

(NASA-CR-167944) ENGINE DYNAMIC ANALYSIS WITH GENERAL NONLINEAR FINITE ELEMENT CODES. N82-33390
 PART 2: BEARING ELEMENT REPRESENTATION. *HCAI*
 OVERALL NUMERICAL CHARACTERISTICS AND BENCHMARKING (Akron Univ., Ohio) 229 p 63/07 Unclas 3517J

1. Report No. NASA CR 167944		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle Engine Dynamic Analysis with General Nonlinear Finite Element-Codes Part-11: Bearing Element Implementation, Overall Numerical Characteristics, and Benchmarking				5. Report Date October 1984	
				6. Performing Organization Code	
7. Author(s) J. Padovan, M. Adams, J. Ferris, I. Zoid, and P. Lam				8. Performing Organization Report No.	
9. Performing Organization Name and Address The University of Akron Department of Mechanical Engineering Akron, OH 44325				10. Work Unit No.	
				11. Contract or Grant No. NSC 3283	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546				13. Type of Report and Period Covered Topical	
				14. Sponsoring Agency Code	
15. Supplementary Notes Technical Monitor: G. C. Chamis, NASA Lewis Research Center Mail Stop 49-6 21000 Brookpark Road Cleveland, OH 44135					
16. Abstract An interactive finite element, developed for turbine engine structures rotor/stator interaction in the first phase of this program, was implanted in a general purpose finite element computer code. Three different numerical integration methods were incorporated as part of the implantation in order to solve the governing structural dynamics equations of the interactive finite element. The general purpose code is used subsequently to predict the structural dynamic response of the rotor/stator coupled structure subjected to unbalance and impulse type excitations. The structural dynamic response predicted includes: (1) Bearing/rotor trajectories, (2) stator trajectory, (3) rotor orbit, and (4) force, velocity and acceleration histories at a given location in the coupled structure. Three-dimensional post-processors have been developed to graphically display the voluminous predicted results in isometric views.					
17. Key Words-(Suggested by Author(s)) Interactive finite element, bearing/rotor/stator, structural dynamics, numerical integration, graphical displays, computer programming, engine structures.				18. Distribution Statement Unclassified, Unlimited	
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages	22. Price*

ENGINE DYNAMIC ANALYSIS WITH GENERAL NONLINEAR
FINITE ELEMENT CODES

PART II
BEARING ELEMENT IMPLEMENTATION, OVERALL
NUMERICAL CHARACTERISTICS AND BENCHMARKING

By

J. Padovan

M. Adams

J. Fertis

I. Zeid

P. Lam

October 1981

Departments of Mechanical and Civil Engineering
The University of Akron
Akron, Ohio 44325

Prepared by
Lewis Research Center
National Aeronautics and Space Administration
Cleveland, Ohio 44135

NASA Grant NSG 3283

ABSTRACT

In an attempt to increase jet engine efficiencies, typically smaller rotor-stator-running clearances are being employed in current and new designs. Because of this and the ever present need to improve maintenance, reliability and structural integrity under various modes of operation, more sophisticated analyses tools are required to realistically model the engine. Due to the wide spread usage of general purpose finite element codes in the open market, this report seeks to adapt the procedure to use in modelling rotor-bearing-stator structure common to the turbine industry. The work outlined in this report covers the second phase of work on a three-phase NASA Lewis sponsored research grant on engine dynamic simulation by developing strategies which enable the use of available finite element codes. The second phase has concentrated on four major objectives, namely: i) Benchmarking the elements developed in Phase 1 by incorporation into a general purpose code (ADINA); ii) Evaluation of the numerical characteristics of finite element type rotor-bearing-stator simulations through the use of various types of currently available as well as specially developed explicit/implicit numerical integration operators; iii) Improving the overall numerical efficiency of the procedure; and, iv) Benchmarking the overall approach in several case studies.

INDEX

Section	Page
1 Introduction	1-3
2 Overall Approach	4-5
3 Governing Finite Element Field Equations	5-10
4 Solution Methodology	10-15
5 ADINA Based Bearing Implant and Associated Graphics Post Processors	15-18
6 Numerical Operational Characteristics	18-23
7 Summary and Conclusions	23-24
8 Acknowledgement	24
9 References	25-26
APPENDIX	
1 Figures	A1-1-45
2 User I/O Information	A2-1-11
3 List of rotor-bearing-station updates to ADINA code	A3-1-95
4 Two-dimensional plot code	A4-1-11
5 Three-dimensional plot code	A5-26
6 Nomenclature	A6-1-2
7 Equation Definition	A7-1

TABLE OF FIGURE CAPTIONS

Figure Number	Page
2.1 Explicit Scheme: Flow of Calculations	A1-2-4
2.2 Implicit Scheme: Flow of Calculations	A1-5-7
4.1 Explicit Scheme	A1-8.
4.2 Implicit Scheme	A1-9.
6.1 Single Bearing Model	A1-10
6.2 Rotor Displacement Newmark Method	A1-11
6.3 Stator Displacement Newmark Method	A1-12
6.4 Rotor Orbit Newmark Method	A1-13
6.5 Rotor Displacement Wilson Method	A1-14
6.6 Stator Displacement Wilson Method	A1-15
6.7 Rotor Orbit Wilson Method	A1-16
6.8 Rotor Displacement Central Difference	A1-17
6.9 Stator Displacement Central Difference	A1-18
6.10 Rotor Orbit Central Difference	A1-19
6.11 Rotor Displacement Central Difference ($ P = 2500 \text{ lbs}$, $ S = 0$)	A1-20
6.12 Stator Displacement ($ P = 2500 \text{ lbs}$, $ S = 0$)	A1-21
6.13 Rotor Orbit ($ P = 2500 \text{ lbs}$, $ S = 0$)	A1-22
6.14 Rotor Displacement ($ P = 2500 \text{ lbs}$, $ S = 1000 \text{ lbs}$)	A1-23
6.15 Stator Displacement ($ P = 2500 \text{ lbs}$, $ S = 1000 \text{ lbs}$)	A1-24
6.16 Rotor Orbit ($ P = 2500 \text{ lbs}$, $ S = 1000 \text{ lbs}$)	A1-25
6.17 Multi bearing Model	A1-26
6.18 Rotor Displacement ($ P = 500 \text{ lbs}$, $ S = 0$)	A1-27

Figure Number	Page
6.19 Stator Displacement ($ P = 500$ lbs, $ S = 0$)	A1-28
6.20 Rotor Orbit ($ P = 500$ lbs, $ S = 0$)	A1-29
6.21 Rotor Displacement ($ P = 1000$ lbs, $ S = 0$)	A1-30
6.22 Stator Displacement ($ P = 1000$ lbs, $ S = 0$)	A1-31
6.23 Rotor Orbit ($ P = 1000$ lbs, $ S = 0$)	A1-32
6.24 Rotor Displacement ($ P = 2500$ lbs, $ S = 0$)	A1-33
6.25 Stator Displacement ($ P = 2500$ lbs, $ S = 0$)	A1-34
6.26 Rotor Orbit ($ P = 2500$ lbs, $ S = 0$)	A1-35
6.27 Rotor Displacement ($ P = 2500$ lbs, $ S = 0$, $\Omega = 750$ rad/sec)	A1-36
6.28 Stator Displacement ($ P = 2500$ lbs, $ S = 0$, $\Omega = 750$ rad/sec)	A1-37
6.29 Rotor Orbit ($ P = 2500$ rad/sec, $ S = 0$, $\Omega = 750$ rad/sec)	A1-38
6.30 Multibearing Model	A1-39
6.31 Rotor Displacement; Point A	A1-40
6.32 Stator Displacement; Point A	A1-41
6.33 Rotor Orbit; Point A	A1-42
6.34 Rotor Displacement; Point B	A1-43
6.35 Stator Displacement; Point B	A1-44
6.36 Rotor Orbit; Point B	A1-45

1. INTRODUCTION

Due to the need to develop increased jet engine efficiencies, typically smaller rotor-stator running clearances are being employed in current and new designs. This together with the ever-present thrust to improve reliability and reduce maintenance costs has generated the need to develop a better understanding of the basic dynamic characteristics of existing new engine configurations.

With the emergence of general purpose (GP) finite element (FE) codes such as ADINA, ANSYS, MARC, NASTRAN, [1,2] the analysis of the dynamics of general structures has been made readily available to structural designers and analysts. This follows from the fact that the overall FE methodology includes numerous important features namely [3,4]:

- 1) Handles hierarchal structural systems with numerous levels and sub-structure;
- 2) Handles large deformation kinematics-kinetics;
- 3) Accommodates general material properties including anisotropy, nonlinear elasticity, plasticity, viscoplasticity, etc.
- 4) Well suited to handle nonlinear boundary conditions as defined by impact, contact, rubbing, and;
- 5) There exist numerous commercially available GP codes with extensive element libraries [1,2].

Because of such capacities, GP FE codes are experiencing a wide spread usage throughout the various aerospace and commercial industries [1,2].

In spite of its far reaching capabilities, the FE procedure has not been extensively employed to analyze rotor-bearing-stator structure. This follows from the lack of elements which can model the interfacial fields generic to gas turbines and in general rotating equipment. Specifically, there is a lack of elements which can properly and efficiently model the interactive force fields originating in roller bearings, squeeze-film dampers, seals and rub/impact events, etc. Together with these shortcomings arises the difficulties associated with the integration of rotor-bearing-stator simulations which, unlike the modal approach say of Adams [5], requires the entire set of simulating equations to be handled directly. While Guyan type reduction schemes [6] can be employed to reduce the number of degrees of freedom of the various linear substructures, the direct integration of the coupled set of simulating equations still requires further work to improve numerical stability as well as computational efficiency. This is a direct outgrowth of the highly nonlinear manner in which various interfacial fields, such as those generated in squeeze-films and rubs, behave.

In the context of the foregoing, the basic emphasis of this three-year study is to develop as well as extend the FE methodology to efficiently handle the transient/steady state response of rotor-bearing-stator structure associated with gas turbine engines. Specially, this includes: (1) the development of specialized elements which enable the characterization of squeeze-film dampers, roller bearings, rubs, impacts, and seals [7]; (2) the benchmarking of the elements through incorporation into a GP code wherein direct integration schemes are used to generate the overall solution; and (3) the evaluation/improvement of the numerical operating characteristics of such simulations.

The thrust of this report is to outline the second year efforts aimed primarily at:

- f) Implanting the squeeze-film damper element into a general purpose FE code (ADINA [8,9]) for testing and evaluation, including such items as:
 - a). Element computational efficiency;
 - b) Adequacy of the perturbational scheme in defining instantaneous stiffness and damping coefficients of fluid film; and,
 - c) Adequacy of difference mesh [7] used in squeeze-film element to model interactive forces.
- ii) Determine running characteristics of direct integration approach of FE-generated rotor-bearing-stator simulation, including:
 - a) Benchmarking scheme;
 - b) Establish/improving numerical efficiency [10,11],
 - c). Establish numerical stability threshold;
 - d) Comparison of explicit [12] vs implicit [13] methodologies of direct integration; and
 - e) Determine extent and source of nonlinearity arising in model.

In the section which follows, detailed discussions are given on the overall implant strategy, the governing field equations, the explicit and implicit integration schemes as well as the results of numerical experiments used to benchmark the operating characteristics of the approach. The appendices include listings of the appropriate coding required to implant the bearing element into the ADINA architecture as well as its associated user instructions. Also included is a listing of complementing graphics post processors enabling various views of rotor trajectories.

2. OVERALL APPROACH

Rotor-squeeze film/bearing-stator simulations are inherently nonlinear. This is an outgrowth of two sources namely:

- i) The inherent behavior characteristic to squeeze films particularly during transient situations, and
- ii) Potential structural interactions wherein either large deformation kinematics or material nonlinearity (plasticity) are excited.

Because of such features, the overall rotor-bearing-stator simulation must be able to incorporate the various sources of nonlinearity. Additionally, to enable efficient solutions in situations wherein various of the model components are linear, the overall simulation scheme must incorporate substructuring capabilities. Furthermore, since transient problems will be considered, such features must be accommodated by the various integration algorithms used to solve the governing model equations.

The central issue of the modelling requirements is the need to identify the separate sources of nonlinearity which may occur. Having done so, such sources can be handled via substructural procedures. Since numerical integration is used to solve the resulting set, the various linear components of the model can all be generated and stored before time stepping occurs. In this context, only the nonlinear substructural components need be updated during integration. After this, the various linear substructural contributions can be assembled into the global scheme. This is the approach taken for this study.

Figures 2-1 and 2-2 illustrate the various aspects associated with the overall flow of control for the foregoing substructural approach. Specifically, Figure 1-1 illustrates the basic flow of calculations for

explicit type schemes. As can be seen three basic loops are preset in the logic flow. These are associated with the linear and nonlinear substructural components as well as the explicit integration process itself. For linear structural components, the substructural stiffness is calculated and globally assembled outside the integration loop. During integration, the reaction forces of such components are obtained simply by post-multiplying the linear stiffness by the appropriate current deflection. This will be discussed in detail in Chapter 4. Such an approach significantly streamlines the calculation flow. For the nonlinear loop, the nodal load due to component reactions must be updated and reassembled during each time step of the explicit integration scheme. This also applies to the bearing group.

For the implicit integration scheme, as seen from Figure 2-2, the overall flow is similar to that of the explicit case but requires the additional calculation of stiffness and damping matrices associated with the linear, nonlinear and bearing components. Again, the linear component stiffness and damping matrices are generated and globally assembled outside the integration loop while the nonlinear and bearing elements are constantly updated within the loop.

3. GOVERNING FINITE ELEMENT FIELD EQUATIONS

Assuming the possibility of large deformation kinematics and kinetics, the Lagrangian type equations of motion take the form [14].

$$\frac{\partial}{\partial a_j} \left(S_{jk} \left(\delta_{ik} + \frac{\partial u_i}{\partial a_k} \right) \right) + \rho_0 F_{0i} = \rho_0 u_{i,tt} \quad (3.1)$$

where a_i ; $i = 1, 2, 3$ are the cartesian coordinates defining the initial

configuration, S_{jk} the 2nd Piola-Kirchhoff stress tensor, δ_{ik} the Kronecker delta, ρ_0 the initial density, u_i the Lagrangian displacement components, F_{oi} the body force and $(\cdot)_{,tt}$ denotes time-differentiation. Based on (3.1), the virtual work principle used to establish the requisite FE formulation is given by [14]

$$\int_R (\delta L_{ij} S_{ij} + \delta u_i \rho_0 (u_{i,tt} + F_{oi})) dv = \int_{\partial R} \delta u_i n_j S_{jk} (\delta_{ik} + \frac{\partial u_i}{\partial a_k}) ds \quad (3.2)$$

where $\delta(\cdot)$ is the virtual operator, ∂R the boundary of R , n_i the normal to ∂R and L_{ij} the Lagrangian strain tensor is defined by [14]

$$L_{ij} = \frac{1}{2} (u_{i,j} + u_{j,i} + u_{k,i} u_{k,j}) \quad (3.3)$$

Employing the usual shape function approximation we have that [3]

$$u(a,t) = [N(a)]Y(t) \quad (3.4)$$

where $[N(a)]$ is the shape function, Y the nodal displacement and

$$u^T = (u_1, u_2, u_3) \quad (3.5)$$

In terms of (3.4), L_{ij} can be recast as [3]

$$L = [[B] + \frac{1}{2} [B_n(Y)][G]]Y \quad (3.6)$$

where $[B]$, $[B_n(Y)]$ and $[G]$ are defined in Appendix A7 and

$$L^T = (L_{11}, L_{22}, L_{33}, L_{12}, L_{23}, L_{31}) \quad (3.7)$$

Substituting (3.4) and (3.6) into (3.2) yields the following FE formulation of the virtual work principle namely

$$\int_R (\delta L^T S + \delta U^T \rho_0 u_{,tt} + \rho_0 F_{o0}) dv = \delta Y^T F \quad (3.8)$$

where

$$S^T = (S_{11}, S_{22}, S_{33}, S_{12}, S_{23}, S_{13}) \quad (3.9)$$

$$\delta Y = \int_{R_e} \delta u_i n_j S_{jk} (\delta_{ik} + \frac{\partial u_i}{\partial a_k}) ds \quad (3.10)$$

$$\begin{aligned} \delta L &= [B^*] \delta Y \\ &= [[B] + [R_H][G]] \delta Y \end{aligned} \quad (3.11)$$

Now, since δY is arbitrary, (3.8) reduces to the following expression for a given e^{th} element namely

$$\int_{R_e} (\rho_0 [N]^T [N] \ddot{Y} + [B^*]^T S + \rho_0 [N] \Gamma_0) dv = F_e \quad (3.12)$$

Note here F_e strictly defines the elements contribution to its attachment nodes. Recasting (3.12) in more convenient matrix form yields the following expression namely

$$[M_e] \ddot{Y} + \int_{R_e} [B^*]^T S dv + f_e = F_e \quad (3.13)$$

where the consistent mass matrix $[M_e]$ and f_e the body force matrix are given by

$$[M_e] = \int_{R_e} \rho_0 [N]^T [N] dv \quad (3.14)$$

$$f_e = \int_{R_e} \rho_0 [N]^T \Gamma_0 dv \quad (3.15)$$

To develop an overall simulation of the rotor-stator support structure, (3.13) must be assembled together with the interactive force representation [7] of the squeeze film damper and roller bearing combinations. This can be achieved in two different ways evolving out of the requirements of explicit or implicit integration.

For the explicit formulation [12], the interacting fluid forces of the squeeze film can be modeled as nodal loads acting on the various appropriate rotor-stator nodes. This type of formulation takes the following assembled form namely

$$[M]\ddot{Y} + \int_R [B^*]^T S dv = - F_b + f + F_{ext} \quad (3.16)$$

where F_{ext} denotes potential externally applied nodal loads, F_b is the interacting squeeze film load, and f the body force is given by

$$f = \int_R \rho_0 [N] f_0 dv \quad (3.17)$$

For this study the loads developed in the squeeze film are simulated in the manner described in Report 1 [7]. Specifically, two alternatives can be employed:

- i) The pressure profiles modelled by the Reynolds simulation are integrated over the entire squeeze film bearing surface to yield a single force vector defining the overall force occurring in the fluid film; or,
- ii) The profiles can be integrated over segments of bearing arc to yield a detailed set of loads defining squeeze film forces, wherein a more detailed analysis admitting deformations of the bearing structure can be accommodated.

For implicit type formulations [13] (3.16) must be expanded in Taylor series retaining solely up to the tangent stiffness and damping expressions. Specifically, if we let

$$Y(t+\Delta t) \sim Y(t) + \Delta Y(t+\Delta t) \quad (3.18)$$

$$\dot{Y}(t+\Delta t) \sim \dot{Y}(t) + \Delta \dot{Y}(t+\Delta t) \quad (3.19)$$

such that

$$||\underline{Y}(t)|| \gg ||\Delta\underline{Y}(t + \Delta t)|| \quad (3.20)$$

$$||\dot{\underline{Y}}(t)|| \gg ||\Delta\dot{\underline{Y}}(t + \Delta t)|| \quad (3.21)$$

then the proper expansions can be taken namely

$$\underline{L}(t + \Delta t) \sim \underline{L}(t) + \Delta\underline{L}(t + \Delta t) \quad (3.22)$$

$$\underline{S}(t + \Delta t) \sim \underline{S}(t) + \Delta\underline{S}(t + \Delta t) \quad (3.23)$$

$$\underline{F}_b(t + \Delta t) \sim \underline{F}_b(t) + \Delta\underline{F}_b(t + \Delta t) \quad (3.24)$$

where

$$\Delta\underline{L}(t + \Delta t) = [B^*(\underline{Y}(t))] \Delta\underline{Y}(t + \Delta t) \quad (3.25)$$

$$\Delta\underline{S}(t + \Delta t) = [D(\underline{Y}(t))] \Delta\underline{L}(t + \Delta t) \quad (3.26)$$

$$\Delta\underline{F}_b(t + \Delta t) = [K_b(\underline{Y}(t))] \Delta\underline{Y}(t + \Delta t) + [C_b(\underline{Y}(t), \dot{\underline{Y}}(t))] \Delta\dot{\underline{Y}}(t + \Delta t) \quad (3.27)$$

such that $[D]$ is the tangent stiffness of the structural partitions of the rotor-stator assembly while $[K_b]$ and $[C_b]$ are the pseudo stiffness and damping matrices developed across the squeeze film [7]. In terms of (3.18 - 3.27), (3.16) can be reduced to the following tangent formulation that is

$$[M]\ddot{\underline{Y}}(t + \Delta t) + [C_b]\Delta\dot{\underline{Y}}(t + \Delta t) + [[K] + [K_b]]\Delta\underline{Y}(t + \Delta t) = -\underline{F}_b(t) + \underline{f}(t + \Delta t) - \int_R [B^*]^T \underline{S} dv \quad (3.28)$$

where

$$[K] = \int_R ([G]^T [S(t)] [G] + [B^*]^T [D(t)] [B^*]) dv \quad (3.29)$$

such that $[S(t)]$ is a matrix of stress defined at time t .

Note for purely elastic structural components, $[D]$ defines the so-called tangent stiffness which is the slope of the stress-strain space. In the case where some structural components may have undergone some plastic deformation, $[D]$ is replaced by $[D_{ep}]$ which is the elastic-plastic counterpart. A good description of the derivation of $[D_{ep}]$ is given by Zienkiewicz [3]. In such situations, (3.28) is replaced by

$$[M]\ddot{Y} + [C_b]\dot{\Delta Y} + [[K_{ep}] + [K_b]]\Delta Y = - F_b(t) + f(t + \Delta t) - \int_R [B^*]^T S|dv \quad (3.30)$$

such that

$$[K_{ep}] = \int_R ([G]^T [S(t)] [G] + [B^*]^T [D_{ep}(t)] [B^*]) dv \quad (3.31)$$

Note for situations wherein purely linear structural components are involved, (3.28) reduces to the form

$$[M]\ddot{Y}(t + \Delta t) + [C_b]\dot{\Delta Y}(t + \Delta t) + [[K_e] + [K_b]]\Delta Y(t + \Delta t) = - F_b(t) + f(t + \Delta t) - \int_R [B]^T S|dv \quad (3.32)$$

where here

$$[K_e] = \int_R [B]^T [D] [B] dv \quad (3.33)$$

For such situations, $[K_e]$ needs to be updated solely at the start of the problem.

4. SOLUTION METHODOLOGY

The main difficulties in the rotor-stator simulations evolve out of the fact that during transient occurrences, the interactive squeeze film forces undergo significant changes in direction and magnitude. This trend

tends to introduce rather strong changes in $[K_b]$ and $[C_b]$ as the calculations proceed from time step to time step. The major result of this is that a perfusion of system harmonics are then introduced in the problem hence reducing the stability threshold of implicit integration schemes. Since explicit schemes tend to be more stable in problems involving a perfusion of system harmonics, as noted earlier, we have adopted the dual approach of employing both explicit and implicit schemes of integration. This enables us to strike a balance between solution efficiency versus numerical stability.

For the current purposes, we have adopted the central difference form of explicit operator [15]. In this context, since [16]

$$\ddot{\underline{Y}} = \frac{1}{(\Delta t)^2} (\underline{Y}(t + \Delta t) - 2\underline{Y}(t) + \underline{Y}(t - \Delta t)) \quad (4.1)$$

the following explicit time stepping scheme can be developed to solve (3.16) namely

$$\underline{Y}(t + \Delta t) = 2\underline{Y}(t) - \underline{Y}(t - \Delta t) + (\Delta t)^2 [M]^{-1} \left(\underline{f} \Big|_t + \underline{F}_{ext} \Big|_t - \underline{F}_b \Big|_t - \int_R [B^*]^T S \Big|_t dv \right) \quad (4.2)$$

Note in (4.2), $\underline{F}_b \Big|_t$ and $\int_R [B^*]^T S \Big|_t dv$ are updated each time step.

For rotor-stator structure involving purely linear media and small deformation kinematics, the following simplification can be employed that is

$$\int_R [B^*]^T S \Big|_t dv = \int_R [B]^T [D] [B] dv \underline{Y}(t) \quad (4.3)$$

where $[D]$ is the material stiffness associated with the typical Hookean constitutive law. In terms of (4.3), (4.2) reduces to the following form

$$\underline{Y}(t + \Delta t) = 2\underline{Y}(t) - \underline{Y}(t - \Delta t) + (\Delta t)^2 [\underline{M}]^{-1} (\underline{f} |_{\underline{t}} + \underline{F}_{\text{ext}} |_{\underline{t}} - \underline{F}_{\text{b}} |_{\underline{t}} - [\underline{K}_{\ell}] \underline{Y}(t)) \quad (4.4)$$

where $[\underline{K}_{\ell}]$ is the constant stiffness matrix defined by

$$[\underline{K}_{\ell}] = \int_R [\underline{B}]^T [\underline{D}] [\underline{B}] dv \quad (4.5)$$

As can be seen from (4.4), the only term requiring updating on the right-hand side is \underline{F}_{b} the interactive squeeze-film force. Hence, apart from requiring small Δt , (4.4) represents an architecturally very streamlined algorithm to employ in rotor-stator simulations.

For situations wherein the rotor-stator structure involves both linear and nonlinear components, the overall calculations can be substructured in this context.....

$$\int_R [\underline{B}^*]^T \underline{S} |_{\underline{t}} dv = \int_{R_N} [\underline{B}^*]^T \underline{S} dv + \int_{R_{\ell}} [\underline{B}]^T [\underline{D}] [\underline{B}] dv \underline{Y}(t) \quad (4.6)$$

where

$$R = R_{\ell} + R_N \quad (4.7)$$

such that R_N and R_{ℓ} are the regions defining the linear and nonlinear component zones respectively. In the context of (4.6), (4.2) can be recast as follows namely...

$$\underline{Y}(t + \Delta t) = 2\underline{Y}(t) - \underline{Y}(t - \Delta t) + (\Delta t)^2 [\underline{M}]^{-1} (\underline{f} |_{\underline{t}} + \underline{F}_{\text{ext}} |_{\underline{t}} - \underline{F}_{\text{b}} |_{\underline{t}} - \int_{R_N} [\underline{B}^*]^T \underline{S} |_{\underline{t}} dv - [\underline{\bar{K}}_{\ell}] \underline{Y}(t)) \quad (4.8)$$

where here

$$[\underline{\bar{K}}_{\ell}] = \int_{R_{\ell}} [\underline{B}]^T [\underline{D}] [\underline{B}] dv \quad (4.9)$$

The overall flow of calculations associated with the foregoing explicit scheme is given in Figure 4.1.

For the implicit scheme, the Wilson θ [16] and Newmark β [17] methods will be employed to integrate the tangent formulation. To start the development of the requisite algorithms, (3.28) is recast in a more appropriate form namely

$$[M]\ddot{Y}(t+\tau) + [C_b]\dot{Y}(t+\tau) + [[K] + [K_b]]\Delta Y(t+\tau) = -F_b(t) + [C_b]\dot{Y}(t) + F(t+\tau) - \int_R [B^*]S|_t dv \quad (4.10)$$

or more simply

$$[M]\ddot{Y}(t+\tau) + [C_b]\dot{Y}(t+\tau) + [K^*]\Delta Y(t+\tau) = F^*(t+\tau) \quad (4.11)$$

where

$$[K^*] = [K] + [K_b] \quad (4.12)$$

$$F^*(t+\tau) = -F_b(t) + [C_b]\dot{Y}(t) + F(t+\tau) - \int_R [B^*]S|_t dv \quad (4.13)$$

Considering the Wilson θ method first, using the usual linear acceleration assumption, it follows that the velocity at $t+\Delta t$ takes the form

$$\dot{Y}(t+\tau) = \dot{Y}(t) + \frac{\tau}{2} (\ddot{Y}(t+\tau) + \ddot{Y}(t)) \quad (4.14)$$

similarly

$$Y(t+\tau) = Y + \tau\dot{Y}(t) + \frac{\tau^2}{6} (\ddot{Y}(t+\tau) + 2\ddot{Y}(t)) \quad (4.15)$$

Employing (4.14, 4.15) in conjunction with (4.10) yields the following expression for $Y(t+\tau)$ namely

$$\Delta Y(t+\tau) = [K_d]^{-1} \{ F^*(t+\tau) + [M] \left(\frac{6}{\tau} \dot{Y}(t) + 2\ddot{Y}(t) + [C_b](2\dot{Y}(t) + \frac{\tau}{2} \ddot{Y}(t)) - [K_d]Y(t) \right) \} \quad (4.16)$$

where

$$[K_d] = \left[\frac{6}{\tau^2} [M] + \frac{3}{\tau} [C_b] + [K] + [K_b] \right] \quad (4.17)$$

Interpolating in the manner of Wilson [16], we let $\tau = \theta \Delta t (\theta > 1)$ so that

$$\ddot{Y}(t+\Delta t) = \left(1 - \frac{1}{\theta}\right) \ddot{Y}(t) + \frac{1}{\theta} \ddot{Y}(t+\tau) \quad (4.18)$$

$$\dot{Y}(t+\Delta t) = \dot{Y}(t) + \frac{\Delta t}{2} (\ddot{Y}(t) + \ddot{Y}(t+\tau)) \quad (4.19)$$

$$Y(t+\Delta t) = Y(t) + \Delta t \dot{Y}(t) + \frac{(\Delta t)^2}{6} (\ddot{Y}(t+\Delta t) + 2\ddot{Y}(t)) \quad (4.20)$$

Note, as per Wilson [16], to yield a stable solution $\theta \geq 1.37$ must be employed. For our purposes, the value of $\theta = 1.4$ is used. In employing equations (4.16) and (4.18 - 4.20), it follows that the dynamic stiffness matrix $[K_d]$ must be updated and reinverted each time either $[K]$, $[K_b]$ or $[C_b]$ is reevaluated.

To establish the requisite algorithms for the Newmark method [17], the following approximations on velocity and displacement are employed namely

$$Y(t+\Delta t) = Y(t) + \Delta t \dot{Y}(t) + \frac{(\Delta t)^2}{2} \ddot{Y}(t) + \alpha (\Delta t)^2 (\ddot{Y}(t+\Delta t) - \ddot{Y}(t)) \quad (4.21)$$

and

$$\dot{Y}(t+\Delta t) = \dot{Y}(t) + \Delta t \ddot{Y}(t) + \delta \Delta t (\ddot{Y}(t+\Delta t) - \ddot{Y}(t)) \quad (4.22)$$

such that for linear situations, it has been found that for unconditional stability, $\delta \geq \frac{1}{2}$ and $\alpha \geq \frac{1}{4} (\delta + \frac{1}{2})^2$. Solving (4.21) and (4.22) for $\ddot{Y}(t+\Delta t)$ and $\dot{Y}(t+\Delta t)$ in terms of $\Delta Y(t+\Delta t)$ yields

$$\ddot{Y}(t+\Delta t) = \ddot{Y}(t) + \frac{1}{\alpha (\Delta t)^2} (\Delta Y(t+\Delta t) - \Delta t \dot{Y}(t) - \frac{(\Delta t)^2}{2} \ddot{Y}(t)) \quad (4.23)$$

and

$$\Delta \ddot{Y}(t+\Delta t) = \Delta t \ddot{Y}(t) + \frac{\delta}{\alpha \Delta t} (\Delta Y(t+\Delta t) - \Delta t \dot{Y}(t) - \frac{(\Delta t)^2}{2} \ddot{Y}(t)) \quad (4.24)$$

To obtain $\Delta Y(t+\Delta t)$, employing (4.23 and 4.24) in conjunction with (4.10) yields

$$\begin{aligned} [M] \left(\ddot{Y}(t) + \frac{1}{\alpha(\Delta t)^2} (\Delta Y(t+\Delta t) - \Delta t \dot{Y}(t) - \frac{(\Delta t)^2}{2} \ddot{Y}(t)) \right) + \\ [C_b] \left(\Delta t \ddot{Y}(t) + \frac{\delta}{\alpha \Delta t} (\Delta Y(t+\Delta t) - \Delta t \dot{Y}(t) - \frac{(\Delta t)^2}{2} \ddot{Y}(t)) \right) + \\ [K^*] \Delta Y(t+\Delta t) = F^*(t+\Delta t) \end{aligned} \quad (4.25)$$

Solving for $\Delta Y(t+\Delta t)$ we have that

$$\begin{aligned} \Delta Y(t+\Delta t) = \\ \left[\frac{1}{\alpha(\Delta t)^2} [M] + \frac{\delta}{\alpha \Delta t} [C_b] + [K^*] \right]^{-1} \left\{ F^*(t+\Delta t) - \right. \\ [M] \left(\ddot{Y}(t) - \frac{1}{\alpha(\Delta t)^2} (\Delta t \dot{Y}(t) + \frac{(\Delta t)^2}{2} \ddot{Y}(t)) - \right. \\ \left. \left. [C_b] (\Delta t \ddot{Y}(t) - \frac{\delta}{\alpha \Delta t} (\Delta t \dot{Y}(t) + \frac{(\Delta t)^2}{2} \ddot{Y}(t))) \right) \right\} \end{aligned} \quad (4.26)$$

As with the Wilson scheme, it follows that the dynamic stiffness appearing in (4.26) must be updated and reinverted each time either $[K]$, $[K_b]$ or $[C_b]$ is reevaluated. This is clearly seen from the overall flow of control depicted in Figure 4.2.

5. ADINA BASED BEARING IMPLANT AND ASSOCIATED GRAPHICS POST PROCESSORS

To benchmark the approach developed sections 2-4 the appropriate general purpose finite element code had to be selected. As noted in the first year report [7], the ADINA code was chosen for this purpose. This follows from the fact that it has the requisite features required for rotor-bearing-stator simulations. These include [7,9]:

- i) Linear and nonlinear substructuring features;
- ii) An extensive element library;
- iii) Capability to handle kinetic, kinematic, and material nonlinearity;
- iv) Explicit and implicit integration loops;
- v) Simplified I/O features;
- vi) Accessible code architecture;
- vii) Extensively benchmarked; and;
- viii) Requisite portability.

In terms of the squeeze film damper element noted in Report 1 [7], extensive modifications were introduced into the ADINA code. These modifications were made general enough so as to handle rotor-bearing-stator simulations involving any number of rotors and associated squeeze film damper elements. Recalling Chapter 4, the program has two available solution procedural loops, namely either explicit [12] or implicit [13].

Since extensive amounts of data are generated during a typical run, graphics post processors have also been developed to simplify output evaluation. These include both 2-D as well as 3-D plotter schemes. The 2-D processor enables the plotting of:—

- i) Bearing/rotor trajectories at a given station;
- ii) Stator trajectories;
- iii) Clearance histories at a given bearing station; as well as,
- iv) Force, velocity and acceleration histories at given bearing stations.

The 3-D processor enables the plotting of isometric views of the rotor trajectories.

For convenience, Appendices 2-5 give detailed discussions and listings of the ADINA modifications and graphics post processors. Included

are extensive instructions on user I/O of both the ADINA Bearing Code and the associated graphics routines. Specifically:

- i) Appendix 2: I/O instruction on modified ADINA code enabling rotor-bearing-stator simulations;
- ii) Appendix 3: Code listing of ADINA modifications;
- iii) Appendix 4: Code listing and I/O instructions for 2-D plot routine;
- iv) Appendix 5: Code listing and I/O instructions for 3-D plot routine.

Note the bearing modifications incorporated into ADINA were developed so as not to disturb the I/O flow. Hence, apart from required bearing data (film properties, clearances, etc.), the standard ADINA I/O remains intact [8]. For specific details, see Appendices 2-5.

6. NUMERICAL OPERATIONAL CHARACTERISTICS

As noted earlier, this section will determine the numerical operational characteristics of the direct integration approach of FE generated rotor-bearing-stator simulations. This will include such items as:

- a) Benchmarking;
- b) Comparison of explicit vs. implicit methodologies of direct integration; and
- c) Demonstration problems.

6.1. Benchmarking

To benchmark the overall procedure, a simple lumped parameter direct integration scheme was developed... [7] This approach was used to check the accuracy of the FE generated scheme involving the ADINA "implant". As the first example of such benchmarking, consider the system defined in Fig. (6.1). The material and geometric properties associated with the dual ported squeeze film damper bearing employed in this and the following sample problems are defined by

Nominal diameter = 6 inches; nominal length = 1.2 inches;

clearance = 1. inch; viscosity = $.1 \times 10^{-5}$;

film rupture pressure = 15 psia;

$\theta_1 = 90^\circ$; $\theta_2 = 270^\circ$; $P_1 = 15$ psi; $P_2 = 55$ psi.

Note as with current practice, the bearing used in the simulation has no centering spring. In terms of this system, Figs. (6.2-6.10) illustrate various aspects of the validation. Specifically, the sets of results given by Figs. (6.2-6.4), (6.5-6.7), and (6.8-6.10) define checks involving respectively the Newmark [18], Wilson [17], and central difference [16]

operators. In this way, both the implicit and explicit schemes are treated. In each of the comparisons, three different aspects of rotor-bearing-stator behavior are depicted specifically:

- a) Rotor displacement trajectories;
- b) Stator displacement trajectories; and,
- c) Relative rotor orbit.

For the present purposes, while extensive benchmarking was undertaken, for convenience, only the case of mild load imbalance is depicted. Note, due to the complexity of the various trajectories, comparisons between the FE and benchmarking schemes [7] are given solely for the relative rotor orbits. These are shown in Figures (6.4), (6.7), and (6.10). As can be seen, for the given Δt steps chosen, good comparisons were obtained by both the implicit and explicit schemes. Such benchmarking was obtained over a wide range of rotor speeds and load imbalance levels. In all cases, good accuracy was yielded.

Similar benchmarking was also performed for multibearing-rotor-stator simulations. Note, so long as Δt was kept small, good accuracy was obtained over a wide range of system variables.

6.2 Comparison of Explicit - vs. - Implicit Methodologies of Direct Integration

For problems involving few degrees of freedom, it was found that for a given accuracy, both the implicit and explicit schemes required about the same overall computational times. This follows from the fact that for "small problems" the architectural overhead associated with the implant strategy programming dominates over the relative algorithmic efficiency. Note for transients initiated by rather severe imbalance loads, it was found that the implicit scheme proved to be more sensitive to the

choice of time step size. Interestingly, such sensitivities were found to occur for problems with small as well as large numbers of degrees of freedom. After performing several parametric studies, it was found that during the course of a typical transient, particularly involving a severe loading, the tangent properties of the fluid film undergo major changes. Because of this, a perfusion of system harmonics are introduced into the transient response as the tangent fluid film properties vary. Note, this behavior is intrinsic to the fluid film and hence, is independent of the number of degrees of freedom of the rotor-stator model. Such properties tend to reduce the stability threshold of the implicit scheme which is best employed when only a few harmonics are excited. For problems involving large numbers of degrees of freedom, the sensitivity of the implicit scheme coupled with the required continuous updating and inversions of the dynamic stiffness tends to reduce the running efficiency of the procedure.

In contrast, the central difference approach tends to be less sensitive to such tangent property fluctuations. This is true so long as the resulting family of harmonics is bound by the spectral characteristics of the rotor-stator system. Because of this, similar Δt can be employed by the implicit and explicit scheme. In this context, the explicit scheme is somewhat more computationally efficient than the implicit approach. This follows from the fact that no continuous inversion is required by the explicit scheme.

6.3 Demonstration Problems

To demonstrate various of the capabilities of the ADINA implant strategy, the results of several example problems will be considered. This will involve single and multiple bearing problems with various types of load

imbalance histories, impact events, and rotor speeds. For example, in terms of the single bearing system given in Fig. (6.1), Figs. (6.11 - 6.13), illustrate various aspects of the response histories to an imbalance load which is applied as a ramp function in time. As this loading is more severe than that applied in Figs. (6.2 - 6.10), the rotor tends to fill its clearance circle as seen in Fig. (6.13).

If the same system is subject to a unidirectional impulse, as might be expected from a rough landing, Figs. (6.14 - 6.16) illustrate the associated response history. By comparing Figs. (6.11, 6.12) with (6.14, 6.15), the effects of the impulse can be clearly seen from the ovalizing of the rotor and stator trajectories. This is directly due to the directional characteristics of the impulse load. Note, comparing Fig. (6.13) with (6.16), we see that the rotor orbits are essentially unchanged by the presents of the shock load. Hence, it follows that the squeeze film damper has essentially no effect on mitigating the worst aspects of unidirectional shocks when large load imbalances are involved...

For the current problem this is a direct outgrowth of the fact that when large orbits occur, the oil film being thin tends to have very high tangent stiffness properties. Because of this, the squeeze film damper does little to filter or dampen the exciting load. In this context, such damper bearing systems will have only a small effect on mitigating the severity of maneuvering and landing loads on rough running engines..

The next series of examples pertains to the rotor-bearing-stator system defined in Fig. (6.17). For this system only one squeeze film damper is employed. To simulate the shafting, the beam elements available in the ADINA system are utilized. Note, the mass effects are handled via the lumped parameter approach. The series of figures given by (6.18 - 6.20), (6.21 - 6.23) and (6.24 - 6.26) respectively illustrate the effects

of increasing the severity of loading on the dynamic response of multi-bearing problems. As can be seen by examining the progression of figures defining the rotor/stator displacements and the rotor orbit, increasingly stiffer squeeze film damper responses are excited. This is clearly seen from the wide rotor orbits which occur. Note, as the load is further increased, the rotor stator trajectories become "locked in".

The next series of figures namely-(6.27 - 6.29), illustrate the effects of suddenly applied load imbalances. Since the rotor speed considered is high, only a small portion of the clearance circle is filled. This follows from the fact that due to the rather rapid changes in orientation of the exciting load, the fluid does not have sufficient time to recirculate. Because of this the rotor is supported by severe velocity gradients which are only in a close neighborhood. Hence, the pressure gradient generated by the inlet and outlet ports of the squeeze film device causes the rotor to settle in the direction of the low pressure port. Note, as the rotor speed is decreased, increasingly larger rotor orbits occur. Similar trends occur as the level of imbalance is increased.

As a last example, consider the system defined in Fig. 6.30. Based on this configuration, Figs. (6.31 - 6.36) depict various aspects of the response of the system configured with two squeeze film dampers. Similar to the previous problems, the exciting forces involve imbalance loads which are applied in a ramp fashion in time. As can be seen from the results depicted for the lightly and heavily loaded squeeze film damper bearings, the orbits are respectively small and large.

Note, based on numerous parametric studies involving systems similar to the foregoing, it was found that all the time integration schemes considered were stable for situations wherein the fluid underwent only moderate

changes in stiffness during the overall cycle. Note, while a perfusion of harmonics is introduced by even moderate changes in stiffness, so long as the resulting spectra are strongly bound by the frequency envelop of the dominate system frequencies, spurious energy flow to higher order modes is insignificant. Specifically, for the implicit scheme, if the choice of time-step size is gauged to the dominate higher order system frequencies, then the introduction of lower order spectra by the squeeze film has little effect on numerical stability. In contrast, if strong stiffness modulations occur, then significant amounts of energy flow are introduced in the ever shifting higher order modes. This leads to solution instabilities unless smaller Δt are introduced. In this context, as noted earlier, since the explicit approach tends to be more efficient for a given Δt , for problems involving strong vs. weak imbalance loads, its use is advocated over the implicit scheme.

7. SUMMARY

The main thrust of this report has been to outline the second year results of the Engine Dynamic Analysis with General Nonlinear Finite Element Codes Grant. This has included two main areas namely:

- i) Implanting the squeeze-film damper element into a general purpose FE code for testing and evaluation and;
- ii) Determining the numerical characteristics of the FE generated rotor-bearing-stator simulation scheme.

Due to the generality of the approach taken, the overall implant strategy and associated algorithmic schemes has several important features namely:

- i) Handles hierarchal rotor-bearing-stator system involving the possibility of series and parallel substructure; hence, series and spool engine configurations can be handled;

- ii) Handles large deformation-kinematics and kinetics;
- iii) Implant architecture is well suited to handle the possibility of rubbing and contact;
- iv) Due to the use of the ADINA code, a wide assortment of element and material types can be used in a simulation;
- v) To enable the incorporation of new time stepping algorithms, both an explicit and implicit formulation has been incorporated in the implant and;
- vi) 2-D and 3-D-graphics plotter schemes have been developed to enable the post processing of response data; this includes;
 - (a) Bearing/Rotor trajectories;
 - (b) Stator trajectory;
 - (c) Rotor Orbit;
 - (d) Force velocity and acceleration histories at a given location and finally;
 - (e) The 3-D processor enables the plotting of isometric views of the rotor shape at anytime.

8. ACKNOWLEDGEMENT

The authors gratefully acknowledge the many fruitful discussions with Chris Chamis of NASA Lewis which made this effort possible.

9. REFERENCES —

1. Noor, A., "Survey of Computer Programs for Solution of Nonlinear Structural and Solid-Mechanics Programs," *Comp. Struct.* 13, 425-465 (1981).
2. Chang, T. Y. and Padovan, J., "General Purpose Finite Element Programs," *Structural Mechanics Software Series, Vol. III*, edited by Perrone, N. and Pilkey, W., University Press of Virginia, Charlottesville (1980).
3. Zienkiewicz, O. C., "The Finite Element Method," McGraw-Hill Book Co., New York (1977).
4. Bathe, K. J. and Wilson, E. I., "Numerical Methods in Finite Element Analysis," Prentice Hall, Englewood Cliffs, New Jersey (1976).
5. Adams, M. L., "Nonlinear Dynamics of Multi-Bearing Flexible Rotors," *Jr. Sound and Vibration*, 71, 129-144 (1980).
6. Guyan, R. J., "Reduction of Stiffness and Mass Matrices," *AIAA Jr.*, 3, 380 (1965).
7. Adams, M. L., Padovan, J. and Fertis, D. G., "Engine Dynamic Analysis with General Nonlinear Finite Element Codes, Part 1: Overall Approach and Development of Bearing Damper Element," ASME Paper No. 81-FT-151. Also, "Finite Element for Rotor/Stator Interactive Forces in General Engine Dynamic Simulation, Part 1: Development of Bearing Damper Element," NASA CR-165214 (1981).
8. Bathe, K. J., "ADINA - A Finite Program for Automatic Dynamic Incremental Nonlinear Analysis," Report 82448-1, Acoustics Lab, Mechanical Engineering Dept., MIT, May 1976 (revised 1977).
9. Padovan, J., Chang, T. Y., "Evaluation of ADINA: Part II, Operating Characteristics," Report No. AUÉ 802, University of Akron, prepared for ONR under Contract N0014-78-C-0691, (1980).
10. Padovan, J., "Self-Adaptive Incremental Newton-Raphson Algorithms," *Symposium on Computational Methods in Nonlinear Structural and Solid Mechanics*, Washington D.C., also NASA CP-2147 (1980).
11. Padovan, J., Tovichakchaikul, S., "Self-Adaptive Predictor-Corrector Algorithms for Static Nonlinear Structural Analysis," *Comp. and Struct.* (in press), also NASA CR-165410 (1981).
12. Belytschko, T., "A Survey of Numerical Methods and Computer Programs for Dynamic Structural Analysis," *Nucl. Eng. and Design*, 37, 23-34 (1976).

13. Hughes, T. J. R., "Stability, Convergence and Growth and Decay of Energy of the Average Acceleration Method in Nonlinear Structural Dynamics," Jr. Comp. and Struct., 6, 313-324 (1976).
14. Fung, Y. C., "Foundations of Solid Mechanics," Prentice-Hall, Inc., Englewood Cliffs, New Jersey (1965).
15. Archer, J. S., "Consistent Mass Matrix for Distributed Mass Systems," Proc. Am. Soc. Civil Engrs., 89, 161-178 (1903).
16. Hildebrand, F. B., "Finite-Difference Equations and Simulations," Prentice-Hall, Inc., Englewood Cliffs, New Jersey (1968).
17. Wilson, E. L., "A Computer Program for the Dynamic Stress Analysis of Underground Structures," Report No. SESM68-1, Department of Civil Engineering, University of California, Berkeley (1968).
18. Newmark, N. M., "A Method of Computation for Structural Dynamics," J. Eng. Mech. Div., ASCE, 85, EM3, 67-94 (1959).

APPENDIX 1: FIGURES

Figure 2.1 EXPLICIT SCHEME: FLOW OF CALCULATIONS - -

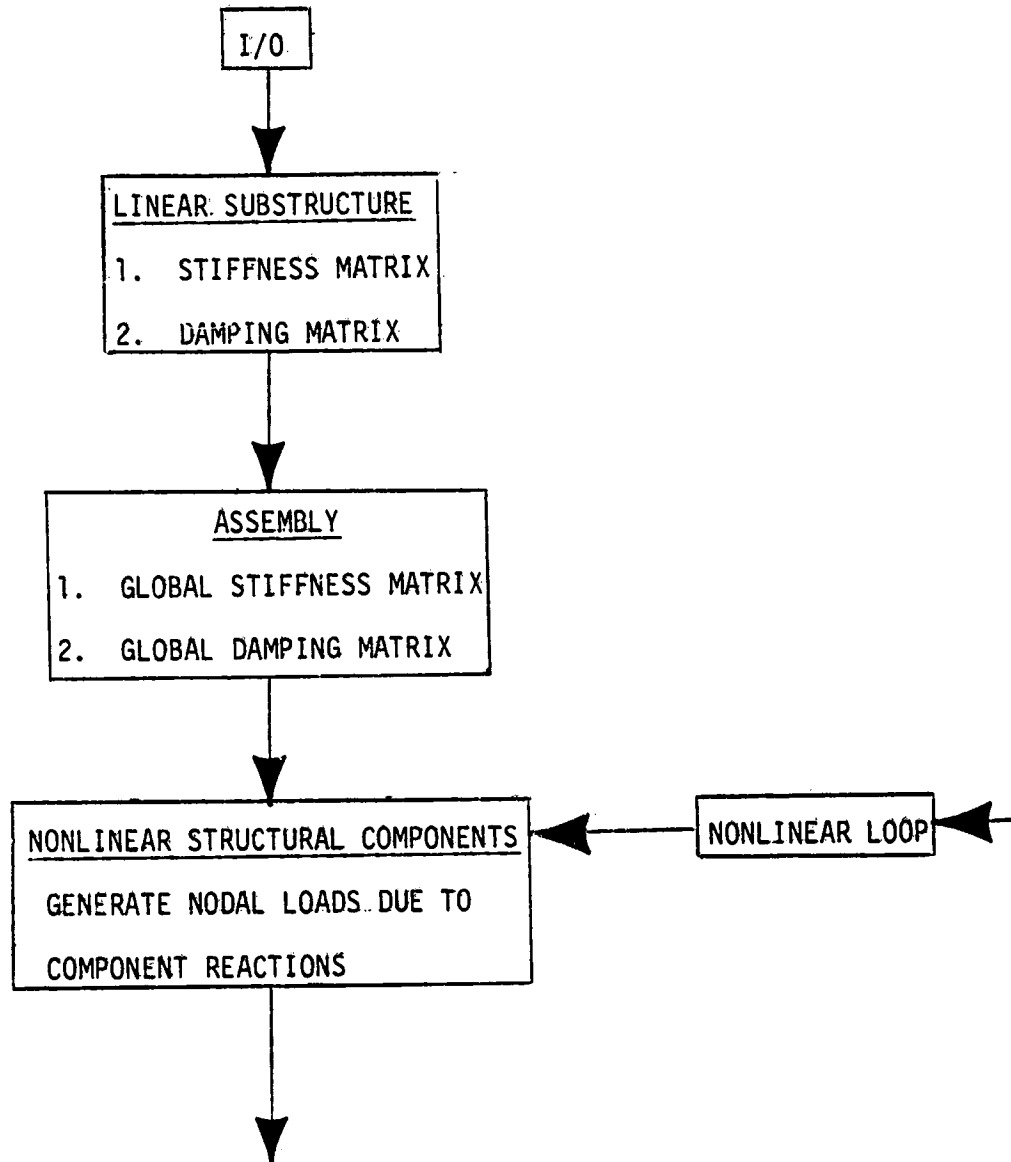


Figure 2.1 EXPLICIT SCHEME; FLOW OF CALCULATIONS (continued)

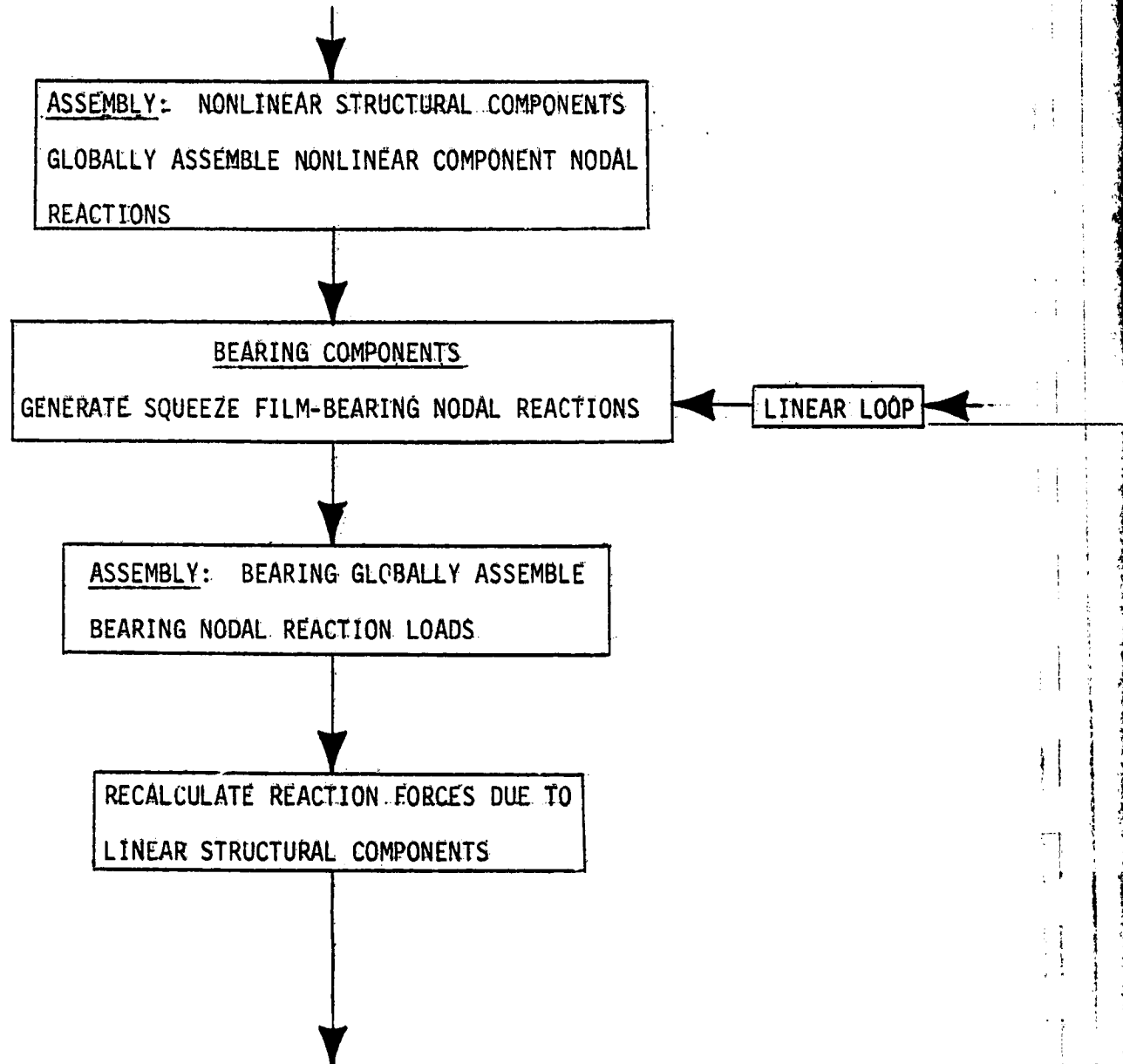


Figure 2.1 EXPLICIT SCHEME; FLOW OF CALCULATIONS (continued)

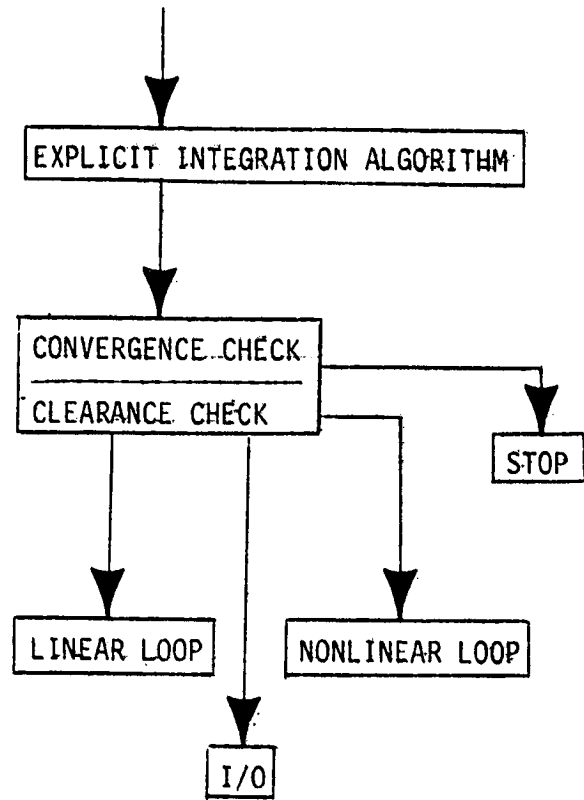


Figure 2.2 IMPLICIT SCHEME; FLOW OF CALCULATIONS

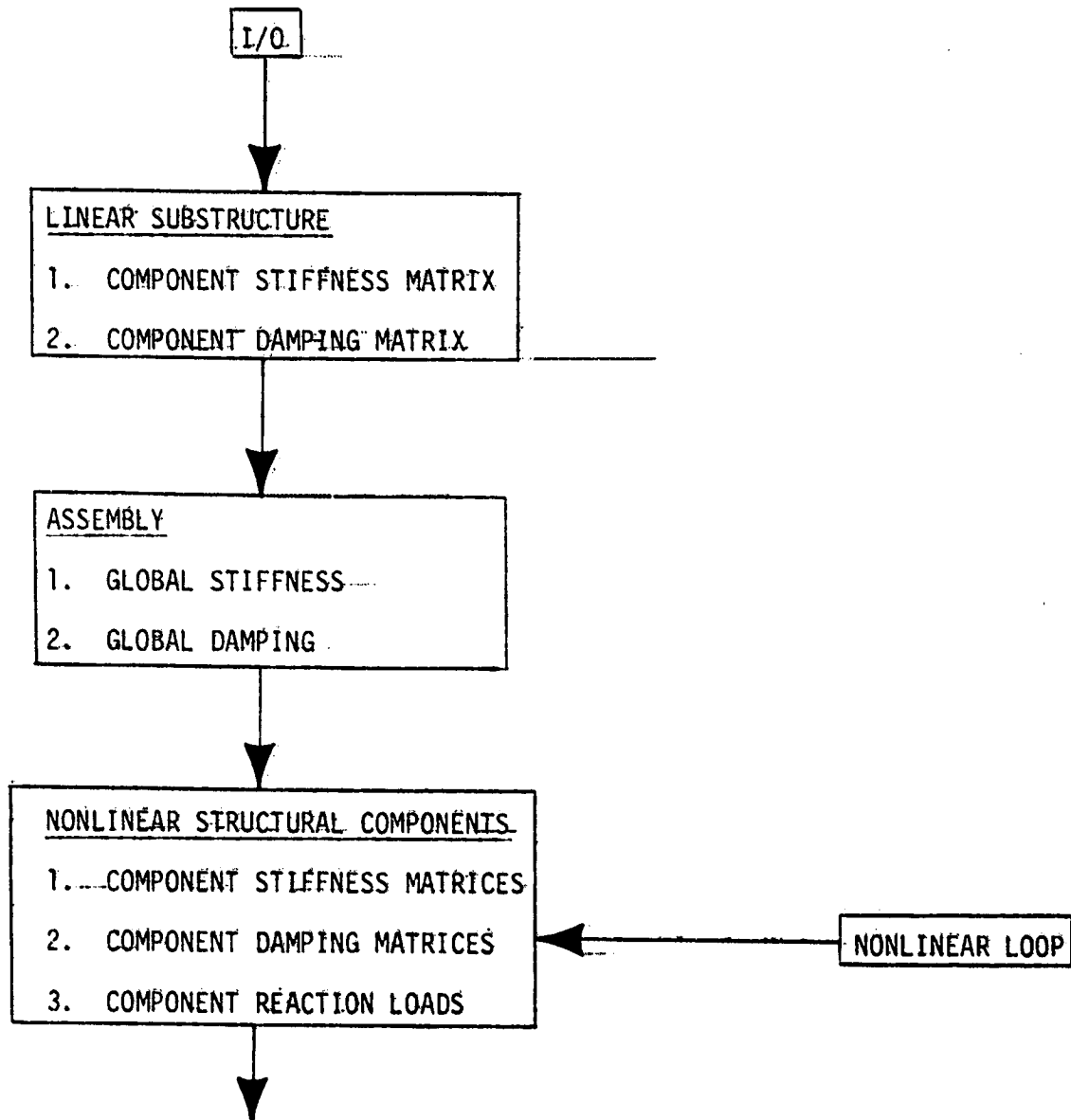


Figure 2.2 IMPLICIT SCHEME; FLOW OF CALCULATIONS (continued)

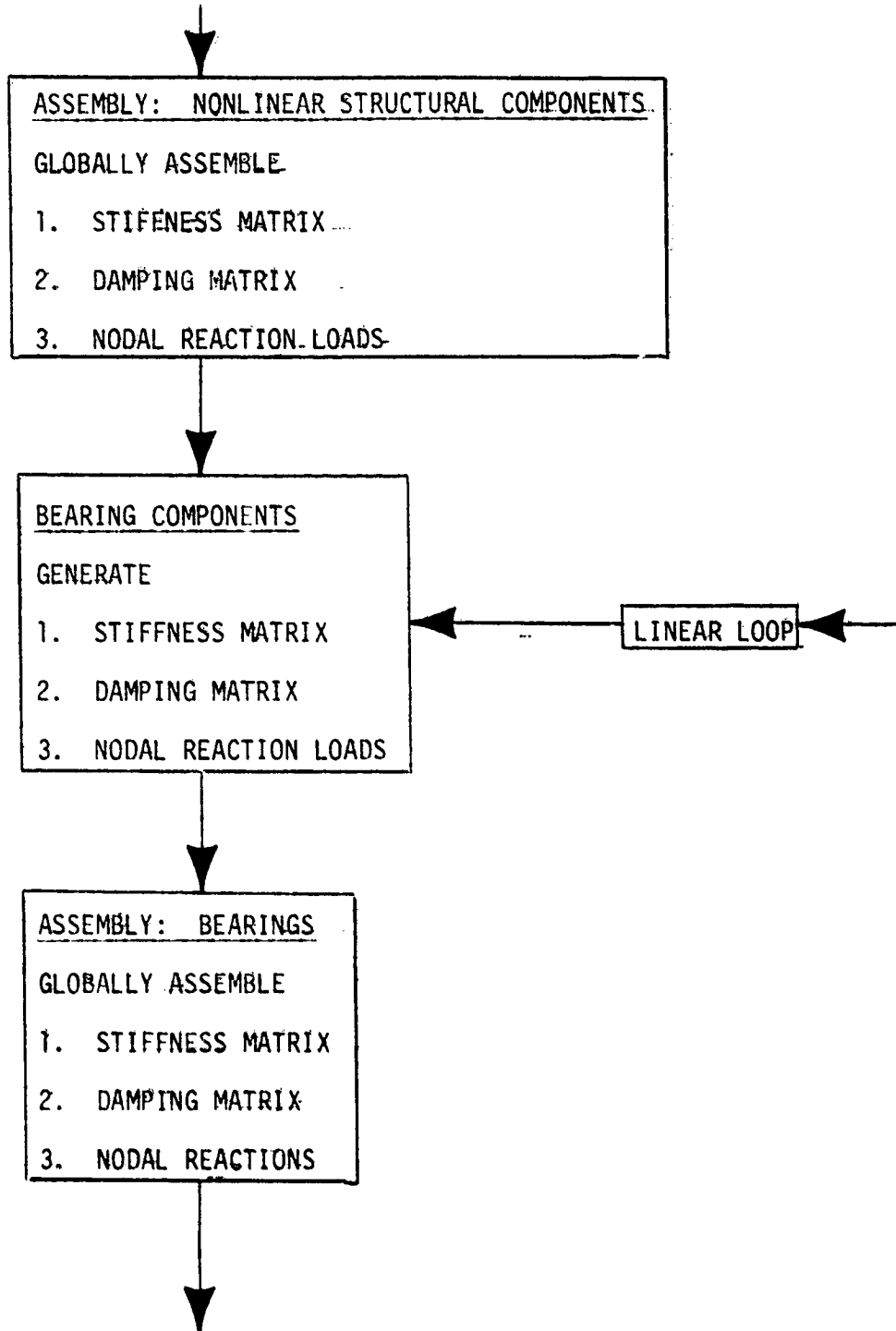


Figure 2.2. IMPLICIT SCHEME; FLOW OF CALCULATIONS (continued)

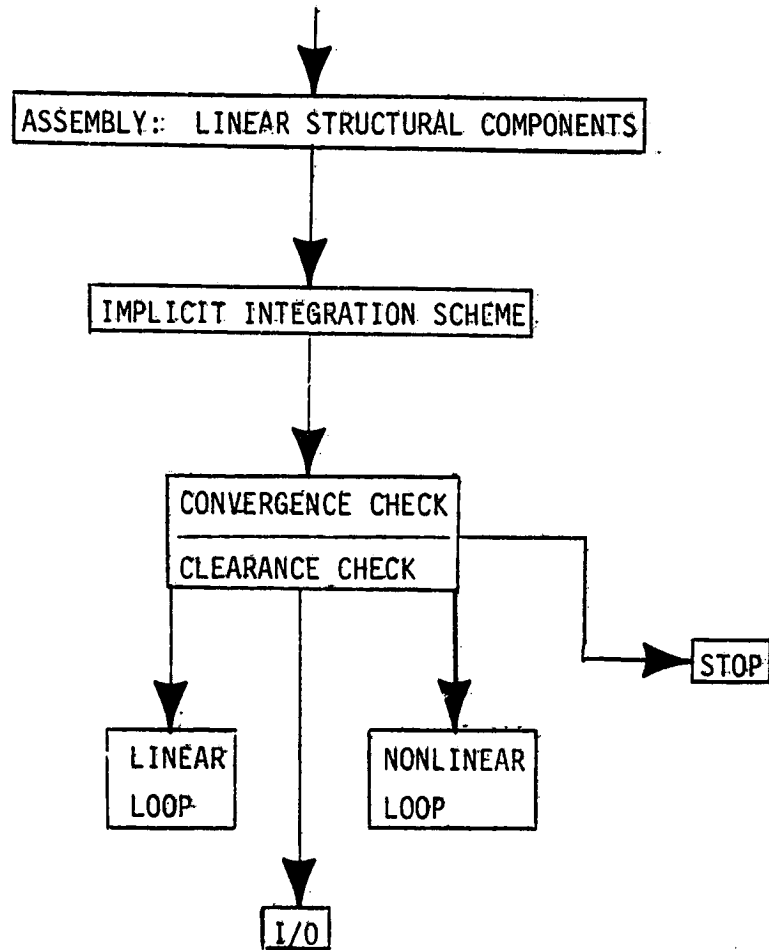


Figure 4.1. EXPLICIT SCHEME

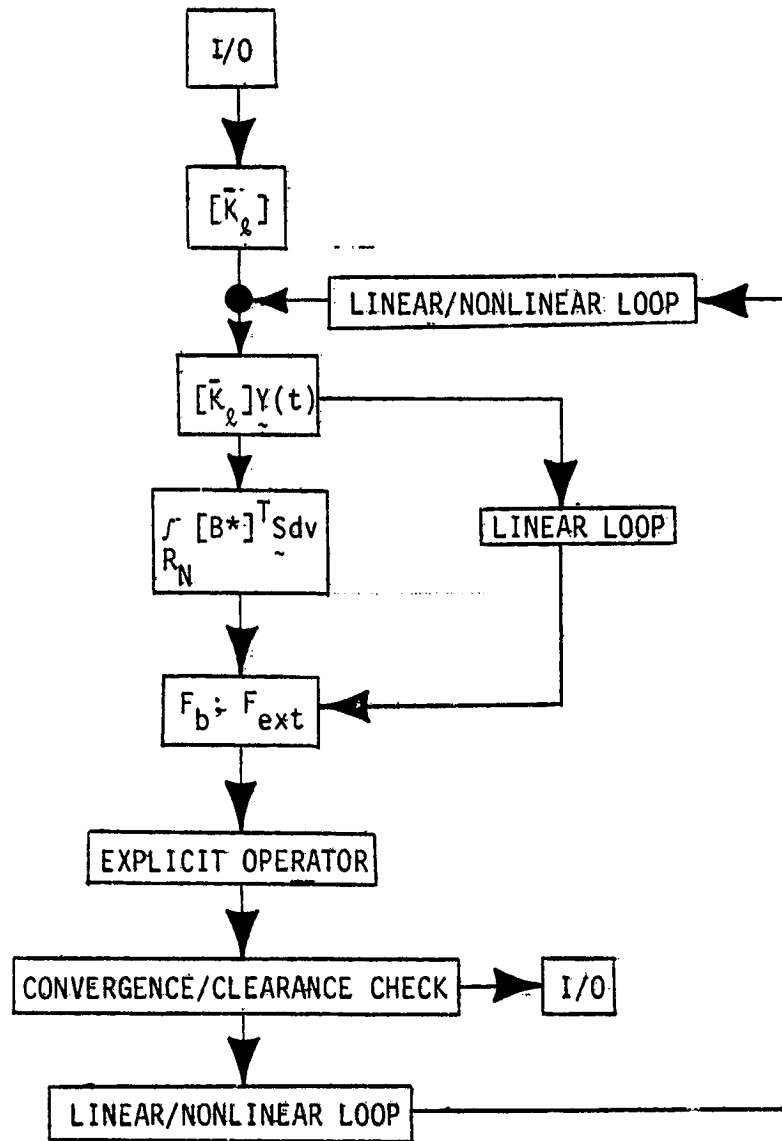
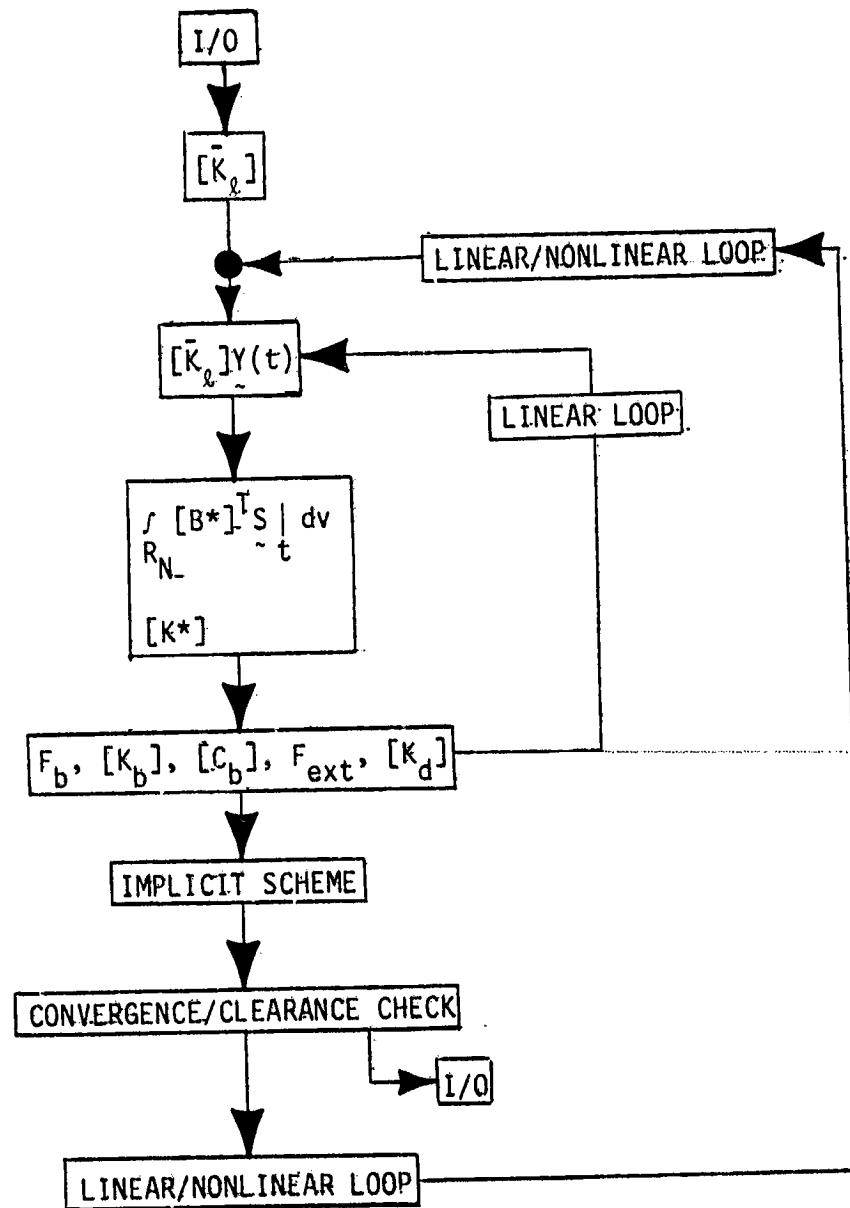


Figure 4.2 IMPLICIT SCHEME



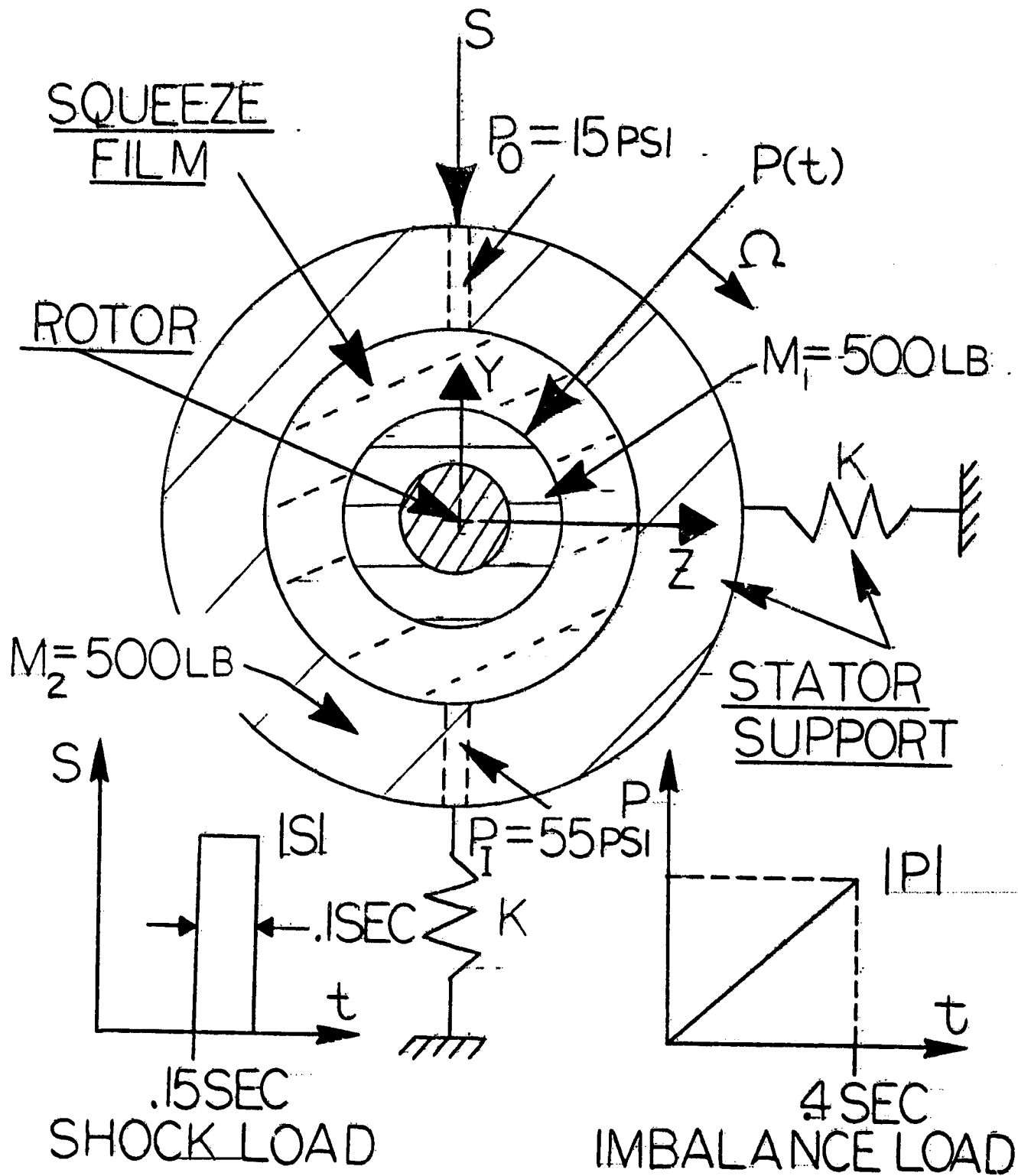


FIG. 6.1 SINGLE BEARING MODEL

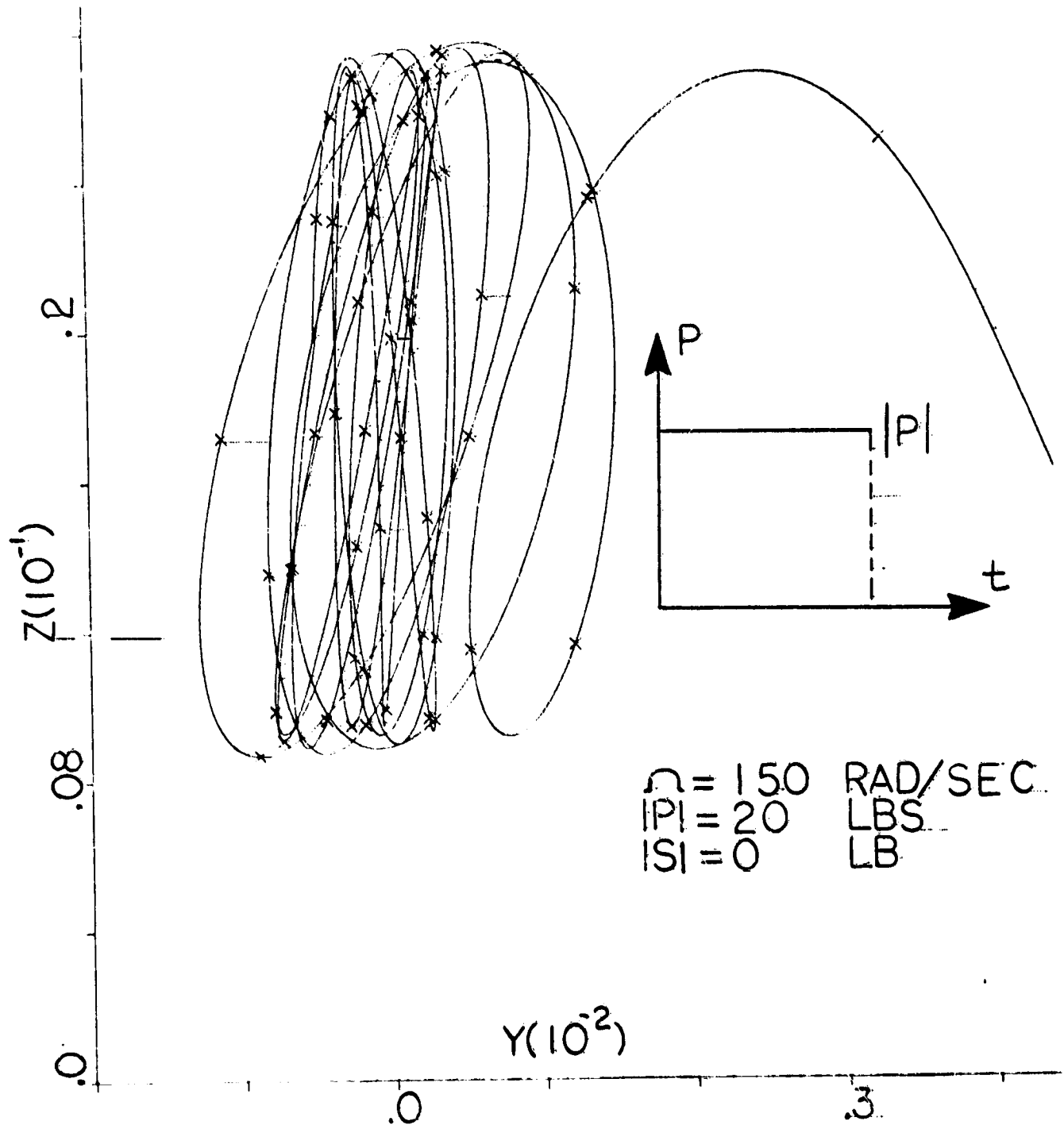


FIGURE 6.2 ROTOR DISPLACEMENT
NEWMARK METHOD
(SEE FIG. 6.1)

ORIGINAL PAGE IS
OF POOR QUALITY

$\omega = 150$ RAD/SEC
 $|P| = 20$ LBS
 $|S| = 0$ LB

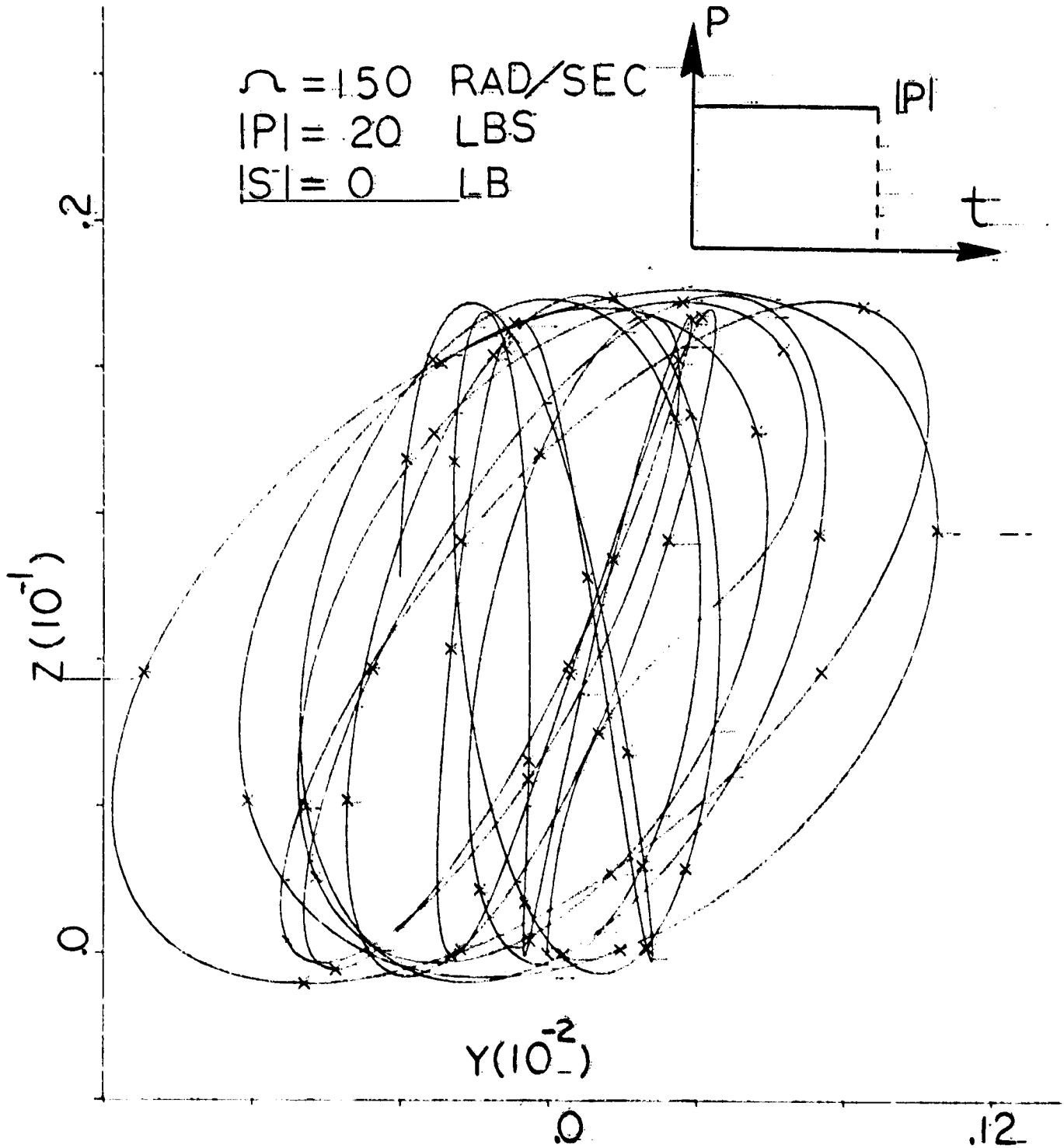
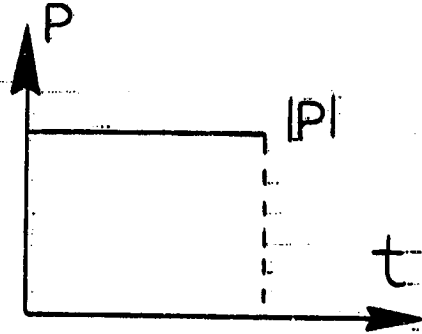


FIGURE 6.3 STATOR DISPLACEMENT
 NEWMARK METHOD
 (SEE FIG. 6.1)

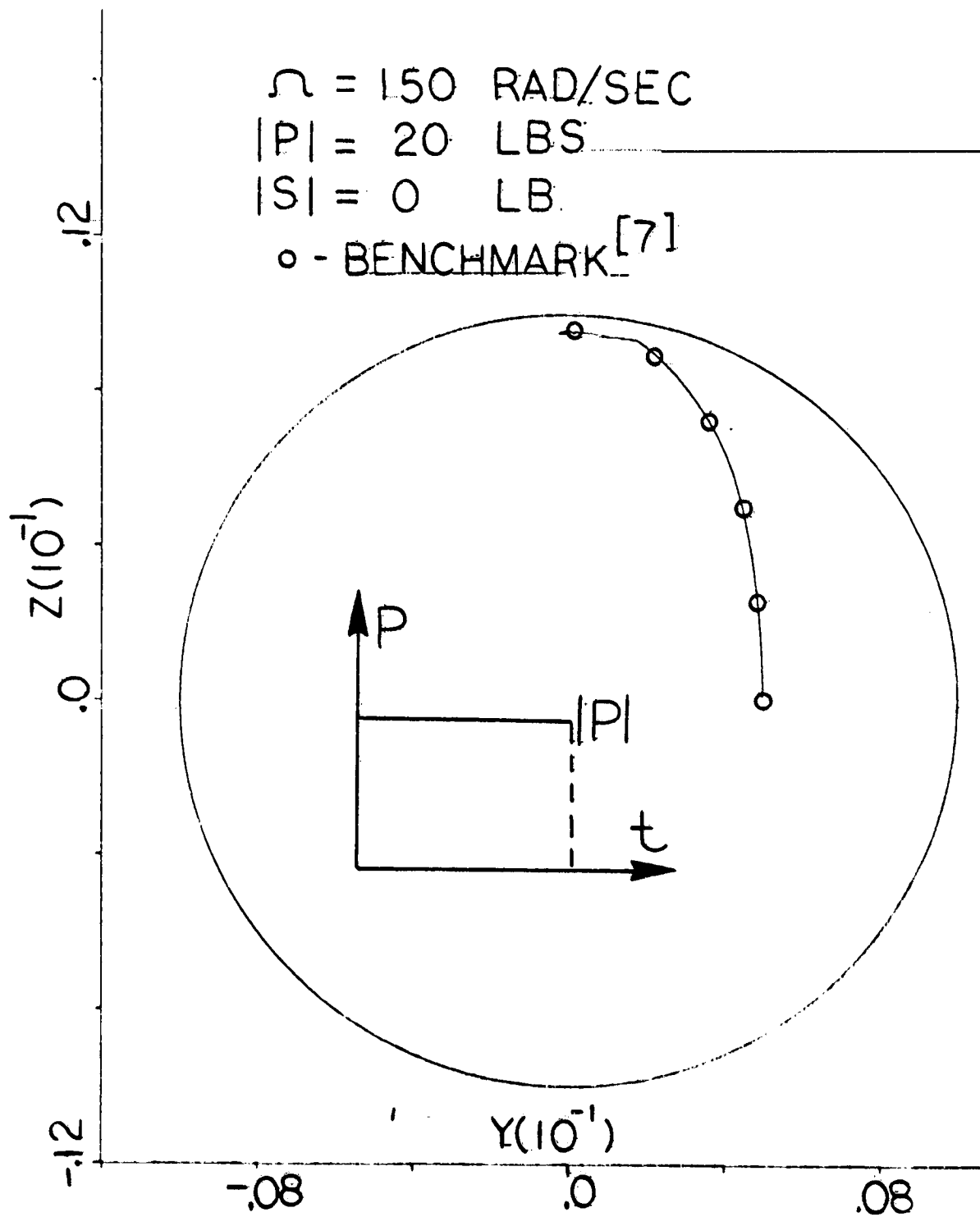


FIGURE 6.4 ROTOR ORBIT
NEWMARK METHOD
(SEE FIG. 6.1)

ORIGINAL PAGE IS
OF POOR QUALITY

A7-14

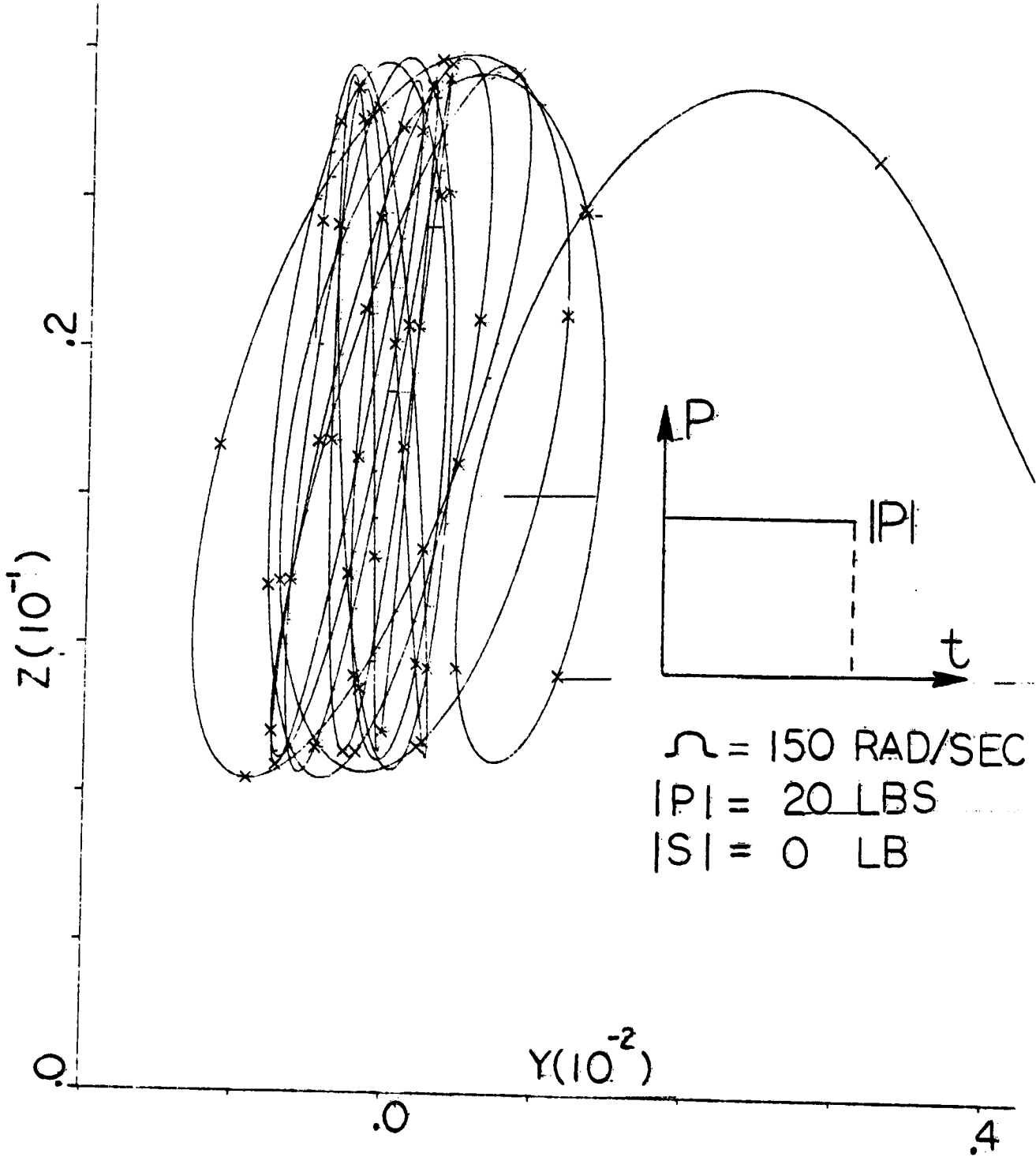


FIGURE 6.5 ROTOR DISPLACEMENT
WILSON METHOD
(SEE FIG. 6.1)

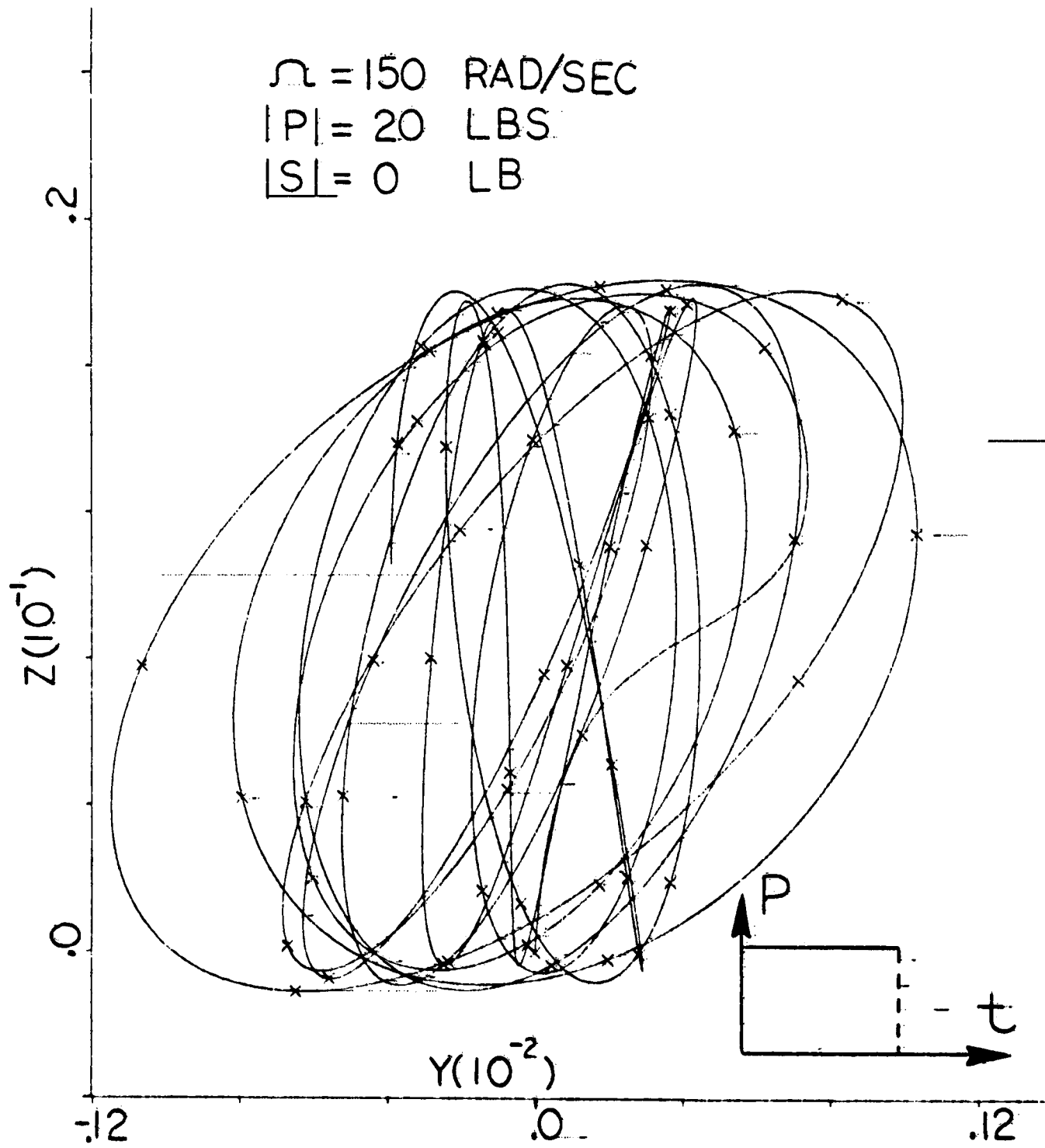


FIGURE 6.6 STATOR DISPLACEMENT
WILSON METHOD
(SEE FIG.6.1)

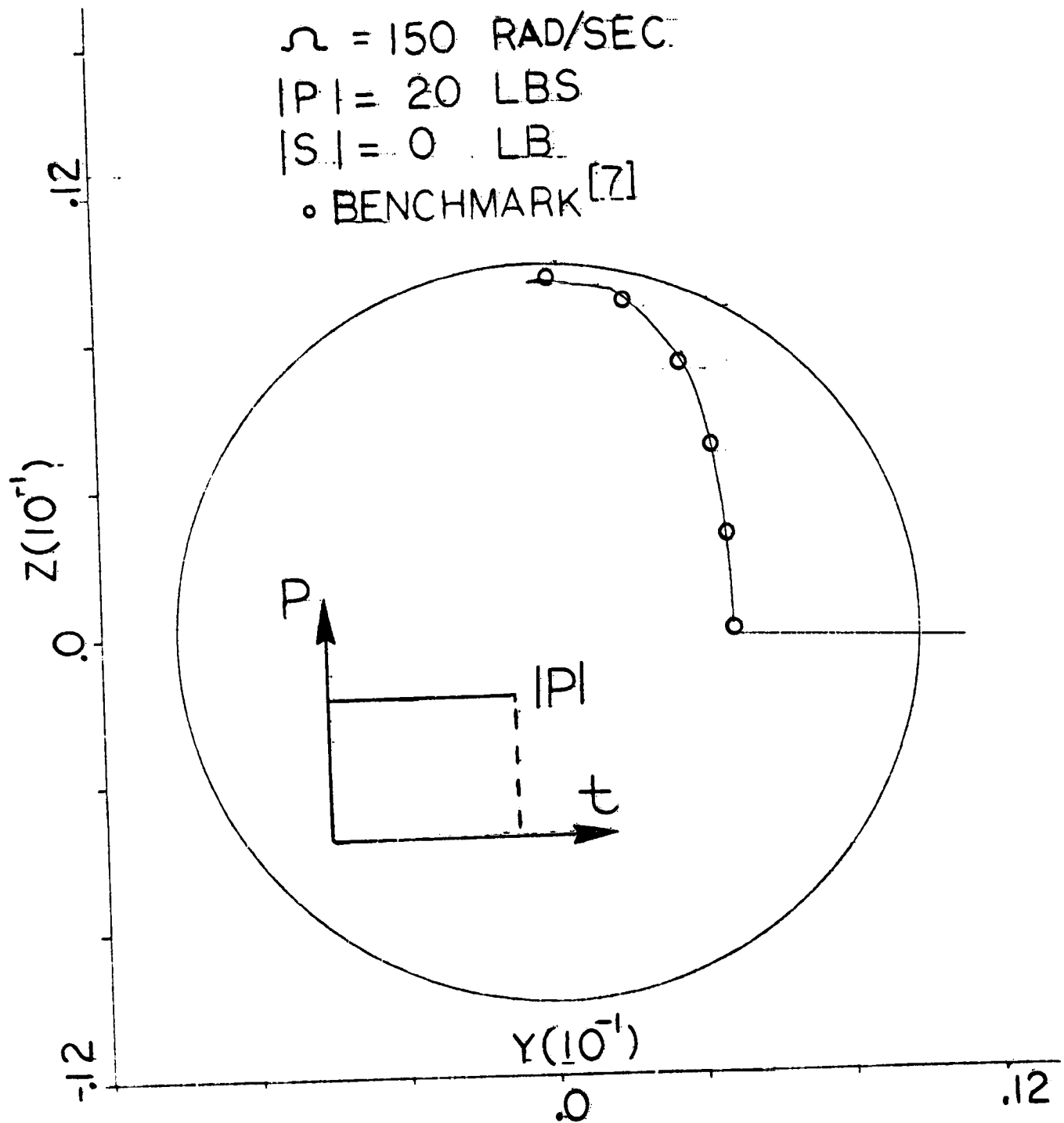


FIGURE 6.7 ROTOR ORBIT
WILSON METHOD
(SEE FIG. 6.1)

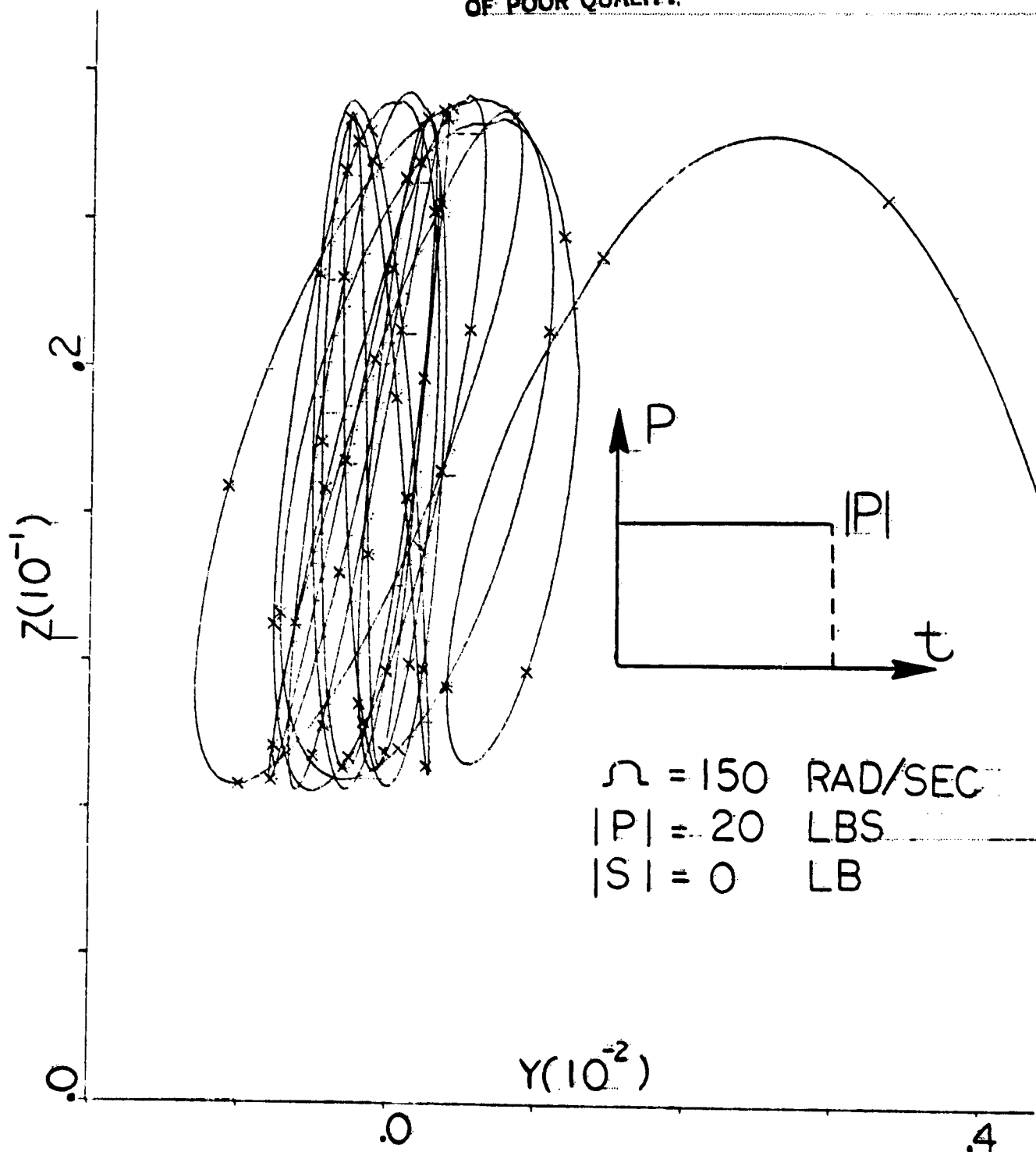


FIGURE 6.8 ROTOR DISPLACEMENT
CENTRAL DIFFERENCE
(SEE FIG. 6.1)

ORIGINAL PAGE IS
OF POOR QUALITY

AT-18

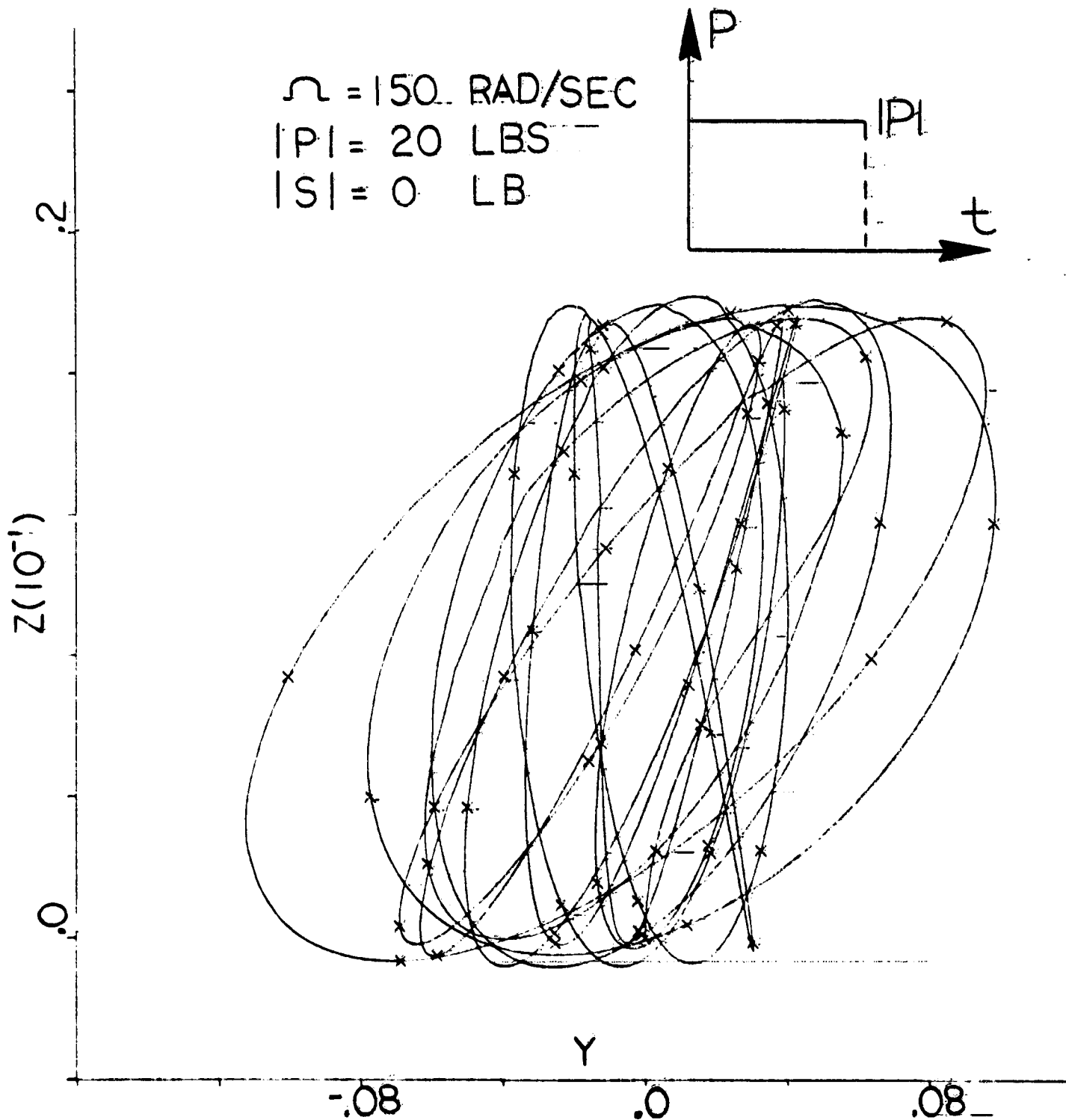


FIGURE 6.9

STATOR DISPLACEMENT
CENTRAL DIFFERENCE
(SEE FIG. 6.1)

$$\Omega = 150 \text{ RAD/SEC}$$

$$|P| = 20 \text{ LBS}$$

$$|S| = 0 \text{ LB}$$

○ BENCHMARK [7]

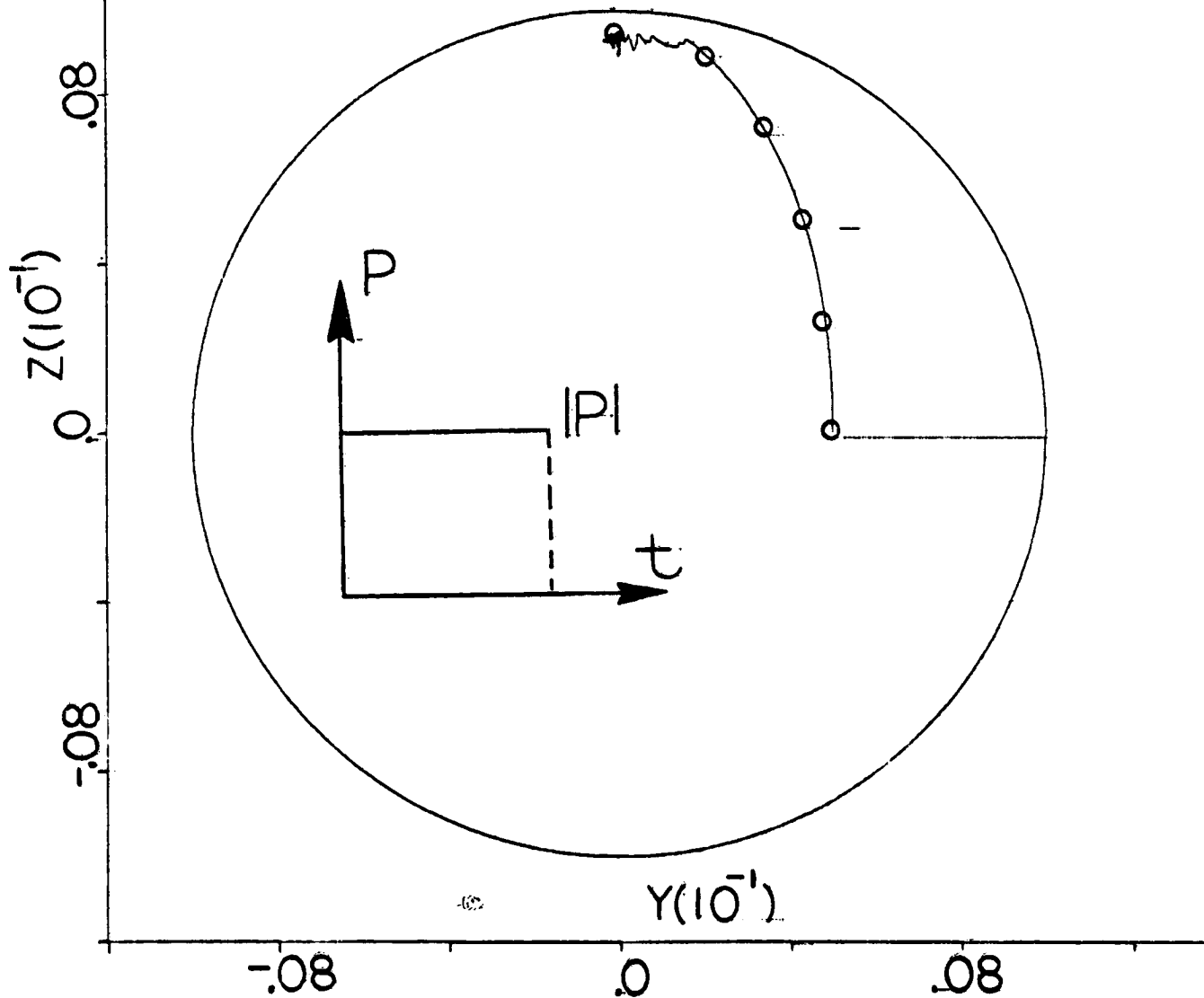


FIGURE 6.10

ROTOR ORBIT
CENTRAL DIFFERENCE
(SEE FIG. 6.1)

ORIGINAL PLOTTING
OF POOR QUALITY---

A1-20

$$\Omega = 150 \text{ RAD/SEC}$$

$$|P| = 2500 \text{ LBS}$$

$$|S| = 0 \text{ LB}$$

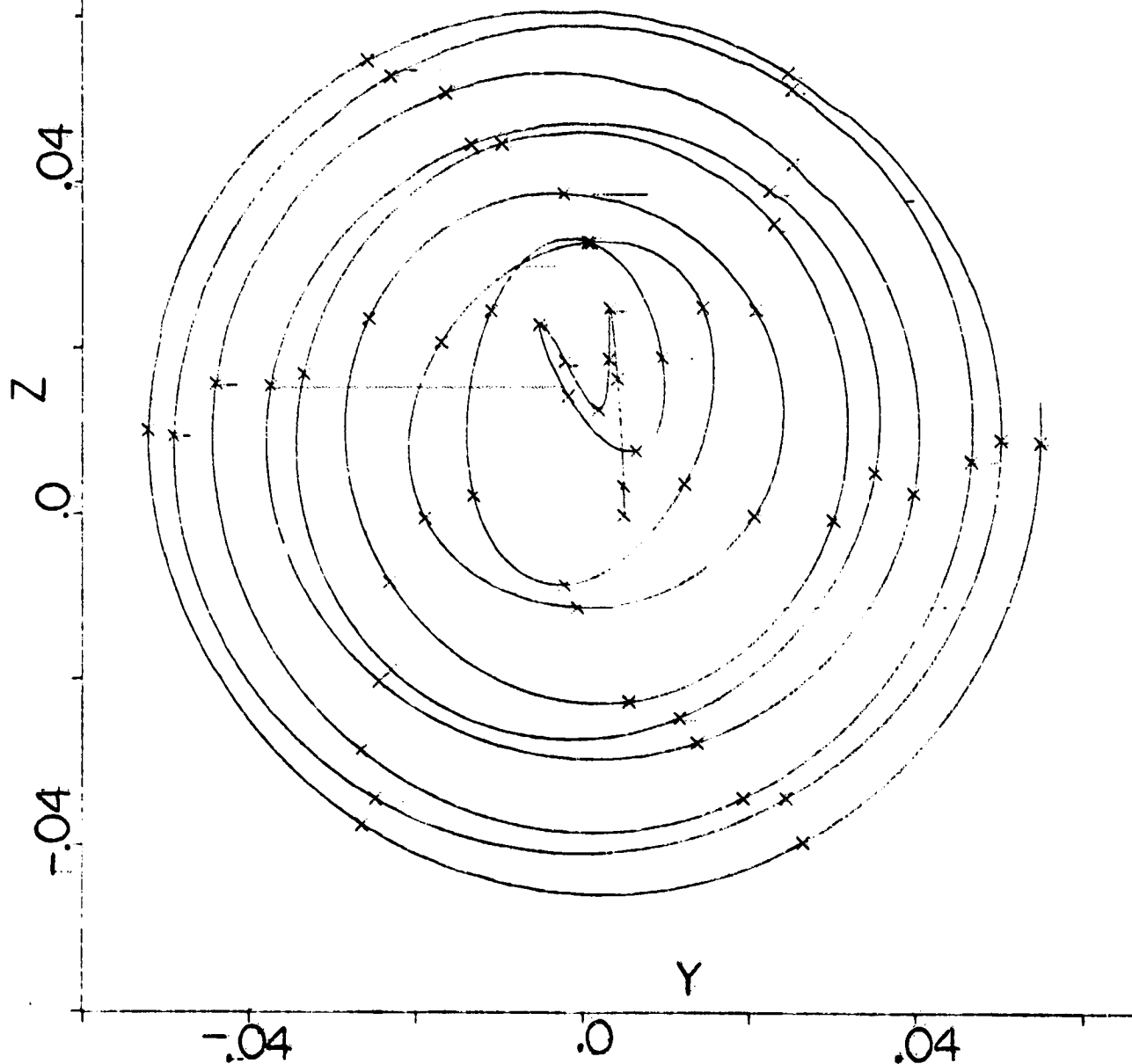


FIGURE 6.11 ROTOR DISPLACEMENT
CENTRAL DIFFERENCE
(SEE FIG. 6.1)

$$\Omega = 150 \text{ RAD/SEC}$$

$$|P| = 2500 \text{ LBS}$$

$$|S| = 0$$

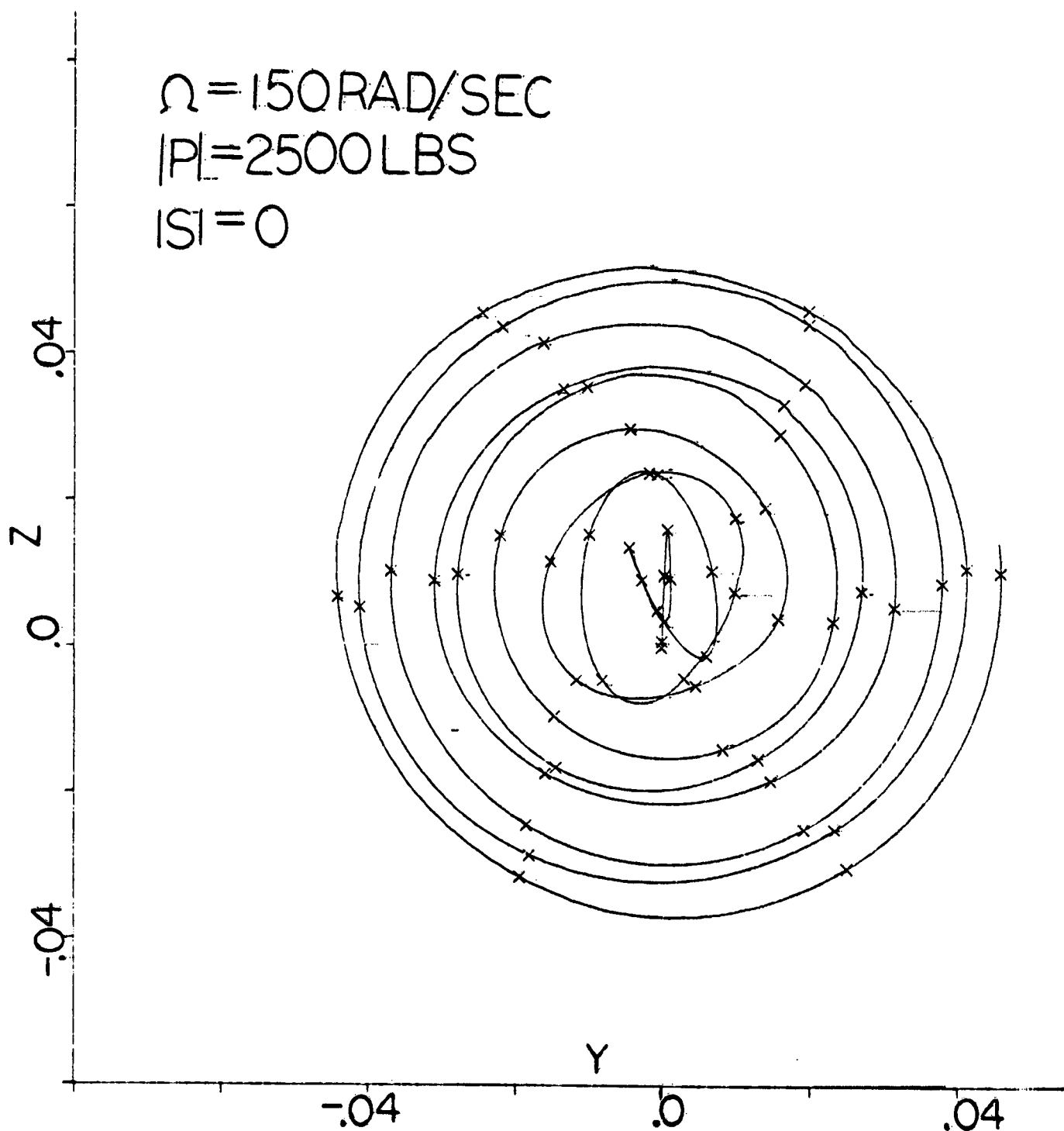


FIGURE 6.12 STATOR DISPLACEMENT
(SEE FIG. 6.1)

ORIGINAL PAGE IS
OF POOR QUALITY

A1-22

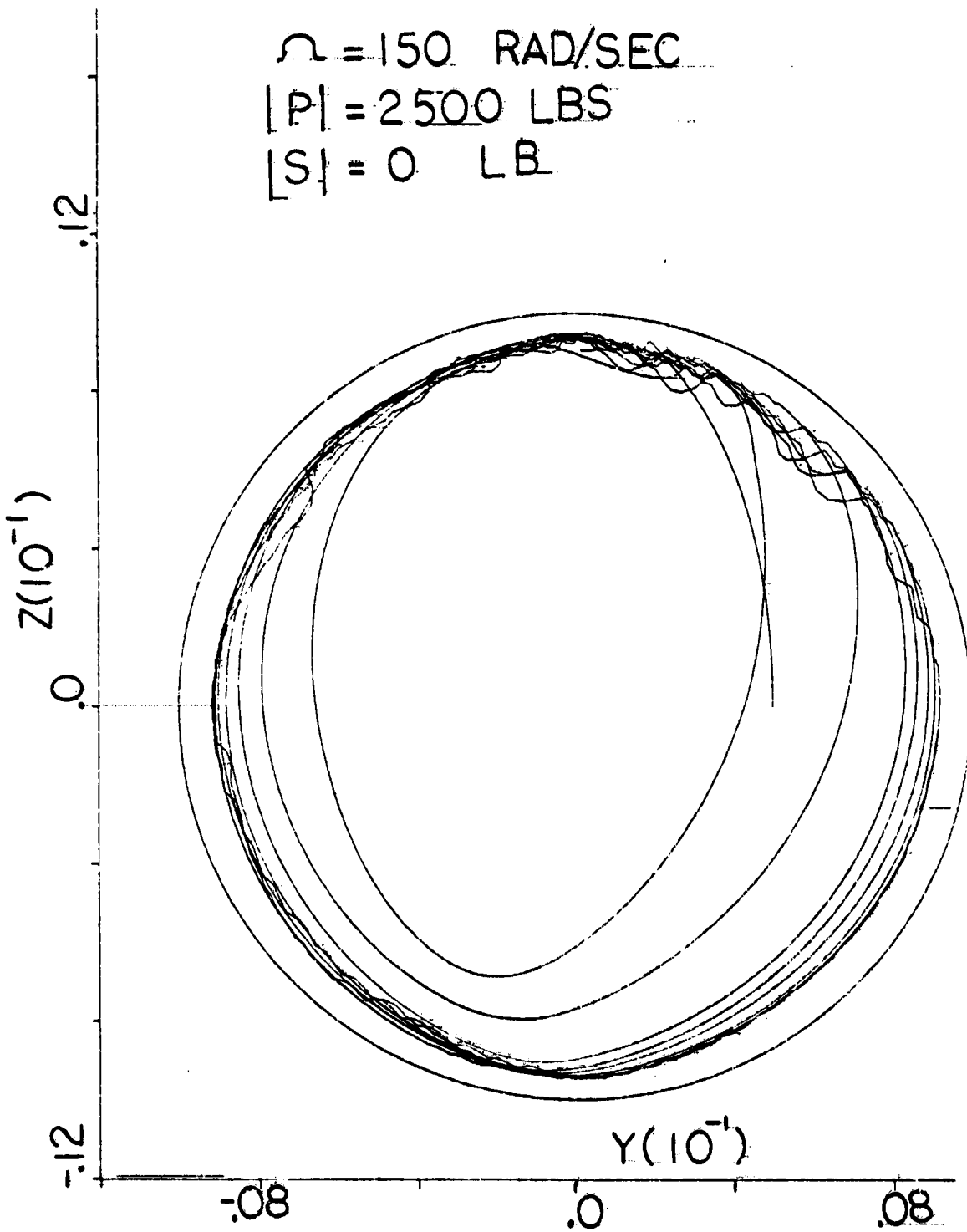


FIGURE 6.13 ROTOR ORBIT
(SEE FIG. 6.1)

$\Omega = 150$ RAD/SEC
 $|P| = 2500$ LBS
 $|S| = 1000$ LBS

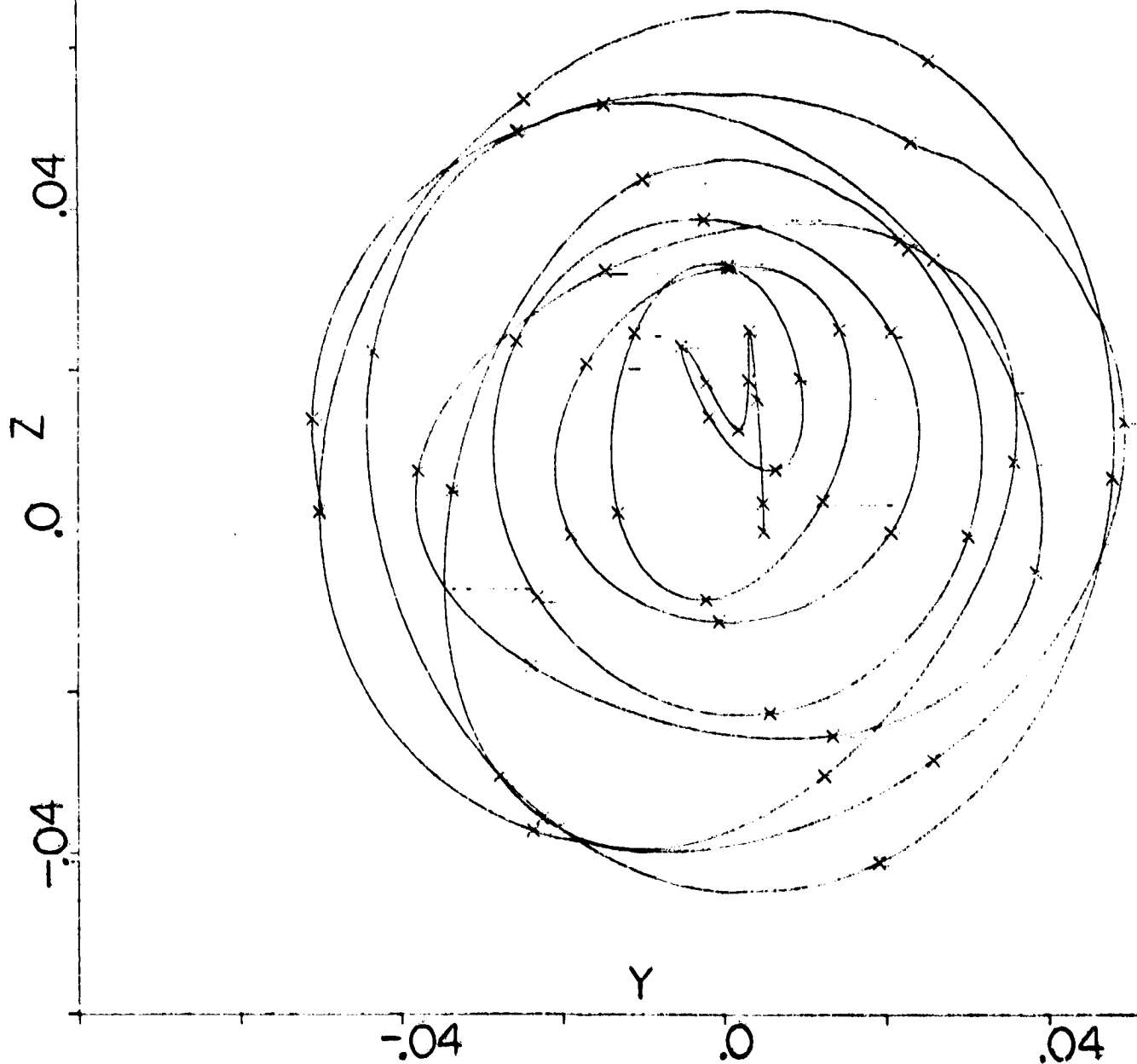


FIGURE 6.14 ROTOR DISPLACEMENT
(SEE FIG. 6.1)

ORIGINAL PAGE IS
OF POOR QUALITY

A1-24

$\Omega = 150 \text{ RAD/SEC}$
 $|P| = 2500 \text{ LBS}$
 $|S| = 1000 \text{ LBS}$

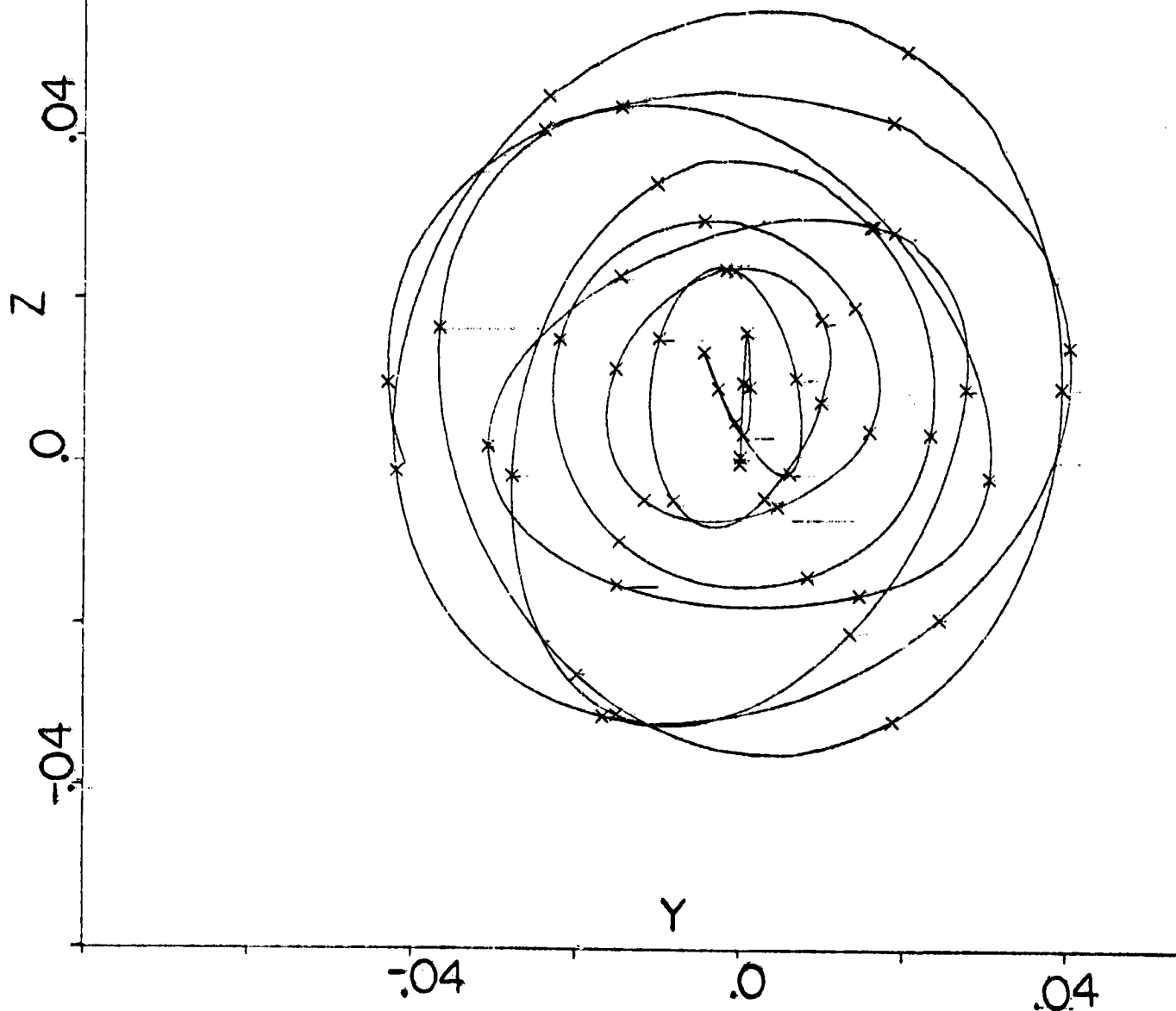


FIGURE 6.15 STATOR DISPLACEMENT
(SEE FIG. 6.1)

ORIGINAL PAPER IS
OF POOR QUALITY

A1-25

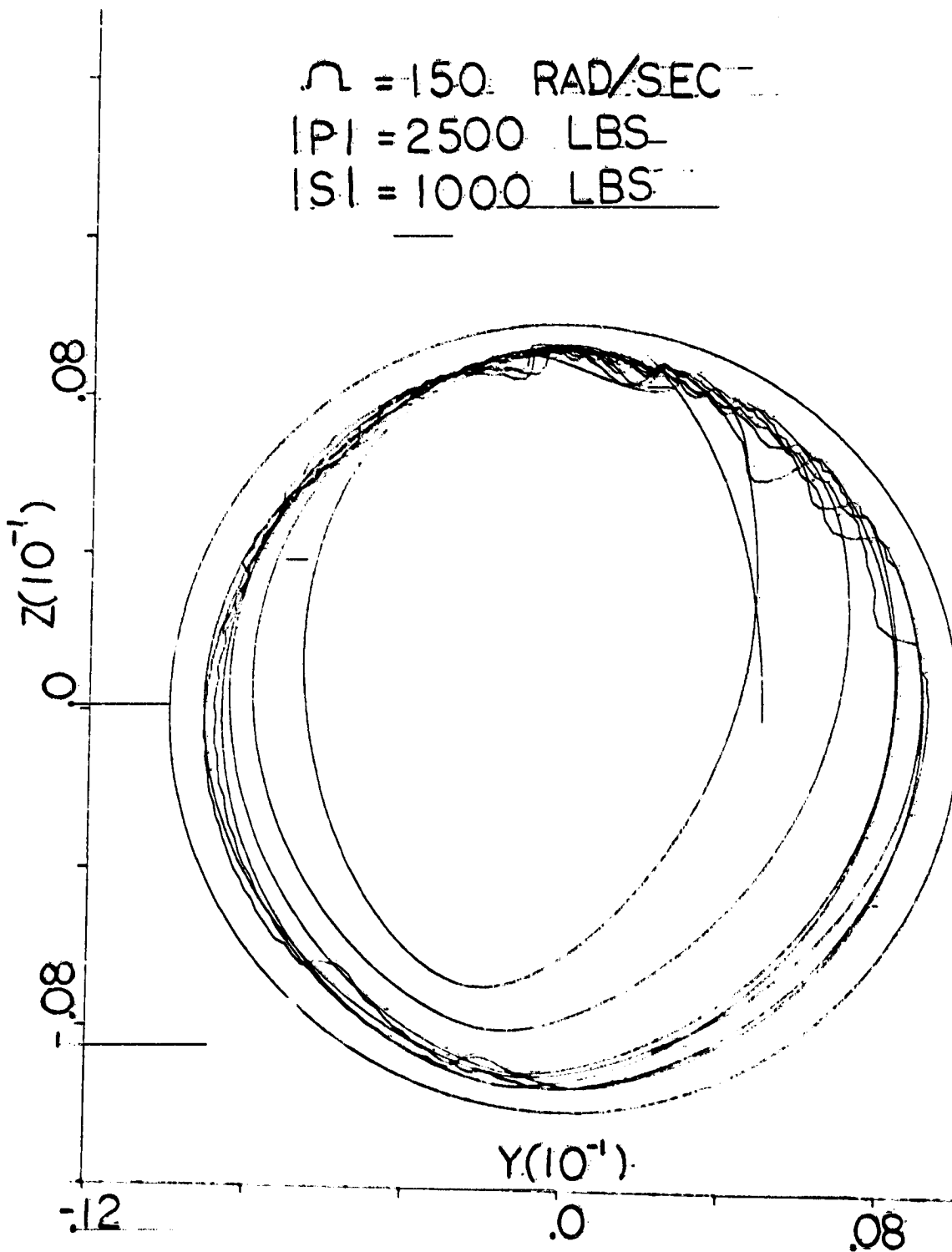


FIGURE 6.16 ROTOR ORBIT
(SEE FIG 6.1)

$A \quad EI \quad B \quad EI \quad C \quad EI \quad D \quad M_S = M_{SFDB} = 500 \text{ LBS}$
 $\leftarrow L \rightarrow \leftarrow L \rightarrow \leftarrow L \rightarrow \quad M = 1000 \text{ LBS}$
 $E = 30 \cdot 10^6 \text{ PSI}, \quad I = 463 \text{ INCH}^4, \quad L = 48 \text{ INCH}$

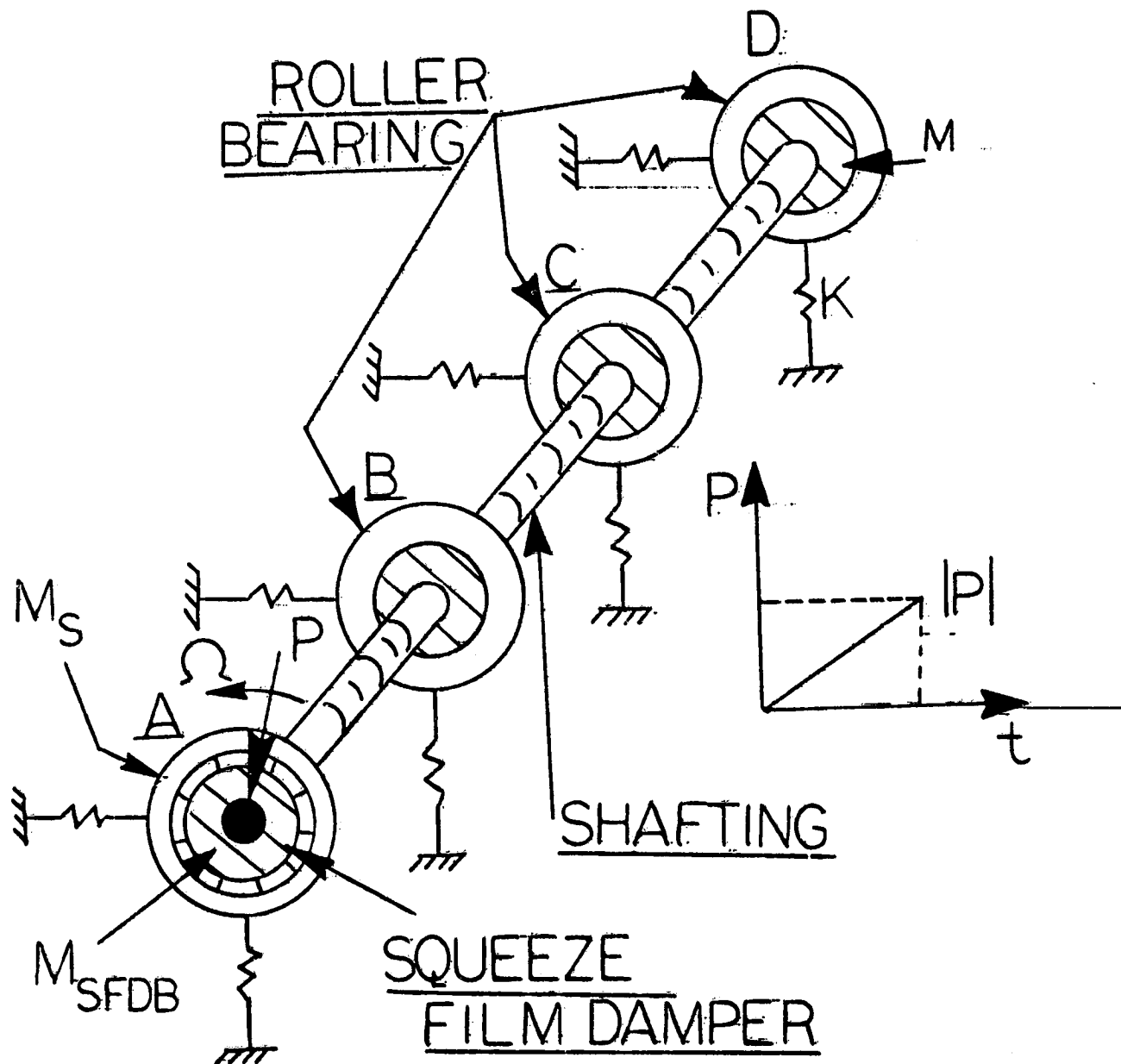


FIG.6.17 MULTI BEARING MODEL

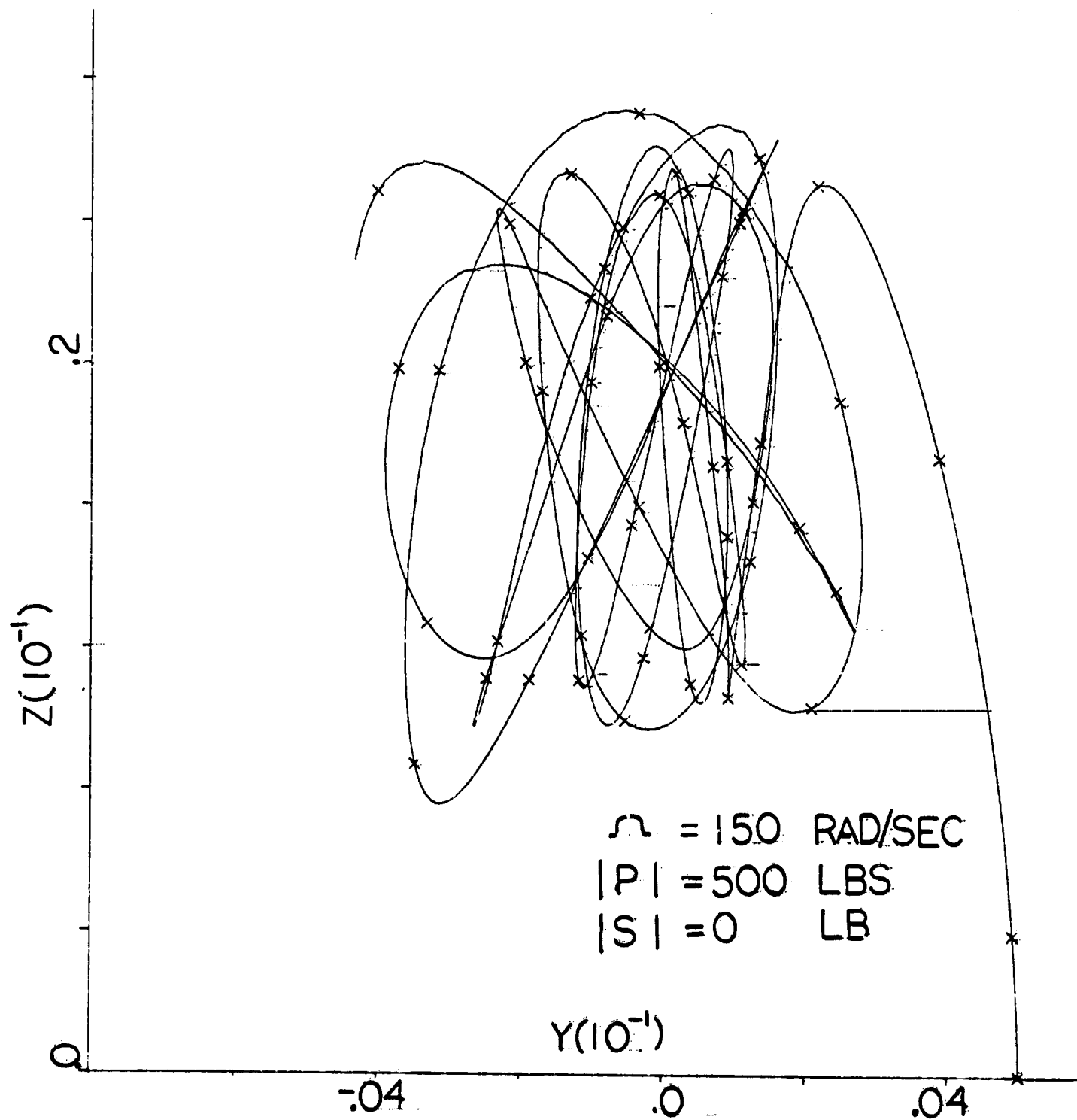
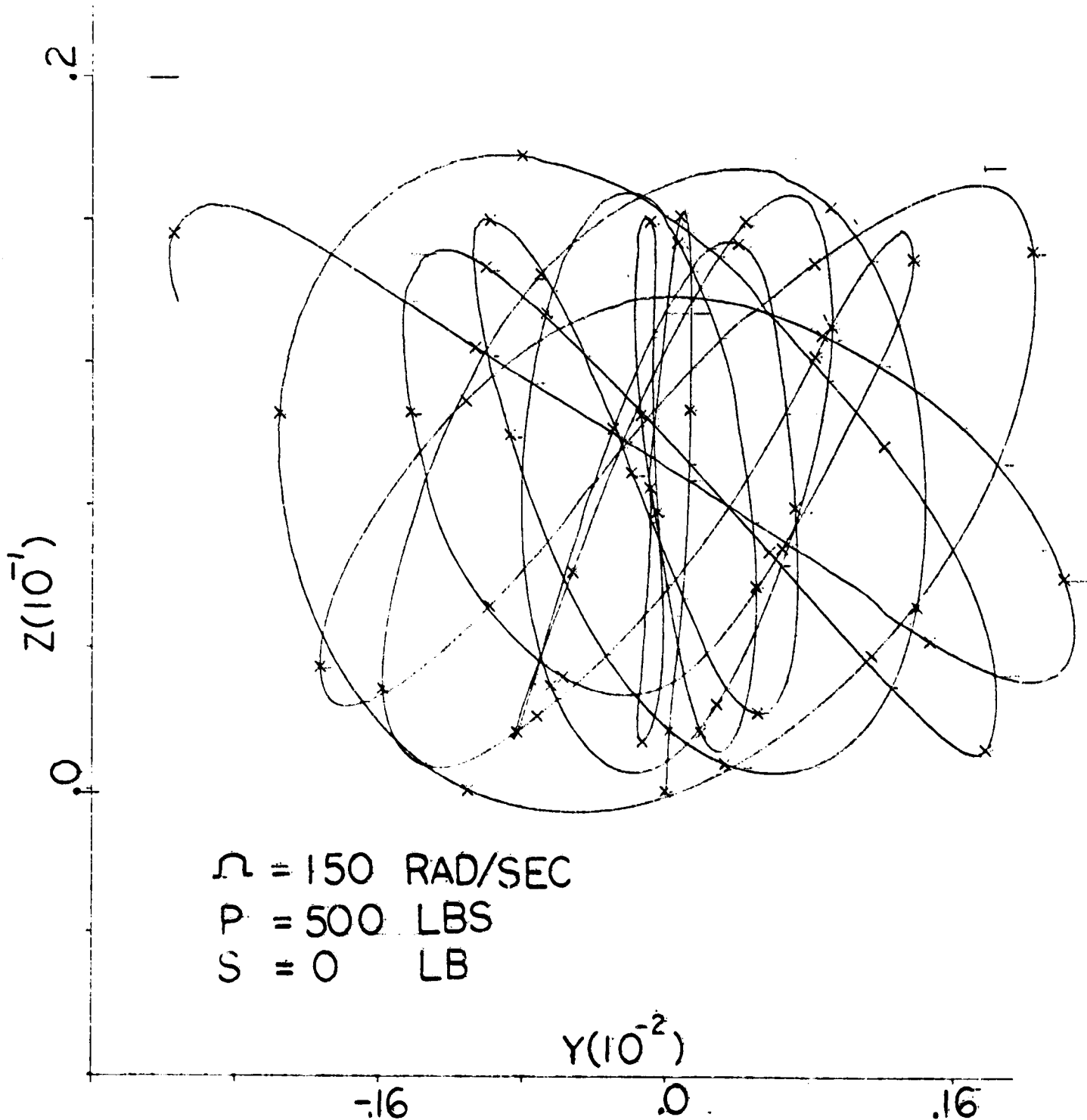


FIGURE 6.18 ROTOR DISPLACEMENT
(SEE FIG. 6.17)

ORIGINAL DRAWING
OF FOUR QUALITYFIGURE 6.19 STATOR DISPLACEMENT
(SEE FIG. 6.17)

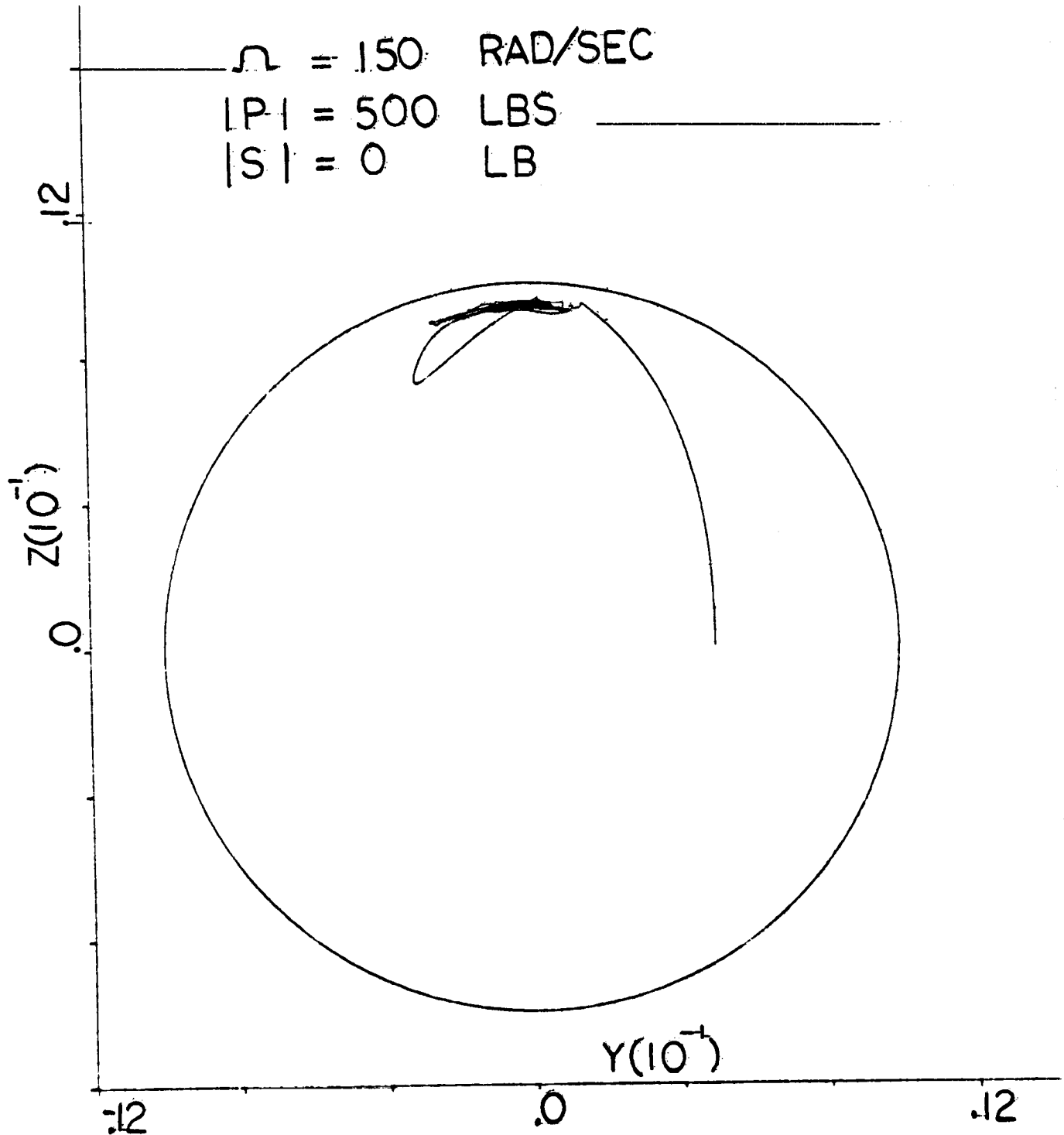


FIGURE 6.20 ROTOR ORBIT
(SEE FIG. 6.17)

ORIGINAL COPY IS
OF POOR QUALITY

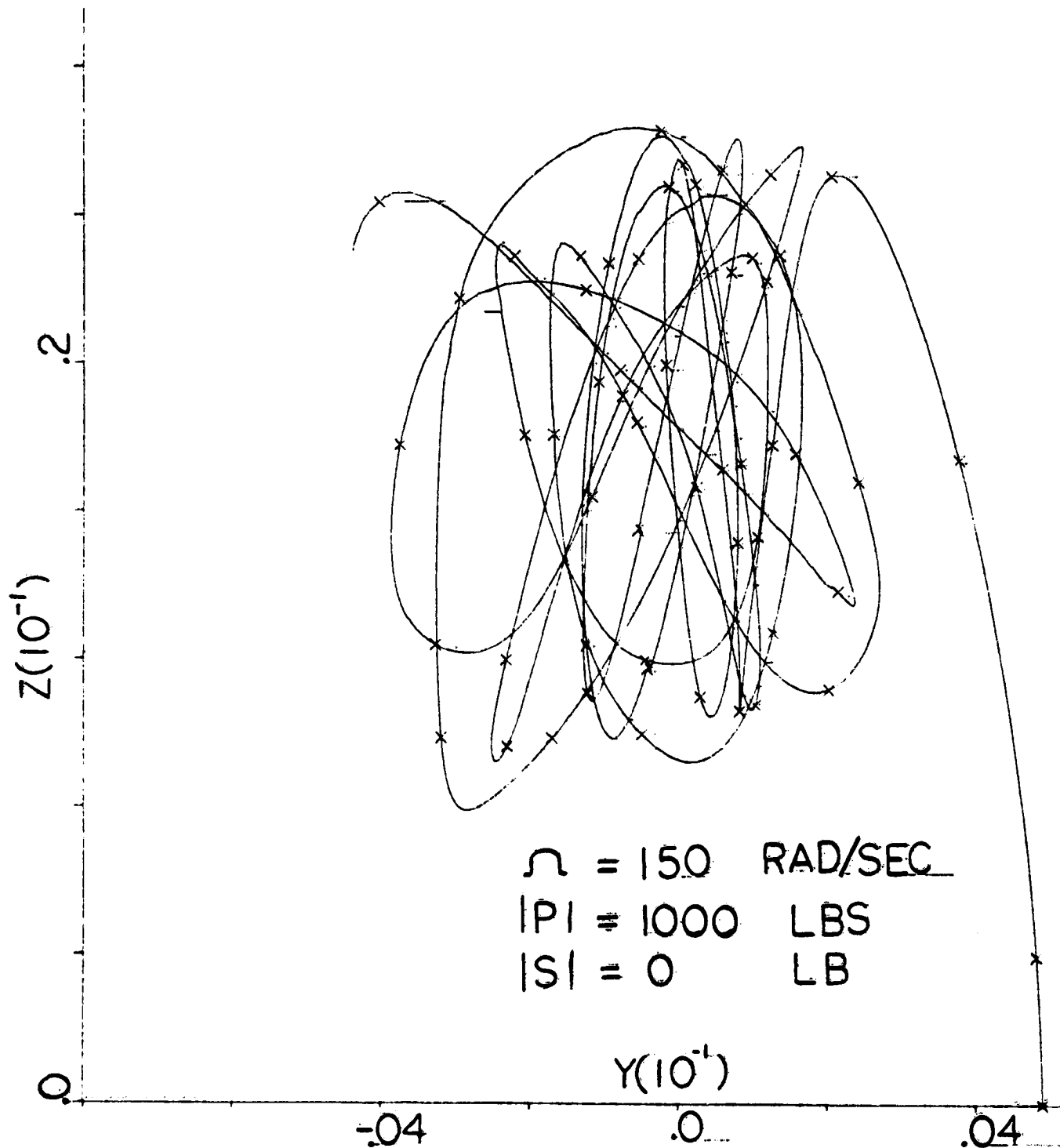


FIGURE 6.21 ROTOR DISPLACEMENT
(SEE FIG. 6.17)

ORIGINAL FILED IN
OF POOR QUALITY

A1-31

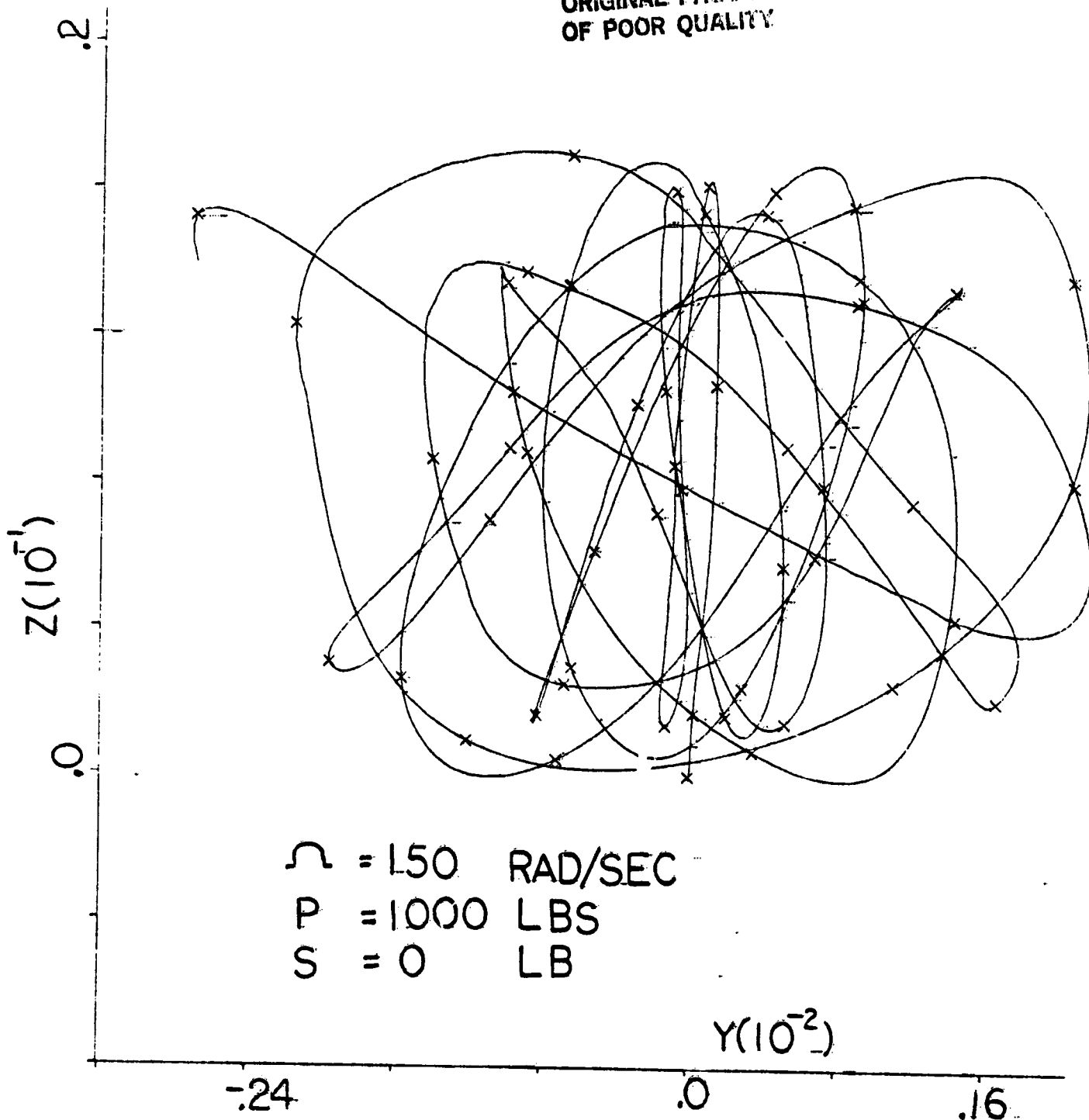


FIGURE 6.22 STATOR DISPLACEMENT
(SEE FIG. 6.17)

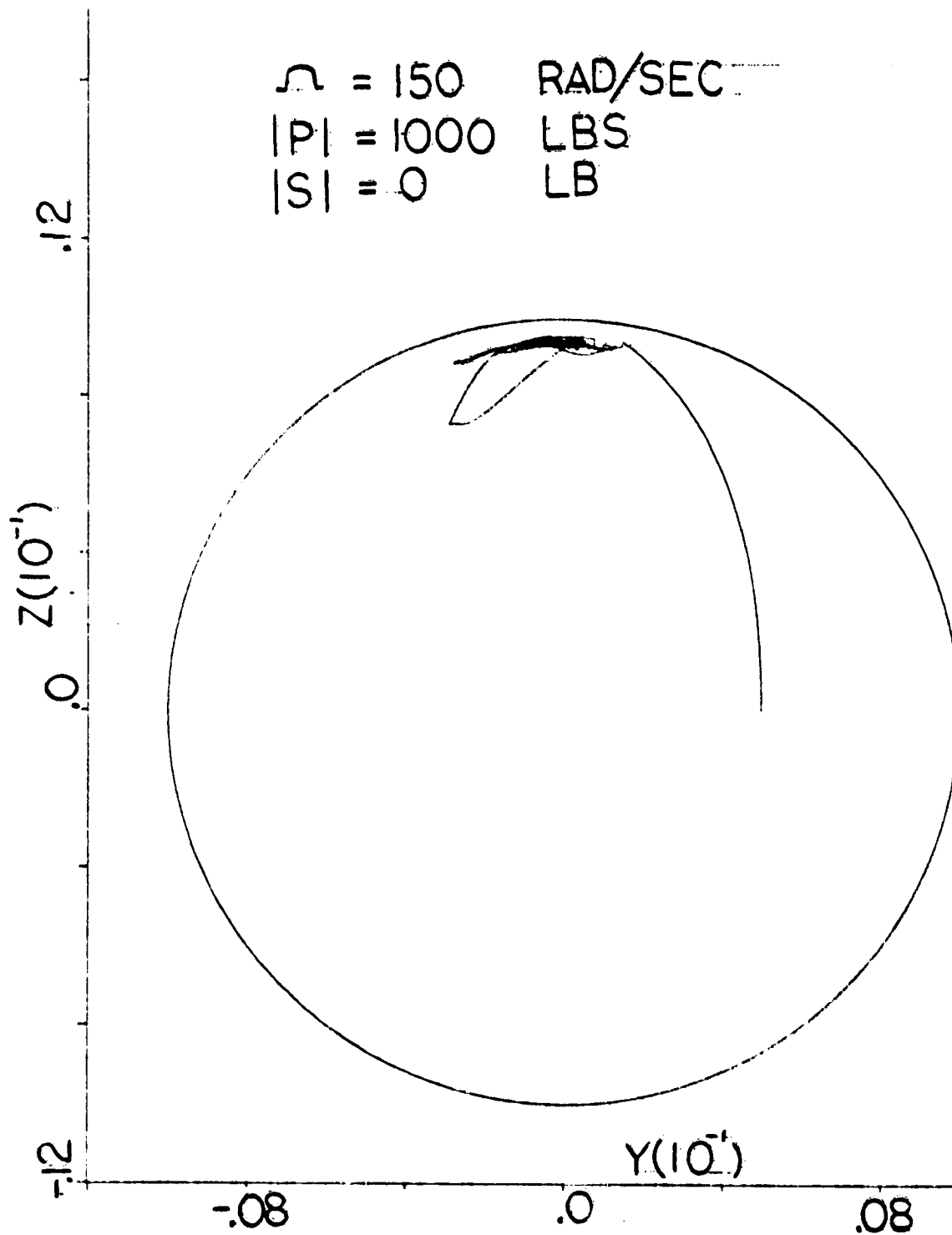


FIGURE 6.23 ROTOR ORBIT
(SEE FIG 6.17)

$\Omega = 150$ RAD/SEC
 $|P| = 2500$ LBS
 $|S| = 0$ LB

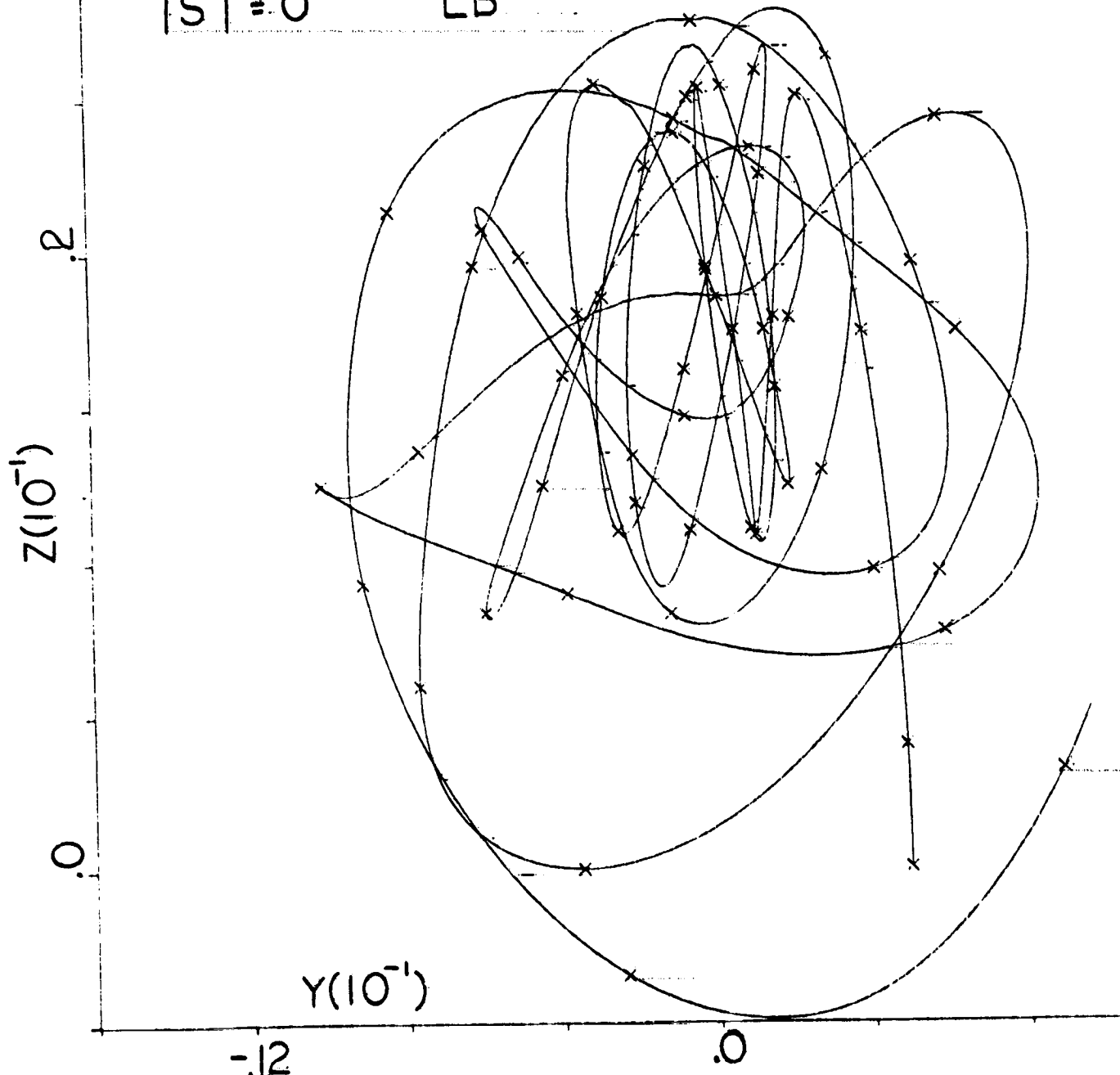


FIGURE 6.24 ROTOR DISPLACEMENT
(SEE FIG. 6.17)

ORIGINAL DRAWING IS
OF POOR QUALITY

A1-34

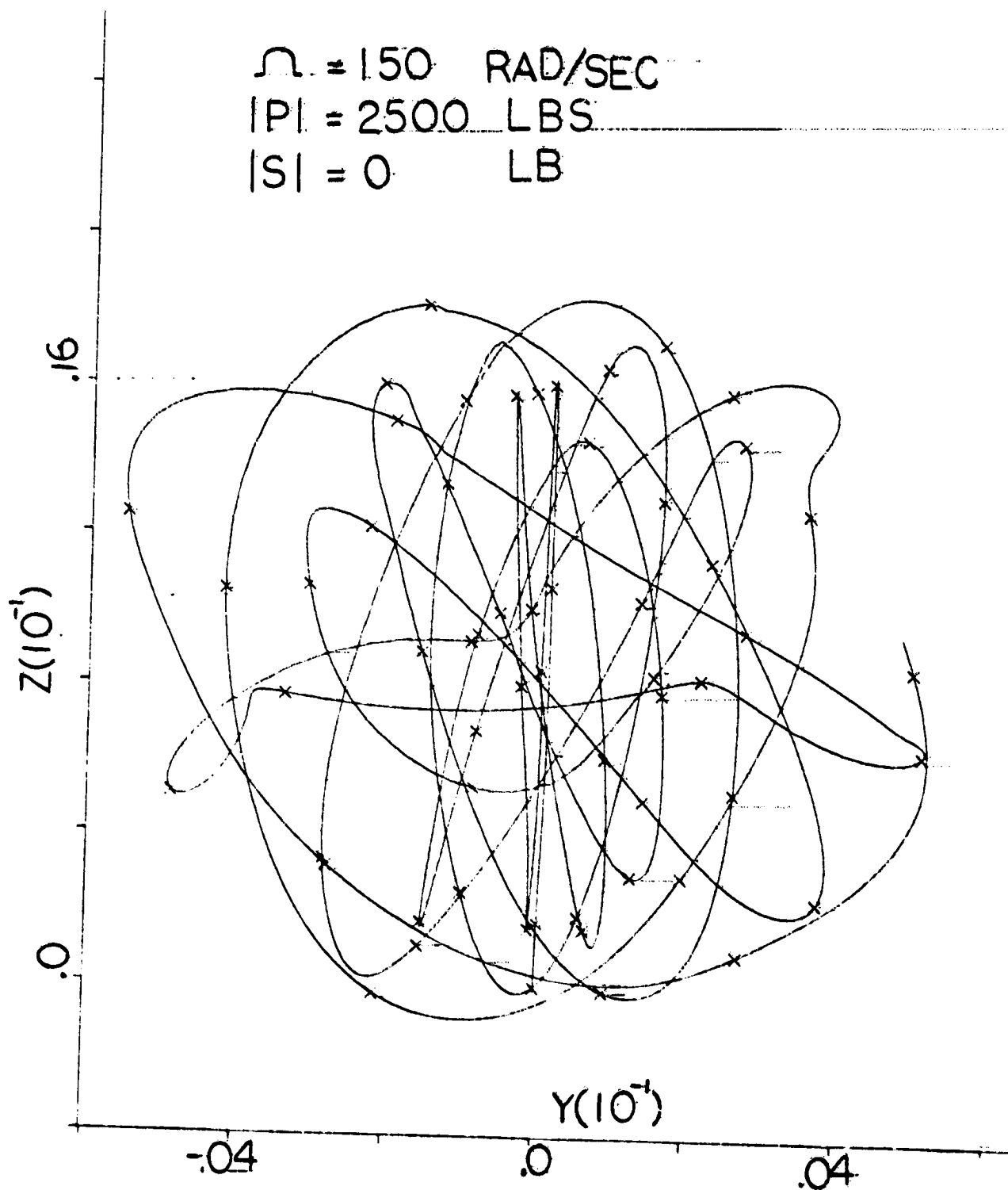


FIGURE 6.25 STATOR DISPLACEMENT
(SEE FIG. 6.17)

ORIGINAL PAGE IS
OF POOR QUALITY

AT-35

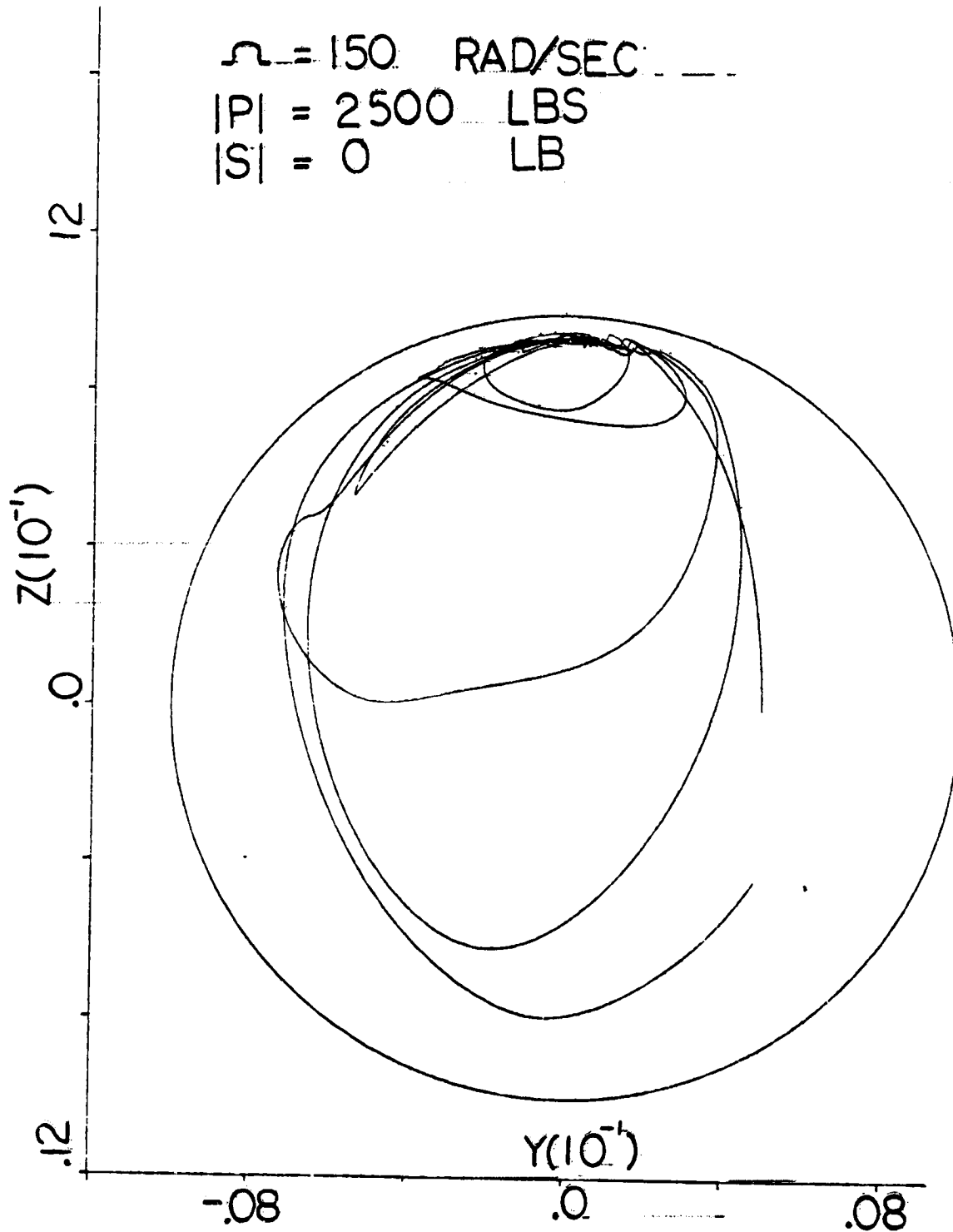


FIGURE 6.26 ROTOR ORBIT
(SEE FIG. 6.17)

ORIGINAL PAGE IS
OF POOR QUALITY

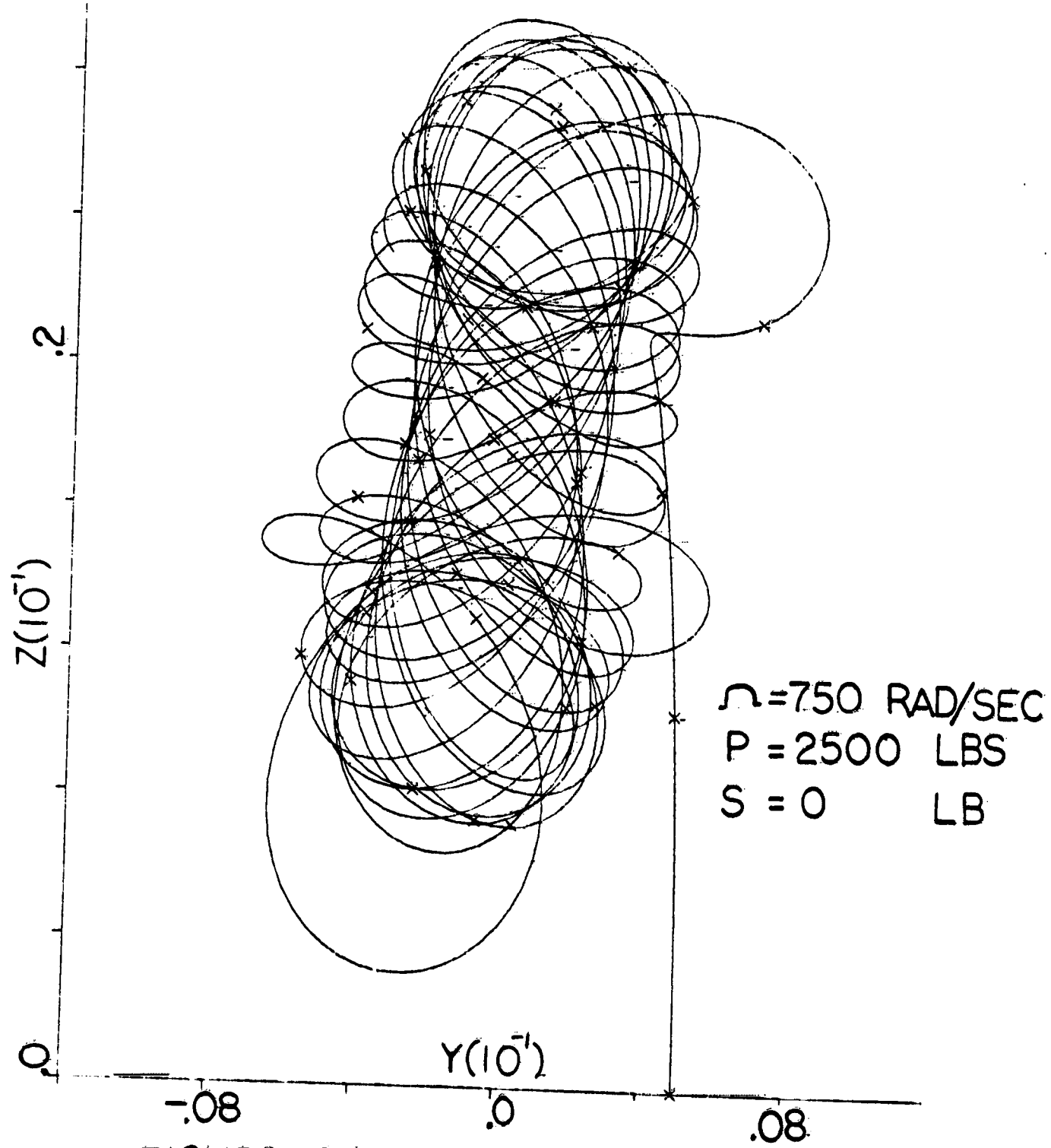


FIGURE 6.27 ROTOR DISPLACEMENT
(SEE FIG. 6.17)

$\Omega = 750$ RAD/SEC
 $|P| = 2500$ LBS
 $|S| = 0$ LB

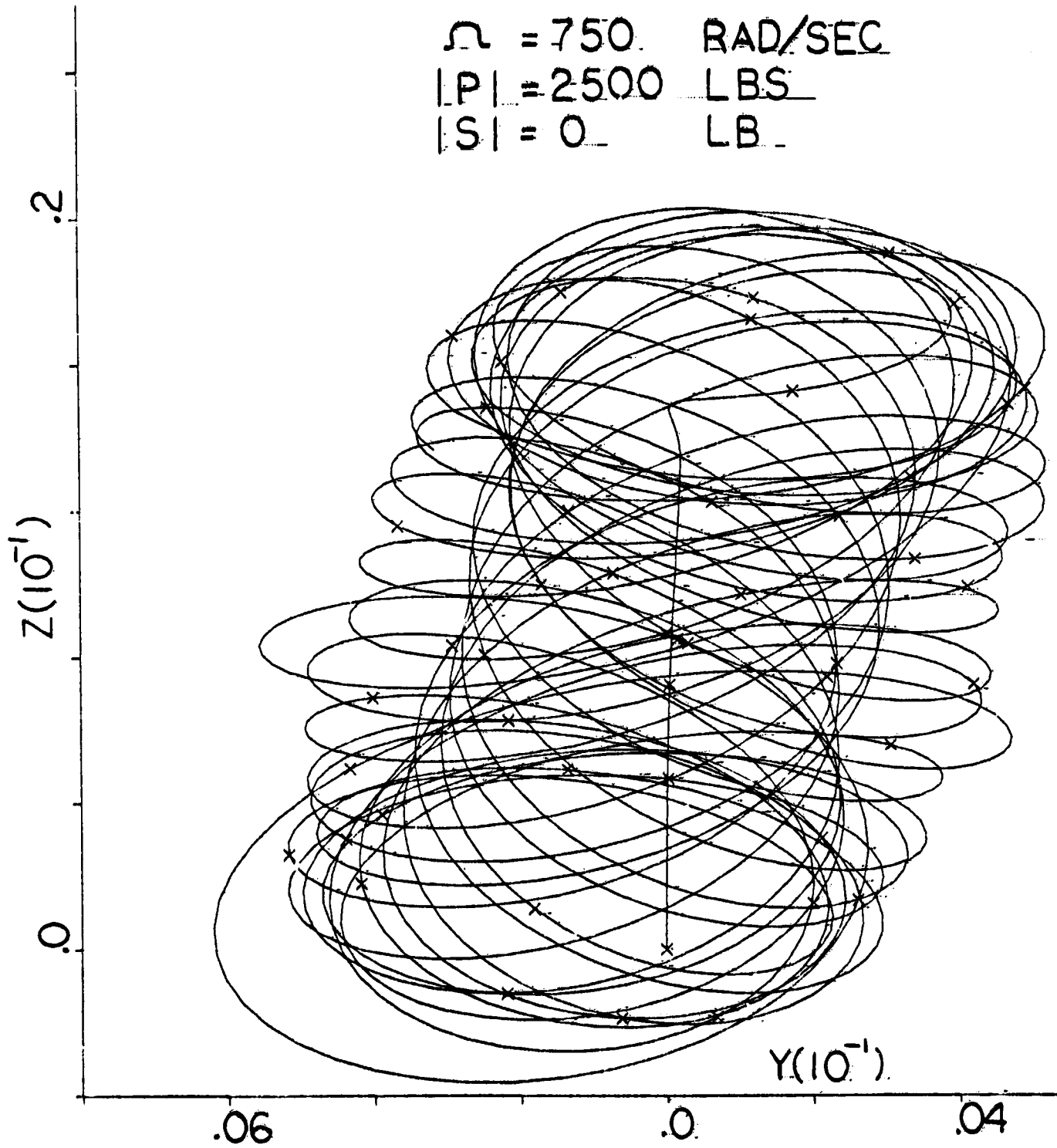


FIGURE 6.28 STATOR DISPLACEMENT
(SEE FIG. 6.17)

ORIGINAL PAGE IS
OF POOR QUALITY

$$\begin{aligned}\Omega &= 750 \text{ RAD/SEC} \\ |P| &= 2500 \text{ LBS} \\ |S| &= 0 \text{ LB}\end{aligned}$$

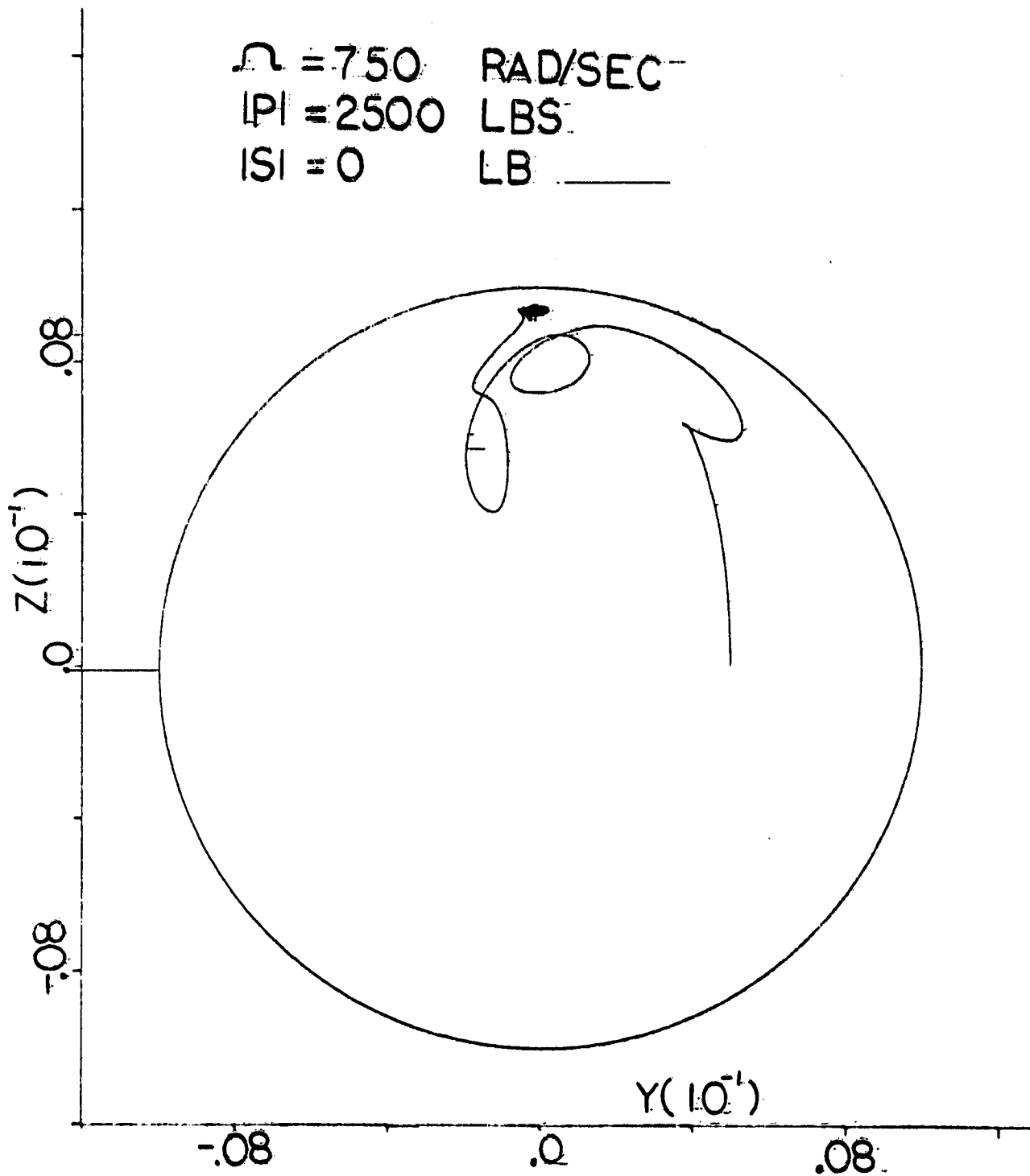


FIGURE 6.29 ROTOR ORBIT
(SEE FIG. 6.17)

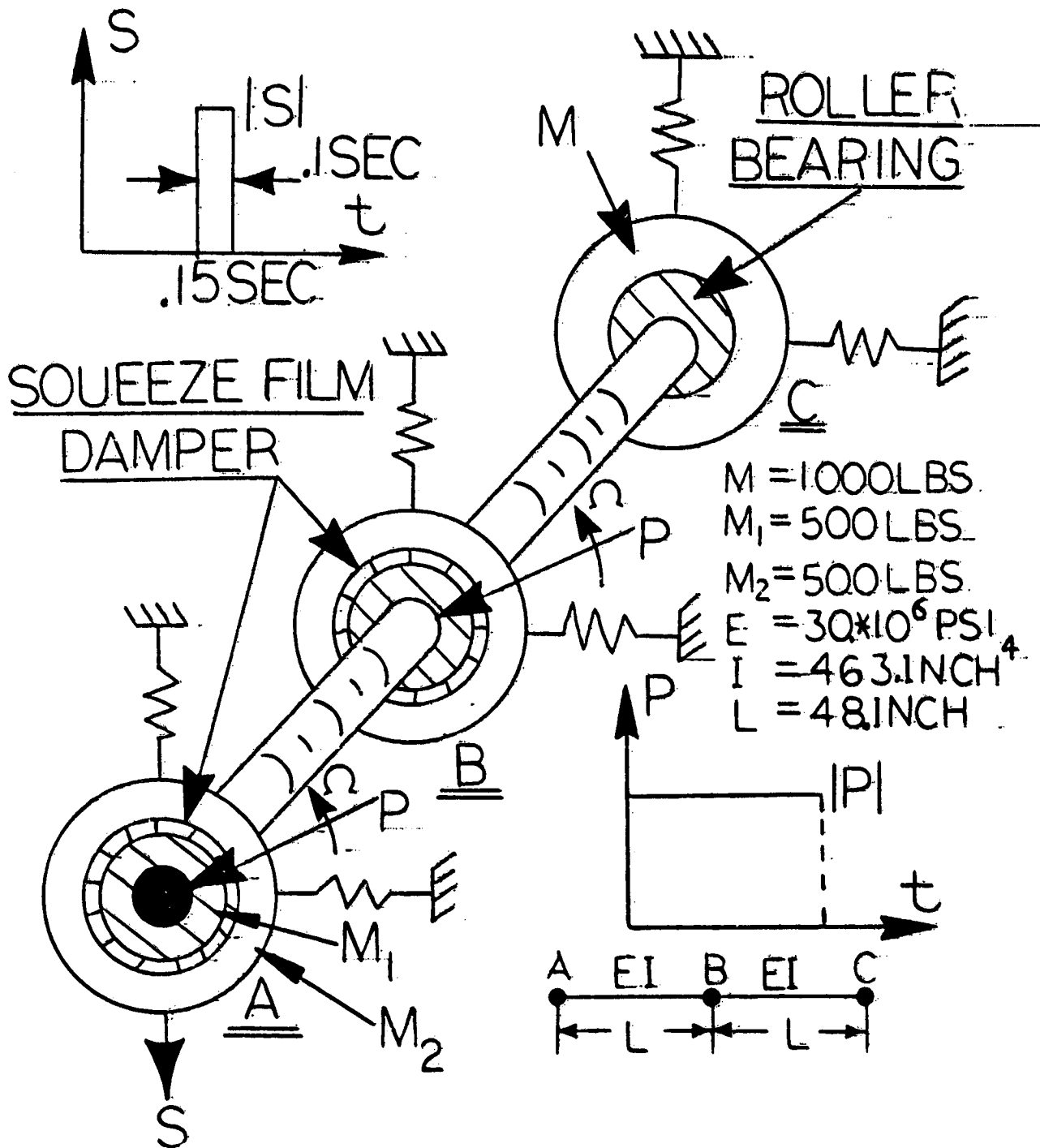


FIG. 6-30 MULTI BEARING MODEL

ORIGINAL PAGE IS
OF POOR QUALITY

A1-40

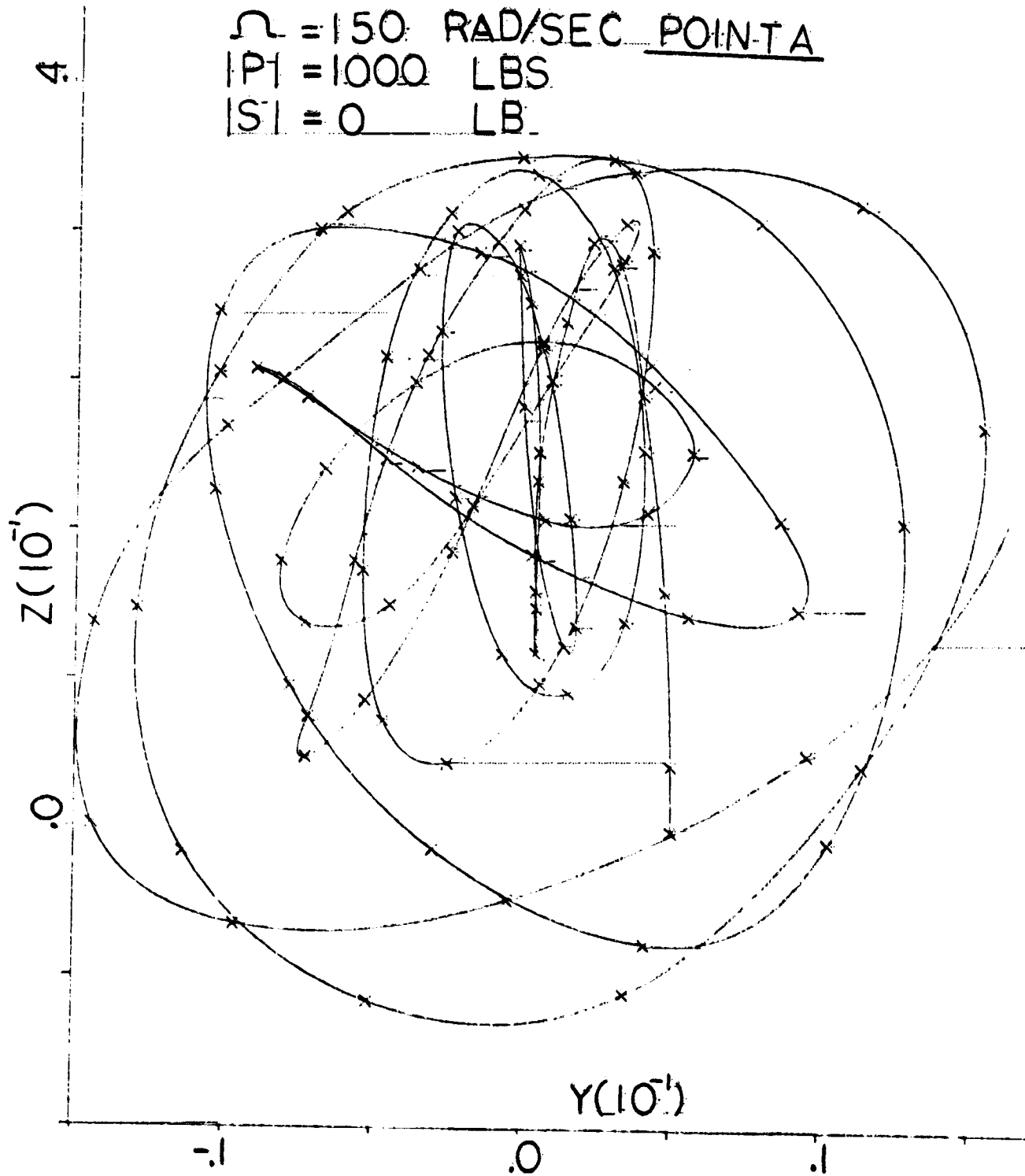


FIGURE 6.31 ROTOR DISPLACEMENT
(SEE FIG. 6.30)

$\Omega = 150$ RAD/SEC POINT A
IPL = 1000 LBS
ISI = 0 LB

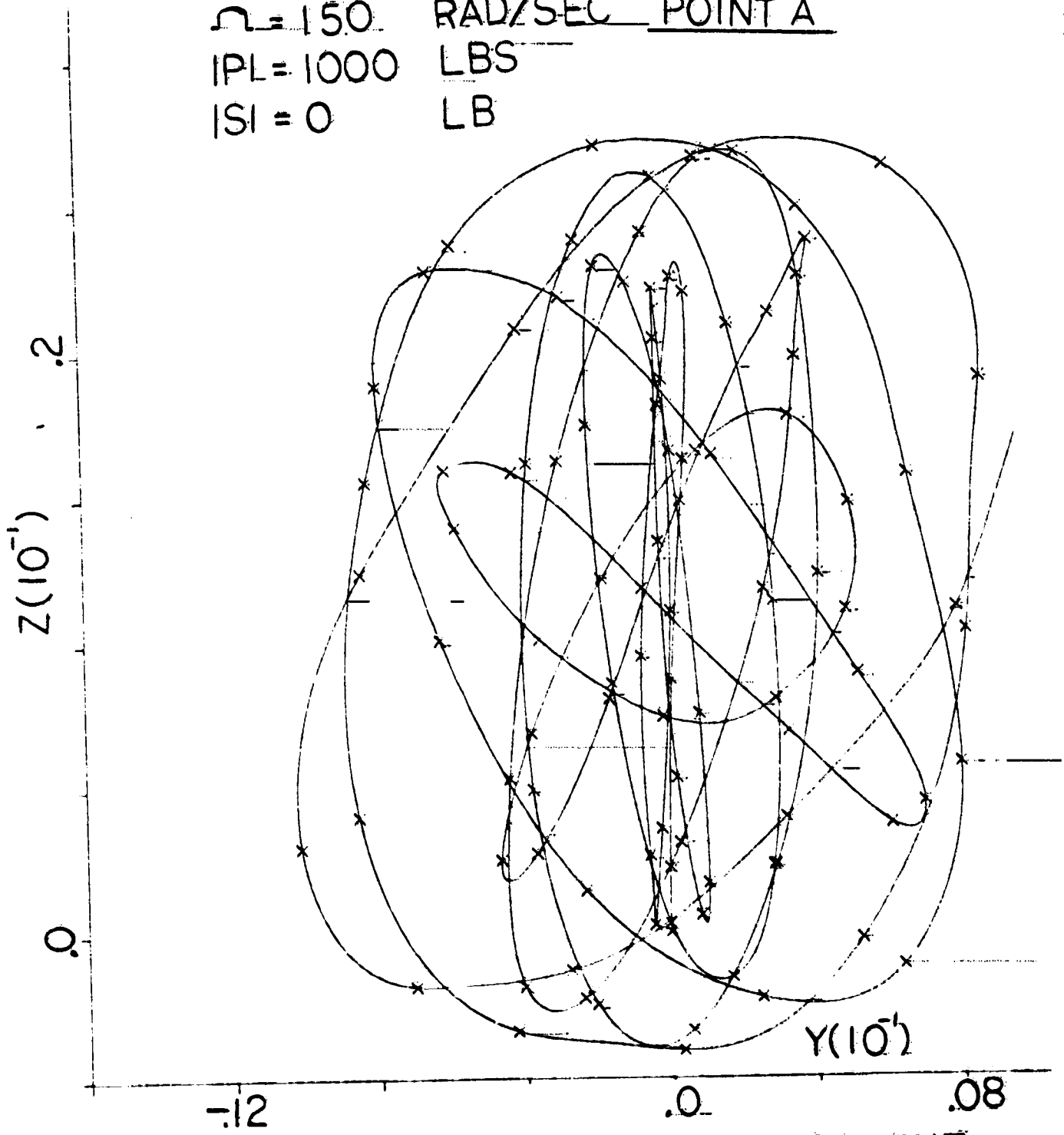


FIGURE 6.32 STATOR DISPLACEMENT
(SEE FIG. 30)

ORIGINAL PAGE IS
OF POOR QUALITY

A1-42

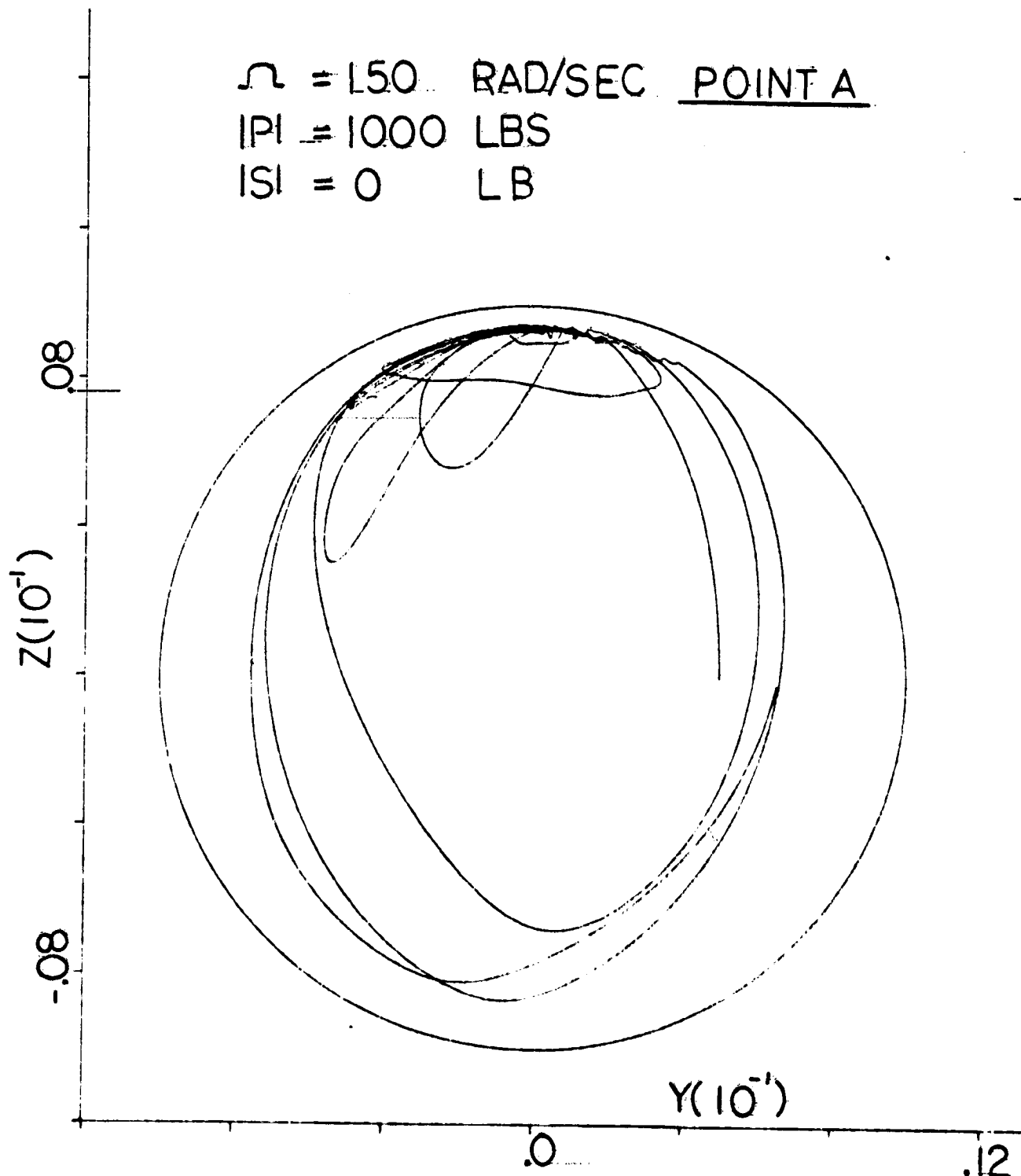


FIGURE 6.33 ROTOR ORBIT
(SEE FIG. 30)

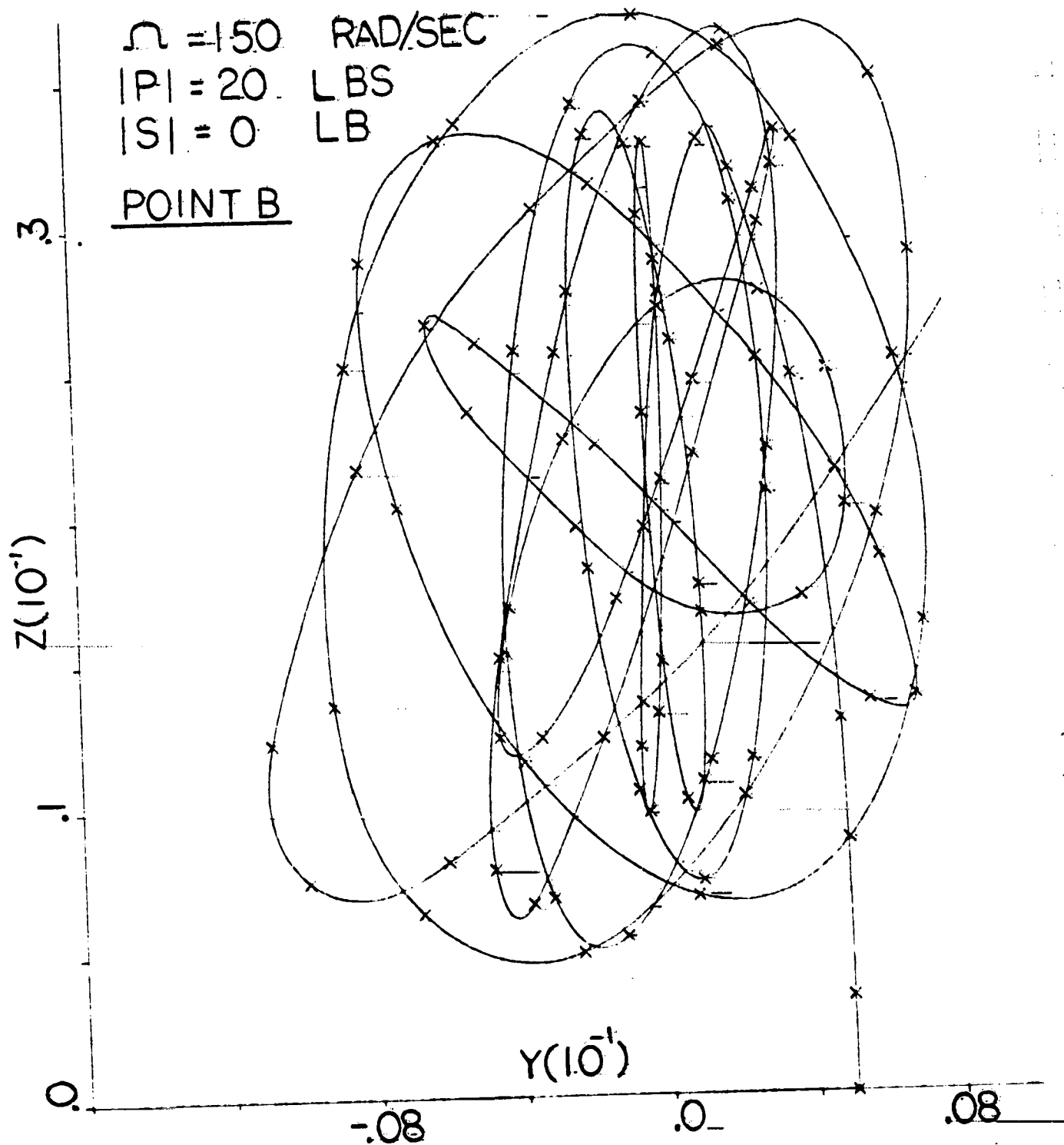


FIGURE 6.34 ROTOR DISPLACEMENT
(SEE FIG. 6.30)

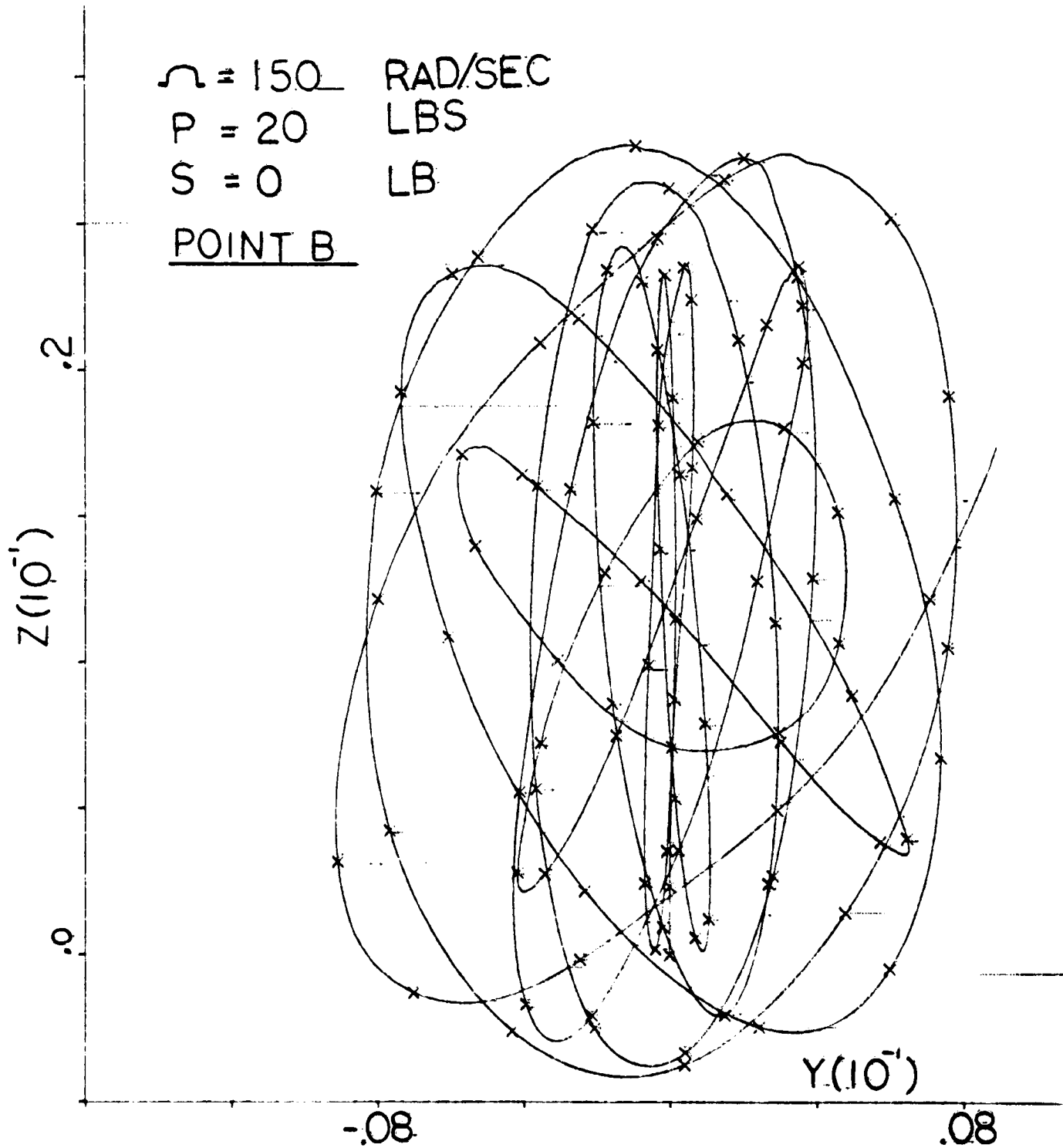


FIGURE 6.35 STATOR DISPLACEMENT.
(SEE FIG. 6.30)

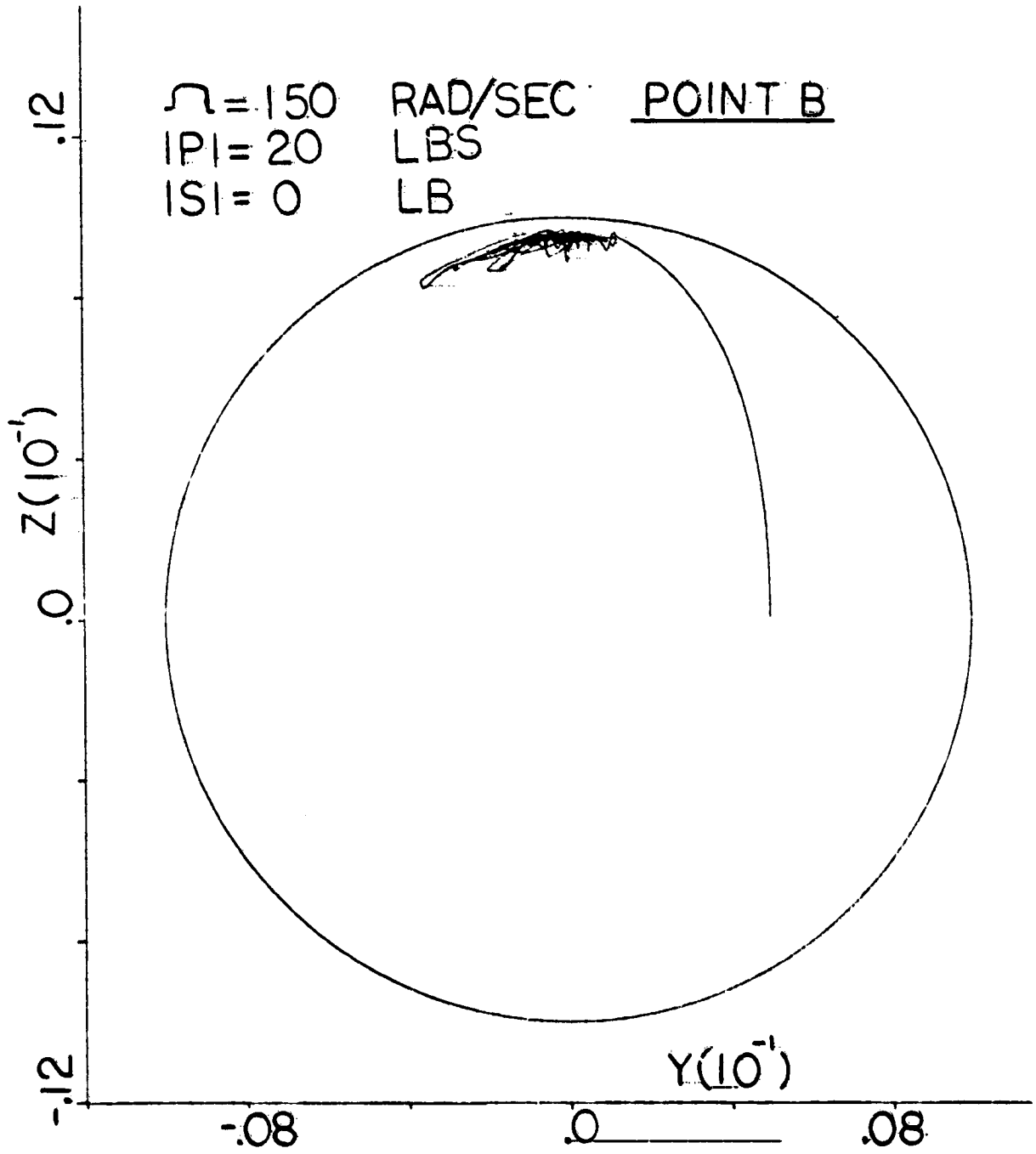


FIGURE 6.36 ROTOR ORBIT
(SEE FIG. 6.30)

APPENDIX 2: USER I/O INFORMATION

This appendix outlines a supplement that must be employed together with the main "ADINA" manual to help model and run general engine dynamic simulations. The manual presents the necessary changes and additions to "ADINA" to implement the bearing element into the code. . .

To simplify its use, the original flow of the ADINA input remains basically the same. This is achieved by adding a new bearing element to the I/O stream. This was made possible by introducing several new features to the I/O namely:

- (1) A master control card;
- (2) A bearing element group;
- (3) A rotating imbalance loads generator; and,
- (4) Plotting capabilities.

In what follows, items i). - iv) will be discussed in detail. As noted earlier, such information is to be used in conjunction with the standard ADINA I/O manual. In keeping with the style of that manual, the following discussion is written in the same form.

A2.1 Master Control Card

Card 3 in ADINA user's manual should be replaced by the following card:

II MASTER CONTROL CARDS (continued)

Card 3: Frequency Solution Request Card (215)

note	columns	variable	entry
(1)	1 - 5	IEIG	Flag indicating frequency solution mode; EQ.0; no frequency solution EQ.1; find lowest frequencies and associated mode shapes

note	columns	variable	entry
(2)	6 - 10	IBEAR	Flag indicating bearing dynamic simulation; EQ.0; no bearing simulation required EQ.1; bearing simulation required

NOTES/

- (1) See ADINA user's manual.
- (2) The control parameter IBEAR determines if the program is to simulate squeeze film damper employing the bearing element. If IBEAR equal to 1, then bearing element group must be input.

A2.2 Bearing Element Group

Bearing elements are members allowed arbitrary orientation in the global coordinate system used in ADINA. The bearing element is 2-node element with only global translational degrees of freedom at each node. The bearing element simulates the squeeze film damper. One of the two nodes of the element represents the damper while the second node represents the stator.

The following input is required for each element group consisting of bearing elements:

Section A Element Group Control Card

This card defines the number of elements in this group, etc.

Section B Element Data Cards

1. Geometric and fluidic properties of squeeze film damper.
The nominal damper annulus diameter, length and radial clearance are defined in this section for each bearing element. The positions of the damper ports, the pressure

at each port, the lubricant viscosity, and the film rupture pressure are also defined. Other variables related to the calculations of the damper tangent stiffness and damping matrices are defined in this section.

2. Element Nodes: The nodal points of bearing elements are input in this part.

A specific element group defined by the above input cards is followed by the input cards of another element group.

A2.2A Element Group Control Card (6T4)

note	columns	variable	entry
	1 - 4	NPAR(1)	Enter number "1"
(1)	5 - 8	NPAR(2)	Number of bearing elements; GE.1
(2)	9 - 12	NPAR(3)	Enter the number "1"
(3)	13 - 16	NPAR(4)	Element birth and death option; EQ.0 elements are active throughout the solu- tion (see note 3)
(4)	17 - 20	NPAR(5)	Element type code; Enter the number "3"
(5)	25 - 28	NPAR(7)	Maximum number of nodes used in any one element; Enter the number "2"

NOTES/

- (1) BEARING element numbers in each group begin with one (1) and end with the total number of elements in the group.

- (2) The parameter NPAR(3) is used to help the program to go through the proper execution path.
- (3) The variable NPAR(4) identifies whether the elements of the element group are active throughout the solution. NPAR(4) can take different modes. For the present time, it takes the value 0. If NPAR(4) .EQ. 0, the elements are active throughout the solution.
- (4) The variable NPAR(5) defines the type of the element used in this element group. Currently, it only takes the value "3".
- (5) NPAR(7) limits the number of nodes that can be used to describe any bearing element in this element group.

A2.2B Element Data Cards

A2.2B.1 Geometric and Fluidic Properties of Bearing Element

The necessary pre-selected geometric parameters of the damper is read in this section. This includes the annulus diameter of the damper, the bearing length, and the bearing clearance. The oil properties (e.g., viscosity and pressure rupture) are also read. The element selection, stiffness matrix formation, and the damping matrix formation options are input in this section also.

Card 1: (I10,6f10.0)

note	columns	variable	entry
(1)	1 - 10	M	Bearing element number; GE.1 and LE.NPAR(2)
	11 - 20	BD	Nominal bearing annulus diameter
	21 - 30	BL	Nominal bearing length
	31 - 40	BC	Bearing annulus radial clearance
	41 - 50	VISC	Bearing lubricant vis- cosity
	51 - 60	PVAP	Film rupture absolute pressure (PSIA)
(2)	61 - 70	TH1	Position angle of lubri- cant port-1 (degrees)

Card 2: (3f10.0,615)

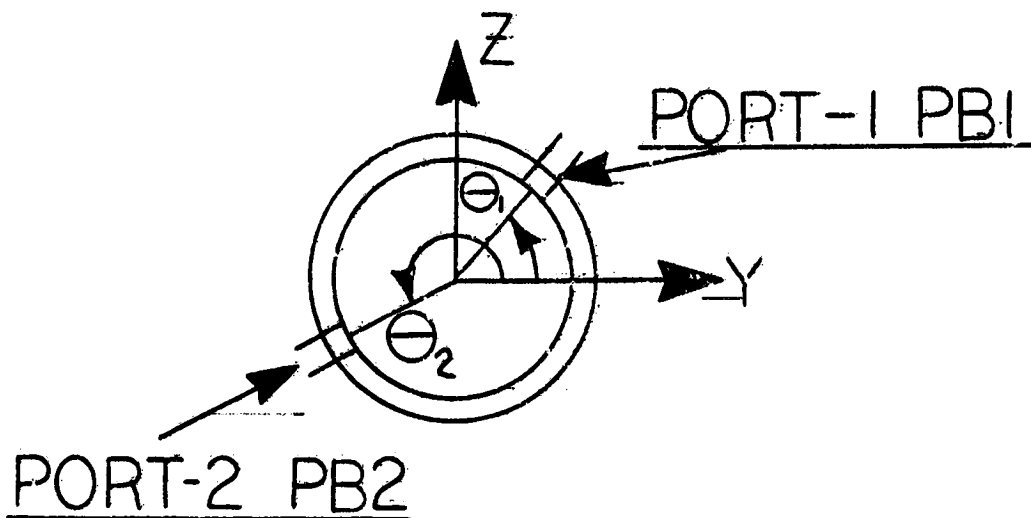
note	columns	variable	entry
(2)	1 - 10	TH2	Position angle of lubri- cant port-2 (degrées)
	11 - 20	PB1	Specific boundary pressure at port-1 (PSIA)
	21 - 30	PB2	Specific boundary pressure at port-2 (PSIA)
	31 - 35	NGRID	Number of finite-difference grid points per damper (odd)
	36 - 40	NSOLN	Solution type; EQ.1 long-bearing solution EQ.2 short-bearing solu- tion EQ.3 Fourier-Series 2-D convergent solution
	41 - 45	NPORT	Number of lubricant ports; GE.0 and LE.2

Card 2: (3f10.0,615) continued...

note	columns	variable	entry
	46 - 50	NFILM	Number of identical annuli for the bearing.
	51 - 55	KOFK	Bearing element stiffness matrix formation option; EQ.0 no stiffness reformation is performed EQ.1 stiffness reformation is performed.
	56 - 60	KOEC	Bearing element damping matrix formation option; EQ.0 no damping matrix formation is performed. EQ.1 damping matrix is performed.

NOTES/

- (1) Elements must be input in increasing element number order. If cards for elements (M+1, M+2, ..., M+J) are omitted, these "J" missing elements can be generated in the same way as ADINA does.
- (2) The position angles of lubricant ports must be input in degrees and are measured as shown below:



A2.2B.2 Element Nodes

Following the previous bearing information, each elements' nodes and their degrees of freedom are input. The degrees of freedom of the nodes which represent the bearing element may be different from those defined in master control card number "1". Accordingly, the degrees of freedom of the bearing element nodes are redefined as applied to this element individually.

Card 3: (1514)

note	columns	variable	entry
	1 - 4	M.	Bearing element number; GE.1 and LE. NPAR(2)
	5 - 8	IFLD	Number of nodes to describe this element
	9 - 12	KG	Node generation increment used to compute the node number for missing elements; EQ.0 default set to "1"
(1)	13 - 16	IDOFB(1)	Rotor node X-translation code; EQ.0: admissible EQ.1: deleted
(1)	17 - 20	IDOFB(2)	Rotor node Y-translation code
(1)	21 - 24	IDOFB(3)	Rotor node Z-translation code
(1)	25 - 28	IDOFB(4)	Rotor node X-rotation code
(1)	29 - 32	IDOFB(5)	Rotor node Y-rotation code
(1)	33 - 36	IDOFB(6)	Rotor node Z-rotation code

Card 3: (1514) continued...

note	columns	variable	entry
(1)	37 - 40	IDOFB(1)	Stator node X-translation code; EQ.0: admissible EQ.1: deleted.
(1)	41 - 44	IDOFB(2)	Stator node Y-translation code
(1)	45 - 48	IDOFB(3)	Stator node Z-translation code
(1)	49 - 52	IDOFB(4)	Stator node X-rotation code
(1)	53 - 56	IDOFB(5)	Stator node Y-rotation code
(1)	57 - 60	IDOFB(6)	Stator node Z-rotation code
(1)	61 - 64	IPLOTÉ	Control parameter for saving bearing responses for future plotting; EQ.0: bearing responses are not saved EQ.1: responses are saved on tape to be used by post-processor 2-D and 3-D plotting programs

Card 4: (215)

note	columns	variable	entry
(1)	1 - 5	NOD(1)	Global node number defining the ROTOR of the bearing element (nodal point 1)
(1)	6 - 10	NOD(2)	Global node number defining the STATOR of the bearing element (nodal point 2)

NOTES/

- (1) The bearing element is a two node member. One node defines the rotor (shaft) of the engine and the other node defines the stator (housing) of the bearing. During the input of the bearing nodes and their degrees of freedom, the node defining the ROTOR must come first (nodal point 1).

A2.3 Rotating Imbalance Loads

In bearing problems, the following card must be used to input the rotating loads data. It follows ADINA regular concentrated loads input cards.

Concentrated Rotating Loads Data. (3I5,2F10.0,15,2F10.0)

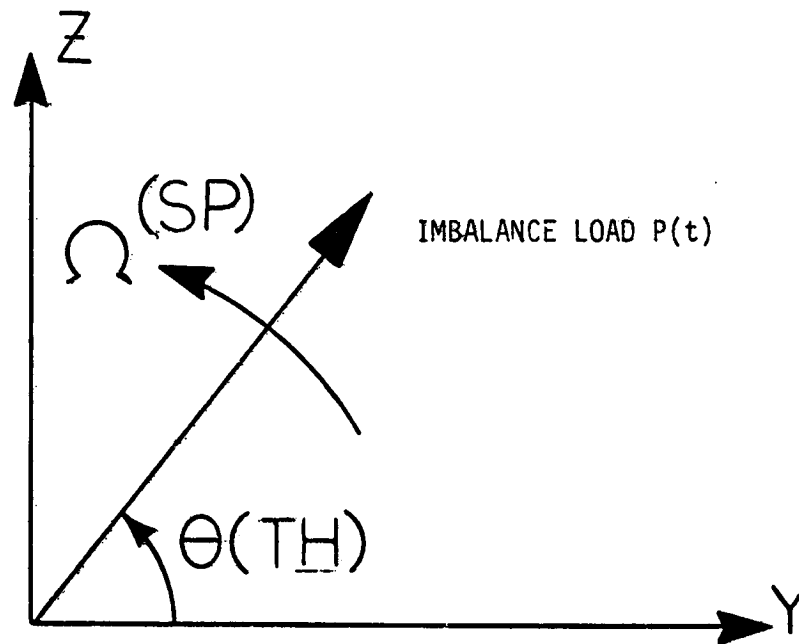
note	columns	variable	entry
(1)	1 - 5	NOD	Node number to which this rotating load is applied; GE.1 and LE. NUMNP
	6 - 10	IDIRN	Degrees of freedom number for this load component; Enter number "2"
	11 - 15	NCUR	Time function number that describes the time dependence $f(t)$ of the load; GE.1 and LE. NLCUR
	16 - 25	FAC	Function multiplier used to scale $f(t)$ to obtain the load at time "t"
	26 - 35	APTM	Arrival time of the forcing function
	36 - 40	KL	Node number increment used to generate loads at the missing nodes; EQ.0 no generation

Concentrated Rotating Loads Data (315,2F10.0,15,2F10.0)
continued ...

note	columns	variable	entry
(2)	41 - 50	TH	Initial phase angle of rotating imbalance load applied at node NOD
(2)	51 - 60	SP	Rotating speed of imbalance load applied at node NOD (RAD/SEC)

NOTES/

- (1) Input as many cards in this section as the number of nodes where rotating imbalance loads are applied; one card for each node.
- (2) The initial phase angle and speed of rotating load measured as shown in the following graph:



A2.4 Plotting Capabilities

During the output phase, the ADINA flow was interrupted to save the necessary bearing responses on a tape to be plotted in a separate plot job. The displacement velocity acceleration and bearing force components of bearing stators and rotors are stored. This is done in subroutine "WRITE". More details on the plotting capabilities are given in Appendices 4 and 5 which list the various plot routines.

APPENDIX 3: LIST OF ROTOR-BEARING-STATOR UPDATES TO ADINA CODE

The listing which follows includes:

- (1) Required JCL —
- (2) Rotor-Bearing-Stator Coding Update

```
//NASAPR JOB XXXXX,ZEID
***JOBPARM SKIP=YES,FORMS=1PWB
// EXEC SAVE, DATASET=CIDATA
XXSAVE PROC TEMP='&&', DATASET=DATASET, PRIME=20, SCNDRY=10, VOL=,
XX DISP=PASS, STATUS=MOD, UNIT=SYSDA, FILE=1, LABEL=SL,
XX DEN=4, BLK=6100, SPACE=TRK, RETPD=0
*****
*** PURPOSE: PROC TO TAKE INPUT CARDS TO A TEMPORARY DISK FILE
*** ID: AGIP0170 MODIFIED: 7 JUL 81 (DEH)
*****
XXSAVE EXEC PGM=AGIP0171, REGION=64K
XXSTEPLIB DD DSN=ACAD.CUMPLIB, DISP=SHR
XX DD DSN=SYS2.PL1LIB, DISP=SHR
XX DD DSN=SYS2.COMPILES, DISP=SHR
XXSYSPRINT DD SYSOUT=A
XXQOUTENT DD DSN=STEMP.&DATASET, DISP=(&STATUS,&DISP), VOL=SER=&VOL,
XX LABEL=(&FILE,&LABEL,RETPD=&RETPD), UNIT=&UNIT,
XX SPACE=(&SPACE,(&PRIME,&SCNDRY),RLSE),
XX DCB=(RECFM=FB,LRECL=80,BLKSIZE=&BLK,DEN=&DEN)
//SYSIN DD * GENERATED STATEMENT
// EXEC USERPAN, DATASET=BEAR
XXUSERPAN PROC TEMP='&&', DATASET=DATASET, PRIME=20, SCNDRY=10, DEN=4,
XX VOL=, LABEL=SL, FILE=1, DISP=PASS, STATUS=MOD, UNIT=SYSDA,
XX BLK=6100, RETPD=0, SPACE=TRK
*****
*** PURPOSE: ALLUWS ACCESS TO USER.PANVALET LIBRARY
*** INSTALLER: GARY SPONSELLER MODIFIED: 7 JUL 81 (DEH)
*****
XXGU EXEC PGM=PAN#1, REGION=128K
XXSTEPLIB DD DSN=PAN.LOADLIB, DISP=SHR
XXPANDD1 DD DSN=USER.PANVALET, DISP=SHR
XXPANDD2 DD DSN=STEMP.&DATASET, UNIT=&UNIT, DISP=(&STATUS,&DISP),
XX SPACE=(&SPACE,(&PRIME,&SCNDRY),RLSE), VOL=SER=&VOL,
XX DCB=(RECFM=FB,LRECL=80,BLKSIZE=&BLK,DEN=&DEN),
XX LABEL=(&FILE,&LABEL,RETPD=&RETPD)
XXPANDD9 DD UNIT=SYSDA, SPACE=(CYL,(3,1,1)),
XX DCB=(RECFM=FB,LRECL=80,BLKSIZE=&BLK,DSORG=PO)
XXSYSPRINT DD SYSOUT=A
XXSYSPUNCH DD SYSOUT=B
//SYSIN DD * GENERATED STATEMENT
// EXEC FORT, DATASET=BEAR
XXFORT PROC SOURCE=, OBJ=NUDECK, MAP=NU, LIST=NU, TEMP='&&',
XX DATASET=DATASET, ID=, CODE=EBCDIC, LINECNT=60, DISP=PASS
*****
*** PURPOSE: COMPILES A FORTRAN PROGRAM (GI COMPILER)
*** MODIFIED: 16 JUL 80 (ELW=W)
*****
XXFORT EXEC PGM=IGIFORT, REGION=128K,
XX PARM=( 'SOURCE, SOURCE, OBJ, &MAP, MAP, &LIST, LIST, &ID, ID, '
XX '&CODE, LINECNT=&LINECNT' )
XXSTEPLIB DD DSN=SYS2.FTGICUMP, DISP=SHR
XXSYSPRINT DD SYSOUT=A
XXSYSPUNCH DD SYSOUT=B
XXSYSLIN DD DSN=&LOADSET, UNIT=SYSDA, SPACE=(400,(1000,200)),
XX DCB=BLKSIZE=400, DISP=(MOD,PASS)
XXSYSIN DD DSN=STEMP.&DATASET, DISP=(OLD,&DISP)
// EXEC GOFORTL, LIB=MELIB2, PRG=ADINA, GOSIZE=950K, DISK=
XXGOFORTL PRJC GOSIZE=128K, EP=MAIN, STATUS=MOD, DISP=PASS, FILE=1,
XX LIB=DUMMY, PRG=DUM4Y, LABEL=SL, UNIT=SYSDA, VOL=, DEN=4,
XX UNIT=SAVED, SYSLIB=DUMMY, TEMP='&&', CC=4, BLK=6100,
XX DATASET=DATASET, MAP=, DISK='DUMMY,', PREC=SP,
```

ORIGINAL PAGE IS
OF POOR QUALITY

```

XX.          PRIME=20,SCNDRY=10,PL0T='DUMMY',.OUTPUT=,PREFIX=PLIB.
*****
***          PURPOSE: EXECUTES A FORTRAN OBJECT PROGRAM RESIDING
***          IN A DISK PDS LIBRARY.
***          DATE MODIFIED: 7 JUL 81 (DEH)
*****
XXGO        EXEC. PGM=LOADER,REGION=&GOSIZE,COND=(&CC,LI),
XX          PARM=*&MAP,MAP,SIZE=&GOSIZE,EP=&EP
XXSYSL0UT  DD  SYSOUT=A
//SYSLIN   DD  DSN=&LOADSET,DISP=OLD
X/SYSLIN   DD  DSN=&PREFIX..&LIB(&PR0G),DISP=SHR.
//          DD  DSN=&PREFIX..&LIB(&PR0G),DISP=SHR
XXSYSLIB   DD  DSN=SYS2.FTMILIB,DISP=SHR.
XX          DD  DSN=PLIB.&SYSLIB,DISP=SHR
XX          DD  DSN=IMSL.FS3D0002.&PREC,DISP=SHR
XX          DD  DSN=SYS2.SUBPGM,DISP=SHR.
XX          DD  DSN=ACAD.SUBLIB,DISP=SHR
//FI01F001 DD  UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(3,3)),
//          DCB=(RECFM=VBS,LRECL=3152,BLKSIZE=3156,BUFNO=1)
X/FT01F001 DD  &DISK,UNIT=SYSDA,SPACE=(CYL,(2,1)),
XX          DCB=(RECFM=VBS,LRECL=1000,BLKSIZE=3156)
//FT02F001 DD  UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(3,3)),
//          DCB=(RECFM=VBS,LRECL=3152,BLKSIZE=3156,BUFNO=1)
X/FT02F001 DD  &DISK,UNIT=SYSDA,SPACE=(CYL,(2,1)),DCB=*.FT01F001
//FT03F001 DD  UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(1,1)),
//          DCB=(RECFM=VBS,LRECL=3152,BLKSIZE=3156,BUFNO=1)
X/FT03F001 DD  &DISK,UNIT=SYSDA,SPACE=(CYL,(2,1)),DCB=*.FT01F001
//FT04F001 DD  UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(3,3)),
//          DCB=(RECFM=VBS,LRECL=3152,BLKSIZE=3156,BUFNO=1)
X/FT04F001 DD  DDNAME=&UNIT4.
XXSAVE()   DD  DSN=&TEMP.&DATASET,DISP=(&STATUS,&DISP),UNIT=&UNIT,
XX          SPACE=(TRK,(&PRIME,&SCNDRY),RLSE),VOL=SER=&VOL.
XX          DCB=(RECFM=FB,LRECL=80,BLKSIZE=&BLK,DEN=&DEN).
XX          LABEL=(&FILE,&LABEL)
XXWORK     DD  &DISK,UNIT=SYSDA,SPACE=(CYL,(2,1)),DCB=*.FT01F001
//FT05F001 DD  DDNAME=SYSIN
X/FT05F001 DD  DDNAME=SYSIN
//FT06F001 DD  SYSOUT=A
X/FT06F001 DD  SYSOUT=(A,&OUTPUT),DCB=RECFM=FA
//FT07F001 DD  UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(3,3)),
//          DCB=(RECFM=VBS,LRECL=3152,BLKSIZE=3156,BUFNO=1)
X/FT07F001 DD  SYSOUT=B
XXPL0TTAPE DD  &PLUT,SYSOUT=G
//FT08F001 DD  UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(3,3)),
//          DCB=(RECFM=VBS,LRECL=3152,BLKSIZE=3156,BUFNO=1)
//FT09F001 DD  UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(1,1)),
//          DCB=(RECFM=VBS,LRECL=3152,BLKSIZE=3156,BUFNO=1)
//FT10F001 DD  UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(3,3)),
//          DCB=(RECFM=VBS,LRECL=3152,BLKSIZE=3156,BUFNO=1)
//FT11F001 DD  UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(3,3)),
//          DCB=(RECFM=VBS,LRECL=3152,BLKSIZE=3156,BUFNO=1)
//FT12F001 DD  UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(3,3)),
//          DCB=(RECFM=VBS,LRECL=3152,BLKSIZE=3156,BUFNO=1)
//FT13F001 DD  UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(3,3)),
//          DCB=(RECFM=VBS,LRECL=3152,BLKSIZE=3156,BUFNO=1)
//FT20F001 DD  DSN=USER.LISTN1,DISP=(NEW,CATLG),
//          SPACE=(TRK,(150,20),RLSE),
//          UNIT=SYSDA,VOL=SER=ACAD01,
//          DCB=(RECFM=FB,BLKSIZE=12960,LRECL=80)
//SYSIN DD  DSN=&CICDATA,DISP=(OLD,DELETE)

```


ORIGINAL PAGE IS
OF POOR QUALITY

```

IEF6531 SUBSTITUTION JCL - DSN=88CIDA,DISP=(MOD,PASS),VOL=SER=,
IEF6531 SUBSTITUTION JCL - LABEL=(1,SL,RETPD=0),UNIT=SYSDA,
IEF6531 SUBSTITUTION JCL - SPACE=(TRK,(20,10),RLSE),
IEF6531 SUBSTITUTION JCL - DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160,DEN=4),
IEF6531 SUBSTITUTION JCL - DSN=88BEAR,UNIT=SYSDA,DISP=(MOD,PASS),
IEF6531 SUBSTITUTION JCL - SPACE=(TRK,(20,10),RLSE),VOL=SER=,
IEF6531 SUBSTITUTION JCL - DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160,DEN=4),
IEF6531 SUBSTITUTION JCL - LABEL=(1,SL,RETPD=0)
IEF6531 SUBSTITUTION JCL - DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160,DSORG=PO)
IEF6531 SUBSTITUTION JCL - PARM=('SOURCE,NOCHECK,NUMAP,NOLIST,ID,')
IEF6531 SUBSTITUTION JCL - *EBCDIC,LINECNT=60*
IEF6531 SUBSTITUTION JCL - DSN=88BEAR,DISP=(OLD,PASS)
IEF6531 SUBSTITUTION JCL - PGM=LOADER,REGION=950K,COND=(4,LT),
IEF6531 SUBSTITUTION JCL - PARM=*MAP,SIZE=950K,EP=MAIN*
IEF6531 SUBSTITUTION JCL - DSN=PLIB.MELIB2(ADINA),DISP=SHR
IEF6531 SUBSTITUTION JCL - DSN=PLIB.MELIB2(ADINA),DISP=SHR
IEF6531 SUBSTITUTION JCL - DSN=PLIB.DUMMY,DISP=SHR
IEF6531 SUBSTITUTION JCL - DSN=IMSL.FS30002.SP,DISP=SHR
IEF6531 SUBSTITUTION JCL - UNIT=SYSDA,SPACE=(CYL,(2,1)),
IEF6531 SUBSTITUTION JCL - UNIT=SYSDA,SPACE=(CYL,(2,1)),DCB=*.FT01F00L
IEF6531 SUBSTITUTION JCL - UNIT=SYSDA,SPACE=(CYL,(2,1)),DCB=*.FT01F00L
IEF6531 SUBSTITUTION JCL - DDNAME=SAVED
IEF6531 SUBSTITUTION JCL - DSN=88DATASET,DISP=(MOD,PASS),UNIT=SYSDA,
IEF6531 SUBSTITUTION JCL - SPACE=(TRK,(20,10),RLSE),VOL=SER=,
IEF6531 SUBSTITUTION JCL - DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160,DEN=4),
IEF6531 SUBSTITUTION JCL - LABEL=(1,SL)
IEF6531 SUBSTITUTION JCL - UNIT=SYSDA,SPACE=(CYL,(2,1)),DCB=*.FT01F001
IEF6531 SUBSTITUTION JCL - SYSOUT=(A,),DCB=RECFM=EA
IEF6531 SUBSTITUTION JCL - DUMMY,SYSOUT=G
IEF0481 INVALID DISP FIELD= PASS SUBSTITUTED

```

ALLOC. FOR NASAPR SAVE

172 ALLOCATED TO STEPLIB

161 ALLOCATED TO

161 ALLOCATED TO

161 ALLOCATED TO SYS00062

JES2 ALLOCATED TO SYSPRINT

26E ALLOCATED TO OUTPUT

JES2 ALLOCATED TO SYSIN

NASAPR SAVE - STEP WAS EXECUTED - COND CODE 0000

ACAD.COMPLIB

KEPT

VOL SER NOS= ACAD02.

SYS2.PL1LIB

KEPT

VOL SER NOS= SPOOL2.

SYS2.CMPILERS

KEPT

VOL SER NOS= SPOOL2.

SYSCTLG.VSPOOL2

KEPT

VOL SER NOS= SPOOL2.

JES2.JOB00137.SU0103

SYSOUT

SYS81229.T091819.RA000.NASAPR.CIDA

PASSED

VOL SER NOS= WORK03.

JES2.JOB00137.S10101

SYSIN

STEP /SAVE / START 81229.0918

STEP /SAVE / STOP 81229.0918 CPU

0MIN 00.41SEC SRB

0MIN 00.06SEC

ALLOC. FOR NASAPR GO

269 ALLOCATED TO STEPLIB

161 ALLOCATED TO SYS00064

172 ALLOCATED TO PAND01

26E ALLOCATED TO PAND02

166 ALLOCATED TO PAND09

```
IEF2371 JES2 ALLOCATED TO SYSPRINT
IEF2371 JES2 ALLOCATED TO SYSPUNCH..
IEF2371 JES2 ALLOCATED TO SYSIN
IEF1421 NASAPR GU - STEP WAS EXECUTED - COND CODE 0000
IEF2851 PAN.LOADLIB KEPT___
IEF2851 VOL SER NUS= STOR02 KEPT___
IEF2851 SYSCTLG.VSPOOL2 KEPT___
IEF2851 VOL SER NUS= SPOOL2. KEPT___
IEF2851 USER.PANVALET KEPT___
IEF2851 VOL SER NUS= ACAD02. PASSED..
IEF2851 SYS81229.T091819.RA000.NASAPR.BEAR. PASSED..
IEF2851 VOL SER NUS= WORK03. DELETED
IEF2851 SYS81229.T091819.RA000.NASAPR.R0000001 DELETED
IEF2851 VOL SER NUS= WORK01.
IEF2851 JES2.JOB00137.S00104. SYSOUT
IEF2851 JES2.JOB00137.S00105. SYSOUT..
IEF2851 JES2.JOB00137.SI0102. SYSIN
IEF3731 STEP /GU / START 81229.0918
IEF3741 STEP /GO / STOP 81229.0918 CPU 0MIN 08.37SEC SRB
IEF2361 ALLOC. FOR NASAPR FORT
IEF2371 162 ALLOCATED TO STEPLIB
IEF2371 161 ALLOCATED TO SYS00066-
IEF2371 JES2 ALLOCATED TO SYSPRINT
IEF2371 JES2 ALLOCATED TO SYSPUNCH
IEF2371 166 ALLOCATED TO SYSLIN-
IEF2371 26E ALLOCATED TO SYSIN
IEF1421 NASAPR FORT - STEP WAS EXECUTED - COND CODE 0000
IEF2851 SYS2.FTGICOMP KEPT
IEF2851 VOL SER NUS= STOR03. KEPT___
IEF2851 SYSCTLG.VSPOOL2 KEPT___
IEF2851 VOL SER NUS= SPOOL2.
IEF2851 JES2.JOB00137.S00106. SYSOUT
IEF2851 JES2.JOB00137.S00107. SYSOUT
IEF2851 SYS81229.T091819.RA000.NASAPR.LOADSET. PASSED
IEF2851 VOL SER NUS= WORK01. PASSED___
IEF2851 SYS81229.T091819.RA000.NASAPR.BEAR
IEF2851 VOL SER NUS= WORK03.
IEF3731 STEP /FORT / START 81229.0918
IEF3741 STEP /FORT / STOP 81229.0922 CPU 1MIN 10.07SEC SRB
IEF2361 ALLOC. FOR NASAPR GU
IEF2371 JES2 ALLOCATED TO SYSLOUT
IEF2371 166 ALLOCATED TO SYSLIN-
IEF2371 164 ALLOCATED TO
IEF2371 161 ALLOCATED TO SYS00068
IEF2371 162 ALLOCATED TO SYSLIB
IEF2371 164 ALLOCATED TO
IEF2371 172 ALLOCATED TO
IEF2371 269 ALLOCATED TO
IEF2371 172 ALLOCATED TO
IEF2371 166 ALLOCATED TO FT01F001
IEF2371 26E ALLOCATED TO FT02F001
IEF2371 26E ALLOCATED TO FT03F001
IEF2371 166 ALLOCATED TO FT04F001
IEF2371 166 ALLOCATED TO SAVED
IEF2371 26E ALLOCATED TO WORK
IEF2371 26E ALLOCATED TO FT05F001
IEF2371 JES2 ALLOCATED TO FT06F001
IEF2371 26E ALLOCATED TO FT07F001
IEF2371 DMY ALLOCATED TO PLOTTAPE
IEF2371 166 ALLOCATED TO FT08F001
IEF2371 166 ALLOCATED TO FT09F001
```

ORIGINAL PAGE IS
OF POOR QUALITY

A3-6

```
IEF2371 166 ALLOCATED TO FT10F001
IEF2371 26E ALLOCATED TO FT11F001
IEF2371 26E ALLOCATED TO FT12F001
IEF2371 166 ALLOCATED TO FT13F001
IEF2371 164 ALLOCATED TO FT20F001
IEF1421 NASAPR GO = STEP WAS EXECUTED = COND. CODE 0000
IEF2851 JES2.JOB00137.S00108 SYSOUT
IEF2851 SYS81229.T091819.RA000.NASAPR.LOADSET... PASSED
IEF2851 VOL SER NOS= WORK01.
IEF2851 PLIB.MELID2 KEPT...
IEF2851 VOL SER NOS= ACAD01.
IEF2851 SYSCTLG.VSPOOL2 KEPT
IEF2851 VOL SER NOS= SPOOL2.
IEF2851 SYS2.FTMLIB KEPT...
IEF2851 VOL SER NOS= STOR03.
IEF2851 PLIB.DUMMY KEPT...
IEF2851 VOL SER NOS= ACAD01.
IEF2851 IMSL.FS3D0002.SP KEPT...
IEF2851 VOL SER NOS= ACAD02.
IEF2851 SYS2.SUBPGM KEPT...
IEF2851 VOL SER NOS= STOR02.
IEF2851 ACAD.SUBLIB KEPT...
IEF2851 VOL SER NOS= ACAD02.
IEF2851 SYS81229.T091819.RA000.NASAPR.R000J002 DELETED.
IEF2851 VOL SER NOS= WORK01.
IEF2851 SYS81229.T091819.RA000.NASAPR.R000J003 DELETED
IEF2851 VOL SER NOS= WORK03.
IEF2851 SYS81229.T091819.RA000.NASAPR.R0000004 DELETED...
IEF2851 VOL SER NOS= WORK03.
IEF2851 SYS81229.T091819.RA000.NASAPR.R000J005 DELETED.
IEF2851 VOL SER NOS= WORK01.
IEF2851 SYS81229.T091819.RA000.NASAPR.DATASET PASSED...
IEF2851 VOL SER NOS= WORK01.
IEF2851 SYS81229.T091819.RA000.NASAPR.R0000006 DELETED...
IEF2851 VOL SER NOS= WORK03.
IEF2851 SYS81229.T091819.RA000.NASAPR.CIDATA DELETED...
IEF2851 VOL SER NOS= WORK03.
IEF2851 JES2.JOB00137.S00109 SYSOUT
IEF2851 SYS81229.T091819.RA000.NASAPR.R0000007 DELETED
IEF2851 VOL SER NOS= WORK03.
IEF2851 SYS81229.T091819.RA000.NASAPR.R0000008 DELETED...
IEF2851 VOL SER NOS= WORK01.
IEF2851 SYS81229.T091819.RA000.NASAPR.R0000009 DELETED
IEF2851 VOL SER NOS= WORK01.
IEF2851 SYS81229.T091819.RA000.NASAPR.R0000010 DELETED...
IEF2851 VOL SER NOS= WORK01.
IEF2851 SYS81229.T091819.RA000.NASAPR.R0000011 DELETED.
IEF2851 VOL SER NOS= WORK03.
IEF2851 SYS81229.T091819.RA000.NASAPR.R0000012 DELETED...
IEF2851 VOL SER NOS= WORK03.
IEF2851 SYS81229.T091819.RA000.NASAPR.R0000013 DELETED
IEF2851 VOL SER NOS= WORK01.
IEF2851 USER.LISTN1 CATALOGED
IEF2851 VOL SER NOS= ACAD01.
IEF3731 STEP /GB / START 81229.0922
IEF3741 STEP /GB / STOP 81229.0923 CPU - 0MIN 20.85SEC SRB
IEF2371 26E ALLOCATED TO SYS0001
IEF2851 SYS81229.T092345.RA000.NASAPR.R0000001 KEPT...
IEF2851 VOL SER NOS= WORK03.
IEF2851 SYS81229.T091819.RA000.NASAPR.BEAR... DELETED
IEF2851 VOL SER NOS= WORK03.
```

ORIGINAL PAGE IS
OF POOR QUALITY

A3-7

```
IEF237I 166 ALLOCATED TO SYS00003
IEF285I SYS81229.T092345.RA000.NASAPR.R0000003      KEPT
IEF285I VOL SER NOS= WORK01.
IEF285I SYS81229.T091819.RA000.NASAPR.LOADSET      DELETED
IEF285I VOL SER NOS= WORK01.
IEF237I 166 ALLOCATED TO SYS00005
IEF285I SYS81229.T092345.RA000.NASAPR.R0000005      KEPT
IEF285I VOL SER NOS= WORK01.
IEF285I SYS81229.T091819.RA000.NASAPR.DATASET      DELETED
IEF285I VOL SER NOS= WORK01.
IEF375I JOB /NASAPR / START 81229.0918.
IEF376I JOB /NASAPR / STOP 81229.0923 CPU 1MIN 39.70SEC SRB
```

MAIN DATE = 81229 09/18/84

C DATA SET AAMAIN AT LEVEL TMP AS OF 08/17/81
C *CDC* #DECK OVL00
C *CDC* #OVERLAY (ADINA,0,0)
C *CDC* #DECK ADINA
C *UNI*)FOR,IS N,ADINA, H,ADINA
C *CDC* PROGRAM ADINA (INPUT, OUTPUT, PUNCH, TAPES=INPUT, TAPES =
C *CDC* 1 OUTPUT, TAPE1, TAPE2, TAPE3, TAPE4,
C *CDC* 2 TAPE7, TAPE8, TAPE9, TAPE10, TAPE11,
C *CDC* 3 TAPE12, TAPE13, TAPE24, TAPE59, TAPE60, TAPES1,
C *CDC* 4 TAPE62, TAPE63)

.....
T A P E A L L O C A T I O N

- TAPE 1 STORES LINEAR ELEMENT GROUP DATA
(WHEN NEGL.NE.0 ONLY)
- TAPE 2 STORES NONLINEAR ELEMENT GROUP DATA
- TAPE 3 STORES EXTERNALLY APPLIED LOADS
- TAPE 4 STORES THE LINEAR STIFFNESS MATRIX
(WHEN NEGL.NE.0 ONLY)
- TAPES 5,6 ARE INPUT,OUTPUT TAPES
- TAPE 7 STORES EFFECTIVE LINEAR COEFFICIENT MATRIX
- TAPE 8 STORES SEQUENTIALLY
(1) ID ARRAY
(2) DURING INPUT
INITIAL DISP, VEL, ACC VECTORS
ON OUTPUT
FINAL DISP, VEL, ACC VECTORS
AND NONLINEAR ELEMENT GROUP DATA FOR RESTART
- TAPE 9 STORES NODAL COORDINATES FOR PRESSURE LOAD
CALCULATIONS
- TAPE 10 STORES
O = L FACTORS OF EFFECTIVE LINEAR OR NONLINEAR
STIFFNESS MATRIX IN TIME INTEGRATION
- TAPE 11 STORES SEQUENTIALLY
(1) CONSISTENT MASS MATRIX (IF LMASS.EQ.2)
(2) DAMPING VECTOR
- TAPE 12 STORES THE EFFECTIVE NONLINEAR STIFFNESS MATRIX
IN THE TIME INTEGRATION - OR -
THE NONLINEAR STIFFNESS MATRIX FOR EIGENSYSTEM SOLUTION
- TAPE 13 STORES MODE SHAPES AND CIRCULAR FREQUENCIES
(IF FREQUENCIES SOLUTION WAS REQUESTED)

ORIGINAL FILED
OF POOR QUALITY

MAIN

DATE = 81229

09/18/44

RUSS.-TDFE, THDFE, BMEL, BANDET, SAVARC *

PRIOR AND AFTER A RANDOM ACCESS READ/WRITE THE FOLLOWING CARDS
HAVE BEEN INCLUDED

* * * * * R A N D O M A C C E S S * * * * *

CREATE RANDOM FILES 2, 10 WITH NUMBER INDEX

* * * * * R A N D O M A C C E S S * * * * *

NR(1)=190
NR(2)=190
LR(1)=3000
LR(2)=3000
DEFINE FILE 10 (190,3000,U,NREC10)
DEFINE FILE 2 (190,3000,U,NREC2)

* * * * * R A N D O M A C C E S S * * * * *

MTOT=40000
MTOT3 = 5000
CDC LTWO=1
ITWO=2
NBCST=MTOT
NBCST=1

200 NUMEST=0
MAXEST=0
NBCEL=0
NREAD=2
NUMREF=0
ITE=0
KPRI=1
KSTEP=0
IND=0
ICOUNT=2

I N P U T P H A S E

CALL SECOND (TIM1)

N0=1 ← NBCST

STORAGE IS ALLOCATED IN ADINI BECAUSE THE REQUIRED
VARIABLES ARE NOT KNOWN YET

CDC CALL OVERLAY (SHADINA,1,0,6HRECALL)
CALL ADINI

MAIN

DATE = 81229

09/18/44

OBTAIN ELEMENT INFORMATION S

CLEAR ARRAY FOR CALCULATION OF COLUMN HEIGHTS

NN=NS + NEQ
 DO 2 I=NS,NN
 2 IA(I)=0
 N6=NS + NEQ

INITIALIZE TEMPERATURE ARRAY

IF (ITP96.EQ.0) GO TO 14
 N6A=NS + NEQ + 1
 N6B=N6A
 N6=N6A + (NUMNP+1)*ITWO
 NN=N6 - 1
 READ (24) (A(I),I=N6A,NN)
 CALL TCHECK (A(N6A),TSTART)
 BACKSPACE 2+

14 CALL ELCAL

CALL SECOND (TIM2)

STORE MAXA ARRAY

IF (IDPE.NE.3) CALL ADDRESS (A(N1),A(N5))

STORAGE CALCULATIONS

TEST FOR AVAILABILITY OF HIGH SPEED STORAGE AND CALCULATE
 MAXIMUM BLOCKSIZE, NUMBER OF BLOCKS, AND BLOCK COUPLING

1. STORAGE FOR LOAD VECTOR CALCULATIONS

MSTORE=(NEQ+1) * NOOF*NUMNP + (NEQ + 2*NPTM + NTFN*NSTE + 2*NLJAD)
 *ITWO + 4*NLJAD + NTFN*ITWO
 IF (IDGRAV.EQ.1) MSTORE=MSTORE + 3*ITWO + NEQ * NEQ*ITWO
 WRITE (6,2200)
 CALL SIZE(MSTORE)

2. STORAGE FOR MATRIX ASSEMBLAGE PHASE AND TIME INTEGRATION

CENTRAL DIFFERENCE METHOD

IF (IDPE.NE.3) GO TO 5
 ISV=(IVL + JVC + 1)/2
 ISA=(IAC + JAC + 1)/2
 MSTORE=NEQNE + (J*ISV+ISA)*NEQ*ITWO + ITEMP*(NUMNP + 1)*ITWO
 NTEMP=NOOF*NUMNP
 IF (NTEMP.LT.NEQ*ITWO) NTEMP=NEQ*ITWO
 MSTORE=MSTORE + MTEMP * MAXES1 + NBCEL
 WRITE (6,2202)
 CALL SIZE (MSTORE)
 NBLUCK=1

ORIGINAL PAGE IS
OF POOR QUALITY

MAIN

DATE = 81229.

09/18/44.

ISTOH=0
GO TO 45.

STATIC ANALYSIS AND IMPLICIT TIME INTEGRATION

5 MSTORE=(NEQ + 1) + 3*NEQ*ITWO + ITEMP*(NUMNP + 1)*ITWO + MAXEST
+ NBCEL
1 MTEMP=2*NEQ*ITWO
IF (MTEMP.LT.(NDUF*NUMNP)). MTEMP=NDUF*NUMNP
MSTORE=MSTORE + MTEMP
IF (ISTAT.EQ.1) MSTORE=MSTORE + 2*NEQ*ITWO
IF (IMASS.EQ.1) MSTORE=MSTORE + NEQ*ITWO
N1A=N1 + NEQ + 1
N1B=N1A + NEQ
IBLOCK=4
NBLUCK=1
10 MELST=IBLOCK*NEGL + 2*IBLOCK
MELST=MELST + (IEIG + 1)*IBLOCK + 1
ISTORL=(MTOT - MSTORE - MELST)/ITWO
IF (ISTOTE.GT.0) ISTORL=ISTOTE
IF (ISTORL.GT.0) GO TO 15
WRITE (6,2203)
STOP

15 CALL SBLOCK (A(N1),A(N1A),A(N1B),ISTORL,NBLOCK,NEQ,NWK,ISTOH)

IF (ISTOTE.GT.0) GO TO 20
IF (NBLOCK.LE.IBLOCK) GO TO 20
IBLOCK=IBLOCK*2
IF (IBLOCK.LT.1000) GO TO 10
WRITE (6,2204)
STOP

20 MAM=NWK/NEQ + 1

3. SPECIAL CASE IF ONE BLOCK CASE AND CONSISTENT MASS MATRIX
IS USED

IF (IMASS.NE.2 .OR. NBLOCK.GT.1) GO TO 30
MM=2*ISTOH
IF (MM.LE.ISTORL) GO TO 30
NBLOCK=2
CALL SBLOCK (A(N1),A(N1A),A(N1B),ISTORL,NBLOCK,NEQ,NWK,ISTOH)

4. STORAGE FOR FREQUENCY ANALYSIS

30 IF (IEIG.EQ.0) GO TO 36
MSTORE=9*NEQ*ITWO + NEQ + 1000
IF (IMASS.EQ.1) MSTORE=MSTORE + NEQ*ITWO
MM=(MTOT - MSTORE)/ITWO
IF (NBLOCK.GT.1 .OR. IMASS.EQ.2) MM=MM/2
IF (MM.GE.ISTOH) GO TO 36
IF (NBLOCK.GT.1 .OR. IMASS.EQ.2) MM=2*MM
NBLOCK=2
CALL SBLOCK (A(N1),A(N1A),A(N1B),MM,NBLOCK,NEQ,NWK,ISTOH)

WRITE TOTAL SYSTEM DATA

36 WRITE (6,2210) NEQ,NWK,MA,MAM,ISTOH,NBLOCK,MTOT

ORIGINAL PAGE IS
OF POOR QUALITY

MAIN

DATE = 81229

09/18/44

```

WRITE (6,2220)
NN=N1A + NBLOCK - 1
WRITE (6,2230) (I,I=1,NBLOCK)
WRITE (6,2240) (IA(I),I=N1A,NN)
NN=N1B + NBLOCK - 1
WRITE (6,2250) (IA(I),I=N1B,NN)
NN=N1A + NBLOCK
DO 40 I=1,NBLOCK
40 IA(NN+I-1)=IA(N1B+I-1)

```

C
C
C
C
C

A S S E M B L A G E O F L I N E A R M A T R I C E S

```

N1B=NN
N1C=N1B + NBLOCK
N1D=N1C + NBLOCK*NEGNL
IF (NBLOCK.EQ.1) N1D=N1C
N2=N1D + (IEIG + 1)*NBLOCK + 1
N3=N2 + ISTOP*ITWO
N4=N3 + ISTOP*ITWO
IF (NBLOCK.EQ.1 .AND. IMASS.LT.2) N4=N3
45 IF (IOPE.EQ.3) N4=N1
N5=N4 + NEQ*ITWO
N6=N5 + NEQ*ITWO
IF (IMASS.EQ.0) N6=N4
IF (ISTAT.EQ.0 .AND. IDGRAV.EQ.1) N6=N5
N7=N6 + MAXEST + NBCEL
WRITE (6,2260)
CALL SIZE(N7)
CALL SECOND(TIM3)
IF (MODEX.GT.0) GO TO 50
IND=2

```

C
C
C
C
C
C
C
C
C

CREATE RANDOM ACCESS FILE 10 WITH ASSOCIATED RECORD NUMBER INDEX

```

* * * * *   R A N D O M   A C C E S S   * * * * *
50 NBLOC1=(IEIG + 1)*NBLOCK + 1
*CDC*      IF (IOPE.NE.3) CALL OPENMS (10,IA(N1D),NBLOC1,0)

```

```

* * * * *   R A N D O M   A C C E S S   * * * * *
IF (MODEX.EQ.0) GO TO 60
IND=1
IREF=0
CALL ASSEM (A(N1),A(N2),A(N3),A(N4),A(N5),A(N6),A(N1A),A(N4),
1 A(N1C),ISTOP,NBLOCK)

```

C
C
C
C
C
C
C
C
C

C A L C U L A T E A N D S T O R E L O A D V E C T O R S

```

*CDC*      60 CALL OVERLAY (SHADINA,1,0,SHRECALL)
60 CALL ADINI

```

C A L C U L A T E S T O R A G E L O C A T I O N S F O R T I M E I N T E G R A T I O N

MAIN

DATE = 81229

09/18/44

```
CALL SECOND(TIM4)
IF (IOPE.EQ.3) N2=N1 + NEQ*ITWO
N3=N2 + NEQ*ITWO
N4=N3 + NEQ*ITWO
N4A=N4 + ISTOH*ITWO
N4B=N4A + ISTOH*ITWO
IF (NBLOCK.EQ.1) N4B=N4A
N5=N4B + NEQ*ITWO
IF (IOPE.EQ.3) N5=N4
N6=N5 + NEQ*ITWO
N6A=N6 + NEQ*ITWO
NN1=NOUF*NUMNP
NN2=2*NEQ*ITWO
IF (NN1.GT.NN2) N6A=N5 + NN1
IF (IOPE.EQ.3) N6=N5
NN2=NEQ*ITWO
IF (NN2.GT.NN1) NN1=NN2
IF (IOPE.EQ.3) N6A=N6 + NN1
N6B=N6A + (ITEMPR=1)*(NUMNP+1)*ITWO
N7=N6B + (NUMNP+1)*ITWO
N8=N7 + NEQ*ITWO
IF (IOPE.EQ.3 .AND. ISV.EQ.0) N9=N7
N9=N8 + NEQ*ITWO
N10=N9 + NEQ*ITWO
IF (IOPE.EQ.3) N10=N7 + (ISV + ISA)*NEQ*ITWO
IF (.IMASS.EQ.2) N10=N9
IF (ISTAT.EQ.0) N10=N7
N11=N10 + MAXEST + N0CEL
WRITE (6,2283)
CALL SIZE (N11)
```

INITIALIZE TEMPERATURE ARRAY

```
IF (ITEMPR.EQ.0) GO TO 64
NN=N6A + (NUMNP+1)*ITWO - 1
READ (24) (A(I), I=N6A, NN)
```

WRITE INITIALIZED DISPLACEMENTS, VELOCITIES, AND ACCELERATIONS
(OR STARTING DISPL/VEL/ACC IE THIS IS A RESTART JOB)

64 CALL WRITE (/ (N1), A(N2), A(N7), A(N8), A(N5), NEQ, NOUF, 1)

IF THIS IS A RESTART JOB TRANSFER NONLINEAR ELEMENT GROUP
DATA TO TAPE 2

```
IF (MODEX.NE.2) GO TO 82
CALL RSTART (A(N1), A(N2), A(N7), A(N8), A(N10), NEQ, IOPE, 2)
```

F R E Q U E N C Y S O L U T I O N

```
82 CALL SECOND (TIM5)
IF (IEIG.EQ.3) GO TO 89
IND=3
TIME=YSTART + DT
CALL ASSEM (A(N1), A(N8), A(N6A), A(N2), A(NJ), A(N10), A(N1A), A(N6B),
A(N1C), ISTOH, NBLOCK)
```

I
I
I

ORIGINAL PAGE IS
OF POOR QUALITY

MAIN

DATE = 81229

09/13/44

09 CALL SECOND (TIM6)

T I M E I N T E G R A T I O N

TSUM1=0.
TSUM2=0.
TSUM3=0.
TSUM4=0.
TSUM5=0.
TSUM6=0.
CALL SECOND (TIM7)
TIM8=TIM7
IF (MODEX.GT.0) GO TO 88
WRITE(6,2030)
GO TO 190

88 IF (NSTE.EQ.0) GO TO 190

TIME=TSTART
TIMEP=TSTART
REWIND 3
IND=4
KPRINT=0
NUMP1=(NUMP+1)*IT#0

IN CASE OF LINEAR ANALYSIS TRIANGULARIZE EFFECTIVE LINEAR
STIFFNESS MATRIX (THE TRIANGULAR FACTORS REMAIN IN CORE
PROVIDED THAT

1. LINEAR ANALYSIS
2. ONE BLOCK CASE
3. IMPLICIT TIME INTEGRATION SCHEME IS USED)

CALL SECOND (TIM7)
IF (KLIN.GT.0 .OR. LOPE.EQ.3) GO TO 94
NTAPE=4
IF (ISTAT.EQ.1) NTAPE=7
CALL COLSOL (A(N1),A(N1A),A(N1B),A(N4),A(N4A),A(N4B),A(NJ),
1 NEQ,NBLOCK,ISTOH,NTAPE,10,1)

94 CALL SECOND (TIM8)
IF (LOPE.EQ.3) GO TO 100

LUMPED MASS MATRIX IS TAKEN INTO CORE AND NODAL DAMPING VECTOR
IS STORED AS FIRST RECORD (IMPLICIT TIME INTEGRATION).

IF (IMASS.NE.1) GO TO 100
REWIND 11
NN=N10 - 1
READ (11) (A(I),I=N9,NN)
NN=N6 + NEQ*IT#0 - 1
READ (11) (A(I),I=N6,NN)
REWIND 11
WRITE (11) (A(I),I=N6,NN)

T I M E S T E P I N C R E M E N T A T I O N

KSTEP .EQ. STEP COUNTER

ORIGINAL PAGES
OF POOR QUALITY

MAIN

DATE = 81229 09/18/94

TIME .EQ. TIME AT WHICH SOLUTION IS REQUIRED

100 KSTEP=KSTEP + 1
TIMEP=TIME + DTA
TIME=TIME + DT

STIFFNESS REFORMATION FLAG
IREF.EQ.0 IF STIFFNESS IS TO BE REFORM: 0.

CALL BLKCNT(KSTEP,NSREFB,IREF,ISREFB,NSTE,1)
IF (IDRE.NE.3 .AND. KSTEP.EQ.1) IREF=0

FLAG FOR EQUILIBRIUM ITERATION
IEQUIT.EQ.0 IF ITERATION IS TO BE PERFORMED
IEQUIT.GT.0 IF NO ITERATION IS TO BE PERFORMED

CALL BLKCNT(KSTEP,NEQITB,IEQUIT,IEQITB,NSTE,2)

FLAG FOR TRIANGULARIZATION AND/OR SIMPLE REDUCTION AND
BACKSUBSTITUTION IN COLSOL
KTR.EQ.1 FOR TRIANGULARIZATION PLUS SOLUTION
KTR.EQ.2 FOR VECTOR SOLUTION ONLY

KTR=1
IF (IREF.NE.0) KTR=2
IF (KLIN.EQ.0) KTR=2

NEQREF IS THE NUMBER OF TIMES THE NONLINEAR STIFFNESS MATRIX
WAS REFORMED

140 NEQREF=0
REWIND 4
REWIND 7

FLAG TO INDICATE CONVERGENCE IN EQUILIBRIUM ITERATION
IEQREF.EQ.0 CONVERGENCE
IEQREF.EQ.1 NORM. OF OUT-OF-BALANCE LOADS IS LARGER THAN NORM
OF INCREMENTAL LOADS (SEE EQUIT)

IEQREF=0

SOLUTION OF INCREMENTAL EQUATIONS

CALCULATE LINEAR EFFECTIVE LOADS BALANCED IN CURRENT CONFIGURATION

CALL SECOND (TIM9)

CALL LOADF (A(N1),A(N1A),A(N2),A(N1),A(N7),A(N8),A(N3),A(N6),
A(N4),A(N9),NBLCK,ISTOM)

CALL SECOND (TIM10)

CALCULATE EFFECTIVE NONLINEAR MATRIX AND FINAL EFFECTIVE LOADS

CALL ASSEM (A(N1),A(N4),A(N4A),A(N2),A(N3),A(N10),A(N1A),A(N6B),

MAIN

DATE = 81229

09/18/44

1- A(NIC),ISTDH,NBLOCK)

CALL SECOND (TIM11)
IF (KSTEP.EQ.1 .AND. IOPE.NE.3) WRITE (6,2300) TIM11...

SOLVE FOR INCREMENT IN DISPLACEMENT VECTOR, NO TRIANGULARIZATION
IF SIMPLE EQUILIBRIUM ITERATION IS TO BE PERFORMED...

IF (IOPE.NE.3) CALL COLSOL (A(N1),A(N1A),A(N1B),A(N4),A(N4A),
A(N4B),A(N3),NEQ,NBLOCK,ISTDH,12,10,KTR)

CALL SECOND (TIM12)
IF (KSTEP.EQ.1 .AND. IOPE.NE.3) WRITE (6,2310) TIM12
TSUM1=TSUM1 + (TIM10 - TIM9)
TSUM2=TSUM2 + (TIM11 - TIM10)
TSUM3=TSUM3 + (TIM12 - TIM11)

ITERATION FOR DYNAMIC EQUILIBRIUM

NO ITERATION IN LINEAR ANALYSIS

IF (KLIN.EQ.0) GO TO 110

IF (IEQUIT.NE.0) GO TO 110

CALL SECOND (TIM13)

CALL EQUIT (A(N4),A(N3),A(N5),A(N2),A(N7),A(N8),A(N1),A(N5),
A(N9),A(N10),A(N4A),A(N4B),A(N1A),A(N1B),ISTDH)

IF NO CONVERGENCE IN ITERATION PROCEED TO NEXT DATA CASE

IF (ITE.GT.ITEMAX) GO TO 190

CALL SECOND (TIM14)
TSUM4=TSUM4 + (TIM14 - TIM13)

CHECK FOR NO CONVERGENCE IN EQUILIBRIUM ITERATION AND
POSSIBLE REFORMATION OF STIFFNESS

IF (IEOREF.EQ.3) GO TO 110

KTR=1
IREF=0

NEOREF=NEOREF + 1

IF (NUMREF.EQ.0) GO TO 112

IF (NEOREF.LE.1) GO TO 140

112 WRITE(6,2043)

GO TO 190

CALCULATE NEW DISP, VEL, ACC VECTORS AT TIME=TSTART + KSTEP*DT
FOR STATIC ANALYSIS AND IMPLICIT TIME INTEGRATION AND ALSO DISP
VECTOR AT TIME=TSTART + (KSTEP + 1)*DT FOR CENTRAL DIFFERENCE
METHOD

110 CALL SECOND (TIM15)

ORIGINAL PAGE IS
OF POOR QUALITY

```

                                MAIN.                DATE = 81229.                09/18/44.
0      43H  CALCULATION OF EFFECTIVE LOAD VECTORS. . . . .F9.2/
7      43H  UPDATING EFFECTIVE STIFFNESS MATRICES . . . . .
9      43H  AND LOAD VECTORS FOR NONLINEARITIES . . . . .F9.2/
A      43H  SOLUTION OF EQUATIONS . . . . .F9.2/
B      43H  EQUILIBRIUM ITERATIONS. . . . .F9.2/
C      43H  CALCULATION AND PRINTING OF DISPLACE- . . . . .
D      43H  MENTS, VELOCITIES, AND ACCELERATIONS. . . . .F9.2/
E      43H  CALCULATION AND PRINTING OF STRESSES . . . . .F9.2/
F      30X,19H STEP-BY-STEP TOTAL .F9.2///
1      49H T O T A L S O L U T I O N T I M E ( S E C ) . . .F9.2)
2200 FORMAT (//30H STORAGE CHECK FOR LOAD INPUT )
2202 FORMAT(//53H STORAGE CHECK FOR MATRIX ASSEMBLAGE, TIME INTEGRATION)
2203 FORMAT (//60H ** STOP ** NO STORAGE AVAILABLE TO STORE STIFFNESS
1 MATRIX. //68H INCREASE MTOT AND/OR BREAK ELEMENTS INPUT INTO MORE
2 ELEMENT GROUPS. )
2204 FORMAT (// 22H STOP ERROR IN INPUT //
1 39H MORE THAN 1000 SOLUTION BLOCKS. REQD )
2210 FORMAT (1H1,
155HTOTAL SYSTEM DATA . . . . .//15X,
255HNUMBER OF EQUATIONS . . . . .(NEQ) =,19//5X,
355HNUMBER OF MATRIX ELEMENTS . . . . .(NWK) =,18//5X,
455HMAXIMUM HALF BANDWIDTH . . . . .(MA) =,18//5X,
555HMEAN HALF BANDWIDTH . . . . .(MAM) =,18//5X,
655HMAXIMUM BLOCK LENGTH . . . . .(ISTOH) =,18//5X,
755HNUMBER OF BLOCKS . . . . .(NBLOCK) =,18//5X,
855HMAXIMUM TOTAL STORAGE AVAILABLE. . . . .( MTOT ) =,18//1
2220 FORMAT(4X,51H NUMBER OF COLUMNS PER BLOCK AND 1ST COUPLING BLOCK)
2230 FORMAT (6X,16H NUMBER OF BLOCK,12X,(2114,/,34X))
2240 FORMAT (6X,28H NUMBER OF COLUMNS PER BLOCK,(2114,/,34X))
2250 FORMAT (6X,21H FIRST COUPLING BLOCK,7X,(2114,/,34X))
2260 FORMAT (//50H0**STORAGE CHECK FOR ASSEMBLAGE OF LINEAR MATRICES )
2270 FORMAT (//50H0**STORAGE CHECK FOR LOAD VECTORS INPUT PHASE )
2280 FORMAT (//50H0**STORAGE CHECK FOR TIME INTEGRATION PHASE )
2290 FORMAT (//15H STEP NUMBER =,15,5X,12H ( AT TIME ,E10.4,2H ) )
2300. FORMAT (//,59H TIME AT ENTERING EQUATION SOLVER FOR FIRST FACTORIS
1ATION =,F10.5)
2310. FORMAT (//,67H TIME AT THE END OF SOLUTION OF EQUATIONS FOR THE FI-
1RST TIME STEP =,F10.5)-
C      DEBUG TRACE, SUBTRACE
C      AT 100
C      TRACE ON
C      AT 190
C      TRACE OFF
C      END

```

ORIGINAL PAGE IS
OF POOR QUALITY

FORTRAN IV G1 RELEASE 2.0

MAIN

DATE = 81223

OPTIONS IN EFFECT NOTERM, ID, EBCDIC, SOURCE, NOLIST, NODECK, LOAD, NUMAP
OPTIONS IN EFFECT NAME = MAIN, LINECNT = 60
STATISTICS SOURCE STATEMENTS = 343, PROGRAM SIZE = 14588
STATISTICS NO DIAGNOSTICS GENERATED

ORIGINAL PAGE IS
OF POOR QUALITY

MAIN

DATE = 81229

09/18/44

```

C      DATA SET ADINI      AT LEVEL TMP AS OF 08/17/81
C *CDC* *DECK OVL1)
C *CDC* OVERLAY (ADINA,1,0)
C *CDC* *DECK ADINI
C *UNI* )FOR,IS N,ADINI, R,ADINI
C *CDC* PROGRAM ADINI
C      SUBROUTINE ADINI
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C      COMMON /SQL/ NUMNP,NEQ,NWK,NWM,NWC,NUMEST,MIDEST,MAXEST,NSTE,MA
C      COMMON /DIM/ NO,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
C      COMMON /EL/ IND,ICDUNT,NPAR(20),NUMEG,NEGL,NEGNL,IMASS,IDAMP,ISTAT
C      COMMON /DOF/ KLIN,IEIG,IMASSN,IDAMPN
C      COMMON /CONST/ DT,DTA,A0,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11
C      COMMON /LOA/ TEND,NTFN,NPTM,NLQAD,LDGRAV,NPR2,NPR3,NODE3
C      COMMON /JUNK/ HED(12),MTOT,LPRG
C      COMMON /VAR/ NG,MODEX,IUPDT,KSTEP,ITEMAX,IEQREF,ITE,KPRI
C      COMMON /NORMS/ RNORM,RENORM,RTOL,ONORMM
C      COMMON /PRCON/ IDATWR,IPRIB,NPB,IOC,IVC,IAC,IPC,IPNODE(3,15)
C      COMMON /ADINA/ OPVAR(7),TSTART,IRINT,ISTOTE
C      COMMON /TEMPRT/ TEMP1,TEMP2,ITEMPR,IIP96,N6A,N6B
C      COMMON /MDFROM/ IDOF(6)
C      COMMON /DPR/ ITWO
C      COMMON /BLOCKS/ NSREFB,NEQITB,NPRIB,NODSVB,LEMSVB,ISREFB(3,10),
C      IEQITB(3,10),IPRIB(3,10),INODB(3,10),IELMB(3,10)
C      COMMON /PORT/ INPORT,JNPORT,NPUTSV,LUNODE,LU1,LU2,LU3,JDC,JVC,JAC
C      COMMON /RANDI/ NOA,NID,IELCPL
C      COMMON /BEAR/ IBEAR,MTUTB
C      COMMON A(1)
C      REAL A
C      DIMENSION BLKNAM(5)
C      DATA BLKNAM /8HSTIFNESS, 8HITERATON, 8HPRINTOUT,
C      8HNODESAVE, 8HELMTSAVE /
C      DATA RECLB1 /8HMASTERCP/, FINAL /6HSTOP.. /
C
C      IND.EQ.2 WHEN LOAD VECTORS ARE TO BE ASSEMBLED
C      IF (IND.EQ.2) GO TO 30
C
C      READ CONTROL INFORMATION
C
C      READ (5,1000) HED
C      IF (HED(1).EQ.FINAL) STOP
C      READ (5,1003) NUMNP,(IDOF(I),I=1,6),NEGL,NEGNL,MODEX,NSTE,
C      1 DT, TSTART, IDATWR, IRINI, IIP96, INPORT, JNPORT, ISTOTE
C      IF (NUMNP.EQ.0) STOP
C      IF (IDATWR.EQ.0 .OR. IDATWR.EQ.2) CALL INLIST (2)
C      READ (5,1001) IMASS, IDAMP, IMASSN, IDAMPN
C.....
C      READ CALCULATION MODES
C      IBEAR = 1 BEARING CALCULATIONS REQUIRED

```

ORIGINAL PAGE IS
OF POOR QUALITY

ADINI

DATE = 81229

09/18/44

```

C..
C.      IBEAR = 0   NO BEARING CALCULATIONS REQUIRED
C.
C.....
4000  READ (5,1002) IFIG,IBEAR
      WRITE (6,4000) IBEAR
      FORMAT (////SX, IBEAR(ADINI)=*,IS)
      READ (5,1020) IUPF,OPVAR
      READ(5,1005) NSREFB,NEQITB,ITEMAX,RTOL
      READ(5,1010) NPRIB,NPB,IDC,IVC,IAC
      READ (5,1010) NPUTSV,NUDSVB,LEMSVB,LUNODE,LU1,LU2,LJ3,JDC,JVC,JAC
C
      IF (NSREFB.EQ.0) GO TO 250
      READ (5,1100)((ISREFB(I,J),I=1,3),J=1,NSREFB)
      IF (NSTE.GT.0 .AND. ISREFB(1,1) .EQ. 0) ISREFB(1,1) = 1
      IF (ISREFB(2,1) .EQ. 0) ISREFB(2,1) = NSTE
      IF (ISREFB(3,1) .EQ. 0) ISREFB(3,1) = 1
      INDEX=1
      IF (NSREFB.LE.1) GO TO 240
      DO 230 I=2,NSREFB
      J=1
      IF (ISREFB(1,J).GT.ISREFB(2,J)) GO TO 235
      IF (ISREFB(1,1).GE.ISREFB(2,J)) GO TO 230
      WRITE (6,3000) BLKNAM(INDEX),I,J
      STOP
230  CONTINUE
240  J=NSREFB
      IF (ISREFB(1,J).LE.ISREFB(2,J)) GO TO 245
235  WRITE (6,3004) BLKNAM(INDEX),J,J
      STOP
245  IF (ISREFB(2,NSREFB).GE.NSTE) GO TO 250
      WRITE (6,3001) BLKNAM(INDEX),ISREFB(2,NSREFB),NSTE
      STOP
C
250  IF (NEQITB.EQ.0) GO TO 350
      READ (5,1100)((IEQITB(I,J),I=1,3),J=1,NEQITB)
      IF (NSTE.GT.0 .AND. IEQITB(1,1) .EQ. 0) IEQITB(1,1) = 1
      IF (IEQITB(2,1) .EQ. 0) IEQITB(2,1) = NSTE
      IF (IEQITB(3,1) .EQ. 0) IEQITB(3,1) = 1
      INDEX=2
      IF (NEQITB.LE.1) GO TO 340
      DO 330 I=2,NEQITB
      J=1
      IF (IEQITB(1,J).GT.IEQITB(2,J)) GO TO 335
      IF (IEQITB(1,1).GE.IEQITB(2,J)) GO TO 330
      WRITE (6,3000) BLKNAM(INDEX),I,J
      STOP
330  CONTINUE
340  J=NEQITB
      IF (IEQITB(1,J).LE.IEQITB(2,J)) GO TO 345
335  WRITE (6,3004) BLKNAM(INDEX),J,J
      STOP
345  IF (IEQITB(2,NEQITB).GE.NSTE) GO TO 350
      WRITE (6,3001) BLKNAM(INDEX),IEQITB(2,NEQITB),NSTE
      STOP
C
350  IF (NPRIB .EQ.0) GO TO 450
      READ (5,1100)((IPRIB(I,J),I=1,3),J=1,NPRIB)

```

ORIGINAL PAGE IS
OF POOR QUALITY

ADINI.

DATE = 81229

09/18/44

```
IF (NSTE.GT.0 .AND. IPRIB(1,1).EQ.0) IPRIB(1,1)=1
IF (IPRIB(2,1).EQ.0) IPRIB(2,1) = NSTE
IF (IPRIB(3,1).EQ.0) IPRIB(3,1) = 1
```

INDEX=3.

IF (NPRIB.LE.1) GO TO 440

DO 430 I=2,NPRIB

J=I - 1

IF (IPRIB(1,J).GT.IPRIB(2,J)) GO TO 435

IF (IPRIB(1,1).GE.IPRIB(2,J)) GO TO 430

WRITE (6,3000) BLKNAM(INDEX),I,J

STOP

430 CONTINUE

440 J=NPRIB

IF (IPRIB(1,J).LE.IPRIB(2,J)) GO TO 445

435 WRITE (6,3004) BLKNAM(INDEX),J,J

STOP

445 IF (IPRIB(2,NPRIB).GE.NSTE) GO TO 450

WRITE (6,3001) BLKNAM(INDEX),IPRIB(2,NPRIB),NSTE

STOP

C

450 IF (NPB .EQ.0) GO TO 550

READ (5,1100) ((IPNODE(I,J),I=1,3),J=1,NPB)

DO 500 I=1,NPB

500 IF (IPNODE(3,I).EQ.0) IPNODE(3,I)=1

550 IF (JNPORT.EQ.0 .OR. NUDSVB.EQ.0) GO TO 650

READ (5,1100)((INODB(I,J),I=1,3),J=1,NUDSVB)

IF (NSTE.GT.0 .AND. INODB(1,1).EQ.0) INODB(1,1) = 1

IF (INODB(2,1).EQ.0) INODB(2,1) = NSTE

IF (INODB(3,1).EQ.0) INODB(3,1) = 1

INDEX=4

IF (NUDSVB.LE.1) GO TO 640

DO 630 I=2,NUDSVB

J=I - 1

IF (INODB(1,J).GT.INODB(2,J)) GO TO 635

IF (INODB(1,1).GE.INODB(2,J)) GO TO 630

WRITE (6,3003) BLKNAM(INDEX),I,J

STOP

630 CONTINUE

640 J=NUDSVB

IF (INODB(1,J).LE.INODB(2,J)) GO TO 645

635 WRITE (6,3004) BLKNAM(INDEX),J,J

STOP

645 IF (INODU(2,NUDSVB).GE.NSTE) GO TO 650

WRITE (6,3001) BLKNAM(INDEX),INODU(2,NUDSVB),NSTE

STOP

C

650 IF (JNPORT.EQ.0 .OR. LEMSVB.EQ.0) GO TO 750

READ (5,1100)((IELMB(I,J),I=1,3),J=1,LEMSVB)

IF (NSTE.GT.0 .AND. IELMB(1,1).EQ.0) IELMB(1,1) = 1

IF (IELMB(2,1).EQ.0) IELMB(2,1) = NSTE

IF (IELMB(3,1).EQ.0) IELMB(3,1) = 1

INDEX=5

IF (LEMSVB.LE.1) GO TO 740

DO 730 I=2,LEMSVB

J=I - 1

IF (IELMB(1,J).GT.IELMB(2,J)) GO TO 735

IF (IELMB(1,1).GE.IELMB(2,J)) GO TO 730

WRITE (6,3000) BLKNAM(INDEX),I,J

ADINI

DATE = 81229

09/18/44

```

STOP
730 CONTINUE
740 J=LEMSVB
   IF ( IELMB(1,J).LE.IELMB(2,J)) GO TO 745
735 WRITE (6,3004) BLKNAM(INDEX),J,J
   STOP
745 IF ( IELMB(2,LEMSVB).GE.NSTE) GO TO 750
   WRITE (6,3001) BLKNAM(INDEX),IELMB(2,LEMSVB),NSTE
   STOP
750 CONTINUE

```

C
C
C

VERIFY AND INITIALIZE SOLUTION VARIABLES

```

NUMEG=NEGL + NEGL
NDOF=6
DO 1, I=1,6
1  NDOF=NDOF - IDOF(I)
   ISTAT=1
   IF ( IMASS.EQ.0) ISTAT=0
   IF ( IOPE.NE.3 .OR. IMASS.EQ.1) GO TO 20
   WRITE (6,3012)
   STOP
20  IF ( IDAMP.EQ.1 .AND. IMASS.EQ.0) GO TO 5
   IF ( IDAMP.NE.0 .AND. IMASS.EQ.0) GO TO 5
   IF ( IMASS.NE.0 .AND. IMASS.EQ.0) GO TO 5
   GO TO 3
5  WRITE(6,3002)
   STOP
3  TH=1.0
   DTA=DT
   KLIN=1
   IF ( NEGL.EQ.0) KLIN=0
   IF ( IOPE.NE.3 .OR. IEIG.EQ.0) GO TO 2
   WRITE (6,3010)
   STOP
2  IF ( IEIG.LE.1) GO TO 4
   WRITE(6,3003)
   STOP
4  IF ( ITEMAX.EQ.0) ITEMAX=15
   IF ( RTOL.EQ.0) RTOL=1.0E-03
   IF ( IOPE.EQ.3 .OR. KLIN.EQ.0) NSREFB=0
   IF ( IOPE.EQ.3 .OR. KLIN.EQ.0) NEQIB=0
   IPC=IDC+IVC+IAC
   IF ( NPB.NE.0) GO TO 6
   IPC=1
   NPB=1
   IDC=1
   IVC=1
   IAC=1
   IPNODE(1,1)=1
   IPNODE(2,1)=NUMNP
   IPNODE(3,1)=1
6  CONTINUE
   IF ( IOPE.EQ.0) IOPE=2
   IF ( IPRINT.EQ.0) IPRINT=9999

```

C

A0=0.
A1=0.

ORIGINAL PAGE IS
OF POOR QUALITY

A3-26

ADINI

DATE = 81229

09/18/44

```
IF(MODEX.NE.2 .AND. JNPORT.GT.0) NPUTSV=1
IF (LUNODE .EQ. 0) LUNODE=60
IF (LU1 .EQ. 0) LU1 =61
IF (LU2 .EQ. 0) LU2 =62
IF (LU3 .EQ. 0) LU3 =63
C
C*** DATA PUTHOLE (START)
C
IF (JNPORT.EQ.0) GO TO 790
RECLAB=RECLB1
WRITE (LUNODE) RECLAB,(HED(1),I=1,12),NUMNP,(IDOF(1),I=1,6),
1 NEGL,NEGNL,MODEX,NSTE,DT,TSTART,IDATWR,LRINT,
2 ITP96,INPURT,JNPORT,IMASS,IDAMP,IMASSN,IDAMPN,
3 IEIG,NSREFB,NEQITB,RTOL,ITEMAX,IOPE,OPVAR(1),
4 OPVAR(2),NPRIB,NUOSVB,LEMSVB,LUNODE,LU1,LU2,LU3,
5 NPB,IOC,IVC,IAC,NPUTSV,JDC,JVC,JAC,
6 ((IPNODE(L,J),I=1,J),J=1,NPJ)
RECLAB=BLKNAM(1)
IF (NSREFB.NE.0) WRITE(LUNODE) RECLAB,((ISREFB(I,J),I=1,J),
1 J=1,NSREFB)
RECLAB=BLKNAM(2)
IF (NEQITB.NE.0) WRITE(LUNODE) RECLAB,((IEQITB(I,J),I=1,J),
1 J=1,NEQITB)
RECLAB=BLKNAM(3)
IF (NPRIB.NE.0) WRITE(LUNODE) RECLAB,((IPRIB(L,J),I=1,J),
1 J=1,NPRIB)
RECLAB=BLKNAM(4)
IF (NUOSVB.NE.0) WRITE(LUNODE) RECLAB,((INOSB(I,J),I=1,3),
1 J=1,NUOSVB)
RECLAB=BLKNAM(5)
IF (LEMSVB.NE.0) WRITE(LUNODE) RECLAB,((IELMB(I,J),I=1,3),
1 J=1,LEMSVB)
C
C*** DATA PUTHOLE (END)
C
790 WRITE (6,2040) HED
IF (ISTAT.EQ.0) GO TO 10
C
C SET TIME INTEGRATION COEFFICIENTS IN CASE OF DYNAMIC PROBLEM
C
CALL OPCOEF(OPVAR)
C
10 IF (IOPE.EQ.1) TH=OPVAR(1)
C
IF (IDATWR.GT.1) GO TO 890
WRITE(6,2045)
NCARD=1
WRITE(6,2055) NCARD
WRITE(6,2000) NUMNP,(IDOF(1),I=1,6),NEGL,NEGNL,MODEX
WRITE (6,2002) NSTE,DT,TSTART,IDATWR
WRITE (6,2023) LRINT,ITP96,INPURT,JNPORT
NCARD=2
WRITE(6,2055) NCARD
WRITE(6,2008) IMASS,IDAMP,IMASSN,IDAMPN
NCARD=3
WRITE(6,2055) NCARD
WRITE(6,2001) IEIG
NCARD = 4
```

ADINI

DATE = 81229

09/18/64

```
WRITE (6,2055) NCARD
WRITE (6,2005) IOPE
IF (IOPE.EQ.1) WRITE(6,2006) OPVAR(1)
IF (IOPE.EQ.2) WRITE(6,2007) OPVAR(1),OPVAR(2)
NCARD=5
WRITE(6,2055) NCARD
WRITE (6,2003) NSREFB,NEQITB,ITEMAX,RTOL
```

C
C
C

PRINT DISPL/VEL/ACC OUTPUT INFORMATION

```
NCARD=6
WRITE(6,2055) NCARD
WRITE (6,2012) NPRIB,NPB,IDC,IVC,IAC
NCARD=7
WRITE(6,2055) NCARD
WRITE (6,2100) NPUTSV,NODSVB,LEMSVB
WRITE (6,2105) LUNODE,LU1,LU2,LU3,JDC,JVC,JAC
IF (NSREFB.NE.0 .OR. NEQITB.NE.0) GO TO 810
IF (NPRIB.NE.0 .OR. JNPORT.NE.0) GO TO 810
IF (NPB.NE.1 .OR. (IPNODE(1,1).NE.1 .OR. IPNODE(2,1).NE.NUMNP))
```

```
1 GO TO 810
GO TO 890
810 WRITE (6,2110)
820 IF (NSREFB.EQ.0) GO TO 830
WRITE (6,2120)
WRITE (6,2130) (J,(J,ISREFB(I,J),I=1,3),J=1,NSREFB)
830 IF (NEQITB.EQ.0) GO TO 840
WRITE (6,2140)
WRITE (6,2150) (J,(J,IEQITB(I,J),I=1,3),J=1,NEQITB)
840 IF (NPRIB.EQ.0) GO TO 850
WRITE (6,2160)
WRITE (6,2170) (J,(J,IPRIB(I,J),I=1,3),J=1,NPRIB)
850 IF (IPNODE(1,1).EQ.1 .AND. IPNODE(2,1).EQ.NUMNP) GO TO 860
WRITE (6,2180)
WRITE (6,2190) (J,(J,IPNODE(I,J),I=1,3),J=1,NPB)
860 IF (NODSVB.EQ.0) GO TO 870
WRITE (6,2200)
WRITE (6,2210) (J,(J,INODB(I,J),I=1,3),J=1,NODSVB)
870 IF (LEMSVB.EQ.0) GO TO 890
WRITE (6,2220)
WRITE (6,2230) (J,(J,IELMB(I,J),I=1,3),J=1,LEMSVB)
880 WRITE (6,2004) ISTAT,KLIN
```

C

```
890 CONTINUE
RTOL=RTOL * RTOL
DNORMM=0.00001
```

C
C
C
C

READ NODAL POINT DATA

```
N0A=N0 + NEGNL
N1=N0A + NEGNL + 1
N2=N1 + NODF*NUMNP
N3=N2 + NUMNP*ITWO
N4=N3 + NUMNP*ITWO
N5=N4 + NUMNP*ITWO
CALL SIZE(N5)
```


ADINL .

DATE = 81229

09/18/44

C
C
CALL INPUT (A(N1),A(N2),A(N3),A(N4),NUMNP,NDUF,NEQ,MODEX)

N3=N2 + NEQ*ITWO
N4=N3 + NEQ*ITWO
N5=N4 + NEQ*ITWO
IF (ISTAT.EQ.0) N5=N3
N6=N5 + NEQ*ITWO
CALL SIZE (N6)

C
C
C
C
C
C
READ LOAD VECTOR CONTROL
INFORMATION

HEAD (5,1010) NLOAD,NTFN,NPTM,IDGRAV,NPR2,NPR3,IPRTYP
IF (IDATWR.LE.1)
WRITE (6,2010) NLOAD,NTFN,NPTM,IDGRAV,NPR2,NPR3,IPRTYP
IF (IDGRAV.EQ.0) OR (ISTAT.EQ.0) GO TO 25
IF (IMASS.EQ.1) GO TO 25
WRITE (6,3020)
STOP

C
25 NUDE3=4
IF (IPRTYP.EQ.1) NUDE3=8
IF (IPRTYP.EQ.2) NUDE3=12

C
C
C
C
C
C
ESTABLISH CONCENTRATED NODAL
MASS AND DAMPING VECTORS

CALL NUDMAS (A(N1),A(N5),NDUF)

C
C
C
C
C
C
READ INITIAL CONDITIONS

CALL INITAL (A(N2),A(N5),A(N3),A(N4),A(N1),NDUF)

C
C
C
C
C
C
REINSTATE NODAL COORDINATES INTO HIGH SPEED STORAGE FROM TAPE9

REWIND 9
N3=N2 + NUMNP*ITWO
N4=N3 + NUMNP*ITWO
N5=N4 + NUMNP*ITWO
NN=N3 = 1
HEAD (9) (A(1),I=N2,NN)
NN=N4 = 1
HEAD (9) (A(1),I=N3,NN)
NN=N5 = 1
HEAD (9) (A(1),I=N4,NN)

C
C
C
C
C
C
GO TO 599

C
C
C
C
C
C
CALCULATE AND STORE LOAD VECTORS

ORIGINAL PAGE IS
OF POOR QUALITY

ADINI

JATE = 81229

09/18/44

C *CJC* 30 CALL OVERLAY (SHADINA,1,1,5HRECALL)
30 CALL LOAD

C
599 CONTINUE
RETURN
1000 FORMAT (12A6)-
1001 FORMAT (4I5)
1002 FORMAT (2I5)
1003 FORMAT (15,6I1,14,3I5,2F10.0,6I5)
1005 FORMAT (3I5,F10.0)-
1010 FORMAT (16I5)
1020 FORMAT (1I0,7F10.0)
1100 FORMAT (15I5,5X)

C
2000 FORMAT(/,5X,
255HNUMBER OF NODAL POINTS (NUMNP) =.15/5X,
355HMASTER X-TRANSLATION CODE (IDOF(1)) =.15/5X,
355HMASTER Y-TRANSLATION CODE (IDOF(2)) =.15/5X,
355HMASTER Z-TRANSLATION CODE (IDOF(3)) =.15/5X,
355HMASTER X-ROTATION CODE (IDOF(4)) =.15/5X,
355HMASTER Y-ROTATION CODE (IDOF(5)) =.15/5X,
355HMASTER Z-ROTATION CODE (IDOF(6)) =.15/5X,
455HNUMBER OF LINEAR ELEMENT GROUPS (NEGL) =.15/5X,
555HNUMBER OF NONLINEAR ELEMENT GROUPS (NEGL) =.15/5X,
D55HSOLUTION MODE (MODEX) =.15/5X,
E55H EQ.0, DATA CHECK
F55H EQ.1, EXECUTION /5X,
G55H EQ.2, RESTART /5X,
2001 FORMAT (/5X,
H55HFREQUENCIES SOLUTION CODE (IEIG) =.15/5X,
I55H EQ.0, NO FREQUENCIES SOLUTION /5X,
J55H EQ.1, FREQUENCIES AND MODE SHAPES /5X,
K55H ARE DETERMINED /5X,
2002 FORMAT (/4X,
A50H NUMBER OF TIME STEPS (NSTE) =15/4X,
B50H TIME STEP INCREMENT (DT) =E11.4//
C 4X,
D50H TIME AT SOLUTION START (TSTART) =E11.4//
E 4X,
F 55H FLAG FOR WRITING INPUT DATA IN CARD IMAGE AND/OR /5X,
G 55H GENERATED FORM (IDATWR) =15/4X,
H 55H EQ.0, BOTH CARD IMAGE LISTING AND DETAILED /4X,
I 55H OUTPUT OF INPUT DATA /4X,
J 55H EQ.1, ONLY DETAILED OUTPUT OF INPUT DATA /4X,
K 55H EQ.2, ONLY A CARD IMAGE LISTING OF INPUT DATA /4X,
L 55H GT.2, NO DETAILED OUTPUT NOR CARD IMAGE /4X,
M 55H LISTING OF INPUT DATA /4X,
2020 FORMAT (/5X,
T55HRESTART SAVE INTERVAL (IRINT) =.15/5X,
U55HTEMPERATURE TAPE FLAG (ITP96) =.15/5X,
V35H EQ.0, TEMPERATURE TAPE NOT USED, /5X,
W30H EQ.1, TEMPERATURE TAPE USED //5X,
A-55HPREPROCESSOR INPUT CONTROL PARAMETER (INPORT) =15/4X,
B 55H EQ.0, NO PREPROCESSOR TAPE USED /4X,
C 50H EQ.1, NODAL AND ELEMENT INFORMATION READ FROM UNITS9 /4X,
D 50H EQ.2, ABOVE INFORMATION AND ID ARRAY READ FROM UNITS9 /4X,
4 55HPORTHOLE PARAMETER (UNPORT) =15/4X,
S 55H EQ.0, PORTHOLE NOT WRITTEN /4X,

ORIGINAL PAGE IS
OF POOR QUALITY

ADINI . . .

DATE = 81229

09/18/44

T 55H EQ.1, PORTHOLE WRITTEN 111

2003 FORMAT (/4X,
 A 55H NO. OF BLOCKS OF EFFECTIVE STIFFNESS 14X.
 B56H REFORMATION TIME STEPS (NSREFB) =15 14X.
 C 55H EQ.0, NO STIFFNESS REFORMATION 14X.
 D 55H NO. OF BLOCKS OF EQUILIBRIUM 14X.
 E56H ITERATION TIME STEPS (NEQITB) =15 14X.
 F 55H EQ.0, NO EQUILIBRIUM ITERATION PERFORMED 15X.
 X55H MAXIMUM NUMBER OF EQUILIBRIUM 15X.
 Y55H ITERATIONS PERMITTED (ITEMAX) =.15/15X.
 I 55H CONVERGENCE TOLERANCE (RTOLL) =E11.4 1

2004 FORMAT (1H0,////,1X,
 A02HA N A L Y S I S - T Y P E (ESTABLISHED USING IMASS,NEGL,NEGLN).
 G//,5X,
 B55H TIME DEPENDENCY CODE (ISTAT) =.15 /5X.
 C55H EQ.0, STATIC ANALYSIS 15X.
 D55H EQ.1, DYNAMIC ANALYSIS 15X.
 E55H NONLINEARITY CODE (KLIN) =.15 /5X.
 F55H EQ.0, LINEAR ANALYSIS 15X.
 G55H EQ.1, NONLINEAR ANALYSIS 1

2005 FORMAT (/5X,
 155H TIME INTEGRATION CODE (IOPE) =.15 /5X.
 255H EQ.1, WILSONS THETA METHOD ./.5X.
 355H EQ.2, NEWMARKS METHOD ./.5X.
 455H EQ.3, CENTRAL DIFFERENCE METHOD ./.5X)

2006 FORMAT (5X,
 155H INTEGRATION PARAMETER (THETA) =.F5.2)

2007 FORMAT (5X,
 155H INTEGRATION PARAMETERS (DELTA) =.F5.2/5X.
 255H (ALPHA) =.F5.2)

2012 FORMAT (/4X,
 A 55H NO. OF BLOCKS OF TIME STEPS FOR NODAL AND ELEMENT 14X.
 B56H ASSOCIATED QUANTITIES PRINT-OUT (NPRIB) =15 14X.
 C 55H EQ.0, NO PRINT-OUT 15X.
 B55H NUMBER OF BLOCKS OF NODAL PRINTOUT (NPB) =.15/15X.
 C55H DISPLACEMENT PRINTOUT CODE (IOC) =.15 /5X.
 D55H EQ.0, NO PRINTING OF DISPLACEMENTS 15X.
 E55H EQ.1, PRINT DISPLACEMENTS 15X.
 F55H VELOCITY PRINTOUT CODE (IVC) =.15 /5X.
 G55H EQ.0, NO PRINTING OF VELOCITIES 15X.
 H55H EQ.1, PRINT VELOCITIES 15X.
 I55H ACCELERATION PRINTOUT CODE (IAC) =.15 /5X.
 J55H EQ.0, NO PRINTING OF ACCELERATIONS 15X.
 K55H EQ.1, PRINT ACCELERATIONS 1

2008 FORMAT (/5X,
 155H MASS MATRIX CODE (IMASS) =.15 /5X.
 255H EQ.0, NO MASS EFFECTS 15X.
 355H EQ.1, LUMPED MASS 15X.
 455H EQ.2, CONSISTENT MASS 15X.
 555H DAMPING MATRIX CODE (IDAMP) =.15 /5X.
 655H EQ.0, NO DAMPING 15X.
 755H EQ.1, DAMPING INCLUDED 15X.
 B55H NUMBER OF NODAL MASSES (IMASSN) =.15/15X.
 C55H NUMBER OF NODAL DAMPERS (IDAMPN) =.15 1

2010 FORMAT (1H1,34H L O A D C O N T R O L D A T A // 4X,
 1 38H NUMBER OF LOADS =.15 / 4X, —
 2 38H NUMBER OF TIME FUNCTIONS =.15 / 4X, —
 3 38H MAX NUMBER OF POINTS IN LOAD CURVES =.15 / 4X,

ORIGINAL PAGE IS
OF POOR QUALITY

A3-31

ADINI

DATE = 81229

09/18/44

4 3BH GRAVITY LOADING CODE . . . (IDGRAV) = .15 / 4X.
5 3BH EQ.0, NO GRAVITY LOADING / 4X.
6 3BH EQ.1, LUMPED GRAVITY LOADING / 4X.
7 3BH NUMBER OF 2/D PRESSURE LOAD SETS = .15 / 4X.
8 3BH NUMBER OF 3/D PRESSURE LOAD SETS = .15 / 4X.
9 3BH MAX. ORDER OF ANY ELEMENT FOR / 4X.
A 25H 3/D PRESSURE LOAD SETS—.12X,1H=.15///—
2040 FORMAT(1H1,12A6,///)
2045 FORMAT(3BH M A S T E R C O N T R O L C A R D S)
2055 FORMAT(///,1X,12HCARD NUMBER ,11)
2100 FORMAT (/,4X,
A 55H PORTHOLE CREATION CONTROL CARD //4X.
B 55H FLAG FOR SAVING INPUT DATA ON TAPE (NPUTSV) =15 /4X.
C 55H EQ.0, WRITE ONLY MAIN HEADER /4X.
D 55H EQ.1, WRITE ALL INPUT DATA ON PORTHOLE //4X.
E 55H NU. OF BLOCKS OF TIME STEPS FOR SAVING /4X.
F 55H NODAL RESPONSES ON TAPE (NODSVB) =15 /4X.
G 55H EQ.0, NO ACTION //4X.
H 55H NU. OF BLOCKS OF TIME STEPS FOR SAVING. /4X.
I 55H ELEMENT RESPONSES ON TAPE (LEMSVB) =15 /4X.
J 55H EQ.0, NO ACTION //)
2105 FORMAT (/,4X,
K 55H NUDE DATA SAVE TAPE NUMBER (LUNODE) =15 /4X.
L 55H TRUSS/B EAM/TIE TAPE NUMBER (LU1) =15 /4X.
M 55H 2/D CONTINUUM TAPE NUMBER (LU2) =15 /4X.
N 55H 3/D CONTINUUM TAPE NUMBER (LU3) =15 /4X.
A 55H DISPLACEMENT SAVE CODE (JDC) =15 /4X.
B 55H EQ.0, NO SAVING OF DISPLACEMENTS. /4X.
C 55H EQ.1, SAVE DISPLACEMENTS ON PORTHOLE //4X.
D 55H VELOCITY SAVE CODE (JVC) =15 /4X.
E 55H EQ.0, NO SAVING OF VELOCITIES /4X.
F 55H EQ.1, SAVE VELOCITIES ON PORTHOLE //4X.
G 55H ACCELERATION SAVE CODE (JAC) =15 /4X.
H 55H EQ.0, NO SAVING OF ACCELERATIONS /4X.
I 55H EQ.1, SAVE ACCELERATIONS ON PORTHOLE //)
2110 FORMAT (1H1,5X,48H S O L U T I O N D E F A I L C A R D S
1)
2120 FORMAT (/,5X,90H (A) B L O C K D E F I N I T I O N C A R D S F O R E F F E C T I V E S T I F F
1 N E S S M A T R I X R E F O R M A T I O N T I M E S T E P S)
2130 FORMAT (/,4X,
A 7H B L O C K ,12 //7X.
B 46H F I R S T S T E P O F T H I S B L O C K (I S R E F B (1 , 1 2 , 3 H)) = 1 5 / 7 X .
C 46H L A S T S T E P O F T H I S B L O C K (I S R E F B (2 , 1 2 , 3 H)) = 1 5 / 7 X .
D 46H I N C R E M E N T I N T I M E S T E P (I S R E F B (3 , 1 2 , 3 H)) = 1 5 / 7)
2140- F O R M A T (/ , 5 X , 8 0 H (B) B L O C K D E F I N I T I O N C A R D S F O R E Q U I L I B R I U M I T E
1 R A T I O N T I M E S T E P S)
2150 F O R M A T (/ , 4 X ,
A 7H B L O C K ,12 //7X.
B 46H F I R S T S T E P O F T H I S B L O C K (I E Q I T B (1 , 1 2 , 3 H)) = 1 5 / 7 X .
C 46H L A S T S T E P O F T H I S B L O C K (I E Q I T B (2 , 1 2 , 3 H)) = 1 5 / 7 X .
D 46H I N C R E M E N T I N T I M E S T E P (I E Q I T B (3 , 1 2 , 3 H)) = 1 5 / 7)
2160 F O R M A T (/ , 5 X , 8 0 H (C) B L O C K D E F I N I T I O N C A R D S F O R P R I N T - O U T T I M E
1 S T E P S)
2170 F O R M A T (/ , 4 X ,
A 7H B L O C K ,12 //7X.
B 46H F I R S T S T E P O F T H I S B L O C K (I P R I B (1 , 1 2 , 3 H)) = 1 5 / 7 X .
C 46H L A S T S T E P O F T H I S B L O C K (I P R I B (2 , 1 2 , 3 H)) = 1 5 / 7 X .
D 46H I N C R E M E N T I N T I M E S T E P (I P R I B (3 , 1 2 , 3 H)) = 1 5 / 7)

ADINI

JATE = 81229

09/18/44

```

2180 FORMAT (/,5X,80H (D) BLOCK DEFINITION CARDS FOR NODAL PARAMETER
1 PRINT=OUT
2190 FORMAT (/,4X,
A 7H BLOCK *I2 //7X.
B 46H FIRST NODE OF THIS BLOCK . . . . (IPNODE(1,12,3H))= 15 //7X.
C 46H LAST NODE OF THIS BLOCK . . . . (IPNODE(2,12,3H))= 15 //7X.
D 46H INCREMENT IN NODE NUMBER . . . . (IPNODE(3,12,3H))= 15 //
2200 FORMAT (/,5X,80H (E) BLOCK DEFINITION CARDS OF TIME STEPS FOR S
1AVING NODAL RESPONSE
2210 FORMAT (/,4X,
A 7H BLOCK *I2 //7X.
B 46H FIRST STEP OF THIS BLOCK . . . . ( INOD3(1,12,3H))= 15 //7X.
C 46H LAST STEP OF THIS BLOCK . . . . ( INOD3(2,12,3H))= 15 //7X.
D 46H INCREMENT IN TIME STEP . . . . ( INOD3(3,12,3H))= 15 //
2220 FORMAT (/,5X,80H (F) BLOCK DEFINITION CARDS OF TIME STEPS FOR S
1AVING ELEMENT RESPONSES
2230 FORMAT (/,4X,
A 7H BLOCK *I2 //7X.
B 46H FIRST STEP OF THIS BLOCK . . . . ( IELMB(1,12,3H))= 15 //7X.
C 46H LAST STEP OF THIS BLOCK . . . . ( IELMB(2,12,3H))= 15 //7X.
D 46H INCREMENT IN TIME STEP . . . . ( IELMB(3,12,3H))= 15 //
3000 FORMAT (1H1,21H ** STOP ** ERROR IN ,A10,13H BLOCK INPUT./
1 14H FIRST STEP OF ,15,34HTH BLOCK IS LESS THAN LAST STEP OF ,15,
1 9HTH BLOCK. ///)
3001 FORMAT (1H1,20H ** STOP ** ERROR IN ,A10,45HBLOCK INPUT. FINAL STEP
1 OF LAST BLOCK INPUT = ,15,18H, LESS THAN NSTE = ,15)
3004 FORMAT (1H1,21H ** STOP ** ERROR IN ,A10,13H BLOCK INPUT./
1 14H FIRST STEP OF ,15,36HTH BLOCK IS LARGER THAN LAST STEP OF ,15,
1 9HTH BLOCK. ///)
3002 FORMAT (///45H **STOP = IMASS MUST BE GT.0 IF CONCENTRATED
1 / 37H MASSES AND/OR DAMPERS ARE SPECIFIED )
3003 FORMAT(///42H **STOP = IEIG.GT.1 NOT PERMITTED IN THIS
* 19HVERSION OF ADINA** )
3010 FORMAT (///,1H1,100H **STOP** IEIG.GT.0 NOT PERMITTED IF CENTRAL D
1 IFFERENCE METHOD IS TO BE USED FOR TIME INTEGRATION )
3012 FORMAT (///,1H1,60H **STOP** IMASS MUST BE EQ. 1 FOR CENTRAL DIFFE
1 RERENCE METHOD )
3020 FORMAT (///,1H1,75H **STOP** IMASS MUST BE EQ. 1 FOR DYNAMIC ANALY
1 SIS INCLUDING GRAVITY LOADS )

```

C
END

ORIGINAL PAGE IS
OF POOR QUALITY

A3-33

FORTRAN IV G1—RELEASE 2.0

ADINI—

DATE = 81229

OPTIONS IN EFFECT NUTERM, LD, EBCDIC, SOURCE, NOLIST, NODECK, LOAD, NUMAP
OPTIONS IN EFFECT NAME = ADINI, LINECNT = 60
STATISTICS SOURCE STATEMENTS = 355, PROGRAM SIZE = 22912
STATISTICS NO DIAGNOSTICS GENERATED

MAIN

DATE = 81229

09/18/44

```

C *CDC* *DECK INLIST
C DATA SET LOADS AT LEVEL TMP AS OF 08/17/81
C *CDC* *DECK LOADS
SUBROUTINE LOADS (ID, RG, R, TIMV, RV, RMASS, MASS, NOD, NCUR, DIRN, FAC,
1 ARTM, KL, RGST, NTFND, NQUE)

```

SUBROUTINE

1. TO READ THE TIME FUNCTIONS AND CALCULATE THE FUNCTIONAL VALUES AT REQUESTED POINTS.
2. TO CALCULATE THE GRAVITY LOADS.
3. TO READ CONCENTRATED NODAL LOADS.
4. TO CALCULATE THE LOAD VECTORS CORRESPONDING TO THE CONCENTRATED LOADS.

VARIABLES

- ID = ARRAY OF BOUNDARY CONDITION CODES
- RG = INTERPOLATED VALUES OF TIME FUNCTIONS
- R = LOAD VECTOR
- TIMV, RV = ABSCISSA AND ORDINATES OF TIME FUNCTIONS
- NOD = NODAL POINTS TO WHICH LOADS ARE APPLIED
- NCUR = TIME FUNCTION NUMBERS OF LOADS
- DIRN = DIRECTION CODES OF LOADS
- FAC = MULTIPLIER OF LOADS
- ARTM = ARRIVAL TIMES OF LOADS
- KL = INCREMENTS IN NODES FOR GENERATION

```

.....
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /SOL/ NUMNP, NEQ, NWK, NWM, NWC, NUMEST, MIDEST, MAXEST, NSTE, MA
COMMON /CONST/ DT, DTA, A0, A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11
1 A12, A13, A14, A15, A16, A17, A18, A19, A20, IOPE
COMMON /VAR/ NG, MDEX, IUPDT, KSTEP, ITEMAX, IEQREF, ITE, KPRI,
1 IREF, IEQUIT, IPRI, KPLUTN, KPLUTE
COMMON /ADINAI/ OPVAR(7), TSTART, IRINT, ISTOTE
COMMON /MDFROM/ IOVF(6)
COMMON /LOA/ TEND, NTFN, NPTM, NLUA, IDGRAV, NPR2, NPR3, NODE3
COMMON /PRCON/ IDATWR, IPRIC, NPB, IDC, IVC, IAC, IPC, IPNODE(3,15)
COMMON /BEAR/ IBEAR, MTOTB

```

```

C DIMENSION ID(NDOF,1), RG(NTFND,1), R(1), TIMV(1), RV(1), RMASS(1),
1 NOD(1), DIRN(1), NCUR(1), FAC(1), ARTM(1), KL(1)
2 DIMENSION RGST(1), MASS(1)
THO(100), SP(100)

```

```

-----
C THO(I) = INITIAL PHASE ANGLE OF ROTATING IMBALANCE LOAD
C APPLIED AT NODE I
C
C SP(I) = ROTATING SPEED OF IMBALANCE LOAD APPLIED AT
C NODE I
C
-----

```

ORIGINAL PAGE IS
OF POOR QUALITY

CLOADS___

DATE = 81229

09/18/44

REWIND 3
REWIND 8

C READ (8) ((ID(I,J),I=1,NDOF),J=1,NUMNP)

C CALCULATION OF TIME FUNCTION DATA AT ALL TIME POINTS

C IF (NSTE.EQ.0) RETURN

C IF (NTFN.GT.0) GO TO 60

C DO 40 I=1,NEQ
40 R(I)=0.
DO 50 K=1,NSTE
50 WRITE (3) (R(I),I=1,NEQ)
RETURN

C 60 CALL TFUNCT (RG,TIMV,RV,RGST,NTFN,NSTE,MODEX,TSTART,DT,DTA)

C CALCULATE GRAVITY LOADING

C NSTEG=0

C IF (IDGRAV.EQ.0) GO TO 100

C CALL GRAVL (ID,IDOF,MASS,RMASS,R,RG,NUMNP,NOOF,NEQ,NTFN,NSTEG,
1 MODEX)___

C ADD CONCENTRATED LOADING.

C 100 IF (NSTEG.EQ.NSTE) RETURN

C IF (NLOAD.EQ.0) GO TO 120
IF (IDATWR.LE.1.AND.IBEAR.EQ.1) WRITE (5,2001) _____
IF (IBEAR.EQ.1) GO TO 102
IF (IDATWR.LE.1) WRITE (5,2000)

102 CONTINUE

C READ IMBALANCE LOAD INFORMATION : EMPLOYED ONLY FOR
C ROTOR BEARING TYPE PROBLEMS

C IF (IBEAR.EQ.1) READ (5,1001) (NOD(I),IDIRN(I),NCUR(I),FAC(I),
1 ARTM(I),KL(I),THO(I),SP(I),I=1,NLOAD)
IF (IBEAR.EQ.1) GO TO 101
READ (5,1000) (NOD(I),IDIRN(I),NCUR(I),FAC(I),ARTM(I),KL(I),
1 IDEJUR,I=1,NLOAD)

101 KL(NLOAD)=0

C DO 139 I=1,NLOAD
IF (IDATWR.GT.1) GO TO 120
IF (IBEAR.EQ.1) WRITE (6,2011)NOD(I),IDIRN(I),NCUR(I),FAC(I),
1 ARTM(I),KL(I),THO(I),SP(I)
IF (IBEAR.EQ.1) GO TO 139

139 CONTINUE

C 120 IF (MODEX.EQ.0) RETURN

ORIGINAL PAGE IS
OF POOR QUALITY.

CLOADS

DATE = 81229

09/18/44

```
NLL=NSTEG + 1
DO 200 K=NLL,NSTE
DO 210 I=1,NEQ
210 R(I)=0.
C
  IF (NLOAD.EQ.0) GO TO 260
C
  DO 220 L=1,NLOAD
  LI=IDIRN(L)
  IF (IDOF(LI).EQ.1) GO TO 220
  LDOF=LI
  LN=NOD(L)
  ARTMT=ARTM(L)
  FACT=FAC(L)
  LC=NCUR(L)
  IF (KL(L).EQ.0) GO TO 222
  DARTM=(ARTM(L+1) - ARTM(L))/((NOD(L+1) - NOD(L))/KL(L))
  FINCR=(FAC(L+1) - FAC(L))/(NOD(L+1) - NOD(L))/KL(L)
222 DO 230 I=1,LDOF
230 IF (IDOF(I).EQ.1) LI=LI - 1
224 NSTEA=ARTMT/DT
  NSTEF=K - NSTEA
  IF (NSTEF.LE.0) GO TO 226
  AFACT=NSTEA - ARTMT/DT + 1.
C
  II=ID(LI, LN)
  IF (II.LE.0) GO TO 226.
  RGFR=RG(LC, NSTEF)
  IF (ARTMT.EQ.0.) GO TO 240
C
  RGFR=RGST(LC)*(1.0 - AFACT) + RGFR*AFACT
  IF (NSTEF.LE.1) GO TO 240
  RGFR=RG(LC, NSTEF=1)*(1.0-AFACT)+RG(LC, NSTEF)*AFACT
240 IF (IBEAR.NE.1) GO TO 245
C
  ADD HORIZONTAL IMBALANCE LOAD COMPONENT TO EXTERNAL LOAD VECTOR
C
  WRITE (6,1987) L,THU(L),SP(L)
1987 FORMAT (5X, 'L=',110,5X, 'THU(L)=' ,D13.5,5X, 'SP(L)=' ,D13.5)
  R(II)=R(II)+RGFR*FACT*CCOS(THU(L)+SP(L)*DT*K)
C
  ADD VERTICAL IMBALANCE LOAD COMPONENT TO EXTERNAL LOAD VECTOR
C
  R(II+1)=R(II+1)+RGFR*FACT*DSIN(THU(L)+SP(L)*DT*K)
  GO TO 246
245 R(II)=R(II)+RGFR*FACT
246 CONTINUE
226 IF (KL(L).EQ.0) GO TO 220
  LN=LN + KL(L)
  IF (LN.GE.NOD(L+1)) GO TO 220
  FACT=FACT + FINCR
  ARTMT=ARTMT + DARTM
  GO TO 224
220 CONTINUE
C
260 IF (IDGRAV.EQ.0) GO TO 362
C
  DO 360 I=1,NEQ
```

ORIGINAL TIME IS
OF POOR QUALITY.

A3-37

CLOADS

DATE = 81229 _____ 09/19/44 _____

```
360 R(I)=R(I) + RMASS(I)
362 WRITE (3) (R(I),I=1,NEQ)
C WRITE (6,3000) K,(R(I),I=1,NEQ)
3000 FORMAT (5X,'TIME STEP-',18,5X,'R(I)(CLOADS):',2//,5X,8(3X,012.5))
IF (IDBUG.EQ.5) WRITE (6,6000) (R(I),I=1,NEQ)
200 CONTINUE
C
RETURN
1000 FORMAT (3I5,2F10.0,15,5X,15)
1001 FORMAT (3I5,2F10.0,15,2F10.0)
2000 FORMAT (///45H C O N C E N T R A T E D L O A D S D A T A // 4X,
1 53H NODE DIRECTION LOAD CURVE LOAD CURVE MULTIPL
2 50H ARRIVAL TIME NODE GENERATION )
2001 FORMAT (///' B E A R I N G C O N C E N T R A T E D L O A
1 D S D A T A //', 4X,' NODE DIRECTION LOAD CURVE LOAD CURV
2E MULTIPL ARRIVAL TIME NODE GENERATION INITIAL ANGLE LOAD
3ROTATING SPEED')
2010 FORMAT (1H0,2X,15,5X,14,9X,14,9X,E13.5,8X,E12.4,7X,15)
2011 FORMAT (1H0,2X,15,5X,14,9X,14,9X,E13.5,8X,E12.4,7X,15,5X,
1 E13.3,5X,E13.5)
6000 FORMAT (10F12.5/)
END
```

ORIGINAL PART IS
OF POOR QUALITY

FORTRAN IV G1 RELEASE 2.0

CLOADS

DATE = 81229

OPTIONS IN EFFECT NOTERM, IO, EBCDIC, SOURCE, NOLIST, NODECK, LOAD, NDMAP
OPTIONS IN EFFECT NAME = CLOADS, LINECNT = 60
STATISTICS SOURCE STATEMENTS = 101, PROGRAM SIZE = 6248
STATISTICS NO DIAGNOSTICS GENERATED

ORIGINAL PAGE IS
OF POOR QUALITY

MAIN DATE = 81229. 09/18/44
DATA SET COLSOL AT LEVEL TMP AS OF 08/17/81

```

C
C
C *CDC* *DECK COLSOL
C *UNI* )FOR,IS N,COLSOL, R,COLSOL
SUBROUTINE COLSOL (MAXA,NCOLBV,ICOPL,A,B,D,V,NEQ,NBLOCK,ISTORL,
1 NSTIF,NRED,KKK)
C
C .....
C P R O G R A M
C TO SOLVE FINITE ELEMENT STATIC EQUILIBRIUM EQUATIONS OUT-OF-
C CORE, USING COMPACTED STORAGE AND COLUMN REDUCTION SCHEME
C .....
C IMPLICIT REAL*8 (A-H,O-Z)
COMMON /ELSTP/ TIME,IDTHF
COMMON /EL/ IND,ICOUNT,NPAR(20),NUMEG,NEGL,NEGNL,IMASS,IDAMP,ISTAT
1 ,NDOF,KLIN,IEIG,IMASSN,IDAMPN
COMMON /RANDI/ NOA,NID,IELCPL
DIMENSION A(ISTORL),B(ISTORL),D(NEQ),V(1)
INTEGER ICOPL(1),NCOLBV(1),MAXA(1)
C
C KHBB=0
C IF (KKK=2) 10,610,610
10 REWIND NSTIF
PIVOT=10.**20
C
C --- FACTORIZE STIFFNESS MATRIX ( LOOP OVER ALL BLOCKS ) ---
C
C DO 600 NJ=1,NBLOCK
C
C READ (NSTIF) A
C WRITE (6,732) (A(I),I=1,ISTORL)
732 FORMAT (5X,'STIFFNESS MATRIX (COLSOL):',//,5X,3(2X,DL3.6))
NCOLB=NCOLBV(NJ)
MM=MAXA(KHBB+1) = 1
IF (NJ.EQ.ICOPL(NJ)) GO TO 300
C
C IK=ICOPL(NJ) - 1
C IM=0
C IF (IK) 300,140,100
100 DO 120 K=1,IK
120 IM=IM + NCOLBV(K)
140 KHB=KHBB = IM
C IK=IK + 1
C NJ1=NJ - 1
C
C REDUCE BLOCK BY THE PRECEDING COUPLING BLOCKS
C
C DO 160 NK=IK,NJ1
C
C ***** R A N D O M A C C E S S *****
C
C NREC10=NK
C CALL READMS (NRED,B,ISTORL,NREC10)

```

COLSOL

DATE = 81229

09/18/44

C * * * * * R A N D O M A C C E S S * * *

KHB=KHB + NCOLBV(NK)
MC=MAXA(IM+1) - 1

DO 200 N=1,NCOLB
KN=MAXA(KHBB+N) - MM
KL=KN + 1
KU=MAXA(KHBB+N+1) - 1 - MM
KH=KU - KL - N + 1
KC=KH - KHB
IF (KC.LE.0) GO TO 200
IC=0
KCL=NCOLBV(NK) - KC + 1
IF (KCL.GT.0) GO TO 210
IC=1 - KCL
KCL=1

210 KCR=NCOLBV(NK)
KLT=KU - IC

DO 220 K=KCL,KCR
IC=IC + 1
KLT=KLT - 1
KI=MAXA(K+IM) - MC
ND=MAXA(K+IM+1) - KI - MC - 1
IF(ND) 220,220,230
230 KK=MINO(IC,ND)
C=0
DO 240 L=1,KK
240 C=C + B(KI+L)*A(KLT+L)
A(KLT)=A(KLT) - C
220 CONTINUE
200 CONTINUE

IM=IM + NCOLBV(NK)

160 CONTINUE

REDUCE BLOCK BY ITSELF

300 DO 400 N=1,NCOLB
KN=MAXA(KHBB+N) - MM
KL=KN + 1
KU=MAXA(KHBB+N+1) - 1 - MM
KDIF=KU - KL
KH=MINO(KDIF,N-1)
KS=N + KHBB
IF (KH) 420,440,460
460 K=N - KH
KLT=KL + KH
IC=0
IF ((N-1).LT.KDIF) IC=KDIF - N + 1

DO 480 J=1,KH
IC=IC + 1
KLT=KLT - 1
KI=MAXA(KHBB+K) - MM

ORIGINAL PAGE IS
OF POOR QUALITY

CUL SOL

DATE = 81229

09/18/44

```

ND=MAXA(KHBB+K+1) - KI - MM - 1
IF (ND) 480,480,500
500 KK=MIN0(IC,ND)
C=0.
DO 520 L=1,KK
520 C=C + A(KI+L)*A(KLT+L)
A(KLT)=A(KLT) - C
480 K=K + 1
C
440 K=KS
E=0.
DO 540 KK=KL,KU
K=K - 1
C=A(KK)/D(K)
E=E + C*A(KK)
540 A(KK)=C
A(KN)=A(KN) - E
C
420 D(KS)=A(KN)
IF (D(KS)) 401,555,400
555 IF (IDTHF.EQ.0) GO TO 560
D(KS)=PIVOT
GO TO 400
560 WRITE (6,2000) KS,D(KS)
STOP
C
401 WRITE(6,2000) KS,D(KS)
400 CONTINUE
C
C KHBB=KHBB + NCOLB
C
C * * * * * R A N D O M A C C E S S * * *
C
NREC10=NJ
CALL WRITMS (NRED,A,ISTORL,NREC10)
C
C * * * * * R A N D O M A C C E S S * * *
C
600 CONTINUE
IF (KLIN.GT.0) GO TO 606
RETURN
C
- - SOLUTION OF EQUATIONS ( LOOP OVER ALL BLOCKS ) - -
C
C REDUCE THE LOAD VECTOR -
C
606 KHBB=0
610 DO 700 NJ=1,NBLOCK
IF (NBLOCK.EQ.1 .AND. (KLIN.EQ.0 .OR. KKK.EQ.1) ) GO TO 710
C
C * * * * * R A N D O M A C C E S S * * *
C
NREC10=NJ
CALL READMS (NRED,A,ISTORL,NREC10)
C
C * * * * * R A N D O M A C C E S S * * *
C
710 NCOLB=NCOLBV(NJ)

```

ORIGINAL PROGRAM
OF POOR QUALITY

COLSOL

DATE = 81229

09/18/44

```

MM=MAXA(KHBB+1) - 1
DO 720 N=1,NCOLB
KL=MAXA(N+KHBB) - MM + 1
KU=MAXA(N+KHBB+1) - MM - 1
IF (KU=KL) 720,730,730
730 KS=N + KHBB
K=KS
C=0.
DO 740 KK=KL,KU
K=K - 1.
740 C=C + A(KK)*V(K)
V(KS)=V(KS) - C
720 CONTINUE
KHBB=KHBB + NCOLB
700 CONTINUE
C
C
BACKSUBSTITUTE
DO 790 N=1,NEQ
790 V(N)=V(N)/D(N)
NBL=NBLOCK
DO 800 NJ=1,NBLOCK
IF (NBLOCK.EQ.1) GO TO 820
C
C
* * * * * R A N D O M A C C E S S * * *
NJBI=NBLOCK - NJ + 1.
CALL READMS (NREQ,A,ISTORL,NJBI)
C
C
* * * * * R A N D O M A C C E S S * * *
NCOLB=NCOLB*(NBL)
820 KHBB=KHBB - NCOLB
MM=MAXA(KHBB+1) - 1
N=NCOLB
DO 860 L=1,NCOLB
KL=MAXA(N+KHBB) - MM + 1
KU=MAXA(N+KHBB+1) - MM - 1
IF (KU=KL) 861,890,890
890 KS=KHBB + N
K=KS
DO 900 KK=KL,KU
K=K - 1
900 V(K)=V(K) - A(KK)*V(KS)
861 N=N-1
860 CONTINUE
NUL=NBL - 1
800 CONTINUE
C
RETURN
2000 FORMAT (// 40H STOP - STIFFNESS NOT POSITIVE DEFINITE ,//
1 32H NONPOSITIVE PIVOT FOR EQUATION ,14,//
2 10H PIVOT = ,E20.12 )
END

```

ORIGINAL PAGE IS
OF POOR QUALITY.

FORTRAN IV G1 RELEASE 2.0 _____ COLSOL _____ DATE = 81229

OPTIONS IN EFFECT NOTERM, ID, EBCDIC, SOURCE, NOLIST, NODECK, LOAD, NUMAP
OPTIONS IN EFFECT NAME = COLSOL, LINECNT = 60
STATISTICS SOURCE STATEMENTS = 154, PROGRAM SIZE = 4280
STATISTICS NO DIAGNOSTICS GENERATED

ORIGINAL PAGE IS
OF POOR QUALITY

RUSS

DATE = 81245

18/23/01

IF (KPRI.EQ.0) GO TO 800
IF (IND.GT.0) GO TO 420

SKIP TO BEARING CALCULATIONS IF NECESSARY

IF (IBEAR.EC.1.AND.ITYP.EQ.3) GO TO 80

.....
: READ AND GENERATE ELEMENT INFORMATION :
:
:

1. READ MATERIAL PROPERTIES

IF (IDATWR.LE.1) WRITE (6,2000)

GO TO (10,20,30,40,40,60,60,70), MODEL

LINEAR ELASTIC (MODEL 1)

10 IF (IDATWR.LE.1) WRITE (6,2010)
DO 15 I=1,NUMMAT
READ (5,1000) N,AREA(N),DEN(N)
READ (5,1010) E(N)
IF (IDATWR.LE.1) WRITE (6,2011) N,AREA(N),DEN(N),E(N)
15 CONTINUE

*** DATA PORTHOLE ***** (START)

IF (JNPORT.EQ.0 .OR. NPUTSV.EQ.0) GO TO 150
RECLAB = RECLB1
WRITE (LUI) RECLAB,NUMMAT,NCON,(DEN(I),I=1,NUMMAT),
1. (E(I),I=1,NUMMAT),(AREA(I),I=1,NUMMAT)

*** DATA PORTHOLE ***** (END)

GO TO 150

NONLINEAR ELASTIC (MODEL 2)

20 IP=NCON/2
KP=1
DO 25 I=1,NUMMAT
READ (5,1000) N,AREA(N),DEN(N)
READ (5,1010) (PRCP(J,N),J=1,NCON)
IF (IDATWR.GT.1) GO TO 25
WRITE (6,2020) N,AREA(N),DEN(N),KP,PRCP(1,N),PRCP(IP+1,N)
DO 22 K=2,IP
KP=K+IP
ETAN=(PRCP(KP,N) - PRCP(KP-1,N))/(PRCP(K,N) - PRCP(K-1,N))
22 WRITE (6,2021) K,PRCP(K,N),PRCP(KP,N),ETAN

ORIGINAL PAGE IS
OF POOR QUALITY

A3-46

RUSS

DATE = 81245

18/23/01

```
25 CONTINUE
   GO TO 98
C
C
   THERMOELASTIC (MODEL 3)
30 DO 35 I=1,NUMMAT
   READ (5,1000) N,AREA(N),DEN(N)
   READ (5,1010) (PROP(J,N),J=1,NCUN)
   IF (IDATWR.GT.1) GO TO 35
   WRITE (6,2030) N,AREA(N),DEN(N)
   *RITE (6,2031) (PROP(J,N),J=1,NCUN)
35 CONTINUE
   GO TO 98
C
C
   ELASTIC-PLASTIC MODELS (MODEL 4 AND MODEL 5)
40 IF (IDATWR.LE.1) WRITE (6,2040)
   DO 45 I=1,NUMMAT
   READ (5,1000) N,AREA(N),DEN(N)
   READ (5,1010) (PROP(J,N),J=1,NCUN)
   IF (IDATWR.GT.1) GO TO 45
   WRITE (6,2011) N,AREA(N),DEN(N),(PROP(J,N),J=1,NCUN)
45 CONTINUE
   GO TO 98
C
C
   THERMO-ELASTIC-PLASTIC AND CREEP MODELS (MODEL 6 AND MODEL 7)
60 DO 65 I=1,NUMMAT
   READ (5,1000) N,AREA(N),DEN(N)
   READ (5,1010) (PROP(J,N),J=1,NCUN)
   IF (IDATWR.GT.1) GO TO 65
   WRITE (6,2030) N,AREA(N),DEN(N)
   *RITE (6,2061) (PROP(J,N),J=1,NCUN)
65 CONTINUE
   GO TO 98
C
C
   USER-SUPPLIED MODEL (MODEL 8)
70 DO 75 I=1,NUMMAT
   READ (5,1000) N,AREA(N),DEN(N)
   READ (5,1010) (PROP(J,N),J=1,NCUN)
   IF (IDATWR.GT.1) GO TO 75
   WRITE (6,2030) N,AREA(N),DEN(N)
   WRITE (6,2071) (J,N,PROP(J,N),J=1,NCUN)
75 CONTINUE
C
C
*** DATA PORTHOLE ***** (START)
98 IF (UNPORT.EQ.0 .OR. NPUTSV.EQ.0) GO TO 150
   RECLAB = RECLB1
   WRITE (LU1) RECLAB,NUMMAT,NCUN,(DEN(I),I=1,NUMMAT),
1          ((PROP(I,J),I=1,NCUN),J=1,NUMMAT),
2          (AREA(I),I=1,NUMMAT)
C
C
*** DATA PORTHOLE ***** (END)
   GO TO 150
```


ORIGINAL PAGE IS
OF POOR QUALITY

A3-49

RUSS

DATE = 81246

18/23/01

```
IF (IBEAR.EQ.1.AND.ITYPT.EQ.3) GO TO 120
IF (MTYP.LE.0) MTYP=1
IF (MTYP.GT.NUMMAT) GO TO 110
IF (KG.LE.0) KG=1
IF (IELE.LE.MXNUDS) GO TO 120
C
WRITE (6,2300) NG,M,IELE,MXNUDS
STOP
C
110 WRITE (6,2310) NG,M,MTYP,NUMMAT
STOP
C
120 IF (M.NE.N) GO TO 200
DO 122 I=1,IELE
122 NODEM(I)=NODE(I)
MTYPE=MTYP
KKK=KG
EPSI=EPS
IPST=IS
ETIM=ETIME
IELD=IELE
IPLOT=IPLOT
C
SAVE ELEMENT INFORMATION
C
200 IF (ITYPT.NE.1) GO TO 195
I=NODEM(I)
XYZ(1,N)=Y(I)
GO TO 201
C
195 L=-2
DO 190 LL=1,IELD
L=L+3
I=NODEM(LL)
XYZ(L,N)=X(I)
XYZ(L+1,N)=Y(I)
190 XYZ(L+2,N)=Z(I)
C
201 MATP(N)=MTYPE
EPSIN(N)=EPS
IPSN(N)=IPST
IELTC(N)=IELD
IPLOTN(N)=IPLOT
IF (ITYPT.EQ.3.AND.IPLOT.EQ.1) MB=MB+1
IF (ITYPT.EQ.3.AND.IPLOT.EQ.1) NR(MB)=NODEM(I)
IF (ITYPT.EQ.3.AND.IPLOT.EQ.1) NS(MB)=NODEM(I)
NU=3+IELD
IF (ITYPT.EQ.1) ND=1
IF (IBEAR.EQ.1.AND.ITYPT.EQ.3) ND=12
IF (ITYPT.EQ.3) GO TO 204
C
INITIALIZE WORKING STORAGE
C
IF (MODEL.LT.3) GO TO 204
C
CALL IMATB (PROP(1,MTYPE),WA(1,N),WA(1,N),NUOGL(I,N),
1 TEMPV1,NODEM,IELD)
```

RUSS

DATE = 81245

18/23/01

```

C 204 IF (IDEATH.EQ.0) GO TO 210
    IF (IDEATH.EQ.2) GO TO 207
    DO 208 L=1,ND
208 EDISH(L,N)=0
    ETIMV(L)=ETIM
    GO TO 210
207 ETIMV(N)=ETIM

C 210 IF (ITYPT.NE.1) GO TO 211
    LDU=1
    IF (IDOF(1).EQ.0) LDC=2
    LM(1,N)=0
    IF (IDOF(2).EQ.1) GO TO 295
    II=NODEM(1)
    LM(1,N)=ID(LDU,II)
    GO TO 295

C 211 DO 290 L=1,ND
290 LM(L,N)=0
    IF (IBEAR.EQ.1.AND.ITYPT.EQ.3) GO TO 230
    LL=1
    DO 240 L=1,3
    IF (IDOF(L).EQ.1) GO TO 240
    LP=L-3
    DO 241 LK=1,IELD
    LP=LP+LK
    II=NODEM(LK)
    LM(LP,N)=ID(LL,II)
C 240 WRITE (6,9887) N,L,LL,LP,II,IDOF(L)
9887 FORMAT (5X,'N=',I5,5X,'L=',I5,5X,'LL=',I5,5X,'LP=',I5,
15X,'II=',I5,5X,'IDOF(L)=',I5)
241 CONTINUE
    LL=LL+1
240 CONTINUE
C 240 WRITE (6,5098) ((ID(KD,MD),KD=1,6),MD=1,7)
C 240 WRITE (6,9786) (LM(III,N),III=1,ND)
5098 FORMAT (5X,'ID(I,J) (TRUSS ELE)='/,8(5X,I10))
295 CONTINUE

C 230 GO TO 233

C 230 CREATE THE BEARING ELEMENT CONNECTIVITY VECTOR LM

C 230 LL = 1
    DO 232 L=1,6
    IF (IDOF(L).EQ.1) GO TO 232
    LP = L-6
    DO 231 LK=1,IELD
    LP = LP + LK
    II = NODEM(LK)
    LM(LP,N) = ID(LL,II)
    IF (IDOF(L).EQ.1) LM(LP,N)=0
C 231 WRITE (6,9887) N,L,LL,LP,II,IDOF(L)
231 CONTINUE
    LL=LL+1
232 CONTINUE
C 232 WRITE (6,3526) NODE

```

ORIGINAL PAGE IS
OF POOR QUALITY

A3-51

RUSS

DATE = 31245

18/25/01

```

3520 FORMAT ('$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ NOOF=',I5)
C *WRITE (6,5098) ((ID(KD,MD),KD=1,6),MD=1,7)
C *WRITE (6,6067) (NODEM(LK),LK=1,IELD)
3067 FORMAT(5X,'NODEM(LK)=' ,2I20)
C DO 239 LK=1,IELD
C LL = NODEM(LK)
C *WRITE( 6,4309) (ID(LL,IL),LL=1,6)
239 CONTINUE
4309 FORMAT(5X,'ID(LL,IL)=' ,6I12)
C *WRITE (6,9780) (LM(III,N),III=1,ND)
9780 FORMAT (8X,'LM=' ,8(1X,I11))
C UPDATE COLUMN HEIGHTS AND BANDWIDTH
C
233 CALL COLFT (HT,ND,LM(1,N))
C
IF (IDATWR.GT.1) GO TO 298
RL(1)=XYZ(1,N)
IF (ITYPT.EQ.0) CALL LENTH1 (XYZ(1,N),RL)
IF (ITYPT.NE.3) *WRITE (6,2210) N,IELTD(N),IPS(N),MATP(N),KKK,
1 EPSIN(N),RL(1),ETIM,(NODEM(LL),LL=1,IELD)
IF (ITYPT.EQ.3) *WRITE (6,2211) N,IELTD(N),KKK,ETIM,
1 (NODEM(LL),LL=1,IELD)
C
C *** DATA PORTHOLE ***** (START)
C
298 IF (JNPORT.EQ.0 .OR. NPUTSV.EQ.0) GO TO 300
RECLAB=RECLB2
*WRITE (LUI) RECLAB,N,IELTD(N),IPS(N),MATP(N),KKK,EPSIN(N),ETIM,
1 (NODEM(LL),LL=1,IELD)
C *** DATA PORTHOLE ***** (END)
C
C
300 IF (N.EQ.NUME) RETURN
N=N+1
DO 220 LL=1,IELD
220 NODEM(LL)=NODEM(LL) + KKK
IF (N.GT.M) GO TO 100
GO TO 120
C
420 IF (IBEAR.EQ.1.AND.ITYPT.EQ.3) GO TO 720
GO TO (440,560,560,700),IND
C
C A S S E M B L E L I N E A R S T I F F N E S S M A T R I C E S
C
440 DO 500 N=1,NUME
IELD=IELTD(N)
ND=IELD*3
IF (ITYPT.EQ.1) ND=1
CALL ECHECK (LM(1,N),ND,ICD,ICDPT)
IF (ICD.EQ.1) GO TO 500
MTPC=MATP(N)
IF (ITYPT.NE.1) GO TO 501
C
RING ELEMENT

```


RUSS

DATE = 81245

10/23/01

```

C
  ND=1
  S(1)=E(MTYPE)*AREA(MTYPE)/XYZ(1,N)
  GO TO 520
C
C   2=4 NODE TRUSS
C
501 AE=AREA(MTYPE)*E(MTYPE)
    CALL LENTH1 (XYZ(1,N),RL)
    CALL STIF1 (XYZ(1,N),AE,S)
C   WRITE (6,1113) (S(L),L=1,78)
C   1113 FORMAT(5X,'TRUSS STIFFNESS (RUSS):',/,/,8(2X,D13.5))
C   1114 FORMAT (5X,'LM(1,N) (RUSS):',/,/,8(3X,I10))
C
520 CALL ADDBAN(A(N2),A(N1),S,RE,LM(1,N),ND,1)
500 CONTINUE
    RETURN
C
C   A S S E M B L E   M A S S   M A T R I C E S
C
500 IF (IMASS.EQ.2) GO TO 550
C
C   LUMPED MASS DISCRETIZATION
C
543 DO 640 N=1,NUME
    MTYPE=MATP(N)
    IELD=IELD(N)
    ND=IELD*3
C
    IF (ITYPT.NE.1) GO TO 546
    XM(1)=XYZ(1,N)*AREA(MTYPE)*DEN(MTYPE)
    ND=1
    GO TO 547
C
546 CALL LENTH1 (XYZ(1,N),RL)
C
    PM(1)=.5*RL(2)
    GO TO (900,620,625,626), IELD
620 PM(2)=PM(1)
    GO TO 627
625 PM(2)=.5*(RL(1)-RL(2))
    PM(3)=.5*RL(1)
    GO TO 627
626 PM(2)=.5*(RL(1)-RL(3))
    PM(3)=.5*RL(3)
    PM(4)=.5*(RL(1)-RL(2))
C
627 K=0
    ADEN=AREA(MTYPE)*DEN(MTYPE)
    DO 628 I=1,IELD
    XMM=ADEN*PM(I)
    DO 628 L=1,3
    K=K+1
628 XM(K)=XMM
C

```

RUSS

DATE = 81249

18223/01

```

547 CALL ADDMA (A(N4),XM,LM(1,N),ND)
C40 CONTINUE
C
C RETURN
C
C CONSISTENT MASS DISCRETIZATION
C
550 DO 650 N=1,NUME
    IELD=IELTD(N)
    ND=IELD*3
    IF (ITYPT.EQ.1) ND=1
    CALL ECHECK(LM(1,N),ND,ICODE,IUPDT)
    IF (ICODE.EQ.1) GO TO 650
    MTYPE=MATR(N)
C
    IF (ITYPT.NE.1) GO TO 651
    XM(1)=XYZ(1,N)*AREA(MTYPE)*DEN(MTYPE)
    ND=1
    GO TO 655
C
651 ADEN=AREA(MTYPE)*DEN(MTYPE)
    CALL LENTH1 (XYZ(1,N),RL)
    CALL MASEAR (XYZ(1,N),ADEN,S)
C
655 CALL ADDBAN (A(N2),A(N1),S,RE,LM(1,N),ND,1)
C50 CONTINUE
C
C RETURN
C
C A S S E M B L E   N O N L I N E A R   F I N A L   S T R U C T U R E
C T U R E   S T I F F N E S S   A N D   E F F E C T I V E   L O A D
C V E C T O R S
C
700 ISTIF=0
    IF (ICOUNT.EQ.3) GO TO 703
    IF (IREF.EQ.0) ISTIF=1
703 CONTINUE
    MADR=N3
    IF (ICOUNT.EQ.3) MADR=N5
C
    DO 710 N=1,NUME
        IELD=IELTD(N)
        ND=IELD*3
        IF (ITYPT.EQ.1) ND=1
        CALL ECHECK (LM(1,N),ND,ICODE,IUPDT)
        IF (ICODE.EQ.1) IELCPL=IELCPL+1
        IF (ICODE.EQ.1) GO TO 710
        IF (IDEATH.EQ.0) GO TO 692
        ETIM=DABS(ETIMV(N))
        IF (IDEATH.LQ.2) GO TO 690
        IF (TIME.LT.ETIM) GO TO 710
        IF (ETIMV(N).GE.0.) GO TO 692
        ETIMV(N)=ETIM
        DO 695 L=1,ND
            I=LM(L,N)
            IF (I.LE.0) GO TO 695
            EDISE(L,N)=X(I)
    
```


RUSS

JATE = 31249

18/23/01

```

C
C CALL SUBROUTINE BEARNG TO CALCULATE BEARING ELEMENT FORCES
C AND STIFFNESSES
C
C WRITE(6,1314) NEQ
1314 FORMAT(SX,'*****%*%*%*%*%*%*%*%*****%*%*%*%*%*%*****',
1 10X 'NEQ=',L10)
C WRITE(6,1315) (X(I),I=1,NEQ)
1315 FORMAT(SX,'X=',6E12.5)
C WRITE(6,1316) (Y(I),I=1,NEQ)
1316 FORMAT(SX,'Y=',6E12.5)
C WRITE(6,1317) (Z(I),I=1,NEQ)
1317 FORMAT(SX,'Z=',6E12.5)
C CALL BEARNG(B(NB),B(NB1),B(NB2),B(NB3),B(NB4),
C 1 B(NB5),B(NB6),B(NB7),B(NB8),B(NB9),
C 2 B(NB10),B(NB11),B(NB12),B(NB13),B(NB14),A(N3),
C 3 IELTD,X,Y,Z,XYZ,LM,S,RE,MAOR,ISTIFB,NDM,NUME)

```

RETURN

S T R E S S C A L C U L A T I O N S

```

C *** DATA PORTHOLE ***** (START)
C 800 IF (JNPORT.EQ.0 .OR. KPLUTE.NE.0) GO TO 802
C RECLAB = RECLB3
C WRITE (LUI) RECLAB,NG,(NPAR(I),I=1,20),KSTEP,TIME,NEGL
C *** DATA PORTHOLE ***** ( END )
C 802 IF (NG.GT.NEGL) GO TO 805
C
C GEOMETRIC AND MATERIAL LINEAR STRESS CALCULATION
C
C IPRINT=0
C IPRINT=JNPORT*KPLUTE
C
C DO 830 K=1,NUME
C IPST=IPS(N)
C IF (IPST.EQ.0) GO TO 830
C IF (IPRI.NE.0) GO TO 801
C IPRINT=IPRINT + 1
C IF (IPRINT.NE.1) GO TO 801
C WRITE (6,2500) NG
801 MTYPE=MATP(N)
C IELD=IELTD(N)
C ND=IELD*3
C
C IF (ITYPT.NE.1) GO TO 811

```

ORIGINAL PAGE IS
OF POOR QUALITY

RUSS _____ DATE = 81245

18/23/01

```

ND=1
EP=EPSIN(N)
I=LM(1,N)
IF (I.GT.0) EP=EP + X(I)/XYZ(1,N)
STN(1)=EP
STS(1)=E(MTYPE)*EP
PP(1)=AREA(MTYPE)*STS(1)
IF (IPRI.EQ.0) WRITE(6,2510) N,ND,PP(1),STS(1),STN(1)
GO TO 816
C
811 CONTINUE
DO 815 J=1,ND
DISP(J)=0.
I=LM(J,N)
IF (I.GT.0) DISP(J)=X(I)
815 CONTINUE
C
CALL LENTHI (XYZ(1,N),RL)
C
DO 814 L=1,NINT
EP=EPSIN(N)
R=XG(L,NINT)
CALL DEKIQ1(XYZ(1,N),R,RQ,XJ)
DO 813 K=1,ND
813 EP = EP + BC(K)*DISP(K)
STN(L)=EP
STS(L)=E(MTYPE)*EP
PP(L)=STS(L)*AREA(MTYPE)
IF (IPRI.EQ.0) WRITE(6,2510) N,L,PP(L),STS(L),STN(L)
814 CONTINUE
IF (IPRI.EQ.0) WRITE(6,2520)
C
C*** DATA PORTHOLE ***** (START)
C
816 IF (UNPORT.EQ.0 .OR. RECLDIE.NE.0) GO TO 830
RECLAB=RECLB4
DO 817 L=1,NINT
WRITE (LUI) RECLAB,L,PP(L),STS(L),STN(L)
817 CONTINUE
C
C*** DATA PORTHOLE ***** (END)
C
C
830 CONTINUE
RETURN
C
C
NONLINEAR STRESS CALCULATION
C
805 IPRNT=0
DO 810 K=1,NUME
IF (IDEATH.EQ.0) GO TO 910
ETIM=DAES(ETIMV(N))
IF (IDEATH.EQ.2) GO TO 890
IF (TIME.LT.ETIM) GO TO 810
GO TO 910

```

RUSS

DATE = 81245

18/23/01

```
350 IF (TIME.GT.ETIM) GO TO 810
310 IPST=IPS(N)
    IF (IPST.EQ.0) GO TO 810
    IF (IPRI.NE.0) GO TO 803
    IPRNT=IPRNT + 1
    IF (IPRNT.NE.1) GO TO 803
    GO TO (870,870,870,875,875,875,875,875,870), MODCL
870 *WRITE (6,2500) NG
    GO TO 803
875 *WRITE (6,2600) NG
C
803 IF (IPRI.EQ.0) *WRITE(6,2520)
    MTYPE=MATP(N)
    IELD=IELTD(N)
    ND=IELD*3
C
    IF (ITYPT.NE.1) GO TO 850
    ND=1
    RST(1)=XYZ(1,N)
    DISP(1)=0.
    I=LM(1,N)
    IF (I.GT.0) DISP(1)=X(I)
    IF (IDEATH.NE.1) GO TO 800
    RST(1)=RST(1) + EDISB(1,N)
    DISP(1)=DISP(1) + EDISB(1,N)
    GO TO 800
C
850 DO 851 L=1,ND
    RST(L)=XYZ(L,N)
    DISP(L)=0.
    I=LM(L,N)
    IF (I.GT.0) DISP(L)=X(I)
    IF (IDEATH.NE.1) GO TO 851
    RST(L)=RST(L) + EDISB(L,N)
    DISP(L)=DISP(L) + EDISB(L,N)
851 CONTINUE
C
860 CALL STIEN1 (N,AREA(MTYPE),PROP(1,MTYPE),WA(1,N),
    NUDGL(1,N),TEMPV1,TEMPV2,EPSIN(N),S,RE,0)
C
C*** DATA PORTHOLE *****-(START)
C
    IF (JNPORT.EQ.0 .OR. KPLOTE.NE.0) GO TO 810
    RECLAB=RECL4
    DO 880 L=1,NINT
    *WRITE (LUL) RECLAB,L,PP(L),STS(L),STN(L)
880 CONTINUE
C
C*** DATA PORTHOLE ***** (END)
C
810 CONTINUE
C
    RETURN
C
900 STOP
```

RUSS

DATE = 81245

13/23/01

CC
C

```
1000 FORMAT (15.4F10.0)
1010 FORMAT (8F10.0)
1100 FORMAT (515.2F10.0)
1200 FORMAT (1615)

2000 FORMAT (///37H MATERIAL CONSTANTS)
2010 FORMAT (///5H SET,10X,5H AREA,10X,5H DEN,10X,5H E)
2011 FORMAT (15.8E15.6)
2020 FORMAT (///5H SET,10X,5H AREA,10X,5H DEN,12X,5H POINT,9X,
1 6H STRAIN,9X,6H STRESS,11X,4H ETANZ 15.2E15.6,12X,15.2E15.6)
2021 FORMAT (47X,15.3E15.6)
2030 FORMAT (///27H MATERIAL CONSTANTS SET NO. 15/
1 14H AREA =.E15.6/14H DENSITY =.E15.6/)
2031 FORMAT (37H TEMP .....( PROP( 1= 6) )... =.6(E15.6)/
1 37H E .....( PROP( 7=12) )... =.0(E15.6)/
2 37H ALPHA .....( PROP(13=18) )... =.0(E15.6)/
3 37H REF TEMP .....( PROP( 19) )... =.E15.6/
2040 FORMAT (///5H SET,10X,5H AREA,10X,5H DEN,10X,5H E
1 10X,5H YIELD,10X,5H ET /)
2061 FORMAT (39H TEMP .....( PROP( 1= 6) )... =.0E15.6/
1 39H E .....( PROP( 7=12) )... =.0E15.6/
2 39H YIELD .....( PROP(13=18) )... =.0E15.6/
3 39H ET .....( PROP(17=24) )... =.0E15.6/
4 39H ALPHA .....( PROP(25=30) )... =.6E15.6/
5 39H REF TEMP .....( PROP( 31) )... =.E15.6/
6 39H CREEP LAW KEY .....( PROP( 32) )... =.F8=0/
7 39H CREEP LAW COEFFICIENTS ...../
8 39H A0 .....( PROP( 33) )... =.E15.6/
9 39H A1 .....( PROP( 34) )... =.E15.6/
A 39H A2 .....( PROP( 35) )... =.E15.6/
1 39H A3 .....( PROP( 36) )... =.E15.6/
2 39H A4 .....( PROP( 37) )... =.E15.6/
3 39H A5 .....( PROP( 38) )... =.E15.6/
4 39H A6 .....( PROP( 39) )... =.E15.6/
2071 FORMAT (6H PROP(.12,1H,12,3H) =.E15.6)
2200 FORMAT (///38H ELEMENT LINE ORDER MAT ION /
1 39X,7H INITIAL/
2 4X,1HN,3X,4HYIELD,3X,3HMIPS,2X,4HMTYP,4X,2HKG,
3 9X,6H STRAIN,12X,A6,9X,5HETIME,13X,
4 34H NODE(1) NODE(2) NODE(3) NODE(4) /)
2201 FORMAT (///4E ELEMENT NODAL INFLUENCE COEFFICIENTS //
1 4X,*N*,3X,*YIELD*,4X,*K*,5X,*ETIME*,10X,*RATOR (NODE(1))*,
2 3X,*STATOR (NODE(2))*, /)
2210 FORMAT (15.316,17.5X,2E15.6,3X,E15.6,2X,417)
2211 FORMAT (15.16,17.9X,E15.6,8X,15.9X,15)
2300 FORMAT (///16H ELEMENT GROUP =.12,16H (TRUSS / RUSS) /
1 17H ELEMENT NUMBER =.14/
2 7H IELD =.13,26H IS GREATER THAN NPAR(7) =.13/
3 5H STOP)
2310 FORMAT (///16H ELEMENT GROUP =.12,16H (TRUSS / RUSS) /
1 17H ELEMENT NUMBER =.14/
2 7H MYP =.13,27H IS GREATER THAN NPAR(10) =.13/5H STOP)
2500 FORMAT (1H1.46HS T H E S S C A L C U L A T I O N S F U R
1 17H ELEMENT GROUP .15,15H (TRUSSES) /4.4X,
2 7HELEMENT,10H LOCATION,8X,5HFURCC,12X,6HSTRESS,11X,
3 6HSTRAIN,9X,11HTEMPERATURE, /)
```

ORIGINAL PAGE IS
OF POOR QUALITY

A3-59

RUSS

DATE = 81245

18223/01

2510 FORMAT (19,110,3(2X,E15.6))
2610 FORMAT (1H)
2630 FORMAT (1H1,48HSTRESS CALCULATIONS FOR
1 27HELEMENT GROUP 15,15H (TRUSSES)-- //,4X,
2 7HELEMENT,10H LOCATION,5X,5HSTATE,8X,5HFORCE,12X,
3 6HSTRESS,11X,6HSTRAIN,9X,11HTEMPERATURE,/) .

END

ORIGINAL PAGE IS
OF POOR QUALITY.

A3-60

RUSS

DATE = 81245

OPTIONS IN EFFECT NOTERM, ID, EECDC, SOURCE, NCLIST, NODECK, LOAD, NOMAP, NOTEST
OPTIONS IN EFFECT NAME = RUSS , LINECNT = 60
STATISTICS SOURCE STATEMENTS = 492, PROGRAM SIZE = 21754
STATISTICS NL DIAGNOSTICS GENERATED

ORIGINAL PAGE IS
OF POOR QUALITY

A3-61

MAIN

DATE = 91229

09/18/44

C
C*****
C*****
C*****
C*****

C SUBROUTINE BEARNG(BD,BL,BC,VISC,PVAP,

C 1 TH1,TH2,PB1,PB2,NGRID,NSCLN,NPORT, ...
C 2 NFILM,KDFK,KDFC,R,IELTD,X,Y,Z,XYZ,LM,
C 3 S,RE,MADR,ISTIFB,NDM,NUME)

C
C P R O G R A M F U N C T I O N

C . TO READ BEARING ELEMENTS INFORMATION
C . TO CALCULATE AND ASSEMBLE ELEMENT STIFFNESSES

C E X E C U T I O N M O D E

C IND=0 BEARING INFORMATION IS READ
C IND=4 ELEMENT STIFFNESSES ARE CALCULATED
C AND ASSEMBLED
C

C IMPLICIT REAL*8 (A-H,O-Z)

C COMMON /EL/IND,ICOUNT,NPAR(20),NUMEG,NEGL,NEGNL,IMASS,

C 1 IJAMP,IJSTAT,NDOFDM,KLIN,IEIG,IMASSN,IDAMPN

C COMMON /DIM/NO,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,

C 1 N14,N15

C COMMON /CONST/DT,DTA,A0,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,

C 1 A11,A12,A13,A14,A15,A16,A17,A18,A19,A20,LOPE

C COMMON /VAR/NS,MODEX,IUPDT,KSTEP,ITEMAX,IEOREF,ITE,

C 1 KPRI,IREF,IEQUIT,IPRI,KPLOTN,KPLUTE

C COMMON /FORCEB/ F1P(100),F2P(100),FR(100)

C COMMON /BEAR/ IBEAR,MTOTS

C COMMON /BEAR2/ B(1)

C COMMON A(1)

C REAL A

BEARNG

JATE = 81229

09/18/44

```
C
C REAL B
C DIMENSION BD(1),BL(1),BC(1),VISC(1),PVAP(1),TH1(1),
C 1 TH2(1),PB1(1),PB2(1),NGRID(1),NSOLN(1),R(1),
C 2 NPORT(1),NFILM(1),KOFK(1),KOFCE(1),IELTO(1),
C 3 X(1),Y(1),Z(1),LM(NOM,1),XYZ(NOM,1),S(1),
C 1 RE(1),VELB(12),ACCB(12)
C IE(IND.GT.0) GO TO 420
C .....
C READ AND GENERATE ELEMENT
C INFORMATION
C .....
C WRITE(6,1999)
C N = 1
C IREAD = 5
C 160 READ (IREAD,1000) M,BDE,BLE,BCE,VISCE,PVAP,E,TH1E,
C 2 TH2E,PB1E,PB2E,NGRIDE,NSOLNE,NPORTE,
C 3 NFILME,KOFKE,KOFCE
C GENERATE MISSING ELEMENT INFORMATION IF NECESSARY
C 120 IF (M.NE.N) GO TO 200
C BDG = BDE
C BLG = BLE
C BCG = BCE
C VISC G = VISCE
C PVAP G = PVAP E
C TH1 G = TH1 E
C TH2 G = TH2 E
C PB1 G = PB1 E
C PB2 G = PB2 E
C NGRID G = NGRIDE
C NSOL G = NSOLNE
C NPORT G = NPORTE
C NFILM G = NFILME
C KOFK G = KOFKE
C KOFCE G = KOFCE
C 200 BD(N) = BDG
C BL(N) = BLG
C BC(N) = BCG
C VISC(N) = VISC G
C PVAP(N) = PVAP G
C TH1(N) = TH1 G
C TH2(N) = TH2 G
C PB1(N) = PB1 G
```

ORIGINAL PAGE IS
OF POOR QUALITY

A3-63

BEARING _____

DATE = 81229

09/18/64

PB2(N) = PB2G
NGRID(N) = NGRIDG
NSOLN(N) = NSOLNG
NPORT(N) = NPORTG
NFILM(N) = NFILMG
KOFK(N) = KOFKG
KOFK(N) = KOFKG
KOFK(N) = KOFKG

WRITE (6,2000) N,BD(N),BL(N),BC(N),VISC(N),PVAP(N),TH1(N),
1 TH2(N),PB1(N),PB2(N),NGRID(N),NSOLN(N),NPORT(N),
1 NFILM(N),KOFK(N),KOFK(N)

FUTURE
ADD RESTART CAPABILITIES IN THE FUTURE IF NECESSARY.

WRITE(6,1312) IND,N,NUME
1312 FORMAT(5X,'IND,N,NUME=',3I10)
IF (N.EQ.NUME) RETURN
N = N + 1
IF (N.GT.M) GO TO 160
GO TO 120

.....
ASSEMBLE INSTANTANEOUS FORCE VECTOR,
TANGENT STIFFNESS(AK) AND TANGENT DAMPING (AC).
MATRICES OF THE BEARING ELEMENT
.....

420 IF (IND.NE.4) RETURN

ND=12

DO 740 N=1,NUME

CHECK IF ELEMENT BELONGS TO THE CURRENT BLOCK _____

CALL ECHEGK (LM(1,N),ND,ICODE,IUPDT)

IF (ICODE.EQ.1) IELCPL = IELCPL + 1
IF (ICODE.EQ.1) GO TO 740

*****FUTURE*****

ADD BIRTH AND DEATH OPTIONS LATER IF NECESSARY

*****FUTURE*****

CALCULATE INERTIAL COORDINATES OF DAMPER
INSIDE AND OUTSIDE SURFACE CENTER-LINE

XRB = XYZ(1,N)

ORIGINAL PAGE IS
OF POOR QUALITY

BEARNG

DATE = 81229

09/18/44

```

C IF (LM(1,N).GT.0) XRB = X(LM(1,N))L
C YRB = XYZ(2,N)
C IF (LM(2,N).GT.0) YRB = X(LM(2,N))
C ZRB=XYZ(3,N)
C IF (LM(3,N).GT.0) ZRB = X(LM(3,N))
C XSB=XYZ(4,N)
C IF (LM(7,N).GT.0) XSB = X(LM(7,N))
C YSB=XYZ(5,N)
C IF (LM(8,N).GT.0) YSB = X(LM(8,N))
C ZSB=XYZ(6,N)
C IF (LM(9,N).GT.0) ZSB = X(LM(9,N))
DO 725 L=1,ND
VELB(L)=0.
ACCB(L)=0.
I=LM(L,N)
IF (I.GT.0) VELB(L)=Y(I)
IF (I.GT.0) ACCB(L)=Z(I)
C
C 725 CONTINUE
C
C XDTR = VELB(1)
C YDTR = VELB(2)
C ZDTR = VELB(3)
C XDTS = VELB(7)
C YDTS = VELB(8)
C ZDTS = VELB(9)
C XDT2R = ACCB(1)
C YDT2R = ACCB(2)
C ZDT2R = ACCB(3)
C XDT2S = ACCB(7)
C YDT2S = ACCB(8)
C ZDT2S = ACCB(9)
C WRITE (6,2500) XRB,YRB,ZRB,XSB,YSB,ZSB
2500 FORMAT (7X,'X=ROTOR',5X,'Y=ROTOR',5X,'Z=ROTOR',4X,'X=STATOR',
1 5X,'Y=STATOR',4X,'Z=STATOR',//,5X,6(2X,D13.6))
C WRITE (6,2550) XDTR,YDTR,ZDTR,XDTS,YDTS,ZDTS
2550 FORMAT (7X,'ROTOR XDT',4X,'ROTOR YDT',4X,'ROTOR ZDT',
1 3X,'STATOR XDT',4X,'STATOR YDT',3X,'STATOR ZDT',//,
2 5X,6(2X,D13.6))
C WRITE (6,2600) XDT2R,YDT2R,ZDT2R,XDT2S,YDT2S,ZDT2S
2600 FORMAT (7X,'ROTOR XDT2',4X,'ROTOR YDT2',3X,'ROTOR ZDT2',
1 3X,'STATOR XDT2',3X,'STATOR YDT2',3X,'STATOR ZDT2',//,
2 5X,6(2X,D13.6))
C
C GET BACK THE OTHER ELEMENT PROPERTIES TO BE
C USED IN SQUEEZE
C
C BDE = BD(N)
C BLE = BL(N)
C BCE = BC(N)
C VISCE = VISC(N)
C PVAPE = PVAP(N)
C THIE = TH1(N)
C TH2E = TH2(N)
C PBIE = PB1(N)

```

BEARING

DATE = 81229

09/18/44

PB2E = PB2(N)
NGRIDE = NGRID(N)
NSULNE = NSULN(N)
NPORTE = NPOR(T(N)
NFILME = NFILM(N)
KOFKE = KOFK(N)
KOFCE = KOFK(N)

C
C

DU 726 I=1,ND
726 RE(I)=0.
DU 727 I=1,78
727 S(I)=0.

C

C

C

C

C

C

C

C

C

C

C

C

CALL SQUEEZ TO CALCULATE INSTANTANEOUS FORCE VECTOR,
STIFFNESS MATRIX(AK) AND DAMPING MATRIX (AC).

CALL SQUEEZ (BDE,BLE,BCE,VISCE,TH1E,TH2E,PB1E,PB2E,

1 YRB,ZRB,YDTR,ZDTR,YSB,ZSB,YDTS,ZDTS,AK11,AK12,
2 AK22,AC11,AC22,F1,F2,NGRIDE,NSULNE,NPORTE,
3 KOFKE,KOFCE,NFILME,PVAPÉ)

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

S(13) = AK11
S(14) = AK12
S(19) = -AK11
S(20) = -AK12
S(24) = AK22
S(29) = -AK12
S(30) = -AK22
S(64) = AK11
S(65) = AK12
S(69) = AK22

RE(2) = -F1
RE(3) = -F2
RE(8) = F1
RE(9) = F2

F1P(N)=F1
F2P(N)=F2
FR(N)=DSURT(F1*F1+F2*F2)

ADD ELEMENT VECTOR RE TO EXTERNAL LOAD VECTOR

WRITE (6,981) (S(I),I=1,78)
981 FORMAT (5X,'BEARING STIFFNESS (BEARING):',//,
1 8(3X,013-6))

CALL ADJUAN (A(MADR),A(NI),S,RE,LM(1,N),ND,2).

ADD ELEMENT STIFFNESS (AK)

ORIGINAL PAGE IS
OF POOR QUALITY

BEARING

DATE = 81229

09/18/44

```

C
C   WRITE (6,461) (LM(I,N),I=1,ND)
461  FORMAT (5X,'LMARRAY(BEARING):',/,5X,8(2X,112))
C   WRITE (6,750) IREF
750  FORMAT (/,/,,'$$$$$$$$$$$$$$$$$$$$ IREF (BEARING) $$$$$$',/,15,/,/,)
      IF (IREF) 641,739,641
739  CALL ADDBAN (A(N4),A(N1),S,RE,LM(1,N),ND,1)
C
C   ADD EFFECT OF BEARING DAMPING MATRIX (AC)
C
641  IF (KOFCE.EQ.0) GO TO 740
      DU 741 I=1,ND
741  RE(I)=0.
      DU 742 I=1,78
742  S(I)=0.
      RE(2)= (A4-1)*YDTR+A5*YDT2R
      RE(3)= (A4-1)*ZDTR+A5*ZDT2R
      RE(8)= (A4-1)*YDTS+A5*YDT2S
      RE(9)= (A4-1)*ZDTS+A5*ZDT2S
C
C   WRITE (6,749) A4,A5,YDTR,YDT2R,YDTS,YDT2S
749  FORMAT (5X,'A4,A5,YDTR,YDT2R,YDTS,YDT2S:',/,/,8(3X,D12.5))
      WRITE (6,759) (RE(I),I=1,ND)
759  FORMAT (5X,'NUMERICAL TIME INTEGRATION VALUES OF VELOCITIES:',
1     //,8(3X,D12.5))
      WRITE (6,751) AC11,AC22
751  FORMAT (5X,'DAMPING MATRIX:',/,2(10X,D13.6))
      S(13)=AC11
      S(19)=-AC11
      S(24)=AC22
      S(30)=-AC22
      S(64)=AC11
      S(69)=AC22
C
C
      LL=0
      LJ=1
      DO 743 L=1,ND
        I=LM(L,N)
        DU 744 K=L,ND
          LL=LL+1
          IF (I.EQ.0) GO TO 744
          R(I)=R(I)-S(LL)*RE(K)
C   WRITE (6,752) L,I,K,LL,R(I),S(LL),RE(K)
752  FORMAT (5X,'L,I,K,LL,R(I),S(LL),RE(K) ',
1     //,35X,4(4X,15),10X,3(4X,D12.5))
744  CONTINUE
      IF (L.EQ.ND) GO TO 743
      LI=L+1
      DU 745 J=LI,ND
        LJ=LJ+1
        IF (J.EQ.ND) LJ=LJ+1
        IJ=LM(J,N)
        IF (IJ.EQ.0) GO TO 745
        R(IJ)=R(IJ)-S(LJ)*RE(L)
C   WRITE (6,765) L,LI,J,LJ,IJ,R(IJ),S(LJ),RE(L)
765  FORMAT (5X,'L,LI,J,LJ,IJ,R(IJ),S(LJ),RE(L) ',
1     //,35X,5(4X,15),3(4X,D12.5))

```

ORIGINAL PAGE IS
OF POOR QUALITY

A3-67

BEARNG

DATE = 81229

09213744

745 CONTINUE
743 CONTINUE
WRITE (6,753) (R(I),I=1,4)
753 FORMAT (5X,'DAMPED EXTERNAL LOAD VECTOR',//,8(3X,D12.5))
740 CONTINUE

C
RETURN

C
1000 FORMAT (110,6F10.0,/,3F10.0,6I5)

C
1999 FORMAT (////////,4X,'BEARING ELEMENT INFORMATION',//,4X,

C
1 'N BD BL BC VISC.

C
2 , 'PVAP TH1 TH2 PB1 PB2 NG

C
3 NS NP NF .KK KC')

C
2000 FORMAT (15,9D11.4,4I5,2I4)

C
END

ORIGINAL PAGE IS
OF POOR QUALITY.

FORTRAN IV G1 RELEASE 2.0

BEARNG

DATE = 81229

OPTIONS IN EFFECT NOTERM, ID, EBCDIC, SOURCE, NOLIST, NUDECK, LOAD, NDMAP
OPTIONS IN EFFECT NAME = BEARNG, LINECNT = 60
STATISTICS SOURCE STATEMENTS = 182, PROGRAM SIZE = 6808
STATISTICS NO DIAGNOSTICS GENERATED

ORIGINAL PAGE IS
OF POOR QUALITY

MAIN

DATE = 8122)

09/18/44

```

C
C
C *****
C *****
C *****
C
C   SUBROUTINE SIZE3(N)
C   IMPLICIT REAL *8(A-H,O-Z)
C   COMMON B(1)
C   COMMON /BEAR/ IBEAR, MTOTB
C
C   WRITE (6,1000)
C   WRITE (6,1001) N
C   IF (N.LT.MTOTB) GO TO 20
C   WRITE (6,1004) N
C   STOP
20  WRITE (6,1003)
1000 FORMAT (/,'.4X,'CORE INFORMATION... (DECIMAL)...')
C 1001 FORMAT (4X,' REQUESTED CORE FOR BEARING INFORMATION =',5X,I6)
C 1004 FORMAT (4X,' NOT AVAILABLE STOP',/,'.4X,'***CHANGE MTOTB',
C 1
C 1 '(IN MAIN) TO (AT LEAST) ',18,'*****')
C 1008 FORMAT (4X,' OBTAINED ',//)
C   RETURN
C   END

```

ORIGINAL PAGE IS
OF POOR QUALITY.

A3-70

FORTRAN IV G1 RELEASE 2.0 _____ SIZEB _____ DATE = 81229 _____

OPTIONS IN EFFECT NOTERM, ID, EBCDIC, SOURCE, NOLIST, NODECK, LOAD, NOMAP:
OPTIONS IN EFFECT NAME = SIZEB = _____ LINECNT = 60
STATISTICS SOURCE STATEMENTS = _____ L6, PROGRAM SIZE = 040
STATISTICS NO DIAGNOSTICS GENERATED

ORIGINAL PAGE IS
OF POOR QUALITY

MAIN

DATE = 81229

09/18/44

SUBROUTINE SQUEEZ(AD,AL,AC,AISC,ATH1,ATH2,ABL,AB2,U,V,...

1 UDT,VDT,UB,VB,UBT,VBT,AK11,AK12,AK22,AC11,AC22,F1,F2,NGRIA,NSOL,A,

2 NPORA,KAFK,KAFC,NFILA,PVAA)---

NONLINEAR TIME-TRANSIENT SQUEEZE-FILM DAMPER INTERACTIVE ELEMENT
THIS CODE COMPUTES INSTANTANEOUS FORCE VECTOR AND ITS
SPATIAL GRADIENTS, I.E., THE TANGENT STIFFNESS AND DAMPING
MATRICES.

NOMENCLATURE

INPUT

BD=NOMINAL DAMPER ANNULUS DIAMETER(IN)

BL=NOMINAL DAMPER ANNULUS LENGTH(IN)

BC=DAMPER ANNULUS RADIAL CLEARANCE(IN)

VISC=DAMPER LUBRICANT VISCOSITY(REYNS)

PVAP=FILM RUPTURE PRESSURE(Psia)

THT(1)=POSITION ANGLE OF LUBRICANT PORT-1(DEG)

THT(2)=POSITION ANGLE OF LUBRICANT PORT-2(DEG)

PB(1)=SPECIFIED BOUNDARY PRESSURE AT PORT-1(Psia)

PB(2)=SPECIFIED BOUNDARY PRESSURE AT PORT-2(Psia)

NGRID=NUMBER OF FINITE-DIFFERENCE GRID POINTS PER DAMPER ARC(JDD)

NSOLN=1,LONG-BEARING SOLUTION USED

 =2,SHORT-BEARING(PARABOLIC) SOLUTION USED

 =3,FOURIER-SERIES 2-D CONVERGENT SOLUTION USED

NPORT=NUMBER OF LUBRICANT PORTS(0,1 OR 2)

IF NPORT=0, JOINED-BOUNDARY CONDITION IS USED

SQUEEZ

DATE = 81224

09/18/44

NFILM=NUMBER OF IDENTICAL ANNULI FOR THE DAMPER _____
KOFK=0, STIFFNESS MATRIX NOT COMPUTED _____
KOFK=1, STIFFNESS MATRIX COMPUTED
KDFC=0, DAMPING MATRIX NOT COMPUTED
KDFC=1, DAMPING MATRIX COMPUTED
X=X=INERTIAL COORDINATE OF DAMPER INSIDE SURFACE CENTER=LINE(IN) _____
Y=Y=INERTIAL COORDINATE OF DAMPER INSIDE SURFACE CENTER=LINE(IN) _____
XDT=X=INERTIAL VELOCITY OF INSIDE SURFACE CENTER=LINE(IN/SEC) _____
YDT=Y=INERTIAL VELOCITY OF INSIDE SURFACE CENTER=LINE(IN/SEC) _____
XB=X=INERTIAL COORDINATE OF DAMPER OUTSIDE SURFACE CENTER=LINE(IN) _____
YB=Y=INERTIAL COORDINATE OF DAMPER OUTSIDE SURFACE CENTER=LINE(IN) _____
XBT=X=INERTIAL VELOCITY OF DAMPER OUTSIDE SURFACE CENTER=LINE(IN) _____
YBT=Y=INERTIAL VELOCITY OF DAMPER OUTSIDE SURFACE CENTER=LINE(IN) _____
OUTPUT

F1=X=FORCE COMPONENT ON INSIDE DAMPER SURFACE(LBS)
F2=Y=FORCE COMPONENT ON INSIDE DAMPER SURFACE(LBS).
FORCE COMPONENTS ON OUTSIDE DAMPER SURFACE ARE EQUAL BUT OPPOSITE
F1 AND F2 BECAUSE FLUID INERTIA EFFECTS ARE NEGLECTED
SYMMETRIC PORTION OF STIFFNESS MATRIX:
AK11=KXX(LBS/IN)
AK12=KXY(LBS/IN)=KYY
AK22=KYY(LBS/IN)
DIAGONAL PORTION OF DAMPING MATRIX
AC11=CXX(LB*SEC/IN)
AC22=CYY(LB*SEC/IN)
IMPLICIT REAL*8 (A-H, O-Z)
COMMON/INPUT1/BD, BL, BC, VISC, THT(2), PB(2), PVAP
COMMON/INPUT2/NGRID, NSOLN, NPORT, NFILM

ORIGINAL PAGE IS
OF POOR QUALITY.

SQUEEZ

DATE = 81229

09/18/44

```

COMMON/COORD/X,Y,XDT,YDT,XB,YB,XBT,YBT
COMMON/FILM/TH(101),H(101),DHOX(101),DHDT(101),STH(101),CTH(101),
1DXD(2),ALFA(2).
COMMON/WORK/FNGDM1,KB,KOUNT
COMMON/INC/HMIN,VEL,DELS,DELST
DIMENSION A(101),B(101),C(101),E(101),RH(101),P(101),ARG1(101),
1 ARG2(101),D(101)
ALLOCATE INPUT NAMES

```

```

BD = AD
BL = AL
BC = AC
VISC = AISC
THT(1) = ATH1
THT(2) = ATH2
PB(1) = AB1
PB(2) = AB2
PVAP = PVAA
X = U
Y = V
XDT = UDT
YDT = VDT
XB = UB
YB = VB
XBT = UBT
YBT = VBT
NGRID = NGRIA
NSOLN = NSOLA
NPORT = NPORA
KOFK = KAFK
KOFK = KAFC
NFILM = NFILA
PI = 3.141592654

```

```

C WRITE (6,2500) X,Y,XB,YB,XDT,YDT,XBT,YBT
2500 FORMAT (7X,'X=ROTUR',5X,'Y=ROTUR',5X,'X=STATOR',4X,'Y=STATOR',
3 5X,'ROTUR XDT',
1 5X,'ROTUR YDT',4X,'STATOR XDT',5X,'STATOR YDT',//,
2 5X,(2X,D13.6))

```

```

C WRITE (6,1999)
1999 FORMAT (////////,4X,'BEARING ELEMENT INFORMATION',//,4X,

```

```

1      'BD      BL      BC      VISC
2      , 'PVAP    TH1    TH2    PB1    PB2

```

```

3      NS      NP      NF      KK      KC
C WRITE (6,10) BD, BL, BC, VISC, PVAP, THT(1), THT(2), PB(1), PB(2),
1      NGRID, NSOLN, NPORT, NFILM, KOFK, KOFK
C 10 FORMAT (9D11.4,4I5,2I4)
C

```

ORIGINAL PAGE IS
OF POOR QUALITY

A3-74

SQUEEZ

JATE = 81229

09/18/44

```
C SET UP
C
KOUNT = 1
MTEST = NGRID + 1
NTEST = MTEST/2
KTEST = 2*NTEST

C IF(KTEST.NE.MTEST) NGRID=NGRID-1
C
FNGDM1 = FLOAT(NGRID-1)
THT(1) = THT(1)*PI/180.
C
IF(NPORT.LT.2) GO TO 20
C
THT(2) = THT(2)*PI/180.
C
IF(THT(2).LT.THT(1)) THT(2)=THT(2)+2.*PI
C
ALFA(1) = THT(2)-THT(1)
ALFA(2) = 2.*PI-ALFA(1)
DXD(1) = 0.5*BD*ALFA(1)/FNGDM1
DXD(2) = 0.5*BD*ALFA(2)/FNGDM1
C
GO TO 40
C
20 DXD(1) = BD*PI/FNGDM1
ALFA(1) = 2.*PI
C
40 CONTINUE
C
WRITE(6,10)DXD(1),DXD(2)
AKXX = 0.0
AKXY = 0.0
AKYX = 0.0
AKYY = 0.0
ACXX = 0.0
ACXY = 0.0
ACYX = 0.0
ACYY = 0.0
FX = 0.0
FY = 0.0
C
C BRANCH ACCORDING TO SOLUTION DESIGNATED
C
60 CONTINUE
CALL INCRNT
GO TO (100,100,300),NSCLN
100 CONTINUE
C
C SOLVE FOR SQUEEZE FILM PRESSURE DISTRIBUTION
C
DO 190 KB=1,NPORT
P(1)=PB(KB)
IF(KB.EQ.1) P(NGRID)=PB(2)
IF(KB.EQ.2) P(NGRID)=PB(1)
DX=XD(KB)
CALL OFILM
A(2)=0.0
```

SQUEEZ

DATE = 81229

09/18/44

```

B(2) = P(1)
DO 110 K=2,NGRID
COEF1 = H(K)**3/OX**2.
COEF2 = (1.5*H(K)**2)*DHDX(K)/ZDX
C(K) = 2.*COEF1
D(K) = COEF1+COEF2.
E(K) = COEF1-COEF2
RH(K) = 12.*VISC*DHD.T(K)
IF(NSOLN.EQ.1) GO TO 110
C(K) = C(K)-(8.*H(K)**3)/BL**2
110 CONTINUE
NGRD = NGRID-1
DO 120 K=2,NGRD
FOCTR = C(K)+E(K)*A(K)
A(K+1) = D(K)/FOCTR
120 B(K+1) = (RH(K)-E(K)*B(K))/FOCTR
NGR = NGRID-2.
DO 130 K=1,NGR
J = NGRID-K
P(J) = A(J+1)*P(J+1)+B(J+1)
IF(P(J).LT.PVAP) P(J)=PVAP
130 CONTINUE
C WRITE(6,135)(P(J),J=1,NGRID)
135 FORMAT(2X,14E9.2)
C
C INTEGRATE PRESSURE DISTRIBUTION TO GET X AND Y FORCE COMPONENTS
C
DO 140 K=1,NGRID
ARG1(K) = P(K)*CTH(K)
140 ARG2(K) = P(K)*STH(K)
A1 = ARG1(1)+ARG1(NGRID)
A2 = ARG2(1)+ARG2(NGRID)
B1 = 0.0
B2 = 0.0
DO 150 K=2,NGRD.2
B2 = B2+ARG2(K)
150 B1 = B1+ARG1(K)
C1 = 0.0
C2 = 0.0
DO 160 K=3,NGR.2
C1=C1+ARG1(K)
160 C2=C2+ARG2(K)
DTHET=2.*UX/BD
FACTR=FLUAT(NFILM)*DTHET/3.
XQ=FACTR*(A1+4.*B1+2.*C1)
YQ=FACTR*(A2+4.*B2+2.*C2)
IF(NSOLN.EQ.1) FACTR = BD*9L/2.
IF(NSOLN.EQ.2) FACTR = BD*9L/3.
XQ=XQ*FACTR
YQ=YQ*FACTR
FX=FX+XQ
FY=FY+YQ
190 CONTINUE
GO TO (500,520,540,560,580),KUUNT
500 F1=FX
F2=FY
IF((KOFK.EQ.0).AND.(KOFK.EQ.0)) RETURN
IF(KOFK.EQ.0) KUUNT=4

```


ORIGINAL PAGE IS
OF POOR QUALITY

A3-76

SQUEEZ.

DATE = 81229

09/18/44

```
IF(KOPK.EQ.0) GO TO 60-
GO TO 590
520 AKXX=(FX=F1)/DELS
    AKYX=(FY=F2)/DELS
    GO TO 590
540 AKXY=(FX=F1)/DELS
    AKYY=(FY=F2)/DELS
C   WRITE (6,657) AKXX,AKXY,AKYY
657 FORMAT (5X,'BEARING STIFF.(SQUEEZ):',3(5X,013.6))
    AK11=AKXX
    AK22=AKYY
    AK12=0.5*(AKXY+AKYX)
C   WRITE (6,657) AK11,AK12,AK22-
C   WRITE (6,658) F1,F2
658 FORMAT (5X,'FORCES ON THE ROTOR:',//,8(2X,013.6))
    IF(KOFC.EQ.0) RETURN
    GO TO 590
560 ACXX=(FX=F1)/DELST
    ACYX=(FY=F2)/DELST
    GO TO 590
580 ACXY=(FX=F1)/DELST
    ACYY=(FY=F2)/DELST
    GO TO 600
590 KOUNT=KOUNT+1
    GO TO 60
600 CONTINUE
    AC11=ACXX
    AC22=ACYY
    RETURN
300 WRITE(6,700)
    RETURN
700 FORMAT(1H1//5X*'FOURIER=SERIES 2=0 OPTION NOT READY FOR USE'//)
END
```

ORIGINAL PAGE IS
OF POOR QUALITY

FORTRAN IV G1 RELEASE 2.0

SQUEEZ

DATE = 91229

OPTIONS IN EFFECT NOTERM, ID, EBCDIC, SOURCE, NOLIST, NODECK, LOAD, NOMAP
OPTIONS IN EFFECT NAME = SQUEEZ, LINECNT = 80
STATISTICS SOURCE STATEMENTS = 159, PROGRAM SIZE = 11932
STATISTICS NO DIAGNOSTICS GENERATED

ORIGINAL PAGE IS
OF POOR QUALITY

MAIN—

DATE = 81229

09/18/44

```

C
C*****
C*****
C*****
C
C      SUBROUTINE DFILM
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C
C      COMMON/INPUT1/BD,BL,BC,VISC,THT(2),PB(2),PVAP
C
C      COMMON/INPUT2/NGRID,NSOLN,NPORT,NFILM
C
C      COMMON/COORD/X,Y,XDT,YDT,XB,YB,XBT,YBT
C
C      COMMON/FILM/TH(101),H(101),DHDX(101),DHDT(101),STH(101),CTH(101),
C
C      L      DXD(2),ALFA(2)
C
C      COMMON/WORK/FNGDM1,KB,KOUNT
C
C      COMPUTE DISTRIBUTIONS FOR H,DHDX AND DHDT
C
C      R=BD/2.
C      DANG=ALFA(KB)/FNGDM1
C      IF(KB.EQ.1) TH(1)=THT(1)
C      IF(KB.EQ.2) TH(1)=THT(2)
C      DO 50 K=2,NGRID
50  TH(K)=TH(K-1)+DANG
C      CDG=DCOS(DANG)
C      SDG=DSIN(DANG)
C      CTH(1)=DCOS(TH(1))
C      STH(1)=DSIN(TH(1))
C      DO 100 K=2,NGRID
100  CTH(K)=CTH(K-1)*CDG-STH(K-1)*SDG
C      STH(K)=STH(K-1)*CDG+CTH(K-1)*SDG
C      DO 150 K=1,NGRID
150  H(K)=BC-(X-XB)*CTH(K)-(Y-YB)*STH(K)
C      IF(H(K).LT.0.) GO TO 500
C      DHDX(K)=((X-XB)*STH(K)-(Y-YB)*CTH(K))/R
150  DHDT(K)=-((XDT-XBT)*CTH(K)-(YDT-YBT)*STH(K))
C      RETURN
500  WRITE(6,600) KB,K
C      STOP
501  RETURN
600  FORMAT(1H1//5X'NEGATIVE FILM THICKNESS ENCOUNTERED AT KB,K=',
1  2I10)
C      END

```

ORIGINAL PAGE IS
OF POOR QUALITY

A3-79

FORTRAN IV G1 RELEASE 2.0

DEILM

DATE = 81229

OPTIONS IN EFFECT NOTERM, ID, EBCDIC, SOURCE, NOLIST, NODECK, LOAD, NUMAP
OPTIONS IN EFFECT NAME = DFILM * LINECNT = 60
STATISTICS SOURCE STATEMENTS = 31 PROGRAM SIZE = 1034
STATISTICS NO DIAGNOSTICS GENERATED

ORIGINAL PAGE IS
OF POOR QUALITY

MAIN

DATE = 81229

09/13/44

```

C*****
C*****
C*****
C
C      SUBROUTINE INCRNT.
C
C      THIS SUBROUTINE INCREMENTS DISPLACEMENTS AND VELOCITIES FOR
C      THE PURPOSE OF NUMERICALLY DIFFERENTIATING THE SQUEEZE FILM
C      FORCE VECTOR.
C
C      IMPLICIT REAL*8 (A-H,O-Z).
C
C      COMMON/INPUT1/BD,BL,BC,VISC,THI(2),PB(2),PVAP
C
C      COMMON/COORD/X,Y,XDT,YDT,XB,YB,XBT,YBT
C
C      COMMON/WORK/FNGDMI,KB,KOUNT
C
C      COMMON/INC/HMIN,VEL,DELS,DELST
C
C      IF(KOUNT.NE.1) GO TO 10
C      SX=X
C      SY=Y
C      SXDT=XDT
C      SYDT=YDT
C      RETURN
10  CONTINUE
C      X=SX
C      Y=SY
C      XDT=SXDT
C      YDT=SYDT
C      HMIN=C*DSQRT((X-XB)**2+(Y-YB)**2)
C      VEL=DSQRT((XDT-XBT)**2+(YDT-YBT)**2)
C      DELS=0.05*HMIN
C      DELST=0.05*VEL
C      IF(KOUNT.EQ.2) X=X+DELS
C      IF(KOUNT.EQ.3) Y=Y+DELS
C      IF(KOUNT.EQ.4) XDT=XDT+DELST
C      IF(KOUNT.EQ.5) YDT=YDT+DELST
C      RETURN
C      END

```

ORIGINAL PAGE IS
OF POOR QUALITY

FORTRAN. IV G1 RELEASE 2.0 _____ INCRNT. _____ DATE = 81229_

OPTIONS IN EFFECT NOTERM, ID, EBCDIC, SOURCE, NOLIST, NODECK, LOAD, NUMAP ..
OPTIONS IN EFFECT NAME = INCRNT . LINECNT = 63
STATISTICS SOURCE STATEMENTS = 27, PROGRAM SIZE = 740
STATISTICS NO DIAGNOSTICS GENERATED

ORIGINAL PAGE IS
OF POOR QUALITY

MAIN

DATE = 81229

09/18/44

```

C      DATA SET TRUSS      AT LEVEL TMP AS OF 08/17/81
C *CDC* *DECK UVL20
C *CDC*   OVERLAY (ADINA,2,0)
C *CDC* *DECK TRUSS.
C *UNI* )FOR, IS N, TRUSS, R, TRUSS
C *CDC*   PROGRAM TRUSS
C      SUBROUTINE TRUSS
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C
C      .....
C
C      S T O R A G E
C
C      N102   DEN      (NECESSARY FOR STATIC PROBLEMS, TOO)
C      N103   AREA
C      N105   LM       CONNECTIVITY
C      N106   XYZ     ELEMENT NODAL COORDINATES
C      N107   MATP    MATERIAL PROPERTY SET NUMBER
C      N108   EPSIN   INITIAL STRAIN
C      N109   IPS     STRESS PRINTING FLAG
C      N110   ETIMV   ELEMENT BIRTH/DEATH TIME (IF IDEATH.GT.0)
C      N111   EDISB   ELEMENT DISPS AT BIRTH TIME (IF IDEATH.EQ.1)
C      N112   WA=IWA  WORKING ARRAY (SEE IDWAS)
C      N113   PROP    MATERIAL CONSTANTS (SEE NMCON)
C      N114   NODGL   GLOBAL NODE NUMBERS (SEE NDWS)
C      N115   IELTD   ELEMENT NUMBER OF NODES
C
C      .....
C
C      COMMON /DIM/  N0,N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
C      COMMON /SOL/  NUMNP,NEQ,NWK,NWL,NWC,NUMEST,MIDEST,MAXEST,NSTE,WA
C      COMMON /EL/   INO,ICOUNT,NPAR(20),NUMEG,NEGL,NEGL,IMASS,IDAMP,ISTAT
C      COMMON /VAR/  NG,NOEX,IUPDT,KSTEP,ITEMAX,IEQREF,ITE,KPRI,
C      IREF,IEQUIT,IPRI,KPLOTN,KPLUTE
C      COMMON /DPR/  ITWO
C      COMMON /JUNK/ HED(12),MTOT
C      COMMON /ELGTH/ NFIRST,NLAST,NBCEL
C      COMMON /ELSTP/ TIME,IDTHF
C      COMMON /PRCON/ IDATWR,IPRIC,NPB,IDC,IVC,IAC,IPC,IPNODE(3,15)
C      COMMON /PORT/ INPORT,INPORT,NPUTSV,LUNODE,LU1,LU2,LU3,JDC,JVC,JAC
C      COMMON /TMODEL/ LINEL,NONEL,ITEL,ISPEL,KINPEL,IEPCL,IEPCK,MODMAX
C      COMMON A(1)
C      COMMON /TEMPRT/ TEMP1,TEMP2,ITEMPR,ITP96,N6A,N6B
C      REAL A
C      DIMENSION IA(1)
C      EQUIVALENCE (A(1),IA(1))
C
C      DIMENSION DATA(20)
C
C      EQUIVALENCE (NPAR(1),NPAR1), (NPAR(2),NUME), (NPAR(3),INDNL),
C      (NPAR(4),IDEATH), (NPAR(5),ITYPT), (NPAR(7),MXNOUS),
C      (NPAR(10),NINT), (NPAR(15),MODEL), (NPAR(16),NUMMAT),
C      (NPAR(17),NCON), (NPAR(19),ITEMP)

```

ORIGINAL PAGE IS
OF POOR QUALITY.

TRUSS

DATE = 81229

09/18/44

DIMENSION NMCON(8),LDWAS(8),NDWS(8)
DATA RECLB1 /8BHTYPE = 1

DATA NMGUN / 1, 0, 19, 3, 3, 39, 39, 0/
DATA LDWAS / 0, 0, 0, 2, 2, 9, 9, 0/
DATA NDWS / 0, 0, 1, 0, 0, 1, 1, 0/

IF (IND.NE.0) GO TO 100

INPUT PHASE

CHECK ON RANGE AND SET DEFAULTS FOR NPAR VECTOR

ISTOP=0

LINEL=1
NONEL=2
ITEL=3
ISPEL=4
KINREL=5
IEPCI=6
IEPCK=7
MODMAX=8

IF (NUME.GT.0) GO TO 10
ISTOP=ISTOP+1
IF (ISTOP.EQ.1) WRITE (6,2100) NG
ISUB=2
IRANGE=1
WRITE (6,2400) ISTOP,ISUB,IRANGE,ISUB,NPAR(ISUB)

10 IF (INDNL.GE.0 .AND. INDNL.LE.2) GO TO 15
ISTOP=ISTOP+1
IF (ISTOP.EQ.1) WRITE (6,2100) NG
ISUB=3
WRITE (6,2200) ISTOP,ISUB,ISUB,NPAR(ISUB)

15 IF (IDEATH.NE.0) IDTHF=1
IF (IDEATH.GE.0 .AND. IDEATH.LE.2) GO TO 20
ISTOP=ISTOP+1
IF (ISTOP.EQ.1) WRITE (6,2100) NG
ISUB=4
WRITE (6,2200) ISTOP,ISUB,ISUB,NPAR(ISUB)

20 IF (ITYPT.EQ.3) GO TO 25
IF (ITYPT.GE.0 .AND. IITYPT.LE.1) GO TO 25
ISTOP=ISTOP+1
IF (ISTOP.EQ.1) WRITE (6,2100) NG
ISUB=5
WRITE (6,2200) ISTOP,ISUB,ISUB,NPAR(ISUB)

25 IF (MXNODS.LE.0) MXNODS=2
IF (MXNODS.LE.4) GO TO 30
ISTOP=ISTOP+1
IF (ISTOP.EQ.1) WRITE (6,2100) NG

ORIGINAL PAGE IS
OF POOR QUALITY

A3-84

TRUSS

DATE = 8.12.29

09/18/44

```
ISUB=7
IRANGE=4
WRITE (6,2300) ISTOP,ISUB,IRANGE,ISUB,NPAR(ISUB)
C
30 NIPT=1
IF (ITYPT.EQ.1) GO TO 34
IF (MXNODS.GT.2) GO TO 32
IF (MODEL.EQ.ITEL) GO TO 32
IF (MODEL.EQ.IEPCI .OR. MODEL.EQ.IEPCK) GO TO 32
GO TO 34
32 NIPT=NINT
IF (NINT.LE.0) NIPT=2
34 NINT=NIPT
IF (NINT.LE.4) GO TO 35
ISTOP=ISTOP+1
IF (ISTOP.EQ.1) WRITE (6,2100) NG
ISUB=10
IRANGE=4
WRITE (6,2300) ISTOP,ISUB,IRANGE,ISUB,NPAR(ISUB)
C
35 IF (MODEL.LE.0) MODEL=1
IF (MODEL.LE.MODMAX) GO TO 40
ISTOP=ISTOP+1
IF (ISTOP.EQ.1) WRITE (6,2100) NG
ISUB=15
WRITE (6,2300) ISTOP,ISUB,MODMAX,ISUB,NPAR(ISUB)
C
40 IF (NUMMAT.LE.0) NUMMAT=1
C
IF (MODEL.EQ.MODMAX) GO TO 50
IF (MODEL.NE.NONEL) GO TO 45
IF (NCUN.GE.4) GO TO 50
ISTOP=ISTOP+1
IF (ISTOP.EQ.1) WRITE (6,2100) NG
ISUB=17
IRANGE=4
WRITE (6,2400) ISTOP,ISUB,NPAR(ISUB),IRANGE
GO TO 50
C
45 NCUN=NMEON(MODEL)
C
CHECK ON COMPATIBILITY BETWEEN ELEMENTS OF NPAR
C
C
C
C
1. COMPATIBILITY OF INDNL AND IDEATH
50 ISUB=3
IF (INDNL.GT.0) GO TO 55
IF (IDEATH.EQ.0) GO TO 52
ISTOP=ISTOP+1
IF (ISTOP.EQ.1) WRITE (6,2100) NG
ISUD=4
WRITE (6,2500) ISTOP,ISUB,NPAR(ISUB),ISUD,NPAR(ISUD)
C
C
C
2. COMPATIBILITY OF INDNL AND ITYPE
52 IF (ITYPT.NE.2) GO TO 54
ISTOP=ISTOP+1
```

ORIGINAL PAGE IS
OF POOR QUALITY

TRUSS

DATE = 81229 09/18/44

```

IF (ISTOP.EQ.1) WRITE (6,2100) NG
ISUD=5
WRITE (6,2500) ISTOP,ISUB,NPAR(ISUB),ISUD,NPAR(ISUD)
C
C
C
3. COMPATIBILITY OF INONL AND MODEL
54 IF (MODEL.EQ.LINEL) GO TO 55
ISTOP=ISTOP+1
IF (.ISTOP.EQ.1) WRITE (6,2100) NG
ISUD=15
WRITE (6,2500) ISTOP,ISUB,NPAR(ISUB),ISUD,NPAR(ISUD)
C
C
C
4. COMPATIBILITY OF MXNUDS AND ITYPT
55 ISUB=5
ISUD=7
IF (LXNUDS.NE.1) GO TO 60
IF (ITYPT.EQ.1) GO TO 60
ISTOP=ISTOP+1
IF (ISTOP.EQ.1) WRITE (6,2100) NG
WRITE (6,2500) ISTOP,ISUB,NPAR(ISUB),ISUD,NPAR(ISUD)
C
C
C
C
CHECK FOR TEMPERATURE TAPE
60 ITEMPR=0
IF (MODEL.EQ.ITEL) ITEMPR=1
IF (MODEL.EQ.IEPCI .OR. MODEL.EQ.IEPCK) ITEMPR=2
IF (ITEMPR.EQ.0) GO TO 70
IF (ITP96.GT.0) GO TO 70
ISTOP=ISTOP+1
IF (ISTOP.EQ.1) WRITE (6,2100) NG
WRITE (6,2600) ISTOP
C
C
70 IITEMP=ITEMPR
IF (ISTOP.EQ.0) GO TO 75
WRITE (6,2700) ISTOP
INPUT=5
BACKSPACE INPUT
HEAD (5,1000) DATA
WRITE (6,2800) (I,I=1,3),DATA
IDATWR=1
C
75 IF (IDATWR.GT.1) GO TO 90
C
C
C
PRINT OUT NPAR VECTOR
WRITE (6,2900) NPAR,NUME,INONL,IDEATH
WRITE (6,2920) ITYPT,MXNUDS,NINT
WRITE (6,2940) MODEL,NUMMAT,NCUN
C
90 IF (ISTOP.EQ.0) GO TO 95
WRITE (6,2750)
STOP
C
C
C
C*** DATA PURTHULE ***** (START)

```


ORIGINAL PAGE IS
OF POOR QUALITY

TRUSS

DATE = 81229

09/18/44

M2=N6

```

C 120 CALL RUSS (A(N1),A(M2),A(M3),A(M4),A(N5),A(N113),A(N102),A(N103),
1         A(N105),A(N106),A(N107),A(N108),A(N109),A(N110),
2         A(N111),A(N112),A(N113),A(N114),A(N115),NCON,NDUE,NDM,
3         IDNA,A(N6A+ITWO),A(N6B+ITWO),MXNDS)
    IF (IND.GT.0) GO TO 150
    DO 140 I=1,20
140 IA(NFIRST + I - 1)=NPAR(I)
150 CONTINUE

```

```

C RETURN

```

```

C 1000 FORMAT (20A4)

```

```

C 2000 FORMAT (//49H LENGTH OF ARRAY NEEDED FOR STORING ELEMENT GROUP/
3         12H DATA (GROUP,13,26H) . . . . .
4         15H( MIDEST ) . . . =.15)
2030 FORMAT (//49H AVAILABILITY OF CORE STORAGE FOR ELEMENT GROUP/
1         49H DATA . . . . . ( NLAST ) .)

```

```

C 2100 FORMAT (1H1.45HERRJR IN ELEMENT GROUP CONTROL CARDS (TRUSS) /
1         16H ELEMENT GROUP =, 15/)
2200 FORMAT (15,7H. NPAR(.12,27H) IS OUT OF RANGE ... NPAR(.12,
1         3H) =.15)
2300 FORMAT (15,7H. NPAR(.12,16H) SHOULD BE .LE.. 12.10H ... NPAR(.12,
1         3H) =.15)
2400 FORMAT (15,7H. NPAR(.12,16H) SHOULD BE .GE.. 12.10H ... NPAR(.12,
1         3H) =.15)
2500 FORMAT (15,7H. NPAR(.12,3H) =.12.10H AND NPAR(.12,3H) =.12.
1         19H ARE NOT COMPATIBLE )
2600 FORMAT (15,37H. TEMPERATURE TAPE SHOULD BE PROVIDED )
2700 FORMAT (//25H TOTAL NUMBER OF ERRORS =.15//
1         48H CARD IMAGE LISTING AND PRINT-OUT OF NPAR VECTOR/
2         48H (WITH DEFAULTS ENFORCED) ARE GIVEN BELOW -----)
2800 FORMAT (//34H CARD IMAGE LISTING OF NPAR VECTOR //29X.8(11,9X)/
1         15H COLUMN NUMBERS,5X.8(10H1234567890)//
2         15H NPAR VECTOR .5X.20A4 // )
2750 FORMAT (//// 23H STOP (ERRORS IN NPAR) )

```

```

C 2900 FORMAT (36H ELEMENT DEFINITION ///.
1         14H ELEMENT TYPE ,13(2H .),16H( NPAR(1) ) . . . =.15/.
2         25H EQ.1. TRUSS ELEMENTS/ .-
3         25H EQ.2. 2-DIM ELEMENTS/ .-
4         25H EQ.3. 3-DIM ELEMENTS/ .-
5         25H EQ.4. BEAM ELEMENTS// .-
6         20H NUMBER OF ELEMENTS..10(2H .),16H( NPAR(2) ) . . . =.15//.
7         28H TYPE OF NONLINEAR ANALYSIS..6(2H .),15H( NPAR(3) ) . .
8         1H=.15/.
9         38H EQ.0. LINEAR / .
A         38H EQ.1. MATERIALLY NONLINEAR ONLY / .
1        45H EQ.2. UPDATED LAGRANGIAN FORMULATION // .
2        32H ELEMENT BIRTH AND DEATH OPTIONS ,4(2H .) .-
3        16H( NPAR(4) ) . . . =.15/.
4        28H EQ.0. OPTION NOT ACTIVE/ .
5        30H EQ.1. BIRTH OPTION ACTIVE / .
6        30H EQ.2. DEATH OPTION ACTIVE // )
2920 FORMAT (18H ELEMENT TYPE CODE,11(2H .),16H( NPAR(5) ) . . . =.15/.

```

ORIGINAL PAGE IS
OF POOR QUALITY

TRUSS

DATE = 81229

09/18/44

```

1      40H      EQ.0. GENERAL 3-D TRUSS
2      40H      EQ.1. RING ELEMENT
3      42H      MAXIMUM NUMBER OF NODES USED TO DESCRIBE 1.4X.
4      16H      ANY ONE ELEMENT,10(2H .).16H( NPAR(7) ). . =.15//.
5      35H      INTEGRATION ORDER FOR STIFFNESS 1.4X.
6      12H      CALCULATION,12(2H .).16H( NPAR(10) ). . =.15//.
7      2940. FORMAT (42H MATERIAL DEFINITION.
1      16H      MATERIAL MODEL,12(2H .).16H( NPAR(15) ). . =.15//.
2      40H      EQ.1. LINEAR ELASTIC
3      40H      EQ.2. NONLINEAR ELASTIC
4      42H      (STRESS-STRAIN LAW SPECIFIED)
5      40H      EQ.3. THERMOELASTIC
6      44H      EQ.4. ELASTIC-PLASTIC (ISOTROPIC)
7      44H      EQ.5. ELASTIC-PLASTIC (KINEMATIC)
8      48H      EQ.6. ELASTIC-PLASTIC-CREEP (ISOTROPIC)
9      48H      EQ.7. ELASTIC-PLASTIC-CREEP (KINEMATIC)
10     48H      EQ.8. USER SUPPLIED MODEL
11     37H      NUMBER OF DIFFERENT SETS OF MATERIAL
12     14H      CONSTANTS,13(2H .).16H( NPAR(16) ). . =.15//.
13     40H      NUMBER OF MATERIAL CONSTANTS PER SET.
14     16H( NPAR(17) ). . =.15)

```

C

END

ORIGINAL PAGE IS
OF POOR QUALITY

A3-89

FORTRAN IV RELEASE 2.0

TRUSS

DATE = 81229

OPTIONS IN EFFECT NOTERM, IO, EBCDIC, SOURCE, NOLIST, NODECK, LOAD, NO MAP
OPTIONS IN EFFECT NAME = TRUSS - LINECNT = 60
STATISTICS SOURCE STATEMENTS = 211, PROGRAM SIZE = 8304
STATISTICS NO DIAGNOSTICS GENERATED

ORIGINAL PAGE IS
OF POOR QUALITY

WRITE

DATE = 01245

10/23/01

```

IF (ISTAT.EQ.0) GO TO 40
IF (ILPE.EQ.3) GO TO 20
READ (8) VEL
READ (8) ACC
GO TO 40

C
20 READ (8) DISPE
IF (MODEX.EQ.2) GO TO 40
ISV=(IVC + JVC + 1)/2
ISA=(IAC + JAC + 1)/2
IF (ISV.EQ.0) READ (8)
IF (ISV.NE.0) READ (8) VEL
IF (ISA.NE.0) READ (8) ACC

C
40 IF (IPRIC.EQ.0) RETURN
WRITE (6,2100)

C
C
C
50 IC=4
IF (KKK.EQ.1) GO TO 60
IF (IDC.EQ.0) GO TO 180
60 WRITE (6,2000)
IC=IC + 5
DO 150 IB=1,NRB
NODE1=IPNODE(1,IB)
IF (NODE1.EQ.0) GO TO 150
NODE2=IPNODE(2,IB)
NODINC=IPNODE(3,IB)
IF (KKK.EQ.1) NODE1=1
IF (KKK.EQ.1) NODE2=NUMNP
IF (KKK.EQ.1) NODINC=1

C
DO 100 I1=NODE1,NODE2,NODINC
IC=IC + 1
IF (IC.LT.56) GO TO 105
WRITE (6,2045)
IC=4
105 DO 110 I=1,6
110 D(I)=0.
IL=0
DO 120 I=1,NDOF
KK=LD(I,11)
115 IL=IL + 1
IF (IL.LE.6) GO TO 117
WRITE (6,3000)
STOP
117 IF (IDOF(IL).EQ.1) GO TO 115
IF (KK.NE.0) D(IL)=DISP(KK)
IF (IOPE.NE.3.OR.IND.NE.0) GO TO 120
IF (KK.NE.0) D(IL)=DISPE(KK)
120 CONTINUE
DO 1000 IX=1,MB
IF (I1.EQ.NR(IX).OR.I1.EQ.NS(IX)) WRITE (20,5000) I1,D(2),D(3)
1000 CONTINUE
5000 FORMAL (110,2013.5)
100 WRITE (6,2010) I1,D
C

```


ORIGINAL PAGE IS
OF POOR QUALITY

*RITE..

DATE = 81245

18/23/01

```

IF (KKK.EQ.1) GO TO 180.
IF (IC.GE.55) GO TO 150
IC=IC+1
WRITE(6,2050)
150 CONTINUE
180 IF (ISTAT.EQ.0) RETURN

```

C
C
C

PRINT VELOCITIES.

```

IF (KKK.EQ.1 .AND. IOPE.NE.3) GO TO 201
IF (IVC.EQ.0) GO TO 280
201 IC=IC + 5 + 10C
IF (IDC.NE.0) WRITE(6,2050)
IF (IC.GE.54) GO TO 205
WRITE(6,2020)
GO TO 206
205 WRITE(6,2022)
IC=4
206 DO 250 IB=1,NPB
NODE1=IPNODE(1,IB)
IF (NODE1.EQ.0) GO TO 250
NODE2=IPNODE(2,IB)
NODINC=IPNODE(3,IB)
IF (KKK.EQ.1) NODE1=1
IF (KKK.EQ.1) NODE2=NUMNP
IF (KKK.EQ.1) NODINC=1
C
DO 200 II=NODE1,NODE2,NODINC
IC=IC + 1
IF (IC.LT.50) GO TO 207
WRITE(6,2022)
IC=4
207 DO 210 I=1,6
210 D(I)=0.
IL=0
DO 220 I=1,NDOF
KK=ID(1,II)
215 IL=IL + 1
IF (IL.LE.6) GO TO 217
WRITE (6,3000)
STOP
217 IF (IDOF(IL).EQ.1) GO TO 215
220 IF (KK.NE.0) D(IL)=VEL(KK)
DO 1001 IX=1,MB
IF (II.EQ.NR(IX).OR.II.EQ.NS(IX)) WRITE(20,5000) II,D(2),D(3)
1001 CONTINUE
200 WRITE(6,2010) II,D

```

C

```

IF (KKK.EQ.1) GO TO 280
IF (IC.GE.55) GO TO 250
IC=IC+1.
WRITE(6,2050)
250 CONTINUE

```

C
C
C

PRINT ACCELERATIONS.

```

280 IF (KKK.EQ.1 .AND. IOPE.NE.3) GO TO 290
IF (IAC.EQ.0) RETURN

```

ORIGINAL PAGE IS
OF POOR QUALITY

A3-93

WRITE

DATE = 81245

19723/01

```
IF (IDC.EQ.0.AND. IVC.EQ.0) GO TO 305
290 IC=IC + 6
IF (IC.GE.54) GO TO 303
WRITE(6,2050)
*RITE(6,2030)
GO TO 308
303 *RITE(6,2032)
IC=4
GO TO 308
305 IC=IC + 5
WRITE(6,2030)
308 DO 350 IB=1,NPB
NODE1=IPNODE(1,IB)
IF (NODE1.EQ.0) GO TO 350
NODE2=IPNODE(2,IB)
NODINC=IPNODE(3,IB)
IF (KKK.EQ.1) NODE1=1
IF (KKK.EQ.1) NODE2=NUMNP
IF (KKK.EQ.1) NODINC=1
C
DO 300 II=NODE1,NODE2,NODINC
IC=IC + 1
IF (IC.LT.56) GO TO 307
WRITE(6,2032)
IC=4
307 DO 310 I=1,6
310 D(I)=0.
IL=0
DO 320 I=1,NDUF
AK=ID(I,II)
315 IL=IL + 1
IF (IL.LE.6) GO TO 317
WRITE (6,3000)
STOP
317 IF (IDOF(IL).EQ.1) GO TO 315
320 IF (KK.NE.0) D(IL)=ACC(AK)
DO 1002 IX=1,M6
IF (II.EQ.NR(IX).OR. II.EQ.NS(IX)) *RITE(20,5000) II,D(2),D(3)
1002 CONTINUE
300 WRITE(6,2010) II,D
C
IF (KKK.EQ.1) RETURN
IF (IC.GE.55) GO TO 350
IC=IC+1
*RITE(6,2050)
350 CONTINUE
C
RETURN
C
2000 FORMAT (/27H D I S P L A C E M E N T S // 7H NODE 12X
114HX=DISPLACEMENT 4X 14HY=DISPLACEMENT 4X 14HZ=DISPLACEMENT
28X 10HX=ROTATION 8X 10HY=ROTATION 8X 10HZ=ROTATION //)
2010 FORMAT (2X,15,8X,6E15.6)
2020 FORMAT (/22H V E L O C I T I E S // 7H NODE 16X 10HX=VELOCITY
16X 10HY=VELOCITY 8X 10HZ=VELOCITY
28X 10HX=ROTATION 8X 10HY=ROTATION 8X 10HZ=ROTATION //)
2022 FORMAT (/11,21H V E L O C I T I E S //7H NODE 16X 10HX=VELOCITY
16X 10HY=VELOCITY 8X 10HZ=VELOCITY 8X 10HX=ROTATION 8X
```

ORIGINAL PAGE IS
OF POOR QUALITY

WRITE

DATE = 01245

18/23/01

310HY=ROTATION 8X 10HZ=ROTATION // 7H NODE 12X
 2030 FORMAT (/27H A C C E L E R A T I O N S // 7H NODE 12X
 114HX=ACCELERATION 4X 14HY=ACCELERATION 4X 14HZ=ACCELERATION
 28X 10HX=ROTATION 8X 10HY=ROTATION 8X 10HZ=ROTATION // 7H NODE 12X
 2032 FORMAT (/1H1.26H A C C E L E R A T I O N S // 7H NODE 12X
 114HX=ACCELERATION 4X 14HY=ACCELERATION 4X 14HZ=ACCELERATION
 28X 10HX=ROTATION 8X 10HY=ROTATION 8X 10HZ=ROTATION // 7H NODE 12X
 2040 FORMAT (/1H1.26H D I S P L A C E M E N T S // 7H NODE 12X
 114HX=DISPLACEMENT 4X 14HY=DISPLACEMENT 4X 14HZ=DISPLACEMENT
 28X 10HX=ROTATION 8X 10HY=ROTATION 8X 10HZ=ROTATION // 7H NODE 12X
 2050 FORMAT (/1H1.38H I N I T I A L C O N D I T I O N S // 7H NODE 12X
 2100 FORMAT (/1H1.38H I N I T I A L C O N D I T I O N S // 7H NODE 12X
 3000 FORMAT (/27H **STOP, ERROR IN DEGREE OF FREEDOM CALCULATIONS // 7H NODE 12X
 1 28H CHECK MASTER CONTROL CARD 1 // 1X)

END

ORIGINAL PAGE IS
OF POOR QUALITY

A3-95

WRITE

DATE = 81248

OPTIONS IN EFFECT NOTERM, ID, EBCDIC, SOURCE, NDLIST, NDOECK, LOAD, NUMAP, NOTEST
OPTIONS IN EFFECT NAME = WRITE, LINECNT = 63
STATISTICS SOURCE STATEMENTS = 193, PROGRAM SIZE = 5976
STATISTICS NO DIAGNOSTICS GENERATED

STATISTICS NO DIAGNOSTICS THIS STEP

APPENDIX 4: TWO-DIMENSIONAL PLOT CODE

During the solution phase, the main program "ADINA" solves for the responses of the bearing structural problem incrementally in time. These responses, namely the displacements, velocities, accelerations, and bearing forces, are stored on an out-of-core tape for future plotting purposes. The subroutine "WRITE" of "ADINA" has been modified to store on a tape the responses of the rotors and stators of squeeze-film dampers of either a single bearing or multi-bearing problems.

The code listed below reads and plots the rotor and stator displacement, velocity, and/or acceleration trajectories of any squeeze-film damper. For any desired plot, separate graphs for either the stator or the rotor responses can be obtained. For the displacement, the difference between the rotor and stator displacements are plotted to show the rotor orbit relative to the stator. For the bearing forces, both the actual force and the filtered force plots can be obtained. A linear filtering, by averaging the forces, has been used here. However, other methods of filtering can be employed by the user.

To help increase the diversity and capabilities of the code, different control flags or parameters have been introduced to give the user the opportunity to turn on or shut off any graph. For example, KPD is a flag that indicates displacement plotting mode. A value of "1" for KPD will request the code to plot displacements of both the rotor and stator.

The following control parameters must be input by the user:

KPD: Control parameter indicating plotting mode;

EQ.1: plot rotor and stator displacement trajectories

EQ.0: plots are not required.

KPV: Control parameter indicating velocity plotting mode;

EQ.1: plot rotor and stator velocity trajectories

EQ.0: plots are not required

KPA: Control parameter indicating acceleration plotting mode;

EQ.1: plot rotor and stator acceleration trajectories

EQ.0: plots are not required.

ORIGINAL PRINTING IS
OF POOR QUALITY

A4-3

```
1 //CASE1 JOB XXXX, *3523 ZE10 * .MSGLEVEL=(2,0)
2 ***JOBPAR* SKIP=YES, FORMS=1PWB
9 // EXEC FORT
//SYSIN DD * GENERATED STATEMENT
10 // EXEC GOFORT, PLOT=.GOSIZE=900K
31 //SYSIN DD * GENERATED STATEMENT
32 //FT23F001 DD DSN=USER.PROBN1, DISP=(OLD,KEEP)
33 //SYSIN DD * GENERATED STATEMENT
//
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
IEF142I CASE1 FORT = STEP WAS EXECUTED = COND CODE 0000
IEF373I STEP /FORT / START 81232.2330
IEF374I STEP /FORT / STOP 81232.2331 CPU 0MIN.05.55SEC SRB
IEF142I CASE1 GO = STEP WAS EXECUTED = COND CODE 0000
IEF373I STEP /GO / START 81232.2331..
IEF374I STEP /GO / STOP 81232.2334 CPU 0MIN 53.21SEC SRB
IEF375I JOB /CASE1. / START 81232.2330
IEF376I JOB /CASE1 / STOP 81232.2334 CPU 0MIN 53.76SEC SRB
```


MAIN-

DATE = 81232

```

TIM(1)=TIME
F1(I+(IP-1)*N)=F1P(IP)
F2(I+(IP-1)*N)=F2P(IP)
9 FR(I+(IP-1)*N)=FRP(IP)
C
C
C READ ROTOR AND STATOR DISPLACEMENTS INTO CORE
2 DO 501 J=1,M
READ (1R,5) II,D(2),D(3)
WRITE (6,5) II,D(2),D(3)
IF (KPD.NE.1) GO TO 501
DO 401 L=1,K
IF (II.EQ.NR(L)) DXR(I+(L-1)*N)=D(2)
IF (II.EQ.NR(L)) DYR(I+(L-1)*N)=D(3)
IF (II.EQ.NS(L)) DXS(I+(L-1)*N)=D(2)
IF (II.EQ.NS(L)) DYS(I+(L-1)*N)=D(3)
401 CONTINUE
501 CONTINUE
C
C
C READ ROTOR AND STATOR VELOCITIES INTO CORE
DO 502 J=1,M
READ (1R,5) II,D(2),D(3)
WRITE (6,5) II,D(2),D(3)
IF (KPV.NE.1) GO TO 502
DO 402 L=1,K
IF (II.EQ.NR(L)) VXR(I+(L-1)*N)=D(2)
IF (II.EQ.NR(L)) VYR(I+(L-1)*N)=D(3)
IF (II.EQ.NS(L)) VXS(I+(L-1)*N)=D(2)
IF (II.EQ.NS(L)) VYS(I+(L-1)*N)=D(3)
402 CONTINUE
502 CONTINUE
C
C
C READ ROTOR AND STATOR ACCELERATIONS INTO CORE.
DO 503 J=1,M
READ (1R,5) II,D(2),D(3)
WRITE (6,5) II,D(2),D(3)
IF (KPA.NE.1) GO TO 503
DO 403 L=1,K
IF (II.EQ.NR(L)) AXR(I+(L-1)*N)=D(2)
IF (II.EQ.NR(L)) AYR(I+(L-1)*N)=D(3)
IF (II.EQ.NS(L)) AXS(I+(L-1)*N)=D(2)
IF (II.EQ.NS(L)) AYS(I+(L-1)*N)=D(3)
403 CONTINUE
503 CONTINUE
500 CONTINUE
504 CONTINUE
DO 510 L=1,K
IF (KPD.NE.1) GO TO 507
WRITE (6,10) (DXR(I+(L-1)*N),I=1,N)
WRITE (6,11) (DYR(I+(L-1)*N),I=1,N)
WRITE (6,12) (DXS(I+(L-1)*N),I=1,N)
WRITE (6,13) (DYS(I+(L-1)*N),I=1,N)
507 IF (KPV.NE.1) GO TO 506
WRITE (6,14) (VXR(I+(L-1)*N),I=1,N)
WRITE (6,15) (VYR(I+(L-1)*N),I=1,N)
WRITE (6,16) (VXS(I+(L-1)*N),I=1,N)

```

ORIGINAL PAGE IS
OF POOR QUALITY

A4-8

MAIN. ---

DATE = 81232

```
WRITE (6,17) (VYS(I+(L-1)*N),I=1,N)
506 IF (KPA.NE.1) GO TO 510
WRITE (6,13) (AXR(I+(L-1)*N),I=1,N)
WRITE (6,19) (AYR(I+(L-1)*N),I=1,N)
WRITE (6,20) (AXS(I+(L-1)*N),I=1,N)
WRITE (6,21) (AYS(I+(L-1)*N),I=1,N)
510 CONTINUE
CALL PLOTS
CALL PLOT (1,0,1,5,-3)
DO 610 L=1,K
NPTS(1)=N
NPTS(2)=N
INC(1)=1
INC(2)=1
LINTYP(1)=35
LINTYP(2)=35
INTEQ(1)=4
INTEQ(2)=0
IF (KPD.NE.1) GO TO 22
C
C
C
PLOT ROTOR DISPLACEMENT TRAJECTORIES
DO 550 I=1,N
XARRAY(I)=DXR(I+(L-1)*N)
550 YARRAY(I)=DYR(I+(L-1)*N)
IF (L.NE.1) CALL PLOT(12,0,0,0,-3)
CALL GRAPH (XARRAY,'Y=DISP',6,7.5,YARRAY,'Z=DISP',6,7.5,
1 'ROTOR DISPLACEMENT TRAJECTORIES',31,0.5,7.75,0.14,1,
2 NPTS,INC,LINTYP,INTEQ)
C
C
C
PLOT STATOR DISPLACEMENT TRAJECTORIES
DO 551 I=1,N
XARRAY(I)=DXS(I+(L-1)*N)
551 YARRAY(I)=DYS(I+(L-1)*N)
CALL PLOT (12,0,0,0,-3)
CALL GRAPH (XARRAY,'Y=DISP',6,7.5,YARRAY,'Z=DISP',6,7.5,
1 'STATOR DISPLACEMENT TRAJECTORIES',32,0.5,7.75,0.14,1,
2 NPTS,INC,LINTYP,INTEQ)
C
C
C
PLOT ROTOR ORBIT RELATIVE TO STATOR
DO 561 I=1,N
XARRAY(I)=DXR(I+(L-1)*N)-DXS(I+(L-1)*N)
561 YARRAY(I)=DYR(I+(L-1)*N)-DYS(I+(L-1)*N)
DTH=3.141593/90.0
DO 552 I=1,180
I2=I+N+2
XARRAY(I2)=8C*COS(DTH*(I-1))
552 YARRAY(I2)=8C*SIN(DTH*(I-1))
XARRAY(181+N+2)=XARRAY(N+3)
YARRAY(181+N+2)=YARRAY(N+3)
NPTS(2)=181
LINTYP(1)=0
LINTYP(2)=0
CALL PLOT (12,0,0,0,-3)
CALL GRAPH (XARRAY,'Y=DISP',6,7.5,YARRAY,'Z=DISP',6,7.5,
1 'ROTOR ORBIT',11,0.5,7.75,0.14,2,
```

ORIGINAL PRICE LIST
OF POOR QUALITY

MAIN

DATE = 91232

2 NPTS, INC, LINTYP, INTEQ)
22 IF (KPV.NE.1) GO TO 23

PLOT ROTOR VELOCITY TRAJECTORIES

DO 580 I=1,N
XARRAY(I)=VXR(I+(L-1)*N)
580 YARRAY(I)=VYR(I+(L-1)*N)
NPTS(2)=N
LINTYP(1)=35
LINTYP(2)=35
CALL PLOT (12.0,0.0,0,-3)
CALL GRAPH (XARRAY, Y=VEL, 6.7.5, YARRAY, Z=VEL, 6.7.5,
1 ROTOR VELOCITY TRAJECTORIES, 27.0.5.7.75, 0.14.2,
2 NPTS, INC, LINTYP, INTEQ)

PLOT STATOR VELOCITY TRAJECTORIES

DO 579 I=1,N
XARRAY(I)=VXS(I+(L-1)*N)
579 YARRAY(I)=VYS(I+(L-1)*N)
CALL PLOT (12.0,0.0,0,-3)
CALL GRAPH (XARRAY, Y=VEL, 6.7.5, YARRAY, Z=VEL, 6.7.5,
1 STATOR VELOCITY TRAJECTORIES, 28.0.5.7.75, 0.14.2,
2 NPTS, INC, LINTYP, INTEQ)
23 IF (KPA.NE.1) GO TO 24

PLOT ROTOR ACCELERATION TRAJECTORIES

DO 600 I=1,N
XARRAY(I)=AXR(I+(L-1)*N)
600 YARRAY(I)=AYR(I+(L-1)*N)
CALL PLOT (12.0,0.0,0,-3)
CALL GRAPH (XARRAY, Y=ACC, 6.7.5, YARRAY, Z=ACC, 6.7.5,
1 ROTOR ACCELERATION TRAJECTORIES, 31.0.5.7.75, 0.14.2,
2 NPTS, INC, LINTYP, INTEQ)

PLOT STATOR ACCELERATION TRAJECTORIES

DO 590 I=1,N
XARRAY(I)=AXS(I+(L-1)*N)
590 YARRAY(I)=AYS(I+(L-1)*N)
CALL PLOT (12.0,0.0,0,-3)
CALL GRAPH (XARRAY, Y=ACC, 6.7.5, YARRAY, Z=ACC, 6.7.5,
1 STATOR ACCELERATION TRAJECTORIES, 32.0.5.7.75, 0.14.2,
2 NPTS, INC, LINTYP, INTEQ)
24 IF (KPF.NE.1) GO TO 610

PLOT HORIZONTAL COMPONENT OF BEARING FORCE

DO 605 I=1,N
XARRAY(I)=TIM(I)
605 YARRAY(I)=F1(I+(L-1)*N)
CALL PLOT (12.0,0.0,0,-3)
CALL GRAPH (XARRAY, TIME, 4.7.5, YARRAY, ROTOR FORCE (HORIZONTAL),
1 23.7.5, ROTOR FORCE TRAJECTORIES, 24.0.5.7.75, 0.14.
2 1, NPTS, INC, LINTYP, INTEQ)

ORIGINAL PAGE IS
OF POOR QUALITY

MAIN

DATE = 81232

C
C
C

PLOT VERTICAL COMPONENT OF BEARING FORCE

```

DO 606 I=1,N
606 YARRAY(I)=F2(I+(L-1)*N)
CALL PLOT (12.0,0.0,-3)
CALL GRAPH (XARRAY,'TIME',4.7.5,YARRAY,'ROTOR FORCE(VERTICAL)',
1 21.7.5,'ROTOR FORCE TRAJECTORIES',24.0.5,7.75,0.14,
2 1,NPTS,INC,LINTYP,INTEQ)

```

C
C
C

PLOT FILTERED HORIZONTAL COMPONENT OF BEARING FORCE:
(LINEAR FILTERING: $F(I)=(F(I)+F(I+1))/2$)

```

N1=N-1
DO 581 I=1,N1
XARRAY(I)=(TIM(I)+TIM(I+1))/2.0
581 YARRAY(I)=(F1(I+(L-1)*N)+F1(I+1+(L-1)*N))/2.0
NPTS(1)=N1
CALL PLOT (12.0,0.0,-3)
CALL GRAPH (XARRAY,'TIME',4.7.5,YARRAY,'ROTOR FORCE(HORIZONTAL)',
1 23.7.5,'FILTERED FORCE TRAJECTORIES',27.0.5,7.75,0.14,
2 1,NPTS,INC,LINTYP,INTEQ)

```

C
C
C

PLOT FILTERED VERTICAL COMPONENT OF BEARING FORCE
(LINEAR FILTERING)

```

DO 582 I=1,N1
582 YARRAY(I)=(F2(I+(L-1)*N)+F2(I+1+(L-1)*N))/2.0
CALL PLOT (12.0,0.0,-3)
CALL GRAPH (XARRAY,'TIME',4.7.5,YARRAY,'ROTOR FORCE(VERTICAL)',
1 23.7.5,'FILTERED FORCE TRAJECTORIES',27.0.5,7.75,0.14,
2 1,NPTS,INC,LINTYP,INTEQ)

```

```

610 CONTINUE
CALL PLOT (12.0,0.0,999)
1 FORMAT (16I5)
5 FORMAT (110.2D13.6)
6 FORMAT (110.4D13.6)
7 FORMAT (50X,110,4(2X,D13.6))
10 FORMAT (///5X,'HORIZONTAL ROTOR DISPLACEMENT',//.8(2X,E13.5))
11 FORMAT (///5X,'VERTICAL ROTOR DISPLACEMENT',//.8(2X,E13.5))
12 FORMAT (///5X,'HORIZONTAL STATOR DISPLACEMENT',//.8(2X,E13.5))
13 FORMAT (///5X,'VERTICAL STATOR DISPLACEMENT',//.8(2X,E13.5))
14 FORMAT (///5X,'HORIZONTAL ROTOR VELOCITY:',//.8(2X,D13.5))
15 FORMAT (///5X,'VERTICAL ROTOR VELOCITY:',//.8(2X,D13.5))
16 FORMAT (///5X,'HORIZONTAL STATOR VELOCITY:',//.8(2X,D13.5))
17 FORMAT (///5X,'VERTICAL STATOR VELOCITY:',//.8(2X,D13.5))
18 FORMAT (///5X,'HORIZONTAL ROTOR ACCELERATION:',//.8(2X,D13.5))
19 FORMAT (///5X,'VERTICAL ROTOR ACCELERATION:',//.8(2X,D13.5))
20 FORMAT (///5X,'HORIZONTAL STATOR ACCELERATION:',//.8(2X,D13.5))
21 PORMAT (///5X,'VERTICAL STATOR ACCELERATION:',//.8(2X,D13.5))
505 FORMAT (1H0///5X,'PRINT OUT FOR ROTOR',15,
1 ' AND STATOR',15,/)
STOP
END

```

ORIGINAL PAGE IS
OF POOR QUALITY

MAIN

DATE = 81232

OPTIONS IN EFFECT NOTERM, IO, EBCDIC, SOURCE, NOLIST, NODECK, LOAD, NDMAP
OPTIONS IN EFFECT NAME = MAIN, LINECNT = 60
STATISTICS SOURCE STATEMENTS = 177, PROGRAM SIZE = 175892
STATISTICS NO DIAGNOSTICS GENERATED

C 2
A 3

APPENDIX 5: THREE-DIMENSIONAL PLOTTING CODE

This Appendix overviews the post-processor programs written to plot three-dimensional pictures of a deformed shaft given the deflected positions of distinct points along the shaft at a sequence of time intervals. The purpose of such plots is to reduce computed rotor vibration events (transient as well as steady-state) to an easily visualized series of "still" pictures. A sequence of such pictures is then suitable to construct a slow-motion animation of rotor vibration phenomena. This Appendix provides a detailed description of the necessary inputs to the post-processor programs as well as describing the post-processor programs themselves.

A5.1 Post-Processor 1, JCL Requirements

There are three JCL requirements for the first of the two post-processor programs. The first of these three requirements is the allocation of FORTRAN logical unit 20 in the modified ADINA program to provide input to the post-processor program. Execution of this post-processor would be simplified if the data that is to be read by this unit exists on the disk data storage of the computer. If this data does exist on a disk dataset, then the JCL necessary for unit 20 is that which will allow this unit to access the appropriate disk dataset.

The second JCL requirement of the first post-processor program deals with the datasets that are created during the execution of this program that facilitate the plotting of

the orbit trajectory. During the execution of this post-processor program, the time-displacement data for all the nodes along the shaft are read and then this data is written in a manner that a dataset is created for each node along the shaft. Each dataset that is created contains only the time-displacement data for one node. The units associated with this manner of data manipulation begin with FORTRAN logical unit 11 and continue through unit 11 plus the number of nodes along the shaft. Each of these units will require the JCL to assign them to different disk datasets that are created during the program. Per the above discussion, if there are 4 nodes present along the shaft, then the JCL supplied will have to define FORTRAN logical units 11, 12, 13, and 14.

The third JCL requirement of the first post-processor program concerns the dataset which contains the output from this segment of the plotting package. This output is the coefficients of the interpolatory polynomial. This data is output to a disk dataset so that it can be retrieved by the second post-processor program. The FORTRAN logical unit associated with this output is unit 21. Therefore, the JCL to create and save a dataset that will be allocated to unit 21 must be provided.

A5.2 Post-Processor 2, JCL Requirements

The JCL requirements for this post-processor have been alluded to in the previous section. These are twofold and both deal with datasets that are created by the first post-processor program.

First, beginning with FORTRAN logical unit 11 and assigning one unit for each node, the datasets that contain the orbit trajectory informa-

tion must be allocated to FORTRAN logical unit. This allocation procedure must be similar to the allocations that were made in executing the first post-processor. This means that the dataset allocated to unit 11 during execution of the first post-processor must be allocated to unit 11 during the execution of the second post-processor.

The second JCL requirement of this post-processor program is the allocation of the dataset (created during the execution of the first post-processor program) which contains the output from the first post-processor. This dataset must be allocated to FORTRAN logical unit 21 during the execution of the second post-processor program.

A5.3 Inputs from Modified ADINA Code

The output from the modified ADINA program serves as input to the first post-processor program contains four different types of records. These records and the order in which they should appear with their formats are as follows:

- (1) Number of nodes which appear along the shaft in a (2X, I10) format.
- (2) X, Y, and Z static, undeformed position of these nodes in a (3(2X, E15.8) format. Note, these nodes should be ordered so that the node closest to the origin appears first and the node farthest away from the origin along the shaft axis appears last.
- (3) The period of shaft revolution in seconds per cycle in a (2X, E15.7) format.
- (4) The node id, time, X, Y, and Z displacement for each node at each time step of integration in a (I10, 4E13.6) format.

The records defined by (1), (2), and (3) will appear once. However, the records defined by item (4) will appear for each node for every time step during the solution of the analysis.

A5.4 User Inputs

A5.4A User Inputs to Post-Processor 1

A5.4A.1 Card No. 1

This card defines the times at which the deformed shaft is to be plotted.

1.	10	11.	20	21	30	
ANGINT		ANGFIN		ANGINC		(3F10.0)

ANGINT - Value of the initial revolution at which the deformed shaft is to be plotted

ANGFIN - Value of the final revolution at which the deformed shaft is to be plotted

ANGINC - Value of the revolution increment at which the deformed shaft is to be plotted. The time of the i^{th} plot will be:

$$\text{TIME} = (\text{ANGINT} + (i-1)*\text{ANGINC}) * \text{PERIOD}$$

Where: PERIOD = period of shaft revolution.

A5.4A.2 Card No. 2

This card defines the direction of the shaft axis.

1		
1		(1A1)

Where: I is an alphanumeric X, Y, or Z depending upon the shaft orientation.

Note: For X1, X2, X3 coordinate systems;

$$X = X1$$

$$Y = X2$$

$$Z = X3$$

A5.4B User Inputs to Post-Processor 2

A5.4B.1 Card No. 1

1.	5	6	15
NSECT	THETA		

(I5,F10.0)

Where: NSECT - The number of sections into which the shaft will be divided for the plotting of its deformed shape. A high value for this number will cause a large CPU time for the plotting procedure. However, a low value for this number will cause the plot to be choppy in appearance.

THETA - The angle at which the shaft axis will be inclined from the horizontal.

A5.4B.2 Card No. 2

This card reads the plot control parameters. The control parameters consist of 6 integers read in a (I5, 5I1) format. The first integer controls the number of deformed shafts that are drawn on each plot. The second integer, read from the sixth card column, controls the usage of the default plot window size. If this parameter is input

as a zero or blank, then the plot drawn will fit in an eleven inch square. If this parameter is input as a one, then the plot size will be read from input card number 3. The third control integer, read from the seventh card column, controls the maximum size of the orbit trajectory. If this parameter is input as a zero or blank, then the maximum orbit trajectory will be scaled to one and one-half inches on the plot. If this parameter is input as a one, then the plot size of the maximum orbit trajectory will be read from input card number 4. The fourth control integer, read from the eighth card column, controls the appearance of the node names on the plot. If this parameter is input as a zero or a blank, then the node names will appear on the plot. If this parameter is input with a value of one, then the node names do not appear. The fifth control parameter, read from the ninth card column, controls the appearance of the line connecting a node's static, undeformed, position with its deformed position. If this parameter is input as zero or blank, then this line appears. If this parameter is input as a one, then this line does not appear on the plot. The final control parameter, read from card column number ten, controls the time range over which the orbit trajectory is plotted. If this parameter is input as zero or blank, then the entire orbit trajectory is plotted. If this parameter is input as a one, then the time range over which the orbit is plotted is defined by the data appearing on card number 5.

Summary:

Parameter	Controls
1	Number of shafts per picture
2	Plot size
3	Orbit trajectory size
4	Node name appearance
5	Static-to-Deformed line appearance
6	Orbit trajectory time range

A5.4B.3 Card No. 3

1	10	
WINDOW		(F10.0)

Window - The size of the shaft appearing on the plot.

A5.4B.4 Card No. 4

This card is used only if the third control parameter on card number

2 is non-zero.

1	10	
DISPSZ		(F10.0)

DISPSZ - The size of the maximum orbit trajectory.

A5.4B.5 Card No. 5

This card is used only if the sixth control parameter on card number

2 is non-zero.

1	10	11	20	
PERMIN		PERMAX		(2F10.0)

PERMIN - The minimum period at which the orbit trajectory
is to be plotted.

PERMAX - The maximum period at which the orbit trajectory
is to be plotted. ---

Note: $TIMIN = PERIOD * PERMIN$ _____

$TIMAX = PERIOD * PERMAX$

A5.5 Program Listing, Post-Processors

ORIGINAL PAGE IS
OF POOR QUALITY

LIST UTILITY

```
//NASAPR   JOB: 04180.NASA.CLASS=A.
// EXEC   PONT
C*****
C*
C*   TITLE : COMPUTER PLOTTING OF TRANSIENT AND STEADY-STATE
C*           ROTOR VIBRATION TIME VARYING DEFLECTIONS.
C*
C*   PURPOSE : THIS PROGRAM IS INTENDED TO SERVE AS THE FIRST OF TWO
C*             PLOTTING POST-PROCESSOR PROGRAMS. THE TASK OF THESE
C*             PROGRAMS IS TO VISUAL DEPICT THE DEFORMATION PROCESS
C*             OF A ROTATING SHAFT. THIS POST-PROCESSOR PROGRAM
C*             SERVES TO COMPUTE THE TIME VECTOR AT WHICH THE
C*             DEFORMED SHAFT WILL BE PLOTTED AS WELL AS THE
C*             GENERATION OF THE COEFFICIENTS FOR THE INTERPOLATORY
C*             POLYNOMIAL THAT WILL BE USED TO DRAW THE SHAFT AT
C*             EACH OF THE SELECTED TIME INTERVALS.
C*
C*   NOTE : FOR FURTHER DISCUSSION OF THIS PLOTTING PACKAGE,
C*           PLEASE SEE THE REPORT RELATING TO THESE
C*           POST-PROCESSOR PROGRAMS.
C*****
C
C   DIMENSION I(50), TIME(1000)
C   COMMON/NUMBER/NBRPLT, NUMNF
C   COMMON/PLTCNT/IPLT, TIMINC, PLTTIM(1000)
C   DATA IX .1Y .1Z /
C   +       1HX .1HY .1FZ /
C
C   INITIALIZE THE PLOT CONTROL COUNTER
C
C       IPLT=1
C
C   OBTAIN THE ADINA INFORMATION FROM THE DATA SET THAT WAS CREATED DURING
C   THE ADINA RUN.
C
C       CALL BDADIN(PERIOD)
C
C   NOW READ THE PLOT TIME CONTROL DATA AND GENERATE THE PLOT TIME
C   VECTOR. THIS VECTOR CONTAINS THE TIMES AT WHICH THE DEFORMED
C   SHAPE OF THE SHAFT IS TO BE VIEWED.
C
C       READ(5,1)ANGINT,ANGFIN,ANGINC
C       1 FORMAT(3F10.0)
C       NBRPLT=0
C       ANGLE=ANGINT
C       IF(ANGINC .GT. 0)GO TO 3
C       WRITE(6,2)
C       2 FORMAT(' **** ANGINC IS EITHER ZERO OR NEGATIVE. ****')
C       +       ' SORRY THIS IS A FATAL ERROR.')
C
C       STOP
C       3 CONTINUE
C       ANGLE=ANGLE+ANGINC
C       NBRPLT=NBRPLT+1
C       IF(ANGLE .LT. ANGFIN)GO TO 3
C       IF(NBRPLT .LE. 1000)GO TO 3
C       WRITE(6,4)
C       4 FORMAT(' **** NBRPLT EXCEEDS ITS MAXIMUM ALLOWABLE VALUE. ****')
```

ORIGINAL PRINTING
OF POOR QUALITY.

LIST UTILITY

```

**1/.
+.      SORRY FATAL ERROR.!)
STOP-
5 CONTINUE
  PLTTIM(1)=ANGINT*PERIOD
  TIMINC=ANGINC*PERIOD
  DO 6, I=2, NBRPLT
  PLTTIM(I)=PLTTIM(I-1)+TIMINC
6 CONTINUE
C
C IDENTIFY, BY USER INPUT, THE SHAFT DIRECTION.
C
  READ(5,10) ID
10 FORMAT(A1)
  ISHAFT=0
  IF(ID .EQ. IX) ISHAFT=1
  IF(ID .EQ. IY) ISHAFT=2
  IF(ID .EQ. IZ) ISHAFT=3
  IF(ISHAFT .NE. 0) GO TO 30
  WRITE(6,20) ID
20 FORMAT(*1 UNKNOWN ID INPUT, ID = ',A1)
  STOP 20
30 CONTINUE
C
C SUBROUTINE TRAJ WILL SET UP THE DATASETS FOR THE TRAJECTORY
C   PLOTTING.
C
  CALL TRAJ(II, AMAX, ISHAFT)
C
C WRITE OUT THE ADINA INFORMATION THAT WILL BE REQUIRED IN THE PLOTTING
C   ROUTINE.
C
  CALL WTADIN(II, ISHAFT, AMAX, PERIOD)
C
C DO THE FOLLOWING FOR EACH PLOTTING WINDOW.
C
  DO 50 I=1, NBRPLT.
C
C READ THE ADINA DISPLACEMENTS FOR THIS TIME WINDOW.
C
  CALL RDDISP(LTIME, TIME(I))
C
C AT THIS POINT IF LTIME IS EQUAL ZERO, THEN WE HAVE NO DATA TO PLOT.
C
  IF(LTIME .EQ. 0) GO TO 49
C
C WRITE THE TIME OF THIS WINDOW TO THE STORAGE DATA SET FOR RETRIEVAL
C   DURING THE EXECUTION OF THE PLOTTING ROUTINE.
C
  WRITE(21,40) TIME(I)
  WRITE(6,40) TIME(I)
40 FORMAT(2X,E13.6)
C
C COMPUTE AND OUTPUT THE COEFFICIENTS OF A FORWARD-DIFFERENCE NEWTON'S
C   INTERPOLATORY POLYNOMIAL FOR THIS TIME WINDOW.
C
  CALL COMDIF(ISHAFT)
49 CONTINUE

```

OF K...
 OF K...

LIST UTILITY

```

50. CONTINUE.
C
C HOPEFULLY, ALL DONE.
C
  STOP
  END
  SUBROUTINE_COMDIF(ISHAFT)
C
C THIS SUBROUTINE COMPUTES AND OUTPUTS THE COEFFICIENTS OF A NEWTON'S
C FORWARD-DIFFERENCE INTERPOLATORY POLYNOMIAL. THESE COEFFICIENTS
C ARE READ AND USED BY THE PLOTTING ROUTINE. THERE IS A SET OF
C COEFFICIENTS GENERATED FOR EACH TIME WINDOW TO BE PLOTTED.
C
  COMMON/DISPLA/DISP(3,50)
  COMMON/NODE /LOC(3,50)
  COMMON/NUMBER/NBRPLT,NUMNP
  REAL LCC
  DIMENSION DIF(50,50)
  DO 70 IDIR=1,3
  IF(IDIR .EQ. ISHAFT)GO TO .60
  N=NUMNP-1
  DO 10 I=1,N
  DIF(1,I)=(DISP(IDIR,I+1)-DISP(IDIR,I))/(LOC(I+1,ISHAFT)
  * = LOC(I,ISHAFT))
  10 CONTINUE
  DO 30 I=2,N
  K=NUMNP-I
  DO 20 L=1,K
  M=I+L
  DIF(I,L)=(DIF(I-1,L+1)-DIF(I-1,L))/(LOC(M,ISHAFT)
  * = LOC(L,ISHAFT))
  20 CONTINUE
  30 CONTINUE
  WRITE(21,40)IDIR
  WRITE(6,40)IDIR
  40 FORMAT(2X,I3)
  WRITE(21,50)DISP(IDIR,1),(DIF(K,1),K=1,N)
  WRITE(6,50)DISP(IDIR,1),(DIF(K,1),K=1,N)
  50 FORMAT((5(2X,E13.6)/))
  60 CONTINUE
  70 CONTINUE
  RETURN
  END
  SUBROUTINE_READIN(PERIOD)...
  COMMON/NODE /LOC(3,50)
  COMMON/NUMBER/NBRPLT,NUMNP
  REAL LOC
C
C READ THE NUMBER OF NODES
C
  READ(20,10)NUMNP
  WRITE(6,10)NUMNP
  10 FORMAT(2X,I10)
C
C READ THE X, Y, AND Z LOCATIONS OF THE NODES
C
  DO 30 I=1,NUMNP
  READ(20,20)(LOC(I,K),K=1,3)

```

OF POOR QUALITY

A5-13

LIST UTILITY

```

WRITE(6,20)(LDC(I,K),K=1,3)
20 FORMAT(3(2X,E15.8))
30 CONTINUE

```

C READ THE PERIOD OF SHAFT REVOLUTIONS

```

READ(20,40)PERIOD
WRITE(6,40)PERIOD
40 FORMAT(2X,E15.7)
RETURN
END
SUBROUTINE ADDISP(ITIME,TIME)
COMMON/DISPLA/DISP(3,50)
COMMON/NUMBER/NBRPLT,NUMNP
COMMON/PLTENT/IPLUT, TIMINC, PLTTIM(1000)
DIMENSION I(50)
ITIME=0
4 CONTINUE
READ(20,10,END=40)I(1),TIME,(DISP(K,1),K=1,3)
DO 5 LL=1,PLT,NBRPLT
IF(ABS(PLTTIM(LL)-TIME)...LE...TIMINC)GO TO 6
5 CONTINUE
DO 11 K=2,NUMNP
READ(20,10)IDUM
10 FORMAT(10,4E13.6)
11 CONTINUE
GO TO 4
6 CONTINUE
ITIME=1
IPLUT=IPLUT+1
DO 20 I=2,NUMNP
READ(20,10)I(1),TIME,(DISP(K,1),K=1,3)
20 CONTINUE
DO 30 I=1,NUMNP
WRITE(6,10)I(1),TIME,(DISP(K,1),K=1,3)
30 CONTINUE
GO TO 60
40 CONTINUE
WRITE(6,50)
50 FORMAT(' ***** END OF ADINA INFORMATION ***** ')
STOP
60 CONTINUE
RETURN
END
SUBROUTINE TRAJC(I1, AMAX, ISHAFT)
COMMON/NUMBER/NBRPLT,NUMNP
DIMENSION I(50),TRAJD(50,2),TRAJA(50,3)
REWIND 20

```

C READ UNWANTED DATA AT THE BEGINNING OF THE FILE

```

READ(20,1)IDUM
1 FORMAT(2X,110)
DO 3 I=1,NUMNP
READ(20,2)IDUM
2 FORMAT(2X,E15.8)
3 CONTINUE
READ(20,2)IDUM

```

ORIGINAL SOURCE
OF POWER QUALITY

LIST UTILITY

```

      AMAX=0.0
5    CONTINUE
      DO 20 I=1,NUMNP
      READ(20,10,END=40)II(I),TIME,(TRAJA(I,K),K=1,3)
10   FORMAT(110,4E13.6)
      IF(ISHAFT .NE. 1)GO TO 11
      TRAJD(I,1)=TRAJA(I,2)
      TRAJD(I,2)=TRAJA(I,3)
      GO TO 19
11   CONTINUE
      IF(ISHAFT .NE. 2)GO TO 12
      TRAJD(I,1)=TRAJA(I,1)
      TRAJD(I,2)=TRAJA(I,3)
      GO TO 19
12   CONTINUE
      TRAJD(I,1)=TRAJA(I,1)
      TRAJD(I,2)=TRAJA(I,2)
19   CONTINUE

C    COMPUTE THE MAGNITUDE OF THE TRAJECTOR PATH FOR THIS DISPLACEMENT
C    DATA.
C
      RANGE=SQRT(TRAJD(I,1)**2+TRAJD(I,2)**2)
C
C    CHECK THIS MAGNITUDE AGAINST THE MAXIMUM AND IF THIS IS LARGER
C    SET THE MAXIMUM EQUAL TO THIS MAGNITUDE.
C
      IF(RANGE .GT. AMAX)AMAX=RANGE
20   CONTINUE
      DO 30 I=1,NUMNP
      IOUT=10+I
      WRITE(IOUT,10)II(I),TIME,(TRAJD(I,K),K=1,2)
30   CONTINUE
      GO TO 5
40   CONTINUE
      REWIND 20

C    READ UNWANTED DATA AT THE BEGINNING OF THE FILE.
C
      READ(20,1)IDUM
      DO 50 I=1,NUMNP
      READ(20,2)DUM
50   CONTINUE
      READ(20,2)DUM
      RETURN
      END
      SUBROUTINE READIN(II, ISHAFT, AMAX, PERIOD)
      COMMON/NCDE /LOC(3,50)
      COMMON/NUMBER/NBRPLT,NUMNP
      DIMENSION II(50)
      REAL LOC
      *RITE(21,10)NUMNP,NBRPLT,ISHAFT,PERIOD
      WRITE(6,10)NUMNP,NBRPLT,ISHAFT,PERIOD
10   FORMAT(3(2X,110),E15.7)
      *RITE(21,11)II(I),I=1,NUMNP)
11   FORMAT((40(1X,14)/))
      *RITE(21,15)AMAX
      *RITE(6,15)AMAX

```

ORIGINAL PAGE IS
OF POOR QUALITY

LIST UTILITY.

```
15 FORMAT(2X,E15.8)
   WRITE(21,20)(LOC(I,ISHAFT),I=1,NUMNP)
   WRITE(6,20)(LOC(I,ISHAFT),I=1,NUMNP)
20 FORMAT((5(2X,E13.6)/))
   RETURN
   END
```

ORIGINAL COPY
OF POOR QUALITY

```

C*****
C*
C*
C*      TITLE:  COMPUTER PLOTTING OF TRANSIENT AND STEADY-STATE
C*              ROTOR VIBRATION TIME VARYING ANALYSIS.
C*
C* DESCRIPTION: THIS PROGRAM IS THE SECOND OF TWO PROGRAMS THAT
C*              ALLOW THE USER TO VIEW THE DEFORMATION OF A
C*              ROTATING SHAFT. THIS PROGRAM CANNOT BE
C*              EXECUTED UNTIL THE FIRST ROUTINE HAS BEEN RUN
C*              AND THE DATASETS THAT IT CREATES ARE ACCESSIBLE
C*              TO THIS PROGRAM. THIS PROGRAM PERFORMS THE
C*              THE ACTUAL GRAPHICS ON THE CALCUMP PLOTTER.
C*
C*
C*****
COMMON/COEF  /A(2,50) , LOC(50)
COMMON/CNTRL/CNTRL(10)
COMMON/DRAWX /X(2,101)
COMMON/NUM  /NUMNP , NBRPLT
COMMON/SCALE/FACTRA , FACTRB
COMMON/TRIG /COSINE , SINE
COMMON/TRJCT/PERIOD , TIMIN , TIMAX
DIMENSION ICR(2) , II(50) , AXIAL(101) , XI(50)
REAL LOC
DATA PI/3.1415926536/,DISPSZ/1.5/,WINDOW/8.0/
XMAX=0.0
XMIN=0.0

C
C READ THE FOLLOWING INFORMATION FROM THE DATASET CREATED BY THE FIRST
C POST-PROCESSOR:
C      1) NUMNP = NUMBER OF NODAL POINTS ON THE SHAFT.
C      2) NBRPLT = NUMBER OF POSSIBLE PLOT TIMES.
C      3) ISHAFT = ID OF THE SHAFT AXIAL COORDINATE.
C      4) PERIOD = PERIOD OF SHAFT REVOLUTION.
C
C      READ(21,10)NUMNP,NBRPLT,ISHAFT,PERIOD
C      10 FORMAT(3(2X,I10),E15.7)
C
C READ THE ID'S OF THE SHAFT NODES.
C
C      READ(21,20)(II(I),I=1,NUMNP)
C      20 FORMAT((4C(1X,I4)/))
C      DO 30 I=1,NUMNP
C      XI(I)=FLGAT(II(I))
C      30 CONTINUE
C
C READ THE MAXIMUM-MAGNITUDE OF THE TRAJECTORY.
C
C      READ(21,40)AMAX
C      40 FORMAT(2X,E15.8)
C

```

CONTROL VECTOR
DEFORM QUALITY

LIST-UTILITY...

```

C READ THE AXIAL LOCATION OF THE NODES.
C
  READ(21,50)(LOC(I),I=1,NUMNP)
  50 FORMAT((5(2X,E13.6)Z))
C
C SEARCH THE SHAFT FOR ITS MAXIMUM AND MINIMUM LOCATION.
C
  DO 60 I=1,NUMNP
  IF(LOC(I) .LT. XMIN)XMIN=LOC(I)
  IF(LOC(I) .GT. XMAX)XMAX=LOC(I)
  60 CONTINUE
C
C READ THE NUMBER OF SECTIONS INTO WHICH THE SHAFT IS TO BE DIVIDED.
C
  READ(5,70)NSECT,THETA
  70 FORMAT(15,F10.0)
C
C READ THE VALUES OF THE CONTROL VECTOR. THE VALUES READ WILL CONTROL
C THE FOLLOWING ITEMS:
C
C ICNTRL
C COMPONENT          CONTROLS
C 1                   THE NUMBER OF DEFORMED SHAFTS THAT ARE PLOTTED IN
C                     EACH WINDOW.
C 2                   THE USAGE OF THE DEFAULT PLOTTING WINDOW SIZE. IF
C                     THE VALUE READ IS ZERO, THEN THE DEFAULT SIZE
C                     8 INCHES IS USED. IF THE VALUE READ IS ONE,
C                     THEN THIS WINDOW SIZE MUST BE INPUT.
C 3                   THE USAGE OF THE DEFAULT AMPLIFICATION FACTOR.
C                     IF THE VALUE READ IS ZERO, THEN THE AMPLI-
C                     FICATION FACTOR USED WILL PRODUCE A MAXIMUM
C                     TRAJECTORY ORBIT OF 1.5 INCHES. IF THE VALUE
C                     READ IS ONE, THEN THE SIZE OF THE MAXIMUM
C 4                   THE APPEARANCE OF THE NODE NAMES ON THE PLOT.
C                     IF THE VALUE READ IS ZERO, THEN THE NAMES
C                     APPEAR. IF THE VALUE READ IS ONE THEN THE
C                     DO NOT APPEAR.
C 5                   THE APPEARANCE OF THE LINE CONNECTING THE STATIC
C                     POSITIONS OF THE NODES TO THEIR DEFORMED
C                     LOCATIONS. IF THE VALUE READ IS ZERO, THEN THIS
C                     LINE APPEARS. IF THE VALUE READ IS ONE, THEN
C                     THIS LINE DOES NOT APPEAR.
C 6                   THE TIME RANGE OF THE TRAJECTORY ORBIT THAT IS
C                     OUTPUT IN EACH PLOTTING WINDOW. IF THE VALUE
C                     READ IS ZERO, THEN THE ENTIRE ORBIT IS DISPLAYED
C                     IN EACH PLOTTING WINDOW. IF THE VALUE READ IS
C                     ONE, THEN ONLY THE RANGE SPECIFIED BY THE USER
C                     IS PLOTTED.
C
  READ(5,80)(ICNTRL(I),I=1,10)
  80 FORMAT(15,9I1)
  WRITE(6,83)(ICNTRL(I),I=1,10)
  83 FORMAT(' ICNTRL = ' // 10(13,2X)////)
C
C DEFALT WILL USE THE VALUES OF THE CONTROL VECTOR TO DETERMINE IF ANY
C ADDITIONAL INPUTS MUST BE READ AND IT WILL THEN READ THESE INPUTS.
C
  CALL DEFALT(DLSPSZ, WINDOW)

```


ORIGINAL PAGE IS
OF POOR QUALITY

LIST UTILITY. —

```

C COMPUTE THE INCREMENTAL AXIAL DISTANCE. —
C   DDIST=(XMAX-XMIN)/FLOAT(NSECT)
C
C CONVERT THETA TO RADIAN FROM DEGREES.
C   THETA=THEIA*PI/180.
C
C TO AVOID MULTIPLE CALLS TO THE TRIG FUNCTIONS, COMPUTE THE SINE
C AND COSINE OF THEIA. —
C   COSINE=CCS(THETA)
C   SINE=SIN(THETA)
C
C FIND THE MAXIMUM X1 RANGE. —
C   XRNGE=ABS(XMAX*CCSINE-XMIN*COSINE)
C
C FIND THE MAXIMUM X2 RANGE. —
C   YRNGE=ABS(XMAX*SINE-XMIN*SINE)
C
C DETERMINE THE MAXIMUM RANGE OF THE PROBLEM.
C   IF(XRNGE .GE. YRNGE)RANGE=XRNGE
C   IF(YRNGE .GT. XRNGE)RANGE=YRNGE
C
C WITH THE MAXIMUM RANGE AND THE SIZE OF THE VIEWING WINDOW, COMPUTE
C THE WINDOW SCALING FACTOR.
C   FACTRB=WINDCW/RANGE.
C
C WITH THE MAXIMUM MAGNITUDE OF THE TRAJECTORY, COMPUTE THE DISPLACEMENT
C SCALING FACTOR.
C   FACTRA=DISPSZ/AMAX
C
C INITIALIZE THE PLOTTER.
C   CALL PLOTS
C
C MOVE AND CENTER THE ORIGIN OF THE PLOTTER
C   CENTER=0.5*(11.0-WINDOW)
C   CALL PLOT(0.0, CENTER, -3)
C
C RESET THE NUMBER OF SECTIONS.
C   NSECT=NSECT+1
C   JJ=ICNTRL(1)
C   IF(JJ .EQ. 0)JJ=1
C
C DO THE FOLLOWING FOR ALL THE POSSIBLE PLOT TIMES.
C   DO 130 LL=1,NBRPLT
C     DO 120 KK=1,JJ
C
C READ THE PLOT TIME.

```

OF FOUR

LIST UTILITY

```

C      READ(21,50,END=140)TIME.
C      DO THE FOLLOWING FOR THE DEFLECTION DIRECTIONS.
C      DO 90 I=1,2.
C      READ THE DIRECTION ID.
C      READ(21,85)IDIR(I)
C      85 FORMAT(2X,I3).
C      READ THE FORWARD NEWTON INTERPOLATORY POLYNOMIAL COEFFICIENTS.
C      READ(21,50)(A(I,K),K=1,NUMNP)
C      90 CONTINUE
C      NOW FOR EACH LOCATION ALONG THE SHAFT.
C      DO 100 I=1,NSECT
C      DETERMINE THE AXIAL LOCATION OF THIS POINT.
C      DIST=XMIN+(DDIST*(I-1))
C      CALL DISP TO EVALUATE THE FORWARD NEWTON INTERPOLATORY POLYNOMIAL AT
C      THIS AXIAL LOCATION.
C      CALL DISP(DIST, X1, X2)
C      STORE THESE COORDINATES IN THE COORDINATE MATRIX.
C      AXIAL(I)=DIST
C      X(1,I)=X1
C      X(2,I)=X2
C      100 CONTINUE
C      DO 110 I=1,NSECT
C      COMPUTE THE SCALED DISPLACEMENTS.
C      X(1,I)=X(1,I)*FACTRA
C      X(2,I)=X(2,I)*FACTRA
C      COMPUTE THE SCALED LOCATIONS.
C      AXIAL(I)=AXIAL(I)*FACTRB
C      NOW PLACE THE SCALED DISPLACEMENTS INTO THE PROPER GLOBAL
C      VIEWING PERSPECTIVE.
C      CALL COORD(AXIAL(I), X(1,I), X(2,I))
C      110 CONTINUE
C      DRAW THE UNDEFORMED, STATIC SHAFT.
C      CALL STATIC
C      CALL DRAW TO PLOT OUT THE DEFLECTION SHAPE OF THE SHAFT AT THIS TIME.

```

ORIGINAL PAGE 13
OF POOR QUALITY

LIST UTILITY_____

```

C
C   CALL DRAW(NSECT)
C
C   CONNECT THE UNDEFORMED NODAL LOCATIONS WITH THEIR DEFORMED LOCATIONS.
C
C   IF(ICNTRL(5) .EQ. 0)CALL LINE
C
C   OUTPUT TO THE PRINTER THE PLOT TIME JUST PLOTTED.
C
C   WRITE(6,115)TIME
115 FORMAT('   PLOT DRAWN FOR TIME = ',E15.8)
C
C   IF MULTIPLE SHAFTS ARE TO APPEAR ON THE SAME PICTURE, THEN THERE IS
C   NO NEED TO REDRAW THE REMAINING COMPONENTS EACH TIME.
C
C   IE(KK .NE. 1)GO TO 120
C
C   PLOT OUT THE SHAFT NODAL TRAJECTORIES.
C
C   CALL TRJPLT(II)
C
C   WRITE OUT THE NAMES OF THE NODES.
C
C   IF(ICNTRL(4) .EQ. 0)CALL NAME(CENTER, XI)
C
C   PRINT OUT THE TIME ASSOCIATED WITH THIS PLOT.
C
C   CALL SYMBCL(3.0, 1.0, 0.25, 7*TIME = , 0.0, 7)
C   CALL NUMBER(999., 999., 0.25, TIME, 0.0, 4)
120 CONTINUE
C
C   MOVE THE PLOT ORIGIN FOR THE NEXT WINDOW IN TIME.
C
C   CALL PLOT(17.0, 0.0, =3)
130 CONTINUE
140 CONTINUE
C
C   HOPEFULLY, ALL DONE.
C
C   CALL PLOT(20.0, 0.0, 999)
STOP
END
SUBROUTINE COORD(DIST, X1, X2)
COMMON/TRIG, ZCOSINE, SINE
X1=X1 + DIST*ZCOSINE
X2=X2 + DIST*SINE
RETURN
END
SUBROUTINE DEFALT(DISPSZ, WINDOW)
COMMON/CNTRC/ICNTRL(10)
COMMON/TRJCT/PERIOD, TIMIN, , TIMAX
IF(ICNTRL(2) .EQ. 0)GO TO 20
READ(5,10)WINDOW
10 FORMAT(F10.0)
20 CONTINUE
IF(ICNTRL(3) .EQ. 0)GO TO 30
READ(5,10)DISPSZ
30 CONTINUE

```

ORIGINAL PAGE IS
OF POOR QUALITY

LIST UTILITY

```

IF(ICNTRL(6) .EQ. 0)GO TO 50
READ(5,40)PERMIN,PERMAX
40 FORMAT(2F10.0)
TIMIN=PERMIN*PERIOD
TIMAX=PERMAX*PERIOD
GO TO 60
50 CONTINUE
TIMIN=0.0
TIMAX=1.E+10
60 CONTINUE
WRITE(6,70)TIMIN,TIMAX
70 FORMAT('      TIMIN = ',E15.7/
+      '      TIMAX = ',E15.7)
RETURN
END
SUBROUTINE DISP(DIST, X1, X2)
COMMON/COEF /A(2,50) , LOC(50)
COMMON/NUM /NUMNP , NERPLT
DIMENSION A1(2,50)
REAL LOC

```

C THIS SUBROUTINE WILL EVALUATE THE NEWTON FORWARD INTERPOLATORY
C POLYNOMIAL THAT WILL RESULT IN THE APPROXIMATION OF THE SHAFT
C DISPLACEMENT AT THE DISTANCE DIST.
C THE DISPLACEMENT IS DEFINED BY DETERMINING THE TWO COMPONENTS OF THE
C SHAFT DISPLACEMENT.

```

N=NUMNP-1
DO 20 I=1,2
DO 10 K=1,NUMNP
A1(I,K)=A(I,K)
10 CONTINUE
20 CONTINUE
DO 40 I=1,2
DO 30 K=1,N
L=NUMNP-K
A1(I,L)=A1(I,L)+(DIST-LOC(L))*A1(I,L+1)
30 CONTINUE
40 CONTINUE
X1=A1(1,1)
X2=A1(2,1)
RETURN
END
SUBROUTINE DRAW(NSECT)
COMMON/COEF /A(2,50) , LOC(50)
COMMON/DRAWX /X(2,101)
COMMON/NUM /NUMNP , NERPLT
COMMON/SCALE1/FACTRA , FACTRB
REAL LOC
DO 15 J=1,2
X1=X(1,1)
X2=X(2,1)-(J-1)*0.015
CALL PLOT(X1, X2, 3)
DO 10 I=2,NSECT
X1=X(1,I)
X2=X(2,I)-(J-1)*0.015
CALL PLOT(X1, X2, 2)
10 CONTINUE

```

LIST UTILITY

```

15 CONTINUE.
   DO 20 I=1,NUMNP
   AXIAL=LOC(1)
   CALL DISP(AXIAL, X1, X2)
   X1=X1*FACTRA
   X2=X2*FACTRA
   AXIAL=AXIAL*FACTRB
   CALL CUCRD(AXIAL, X1, X2)
   CALL SYMBCL(X1, X2, 0.1, 11, 0.0, -1)
   CALL SYMBCL(X1, X2, 0.1, 0, 0.0, -1)
20 CONTINUE
   RETURN
   END
   SUBROUTINE LINE
   COMMON/COEF /A(2,50), LOC(50)
   COMMON/NUM /NUMNP, NBRPLT
   COMMON/SCALE1/FACTRA, FACTRB
   COMMON/TRIG /COSINE, SINE
   REAL LOC
   DO 10 I=1,NUMNP
   X1=LOC(I)*COSINE*FACTRB
   X2=LOC(I)*SINE*FACTRB
   CALL PLOT(X1, X2, 3)
   CALL DISP(LOC(I), X1, X2)
   X1=X1*FACTRA
   X2=X2*FACTRA
   POS=LOC(I)*FACTRB
   CALL CUCRD(POS, X1, X2)
   CALL PLOT(X1, X2, 2)
10 CONTINUE
   RETURN
   END
   SUBROUTINE NAME(CENTER, X1)
   COMMON/COEF /A(2,50), LOC(50)
   COMMON/NUM /NUMNP, NBRPLT
   COMMON/SCALE1/FACTRA, FACTRB
   COMMON/TRIG /COSINE, SINE
   REAL LOC
   DIMENSION X1(50)
   DO 10 I=1,NUMNP
   X1=LOC(I)*FACTRB*COSINE
   X2=LOC(I)*FACTRB*SINE
   X1=X1+0.6*CENTER
   X2=X2+0.8*CENTER
   CALL PLOT(X1, X2, 2)
   X=X1(I)
   CALL NUMBER(X1, X2, 0.20, X, 0.0, -1)
10 CONTINUE
   RETURN
   END
   SUBROUTINE STATIC
   COMMON/COEF /A(2,50), LOC(50)
   COMMON/NUM /NUMNP, NBRPLT
   COMMON/SCALE1/FACTRA, FACTRB
   COMMON/TRIG /COSINE, SINE
   REAL LOC
   RANGE=LOC(NUMNP)*LOC(1)

```

ORIGINAL PAGE IS
OF POOR QUALITY

LIST UTILITY

```

DDIST=RANGE/42.0
DIST=LDC(1)
DO 10 I=1,22.2
X1=(DIST*(1-I)*DDIST)*COSINE*FACTRA
X2=(DIST*(1-I)*DDIST)*SINE*FACTRB
CALL PLOT(X1, X2, 3)
X1=(DIST+I*DDIST)*COSINE*FACTRA
X2=(DIST+I*DDIST)*SINE*FACTRB
CALL PLOT(X1, X2, 2)
10 CONTINUE
RETURN
END
SUBROUTINE TRUPLT
COMMON/COEF /A12,50) , LCC(50)
COMMON/NUM /NUMNP , NUMRPLT
COMMON/SCALE1/FACTRA , FACTRB
COMMON/TRIG /COSINE , SINE
COMMON/TRJCT/PERIOD , TIMIN ,TIMAX
REAL LCC
HGT=0.05
HGT2=0.100
ISYM=50
DO 30 I=1,NUMNP
ISYM1=1
IN=10+1
1 CONTINUE
READ(IN,5)I1,TIME,TRAJ1,TRAJ2
5 FORMAT(110,3E13.8)
IF(TIME .LT. TIMIN)GO TO 1
X1=TRAJ1*FACTRA
X2=TRAJ2*FACTRA
POS=LCC(I)*FACTRB
CALL COORD(POS, X1, X2)
IF(X1 .GT. 1000. .OR. X2 .GT. 1000.)STOP 9900
CALL PLOT(X1, X2, 3)
CALL SYMBOL(X1, X2, HGT, -1, 0.0, -1)
10 CONTINUE
READ(IN,5,END=20)I1,TIME,TRAJ1,TRAJ2
IF(TIME .GT. TIMAX)GO TO 20
ISYM1=ISYM1+1
X1=TRAJ1*FACTRA
X2=TRAJ2*FACTRA
POS=LCC(I)*FACTRB
CALL COORD(POS, X1, X2)
IF(X1 .GT. 1000. .OR. X2 .GT. 1000.)STOP 9900
CALL PLOT(X1, X2, 2)
IF(ISYM1 .NE. ISYM)GO TO 10
ISYM1=1
CALL SYMBOL(X1, X2, HGT, 1, 0.0, -1)
GO TO 10
20 CONTINUE
READ(IN) IN
30 CONTINUE
RETURN
END

```

750 RECORDS PRINTED. END OF LIST UTILITY

A5.6 Sample Run of the 3-D Plotting Post-Processors

A5.6A ADINA Generated Data

The necessary data from the modified version of ADINA was obtained through Fortran logical unit 20. The JCL used to allocate this unit was

```
//FT20E001 DD DSN=USER.MULTI1,DISP=(NEW,CATLG),  
//          SPACE=(TRK,(150,20),RLSE),  
//          UNIT=SYSDA,VOL=SER=ACAD01,  
//          DCB=(RECFM=FB,BLKSIZE=12960,LRECL=80)
```

A5.6B Post-Processor / JCL Requirements

The JCL required by the first post-processor program for this sample problem that contained three nodes along the shaft was:

```
// EXEC GOFORT  
//FT11F001 DD DSN=USER.EFHPLT1,DISP=(NEW,CATLG),UNIT=SYSDA,  
//          VOL=SER=ACAD01,SPACE=(TRK,(50,5),RLSE),  
//          DCB=(RECFM=FB,BLKSIZE=6160,LRECL=80)  
//FT12F001 DD DSN=USER.EFHPLT2,DISP=(NEW,CATLG),UNIT=SYSDA,  
//          VOL=SER=ACAD01,SPACE=(TRK,(50,5),RLSE),  
//          DCB=(RECFM=FB,BLKSIZE=6160,LRECL=80)  
//FT13F001 DD DSN=USER.EFHPLT3,DISP=(NEW,CATLG),UNIT=SYSDA,  
//          VOL=SER=ACAD01,SPACE=(TRK,(50,5),RLSE),  
//          DCB=(RECFM=FB,BLKSIZE=6160,LRECL=80)  
//FT20F001 DD DSN=USER.MULTI1,DISP=SHR  
//FT21F001 DD DSN=USER.EFHPLT,DISP=(NEW,CATLG),UNIT=SYSDA,  
//          VOL=SER=ACAD01,SPACE=(TRK,(50,5),RLSE),  
//          DCB=(RECFM=FB,BLKSIZE=6160,LRECL=80)
```

A5.6C Post-Processor / Input

The user inputs to the first post-processor program for this sample run was:

```
0.0    90.0    0.3  
X
```

A5.6D Post-Processor 2, JCL Requirements

The JCL required by the second post-processor program for this sample
was:

```
// EXEC GOFORT,GOSIZE=512K,PLOT=  
//FT11F001 DD DSN=USER.EFHPLT1,DISP=SHR  
//FT12F001 DD DSN=USER.EFHPLT2,DISP=SHR  
//FT13F001 DD DSN=USER.EFHPLT3,DISP=SHR  
//FT21F001 DD DSN=USER.EFHPLT,DISP=SHR
```

A5.6E Post-Processor 2 Input

The user inputs for the second post-processor program for this sample
run was:

```
      20      35  
      3
```


A5.6F Post-Processor 2 Output

Attached are some selected plots that were obtained during this -
sample run.

ORIGINAL PAGE IS
OF POOR QUALITY

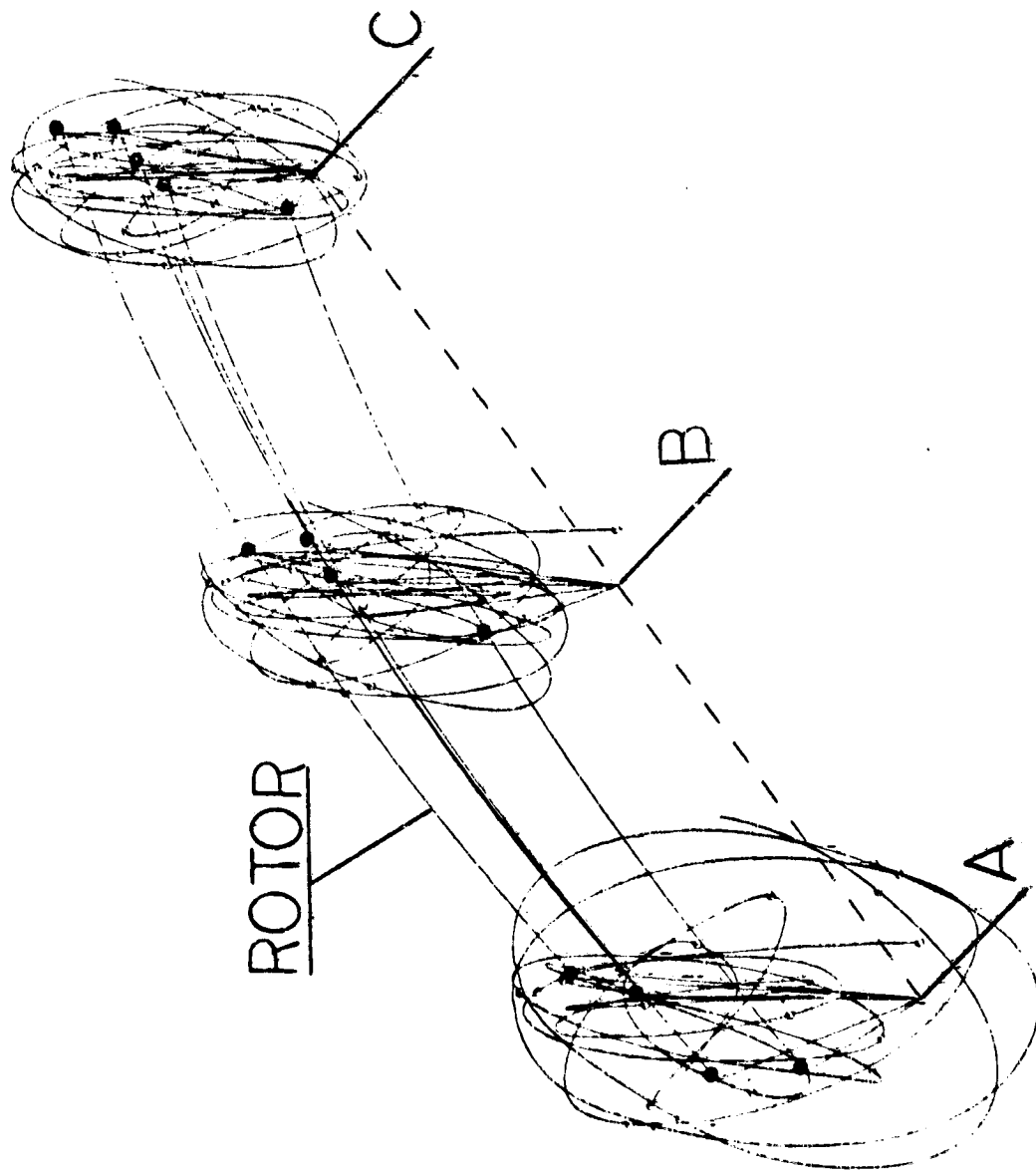


FIG. A5-5 ROTOR TRAJECTORY

ORIGINAL PAGE IS
OF POOR QUALITY

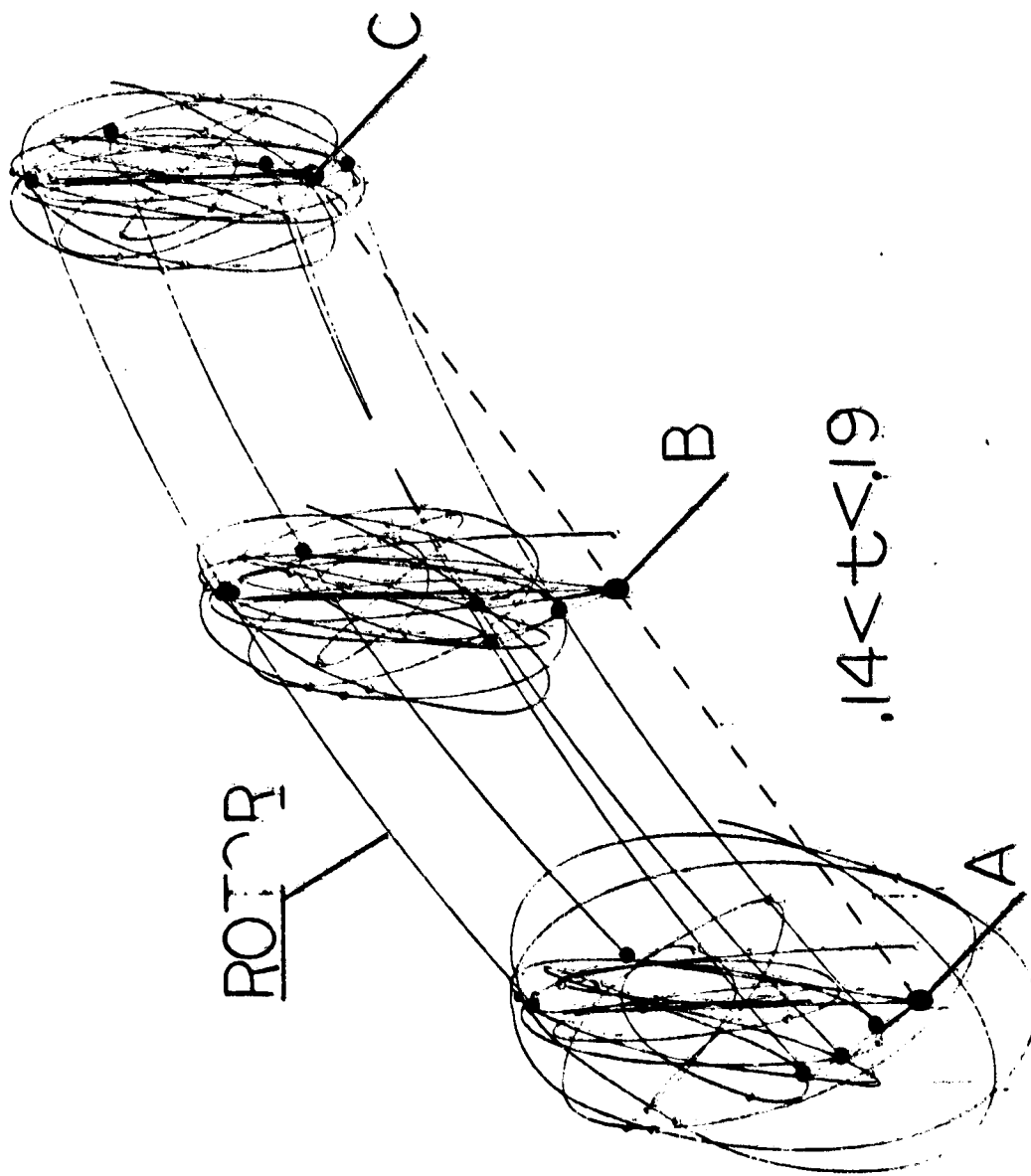


FIG. A5-4 ROTOR TRAJECTORY

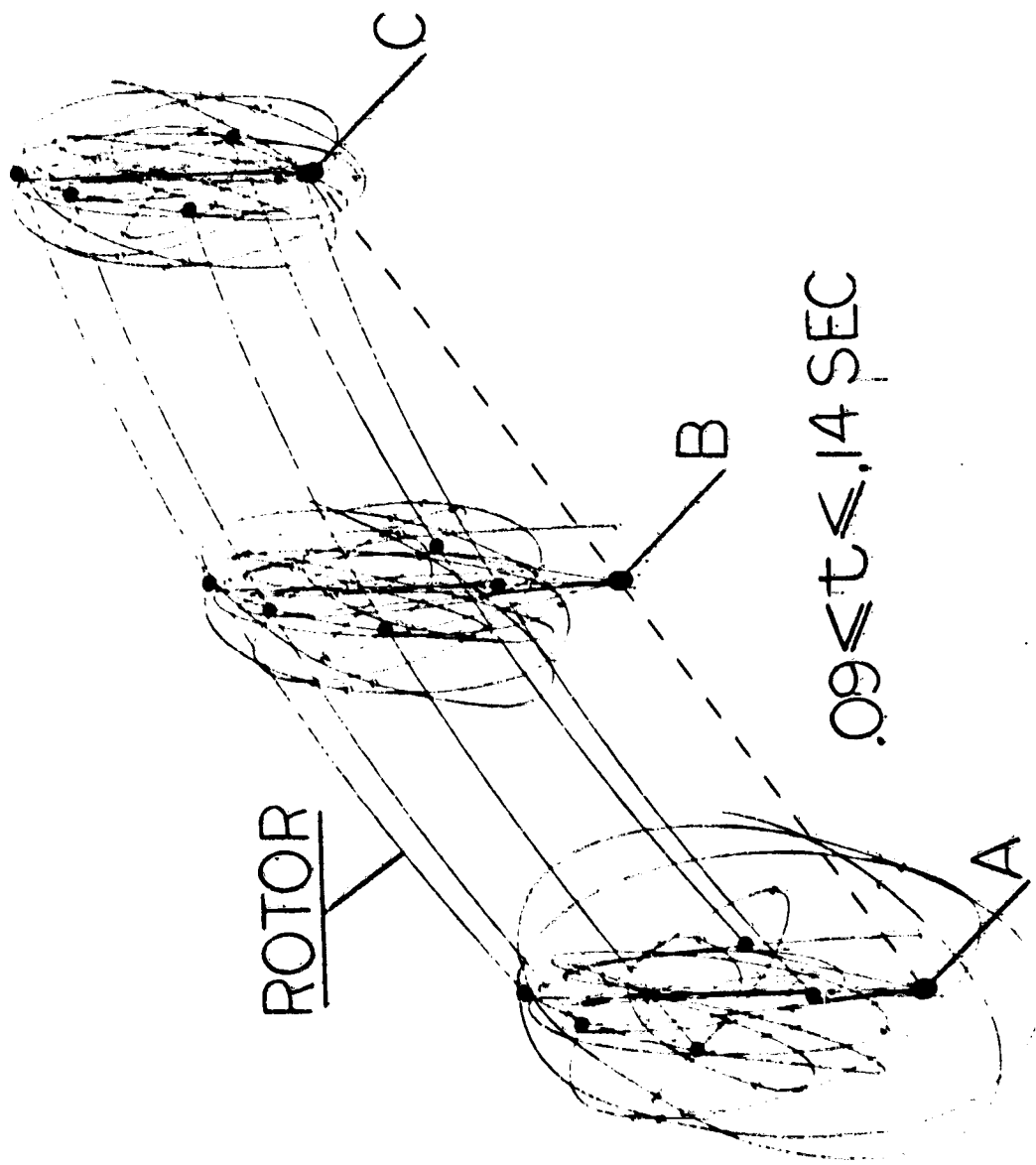
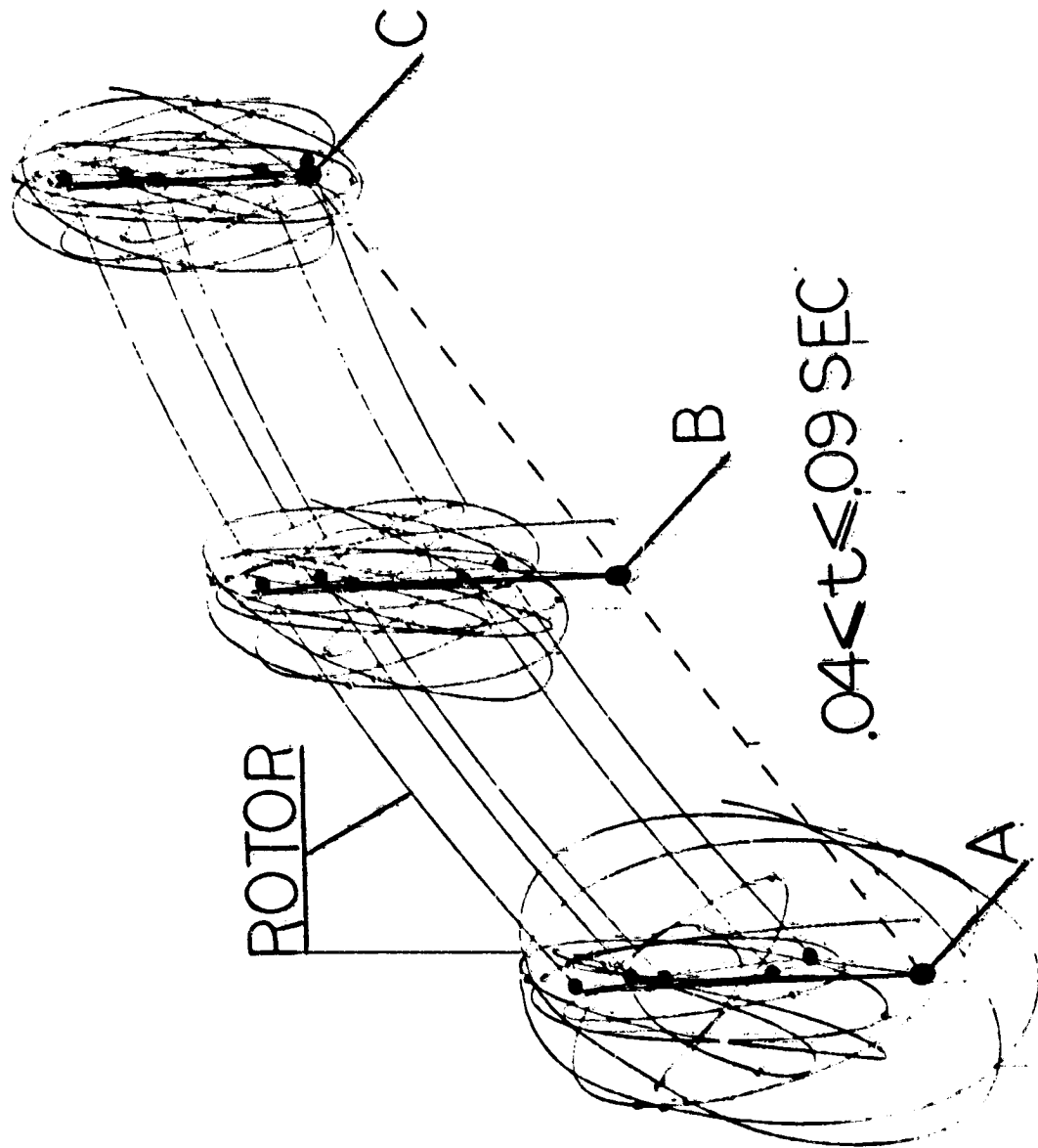


FIG. A5-3 ROTOR TRAJECTORY

ORIGINAL PAGE IS
OF POOR QUALITY



FIGA5-2 ROTOR TRAJECTORY

ORIGINAL PAGE IS
OF POOR QUALITY

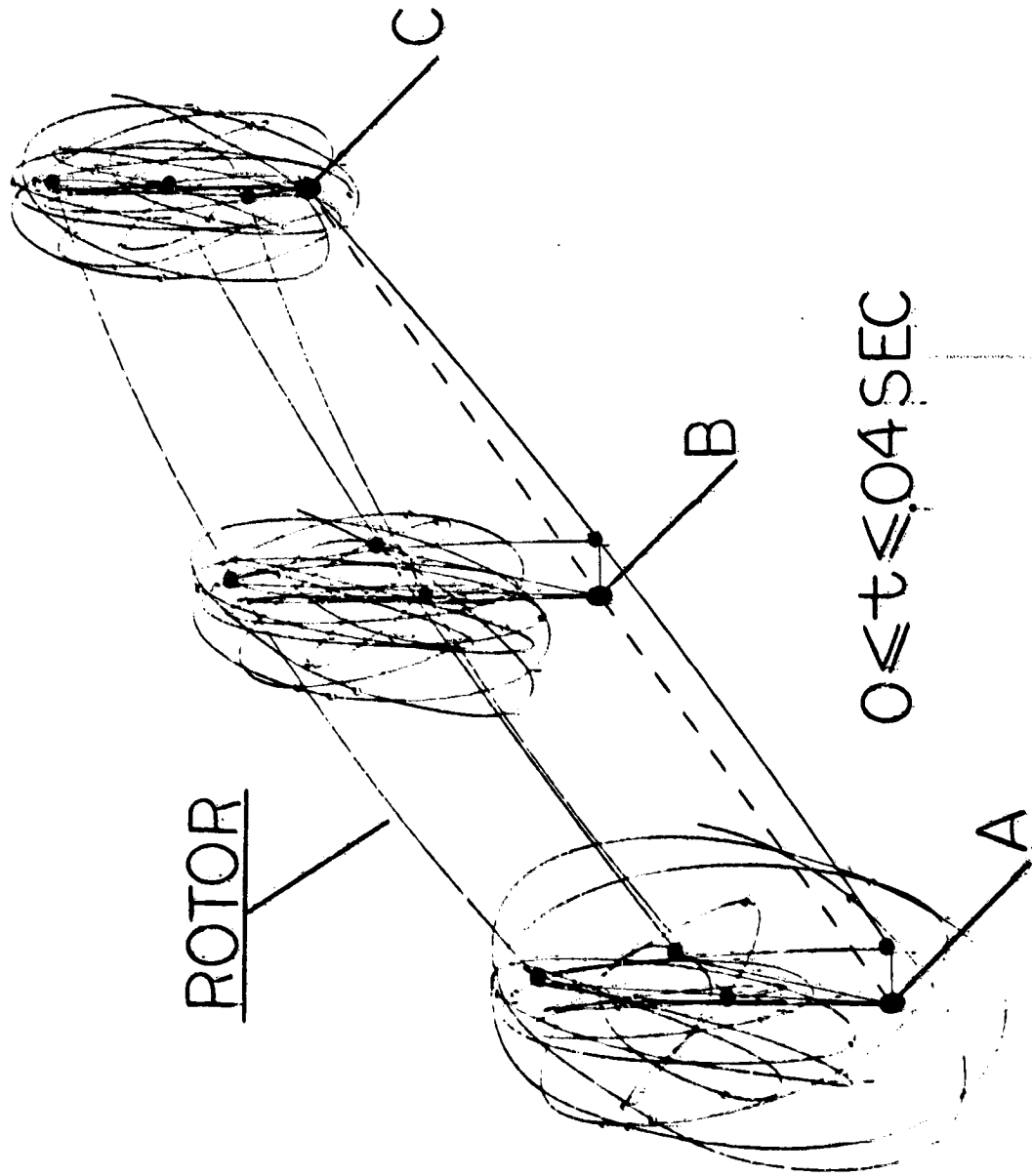


FIG. A5-1 ROTOR TRAJECTORY

APPENDIX 6: NOMENCLATURE

- a_j - Cartesian Lagrangian Coordinates
- $[B^*]$ - Matrix Coefficients
 $[B]$
- $[C_b]$ - Tangential bearing damping matrix
- F_e - Elemental nodal force matrix
- F_b - Bearing nodal load
- f_e - Element body force
- $[K]$ - Global stiffness
- $[K_e]$ - Element stiffness
- $[K_{ep}]$ - Elastic-plastic stiffness
- $[K_l]$ - Linearized stiffness
- $[K_d]$ - Dynamic stiffness
- L_{ij} - Lagrangian strain tensor
 \tilde{L}
- $[M_e]$ - Element mass matrix
- $[M]$ - Global mass matrix
- $[N]$ - Shape function
- P_i, P_o - Inlet-outlet pressures of Damper Bearing
- R - Region occupied by system
- S_{ij} - 2nd Piola-Kirchhoff Stress tensor
 \tilde{S}
- t - Time
- U_i - Displacements
 \tilde{U}
- Y - Nodal displacements
- v - Volume
- ΔY - Incremental nodal displacement
- Δt - Time increment
- $\delta()$ - Variational operator

ρ_0 - Density
(\cdot) - Column Matrix
[] - Matrix
(\cdot)^T - Transpose
(\cdot)_t - Time differentiation
(\cdot)^T - Matrix Transpose
|| || - Euclidian Norm

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX 7: EQUATION DEFINITION.

$$[B] = [[B_i], [B_j], \dots]$$

$$[B_\ell] = \begin{bmatrix} N_{\ell,1} & 0 & 0 \\ 0 & N_{\ell,2} & 0 \\ 0 & 0 & N_{\ell,2} \\ N_{\ell,2} & N_{\ell,1} & 0 \\ 0 & N_{\ell,3} & N_{\ell,2} \\ N_{\ell,3} & 0 & N_{\ell,1} \end{bmatrix} ; \ell = i, j, \dots$$

$$[G] = [[G_i], [G_j], \dots]$$

$$[G_\ell] = \begin{bmatrix} N_{\ell,1} & 0 & 0 \\ N_{\ell,2} & 0 & 0 \\ N_{\ell,3} & 0 & 0 \\ 0 & N_{\ell,1} & 0 \\ 0 & N_{\ell,2} & 0 \\ 0 & N_{\ell,3} & 0 \\ 0 & 0 & N_{\ell,1} \\ 0 & 0 & N_{\ell,2} \\ 0 & 0 & N_{\ell,3} \end{bmatrix} ; \ell = i, j, \dots$$