

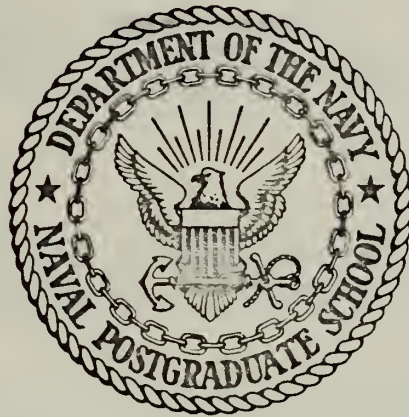
THE MODELING AND ANALYSIS
OF AN ELASTIC MECHANISM
WITH CLEARANCES

CHARLES WILLIAM GNILKA

LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIF. 93940

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

THE MODELING AND ANALYSIS OF AN ELASTIC
MECHANISM WITH CLEARANCES

by

Charles William Gnilka

Thesis Advisor:

D. Salinas

December 1971

The Modeling and Analysis of An Elastic
Mechanism With Clearances

by

Charles William Gnilka
Lieutenant, United States Navy
B.S., Villanova University, 1964

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1971

ABSTRACT

The classical approach for the analysis of linkages has been to assume rigid elements during operation. As the operational speeds of mechanisms increase, this assumption creates larger errors in the analysis. Thus a better model of a machine that considers elastic deformations will aid in more efficient design as well as provide improved accuracy and performance.

This thesis presents a simulation for the dynamic response of an elastic link model with three clearances. The simulation has been developed for an automobile, cam actuated valve train. The computer program is coded in FORTRAN IV, for an IBM-360 computer and is included as a portion of this work. The capability to visually observe the dynamic action of the model is included by a graphic display routine. This routine is implemented in SDS FORTRAN IV for a SDS-9300 computer interfaced with an ADAGE Graphic Display Terminal, Model 10.

Sample problems for various cam speeds of interest are included utilizing graphs and photos from the graphic display.

TABLE OF CONTENTS

I.	INTRODUCTION - - - - -	17
II.	PHYSICAL MODEL - - - - -	21
	A. GENERAL DESCRIPTION - - - - -	22
	1. Elements of the Model - - - - -	22
	a. Follower - - - - -	22
	b. Left Rocker Arm - - - - -	22
	c. Right Rocker Arm - - - - -	22
	d. Valve Spring - - - - -	23
	e. Valve - - - - -	23
	f. Cam - - - - -	23
	g. Valve Seat - - - - -	23
	2. Clearance Parameters - - - - -	23
	B. ELEMENT COORDINATES - - - - -	24
	C. SYSTEM COORDINATES - - - - -	26
	D. DEFINITIONS OF CONFIGURATIONS - - - - -	27
	1. External to the System - - - - -	27
	a. Clearance 1 - - - - -	27
	b. Clearance 3 - - - - -	27
	2. Internal to the System - - - - -	27
	a. Clearance 2 - - - - -	27
	b. Clearance 4 - - - - -	27
	E. SYSTEM CONFIGURATIONS - - - - -	28
	F. SHIFTING OF CONFIGURATIONS - - - - -	32

III.	MATHEMATICAL MODEL - - - - -	34
A.	EQUATIONS OF MOTION - - - - -	37
B.	MASS AND STIFFNESS MATRICES - - - - -	37
C.	DAMPING MATRIX - - - - -	37
	1. Uncoupling the Equations of Motion - - - - -	38
	2. Relation of System Parameters to Model Parameters - - - - -	40
	3. Calculation of System Damping - - - - -	40
D.	CONFIGURATION SHIFTING ALGORITHM - - - - -	42
E.	METHOD OF CALCULATIONS - - - - -	47
	1. Solution to the Governing Equations - - - - -	48
	2. Precalculated Matrices - - - - -	54
F.	REFINEMENTS IN THE SOLUTION - - - - -	56
	1. Incremental Time - - - - -	56
	2. Impact Forces - - - - -	56
IV.	IBM-360 COMPUTER PROGRAM - - - - -	59
A.	GENERAL DESCRIPTION - - - - -	59
B.	GENERALIZATION MATRICES - - - - -	59
	1. Matrix CONFIG - - - - -	60
	2. Matrix IDIM - - - - -	61
	3. Matrix LAYER - - - - -	61
	4. Matrix IQ - - - - -	62
	5. Matrix IDP - - - - -	63
	6. Matrix ICORD - - - - -	64
	7. Matrix IMPACT - - - - -	64
C.	DYNAMIC DIMENSIONING - - - - -	64
D.	PROGRAM STRUCTURE - - - - -	68

E. SUBROUTINES - - - - -	68
1. Subroutine THESIS - - - - -	70
2. Subroutine BKSTEP - - - - -	71
3. Subroutines MCK, DVVS, and DINVS - - - - -	71
4. Newmark's Method Subroutines- - - - -	71
a. Subroutine DEL - - - - -	71
b. Subroutine NEWMRK - - - - -	72
5. Subroutines for Shifting Configurations - - - - -	72
a. Subroutine MONTOR - - - - -	72
b. Subroutine DTPRIM - - - - -	72
c. Subroutine SHIFT - - - - -	73
6. Subroutine INPUT - - - - -	73
7. Subroutine CONMAT - - - - -	73
8. Subroutine INITAL - - - - -	74
9. Subroutine CAM - - - - -	74
10. Subroutine BRANCH - - - - -	74
11. Subroutine ADJUST - - - - -	75
12. Subroutine FIXFOR - - - - -	75
13. Output Subroutines - - - - -	75
a. Subroutine AGT - - - - -	75
b. Subroutines, ARRAY, GRAPH, GPRINT, OLPLOT - -	76
c. Subroutines, DRU -- - - - -	76
d. Subroutine LIST - - - - -	76
e. Subroutine PRNTMX - - - - -	77

V.	SDS-9300 COMPUTER PROGRAM - - - - -	95
A.	METHOD OF ELASTICITY CALCULATIONS - - - - -	95
B.	PROGRAM STRUCTURE - - - - -	96
	1. General Description - - - - -	96
	2. Sense Lights - - - - -	99
	a. Sense Light 1 - - - - -	99
	b. Sense Light 2 - - - - -	99
	c. Sense Light 4 - - - - -	99
	d. Sense Light 6 - - - - -	99
C.	MAIN PROGRAM - - - - -	100
D.	SUBROUTINES - - - - -	100
	1. Subroutine INITAL - - - - -	101
	2. Subroutine READ - - - - -	101
	3. Subroutine SPRCTR - - - - -	102
	4. Subroutine MIDBAR - - - - -	102
	5. Subroutine DISPLY - - - - -	102
	6. Subroutine CAM - - - - -	102
	7. Subroutine BAR1 - - - - -	103
	8. Subroutine BAR2 - - - - -	103
	9. Subroutine BAR3 - - - - -	103
	10. Subroutine BAR4 - - - - -	103
	11. Subroutine JOINT - - - - -	104
VI.	NUMERICAL RESULTS - - - - -	105
A.	MODE SHAPES AND NATURAL FREQUENCIES - - - - -	106
B.	PRE-CALCULATED SYSTEM FREQUENCIES - - - - -	107

C. SAMPLE PROBLEMS - - - - -	132
1. Example 1A, THREE CLEARANCES, 11,000 RPM, VALVE SEAT - - - - -	133
2. Example 1B, THREE CLEARANCES, 11,000 RPM, NO VALVE SEAT - - - - -	150
3. Example 1C, TWO CLEARANCES, 11,000 RPM, VALVE SEAT- - - - -	164
4. Example 1D, TWO CLEARANCES, 11,000 RPM, NO VALVE SEAT - - - - -	178
5. Example 2 & 3, THREE CLEARANCES, 9,000 & 10,000 RPM, VALVE SEAT - - - - -	187
VII. CONCLUSION- - - - -	188
APPENDIX A MASS AND STIFFNESS MATRICES - - - - -	190
APPENDIX B NEWMARK'S β PARAMETER METHOD - - - - -	196
APPENDIX C FOUR CLEARANCE PROBLEM - - - - -	199
APPENDIX D COMPUTER CONTROL CARDS - - - - -	202
APPENDIX E IBM-360 COMPUTER PROGRAM DATA CARDS - - - - -	209
APPENDIX F GRAPHS FOR EXAMPLES 2 and 3 - - - - -	210
IBM COMPUTER PROGRAM OUTPUT - - - - -	229
IBM-360 COMPUTER PROGRAM LISTING - - - - -	230
SDS-9300 COMPUTER PROGRAM LISTING - - - - -	311
LIST OF REFERENCES- - - - -	338
INITIAL DISTRIBUTION LIST - - - - -	340
FORM DD 1473- - - - -	341

LIST OF TABLES

- II.1 CONFIGURATION REFERENCE TABLE
- II.2 CONFIGURATION REFERENCE TABLE (continued)
- VI.1 GRAPH REFERENCE POINTS AND DEGREE OF CAM ROTATION FOR FIGURES 6.27 - 6.32
- VI.2 GRAPH REFERENCE POINTS AND DEGREES OF CAM ROTATION FOR FIGURES 6.35 - 6.39
- VI.3 GRAPH REFERENCE POINTS AND DEGREES OF CAM ROTATION FOR FIGURES 6.41 - 6.45
- VI.4 GRAPH REFERENCE POINTS AND DEGREES OF CAM ROTATION FOR FIGURES 6.48 - 6.50

LIST OF FIGURES

- 2.1 CAM ACTUATED VALVE TRAIN
- 2.2 AXIAL ELEMENT
- 2.3 BENDING ELEMENT
- 2.4 COMBINED ELEMENT COORDINATES
- 2.5 FULL SYSTEM COORDINATES
- 2.6 CONFIGURATION SHIFTING DIAGRAM
- 3.1 CONFIGURATION MATRIX
- 3.2 SHIFTING EXAMPLE NO. 1
- 3.3 SHIFTING EXAMPLE NO. 2
- 3.4 MATRIX LAYERS
- 3.5 INCREMENTAL TIME ΔT
- 3.6 SYSTEM CURVES FOR IMPACT CONSIDERATIONS
- 4.1 MATRIX CONFIG
- 4.2 MATRIX IDIM
- 4.3 MATRIX LAYER
- 4.4 MATRIX IQ
- 4.5 MATRIX IDP
- 4.6 MATRIX ICORD
- 4.7 DYNAMIC DIMENSIONING OF ARRAY SYSM
- 4.8 DYNAMIC DIMENSIONING OF ARRAY BMIV
- 4.9 IBM-360 COMPUTER PROGRAM STRUCTURE
- 4.10 SUBROUTINE THESIS FLOW DIAGRAM
- 4.11 SUBROUTINE BKSTEP FLOW DIAGRAM
- 4.12 SUBROUTINE DEL FLOW DIAGRAM
- 4.13 SUBROUTINE NEWMRK FLOW DIAGRAM
- 4.14 SUBROUTINE MONTOR FLOW DIAGRAM

- 4.15 SUBROUTINE CAM FLOW DIAGRAM
- 4.16 SUBROUTINE DTPRIM FLOW DIAGRAM
- 4.17 SUBROUTINE INPUT FLOW DIAGRAM
- 4.18 SUBROUTINE CONMAT FLOW DIAGRAM
- 4.19 SUBROUTINE BRANCH FLOW DIAGRAM
- 4.20 SUBROUTINE ADJUST FLOW DIAGRAM
- 4.21 SUBROUTINE FIXFOR FLOW DIAGRAM
- 4.22 SUBROUTINE AGT FLOW DIAGRAM
- 4.23 SUBROUTINE ARRAY FLOW DIAGRAM
- 4.24 SUBROUTINE GRAPH FLOW DIAGRAM
- 4.25 SUBROUTINE GPRINT FLOW DIAGRAM
- 5.1 SDS-9300 COMPUTER PROGRAM STRUCTURE
- 5.2 OVER LAY STRUCTURE
- 6.1 FULL SYSTEM MODE SHAPES AND NATURAL FREQUENCIES
- 6.2 REDUCED SYSTEM MODE SHAPES AND NATURAL FREQUENCIES
- 6.3 CONFIGURATION I - Mass, Damping, and Stiffness
- 6.4 CONFIGURATION I - Precalculated Matrices; BMIV, DA,
and DV
- 6.5 CONFIGURATION II - Mass, Damping, and Stiffness
- 6.6 CONFIGURATION II - Precalculated Matrices; COLM, COLC,
and COLK
- 6.7 CONFIGURATION II - Precalculated Matrices; BMIV, DA,
and DV
- 6.8 CONFIGURATION III - Mass, Damping, and Stiffness
- 6.9 CONFIGURATION III - Precalculated Matrices; COLM, COLC,
and COLK

- 6.10 CONFIGURATION IV - Mass, Damping, and Stiffness
- 6.11 CONFIGURATION IV - Precalculated Matrices; COLM, COLC, and COLK
- 6.12 CONFIGURATION IV - Precalculated Matrices; BMIV, DA, and DV
- 6.13 CONFIGURATION V - Mass, Damping, and Stiffness
- 6.14 CONFIGURATION V - Precalculated Matrices; COLM, COLC, and COLK
- 6.15 CONFIGURATION V - Precalculated Matrices; BMIV, DA, and DV
- 6.16 CONFIGURATION VI - Mass, Damping, and Stiffness
- 6.17 CONFIGURATION VI - Precalculated Matrices; COLM, COLC, and COLK
- 6.18 CONFIGURATION VI - Precalculated Matrices; COLM, COLC, and COLK
- 6.19 CONFIGURATION VI - Precalculated Matrices; BMIV, DA, and DV
- 6.20 CONFIGURATION VII - Mass, Damping, and Stiffness
- 6.21 CONFIGURATION VII - Precalculated Matrices; COLM, COLC, and COLK
- 6.22 CONFIGURATION VII - Precalculated Matrices; BMIV, DA, and DV
- 6.23 CONFIGURATION VIII - Mass, Damping, and Stiffness
- 6.24 CONFIGURATION VIII - Precalculated Matrices; COLM, COLC, and COLK
- 6.25 CONFIGURATION VIII - Precalculated Matrices; BMIV, DA, and DV
- 6.26 SAMPLE NUMERICAL RESULTS
- 6.27 - 6.34 EXAMPLE 1A, THREE CLEARANCES, 11,000 RPM, VALVE SEAT
- 6.35 - 6.40 EXAMPLE 1B, THREE CLEARANCES, 11,000 RPM, NO VALVE SEAT

- 6.41 - 6.47 EXAMPLE 1C, TWO CLEARANCES, 11,000 RPM,
 VALVE SEAT
- 6.48 - 6.51 EXAMPLE 1D, TWO CLEARANCES, 11,000 RPM,
 NO VALVE SEAT
- A.1 FULL SYSTEM MASS MATRIX
- A.2 FULL SYSTEM STIFFNESS MATRIX
- A.3 REDUCED SYSTEM MASS MATRIX
- A.4 REDUCED SYSTEM STIFFNESS MATRIX
- B.1 TIME STEP ACCELERATION CURVE
- C.1 FOUR CLEARANCE SYSTEM CONFIGURATIONS
- C.2 FOUR CLEARANCE SYSTEM CONFIGURATION SHIFTING DIAGRAM
- D.1 INITIALIZE DISK SPACE
- D.2 LOAD SUBROUTINE ONTO DISK
- D.3 LIST DATA SET VTOC AND PDS
- D.4 EXECUTE PROGRAM WITH SUBROUTINES ON DISK
- D.5 LOAD MAIN PROGRAM ONTO DISK
- D.6 EXECUTE ENTIRE PROGRAM FROM DISK
- D.7 READ DATA SET MEMBERS ONTO SCRATCH DISK
- D.8 COPY ALL DATA SET MEMBERS
- D.9 COPY PART OF DATA SET MEMBERS
- D.10 TRANSFERRING FORTRAN SOURCE CODE FROM CARDS TO MAGNETIC TAPE
- D.11 PROGRAM EXECUTION FROM FORTRAN OBJECT ON MAGNETIC TAPE
- D.12 PLACING PROGRAM AS BINARY CODE ON MAGNETIC TAPE
- D.13 PROGRAM EXECUTION FROM BINARY CODE ON MAGNETIC TAPE
- F.1 - F.8 EXAMPLE 2, THREE CLEARANCES, 9,000 RPM, VALVE SEAT
- F.9 - F.16 EXAMPLE 3, THREE CLEARANCES, 10,000 RPM, VALVE SEAT

LIST OF PHOTOGRAPHS

- 1.1 AGT Display
- 5.1 AGT Listing Question
- 6.1 AGT Title Page
- 6.2 - 6.23 Example 1A, Three Clearances, 11,000 RPM,
Valve Seat
- 6.24 - 6.42 Example 1B, Three Clearances, 11,000 RPM,
No Valve Seat
- 6.43 - 6.56 Example 1C, Two Clearances, 11,000 RPM,
Valve Seat
- 6.57 - 6.68 Example 1D, Two Clearances, 11,000 RPM,
No Valve Seat

LIST OF SYMBOLS

A	Area
[C]	System damping matrix
c_{ij}	Element of damping matrix
E	Elastic modulus
F_i	Forces
$\{F^*\}$	Normalized force vector
g	Gravitational Constant
I	Moment of inertia
[K]	System stiffness matrix
k_{ij}	Element of stiffness matrix
L	Length
[M]	System mass matrix
M	Mass
m_{ij}	Elements of mass matrix
$\{p\}$	Force vector used in integration solution
q	System coordinate
t	Time
ΔT	Time increment used in integration solution
T_i	Period
u	Element coordinate
$\{\alpha\}$	Vector used in integration solution
[β]	Transformation matrix
ζ	Damping coefficient
	Cam angle of rotation

$\{\phi\}$ Eigenvectors listed in columns, modal matrix
 $\{\psi\}$ Vector used in integration solution
 ω_n, Ω Natural frequency
 $\langle \rangle$ Row Vector

ACKNOWLEDGEMENTS

The author wishes to express his appreciation to his advisors, Dr. Richard C. Winfrey currently at the Naval Civil Engineering Laboratory, Port Hueneme, who introduced this author to the project and for his constructive supervision; and Professor David Salinas of the Naval Postgraduate School for his moral support and guidance.

The Naval Postgraduate School Computer Center provided facilities for the computer work, except the computer graphic portion. This portion was accomplished with facilities and technical advice provided by the Electrical Engineering Department Computer Laboratory. In this regard, the author wishes to express his sincere thanks to Mr. Robert L. Limes, computer laboratory Supervisor, and the staff Mr. William A. Thomas and Mr. Albert Wong for the hours in assistance with graphic computer programing and hardware difficulties.

Finally, the author wishes to acknowledge the encouragement, patience, and moral support of his wife, Martha and for her assistance in preparing the manuscript.

I. INTRODUCTION

The classical approach to the dynamic analysis of linkages has been to assume that all members of the machine move without deformation during operation, i.e. that they move as rigid bodies. Based on this assumption, material choices and sizing to withstand stresses for the applied loads, the linkage deformations and the response motion have been severely idealized. Application of factors of safety have been necessary to account for this idealization. At lower operational speeds design criteria are met. However as the speeds increase the resulting large inertial forces and elastic deflections are more severe and thus the possibility of failure more acute. The design becomes invalid when the critical speed is reached where the machine either fails or where it no longer performs its intended task. Thus expensive and time consuming models must be built and tested, for elastic deformations over the range of operational speeds before the kinematic analysis can be fully accepted. The goal then is to obtain a more accurate dynamic model of a machine in order to better predict and understand its performance.

With the advent of the digital computer, more complex mathematical models of machine are being developed. Engineers are able to accomplish more refined designs with a higher degree of operational confidence, less of a factor of safety, better material choice, and fewer physical model experiments.

This work presents a simulation for the dynamic response of a planar elastic link mechanism, free to move in its rigid body motion

as well as its elastic motion. The structural elements deform elastically due to forces applied by neighboring elements, and due to inertial forces. This simulation has been developed for an automobile, cam actuated valve train, with three clearances. The clearances are considered to exist at the (1) cam-pushrod, (2) valve-valve seat, and (3) rocker arm-pushrod. To simplify the required derivations, the model is assumed to initially consist of straight members with constant circular cross-sectional bars. The assumption of constant circular cross-sectional bars is selected for ease in computing beam section properties. The use of straight links greatly simplifies the equations of elastic deformation.

An existing computer program written by Anderson and Winfrey (Ref. 2), was dependent on separately coded sections for each physical configurations of the two clearance model of a cam actuated valve train. As the number of system clearances are increased, the number of system configurations increase exponentially and separately coded program sections become infeasible.

The extent of this author's objectives and work on the existing program is to:

- (1) Restructure and reprogram the simulation to provide for a generalized method for establishing the physical model configurations.
- (2) Modify this program to allow for more than two clearances, i.e. add clearances internal to the system.
- (3) Establish a method for the dynamic graphical display of the model showing the resulting elastic deformations of the model members at high operational speeds.

Objectives (1) and (2) have been accomplished using the IBM-360 computer for the simulation. Objective (3) has been accomplished with the aide of the ADAGE Graphics Terminal model 10 (AGT-10) interfaced with a Scientific Data System computer model 9300 (Photo 1.1). Data generated by the IBM computer is loaded into the SDS computer by means of punched cards or magnetic tape. This data is then manipulated into a form acceptable for display by the AGT-10.

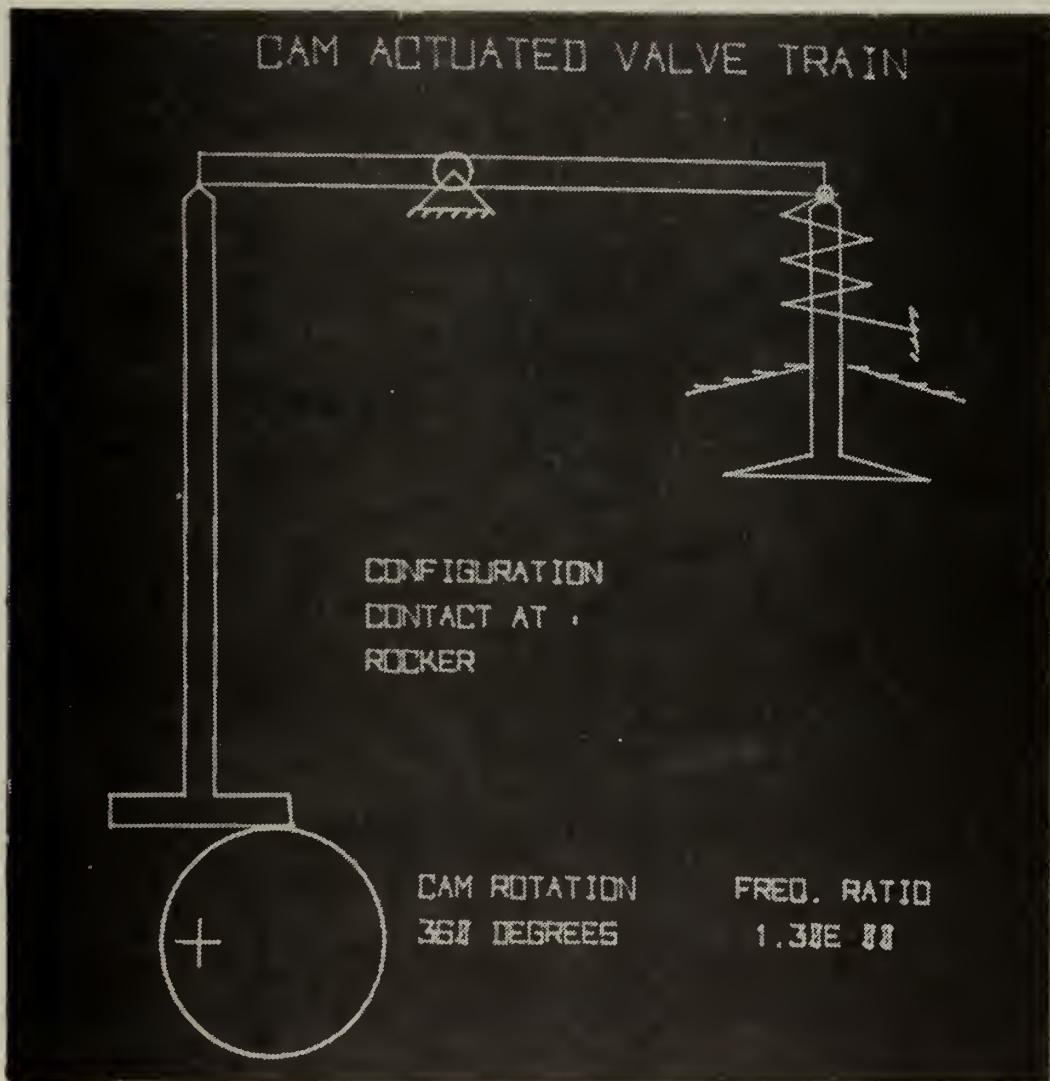


Photo 1.1 AGT DISPLAY

In the following chapters, the physical model, and the concepts and formulation for the corresponding mathematical model are described. The dynamic analysis and the elastic deformation computer programs are discussed and listed at the end of this work. Finally, several numerical examples are given with the model responses shown as AGT photographs and CALCOMP graphs.

In the following chapters, the physical model, and the concepts and formulation for the corresponding mathematical model are described. The dynamic analysis and the elastic deformation computer programs are discussed and listed at the end of this work. Finally, several numerical examples are given with the model responses shown as AGT photographs and CALCOMP graphs.

II. PHYSICAL MODEL

In this chapter the physical system, Figure 2.1, is idealized to enable formulation of a mathematical model whose analysis is tractable. The description covers the interaction between the system parameters and the physical constraints that limit the model's motion. The physical elements that comprise the system are listed and their functions explained. Additionally their physical properties, size and material content, are noted.

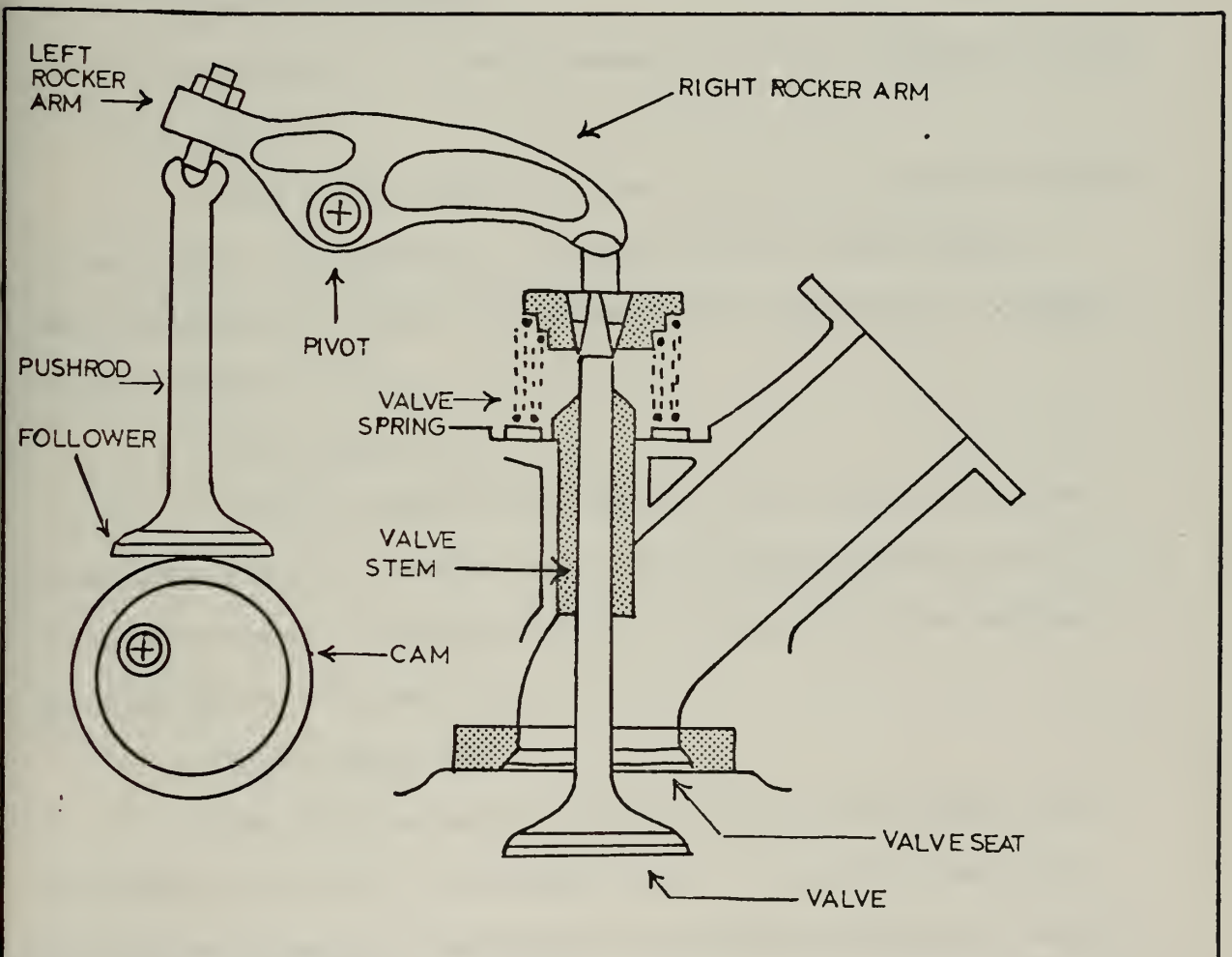


Figure 2.1 CAM ACTUATED VALVE TRAIN

A. GENERAL DESCRIPTION

In order to relate the formulation of this mathematical model to the real world, an automobile car engine valve train assembly is considered. However, any similar physical model might have been chosen. All members of the model are assumed to be manufactured from steel with an elastic modulus of 30×10^6 pounds per square inch and a specific weight of .283 pounds per cubic inch. Additionally longitudinal members are considered as rods with circular cross sections. The diameter and length of each bar are specified.

1. Elements of the Model

The elements of the valve assembly pictured in Figure 2.1 are:

a. Follower

This reciprocating flat faced follower is rigidly attached to the pushrod. The pushrod is considered as an elastic member. These members have harmonic displacement. The diameter is .3 inches; length 9 inches.

b. Left Rocker Arm

The left rocker arm is pinned to and rotates about the fixed pivot point at its right end. This element provides for an "internal" clearance between itself and the pushrod at its left end. The diameter is .6 inches; length 1 inch.

c. Right Rocker Arm

The right rocker arm is also pinned to and rotates about the fixed pivot point. At its right end it is pinned to the valve stem and valve spring. The diameter is .6 inches; length 1.5 inches.

d. Valve Spring

The free end of the valve spring is connected to the rocker arm and valve stem pin, the other end is rigidly fixed to ground. The spring is assumed to have a spring constant of 245 pounds per inch. The diameter is .07 inches; free length 36 inches.

e. Valve

This reciprocating valve is rigidly attached to the valve stem. The valve stem is assumed to be elastic. The upper end of the valve stem is connected to the common pin with the valve spring and right rocker arm. The diameter is .3 inches; length 3 inches.

To complete the model two additional elements are added external to the system. Both provide for the "external" clearances. They are:

f. Cam

The offset circular disk cam imparts sinusoidal motion to the follower.

g. Valve Seat

This rigid seating surface for contact by the valve may be removed from the system when desired.

The rocker arm, although modeled as two elements, is considered to be one continuous member. Thus only rotation about the fixed pivot point is permitted.

2. Clearance Parameters

The clearance or interference at the valve seat may be eliminated by the introduction of a predetermined parameter. Due to the choice of the coordinate system, the parameter set less than the minimum value of the valve displacement has the effect of removing

the valve seat from the system. This provides for the flexibility to conduct studies of the valve motion unimpeded by the valve seat.

Additional flexibility of the model is provided by introduction of another clearance parameter. This parameter has the effect of eliminating the internal clearance between the pushrod and left rocker arm. Thus these two members can be pinned together when desired.

The circular disk cam rotates counterwise. This motion first lifts the follower by direct contact to its extreme position and next permits the follower to return to its initial position. The harmonic cycle is then repeated. For systems with cams of this type, the return motion of the follower must be accomplished by some force external to the cam. Therefore the valve spring is introduced to supply this force. For this work, simple harmonic motion is specified for the follower. The follower then oscillates about the midstroke position with an acceleration proportional to its distance from this position. The amplitude is half the stroke. The cam profile used to obtain the harmonic motion has been generated by the equation $.5 \sin(\omega t)$ inches, where ω is the cam speed in radians per second. Similar studies of various other cam profiles are discussed in Refs. 14-18.

B. ELEMENT COORDINATES

The model of the mechanism consists of two basic elements which account for all the relevant energy. The axial element, Figure 2.2 considers tension and compression forces. The bending element, Figure 2.3, considers only bending energy. The coordinates of the five

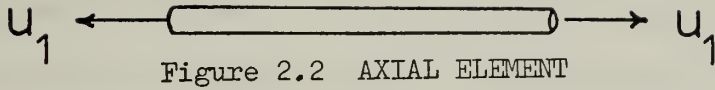


Figure 2.2 AXIAL ELEMENT

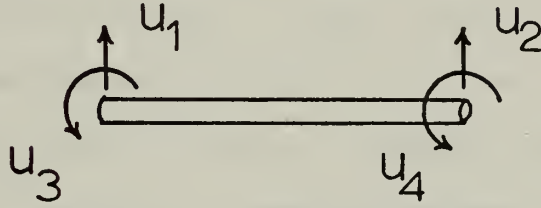


Figure 2.3 BENDING ELEMENT

elements of the system, i.e. pushrod, rocker arms, valve, and spring are chosen as indicated in Figure 2.4.

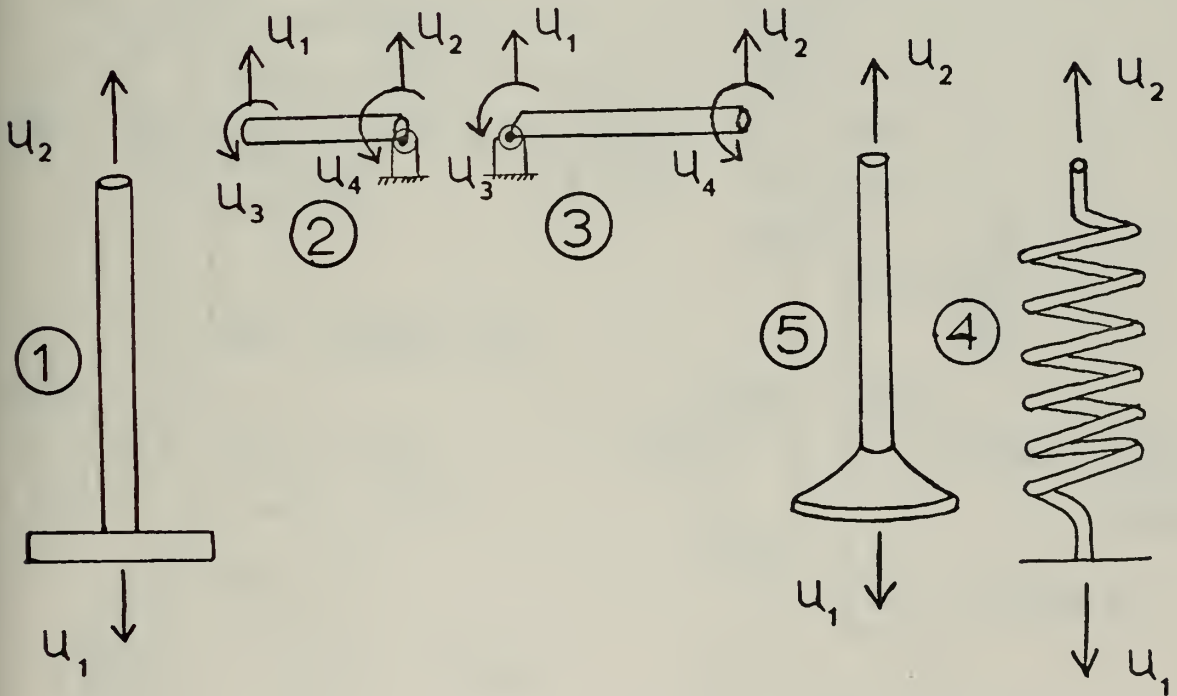


Figure 2.4 COMBINED ELEMENT COORDINATES

C. SYSTEM COORDINATES

Assuming small deformation theory, the linear model is composed of the above elements by superposition. The external members, the cam and valve seat, are also assigned coordinates for reference. As noted above, the pivot point and lower end of the spring are connected to ground and are therefore rigid. The systems coordinates $\{q\}$ define motion. Since no motion is assumed for these two connections, no coordinates are necessary. Recalling that each member was assumed straight, the system coordinates will generally coincide with the element coordinates. The coordinate indices and their positive senses are shown in Figure 2.5.

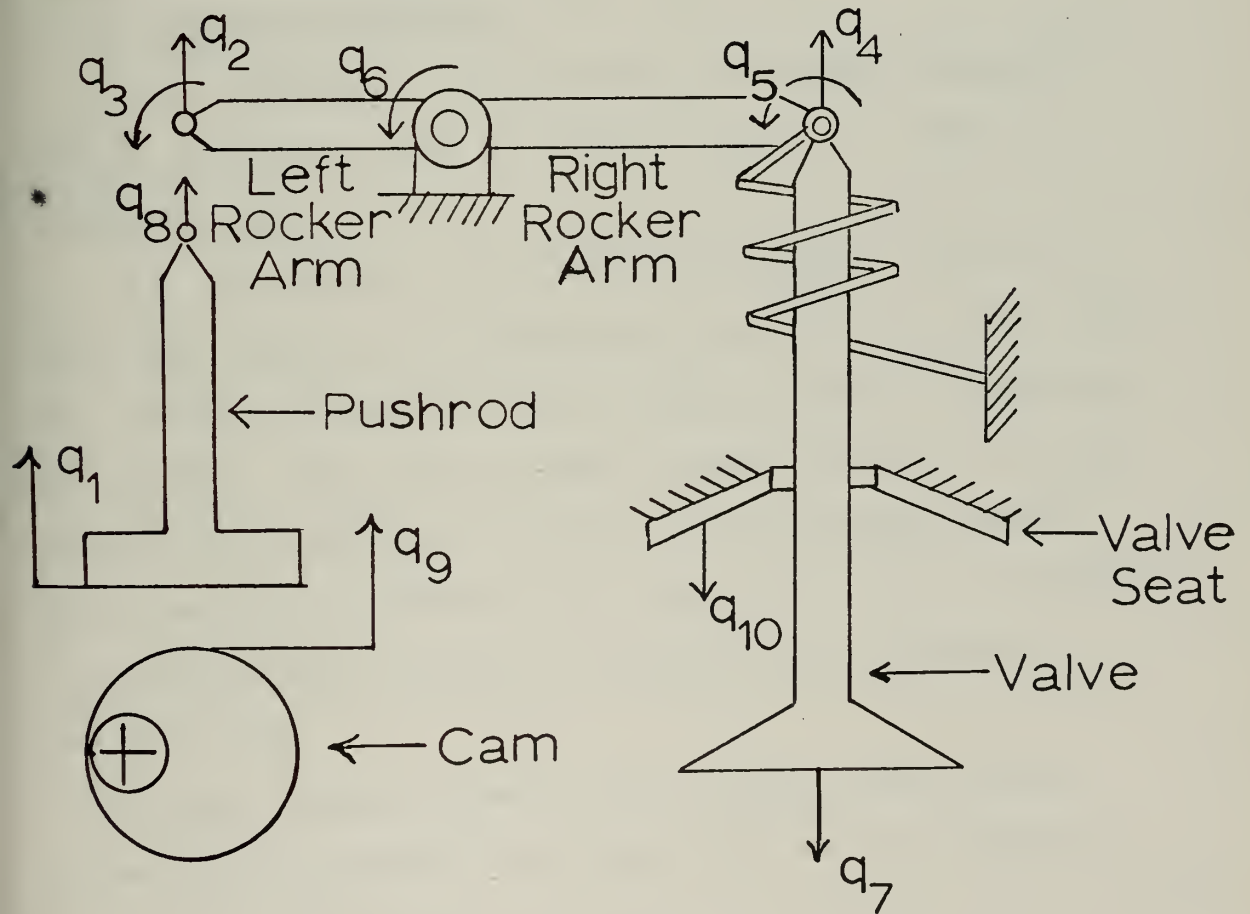


Figure 2.5 FULL SYSTEM COORDINATES

D. DEFINITIONS OF CONFIGURATIONS

The model shown in Figure 2.5 has two external clearances and one internal clearance. The term external refers to a displacement difference between a member of the system and an external member. Internal clearances are displacement differences between two internal members of the system. The clearances are:

1. External To The System

a. Clearance 1. The displacement difference between the cam, (coordinate nine), and the follower, (coordinate one).

b. Clearance 3. The displacement difference between the valve, (coordinate seven), and the valve seat, (coordinate ten).

2. Internal To The System

a. Clearance 2. The displacement difference between the pushrod, (coordinate eight) and left end of the left rocker arm, (coordinate two).

b. Clearance 4. The displacement difference between the valve stem, free end of the spring, and right end of the right rocker arm. The basic concept for the four clearance problem is presented in Appendix C. Since only the three clearance problem is considered for this work, clearance four is zero. Thus the members remain pinned and all move together at this connection. Only one coordinate is required to model the motion of the joint and is designated coordinate four.

Thus the external clearances define positions where external forces must be considered, i.e. forces associated with coordinate

one and coordinate seven. The internal clearances produce internal forces. These equal and opposite forces act upon the coordinate pair two and eight only when they are in contact.

The three clearances, where positive displacement differences between the coordinate pairs may exist, will be referred to as noted above, i.e. clearances 1, 2, and 3. Since clearance 4 will not be modeled, reference to this position will be referred to as coordinate four. Forces between clearances will be discussed in section 3.E.

E. SYSTEM CONFIGURATION

Depending upon the value of the force and displacement at these three clearances, the model can exist in one of eight possible configurations. If clearance 2 is considered pinned, the difference between coordinates two and eight is zero. The possible configurations is then reduced to four and is the study conducted by Anderson (Ref. 2).

The eight configurations are diagramed in Table 2.1 and 2.2.

They are:

1. No contact at clearance 1, clearance 2, or clearance 3.
2. Contact at clearance 3; no contact at clearance 1, or clearance 2.
3. Contact at clearance 2; no contact at clearance 1, or clearance 3.
4. Contact at clearance 2, and clearance 3; no contact at clearance 1.
5. Contact at clearance 1; no contact at clearance 2, and clearance 3.

CONFIGURATION		CONTACT AT CLEARANCE	MONITORED VALUES		ASSUMED VALUES	
No.	Figure		Force	DISPL DIFF	Force	Displacements
I		NONE (NO CONTACT)	--	$q_1 - q_9$ $q_2 - q_8$ $q_7 - q_{10}$	$F_1=0$ $F_2=0$ $F_7=0$ $F_8=-F_2$	ALL VALUES CALCULATED
II		3 (VALVE SEAT)	F_7	$q_1 - q_9$ $q_2 - q_8$	$F_1=0$ $F_2=0$ $F_8=-F_2$	$\ddot{q}_7 = \ddot{q}_{10}$; $\dot{q}_7 = \dot{q}_{10}$; $q_7 = q_{10}$
III		2 (ROCKER)	F_2	$q_1 - q_9$	$F_1=0$	$\ddot{q}_2 = \ddot{q}_8$; $\dot{q}_2 = \dot{q}_8$; $q_2 = q_8$. ∴ Seven Coordinate Reduced System
IV		2 and 3 (VALVE SEAT ROCKER)	F_2 F_7	$q_1 - q_9$	$F_1=0$ $F_8=-F_2$	$\ddot{q}_2 = \ddot{q}_8$; $\dot{q}_2 = \dot{q}_8$; $q_2 = q_8$ ∴ Seven Coordinate Reduced System $\ddot{q}_7 = \ddot{q}_{10}$; $\dot{q}_7 = \dot{q}_{10}$; $q_7 = q_{10}$

TABLE 2.1 CONFIGURATION REFERENCE TABLE

CONFIGURATION		CONTACT AT CLEARANCE	MONITORED VALUES		ASSUMED VALUES	
No.	Figure		Force	DISPL DIFF	Force	Displacements
V		1 (CAM)	F_1	$q_2 - q_8$ $q_7 - q_{10}$	$F_2=0$ $F_7=0$ $F_8=-F_2$	$\ddot{q}_1 = \ddot{q}_9; \dot{q}_1 = \dot{q}_9; q_1 = q_9$
VI		1 and 3 (CAM VALVE SEAT)	F_1 F_7	$q_2 - q_8$	$F_2=0$ $F_8=-F_2$	$\ddot{q}_1 = \ddot{q}_9; \dot{q}_1 = \dot{q}_9; q_1 = q_9$ $\ddot{q}_7 = \ddot{q}_{10}; \dot{q}_7 = \dot{q}_{10}; q_7 = q_{10}$
VII		1 and 2 (CAM ROCKER)	F_1 F_2	$q_7 - q_{10}$	$F_7=0$ $F_8=-F_2$	$\ddot{q}_2 = \ddot{q}_8; \dot{q}_2 = \dot{q}_8; q_2 = q_8$ \therefore Seven Coordinate Reduced System $\ddot{q}_1 = \ddot{q}_9; \dot{q}_1 = \dot{q}_9; q_1 = q_9$
VIII		1, 2, and 3 (CAM ROCKER VALVE SEAT)	F_1 F_2 F_7	- - -	$F_8=-F_2$	$\ddot{q}_2 = \ddot{q}_8; \dot{q}_2 = \dot{q}_8; q_2 = q_8$ \therefore Seven Coordinate Reduced System $\ddot{q}_1 = \ddot{q}_9; \dot{q}_1 = \dot{q}_9; q_1 = q_9$ $\ddot{q}_7 = \ddot{q}_{10}; \dot{q}_7 = \dot{q}_{10}; q_7 = q_{10}$

TABLE 2.2 CONFIGURATION REFERENCE TABLE (continued)

6. Contact at clearance 1, and clearance 3; no contact at clearance 2.
7. Contact at clearance 1, and clearance 2; no contact at clearance 3.
8. Contact at clearance 1, clearance 2, and clearance 3.

These eight configurations will be referred to as configurations I through VIII respectively.

The physical constraints of the system require that at any given instant of time, one and only one configuration will be satisfied. The forces and displacement differences at clearances 1, 2, and 3 are monitored to determine which configuration is valid for the calculated values. For example when the model is in a configuration that requires contact at a clearance and the calculated force at this clearance becomes zero (or negative), then contact is lost at that coordinate pair. Similarly, when the model is in a configuration that requires no contact at a clearance and the calculated displacement becomes zero (or negative), an interference exists and contact is regained at the coordinate.

The configuration and the values monitored for the appropriate coordinates are listed in Tables 2.1 and 2.2. Since this model always has contact at coordinate four contact at clearance 2 will be referred to as contact at the "rocker", (between left rocker arm and pushrod).

F. SHIFTING OF CONFIGURATIONS

As discussed above, the model shifts from one configuration to another, depending on first, the configuration the model is presently in; and second, the values of the monitored parameters. This shifting of configurations can be more easily understood by considering the flow diagram in Figure 2.6. The configuration schematic of the two clearance problem of Anderson (Ref. 2) appeared as a square, (face 3-4-8-7). As seen here the schematic of the three clearance problem becomes a cube. The solid lines and adjacent force parameter indicate a shift in the direction of the arrow caused by that particular force parameter. Similarly the dashed lines and adjacent coordinate parameters indicate a shift in the direction of the arrow caused by that particular negative displacement difference. Also indicated in the configuration flow diagram is an impact indicator. When a positive clearance becomes zero (or negative) a rapid change in the associated coordinate acceleration occurs which lends to high forces, i.e. impact. This is further discussed in section 3.F2. Impact is indicated by the rectangular boxes associated with the dashed shift lines. The figures within these boxes are the clearances at which the impact occurs.

III. MATHEMATICAL MODEL

In this chapter the equations of motion which apply to the model are presented and analyzed. The development of the system mass, and stiffness matrices; the two eigenvalue problems; and the formulation of the system damping are discussed. An algorithm to shift the model to the configuration corresponding to the values of the monitored parameters is presented. Finally the numerical methods by which the solutions are obtained is explained.

A. EQUATIONS OF MOTION

The model is considered to be a linear viscous damping spring-mass system, excited by time varying forces. The differential equations of motion can be written in terms of the system coordinates as

$$[M] \{\ddot{q}\} + [C] \{\dot{q}\} + [K] \{q\} = \{F\} \quad (3.1)$$

where in the usual terminology, $[M]$ represents the system mass matrix, $[C]$ the system damping matrix, and $[K]$ the system stiffness matrix.

The formulation of these matrices has been developed using Lagrange's equations, with the matrix elements based on finite element techniques, Rubenstein and Hurty (Refs. 8 and 9).

The elements of the model, as developed in section 2.B, have a discrete set of element coordinates ($u_i = 1, 2, \dots, n$) that represent the "n", rotational and translational element displacements. These element coordinates $\{u\}$ are directly related to the system

coordinates $\{q\}$ by

$$\{u\} = [\beta]\{q\} \quad (3.2)$$

The transformation matrix $[\beta]$ is formed by applying the conditions of compatibility to the mechanism. For example, if element coordinate u_4 of element three is rotated by one unit, recalling that linear theory is assumed, it is seen that the relation is

$$(u_4)_3 = q_5 \quad (3.3)$$

This scalar equation viewed as the matrix equation, (3.2), becomes

$$(u_4)_3 = \langle 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \rangle \begin{pmatrix} q_1 \\ q_2 \\ \cdot \\ q_5 \\ \cdot \\ \cdot \\ q_8 \end{pmatrix}$$

Thus the fifth row of the transformation matrix $[\beta]$ has been obtained. By similar procedures, the entire transformation matrix for the eight coordinate system is developed. Therefore equation (3.2) can now be written as equation (3.4). Where the column matrix of the element coordinates, $\{u\}$, are partitioned by their appropriate elements, and the 14×8 transformation matrix, $[\beta]$ is partitioned into sub-matrices corresponding to the respective elements. A more concise form may be written for equation (3.4) as seen in equation (3.5).

$$\left\{ \begin{array}{l} \left. \begin{array}{l} u_1 \\ u_2 \end{array} \right\} \quad 1 \\ \left. \begin{array}{l} u_1 \\ u_2 \\ u_3 \\ u_4 \end{array} \right\} \quad 2 \\ \left. \begin{array}{l} u_1 \\ u_2 \\ u_3 \\ u_4 \end{array} \right\} \quad 3 \\ \left. \begin{array}{l} u_1 \\ u_2 \end{array} \right\} \quad 4 \\ \left. \begin{array}{l} u_1 \\ u_2 \end{array} \right\} \quad 5 \end{array} \right\} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \left\{ \begin{array}{l} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \\ q_8 \end{array} \right\} \quad (3.4)$$

$$\left\{ \begin{array}{l} \{u\}_1 \\ \{u\}_2 \\ \{u\}_3 \\ \{u\}_4 \\ \{u\}_5 \end{array} \right\} = \begin{bmatrix} [\beta]_1 \\ [\beta]_2 \\ [\beta]_3 \\ [\beta]_4 \\ [\beta]_5 \end{bmatrix} \{q\} \quad (3.5)$$

B. MASS AND STIFFNESS MATRICES

The element mass $[M_e]$ and stiffness $[K_e]$ matrices are inherent characteristics of each element and can readily be formulated with finite element methods, (Refs. 1 and 8). The system mass $[M]$ and stiffness $[K]$ matrices can be formed with the aid of the transform matrix $[\beta]$ as:

$$[M] = \sum_{i=1}^5 [\beta]_i^T [M_e]_i [\beta]_i \quad (3.6)$$

$$[K] = \sum_{i=1}^5 [\beta]_i^T [K_e]_i [\beta]_i \quad (3.7)$$

where $[\beta]_i$ is the i^{th} element submatrix of $[\beta]$; $[K_e]_i$ is the i^{th} element stiffness matrix; and $[M_e]_i$ is the i^{th} element mass. The complete formulation is discussed in appendix A.

C. DAMPING MATRIX

The damping forces and hence system damping matrix is not as easily formulated since these values may depend upon the vibration of the system as well as elements exterior to it. To reduce the complexity of the solution of the differential equations of motion linear viscous damping is assumed.

The system damping is formulated by Hurty and Rubinstein, (Ref. 9) by: (1) solving the eigenvalue problem associated with the equations of motion of the system in undamped free vibrations,

for the modal matrix; (2) using this modal matrix to form the diagonalized system mass and stiffness matrices; (3) from these matrices forming the diagonalized system damping matrix; and (4) using this diagonalized damping matrix to solve for the system damping matrix.

The dynamic model consists of two eigenvalue problems. One eigenvalue problem is for the system of Figure 2.5. Here eight coordinates are required to solve the dynamic equations, since the pushrod and rocker arm are separated, coordinate two and coordinate eight have independent motion. The other eigenvalue problem is associated with the model in contact at clearance 2 and takes the form derived by Anderson (2). When the motion at coordinates two and eight are equal, the eight system equations must be reduced to a lower order system by removal of the redundant equation to facilitate a solution. Thus the system damping matrix, as well as the system mass and stiffness matrices, become order seven. This will be considered in section 3.E.

1. Uncoupling the Equations of Motion

When the differential equations of motion are derived from a set of convenient coordinates, the equations are normally coupled. If the system displacements $\{q\}$ are expressed in terms of the normal modes, the governing equations will uncouple. Thus from the equations of free undamped vibration

$$[M] \{\ddot{q}\} + [K] \{q\} = 0 \quad (3.8)$$

the uncoupled, diagonal mass $[m]$ and stiffness $[k]$ matrices, and the modal force vector $\{F^*\}$, Hurty and Rubinstein, (Ref. 9).

2. Relation of System Parameter To Modal Parameter

The diagonal matrices, $[m]$, $[c]$ and $[k]$ are each related to their corresponding system matrices by transformation with the modal matrix.

$$[m] = [\Phi]^T [M] [\Phi] \quad (3.14)$$

$$[c] = [\Phi]^T [C] [\Phi] \quad (3.15)$$

$$[k] = [\Phi]^T [K] [\Phi] \quad (3.16)$$

Additionally the modal acceleration $\{\ddot{\eta}\}$ and displacement $\{\eta\}$ are related to the system acceleration $\{\ddot{q}\}$ and displacement $\{q\}$ by the modal matrix.

$$\{\ddot{q}\} = [\Phi] \{\ddot{\eta}\} \quad (3.17)$$

$$\{q\} = [\Phi] \{\eta\} \quad (3.18)$$

3. Calculation of the System Damping

The system damping matrix may be obtained from equation (3.15) by matrix multiplication. Thus the diagonal elements of $[c]$ must be calculated before the desired result, $[C]$, can be found. The governing equations are now uncoupled and the uncoupled mass $[m]$ and stiffness $[k]$ matrices are known. Thus the solution formulated for the single degree of freedom system may be applied to the n degree of freedom system.

Equations (3.19) through (3.21), from Thomson (Ref. 3), may be utilized to obtain the desired diagonal elements of the uncoupled damping matrix $\begin{bmatrix} c \end{bmatrix}$. The natural frequency in radians per second of the undamped oscillation for a single degree freedom system is

$$\omega_n = \sqrt{k/m} \quad (3.19)$$

The damping factor (ζ), is defined as the ratio of the viscous damping (c) to the critical damping factor (c_c).

$$\zeta = c/c_c \quad (3.20)$$

The critical damping factor is also given by

$$c_c = 2 m \omega_n = 2\sqrt{m k} \quad (3.21)$$

Upon substitution of equation (3.21) into equation (3.20), the viscous damping becomes

$$c = 2 \zeta \sqrt{k m} \quad (3.22)$$

In terms of the uncoupled system, the value of the diagonal element of the natural frequencies, corresponding to equation (3.19), are

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_n \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{k_{11}}{m_{11}} \\ \frac{k_{22}}{m_{22}} \\ \vdots \\ \frac{k_{nn}}{m_{nn}} \\ 0 \end{bmatrix} \quad (3.23)$$

where the k's and m's are the diagonal elements from $\begin{bmatrix} k \end{bmatrix}$ and $\begin{bmatrix} m \end{bmatrix}$.

Similarly equation (3.22) can be written as

$$c_{ii} = 2\zeta_i \sqrt{k_i m_i} \quad (i = 1, 2, \dots, n) \quad (3.24)$$

and thus the diagonal values of the uncoupled damping matrix $\begin{bmatrix} c \end{bmatrix}$ are obtained.

An arbitrary choice of values for the damping factor vector $\{\zeta\}$ was considered on the basis of the natural frequencies. The damping factor, (ζ_i) , corresponding to the fundamental frequency is set equal to $\frac{1}{2}$. The remaining elements of $\{\zeta\}$ are set to equal $1/10$.

Having established $\begin{bmatrix} c \end{bmatrix}$, the damping matrix $[C]$ may be formed from equation (3.15) by matrix multiplication on the left by $[\Phi^T]^{-1}$ and on the right by $[\Phi]^{-1}$ as

$$[\Phi^T]^{-1} \begin{bmatrix} c \end{bmatrix} [\Phi]^{-1} = [\Phi^T]^{-1} [\Phi]^T [C] [\Phi] [\Phi]^{-1} \quad (3.25)$$

This gives the desired result

$$[C] = [\Phi^T]^{-1} \begin{bmatrix} c \end{bmatrix} [\Phi]^{-1} \quad (3.26)$$

D. CONFIGURATION SHIFTING ALGORITHM

For the two clearance problem four configurations were established. This work provided for three clearances having eight possible model configurations. For the addition of the clearance at coordinate four, a total of 4 clearances, there would exist sixteen possible system configurations. Thus the number of possible configurations

of an "n" clearance system would be 2^n . A generalized method of selecting the proper configuration on the basis of the monitored parameters is therefore necessary.

The physical model will shift configurations only on the basis of two general parameters, displacement difference and force. Thus a binary number system facilitates the problem of "computer book-keeping". This is accomplished by assigning the value one to a clearance that is in contact and a zero to a clearance not in contact. For the unique choice of configurations established in chapter 2, Figures 2.4 and 2.5, the configuration matrix is developed as shown in Figure 3.1.

		CLEARANCES		
		#1	#2	#3
C	1	0	0	0
O	2	0	0	1
N	3	0	1	0
F	4	0	1	1
I	5	1	0	0
G	6	1	0	1
U	7	1	1	0
R	8	1	1	1
A				
T				
I				
O				
N				

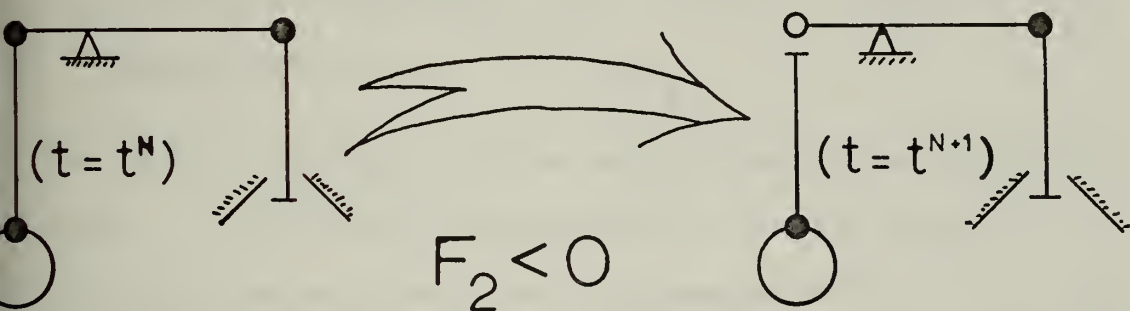
Figure 3.1 Configuration Matrix

The columns of this matrix are associated with clearances 1, 2, and 3 as indicated. The rows are associated with the configuration number that the physical system has taken. Thus to retrieve the correct configuration the entire row is taken as a binary number and converted to a decimal number. Since it is not

convenient to have a configuration numbered zero, a one was arbitrarily added.

For example, when no contact exists at all clearances, the binary number representing this configuration is 0 0 0. This binary number converts to decimal as 0. Upon adding the above mentioned value one, the decimal value becomes 1, which is the first row of the matrix and hence configuration I.

The binary number 1 1 1 represents contact at all clearances and converts to decimal 7. Adding one yields the value 8, the 8th row of the configuration matrix and hence configuration VIII.



CONFIGURATION VII

CONFIGURATION V

Step 1: VII \rightarrow 1 1 0
 Step 2: \downarrow
 1 \oplus 1 = 1 0
 Step 3: \downarrow
 1 0 0
 Step 4: 1 0 0 \rightarrow V

\oplus indicates binary addition

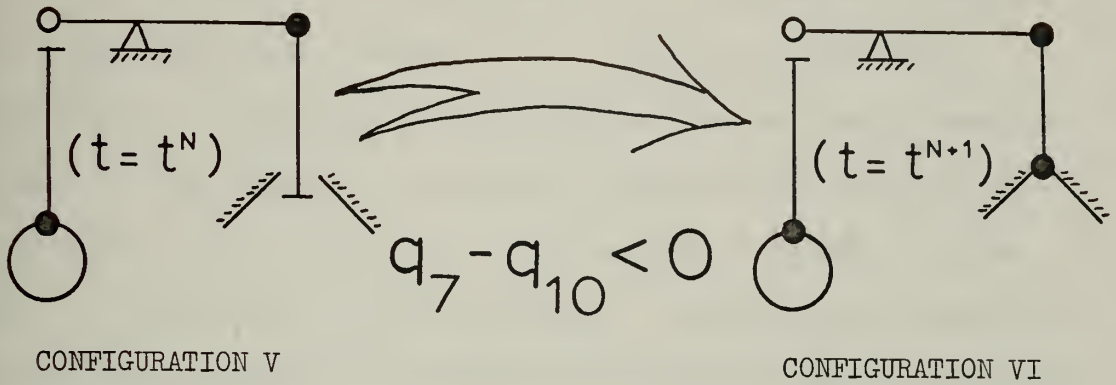
Figure 3.2 Shifting Example No. 1

The manner in which the program determines which configuration the model has shifted into is of primary concern. The developed algorithm is an extension of the above concepts, and best described

by way of examples. For the dynamic behavior pictured in Figure 3.2, the sequence of events is:

- (1) The model is in one of the eight given configurations, i.e. VII at time t^n .
- (2) The program calculates the required parameters for the next time step, t^{n+1} . Among these parameters are the monitored values. In this example the monitored value F_2 has changed during the time step, i.e. F_2 has gone from $F_2 > 0$ to $F_2 < 0$.
- (3) The negative force at coordinate two indicates the loss of contact at clearance 2. Thus the model now shifts to configuration V for time t^{n+1} , (Figure 2.6).

The four step shifting algorithm accomplishes this as outlined in Figure 3.2. The binary representation for configuration VII is retrieved from row seven in the configuration matrix, 1 1 0. Next the column of the binary number corresponding to the clearance that has been altered is removed. In this example, the element being the second digit of the binary number which correspond to the clearance 2 column. A one is then added, by binary addition, to this element. Carries are disregarded. The new second digit 0 is placed into the binary number. Finally the new resulting binary number is converted to decimal and yields the new configuration number, V.



Step 1: $V \rightarrow 1\ 0\ 0$
 Step 2: $\begin{array}{c} \downarrow \\ 0 \oplus 1 = 1 \\ \downarrow \end{array}$
 Step 3: $1\ 0\ 1$
 Step 4: $1\ 0\ 1 \rightarrow VI$

\oplus indicates binary addition

Figure 3.3 Shifting Example No. 2

The second example for the shifting algorithm is shown in

Figure 3.3. The sequence of events for the dynamic behavior is:

- (1) The model is in configuration V at time t^n .
- (2) The program calculates the required parameters at the next time step t^{n+1} . In this example, the monitored displacement difference value $q_7 - q_{10}$ has changed from $q_7 - q_{10} > 0$ to $q_7 - q_{10} < 0$.
- (3) The negative displacement difference at the coordinate pair 7 - 10 indicates that contact is established at clearance 3. Thus the model now shifts to configuration VI for time t^{n+1} .

The shifting algorithm accomplishes this as outlined in Figure 3.3. The binary representation for configuration V is retrieved from row five in the configuration matrix, 1 0 0. Next the column of the binary number corresponding to the clearance that has been altered is removed. In this example, the element being the third digit of the binary number which corresponds to the clearance 3 column. The new digit, 1 is placed into the binary number. Finally the new resulting binary number is converted to decimal and yields the new configuration number, VI.

E. METHOD OF CALCULATIONS

Newmark's β parameter method, (Ref. 4), forms the basis for the calculations. This one step numerical intergration procedure assumes the accelerations of the system coordinates are a linear function of time. Errors introduced by this assumption are given by $\omega_n \Delta T/6$, where ω_n is the highest natural frequency of interest. For small time steps this error is negligible. In this study the time step is taken as

$$\Delta T = 1 \times 10^{-6} \text{ seconds} \quad (3.27)$$

It will be shown, by example that the solution to the equations of motion takes a convenient form. Precalculated matrix expressions may generally be used, resulting in large savings of computer execution time. A basic familiarity with the formulation of Newmark's method (Appendix B) will prove helpful and aid in the understanding of the following discussion.

The development of the problem calculations will assume the model has been operating in configuration VI, contact at the cam and valve seat. The procedures are identical by extension, to the remaining seven configurations. Thus the problem unknown quantities have been calculated for time $t = t^n$. The object is to solve the governing equations for the next time increment $t = t^{n+1}$.

1. Solution to the Governing Equations

The full system matrices $[M]$, $[C]$, and $[K]$ have been developed in sections 3.B and 3.C. The governing equations of motion are expressed as the matrix equation 3.1 in section 3.A. Written in concise form the governing equations for the eight system coordinates are

$$[m_{ij}] \{\ddot{q}_i\}^{n+1} + [c_{ij}] \{\dot{q}_i\}^{n+1} + [k_{ij}] \{q_i\}^{n+1} = \{F_i\}^{n+1} \quad (3.28)$$

$$(i = 1, 2, \dots, 8; j = 1, 2, \dots, 8)$$

Exterior forces, F_1^{n+1} and F_7^{n+1} , are unknown. The remaining applied forces F_i^{n+1} ($i = 2, 3, \dots, 6, 8$) are equal to zero. Since the cam and follower are in contact, the known cam values at reference coordinate nine, \ddot{q}_9^{n+1} , \dot{q}_9^{n+1} , and q_9^{n+1} are equal to the corresponding follower values at system coordinate one, \ddot{q}_1^{n+1} , \dot{q}_1^{n+1} , and q_1^{n+1} . Similarly, the known values for the stationary valve seat at reference coordinate ten, \ddot{q}_{10}^{n+1} , \dot{q}_{10}^{n+1} and q_{10}^{n+1} are set equal to those corresponding values for the valve at system coordinate seven, \ddot{q}_7^{n+1} , \dot{q}_7^{n+1} , and q_7^{n+1} . Displacements

q_i^{n+1} ($i = 2, 3, \dots, 6, 8$) are unknown. Expanding and partitioning equations 3.28 to take advantage of the known values yields

$$\begin{array}{c}
 \left[\begin{array}{cccccc}
 m_{11} & m_{12} & \cdots & m_{16} & m_{17} & m_{18} \\
 m_{21} & m_{22} & \cdots & m_{26} & m_{27} & m_{28} \\
 \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\
 \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\
 m_{61} & m_{62} & \cdots & m_{66} & m_{67} & m_{68} \\
 m_{71} & m_{72} & \cdots & m_{76} & m_{77} & m_{78} \\
 m_{81} & m_{82} & \cdots & m_{86} & m_{87} & m_{88}
 \end{array} \right] \left\{ \begin{array}{c} \ddot{q}_1 \\ \ddot{q}_2 \\ \vdots \\ \vdots \\ \ddot{q}_6 \\ \ddot{q}_7 \\ \ddot{q}_8 \end{array} \right\}^{n+1} + \left[\begin{array}{cccccc}
 c_{11} & c_{12} & \cdots & c_{16} & c_{17} & c_{18} \\
 c_{21} & c_{22} & \cdots & c_{26} & c_{27} & c_{28} \\
 \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\
 \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\
 c_{61} & c_{62} & \cdots & c_{66} & c_{67} & c_{68} \\
 c_{71} & c_{72} & \cdots & c_{76} & c_{77} & c_{78} \\
 c_{81} & c_{82} & \cdots & c_{86} & c_{87} & c_{88}
 \end{array} \right] \left\{ \begin{array}{c} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \vdots \\ \dot{q}_6 \\ \dot{q}_7 \\ \dot{q}_8 \end{array} \right\}^{n+1} \\
 \\
 + \left[\begin{array}{cccccc}
 k_{11} & k_{12} & \cdots & k_{16} & k_{17} & k_{18} \\
 k_{21} & k_{22} & \cdots & k_{26} & k_{27} & k_{28} \\
 \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\
 \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\
 \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\
 k_{61} & k_{62} & \cdots & k_{66} & k_{67} & k_{68} \\
 k_{71} & k_{72} & \cdots & k_{76} & k_{77} & k_{78} \\
 k_{81} & k_{82} & \cdots & k_{86} & k_{87} & k_{88}
 \end{array} \right] \left\{ \begin{array}{c} q_1 \\ q_2 \\ \vdots \\ \vdots \\ q_6 \\ q_7 \\ q_8 \end{array} \right\}^{n+1} = \left\{ \begin{array}{c} F_1 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ F_7 \\ 0 \end{array} \right\}^{n+1} \tag{3.29}
 \end{array}$$

The next step creates two sets of equations. The first set, equations 3.30 and 3.31 consists of individual equations in which the displacement, velocity, and accelerations are known but the force is not. The second set, equation 3.32 consists of the matrix equations set equal to the zero force vector. In these equations all displacement, velocity, and acceleration quantities are unknown.

These equations of motion for the system are now written as:

$$F_1 = \begin{matrix} < m_{11} & \dots & m_{18} > \\ \left. \begin{matrix} \dot{q}_1 \\ \dots \\ \dot{q}_8 \end{matrix} \right\} < c_{11} & \dots & c_{18} > \\ < q_1 & \dots & q_8 > \end{matrix} + \begin{matrix} < k_{11} & \dots & k_{18} > \\ \left. \begin{matrix} \dot{q}_1 \\ \dots \\ \dot{q}_8 \end{matrix} \right\} < c_{11} & \dots & c_{18} > \\ < q_1 & \dots & q_8 > \end{matrix} \quad (3.30)$$

$$F_7 = \begin{matrix} < m_{71} & \dots & m_{78} > \\ \left. \begin{matrix} \dot{q}_1 \\ \dots \\ \dot{q}_8 \end{matrix} \right\} < c_{71} & \dots & c_{78} > \\ < q_1 & \dots & q_8 > \end{matrix} + \begin{matrix} < k_{71} & \dots & k_{78} > \\ \left. \begin{matrix} \dot{q}_1 \\ \dots \\ \dot{q}_8 \end{matrix} \right\} < c_{71} & \dots & c_{78} > \\ < q_1 & \dots & q_8 > \end{matrix} \quad (3.31)$$

$$\begin{matrix} \boxed{\begin{matrix} m_{22} & m_{23} & \dots & m_{26} & m_{28} \\ m_{32} & m_{33} & \dots & m_{36} & m_{38} \\ \cdot & \cdot & \dots & \cdot & \cdot \\ \cdot & \cdot & \dots & \cdot & \cdot \\ m_{62} & m_{63} & \dots & m_{66} & m_{68} \\ m_{82} & m_{83} & \dots & m_{86} & m_{88} \end{matrix}} & \left. \begin{matrix} \dot{q}_2 \\ \dot{q}_3 \\ \cdot \\ \cdot \\ \dot{q}_6 \\ \dot{q}_8 \end{matrix} \right\}^{n+1} & \boxed{\begin{matrix} c_{22} & c_{23} & \dots & c_{26} & c_{28} \\ c_{32} & c_{33} & \dots & c_{36} & c_{38} \\ \cdot & \cdot & \dots & \cdot & \cdot \\ \cdot & \cdot & \dots & \cdot & \cdot \\ c_{62} & c_{63} & \dots & c_{66} & c_{68} \\ c_{82} & c_{83} & \dots & c_{86} & c_{88} \end{matrix}} & + & \left. \begin{matrix} \dot{q}_2 \\ \dot{q}_3 \\ \cdot \\ \cdot \\ \dot{q}_6 \\ \dot{q}_8 \end{matrix} \right\}^{n+1} & \boxed{\begin{matrix} k_{22} & k_{23} & \dots & k_{26} & k_{28} \\ k_{32} & k_{33} & \dots & k_{36} & k_{38} \\ \cdot & \cdot & \dots & \cdot & \cdot \\ \cdot & \cdot & \dots & \cdot & \cdot \\ k_{62} & k_{63} & \dots & k_{66} & k_{68} \\ k_{82} & k_{83} & \dots & k_{86} & k_{88} \end{matrix}} & + & \left. \begin{matrix} q_2 \\ q_3 \\ \cdot \\ \cdot \\ q_6 \\ q_8 \end{matrix} \right\}^{n+1} \end{matrix} \quad (3.32)$$

$$+ \Delta P_1^{n+1} + \Delta P_2^{n+1} = \left. \begin{matrix} 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{matrix} \right\}$$

Where

$$\Delta P_1^{n+1} = \begin{pmatrix} m_{21} \\ m_{31} \\ \vdots \\ m_{61} \\ m_{81} \end{pmatrix} \ddot{q}_1^{n+1} + \begin{pmatrix} c_{21} \\ c_{31} \\ \vdots \\ c_{61} \\ c_{81} \end{pmatrix} \dot{q}_1^{n+1} + \begin{pmatrix} k_{21} \\ k_{31} \\ \vdots \\ k_{61} \\ k_{81} \end{pmatrix} q_1^{n+1} \quad (3.33)$$

and

$$\Delta P_2^{n+1} = \begin{pmatrix} m_{27} \\ m_{37} \\ \vdots \\ m_{67} \\ m_{87} \end{pmatrix} \ddot{q}_7^{n+1} + \begin{pmatrix} c_{27} \\ c_{37} \\ \vdots \\ c_{67} \\ c_{87} \end{pmatrix} \dot{q}_7^{n+1} + \begin{pmatrix} k_{27} \\ k_{37} \\ \vdots \\ k_{67} \\ k_{87} \end{pmatrix} q_7^{n+1} \quad (3.34)$$

The set of consistent system equations written in concise form are

$$F_1 = \sum_{i=1}^8 (m_{1i} \ddot{q}_i^{n+1} + c_{1i} \dot{q}_i^{n+1} + k_{1i} q_i^{n+1}) \quad (3.35)$$

$$F_7 = \sum_{i=1}^8 (m_{7i} \ddot{q}_i^{n+1} + c_{7i} \dot{q}_i^{n+1} + k_{7i} q_i^{n+1}) \quad (3.36)$$

and

$$[m_{ij}] \{\ddot{q}_i\}^{n+1} + [c_{ij}] \{\dot{q}_i\}^{n+1} + [k_{ij}] \{q_i\}^{n+1} + \sum_{j=1}^2 \Delta P_j^{n+1} = \{0_i\} \quad (3.37)$$

$$(i = 2, 3, \dots, 6, 8; j = 2, 3, \dots, 6, 8)$$

The next step is the solution of the matrix equation 3.37 by Newmark's method, for the unknown accelerations. By integration the velocities and displacements are written in terms of the accelerations as:

$$\{\dot{q}_i\}^{n+1} = \{\dot{q}_i\}^n + \frac{\Delta T}{2} \{\ddot{q}_i\}^n + \frac{\Delta T}{2} \{\ddot{q}_i\}^{n+1} \quad (3.38)$$

and

$$\{q_i\}^{n+1} = \{q_i\}^n + \Delta T \{\dot{q}_i\}^n + \frac{\Delta T^2}{3} \{\ddot{q}_i\}^n + \frac{\Delta T^2}{6} \{\ddot{q}_i\}^{n+1} \quad (3.39)$$

$$(i = 2, 3, \dots 6, 8)$$

Let

$$\{\alpha_i\}^n = \{\dot{q}_i\}^n + \frac{\Delta T}{2} \{\ddot{q}_i\}^n \quad (3.40)$$

and

$$\{\beta_i\}^n = \{q_i\}^n + \Delta T \{\dot{q}_i\}^n + \frac{\Delta T^2}{3} \{\ddot{q}_i\}^n \quad (3.41)$$

$$(i = 2, 3, \dots 6, 8)$$

Substituting for $\{\alpha\}^n$ and $\{\beta\}^n$ equations 3.38 and 3.39 become:

$$\{\dot{q}_i\}^{n+1} = \{\alpha_i\}^n + \frac{\Delta T}{2} \{\ddot{q}_i\}^{n+1} \quad (3.42)$$

$$\{q_i\}^{n+1} = \{\beta_i\}^n + \frac{\Delta T^2}{6} \{\ddot{q}_i\}^{n+1} \quad (3.43)$$

$$(i = 2, 3, \dots 6, 8)$$

Next substituting for $\{\dot{q}_i\}^{n+1}$ and $\{q_i\}^{n+1}$ into equations 3.37 and

factoring yields:

$$\begin{aligned}
 & \left([m_{ij}] + [c_{ij}] \frac{\Delta T}{2} + [k_{ij}] \frac{\Delta T^2}{6} \right) \{\ddot{q}_i\}^{n+1} \\
 & + [c_{ij}] \{\alpha_i\}^n + [k_{ij}] \{\beta_i\}^n + \sum_{k=1}^2 \Delta P_k^{n+1} = \{0_i\}
 \end{aligned} \quad (3.44)$$

(i = 2,3 .. 6,8; j = 2,3, .. 6,8)

Let

$$[BMIV_{ij}]^{-1} = [m_{ij}] + [c_{ij}] \frac{\Delta T}{2} + [k_{ij}] \frac{\Delta T^2}{6} \quad (3.45)$$

(i = 2,3, .. 6,8; j = 2,3, .. 6,8)

Substituting for $[BMIV]^{-1}$ in equation 3.44 and moving the knowns to the right side of the equation yields

$$[BMIV_{ij}]^{-1} \{\ddot{q}_i\}^{n+1} = - [c_{ij}] \{\alpha_i\}^n - [k_{ij}] \{\beta_i\}^n - \sum_{k=1}^2 \Delta P_k^{n+1} \quad (3.46)$$

(i = 2,3,..6,8 ; j = 2,3,..6,8)

Inverting the matrix $[BMIV]^{-1}$ yields the expression for the unknown accelerations

$$\{\ddot{q}_i\}^{n+1} = -[BMIV_{ij}][c_{ij}]\{\alpha_i\}^n - [BMIV_{ij}][k_{ij}]\{\beta_i\}^n - [BMIV_{ij}] \left(\sum_{k=1}^2 \Delta P_k^{n+1} \right) \quad (3.47)$$

(i = 2,3,..6,8; j = 2,3,..6,8)

In equation 3.47 the non-time function matrix multiplications operations are precalculated as

$$[DA_{ij}] = [BMIV_{ij}] [c_{ij}] \quad (3.48)$$

and

$$[DV_{ij}] = [BMIV_{ij}] [k_{ij}] \quad (3.49)$$

$$(i = 2,3,..6,8; j = 2,3,..6,8)$$

The system accelerations, except the initial known values, are now calculated by

$$\{\ddot{q}_i\}^{n+1} = - [DA_{ij}] - [DV_{ij}] - [BMIV_{ij}] \left(\sum_{k=1}^2 \Delta P_k^{n+1} \right) \quad (3.50)$$

$$(i = 2,3,\dots,6,8; j = 2,3,\dots,6,8)$$

The system velocities and displacements are calculated from

$$\{\dot{q}_i\}^{n+1} = \{\alpha_i\} + \frac{\Delta T}{2} \{\ddot{q}_i\}^{n+1} \quad (3.51)$$

$$\{q_i\}^{n+1} = \{\beta_i\} + \frac{\Delta T^2}{6} \{\ddot{q}_i\}^{n+1} \quad (3.52)$$

$$(i = 2,3,\dots,6,8; j = 2,3,\dots,6,8)$$

The remaining system unknowns, F_1 and F_7 may be directly calculated from equations 3.30 or 3.35 and 3.31 or 3.36. This completes the calculations for the required solution to the problem for this time step.

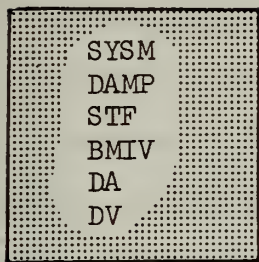
2. Precalculated Matrices

Returning to equation 3.32 it is noted that the mass, damping, and stiffness matrices are unique to configuration VI. Similarly from equation 3.30 and 3.31 the mass, damping, and stiffness column vectors are also unique to configuration VI. Thus these matrices need only be calculated once if stored as layers to the full system matrices. Figure 3.4 graphically indicates the matrix dimensions and elements for this and the remaining seven configurations. Additionally for the eight sets of precalculated matrices $[M]$, $[C]$, and $[K]$ the matrix variables $[BMIV]$, $[DA]$, and $[DV]$ given by equations 3.45, 3.48, and 3.49 may be precalculated for the standard ΔT and stored.

FULL SYSTEM
8 COORDINATE
MATRICES

REDUCED SYSTEM
7 COORDINATE
MATRICES

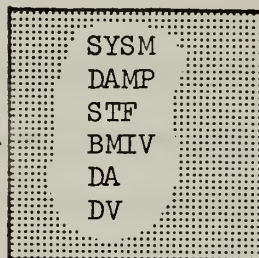
8x8



{ FIRST
EIGENVALUE
PROBLEM }

{ SECOND
EIGENVALUE
PROBLEM }

7x7

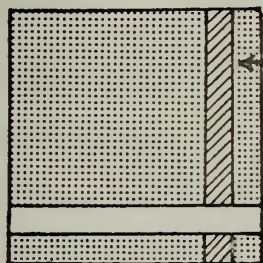


Ⓡ

CONFIGURATION

Ⓢ

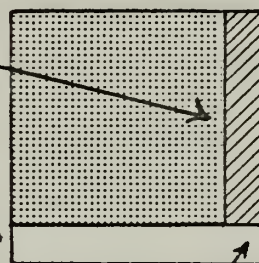
7x7



{ COLM
COLC
COLK }

{ COLM
COLC
COLK }

6x6

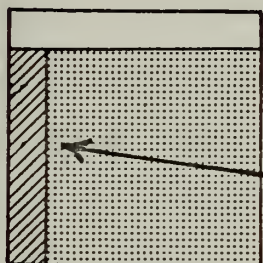


Ⓡ

CONFIGURATION

Ⓢ

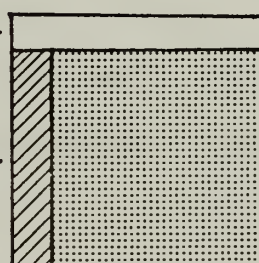
7x7



{ COLM
COLC
COLK }

{ COLM
COLC
COLK }

6x6

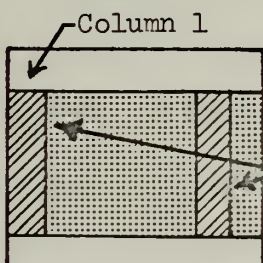


Ⓡ

CONFIGURATION

Ⓢ

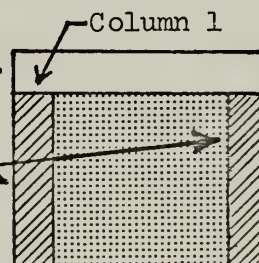
6x6



{ COLM
COLC
COLK }

{ COLM
COLC
COLK }

5x5



Ⓡ

CONFIGURATION

Ⓢ

Figure 3.4 MATRIX LAYERS

F. REFINEMENTS IN THE SOLUTION

1. Incremental Time

As seen in section 3.E, equation 3.27, the normal time step is established as 1×10^{-6} seconds. Generally the system configuration will change in a time step interval, rather than at either of the end points. When this occurs the present negative monitored value calculated at $t = t^{n+1}$ and the past stored monitored value that was calculated at $t = t^n$ are used for a linear interpolation to obtain the zero point, i.e. zero force or zero displacement difference.

The ΔT , now designated as $\Delta T'$, corresponding to the zero point is used to calculate the system values based on the old configurations.

If the system has more than one change in configurations for a given time step, the minimum $\Delta T'$ is used for the calculations, $\Delta T'_{\min}$.

The newly calculated critical value is then monitored, and if still negative, the time is again reduced by a small amount until a positive value of the monitored parameter is obtained. Figure 3.5 summarizes the above procedures for a negative force parameter.

2. Impact Forces

Anderson, in Chapter IV (Ref. 2), puts forth a good account of the study of impact forces. Numerous references are cited. Thus the historical background of impact forces in relation to this study will not be discussed here.

When the displacement difference values reduce to zero (or negative), contact at a clearance exists. This produces a step input to the velocity function and an impulse to the acceleration

function. Finite time steps are utilized to obtain solutions. Therefore one time step is sufficient to reach the step input value for the velocity curve, but two time intervals are required to account for the acceleration caused by the impulse, Figure 3.6.

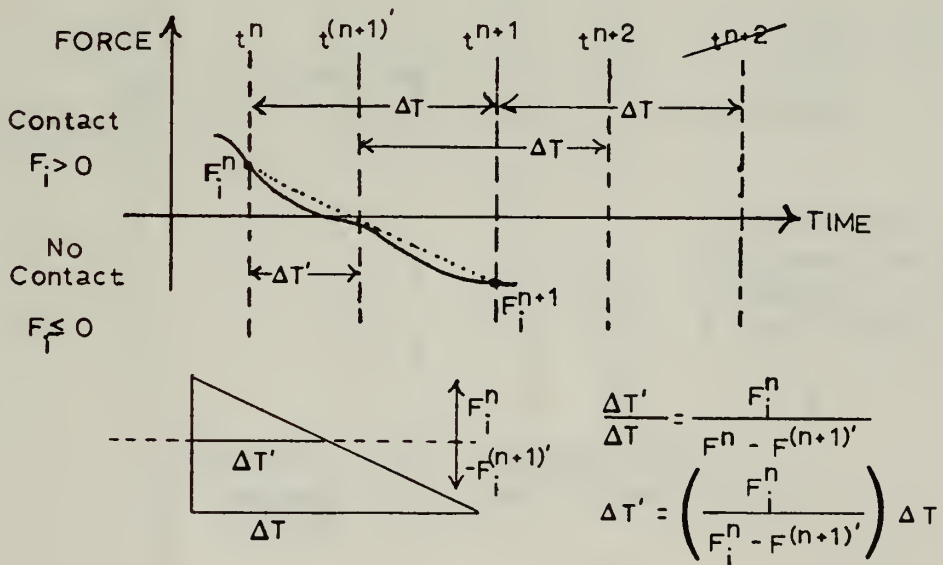
Thus the acceleration at time t^{n+1} is

$$\ddot{q}_i^{n+1} = (\dot{q}_i^{n+1} - \dot{q}_i^n) / \Delta T \quad (i = 1, 2, 7) \quad (3.53)$$

This acceleration is readily calculated for impact at the cam or valve seat ($i = 1$ or 7). However for contact at the rocker, neither \ddot{q}_2^{n+1} or \dot{q}_2^{n+1} is known. In order to solve the equation the assumption is made that

$$\dot{q}_2^{n+1} = \text{SCALE} * \dot{q}_2^n \quad (3.54)$$

The value for SCALE is arbitrarily chosen from 1.0 to 2.0, until a good system response results. Further study is suggested in this area.



then $\omega t^{(n+1)'} = \omega t^n + \omega \Delta T'$

is used to calculate system values.

Figure 3.5 Incremental Time $\Delta T'$

{ NO CONTACT
AT CAM }

{ IMPACT & CONTACT
AT CAM }

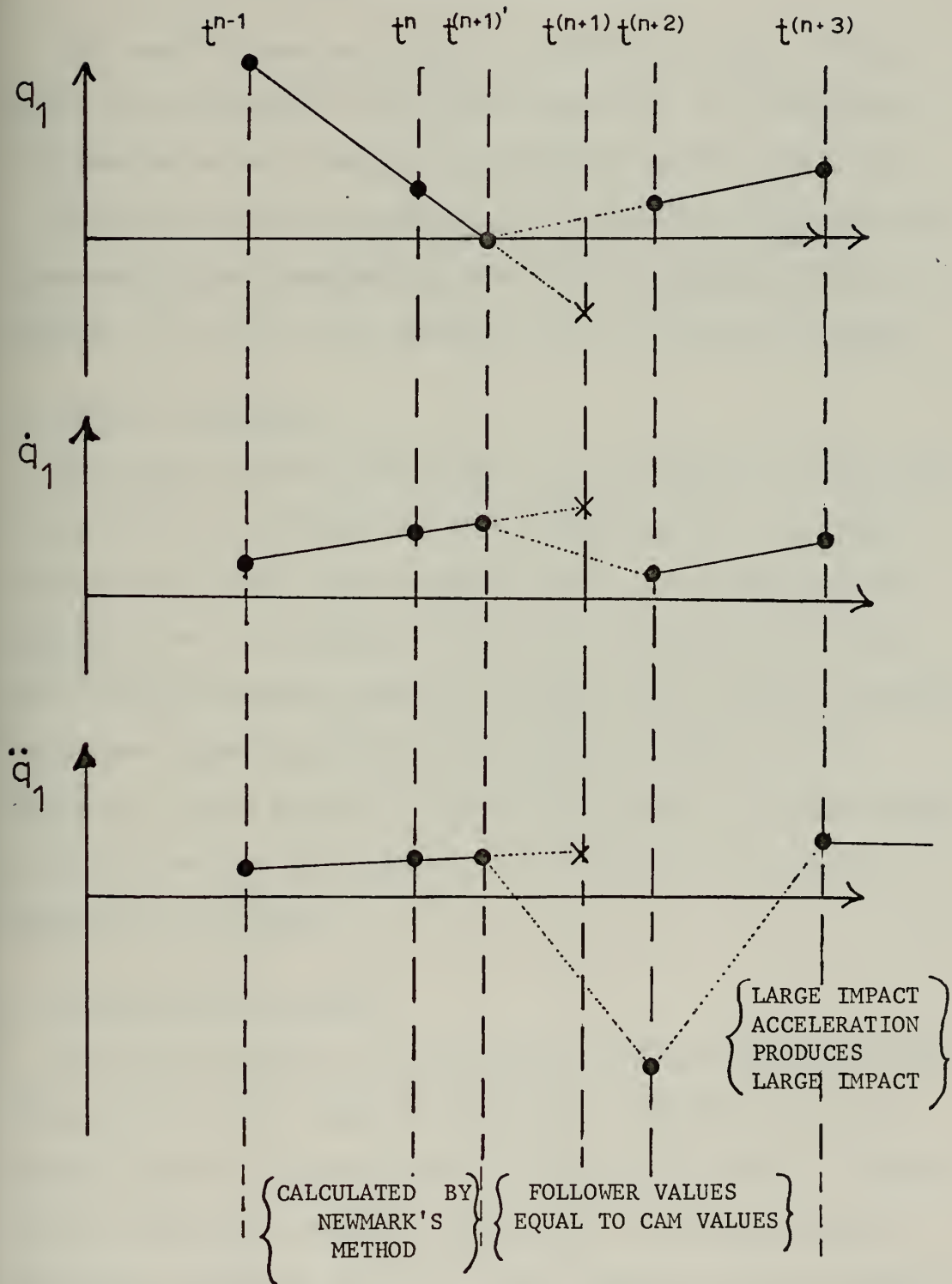


Figure 3.6 System Curves For Impact Considerations

IV. IBM-360 COMPUTER PROGRAM

This computer program is coded in FORTRAN IV, (Refs. 10-13), to run on an IBM-360/67, with OS/360 release 18. The simulation is modeled after basic concepts put forth by Anderson, (Ref. 2).

The original model program has been completely restructured and formulated to yield general procedures for the analysis of the dynamics of an elastic link mechanism with an internal clearance.

A. GENERAL DESCRIPTION

The program requires 234 K bytes of core storage for 850 degrees of cam rotation, including the CALCOMP graph buffer. The offline printer array (GRID) is not included in this core storage figure. Execution time is a function of the speed of rotation of the cam. For 11,000 RPM with 850 degrees, execution time is about 4.3 minutes. The primary output parameters fill arrays for SYSOUT = B and SYSOUT = C. The B queue is for transferring data, via punched cards, for use in the SDS-9300 computer. The C queue is for plotting graphs via the CALCOMP.

B. GENERALIZATION MATRICES

As noted in section 3.D, the number of system configurations is 2^n , where "n" is the number of clearances. Therefore, the most general computer programming was utilized, where possible. Several arrays of prestored computer "instructions" are either computer generated or formed by the user. These arrays set looping variables,

dynamically dimension work arrays, and establish program calculation flow indicators.

1. Matrix CONFIG

The 8 by 3 matrix CONFIG, Figure 4.1 was described in section 3.D. This integer array indicates the physical configurations that the model has taken based on the value of the monitored parameters. The matrix is formed by the program. The user must supply the value for the maximum number of configurations the system is permitted to take.

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Figure 4.1 Matrix CONFIG

2. Matrix IDM

The 8 by 1 column vector IDM indicates the dimension of the matrices that are used for calculations. This array is indicated in Figure 4.2 for the three clearance system. The program parameter ISYS is the matrix dimension of the equivalent system mass, stiffness, and damping matrices and for the calculated program matrix variables BMIV, DA, and DV. The dimensions of the various layers correspond to their configuration number and to the rows of IDM, i.e. when the system is in configuration I, layer one matrices have the dimension indicated in row one of IDM. This integer array is calculated by the program.

$$\begin{pmatrix} 8 \\ 7 \\ 7 \\ 6 \\ 7 \\ 6 \\ 6 \\ 5 \end{pmatrix}$$

Figure 4.2 Matrix IDM

3. Matrix LAYER

The 8 by 1 array LAYER, Figure 4.3, is an integer column vector supplied by the user. The elements of the vector consist of ones or threes. As the name implies, the layer of the mass, stiffness, damping, BMIV, DA, and DV matrices used in the calculations correspond to the configuration number the system has taken. The eigenvalue

problem for contact at the internal clearance (layer 1) or no contact (layer 3), determines the elements of LAYER. Configurations III, IV, VII, or VIII for contact at the rocker-pushrod, use the same eigenvalue problem for the reduced system, i.e. the damping matrix is calculated with layer 3 matrices. Configurations I, II, V, or VI for no contact at the rocker-pushrod, use the full system eigenvalue problem. The damping matrix is calculated using layer 1 matrices. Figure 4.3 shows these values of LAYER for the three clearance model with two eigenvalue problems.

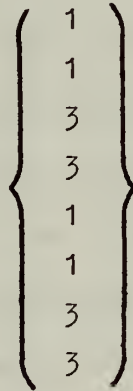


Figure 4.3 Matrix LAYER

4. Matrix IQ

The 8 by 8 integer array IQ, Figure 4.4 indicates the system coordinates for which the Newmark routine will calculate the unknown quantities i.e. displacements, velocities, accelerations, and forces for a given time step. The columns of the array correspond to the configuration number except column 1 and 3. These are additional columns used with configurations VI and VIII respectively. The array is user supplied.

1	1	1	1	2	2	2	2
2	2	2	2	3	3	3	3
3	3	3	3	4	4	4	4
4	4	4	4	5	5	5	5
5	5	5	5	6	6	6	6
6	6	6	6	7	8	7	0
7	8	7	0	8	0	0	0
8	0	0	0	0	0	0	0

Figure 4.4 Matrix IQ

5. Matrix IDP

This 8 by 3 integer array, Figure 4.5, consists of zero elements except where impact may be expected to occur at the external clearances. Here again the rows correspond to the configuration number and the columns correspond to the clearance number. The matrix IDP is used in the calculation of ΔP as a portion of Newmark's method. The array is calculated by the program.

0	0	0
0	0	7
0	0	0
0	0	7
1	0	0
1	0	7
1	0	0
1	0	7

Figure 4.5 Matrix IDP

6. Matrix ICORD

The 3 by 2 integer array ICORD, Figure 4.6 indicates the coordinate pairs of the internal and external clearances. The elements are stored row wise by pairs. The rows correspond to the clearance numbers 1, 2, and 3 respectively. The first column corresponds to the index of the system coordinate that has the greater positive displacement when no contact exists at the clearance in question. The second column is assigned to the remaining index of the clearance pair. The array ICORD is user generated. Figure 4.6 shows the matrix for the three clearance problem defined in chapter 2.

1	9
2	8
7	10

Figure 4.6 Matrix ICORD

7. Matrix IMPACT

The 3 by 1 column vector IMPACT is program generated. The integer elements are continuously modified during program execution. The rows correspond to the clearance number pairs. Values 1, 2, or 7, of the system coordinates are stored when impact occurs for a given time step at a corresponding clearance.

C. DYNAMIC DIMENSIONING

The principle of dynamic dimensioning is widely used in advanced programming and often misunderstood. Thus a short example of its concepts as applied to this work is presented here. Reference to Figures 4.7 and 4.8 will aid in the understanding of the concepts.

Subroutine: THESIS

BKSTEP

DEL

DIMENSION: (8,8,8)

(8,8,1)

(8,8)

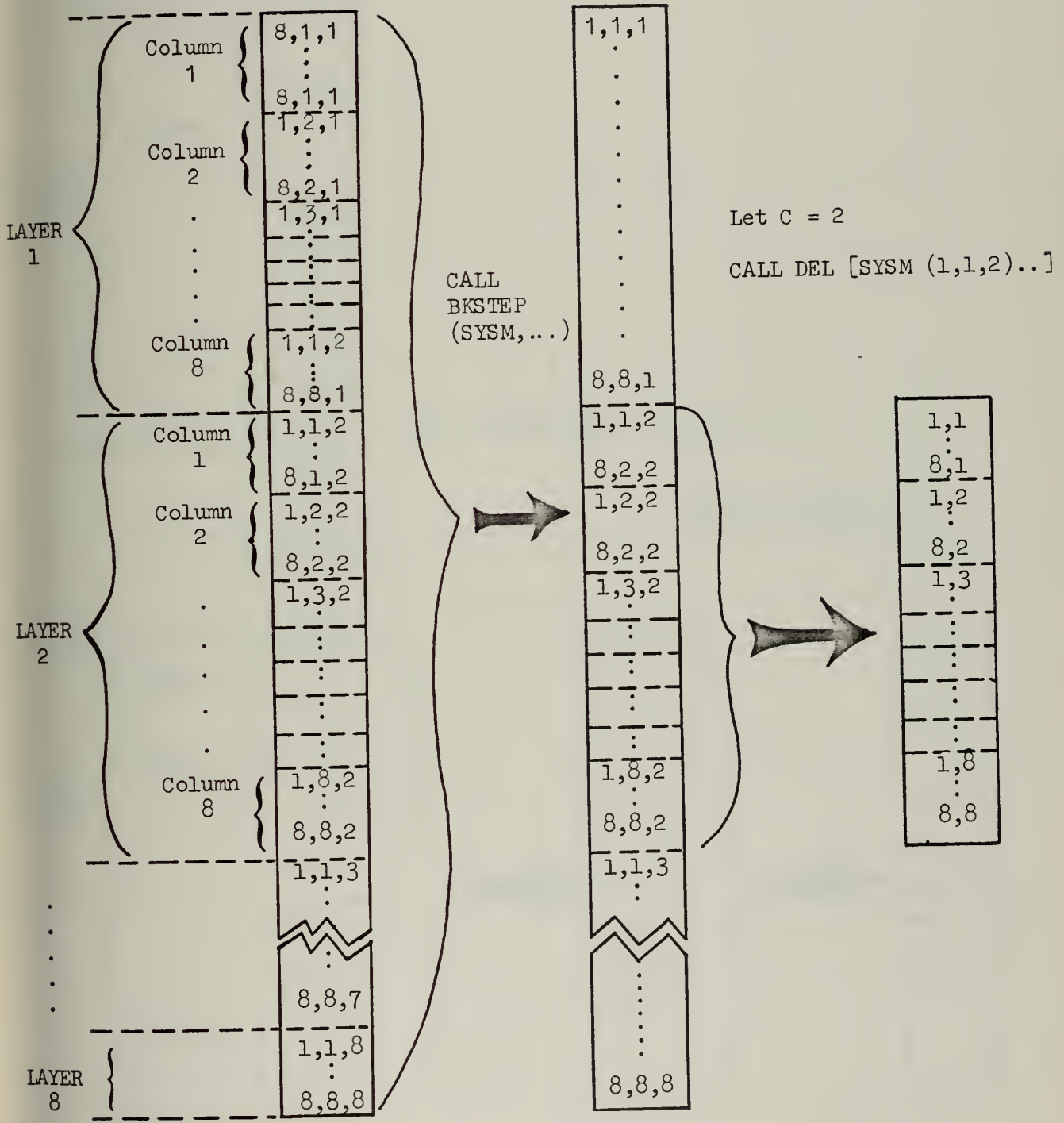


Figure 4.7 Dynamic Dimensioning of Array SYSM

Subroutine:	<u>THESIS</u>	<u>BKSTEP</u>	<u>DEL</u>
DIMENSION:	(8,8,9)	(8,8)	(8,1)

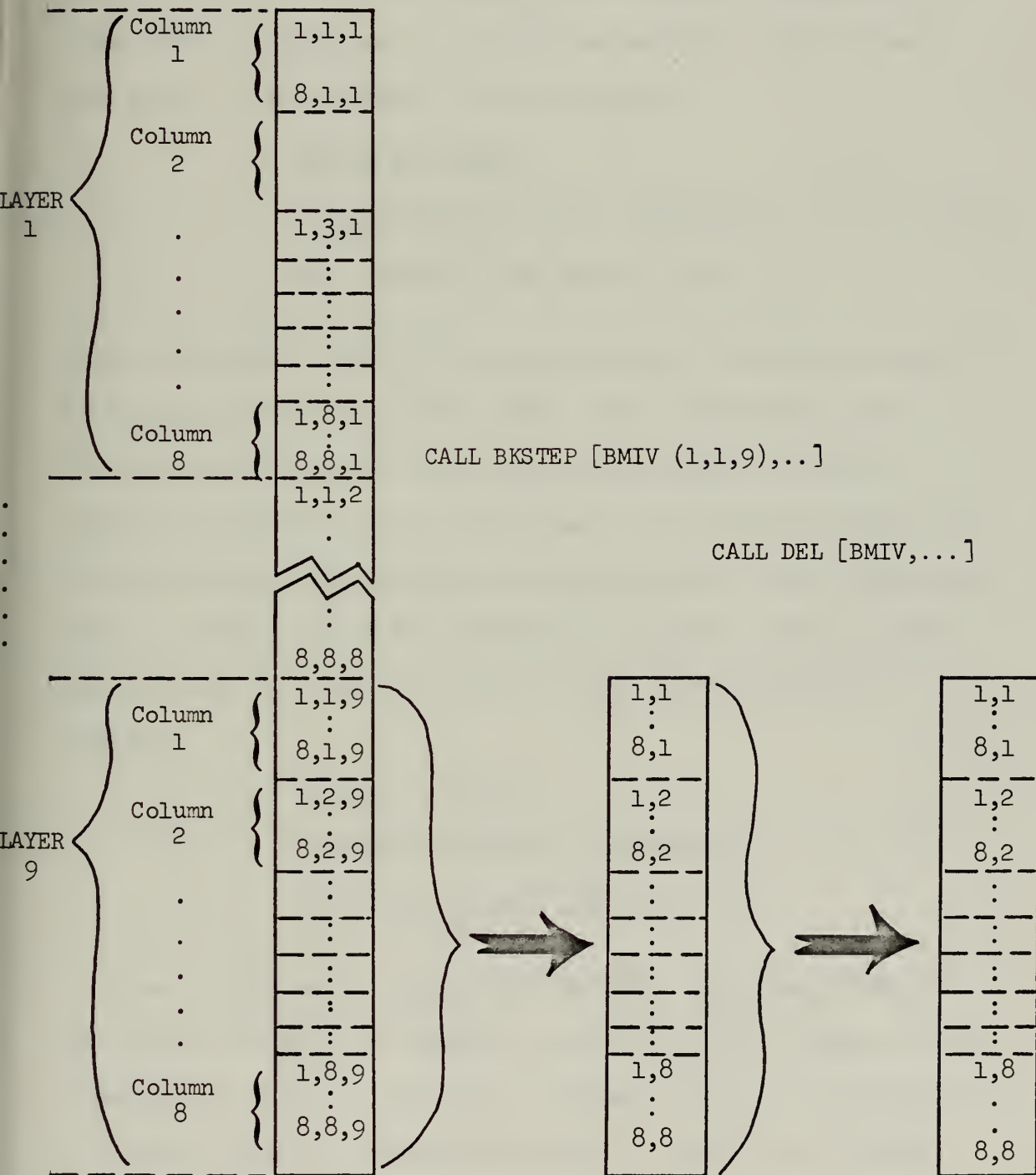


Figure 4.8 Dynamic Dimensioning of Array BMIV

In subroutine THESIS the mass array SYSM and the calculation array BMIV are exactly dimensioned indicating the total number of elements which are stored by columns in the computer. When subroutine BKSTEP is called, the entire array SYSM is passed. However the array BMIV is initialized in the call as the first value of the last layer, in this example. This is shown as:

```
Subroutine THESIS
DIMENSION SYSM (8,8,8), BMIV(8,8,9)
CALL BKSTEP (SYSM, BMIV(1,1,9), ...)
```

Thus in subroutine BKSTEP, the array SYSM must be dimensioned with the exact number of rows and columns. Since the computer has already provided for the total space of the array, the third dimension is dynamic and if entered may be any positive number equal to or less than the originally assigned dimension. Since the entire array is utilized the third dimension is retained. For the array BMIV only the last layer is used, thus the third dimension is dropped.

```
Subroutine BKSTEP
DIMENSION SYSM(8,8,1), BMIV(8,8)
CALL DEL (SYSM(1,1,C), BMIV, ...)
```

When subroutine DEL is called from BKSTEP the array, SYSM is now initialized with the variable C, ($C = 1, 2, \dots, 8$). BMIV, now only considered an 8 by 8 array, is not initialized here. The dimensioning in subroutine DEL is accomplished similar to that done in BKSTEP.

The array SYSM must again be dimensioned as above with the third dimensioning parameter being dynamic. The array BMIV is used as a double dimensioned array in subroutine DEL. By arguments similar to those above, the columns of the array may now also be dynamically dimensioned as:

```
Subroutine DEL  
DIMENSION SYSM(8,8), BMIV(8,1)
```

Figure 4.7 for array SYSM and Figure 4.8 for array BMIV graphically indicate the manner in which these arrays are passed from the subroutines, THESIS-BKSTEP-DEL.

D. PROGRAM STRUCTURE

The main program only sets up the plotting arrays and then calls subroutine THESIS, the main routine that controls the program execution. The calling order is presented in Figure 4.9. The CALCOMP graphing subroutines indicated by an asterisk are supplied by the computer center.

E. SUBROUTINES

The function of each of the subroutines and a description of their execution are included in this section. The concept of execution of object modules stored on disk was used to a large extent. This accounts for the large number of subroutines and their limited content. Flow graphs for the majority of the routines are collected and presented at the end of this section. Each subroutine is listed in its entirety in the IBM-360 Computer Program Listing chapter.

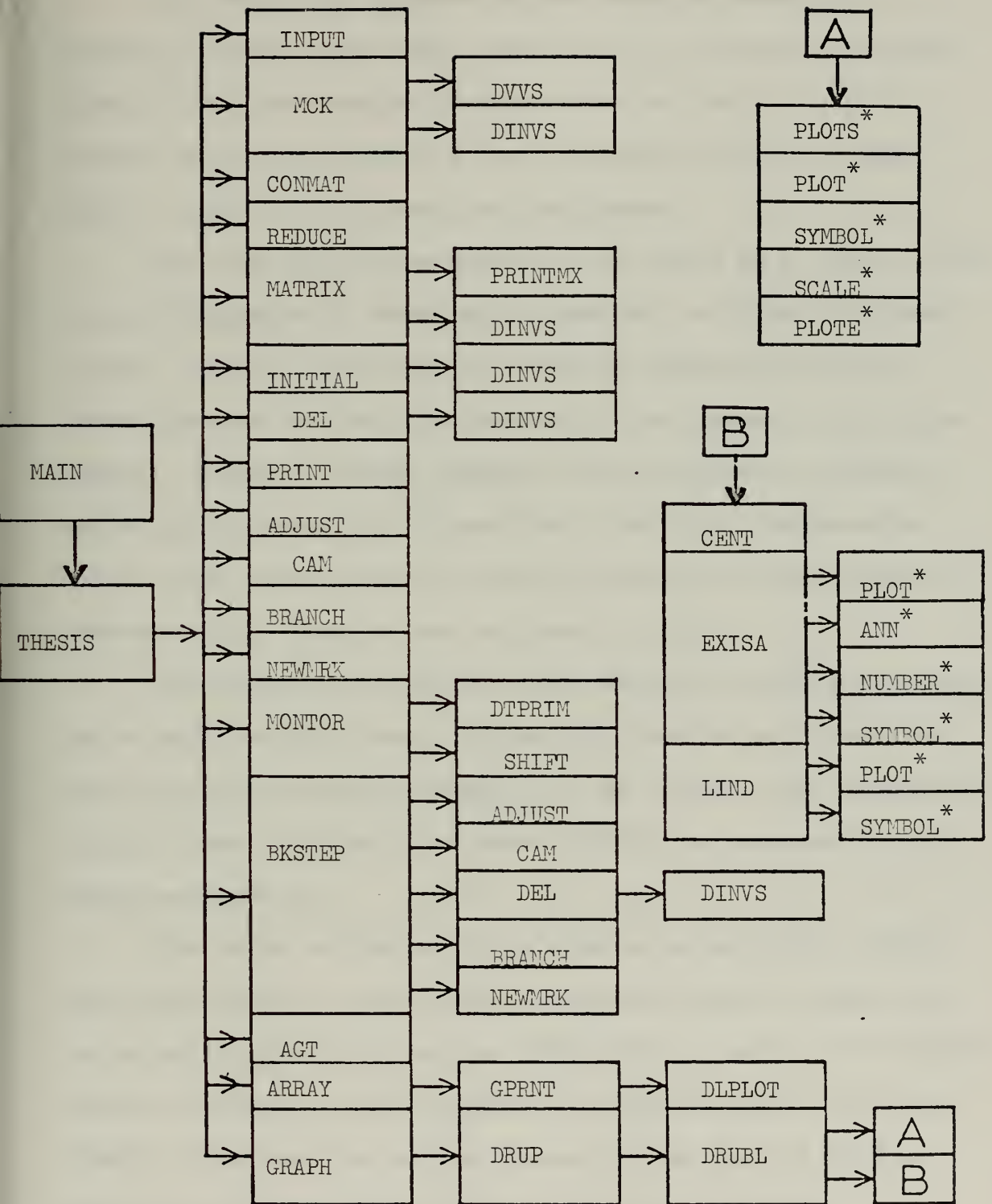


Figure 4.9 IBM-360 Computer Program Structure

1. Subroutine THESIS

This subroutine controls program execution, establishes calculation arrays (MCK, CONMAT, MATRIX, DEL), initialing parameters (INPUT, INITAL) and contains the major time step loop, $\Delta T = 10^{-6}$ seconds. As seen in Figure 4.9, most subroutines are called from THESIS. Figure 4.10 indicates the flow diagram.

The time parameter ωt advances by an amount $\omega \Delta T$. The previous values of displacement, velocity, acceleration, and force are stored (ADJUST). The new cam values are calculated (CAM) and the known system parameter are initialed according to the present configuration (BRANCH). Newmark's method (NEWMARK) is then employed to calculate the remaining unknown system quantities. The system configuration value is next stored prior to checking the newly calculated quantities (MONTOR) for any change in the configuration number.

The system is now checked to see whether an impact has occurred and at what coordinate index. If the configuration has changed, the quantities are recalculated (BKSTEP) for the time that the configuration changed. Then force values are reset (FIXFOR) to coorespond to the system configuration.

The amount of degrees of cam rotation is calculated. If the next whole degree has been reached the system values are output for use as the graphics (AGT) and draw (ARRAY) data. Next if the required number of degrees have been evaluated the desired graphs are plotted (GRAPH). Otherwise, the program branches to the start of the time step loop (statement number 360) and the cycle is repeated.

2. Subroutine BKSTEP

This subroutine is similar to subroutine THESIS in that all program variables are recalculated to correspond to the time at which the system configuration changed. Once this routine is entered only the parameter corresponding to the configuration shift is monitored until a positive value is calculated. Figure 4.11 indicates the flow diagram.

3. Subroutine MCK, DVVS, and DINVS

This subroutine originally coded by Anderson and Winfrey, was modified to account for the additional eigenvalue problem. The sizes and material properties of the system members are assigned. The element of the full and reduced system mass and stiffness are assigned to their respective arrays, SYSM and STF. The damping ratio, (ZTA) is set on the basis of the fundamental frequency of the system being considered. The eigenvalue problems are solved by subroutine DVVS. Subroutine DINVS is called to obtain matrix inverses..

4. Newmark's Method Subroutines

Subroutines NEWMRK and DEL are the primary program calculation routines, and considered together comprise Newmark's method. The flow diagram for subroutine DEL is seen in Figure 4.12 and for subroutine NEWMRK, Figure 4.13.

(a) Subroutine DEL

Subroutine DEL calculates the eight layers of the matrix arrays BMIV, DA, and DV, based on the full time step ΔT . These matrices are precalculated and stored. When DEL is called from subroutine BKSTEP, an additional layer of the above matrices is calculated as required, with a partial time step, $\Delta T_{\min} < 10^{-6}$ seconds.

(b) Subroutine NEWMRK

Subroutine NEWMRK calculates the coordinate accelerations by Newmark's method. The theory of this method is presented in appendix B. Its relation to this study is presented in section 3.E. After the system accelerations are known, the displacements and velocities can be calculated. Finally the forces are calculated at clearance coordinates 1, 2, and 7.

5. Subroutine for Shifting Configurations

The two subroutines that provide the program the ability to shift from one configuration to another, based on the values of the monitored parameters, are MONTOR and SHIFT. Figure 4.14 indicates the flow diagram for subroutine MONTOR. The subroutine that determines the amount by which the time step should be reduced, after a change in the configuration has been indicated, is DTPRIM. The flow diagram for this subroutine is seen in Figure 4.16.

(a) Subroutine MONTOR

Subroutine MONTOR checks the forces and/or the displacement differences at the three clearances to determine if any has become negative. As indicated by the shifting flow diagram, Figure 2.6, only three quantities must be monitored at each clearance, either forces and/or displacements differences. When at least one of the values becomes negative, subroutine DTPRIM is called.

(b) Subroutine DTPRIM

Subroutine DTPRIM linearly interpolates for a best estimate of the time the monitored value became negative, i.e. when the system changed to a different configuration. The calculated time

is stored in array DTP. If more than one change to the system configuration occurs in a single time step, the first change is chosen, i.e. the minimum value among the elements of the array DTP. The clearance index number that causes the system to change is then sent to subroutine SHIFT.

(c) Subroutine SHIFT

Subroutine SHIFT, described in section 3.D returns the new configuration based on the parameters sent by MONTOR.

6. Subroutine INPUT

This subroutine reads as data the formal parameters that are frequently varied. Figure 4.17 presents the flow diagram for this subroutine. These parameters include revolutions per minute (RPM) of the cam, clearance indicators (CLEAR, PIN 2&8) printing indicators (MPRINT, WIDEQ, KPRINT, NCOUNT, IPRINT), output data indicators (IDRAW, AGTOUT, CARDS, TAPE IPLOT) and the tipe step (DELT). Graph titles are read into the arrays XNAME, YNAME, LABEL, and TITLE. The array TITLE is modified to indicate programs without a valve seat. The array LABEL is modified to indicate a two clearance problem. The lengths of the member elements are assigned (REALTH, AGTLTH) and the lever arm of the rocker (ROCKER) is calculated.

7. Subroutine CONMAT

This subroutine establishes two of the program generalization arrays, CONFIG and IDP. Figure 4.18 shows the flow diagram. The routine is called only once during the program execution. The coding for both arrays may be easily expanded to provide for problems with more than three clearances.

8. Subroutine INITAL

This subroutine initiates the starting values of displacement, velocity, acceleration, and force for the system coordinates. The starting configuration VII, requires that layer seven matrices be used for all calculations. Subroutine DINVS is called to establish the inverse stiffness matrix necessary for the displacement calculations. The program variables for the follower are set equal to the cam values and those for the left rocker arm are set equal to the pushrod values by calling subroutine ADJUST.

9. Subroutine CAM

This short subroutine calculates the displacement, velocity, and acceleration at the cam (coordinate 9) for the given ωt . The flow diagram is presented as Figure 4.15. The displacement is given by the equation:

$$q_9 = .5 \sin \omega t + XZERO$$

This routine is called at least once, every time increment from subroutine THESIS. Additionally cam is called by subroutine BKSTEP for intermediate values of ωt .

10. Subroutine BRANCH

This subroutine is called prior to making calculations for a new time step. Figure 4.19 indicates the flow diagram. The eight sections of the routine initialize the system parameters corresponding to its present configuration. Additionally the dimension parameter ISYS is set from the array IDIM. A correction of the above established acceleration is accomplished when the array IMPACT indicates that a

clearance displacement difference is zero. The adjustment of the acceleration produces a corresponding increase in the later calculated force.

11. Subroutine ADJUST

This short subroutine equivalences the displacement, velocity, and acceleration quantities at a system coordinate to those at another system coordinate. For example, when contact exists between cam and follower the above named quantities for coordinate one are set equal to those for coordinate nine. Figure 4.20 presents the flow diagram.

12. Subroutine FIXFOR

This subroutine is primarily used to zero the calculated forces on the basis of the system configuration. This is necessary to eliminate small negative or small positive forces from being plotted. This routine has no effect on the program calculations. Its flow diagram is seen as Figure 4.21.

13. Output Subroutines

(a) Subroutine AGT

The subroutine AGT outputs data on punched cards or magnetic tape for use in the SDS-9300 program calculations of the graphic display. The flow diagram is indicated in Figure 4.22. This subroutine first adjusts the system displacement coordinates by an amount XZERO so as to present the initial position of the graphics display as horizontal. In addition to the displacement coordinates, reference values are sent for the cam rotation and the valve seat position.

(b) Subroutines ARRAY, GRAPH, GPRINT, OLPLOT

The subroutine ARRAY is called for the first value of each whole degree and establishes the array FPLOT, which is used for the plotting of graphs on the CALCOMP or offline printer. Figure 4.23 presents the flow diagram. Subroutine GRAPH is the last subroutine called by THESIS after the desired number of degrees has been evaluated. This subroutine, as seen in the flow diagram, Figure 4.24, organizes the calls to the offline printer subroutine OLPLOT for IPLOT equal to YES and/or the CALCOMP subroutine GPRINT for IDRAW equal to YES. The flow diagram for subroutine GPRINT is indicated in Figure 4.25. Subroutine OLPLOT is a modified version of the Computer Center provided routine UTPLOT. One of the modifications provides for an expansion of the graph output to a maximum of six pages. The number of pages desired is assigned to the program parameter, IP, by the user. Additionally the users must adjust the dimensions of the main program array, GRID to conform with the desired pages. GRID must be dimensioned as LINES by 81 where the dimension parameter LINES is:

$$\text{LINES} = (80)(\text{IP}) + 1$$

(c) Subroutine DRU--

The many subroutines beginning with DRU-- were written for plotting graphs on the CALCOMP plotter by Winfrey. This package also contains computer center provided routines, indicated by an asterick in Figure 4.9.

(d) Subroutine LIST (DEBUG)

Subroutine LIST outputs the numerical formal parameters on the offline printer for either "debugging" purposes or as reference values when viewing the graphs/graphics.

(e) Subroutine PRNTMX

Subroutine PRNTMX outputs all the system matrices,
when the program parameter MPRINT is equal to YES.

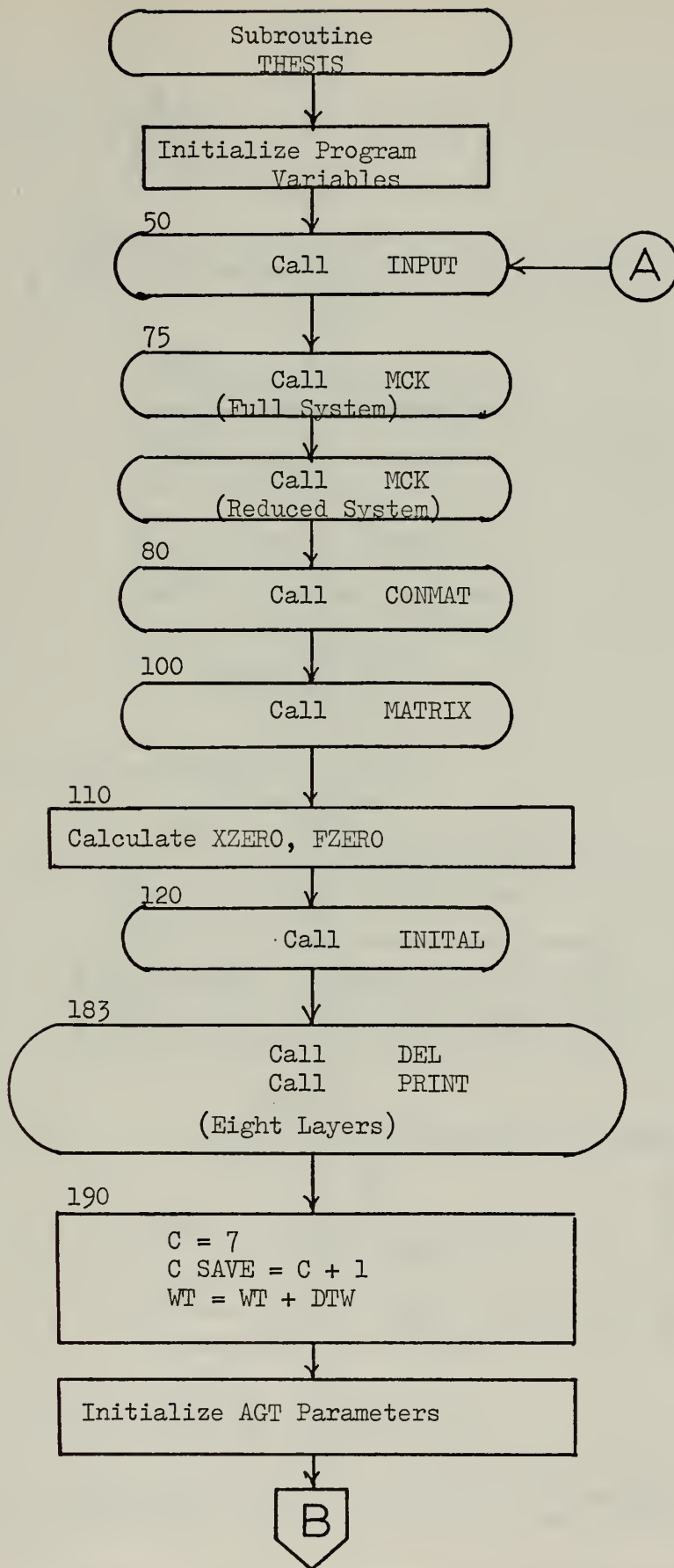


Figure 4.10 Subroutine THESIS Flow Diagram

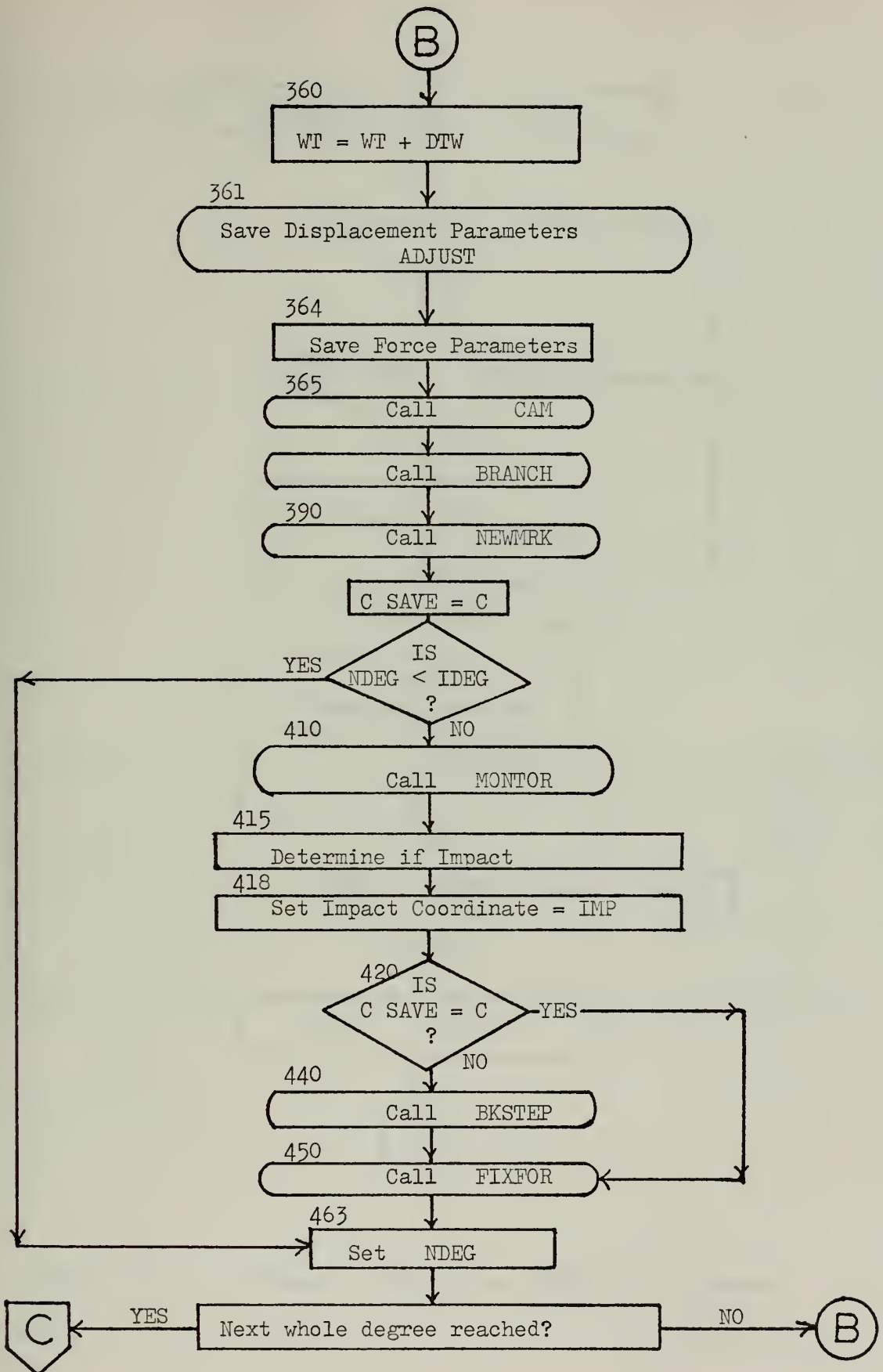


Figure 4.10 Continued - Subroutine THESIS Flow Diagram

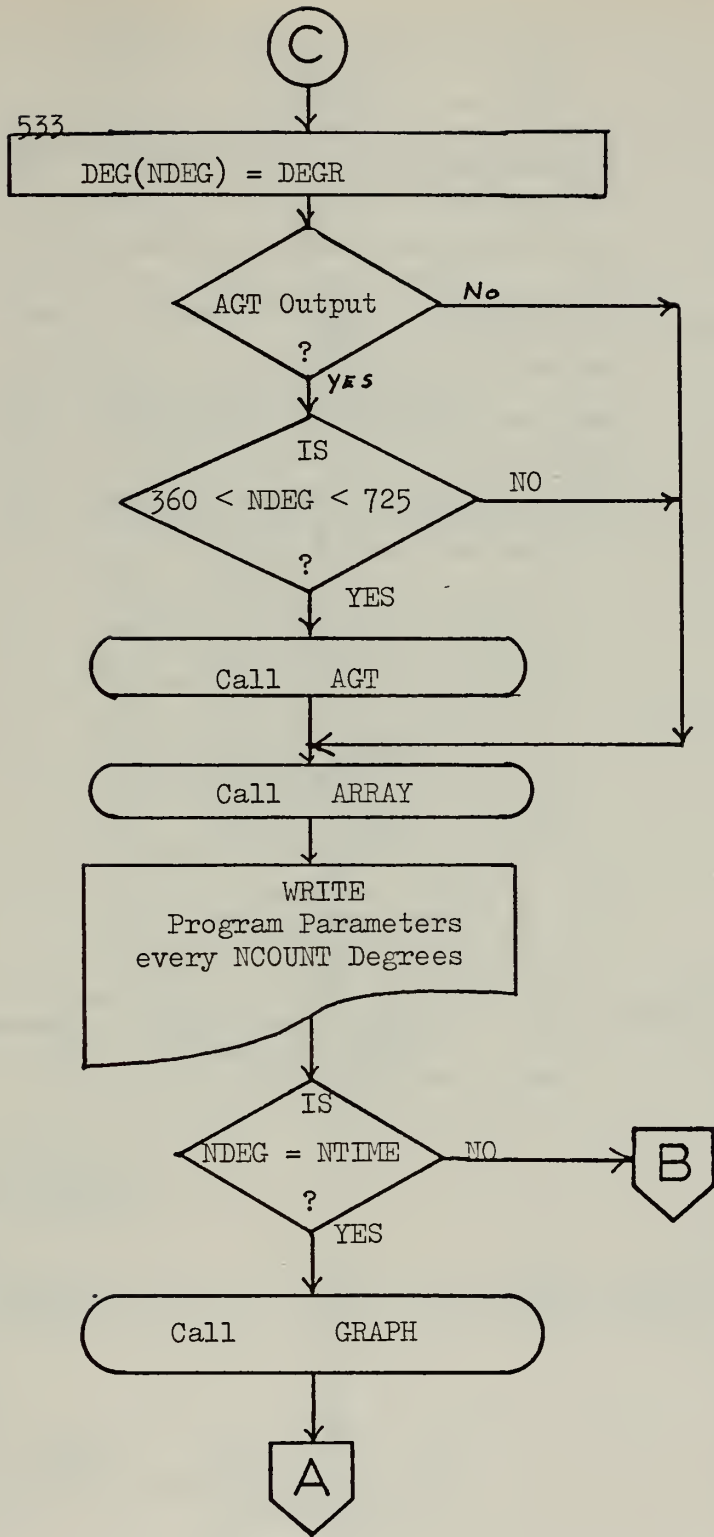


Figure 4.10 Continued - Subroutine THESIS Flow Diagram

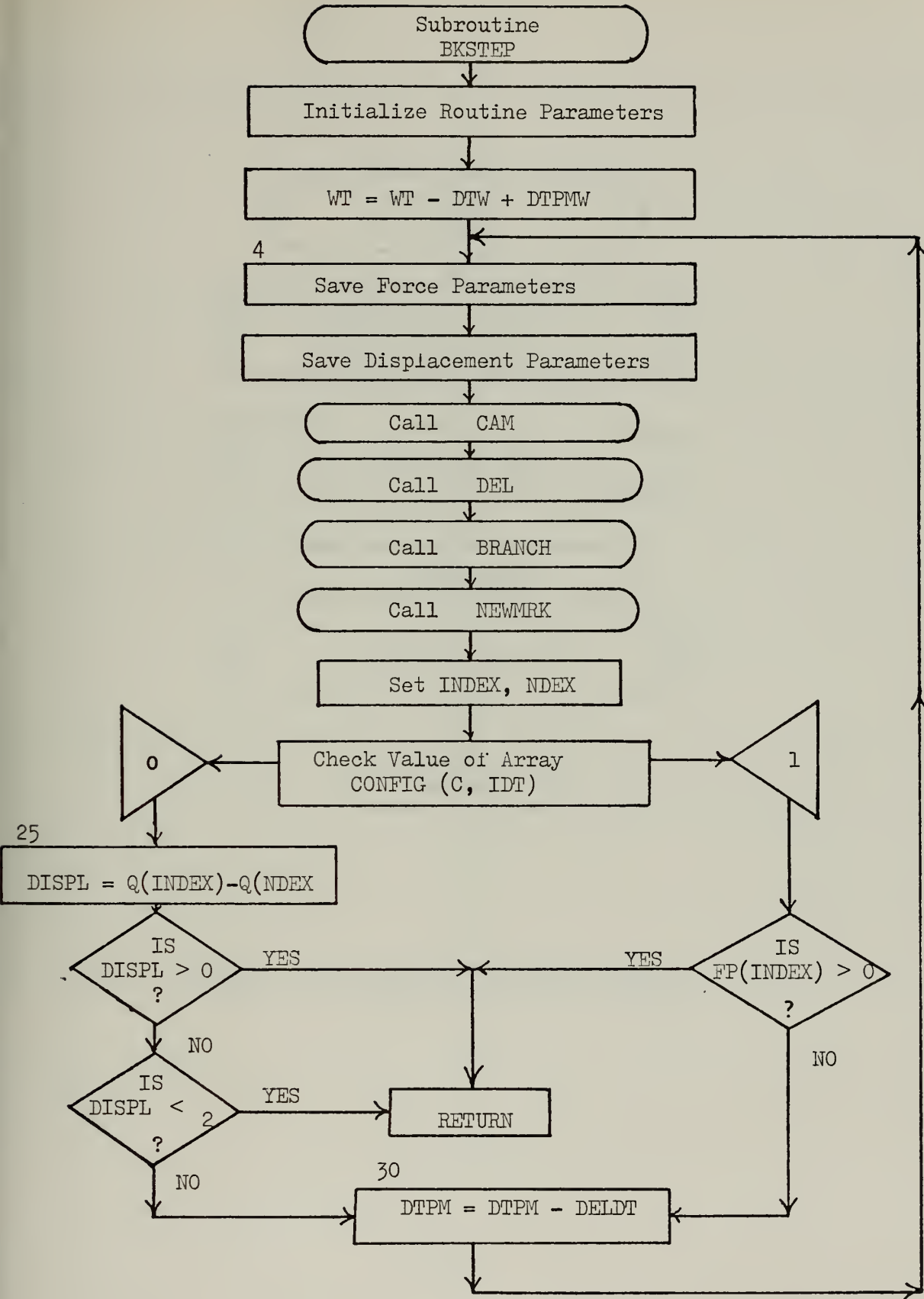


Figure 4.11 Subroutine BKSTEP Flow Diagram

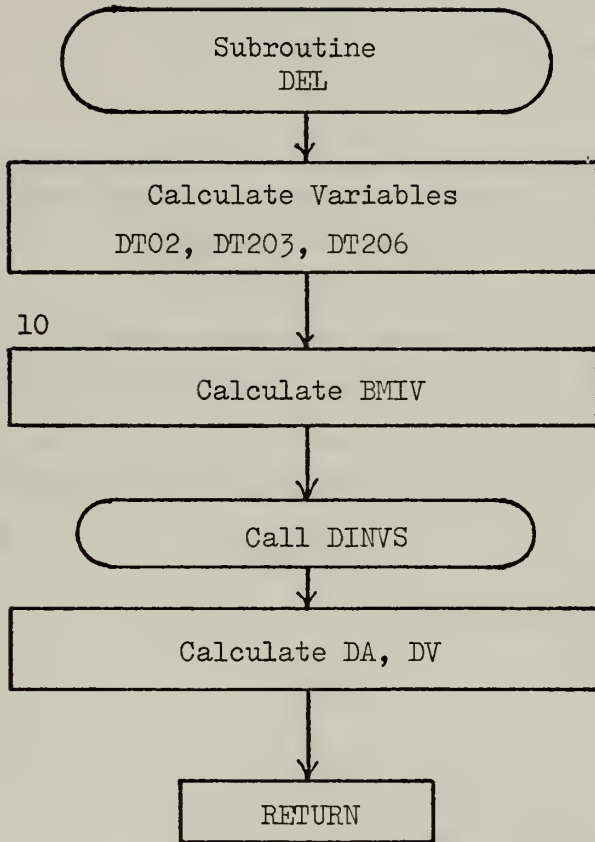


Figure 4.12 Subroutine DEL Flow Diagram

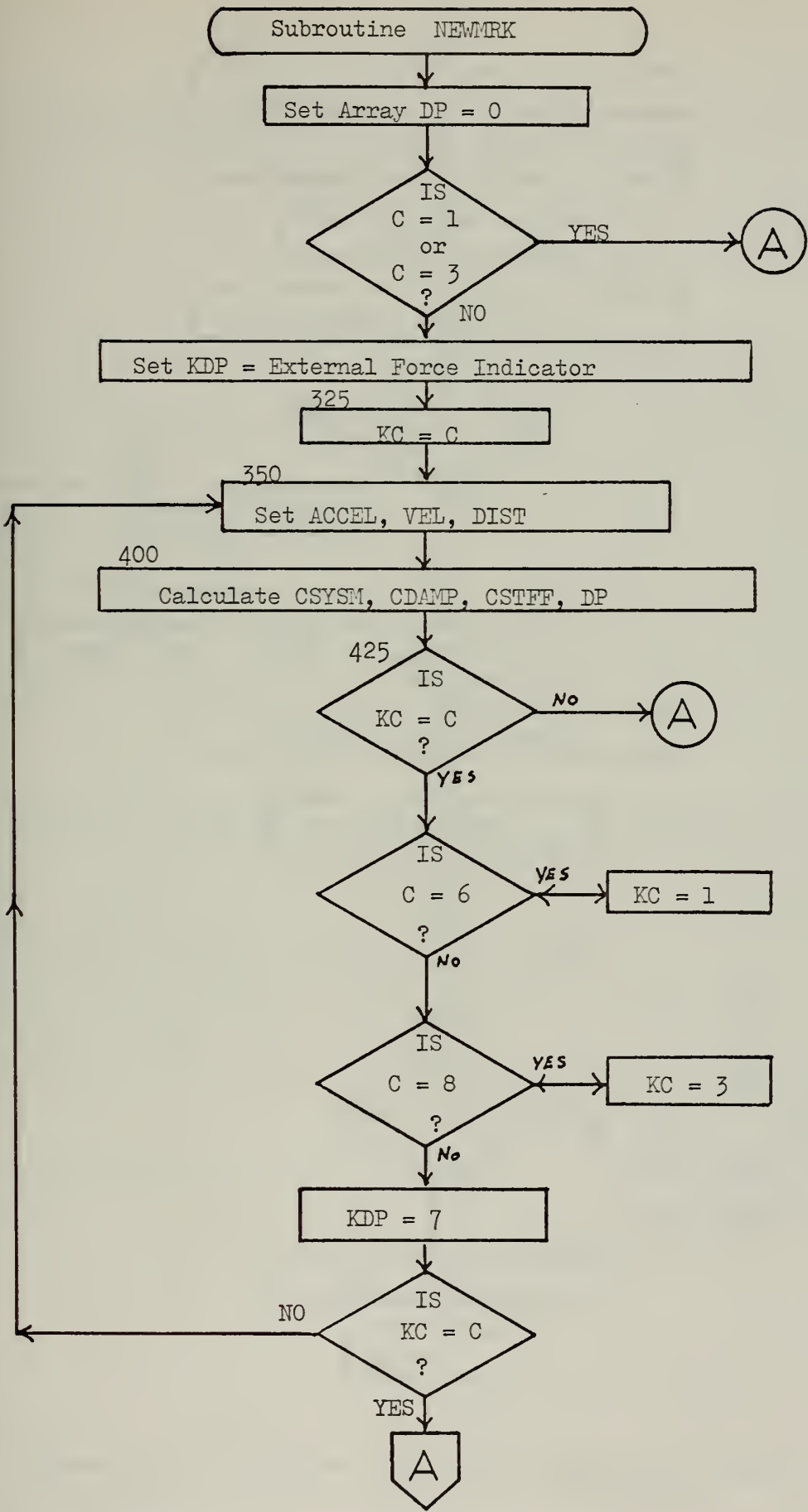


Figure 4.13-1 Subroutine NEWMRK Flow Diagram
83

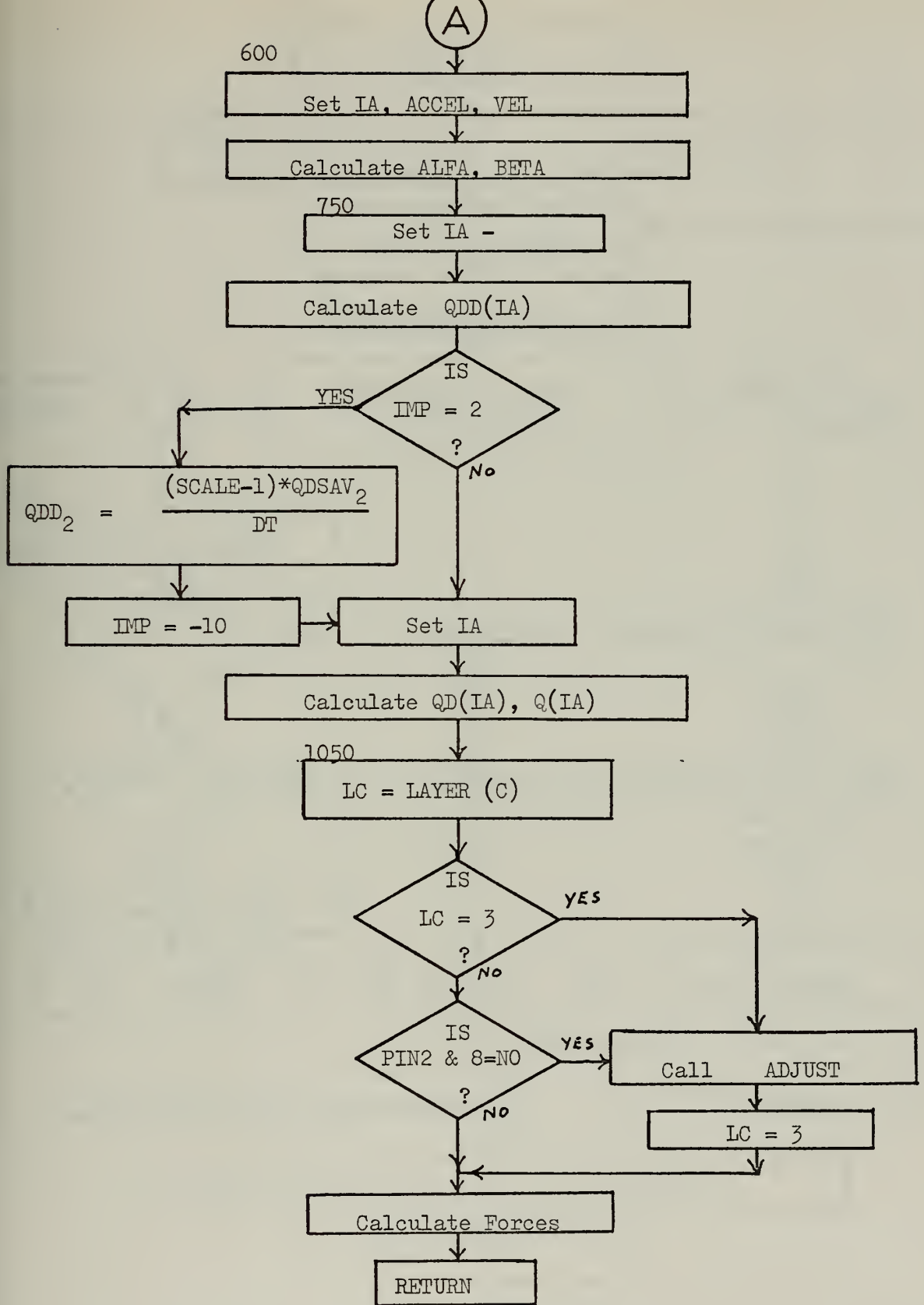


Figure 4.13-2 Continued - Subroutine NEWMRK Flow Diagram

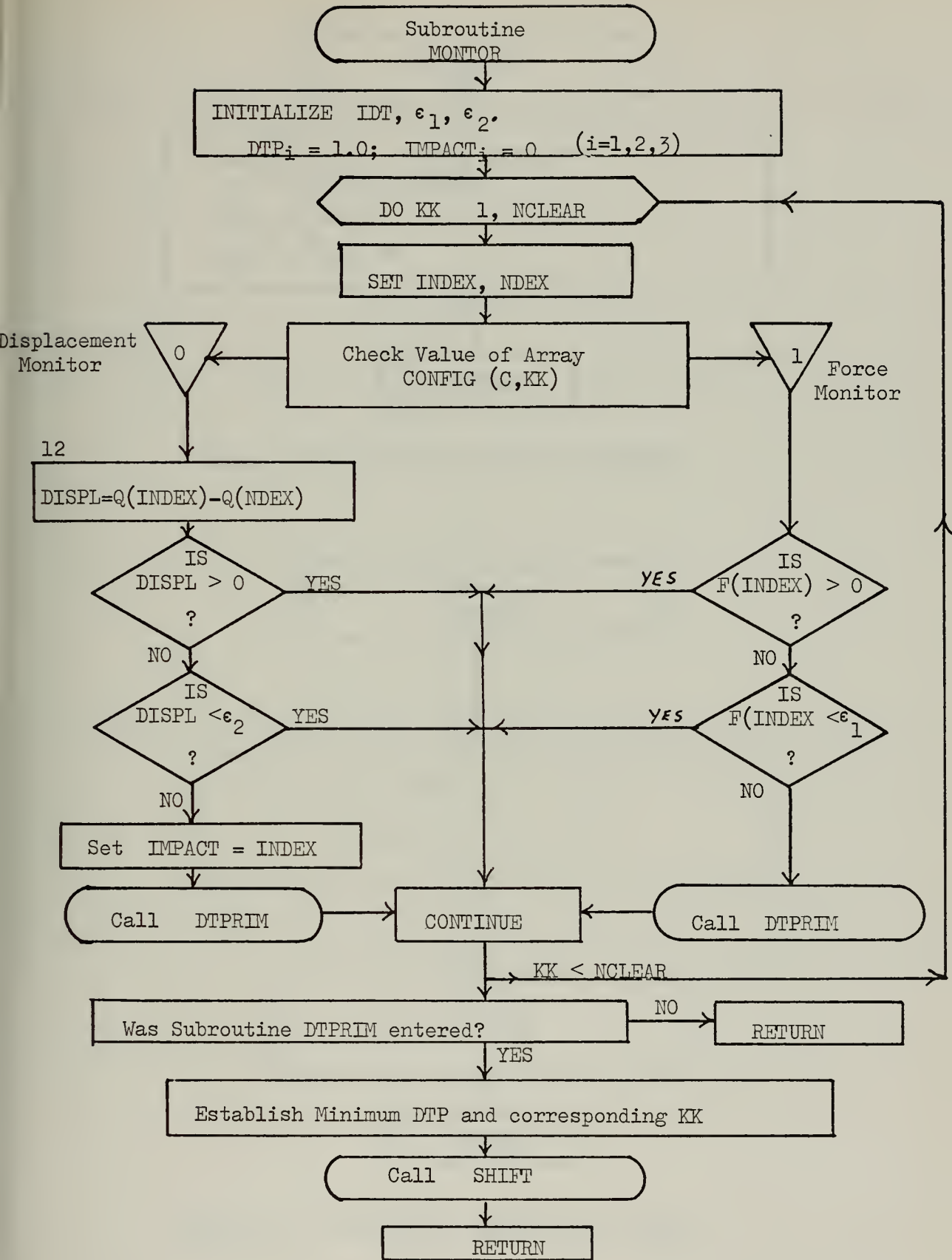


Figure 4.14 Subroutine MONTOR Flow Diagram

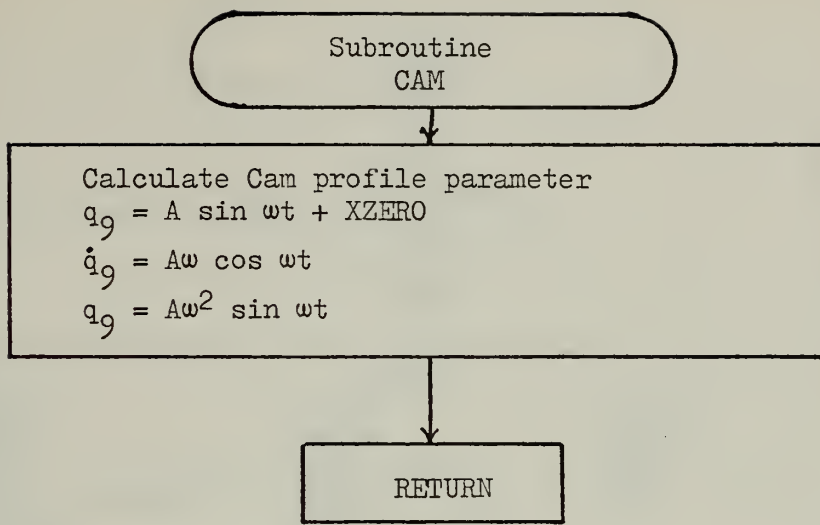


Figure 4.15 Subroutine CAM Flow Diagram

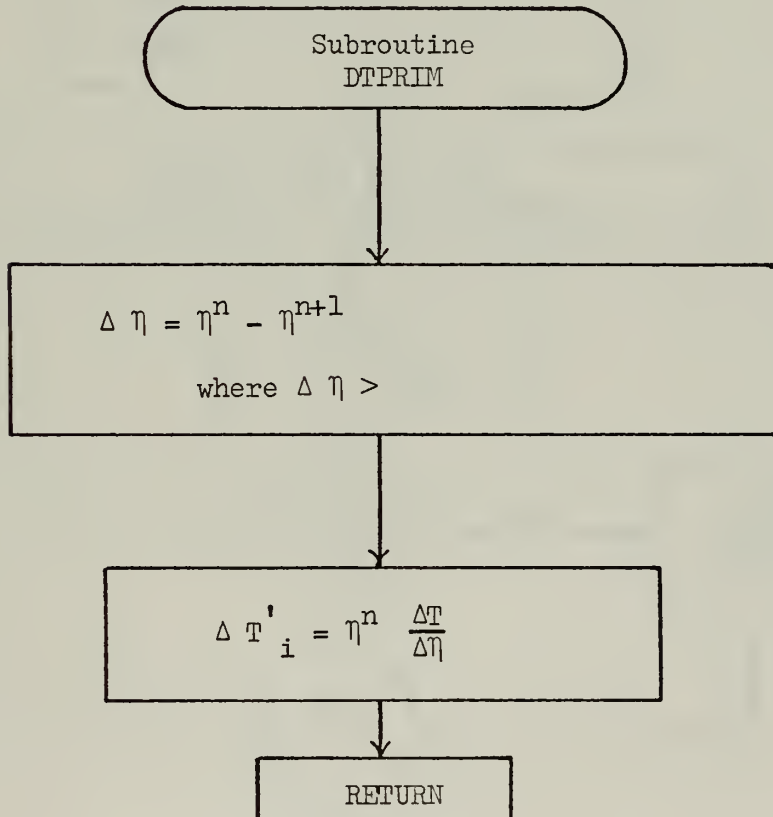


Figure 4.16 Subroutine DTPRIM Flow Diagram

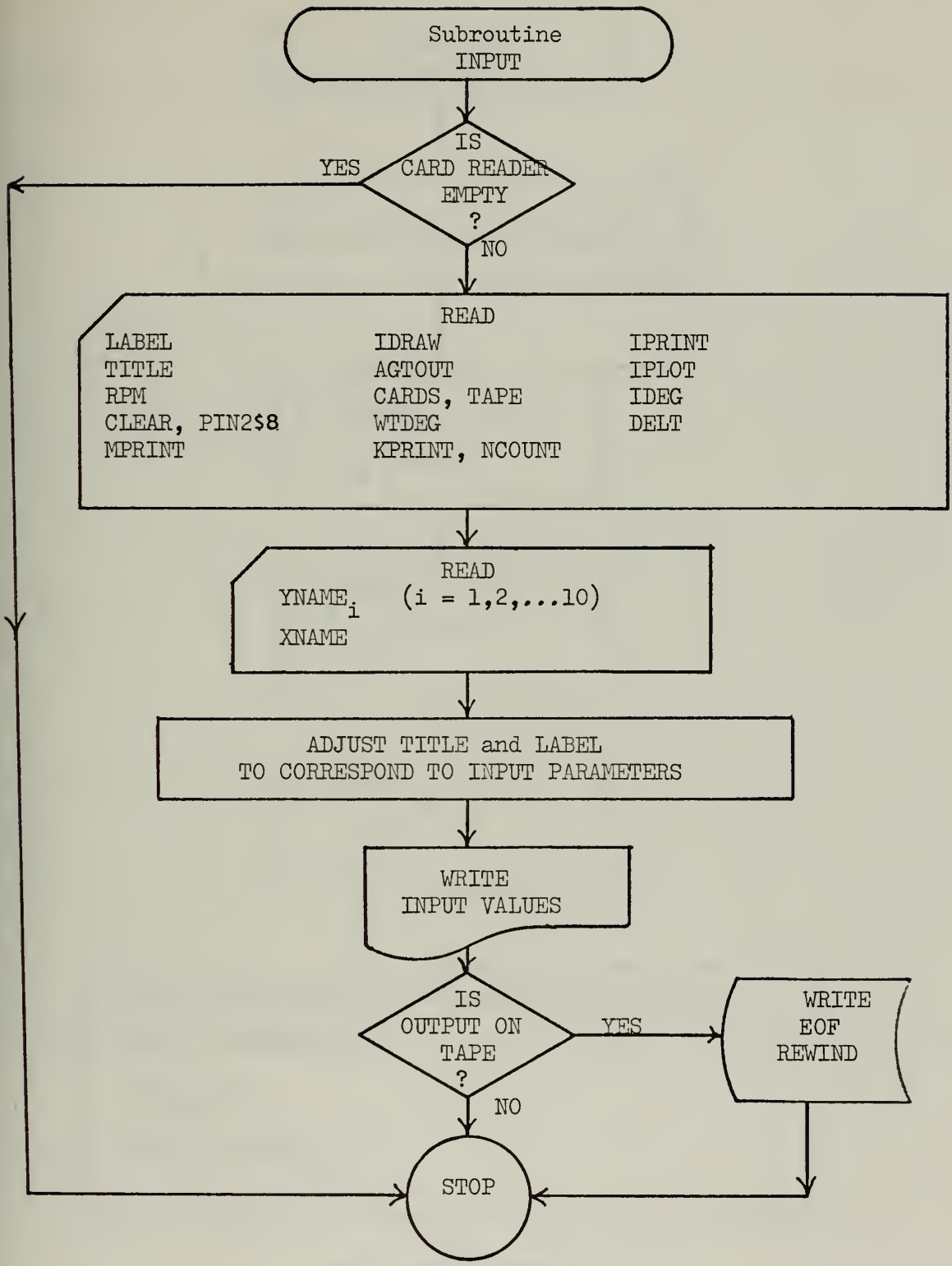


Figure 4.17 Subroutine INPUT Flow Diagram

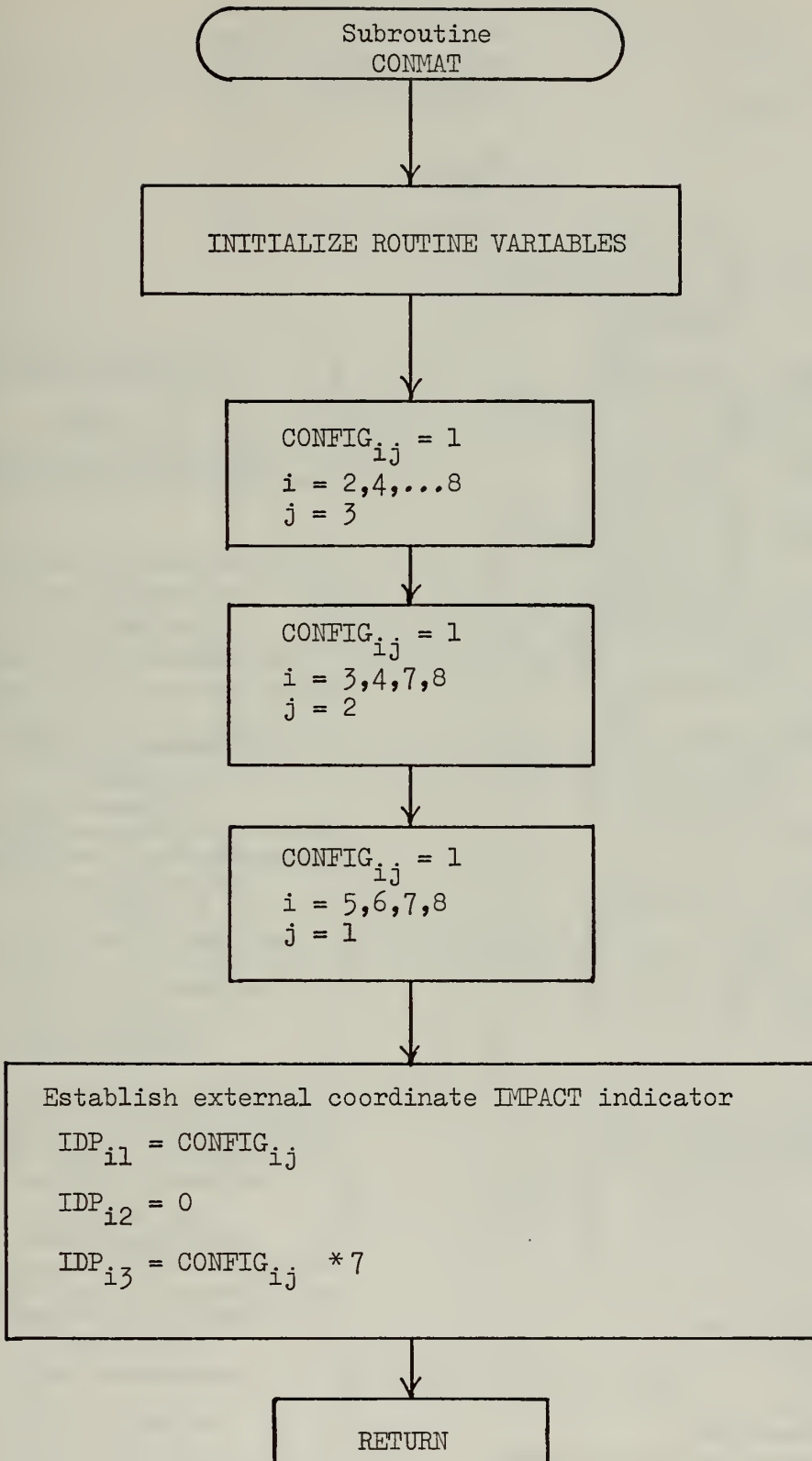


Figure 4.18 Subroutine CONMAT Flow Diagram

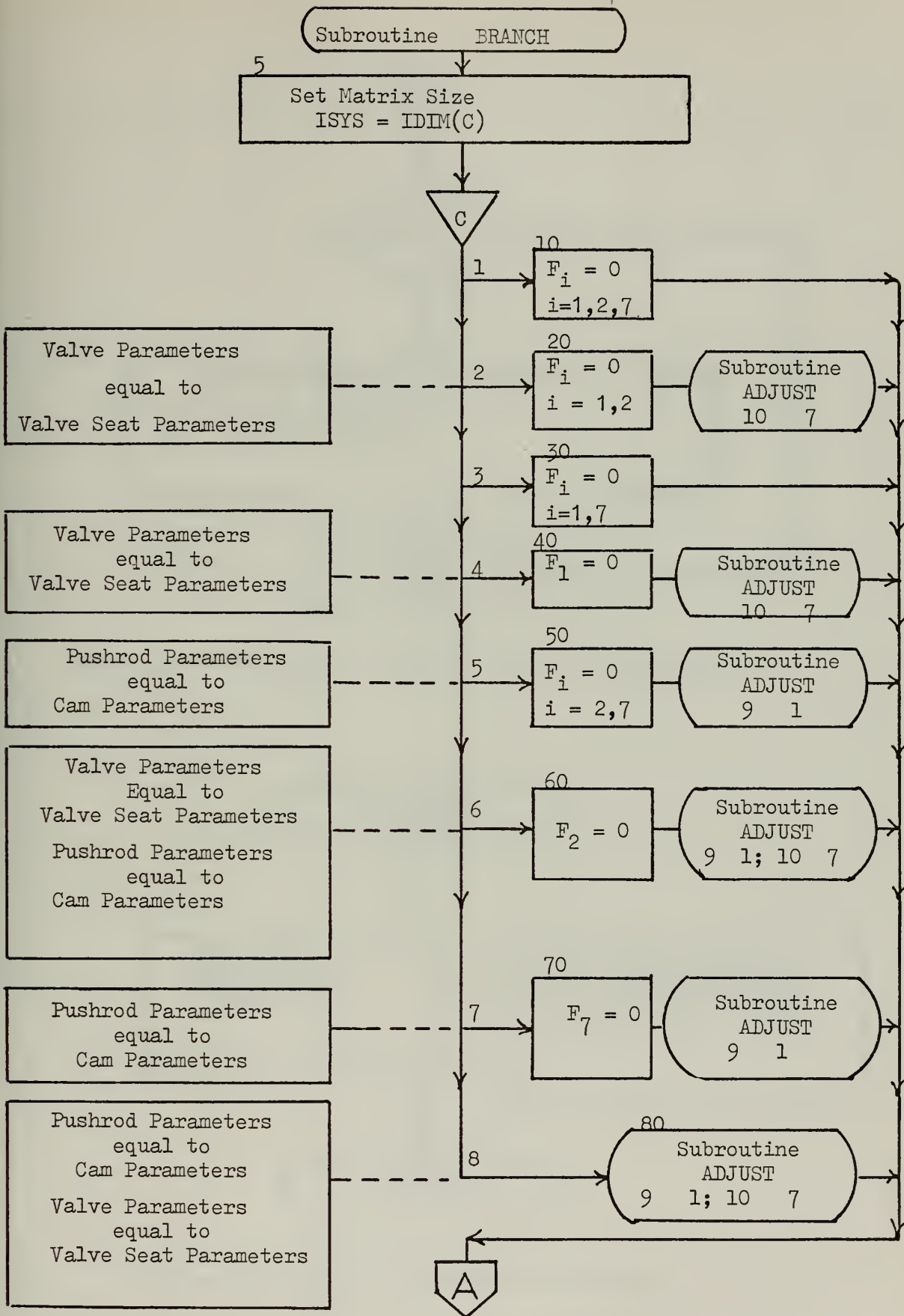


Figure 4.19 Subroutine BRANCH Flow Diagram

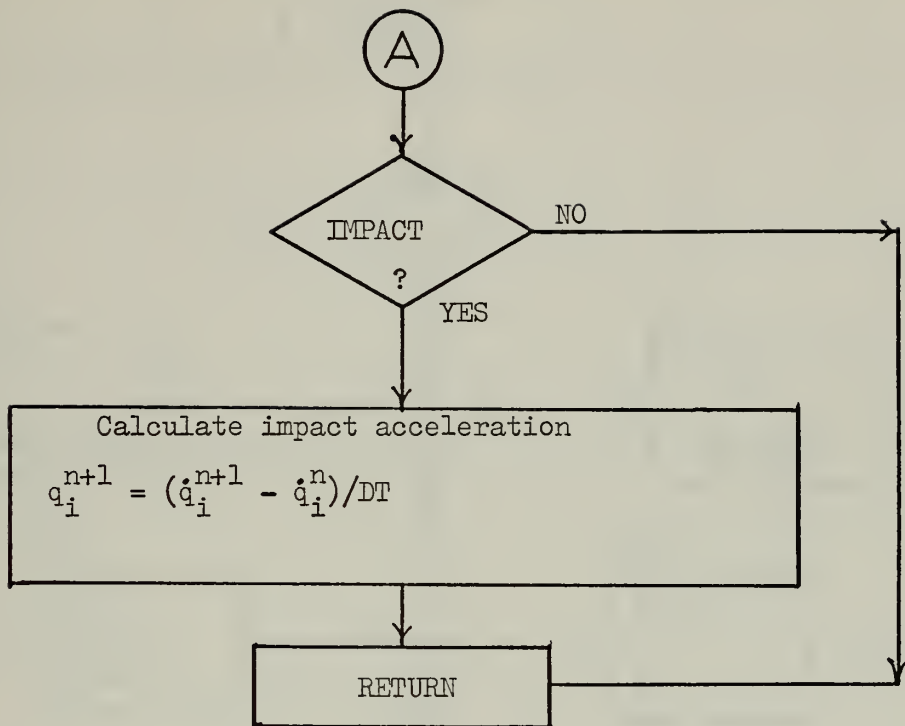


Figure 4.19 Continued - Subroutine BRANCH Flow Diagram

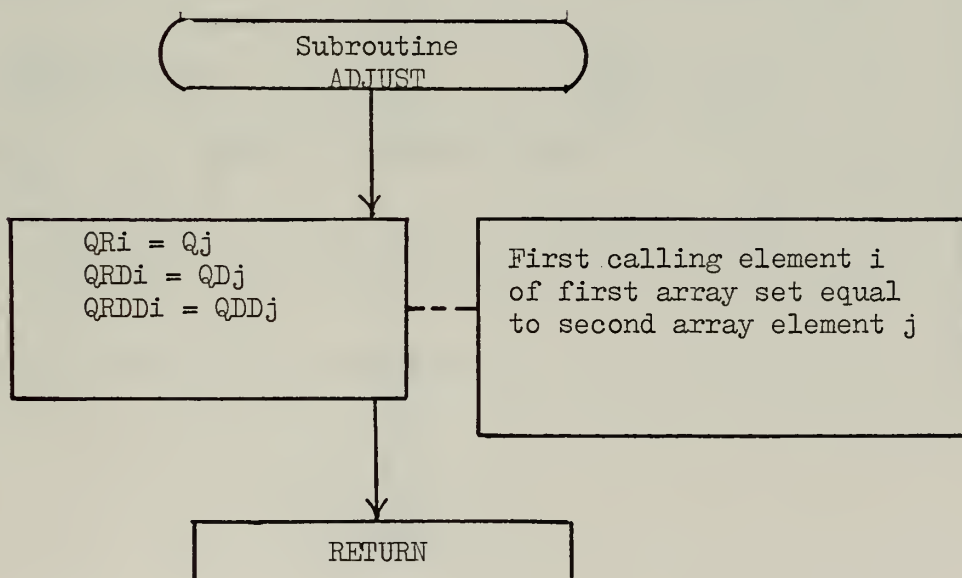


Figure 4.20 Subroutine ADJUST Flow Diagram

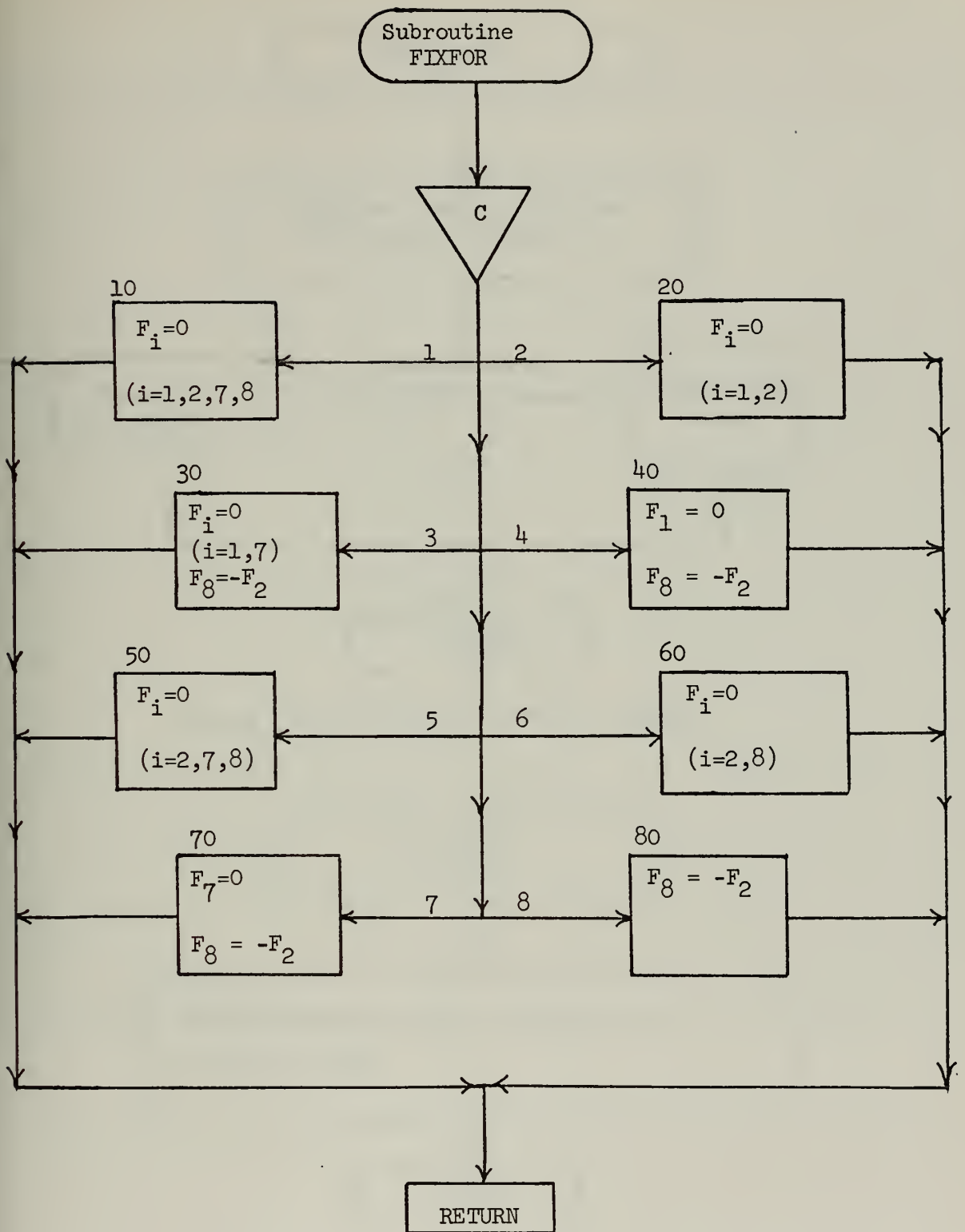


Figure 4.21 Subroutine FIXFOR Flow Diagram

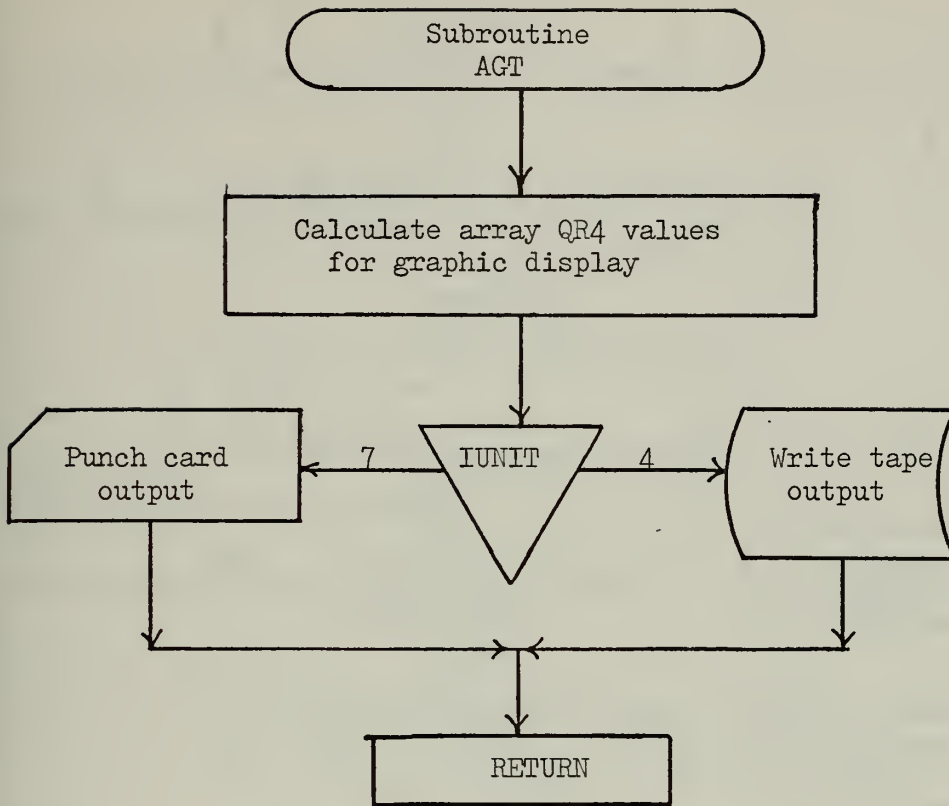


Figure 4.22 Subroutine AGT Flow Diagram

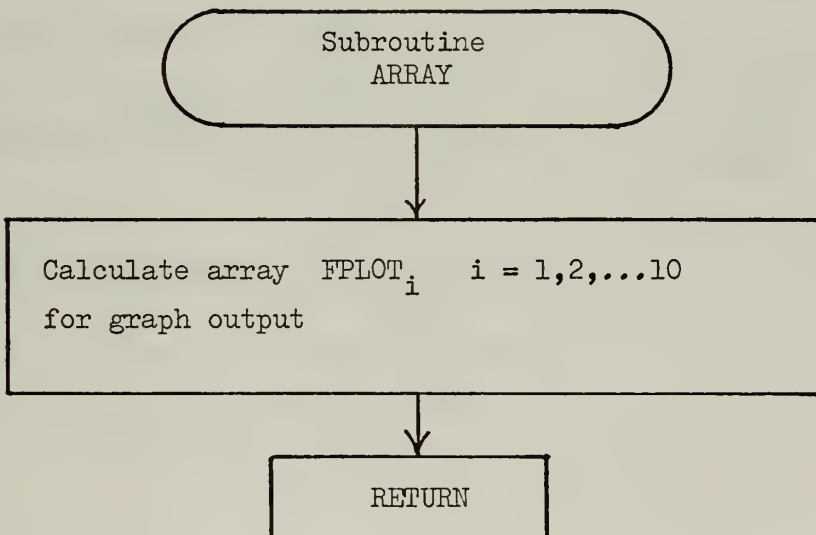


Figure 4.23 Subroutine ARRAY Flow Diagram

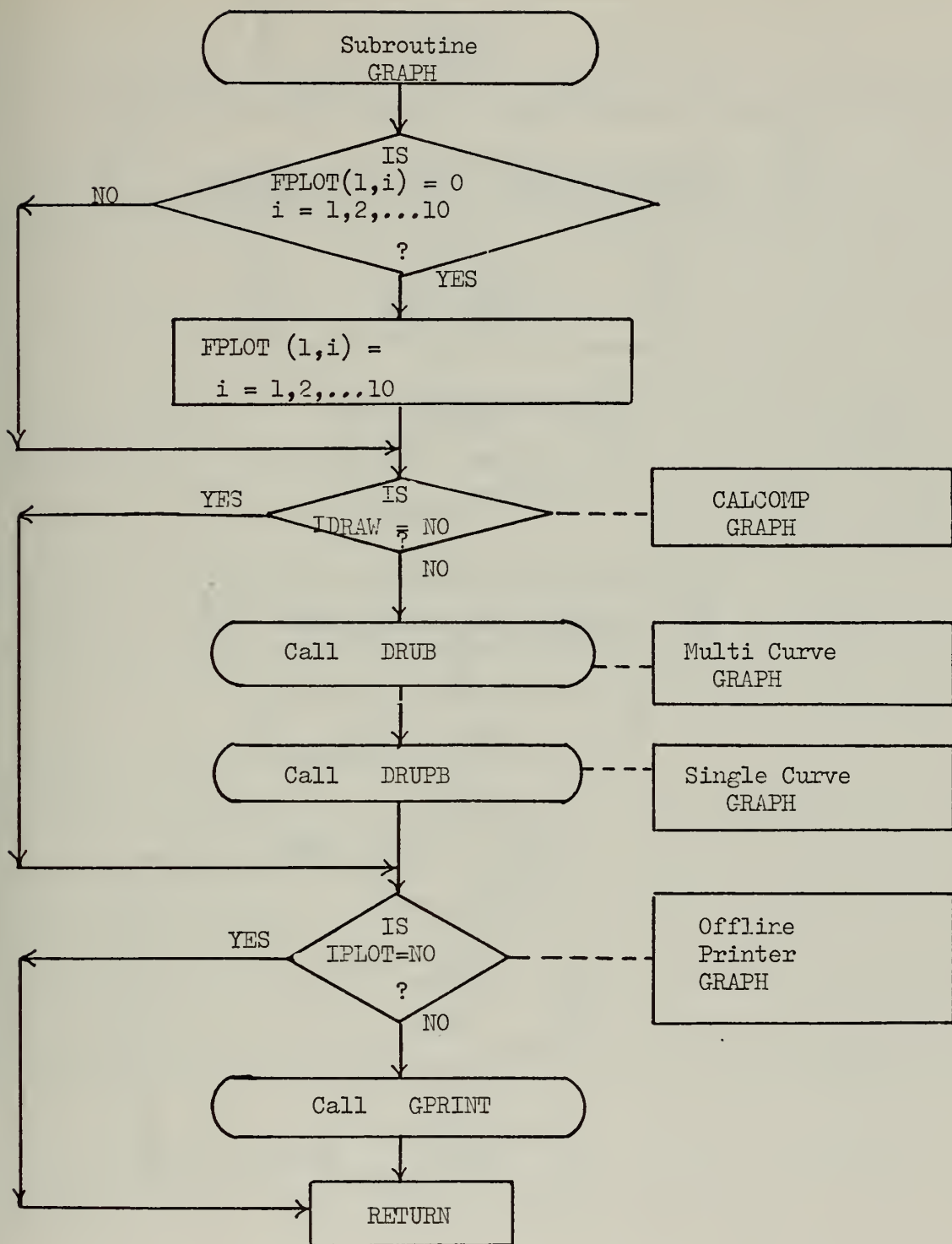


Figure 4.24 Subroutine GRAPH Flow Diagram

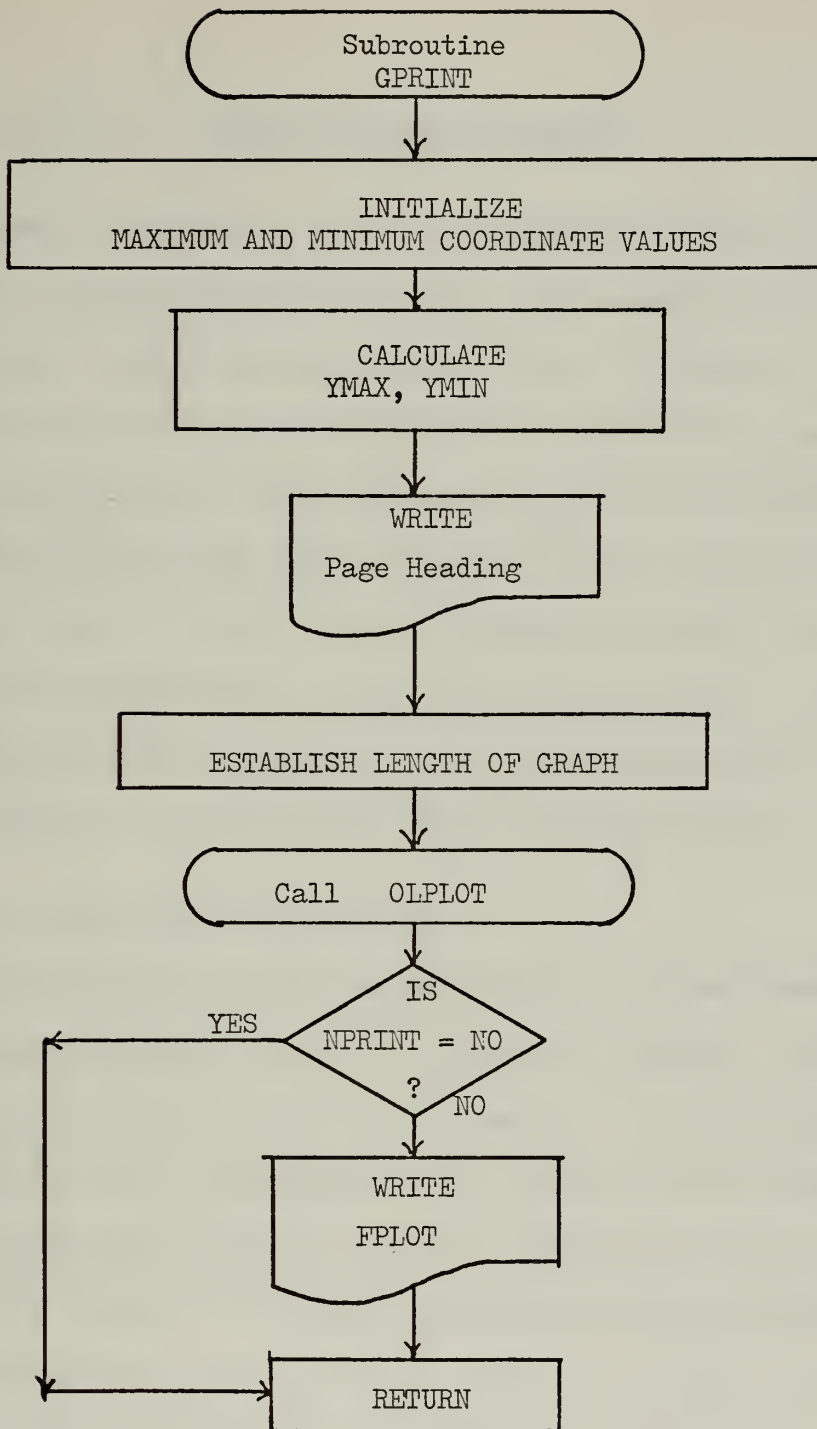


Figure 4.25 Subroutine GPRINT Flow Diagram

V. SDS-9300 COMPUTER PROGRAM

This computer program is coded in SDS FORTRAN IV, (Ref. 5-7), to run on a Scientific Data System Model 9300 computer interfaced with an ADAGE Graphics Display Terminal, Model 10 (AGT-10). This program provides the user the capability to observe the dynamic response of 360 degrees of cam rotation of the valve train assembly. The displacement data generated by the previously described program is the major input. Finite element techniques are used to reformulate the input displacement data to provide for the pictorial responses of bending and axial deformations. Due to the limited core storage of this computer, overlaying techniques have been employed.

A. METHOD OF ELASTICITY CALCULATIONS

As shown in Figure 2.4, the system consists of five elements. Three of these elements, (1, 4, and 5) are axial members. The remaining two, (2 and 3) are bending members. Thus the elastic mode shapes for each of the elements as a function of the distance along the member will yield the required deformations. R. C. Winfrey, reference (1) Chapter V, has done considerable work in the formulation of the required mode shapes by finite element techniques. The mode shapes for the axial elements as a function of x are:

$$\phi_1(x) = \frac{x}{L} \quad (5.1)$$

$$\phi_2(x) = \frac{x}{L} - 1 \quad (5.2)$$

where L is the longitudinal dimension of the bar.

The mode shapes for the bending elements as a function of x are:

$$\Phi_1(x) = \frac{1}{L^3} (L^3 - 3x^2L + 2x^3) \quad (5.3)$$

$$\Phi_2(x) = \frac{1}{L^3} (3x^2L - 2x^3) \quad (5.4)$$

$$\Phi_3(x) = \frac{1}{L^2} (xL^2 - 2x^2L + x^3) \quad (5.4)$$

$$\Phi_4(x) = \frac{1}{L^2} (x^2L - x^3) \quad (5.6)$$

where L is the longitudinal dimension of the bar.

The system displacements are calculated by the IBM-360 program and sent as data to the SDS-9300. These displacements and the above mode shapes are superimposed to yield the required rigid body motion and elastic deformations.

B. PROGRAM STRUCTURE

1. General Description

The main program controls the program execution. The calling order is presented in Figure 5.1. The subroutines indicated by an asterick are supplied by the computer laboratory, and are used to manipulate and initialize data sent to the ADAGE Graphics Terminal. The SDS computer total core available for program execution, exclusive of the control and real time monitoring functions, is approximately 18 K bytes. Since the total program length greatly exceeds the core available, a program overlay technique consisting of storing only active subroutines during execution is employed. This achieves maximum use of the main core storage and allows the program to execute.

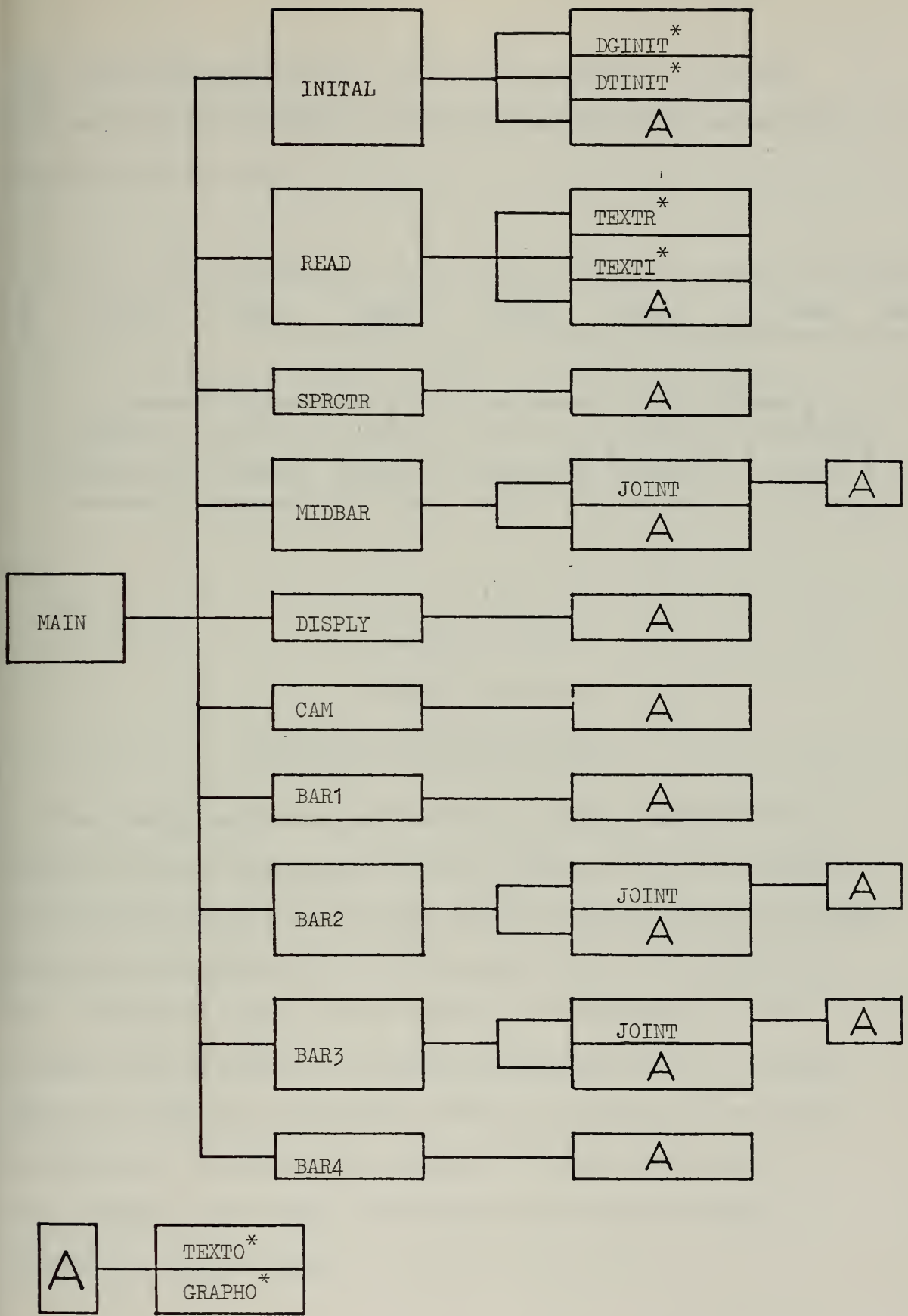
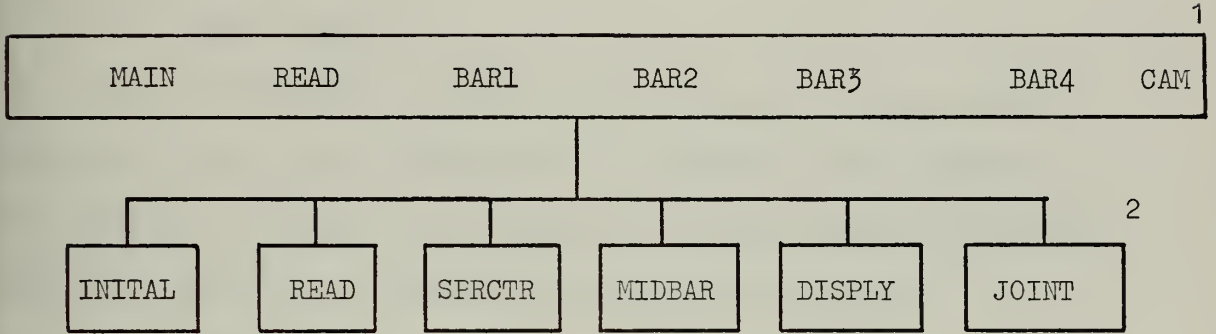


Figure 5.1 SDS-9300 Computer Program Structure

The overlay tree structure is depicted in Figure 5.2. Although necessary for this program, overlaying significantly increases the program execution time.



1. Remains in core
2. Overlay structure

Figure 5.2 Overlay Structure

The bowing and thinning of the axial elements, as seen on the graphics display photographs, have been added as an aid in observing the deformations of these members. The mode shapes given by equations 5.1 and 5.2 are only valid for the elastic line of a one dimensional bar. However the small uniform extension and compression of the element were impossible to satisfactorily reproduce on the display. Thus the bowing and thinning were added as a function of the axial deformation. The thinner the bar, the more axial compression it has undergone. Conversely, the thicker the bar the more axial extension it has undergone.

2. Sense Lights

The SDS computer being a hands-on system, has several physical sense lights. When lit, logic within the program may alter computer instructions as desired.

a. Sense Light 1.

Activation of sense light one branches the execution to the head of the program. Execution then continues and variables are reinitialized as coded in the program. New data points may also be read in, displaying a different set of displacement data.

b. Sense Light 2.

During program execution this sense light indicates that the display will utilize ADAGE Graphics Terminal, number two. When not lit, the SDS computer is interfaced with the AGT unit number one.

c. Sense Light 4.

This sense light branches the program to a convenient point to reinitialize the originally coded program parameters. This is accomplished by activation of the SDS control console teletype from which the user may input any program variables that appear in the namelist statement.

d. Sense Light 6.

Activation of sense light six calls the subroutine TEXTR for each cam rotation increment. This enables the motion of the display to be halted for further study at any point of interest in cam rotation. Depression of the return key on the AGT terminal will advance the display to the next display of cam rotation.

C. MAIN PROGRAM

The main program initializes parameters, establishes the data arrays and contains the calculation loop. Prior to the execution of the program the desired sense lights should be activated. The program is halted almost immediately after execution by an INPUT statement. This statement allows the user to have positive control over assignment of the AGT unit. The designation of the wrong unit at this point will cause a core image failure. The program then must be reloaded. Therefore at this point the 9300 teletype control console should be checked to insure the correct unit is listed in the message to the user. To proceed with the program execution a "*" must be typed on the control console followed by the carriage return key.

Subroutines are called to display the fixed elements of the model. The two arrays that reproduce the circular cam are pre-calculated. Next the frequency ratio is calculated and displayed. The cycle loop is then entered where subroutines are called to display the movable elements. This loop is incremented by a specified amount of cam rotation until 360 degrees of rotation has been reached. At this point the cycle is repeated. For each cam increment, the sense lights are tested.

D. SUBROUTINES

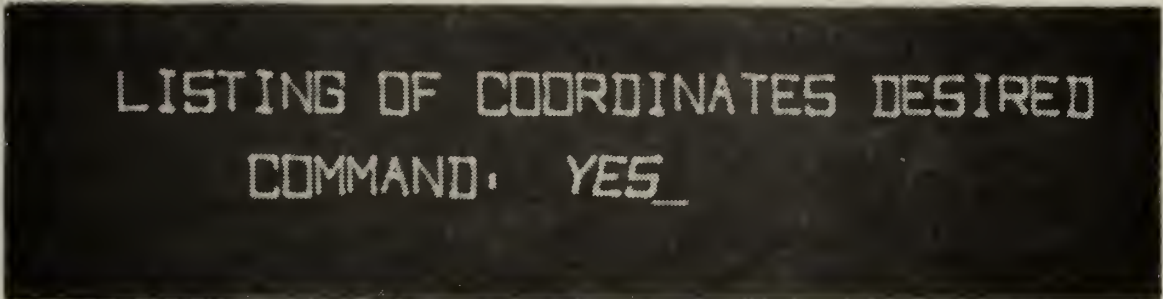
The function of each of the subroutines and a description of their execution are included. Each subroutine is listed in its entirety in the SDS-9300 Computer Program Listing.

1. Subroutine INITIAL

This subroutine initializes the text and graphics directory arrays, DTINIT and DGINIT, respectively. The text response, TEXTR is initialized to block number one at display location 39, 46. The title display is encoded and presented on the AGT. The graphic data blocks are initialized to zero, assigned the appropriate first word header (IHEAD), and then assigned sequence numbers in DGINIT for later use.

2. Subroutine READ

This subroutine reads the title and label for the program and prints them on the offline printer. The program parameters are read, (XZERO, AMP, RPM, WN, ISEAT, and PIN28) and also printed for reference. The user is then questioned from the AGT display, Photo 5.1.



LISTING OF COORDINATES DESIRED
COMMAND: YES_

Photo 5.1 AGT QUESTION

An affirmative response will list the IBM-360 displacement data on the offline printer. The response is entered via the graphics terminal.

The displacement data is next read from magnetic tape or punched cards, depending on the control specification at the time the program was loaded. The array parameters read are, Q1 through Q9, OMEGA, and CONFIG. The Q's are as defined in the IBM program but

modified by the amount XZERO. OMEGA is the amount of revolution of the cam in radians. CONFIG is the corresponding configuration number. Next the displacements are printed, if desired. Finally the title screen is nulled and control is returned to the main program.

3. Subroutine SPRCTR

This subroutine is only called once for a given problem. The display for the rigid portion of the spring is calculated and displayed. The valve seat position and geometry is calculated. The parameter ISEAT is then checked to determine whether the problem considers contact at the valve seat. For those problems with no contact, the valve seat location is adjusted to better picture the position. Finally the valve seat is displayed and control returns to the main program.

4. Subroutine MIDBAR

This subroutine is called once. Subroutine MIDBAR calculates and displays the fixed pivot point.

5. Subroutine DISPLY

This subroutine is called once during each cam increment. The function of this routine is to display the text for the system configurations.

6. Subroutine CAM

Subroutine CAM is called once during each cam increment. This routine calculates the cam position, given the cam rotation (OMEGA) and displacement of the top of the cam (q_0).

7. Subroutine BAR1

This subroutine is called once during each cam increment. Coordinates q_1 and q_8 are used to calculate the position of the pushrod. Equations 5.1 and 5.2 calculate the deformations. The pushrod is then displayed on the AGT.

8. Subroutine BAR2

This subroutine is called once during each cam increment. Coordinates q_2 , q_3 , and q_6 , and equations 5.3, 5.4, and 5.5 are used in the calculations of position, geometry, and deformations of the left rocker arm. The parameter PIN28 is checked to determine whether the pin should be displayed at clearance two. The left rocker arm and pin, if required, are displayed on the AGT.

9. Subroutine BAR3

This subroutine is called once during each cam increment. Coordinates q_4 , q_5 , and q_6 , and equations 5.3, 5.4, and 5.5 are used in the calculation of position, geometry, and deformations of the right rocker arm. Calculation of the position and image of the rocker arm- valve stem- valve spring joint pin is accomplished by subroutine JOINT. The right rocker arm and pin are then displayed on the AGT.

10. Subroutine BAR4

This subroutine is called once during each cam increment. Coordinates q_4 and q_7 are used to calculate the position of the valve stem. Equations 5.1 and 5.2 calculate the deformations. The valve- valve stem is then displayed on the AGT.

11. Subroutine JOINT

This subroutine is called once for the right rocker arm pin and, if required, once for the left rocker arm pin, during each cam increment. The pins' position and image are calculated and displayed on the AGT.

VI. NUMERICAL RESULTS

The precalculated system matrices for the physical properties assumed in chapter 2, are presented in this chapter. Numerical examples have been chosen for their interest and for comparison with one another. The relationship between examples is indicated in Figure 6.26. Example results are presented both as CALCOMP graphs and/or AGT display photos. Photo 6.1 is the title page from the graphics terminal display.

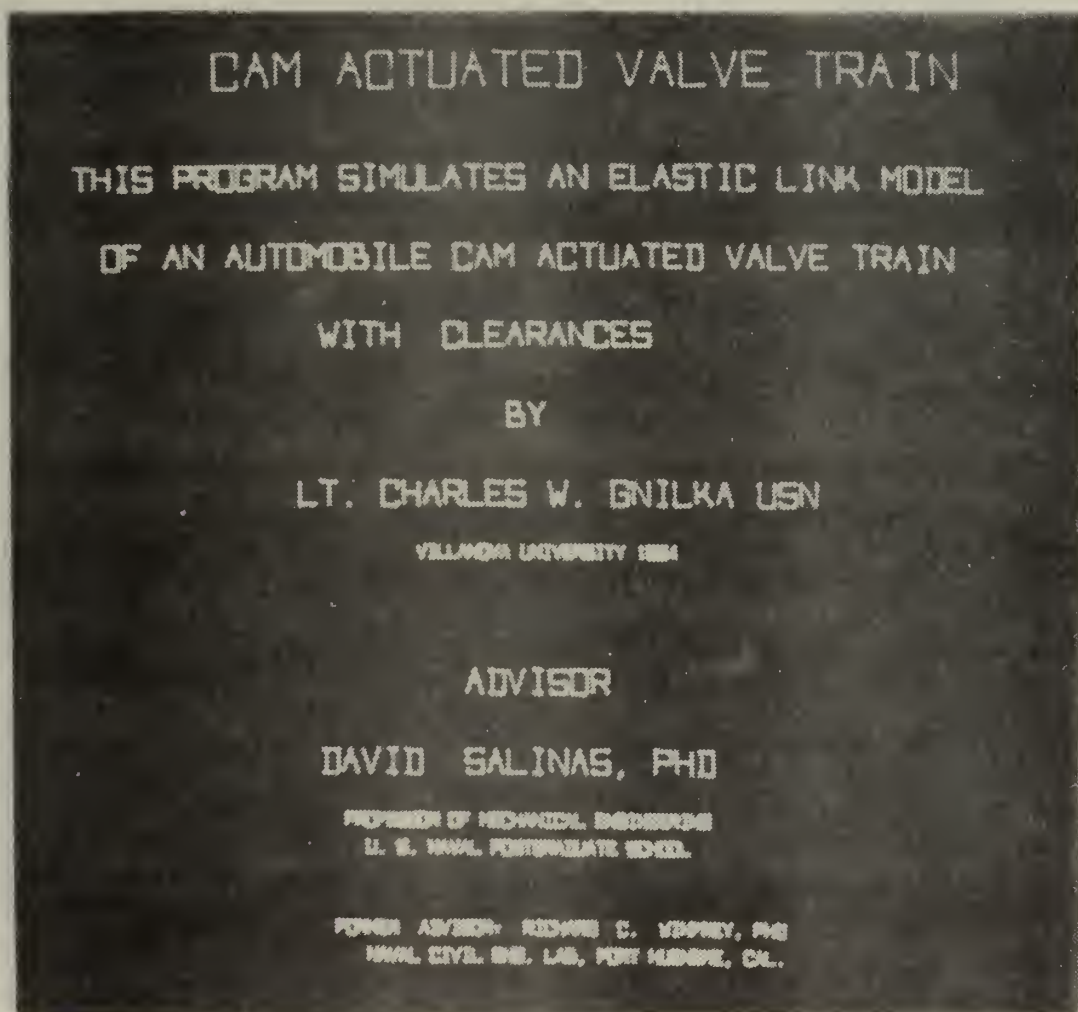


Photo 6.1 AGT Title Page

A. MODE SHAPES AND NATURAL FREQUENCIES

The mode shapes and natural frequencies presented in Figure 6.1 result from the eigenvalue problem for the full eight coordinate system. Figure 6.2 shows the values resulting from the seven coordinate reduced system, for contact at the rocker.

NATURAL FREQUENCIES ω_n (EIGENVALUES) $\times 10^3$							
0.0	63.6	638.7	0.886	263.8	1371.1	149.8	77.86
MODE SHAPES (EIGENVECTORS) IN COLUMNS [ϕ]							
46.3	0.0	0.0	0.0	0.0	0.0	0.0	-80.2
0.0	115.0	-109.5	-37.1	-57.2	-172.7	33.3	0.0
0.0	-131.3	581.0	37.1	149.1	1538.0	-56.0	0.0
0.0	21.5	-34.2	55.6	60.9	18.7	42.5	0.0
0.0	58.7	-499.4	37.0	347.7	376.9	-29.8	0.0
0.0	-70.9	-340.0	37.1	-146.7	381.4	24.6	0.0
0.0	-35.0	-18.9	-55.6	52.8	9.6	120.0	0.0
46.3	0.0	0.0	0.0	0.0	0.0	0.0	80.2

Figure 6.1 FULL SYSTEM MODE SHAPES AND NATURAL FREQ.

NATURAL FREQUENCIES ω_n (EIGENVALUES) $\times 10^3$						
0.691	73.6	521.9	25.9	247.1	1070.2	146.7
MODE SHAPES (EIGENVECTORS) IN COLUMNS [ϕ]						
28.96	-69.26	15.64	45.72	11.22	12.01	-8.39
28.95	64.03	-30.77	20.91	-20.84	-23.92	13.67
-28.95	-70.49	503.13	-30.45	136.63	799.47	-42.66
-43.37	9.09	-34.11	39.89	61.27	25.02	38.35
-28.90	32.45	-427.34	39.72	313.46	471.17	-45.62
-28.93	-42.34	-215.68	-1.53	-164.75	436.79	44.23
43.37	-18.08	-19.75	-42.97	57.06	12.96	118.61

Figure 6.2 REDUCED SYSTEM MODE SHAPES AND NATURAL FREQ.

B. PRE-CALCULATED SYSTEM MATRICES

The remaining system matrices are presented and grouped according to their layer (or configuration) number. The mass, stiffness, and damping matrices for layer one are unaltered and directly correspond to the full eight coordinate system eigenvalue problem. The mass, stiffness, and damping matrices in layer three are unaltered and directly correspond to the reduced seven coordinate system eigenvalue problem.

The pre-calculated matrices COLM, COLC, and COLK located in layers one and three are associated with the calculations for configurations six and eight respectively. Thus, these matrices are not listed in their correct program layer, but with the correct configuration with which they are used.

The order of presentation within each layer is: (1) Mass, (2) Damping, (3) Stiffness, (4) COLM, (5) COLC, (6) COLK, (7) BMIV, (8) DA, and (9) DV.

CONFIGURATION AND MATRIX LAYER = 1

MATRICES ARE 8X8

-- (BMIV) --

```

8.52D 03      0.0 0.99D 04      0.0 0.0 02      0.0 0.31D 04      0.0 0.34D 02      -4.23D 03
0.0           4.96D 05      -2.66D 05      -1.14D 04      -1.14D 04      -2.34D 03
0.0           -2.66D 02      1.97D 06      1.89D 03      1.89D 03      3.94D 03
0.0           -4.90D 02      1.95D 03      1.02D 04      1.24D 04      4.96D 03
0.0           -1.14D 04      1.89D 05      4.24D 04      4.40D 05      2.09D 04
0.0           -1.31D 04      2.18D 05      1.16D 03      1.97D 05      4.04D 03
0.0           -2.34D 02      3.94D 03      4.96D 03      2.09D 04      2.15D 04
-4.23D 03      0.0 0.0           0.0 0.0           0.0 0.0           0.0 0.0           8.52D 03
    
```

-- (DA) --

```

7.72D 03      0.0 0.56D 04      0.0 0.31D 04      0.0 0.0 02      0.0 0.73D 03      0.0 0.68D 02
0.0           -1.53D 05      -2.01D 05      0.0 0.0 05      -6.00D 02      -9.68D 03
0.0           2.53D 03      5.51D 02      4.95D 03      4.95D 03      1.29D 03
0.0           1.81D 03      1.52D 04      6.77D 03      6.77D 03      9.68D 03
0.0           3.66D 04      1.52D 04      8.42D 04      8.42D 04      2.01D 04
0.0           7.47D 04      2.53D 04      4.78D 04      2.83D 04      1.49D 03
0.0           -4.46D 02      7.92D 01      2.01D 04      1.26D 03      2.25D 04
-7.72D 03      0.0 0.0           0.0 0.0           0.0 0.0           0.0 0.0           7.72D 03
    
```

-- (DV) --

```

3.00D 09      0.0 0.51D 11      0.0 0.16D 10      0.0 0.87D 09      0.0 0.36D 10      0.0 0.12D 08
0.0           -2.05D 12      -1.51D 12      1.16D 11      1.47D 11      -6.36D 10      -5.12D 08
0.0           1.90D 10      8.62D 09      -1.93D 09      1.85D 10      7.70D 11      8.40D 09
0.0           1.73D 10      8.62D 09      -8.15D 09      1.85D 10      1.85D 10      1.07D 10
0.0           4.15D 11      2.05D 11      -2.51D 11      2.53D 11      4.00D 11      4.48D 10
0.0           4.83D 11      2.39D 11      -2.03D 11      1.54D 11      4.07D 11      8.62D 09
0.0           8.59D 09      4.28D 09      -9.38D 09      9.13D 09      9.16D 09      1.87D 10
-3.00D 09      0.0 0.0           0.0 0.0           0.0 0.0           0.0 0.0           0.0 0.0           3.00D 09
    
```

Figure 6.4 CONFIGURATION I - Precalculated Matrices; BMIV, DA, and DV

(This page intentionally blank)

CONFIGURATION AND MATRIX LAYER = 2
 MATRICES ARE 7X1

COLUMN MASS	COLUMN DAMPING	COLUMN STIFFNESS
0.0	0.0	0.0
0.0	-7.00D-02	0.0
0.0	-1.01D-02	0.0
-2.59D-05	9.12D-01	7.07D 05
0.0	-1.09D-01	0.0
0.0	5.66D-02	0.0
0.0	0.0	0.0

Figure 6.6 CONFIGURATION II - Precalculated Matrices; COIM, COLC, and COIK

MATRICES ARE 7X7

```

- ( BMIV ) -
8.52D 03 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 4.99D 04 -2.66D 05 0.0 66D 06 0.0 36D 02 0.0 11D 04 0.0 31D 04 -4.23D 03
0.0 -2.66D 05 1.97D 06 7.04D 03 9.03D 03 -1.85D 04 2.17D 05 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 -4.36D 02 -1.11D 04 3.76D 04 4.20D 05 1.93D 05 7.23D 03 2.29D 05 8.52D 03
0.0 -1.11D 04 1.85D 05 7.23D 03 2.17D 05 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
-4.23D 03 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

```

```

- ( DA ) -
7.72D 03 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 7.72D 03 -1.35D 04 -2.31D 04 4.78D 03 -5.86D 02 0.0 74D 03 -7.72D 03
0.0 0.0 2.53D 05 0.0 1D 05 1.26D 04 4.71D 03 6.22D 04 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 1.91D 03 5.32D 02 -1.10D 03 6.48D 03 3.28D 02 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 3.70D 04 1.51D 04 -8.22D 04 8.30D 04 6.01D 04 9.49D 04 7.72D 03
0.0 7.48D 04 2.53D 04 -5.16D 04 2.81D 04 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
-7.72D 03 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

```

```

- ( DV ) -
3.00D 09 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 -2.05D 11 -1.51D 11 0.0 17D 10 -8.77D 09 -6.35D 10 -3.00D 09
0.0 1.90D 12 1.28D 12 1.95D 11 1.46D 11 7.68D 11 0.0 0.0 0.0 0.0 0.0
0.0 1.53D 10 7.64D 09 -1.03D 10 1.64D 10 1.64D 10 0.0 0.0 0.0 0.0 0.0
0.0 4.07D 11 2.02D 11 -2.60D 11 2.44D 11 3.91D 11 0.0 0.0 0.0 0.0 0.0
0.0 4.82D 11 2.39D 11 -2.05D 11 1.53D 11 4.05D 11 0.0 0.0 0.0 0.0 0.0
-3.00D 09 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

```

Figure 6.7 CONFIGURATION II - Precalculated Matrices; BMIV, DA, and DV

MATRICES ARE 7X7

---- SYSM ----						
1.55D-04	7.77D-05	0.0	0.0	0.0	0.0	0.0
7.77D-05	2.32D-04	1.09D-05	0.0	0.0	-6.42D-06	0.0
0.0	1.09D-05	1.97D-06	0.0	0.0	-1.48D-06	0.0
0.0	0.0	0.0	2.01D-04	-2.44D-05	1.44D-05	-2.59D-05
0.0	0.0	0.0	-2.44D-05	6.66D-06	-5.00D-06	0.0
0.0	-6.42D-06	-1.48D-06	1.44D-05	-5.00D-06	8.64D-06	0.0
0.0	0.0	0.0	-2.59D-05	0.0	0.0	5.18D-05
---- DAMP ----						
2.48D 00	1.05D 00	-7.37D-03	1.72D 00	-1.60D-01	9.80D-02	-6.62D-01
1.05D 00	3.49D 00	3.58D-01	2.09D 00	-3.08D-01	3.16D-01	-6.69D-01
-7.37D-03	3.58D-01	2.03D-01	1.02D-01	-3.48D-02	2.59D-03	-1.65D-02
-1.72D 00	2.09D 00	1.02D-01	3.10D 00	-4.83D-01	3.25D-02	2.99D-01
-1.60D-01	-3.08D-01	-3.48D-02	-4.83D-01	2.86D-01	-8.02D-02	-5.40D-02
-9.80D-02	3.16D-01	2.59D-03	3.25D-02	-8.02D-02	4.91D-01	2.90D-02
-6.62D-01	-6.69D-01	-1.65D-02	2.99D-01	-5.40D-02	2.90D-02	1.17D 00
---- STF ----						
2.36D 05	-2.36D 05	0.0	0.0	0.0	0.0	0.0
-2.36D 05	2.53D 06	1.15D 06	0.0	0.0	1.15D 06	0.0
0.0	1.15D 06	7.63D 05	0.0	0.0	3.82D 05	0.0
0.0	0.0	0.0	1.39D 06	-5.09D 05	-5.09D 05	7.07D 05
0.0	0.0	0.0	-5.09D 05	5.09D 05	2.54D 05	0.0
0.0	1.15D 06	3.82D 05	7.07D 05	2.54D 05	1.27D 06	0.0
0.0	0.0	0.0	0.0	0.0	0.0	7.07D 05

Figure 6.8 CONFIGURATION III - Mass, Damping, and Stiffness

MATRICES ARE 7X7

-- (BMIV) --
 8.18D 03 1.90D 04 1.34D 01 7.67D 02 9.06D 02 3.82D 01
 -3.62D 03 -3.81D 04 -7.89D 01 -1.58D 03 -1.82D 03 -2.14D 01
 1.90D 04 -3.81D 04 -7.89D 01 -1.58D 03 -1.82D 03 -2.14D 01
 1.34D 01 5.68D 03 1.01D 04 4.23D 04 8.03D 03 4.97D 03
 7.67D 02 -1.58D 03 4.23D 04 4.38D 05 1.94D 05 2.08D 04
 9.06D 02 -1.82D 03 8.03D 03 1.94D 05 2.27D 05 3.98D 03
 3.82D 01 -2.14D 01 2.79D 03 2.08D 04 3.98D 03 2.15D 04

-- (DA) --
 1.63D 04 2.88D 03 8.18D 03 7.17D 02 9.49D 01 3.28D 03
 -1.09D 03 7.55D 03 5.46D 03 5.96D 02 1.06D 03 -1.84D 03
 3.14D 03 1.73D 05 -1.18D 04 5.78D 03 5.79D 04 -2.73D 03
 8.08D 03 9.12D 03 6.32D 02 6.10D 03 1.01D 04 6.76D 03
 7.20D 03 4.42D 04 1.61D 04 8.36D 04 6.17D 04 1.74D 04
 1.65D 03 7.66D 04 2.57D 04 2.80D 04 9.60D 04 1.22D 03
 -8.56D 03 -8.19D 03 -4.62D 01 1.98D 03 1.07D 03 2.56D 04

-- (DV) --
 2.78D 09 1.18D 10 -8.06D 08 6.14D 08 4.46D 09 3.65D 07
 -2.56D 09 -2.66D 10 1.61D 09 -1.23D 09 -4.93D 09 -7.09D 07
 1.35D 10 9.37D 11 -1.38D 11 1.06D 11 4.73D 11 5.99D 09
 2.17D 07 1.55D 10 7.31D 09 1.84D 10 1.79D 10 1.07D 10
 5.54D 08 3.72D 11 1.75D 11 2.51D 11 3.86D 11 4.46D 10
 6.43D 08 4.34D 11 2.04D 11 1.52D 11 3.91D 11 8.49D 09
 1.41D 07 7.69D 09 3.63D 09 9.09D 09 8.87D 09 1.87D 10

Figure 6.9 CONFIGURATION III - Precalculated Matrices; BMIV, DA, and DV

MATRICES ARE 6X6

---- SYSM ----

```

1.55D-04  7.77D-05  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
7.77D-05  2.32D-04  1.09D-05  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
0.0  1.09D-05  1.97D-06  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
0.0  0.0  0.0  2.01D-04  -2.44D-05  0.0  0.0  0.0  0.0  0.0  0.0  0.0
0.0  0.0  0.0  -2.44D-05  6.66D-06  -5.00D-06  8.64D-06  0.0  0.0  0.0  0.0
0.0  -6.42D-06  -1.48D-06  1.44D-05  -5.00D-06  -5.00D-06  8.64D-06  0.0  0.0  0.0  0.0
    
```

---- DAMP ----

```

2.48D 00  1.05D 00  -7.37D-03  1.72D 00  -1.60D-01  9.80D-02
1.05D 00  3.49D 00  3.58D-01  2.09D 00  -3.08D-01  3.16D-01
-7.37D-03  3.58D-01  2.03D-01  1.02D-01  -3.48D-02  2.59D-03
-1.72D 00  2.09D 00  1.02D-01  3.10D 00  -4.83D-01  3.25D-02
-1.60D-01  -3.08D-01  -3.48D-02  -4.83D-01  2.86D-01  -8.02D-02
9.80D-02  3.16D-01  2.59D-03  3.25D-02  -8.02D-02  4.91D-01
    
```

---- STF ----

```

-2.36D 05  -2.36D 05  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
-2.36D 05  1.15D 06  1.15D 06  0.0  0.0  0.0  0.0  0.0  0.0  0.0
0.0  1.15D 06  7.63D 05  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
0.0  0.0  0.0  0.0  1.39D 06  -5.09D 05  -5.09D 05  5.09D 05  5.09D 05  2.54D 05  2.54D 06
0.0  0.0  0.0  0.0  -5.09D 05  -5.09D 05  5.09D 05  5.09D 05  2.54D 05  2.54D 06  1.27D 06
0.0  1.15D 06  3.82D 05  -5.09D 05  -5.09D 05  5.09D 05  5.09D 05  2.54D 05  2.54D 06  1.27D 06
    
```

Figure 6.10 CONFIGURATION IV - Mass, Damping, and Stiffness

CONFIGURATION AND MATRIX LAYER = 4
 MATRICES ARE 6X1

COLUMN MASS	COLUMN DAMPING	COLUMN STIFFNESS
0.0	-6.62D-01	0.0
0.0	-6.69D-01	0.0
0.0	-1.65D-02	0.0
-2.59D-05	2.99D-01	7.07D 05
0.0	-5.40D-02	0.0
0.0	2.90D-02	0.0

Figure 6.11 CONFIGURATION IV - Precalculated Matrices; COLM, COLC, and COLK

MATRICES ARE 6X6

-- (BMIV) --

```

8.18D 03 -3.62D 03 1.90D 04 4.55D 00 7.30D 02 8.99D 02
-3.62D 03 -7.24D 03 -3.81D 04 -7.39D 01 -1.56D 03 -1.82D 03
1.90D 04 -3.81D 04 7.49D 05 5.03D 03 1.32D 05 1.56D 05
4.55D 00 -3.39D 01 5.03D 03 9.00D 03 3.75D 04 7.11D 03
7.30D 02 -1.56D 01 1.32D 05 3.75D 04 4.18D 05 1.90D 05
8.99D 02 -1.82D 03 1.56D 05 7.11D 03 1.90D 05 2.26D 05
    
```

-- (DA) --

```

1.63D 04 2.90D 03 2.48D 03 8.16D 03 -7.20D 02 9.30D 01
-1.10D 03 7.55D 03 -5.07D 03 5.47D 03 -5.94D 02 1.06D 03
-4.26D 03 1.74D 05 1.35D 05 -1.33D 04 5.52D 03 5.78D 04
1.01D 04 1.10D 04 6.32D 02 1.04D 04 5.64D 03 7.64D 02
1.55D 04 5.21D 04 1.61D 04 -6.78D 04 8.17D 04 6.07D 04
3.24D 03 7.81D 04 2.57D 04 -4.86D 04 2.76D 04 9.58D 04
    
```

-- (DV) --

```

2.78D 09 1.18D 10 1.07D 10 -8.23D 08 5.98D 08 4.45D 09
-2.56D 09 -2.66D 10 -2.15D 10 1.62D 09 -1.22D 09 -8.93D 09
1.35D 10 9.36D 11 5.88D 11 -1.40D 11 1.04D 11 4.72D 11
1.85D 07 1.37D 10 6.47D 09 -1.02D 10 1.63D 10 1.58D 10
5.40D 08 3.65D 11 1.72D 11 -2.57D 11 2.42D 11 3.77D 11
6.41D 08 4.32D 11 2.03D 11 -2.02D 11 1.50D 11 3.90D 11
    
```

Figure 6.12 CONFIGURATION IV - Precalculated Matrices; BMIV, DA, and DV

MATRICES ARE 7X7

--- SYSM ---

7.70D-05	1.09D-05	0.0	0.0	-6.42D-06	0.0	0.0	0.0
1.09D-05	1.97D-06	0.0	0.0	-1.48D-06	0.0	0.0	0.0
0.0	0.0	2.01D-04	-2.44D-05	-1.44D-05	-2.59D-05	0.0	0.0
0.0	0.0	-2.44D-05	6.66D-06	-5.00D-06	0.0	0.0	0.0
-6.42D-06	-1.48D-06	-1.44D-05	-5.00D-06	8.64D-06	0.0	0.0	0.0
0.0	0.0	-2.59D-05	0.0	0.0	5.18D-05	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.55D-04

--- DAMP ---

1.32D 00	2.93D-01	5.54D-01	-1.69D-01	2.49D-01	-7.00D-02	0.0	0.0
2.93D-01	1.42D-01	1.00D-01	-3.70D-02	2.01D-02	-1.01D-02	0.0	0.0
5.54D-01	1.00D-01	1.51D 00	-3.36D-01	-5.39D-02	9.12D-01	0.0	0.0
-1.69D-01	-3.70D-02	-3.36D-01	2.71D-01	-6.93D-02	-1.09D-01	0.0	0.0
-2.49D-01	2.01D-02	-5.39D-02	-6.93D-02	4.69D-01	5.66D-02	0.0	0.0
-7.00D-02	-1.01D-02	9.12D-01	-1.09D-01	5.66D-02	9.31D-01	0.0	6.05D-01
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

--- STF ---

2.29D 06	1.15D 06	0.0	0.0	1.15D 06	0.0	0.0	0.0
1.15D 06	7.63D 05	0.0	0.0	3.82D 05	0.0	0.0	0.0
0.0	0.0	1.39D 06	-5.09D 05	-5.09D 05	7.07D 05	0.0	0.0
0.0	0.0	-5.09D 05	5.09D 05	2.54D 05	0.0	0.0	0.0
1.15D 06	3.82D 05	-5.09D 05	2.54D 05	1.27D 06	0.0	0.0	0.0
0.0	0.0	7.07D 05	0.0	0.0	7.07D 05	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.36D 05

Figure 6.13 CONFIGURATION V - Mass, Damping, and Stiffness

CONFIGURATION AND MATRIX LAYER = 5
 MATRICES ARE 7X1

COLUMN MASS	COLUMN DAMPING	COLUMN STIFFNESS
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0
7.77D-05	-6.05D-01	-2.36D 05

Figure 6.14 CONFIGURATION V -- PrecaLculated Matrices; COLM, COLC, and COLK

MATRICES ARE 7X7

-- (BMIV) --

4.99D 04	-2.66D 05	-4.90D 02	-1.14D 04	-1.31D 04	04	-2.34D 02	0.0
-2.66D 05	1.97D 06	7.95D 03	1.89D 05	2.18D 05	05	3.94D 03	0.0
-4.90D 02	1.89D 03	1.02D 04	4.24D 04	8.16D 03	03	4.96D 03	0.0
-1.14D 04	1.89D 05	4.24D 05	4.40D 05	1.97D 05	05	2.09D 04	0.0
-1.31D 04	2.18D 05	8.16D 03	1.97D 05	2.30D 05	05	4.04D 03	0.0
-2.34D 02	3.94D 03	4.96D 03	2.09D 04	4.04D 03	03	2.15D 04	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.42D 03

-- (DA) --

-1.35D 04	-2.31D 04	4.56D 03	-6.00D 02	1.73D 03	03	-9.68D 02	0.0
-2.53D 05	2.01D 05	-8.87D 03	4.95D 03	6.25D 04	04	1.29D 03	0.0
1.81D 03	5.51D 02	-5.75D 04	6.77D 03	6.55D 02	02	9.68D 03	0.0
3.66D 04	1.52D 04	-6.27D 04	8.42D 04	6.15D 04	04	2.01D 04	0.0
7.47D 04	2.53D 04	-4.78D 04	2.83D 04	9.52D 04	04	1.49D 03	0.0
-4.46D 02	7.92D 01	2.01D 04	1.26D 03	1.42D 03	03	2.25D 04	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.88D 03

-- (DV) --

-2.05D 11	-1.51D 11	1.16D 10	-8.87D 09	-6.36D 10	10	-5.12D 08	0.0
1.90D 12	1.28D 12	-1.93D 11	1.47D 11	7.70D 11	11	8.40D 09	0.0
1.73D 10	8.62D 09	-8.15D 09	1.85D 10	1.85D 10	10	1.07D 10	0.0
4.15D 11	2.06D 11	-2.51D 11	2.53D 11	4.00D 11	11	4.48D 10	0.0
4.83D 11	2.39D 11	-2.03D 11	1.54D 11	4.07D 11	11	8.62D 09	0.0
8.59D 09	4.28D 09	9.38D 09	9.13D 09	9.16D 09	09	1.87D 10	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.51D 09

Figure 6.15 CONFIGURATION V -- Precalculated Matrices; BMIV, DA, and DV

CONFIGURATION AND MATRIX LAYER = 6

MATRICES ARE 6X6

--- SYSM ---

```

7.70D-05  1.09D-05  0.0  0.0  -6.42D-06  0.0
1.09D-05  1.97D-06  0.0  0.0  -1.48D-06  0.0
0.0  0.0  2.01D-04  -2.44D-05  1.44D-05  0.0
0.0  0.0  -2.44D-05  6.66D-06  -5.00D-06  0.0
-6.42D-06  -1.48D-06  1.44D-05  -5.00D-06  8.64D-06  0.0
0.0  0.0  0.0  0.0  0.0  1.55D-04
    
```

--- DAMP ---

```

1.32D 00  2.93D-01  5.54D-01  -1.69D-01  2.49D-01  0.0
2.93D-01  1.42D-01  1.00D-01  -3.70D-02  2.01D-02  0.0
5.54D-01  1.00D-01  1.51D 00  -3.36D-01  -5.39D-02  0.0
-1.69D-01  -3.70D-02  -3.36D-01  2.71D-01  -6.93D-02  0.0
2.49D-01  2.01D-02  -5.39D-02  -6.93D-02  4.69D-01  0.0
0.0  0.0  0.0  0.0  0.0  6.05D-01
    
```

--- STF ---

```

2.29D 06  1.15D 06  0.0  0.0  1.15D 06  0.0
1.15D 06  7.63D 05  0.0  0.0  3.82D 05  0.0
0.0  0.0  1.39D 06  -5.09D 05  -5.09D 05  0.0
0.0  0.0  -5.09D 05  5.09D 05  2.54D 05  0.0
1.15D 06  3.82D 05  -5.09D 05  2.54D 05  1.27D 06
0.0  0.0  0.0  0.0  0.0  2.36D 05
    
```

Figure 6.16 CONFIGURATION VI - Mass, Damping, and Stiffness

CONFIGURATION AND MATRIX LAYER = 6
 MATRICES ARE 6X1

COLUMN MATRICES FOR CALCULATION OF DELTA P		
COLUMN MASS	COLUMN DAMPING	COLUMN STIFFNESS
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0
7.77D-05	-6.05D-01	-2.36D 05

Figure 6.17 CONFIGURATION VI - Precalculated Matrices; COLM, COLC, and COLK

CONFIGURATION = 6 MATRIX LAYER = 1
 MATRICES ARE 6X1

COLUMN MASS	COLUMN DAMPING	COLUMN STIFFNESS
0.0	-7.00D-02	0.0
0.0	-1.01D-02	0.0
-2.59D-05	9.12D-01	7.07D 05
0.0	-1.09D-01	0.0
0.0	5.66D-02	0.0
0.0	0.0	0.0

Figure 6.13 CONFIGURATION VI - Precalculated Matrices; COLM, COLC, and COLK

CONFIGURATION AND MATRIX LAYER = 6

MATRICES ARE 6X6

```

-- ( BMIV ) --
4.99D 04 -2.66D 05 -4.36D 02 -1.11D 04 -1.31D 04 0.0
-2.66D 05 1.97D 06 7.04D 03 1.85D 05 2.17D 05 0.0
-4.36D 02 7.04D 03 9.03D 03 3.76D 04 7.23D 03 0.0
-1.11D 04 1.85D 05 3.76D 04 4.20D 05 1.93D 05 0.0
-1.31D 04 2.17D 05 7.23D 03 1.93D 05 2.29D 05 0.0
0.0 0.0 0.0 0.0 0.0 6.42D 03

-- ( DA ) --
-1.35D 04 -2.31D 04 4.78D 03 -5.86D 02 1.74D 03 0.0
2.53D 05 2.01D 05 -1.26D 04 4.71D 03 6.22D 04 0.0
1.91D 03 5.32D 02 1.10D 03 6.48D 03 3.28D 02 0.0
3.70D 04 1.51D 04 -8.22D 04 8.30D 04 6.01D 04 0.0
7.48D 04 2.53D 04 -5.16D 04 2.81D 04 9.49D 04 0.0
0.0 0.0 0.0 0.0 0.0 3.88D 03

-- ( DV ) --
-2.05D 11 -1.51D 11 1.17D 10 -8.77D 09 -6.35D 10 0.0
1.90D 12 1.28D 12 1.95D 11 1.46D 11 7.68D 11 0.0
1.53D 10 7.64D 09 -1.03D 10 1.64D 10 1.64D 10 0.0
4.07D 11 2.02D 11 -2.60D 11 2.44D 11 3.91D 11 0.0
4.82D 11 2.39D 11 -2.05D 11 1.53D 11 4.05D 11 0.0
0.0 0.0 0.0 0.0 0.0 1.51D 09

```

Figure 6.19 CONFIGURATION VI -- Precalculated Matrices; BMIV, DA, and DV

MATRICES ARE 6X6

--- SYSM ---

2.32D-04	1.09D-05	0.0	0.0	-6.42D-06	0.0
1.09D-05	1.97D-06	0.0	0.0	-1.48D-06	0.0
0.0	0.0	2.01D-04	-2.44D-05	-1.44D-05	-2.59D-05
0.0	0.0	-2.44D-05	6.66D-06	-5.00D-06	0.0
-6.42D-06	-1.48D-06	1.44D-05	-5.00D-06	8.64D-06	0.0
0.0	0.0	-2.59D-05	0.0	0.0	5.18D-05

--- DAMP ---

3.49D 00	3.58D-01	2.09D 00	-3.08D-01	3.16D-01	-6.69D-01
3.58D-01	2.03D-01	1.02D-01	-3.48D-02	2.59D-03	-1.65D-02
2.09D 00	1.02D-01	3.10D 00	-4.83D-01	3.25D-02	2.99D-01
-3.08D-01	-3.48D-02	-4.83D-01	2.86D-01	-8.02D-02	-5.40D-02
3.16D-01	2.59D-03	3.25D-02	-8.02D-02	4.91D-01	2.90D-02
-6.69D-01	-1.65D-02	2.99D-01	-5.40D-02	2.90D-02	1.17D 00

--- STF ---

2.53D 06	1.15D 06	0.0	0.0	1.15D 06	0.0
1.15D 06	7.63D 05	0.0	0.0	3.82D 05	0.0
0.0	0.0	1.39D 06	-5.09D 05	-5.09D 05	7.07D 05
0.0	0.0	-5.09D 05	5.09D 05	-2.54D 05	0.0
1.15D 06	3.82D 05	-5.09D 05	2.54D 05	1.27D 06	0.0
0.0	0.0	-7.07D 05	0.0	0.0	7.07D 05

Figure 6.20 CONFIGURATION VII - Mass, Damping, and Stiffness

CONFIGURATION AND MATRIX LAYER = 7
 MATRICES ARE 6X1

COLUMN MASS	COLUMN DAMPING	COLUMN STIFFNESS
7.77D-05	1.05D 00	-2.36D 05
0.0	-7.37D-03	0.0
0.0	1.72D 00	0.0
0.0	-1.60D-01	0.0
0.0	9.80D-02	0.0
0.0	-6.62D-01	0.0

Figure 6.21 CONFIGURATION VII - Precalculated Matrices; COLM, COLC, and COLK

MATRICES ARE 6X6

-- (BMIV) --

5.65D	03	-2.97D	04	-7.29D	01	-1.24D	03	-1.42D	03	-4.55D	00
-2.97D	04	7.05D	05	5.65D	03	1.33D	05	1.54D	05	2.71D	03
-7.29D	01	5.65D	03	1.01D	04	4.23D	04	8.03D	03	4.97D	03
-1.24D	03	1.33D	05	4.23D	04	4.38D	05	1.93D	05	2.08D	04
-1.42D	03	1.54D	05	8.03D	04	1.93D	05	2.27D	05	3.97D	03
-4.55D	00	2.71D	03	4.97D	03	2.08D	04	3.97D	03	2.15D	04

-- (DA) --

8.83D	03	-3.98D	03	9.07D	03	-9.13D	02	1.10D	03	-3.29D	03
1.67D	05	1.29D	05	-3.08D	04	7.44D	03	5.77D	04	1.04D	04
9.12D	03	6.28D	02	1.32D	04	6.10D	03	1.01D	03	6.76D	03
4.39D	04	1.59D	04	-5.67D	04	8.37D	04	6.17D	04	1.77D	04
7.63D	04	2.54D	04	-4.72D	04	2.81D	04	9.60D	04	1.59D	03
-8.21D	03	-1.21D	01	1.22D	04	1.98D	03	1.07D	03	2.56D	04

-- (DV) --

-2.14D	10	-1.68D	10	1.25D	09	-9.58D	08	-6.96D	09	-5.48D	07
-9.09D	11	5.63D	11	-1.37D	11	1.04D	11	4.62D	11	5.91D	09
1.55D	10	7.29D	09	-8.04D	09	1.84D	10	1.79D	10	1.07D	10
3.71D	11	1.74D	11	-2.48D	11	2.50D	11	3.85D	11	4.46D	10
4.32D	11	2.03D	11	-2.00D	11	1.52D	11	3.91D	11	8.48D	09
7.64D	09	3.58D	09	9.44D	09	9.09D	09	8.85D	09	1.87D	10

Figure 6.22 CONFIGURATION VII - Precalculated Matrices; BMIV, DA, and DV

MATRICES ARE 5X5

--- SYSM ---

```

2.32D-04  1.09D-05  0.0  0.0  -6.42D-06
1.09D-05  1.97D-06  0.0  0.0  -1.48D-06
0.0  0.0  2.01D-04  -2.44D-05  1.44D-05
0.0  0.0  -2.44D-05  6.66D-06  -5.00D-06
-6.42D-06  -1.48D-06  1.44D-05  -5.00D-06  8.64D-06
    
```

--- DAMP ---

```

3.49D 00  3.58D-01  2.09D 00  -3.08D-01  3.16D-01
3.58D-01  2.03D-01  1.02D-01  -3.48D-02  2.59D-03
2.09D 00  -1.02D-01  3.10D 00  -4.83D-01  3.25D-02
-3.08D-01  -3.48D-02  -4.83D-01  2.86D-01  -8.02D-02
3.16D-01  2.59D-03  3.25D-02  -8.02D-02  4.91D-01
    
```

--- STF ---

```

2.53D 06  1.15D 06  0.0  0.0  1.15D 06
1.15D 06  7.63D 05  0.0  0.0  3.82D 05
0.0  0.0  1.39D 06  -5.09D 05  -5.09D 05
0.0  0.0  -5.09D 05  5.09D 05  2.54D 05
1.15D 06  3.82D 05  -5.09D 05  2.54D 05  1.27D 06
    
```

Figure 6.23 CONFIGURATION VIII - Mass, Damping, and Stiffness

CONFIGURATION AND MATRIX LAYER = 8

MATRICES ARE 5X1

COLUMN MASS	COLUMN DAMPING	COLUMN STIFFNESS
7.77D-05	1.05D 00	-2.36D 05
0.0	-7.37D-03	0.0
0.0	1.72D 00	0.0
0.0	-1.60D-01	0.0
0.0	9.80D-02	0.0

CONFIGURATION = 8, MATRIX LAYER = 3

MATRICES ARE 5X1

COLUMN MASS	COLUMN DAMPING	COLUMN STIFFNESS
0.0	-6.69D-01	0.0
0.0	-1.65D-02	0.0
-2.59D-05	2.99D-01	7.07D 05
0.0	-5.40D-02	0.0
0.0	2.90D-02	0.0

Figure 6.24 CONFIGURATION VIII - Pre-calculated Matrices; COLM, COLC, and COLK

MATRICES ARE 5X5

- (BMIV) -

```

5.65D 03 -2.97D 04 -7.19D 01 -1.24D 03 -1.42D 03
-2.97D 04 -7.05D 05 5.02D 03 1.31D 05 1.54D 05
-7.19D 01 5.02D 03 9.00D 03 3.75D 04 7.11D 03
-1.24D 03 1.31D 05 3.75D 04 4.18D 05 1.90D 05
-1.42D 03 1.54D 05 7.11D 03 1.90D 05 2.26D 05
    
```

- (DA) -

```

8.83D 03 -3.98D 03 9.07D 03 -9.12D 02 1.10D 03
1.68D 05 1.29D 05 -3.23D 04 7.19D 03 5.75D 04
1.10D 04 6.31D 02 1.04D 04 5.64D 03 7.64D 02
5.19D 04 1.59D 04 -6.85D 04 8.18D 04 6.07D 04
7.78D 04 2.54D 04 -4.95D 04 2.77D 04 9.58D 04
    
```

- (DV) -

```

-2.14D 10 -1.68D 10 1.26D 09 -9.56D 08 -6.96D 09
-9.08D 11 5.63D 11 -1.38D 11 1.03D 11 4.61D 11
1.37D 10 6.47D 09 -1.02D 10 1.63D 10 1.58D 10
3.64D 11 1.71D 11 -2.57D 11 2.42D 11 3.77D 11
4.31D 11 2.02D 11 -2.02D 11 1.50D 11 3.89D 11
    
```

Figure 6.25 CONFIGURATION VIII -- Precalculated Matrices; BMIV, DA, and DV

C. SAMPLE PROBLEMS

One set of results for 11,000 RPM was obtained in CALCOMP graph form and as AGT graphic display photographs. At this high speed of cam rotation, the pushrod is frequently thrown off the cam. The results are completely analyzed with numerous reference points plotted on the graphs.

The second example problem was chosen at a slower speed, 9,000 RPM. At this speed the pushrod remains in contact with the cam. A third intermediate example, 10,000 RPM was chosen for a comparison between the two above examples. Figure 6.26 shows the relationship between the sample problems.

Photographs from the AGT display are presented after the CALCOMP graphs for each case of example one. It is noted that large scale factors are introduced for both bending and axial deformation as an aid to the user.

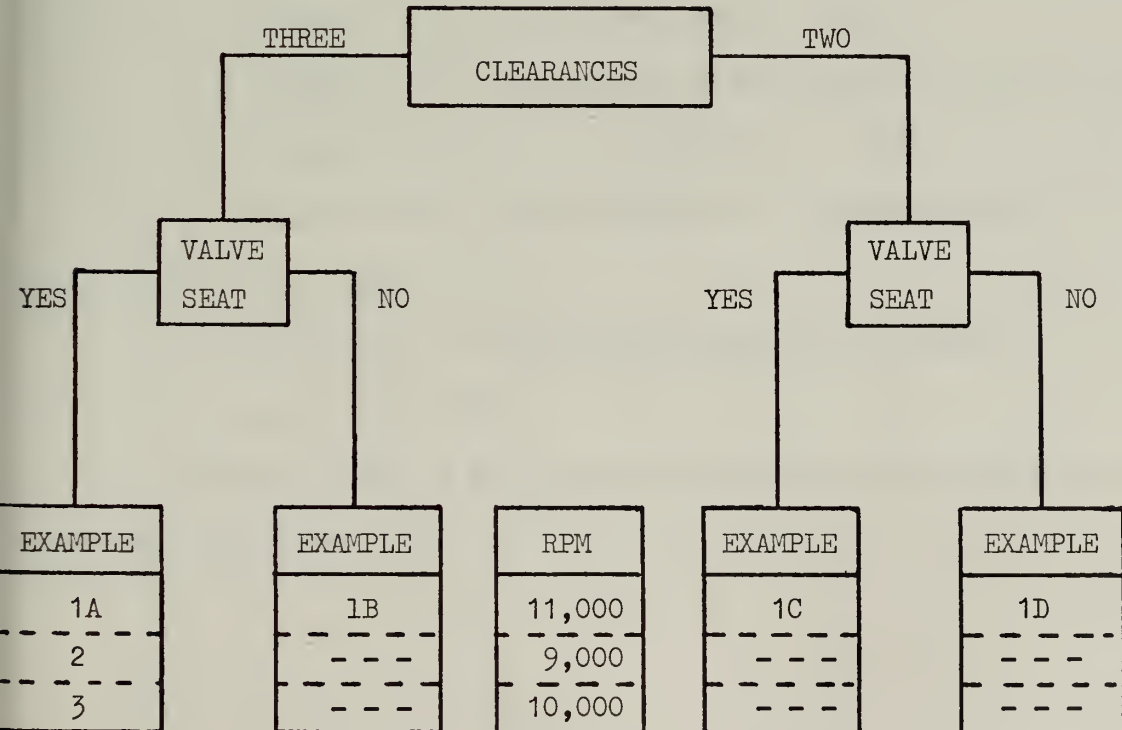


Figure 6.26 SAMPLE NUMERICAL RESULTS

1. Example 1A, Three Clearance, 11,000 RPM, Valve Seat

This analysis will commence at 270 degrees of cam rotation.

One cycle will be discussed, from point (A) to point (AA). The following figures are utilized for the analysis:

a. Figure 6.27

(1) Displacement difference between cam and pushrod

$$(q_1 - q_9) \times 10^{-2}$$

(2) Displacement of lower end of pushrod i.e. follower,

$$q_1 \times 10^{-1}$$

(3) Cam Profile $q_9 \times 10^{-1}$

b. Figure 6.28

(1) Displacement difference between pushrod and rocker

$$(q_2 - q_8) \times 10^{-2}$$

(2) Displacement difference between cam and pushrod

$$(q_1 - q_9) \times 10^{-2}$$

c. Figure 6.29 Force on the Pushrod $F_1 \times 10^3$

d. Figure 6.30 Force on the Rocker $F_2 \times 10^2$

e. Figure 6.31 Displacement of the Valve $q_7 \times 10^{-1}$

f. Figure 6.32 Force on the Valve $F_7 \times 10^3$

g. Figure 6.33 Axial Oscillation of the Valve Stem

$$(q_7 - q_6) \times 10^{-3}$$

h. Figure 6.34 Axial Oscillations of the Pushrod

$$(q_8 - q_1) \times 10^{-3}$$

Figures 6.33 and 6.34 have been included for completeness and are not annotated.

The following reference points approximately correspond to the indicated degrees of cam rotation, Table 6.1

Reference Point	Degrees of Cam Rotation	Reference Point	Degrees of Cam Rotation
A	270	I	380
B	288	J	418
C	294	K	430
D	300	L	448
E	336	M	459
F	346	N	579
G	357	P	583
H	363	AA	630

Table 6.1 Graph Reference Points and Degrees of Cam Rotation for Figures 6.27 - 6.32

At point (A), the cam is at its lowest extreme and is starting to rise. The pushrod is floating between the cam and rocker, with downward motion. The valve has just undergone a severe oscillation and has resumed its upward motion.

At point (B), the cam and pushrod impact and remain in contact for a short period, (B-C). The rising cam and force of impact create severe oscillation of the pushrod and a bounce occurs. The pushrod is again thrown clear of the cam and floats (C-E).

At point (D), the valve has bounced against the valve seat, creating severe oscillation and more bouncing of the valve (D-E). The pushrod is floating thus is not affected by this impact.

At point (E), the rising pushrod now impacts against the rocker, and valve against the valve seat. At point (E), the valve has again reseated and the rocker has become almost stationary. This allows the rising pushrod to impact against the rocker.

The pushrod bounces off the rocker (F-H) and again floats (F-H), now having downward motion and also creating downward motion of the valve.

The rising cam almost immediately contacts the pushrod (G). The resulting motion creates positive contact between cam, pushrod, and rocker (H-I), and the valve bounces cease and it resumes its downward harmonic motion. At point (I), the speed of the cam throws the pushrod clear. Immediately separation occurs at the rocker- pushrod due to inertial effects. Several bounces at this clearance are observed (J-K-L-M), until the system reaches its extreme, point (M), i.e. the cam now with downward motion begins falling away from the pushrod. The valve spring, having reached the maximum extension, reverses the motion of the valve upward. The rocker and pushrod are now in contact (M) and both have downward motion.

At point (N), the valve impacts against the seat which results in severe oscillations and valve bounce (P-AA). With the gross motion of the rocker halted, the pushrod floats clear (P-AA). The cycle then repeats.

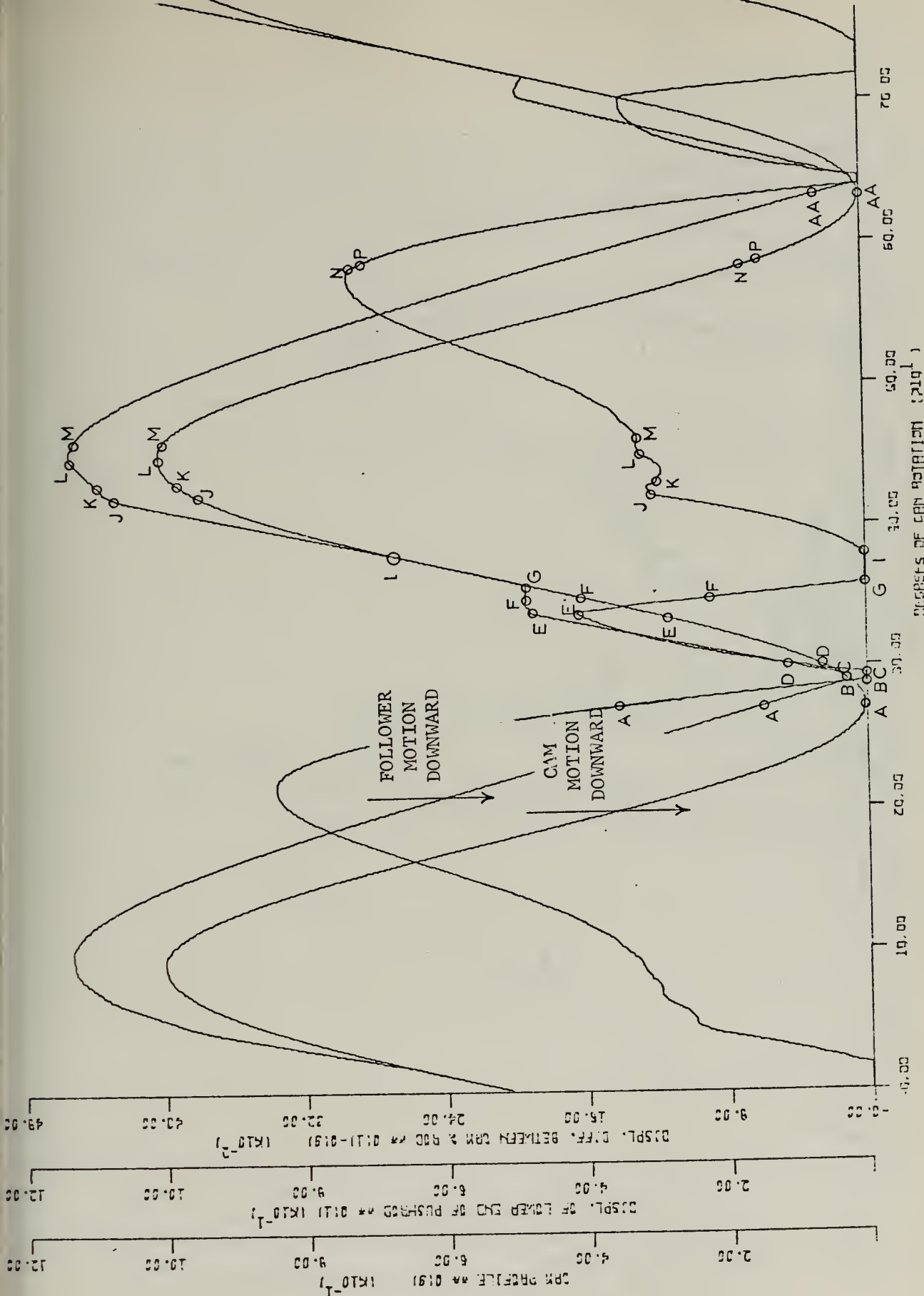


Figure 6.27 THREE CLEARANCES 11.000 RPM VALVE SEAT

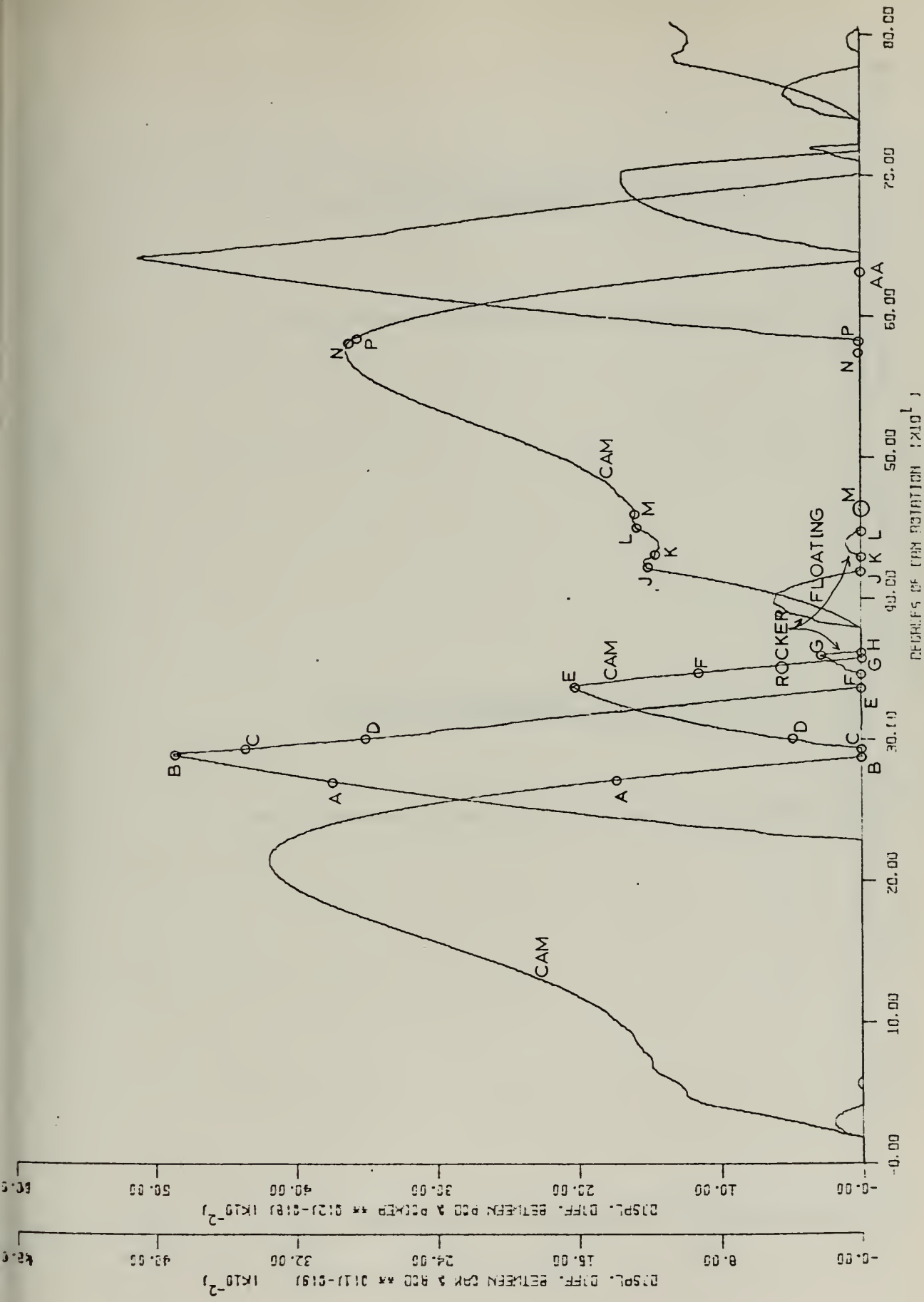


Figure 6.28 THREE CLEARANCES 11,000 RPM VALVE SEAT

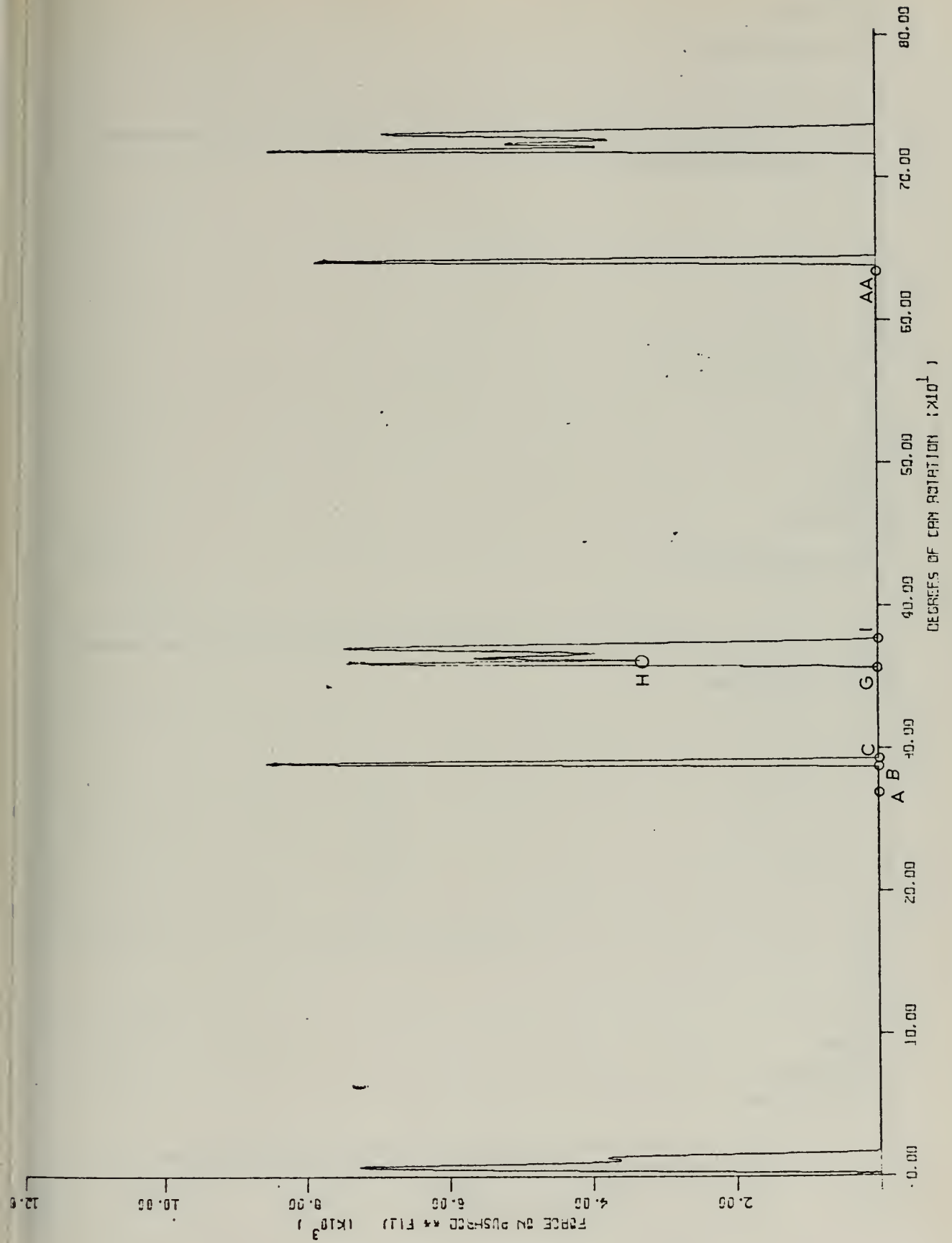


Figure 6.29 THREE CLEARANCES 11,000 RPM VALVE SEAT

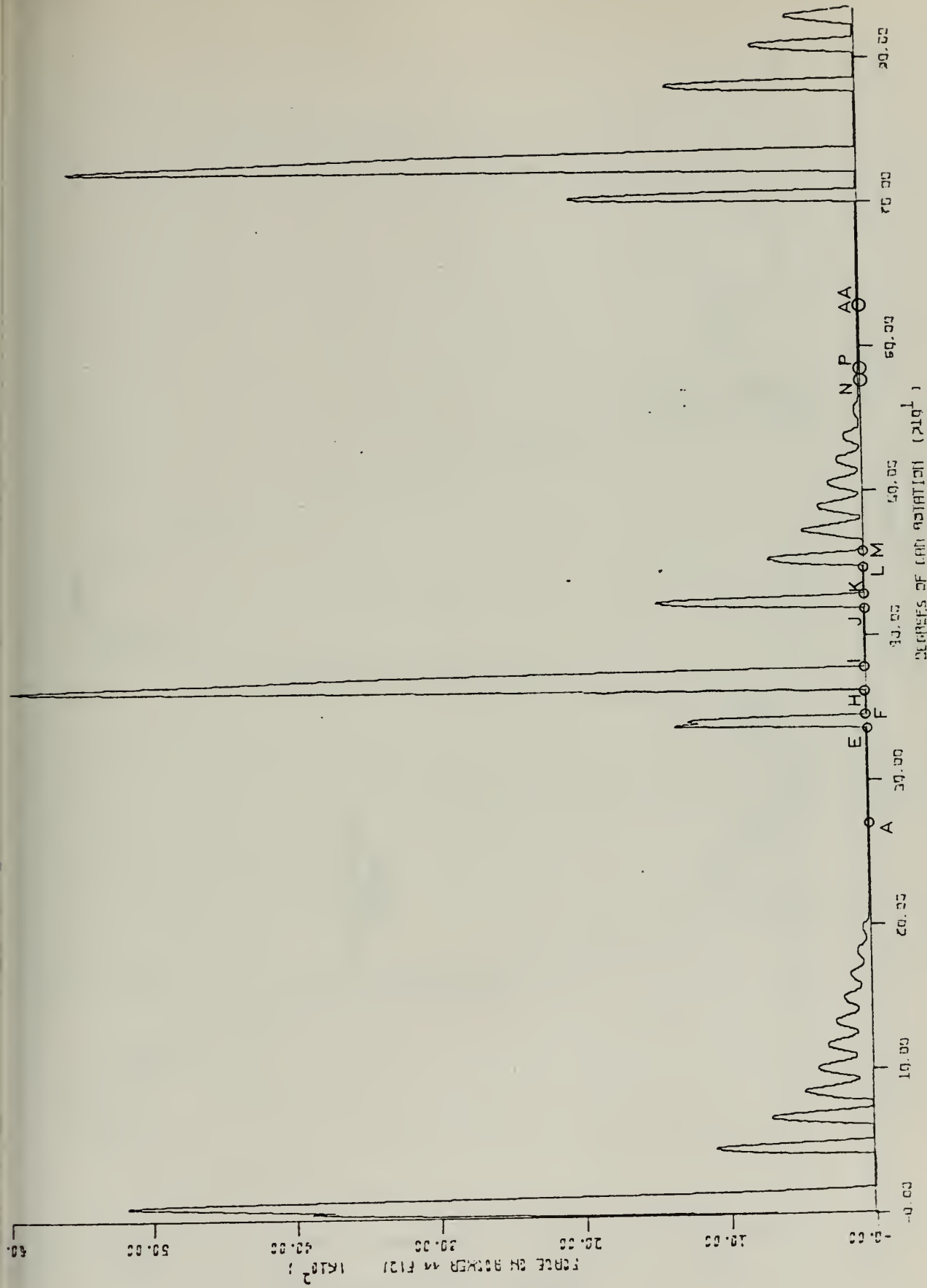


Figure 6.30 THREE CLEARANCES 11,000 RPM VALVE SEAT

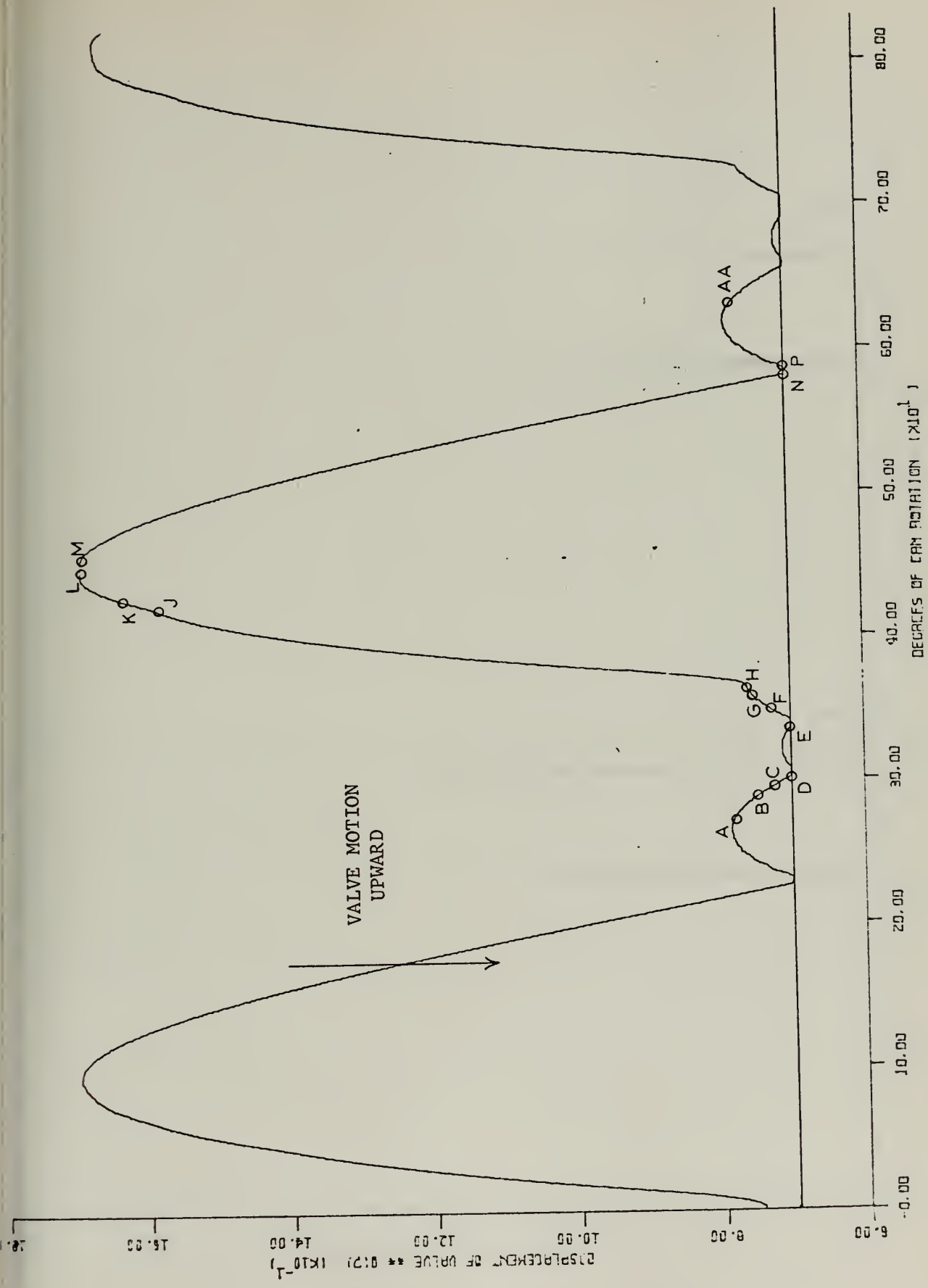
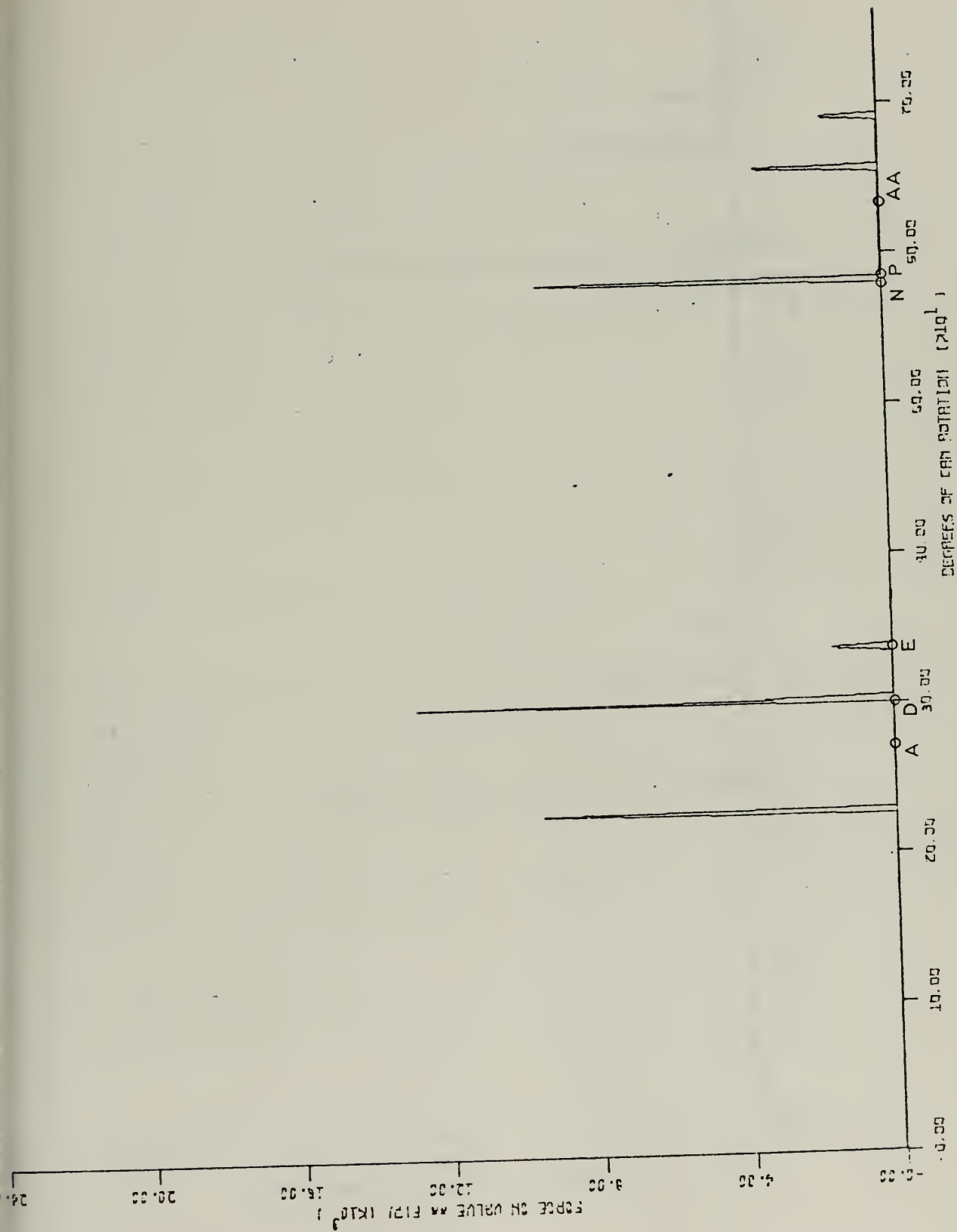


Figure 6.31 THREE CLEARANCES 11,000 RPM VALVE SEAT



11,000 RPM VALVE SEAT

Figure 6.32 THREE CLEARANCES

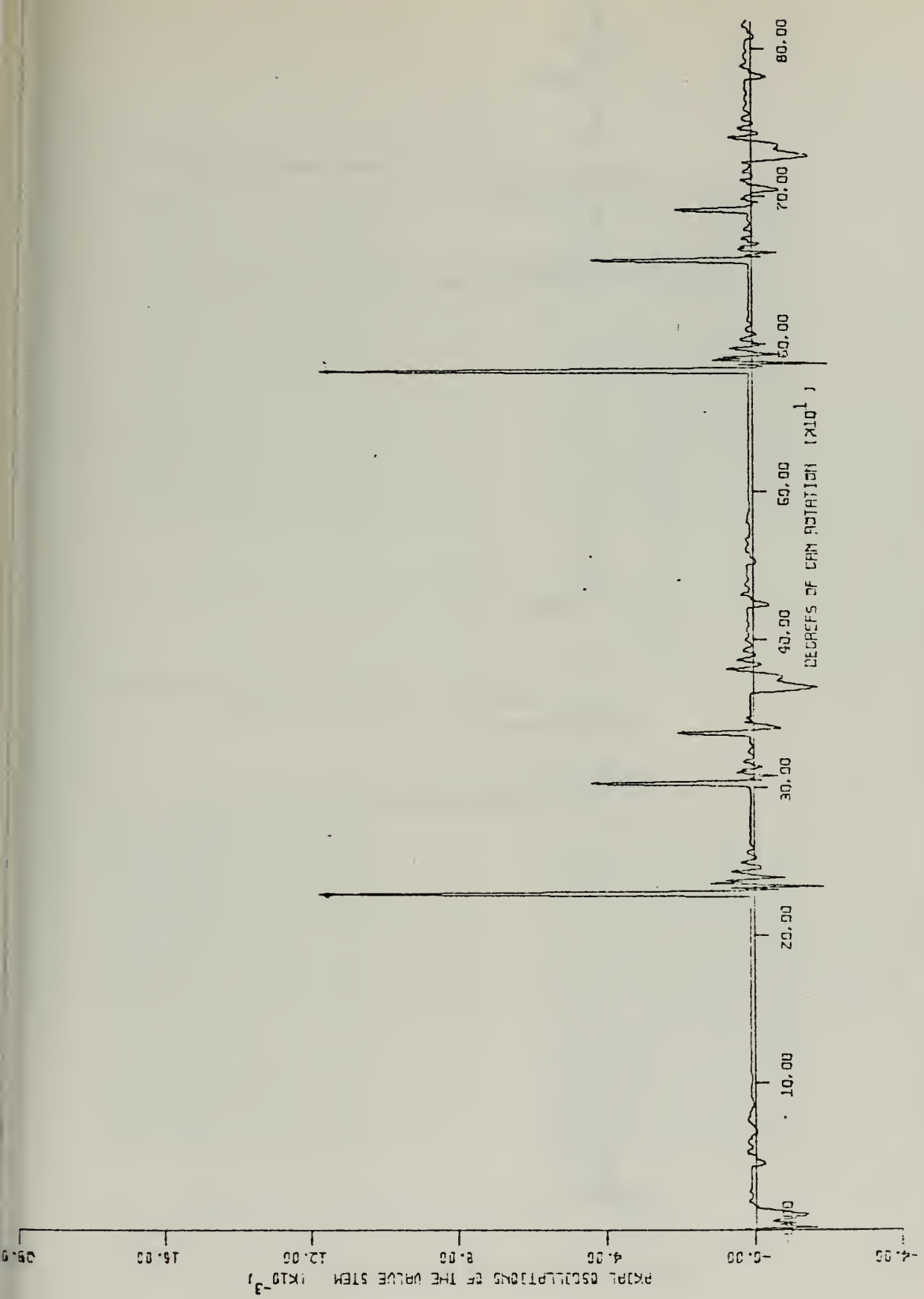


Figure 6.33 THREE CLEARANCES 11,000 RPM VALVE SEAT

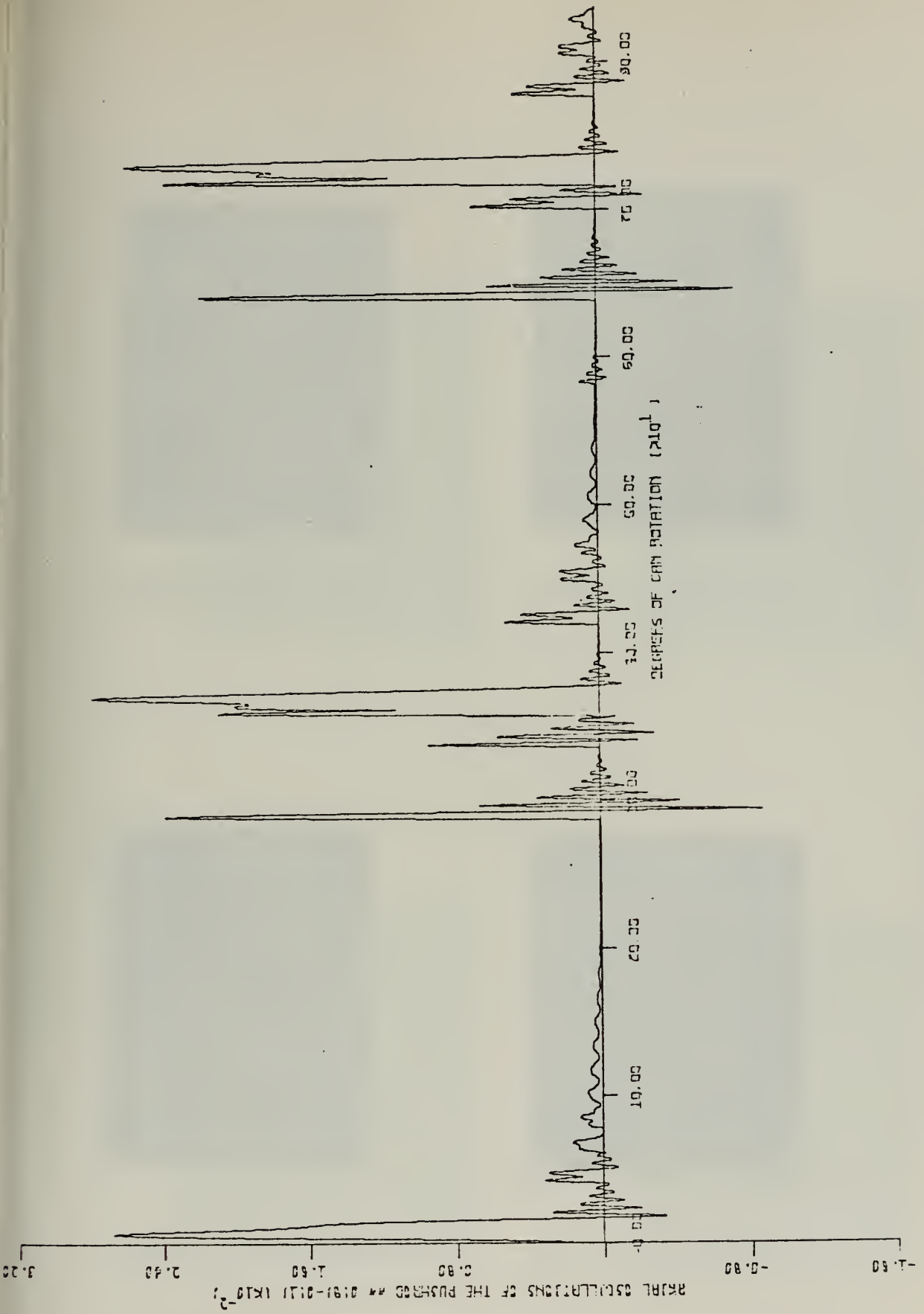


Figure 6.34 THREE CLEARANCES 11,000 RPM VALVE SEAT

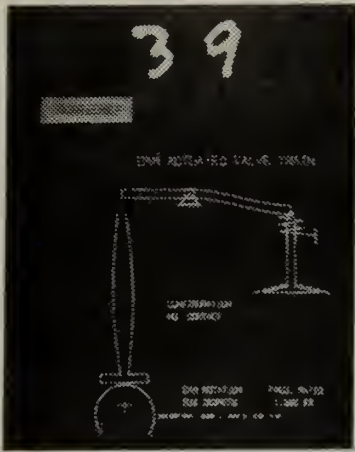


Photo 6.2 - 630 degrees

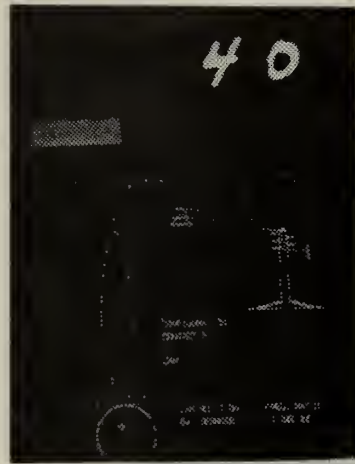


Photo 6.3 - 641 degrees



Photo 6.4 - 650 degrees

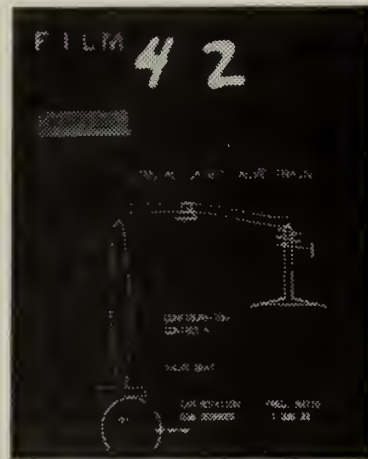


Photo 6.5 - 658 degrees

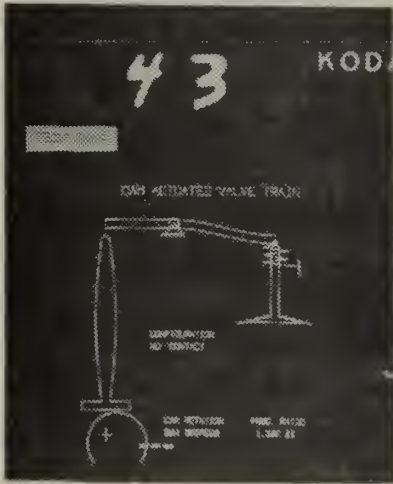


Photo 6.6 - 664 degrees

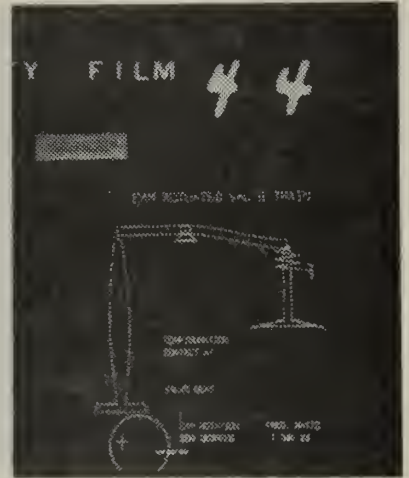


Photo 6.7 - 682 degrees

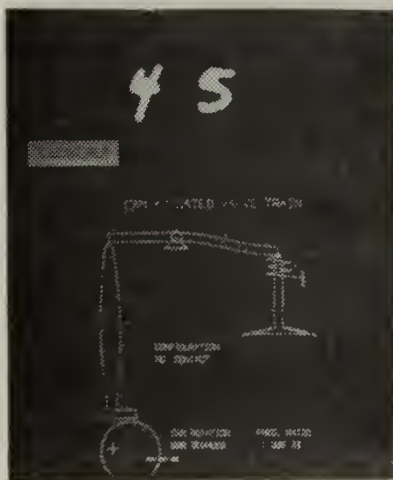


Photo 6.8 - 686 degrees

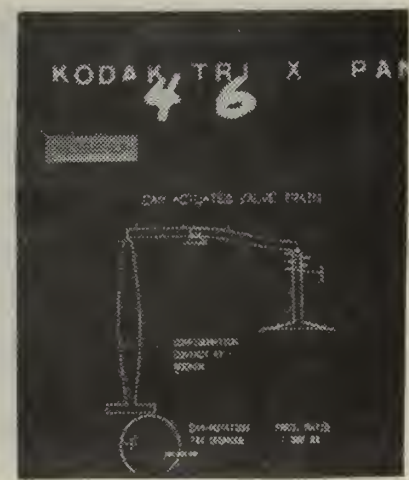


Photo 6.9 - 704 degrees

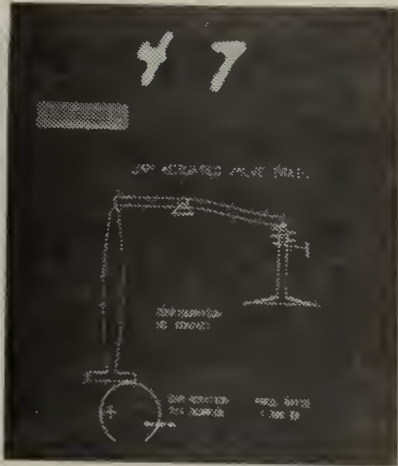


Photo 6.10 - 714 degrees

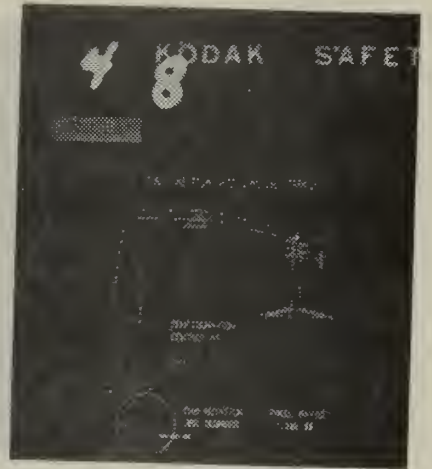


Photo 6.11 - 362 degrees

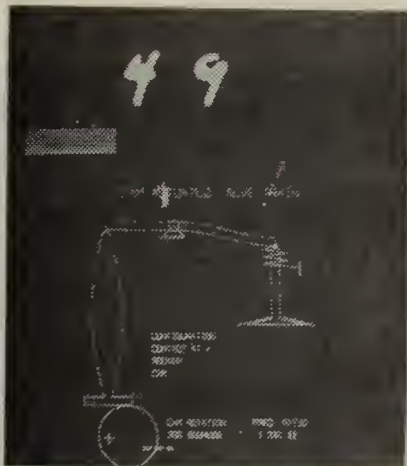


Photo 6.12 - 366 degrees

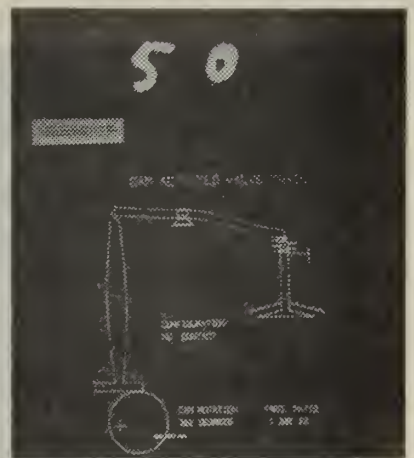


Photo 6.13 - 380 degrees

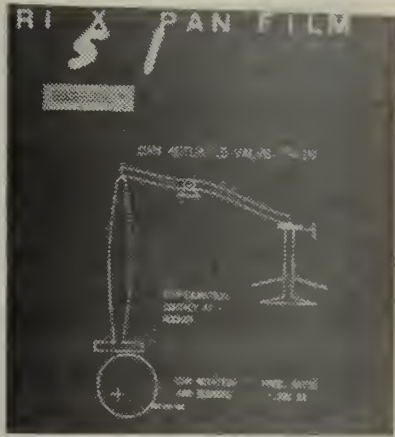


Photo 6.14 - 422 degrees

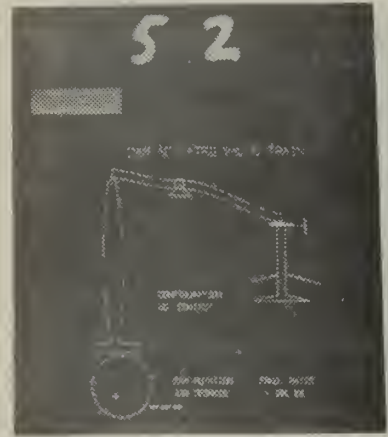


Photo 6.15 - 432 degrees

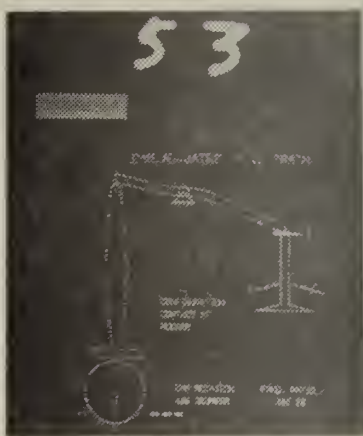


Photo 6.16 - 456 degrees

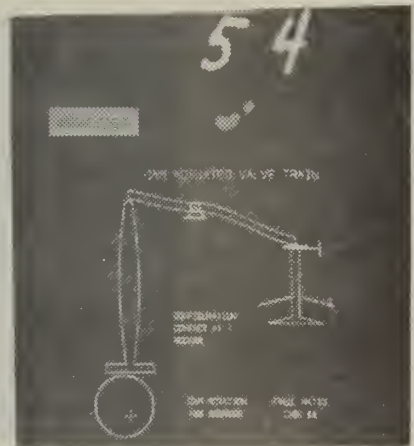


Photo 6.17 - 468 degrees

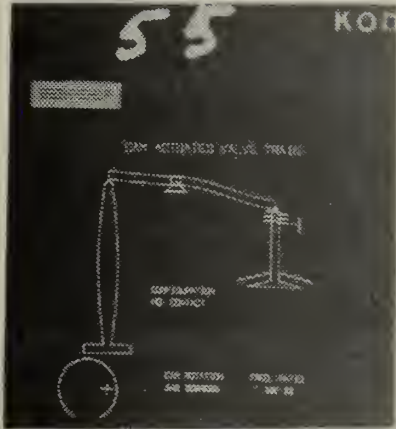


Photo 6.18 - 546 degrees

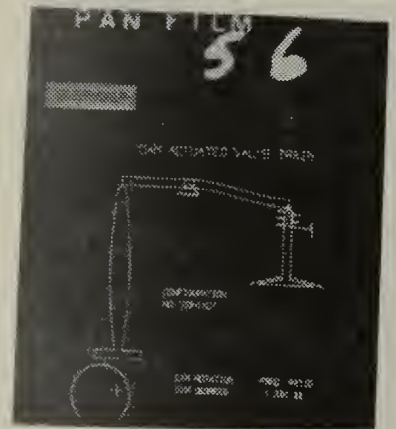


Photo 6.19 - 572 degrees

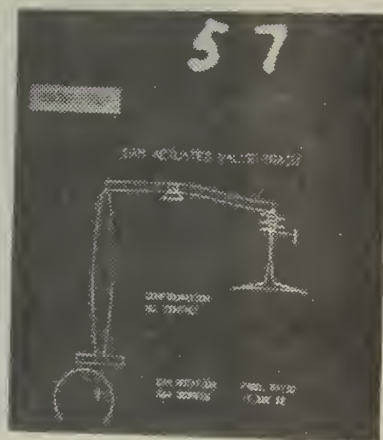


Photo 6.20 - 582 degrees

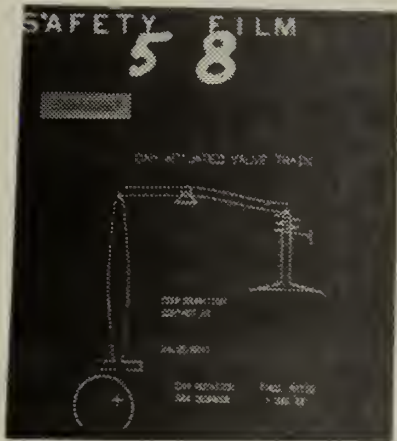


Photo 6.21 - 584 degrees

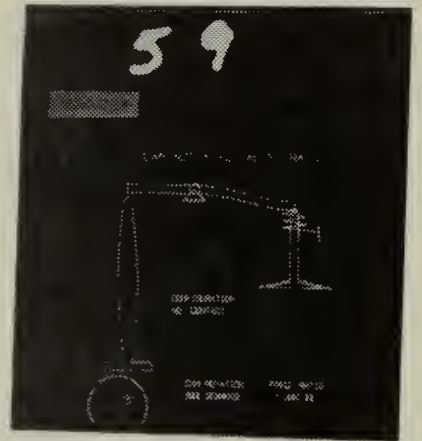


Photo 6.22 - 600 degrees

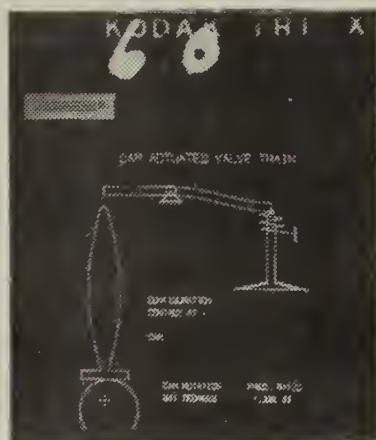


Photo 6.23 - 641 degrees

2. Example 1B, Three Clearances, 11,000 RPM, No Valve Seat

This analysis starts at 274 degrees of cam rotation, point

(A). The following figures are utilized for the analysis.

a. Figure 6.35

(1) Displacement difference between cam and pushrod

$$(q_1 - q_9) \times 10^{-2}$$

(2) Displacement of lower end of pushrod, i.e. follower

$$q_1 \times 10^{-1}$$

(3) Cam profile, $q_9 \times 10^{-1}$

b. Figure 6.36

(1) Displacement difference between pushrod and rocker

$$(q_2 - q_8) \times 10^{-2}$$

(2) Displacement difference between cam and pushrod

$$(q_1 - q_9) \times 10^{-2}$$

c. Figure 6.37 Forces on the pushrod, $F_1 \times 10^3$

d. Figure 6.38 Forces on the rocker, $F_2 \times 10^2$

e. Figure 6.39 Displacement of Valve, $q_7 \times 10^2$

f. Figure 6.40 Axial oscillation of the pushrod,

$$(q_8 - q_1) \times 10^{-3}$$

Figure 6.40 has been included for completeness and is not annotated.

The following reference points approximately correspond to the indicated degrees of cam rotation, Table 6.2

Reference Point	Degrees of Cam Rotation	Reference Point	Degrees of Cam Rotation
A	274	H	401
B	288	I	413
C	295	J	419
D	314	K	449
E	358	L	581
F	364	AA	634
G	382		

Table 6.2 Graph Reference Points and Degrees of Cam Rotation For Figures 6.35-6.39.

The analysis is similar to example 1A, except here the motion of the system is not impeded by the valve seat.

At point (A), the cam is at its lowest extreme and is starting to rise. The pushrod is floating between the cam and rocker, with downward motion.

At point (B), the cam and pushrod impact against one another and remain in contact for a short period, (B-C). The rising cam and force of impact create severe oscillations of the pushrod and a bounce occurs. The system remains intact, i.e. contact at the cam and rocker, (C-D). The valve has reached its highest extreme and reverses motion downwards.

At point (D), the pushrod and cam separate. Immediately, contact is also lost at the rocker and the pushrod floats free, (D-E).

At point (E), the rising pushrod now impacts against the rocker. This slows the motion of the pushrod sufficiently for the rising cam to again impact against the rocker, point (F). The system then remains in contact for a short period (F-G).

At point (G), the speed of the cam throws the pushrod clear. Immediately separation also occurs at the rocker - pushrod due to inertial effects. The pushrod floats free for a short time, (G-H), and then bounces at the rocker-pushrod clearance, point (I). The system next reaches its extreme value, point (K), i.e. the cam now with downward motion begins falling away from the pushrod. The valve spring, having reached the maximum extension, reverses the harmonic motion of the valve, upward. The rocker and pushrod are now in contact (J), and both have downward harmonic motion.

Finally, the compression of the valve spring slows the rocker valve assembly until the pushrod loses contact at the rocker and again floats free, point (L).

At point (AA), 634 degrees of cam rotation, the cycle then repeats.

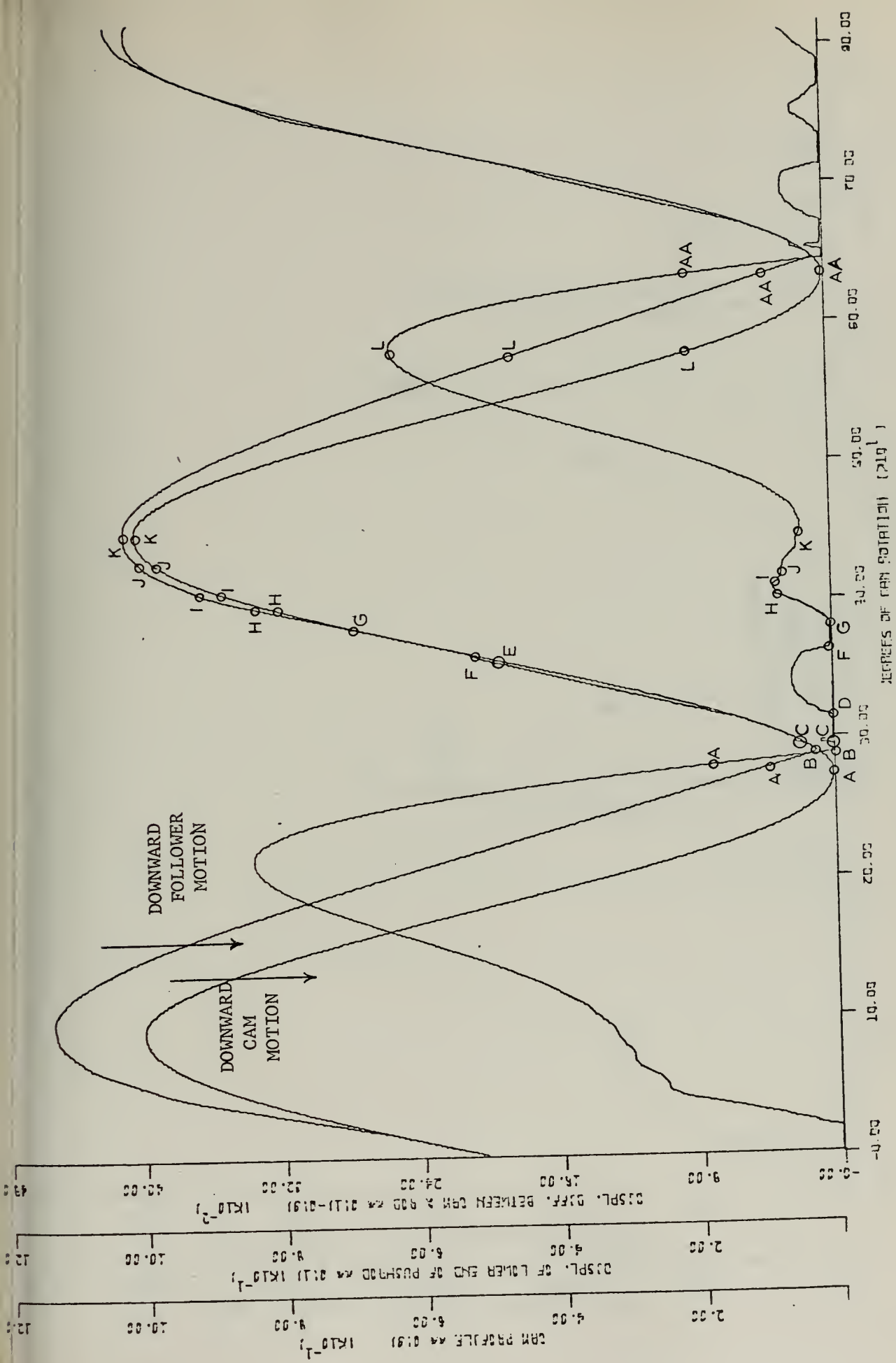


Figure 6.35 THREE CLEARANCES 11,000 RPM NO VALVE SEAT

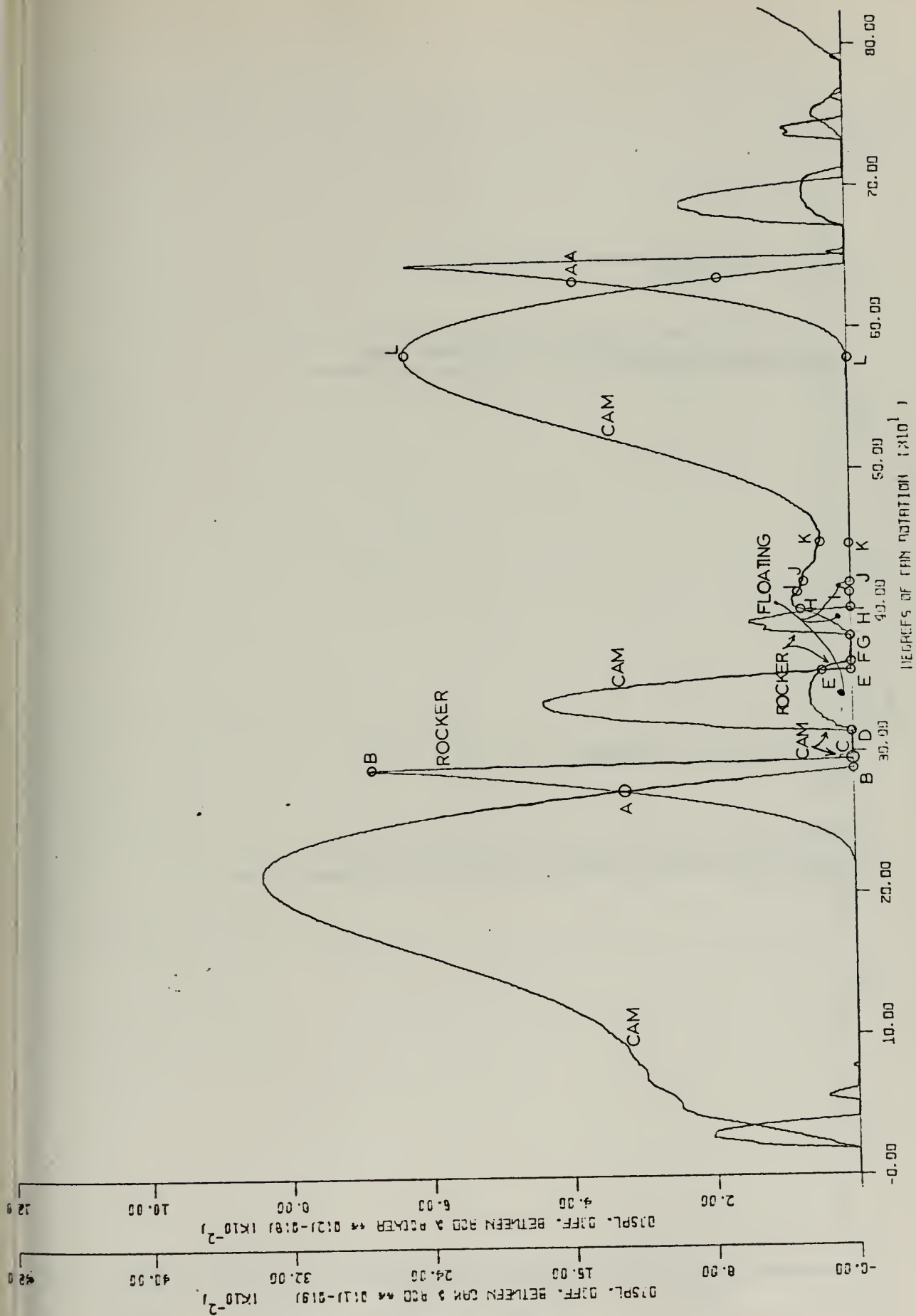
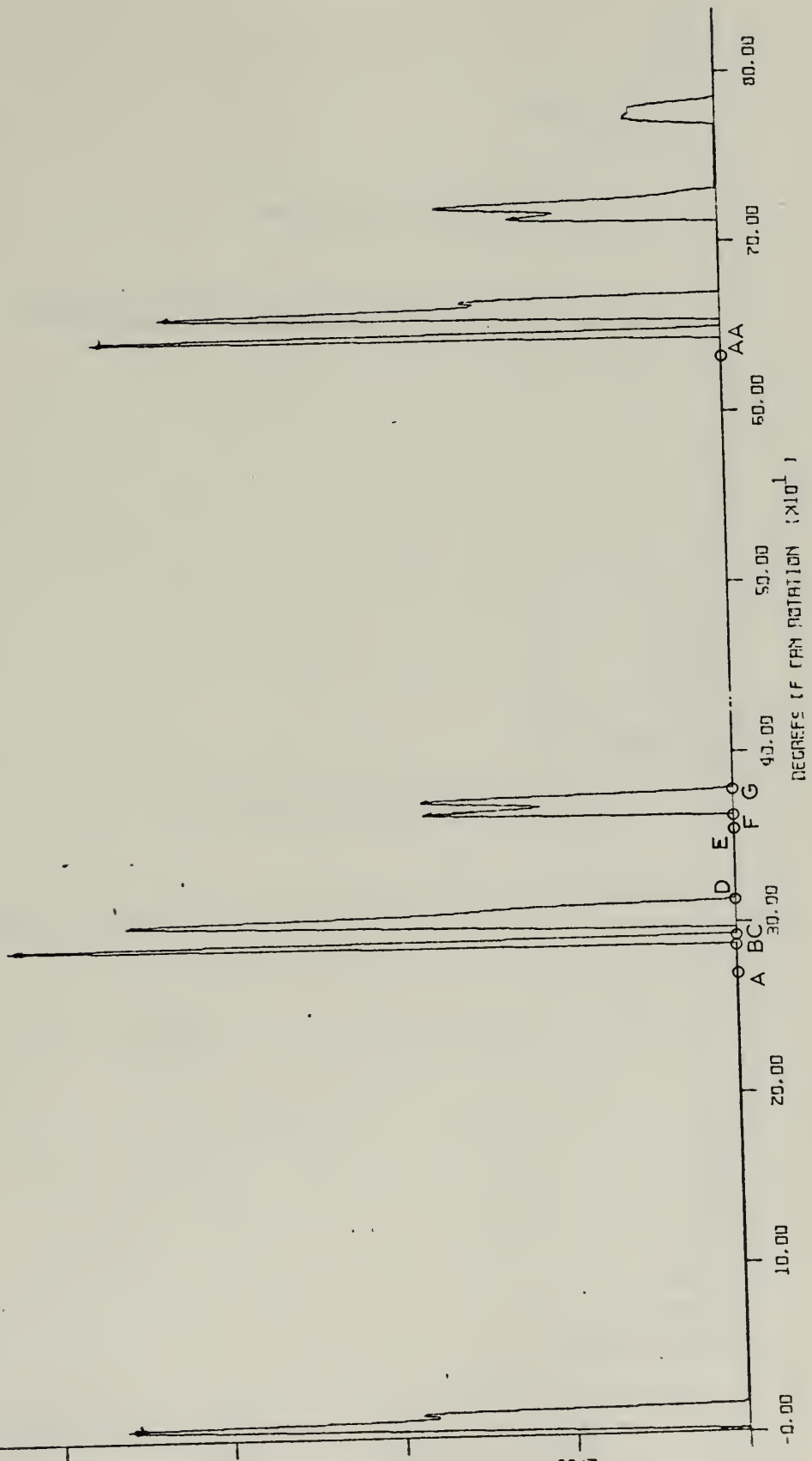


Figure 6.36 THREE CLEARANCES 11,000 RPM NO VALVE SEAT

FORCE ON PUSHERS $\times 10^3$ L



11,000 RPM NO VALVE SEAT

THREE CLEARANCES

Figure 6.37

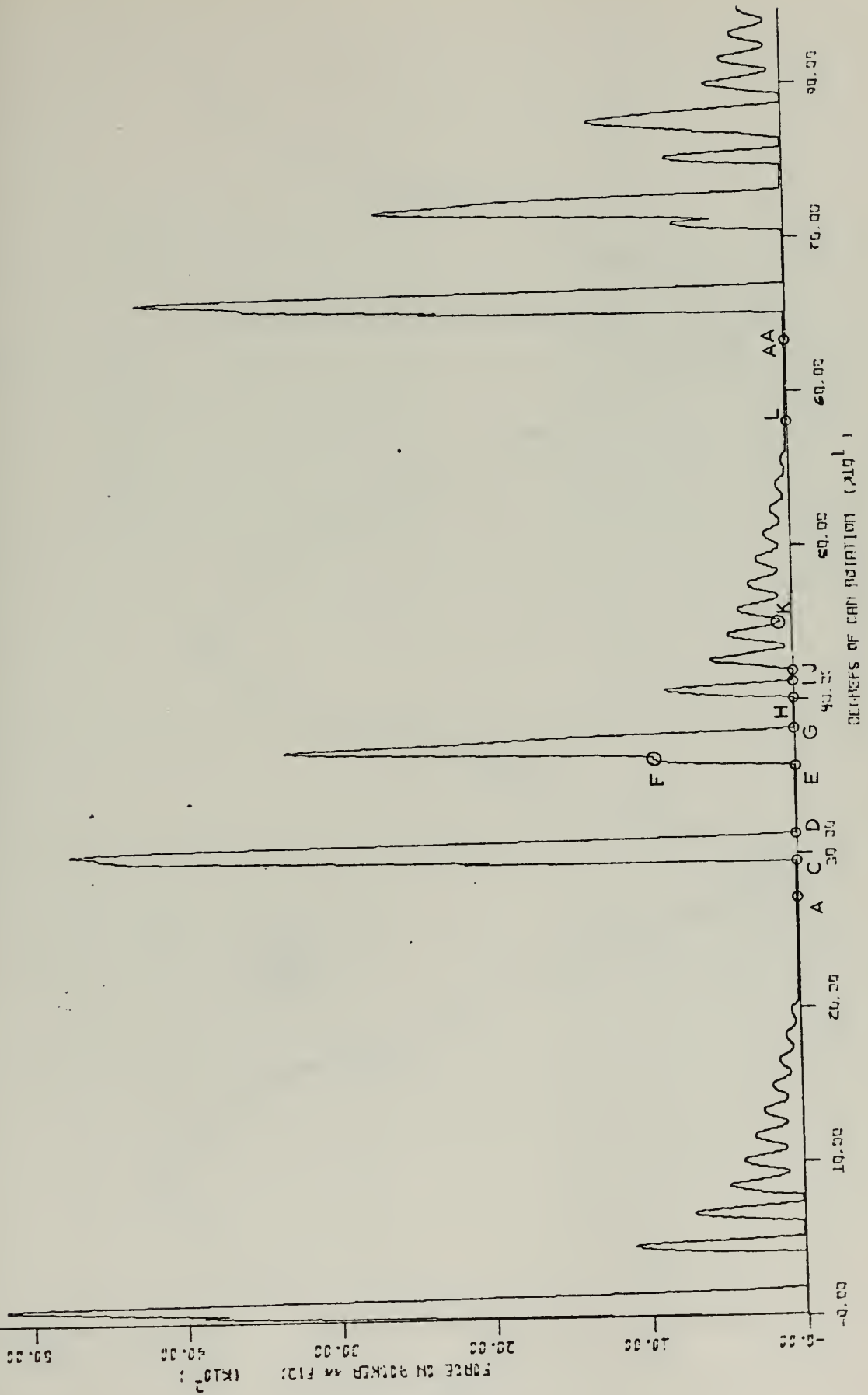


Figure 6.38 THREE CLEARANCES 11.000 RPM NO VALVE SEAT

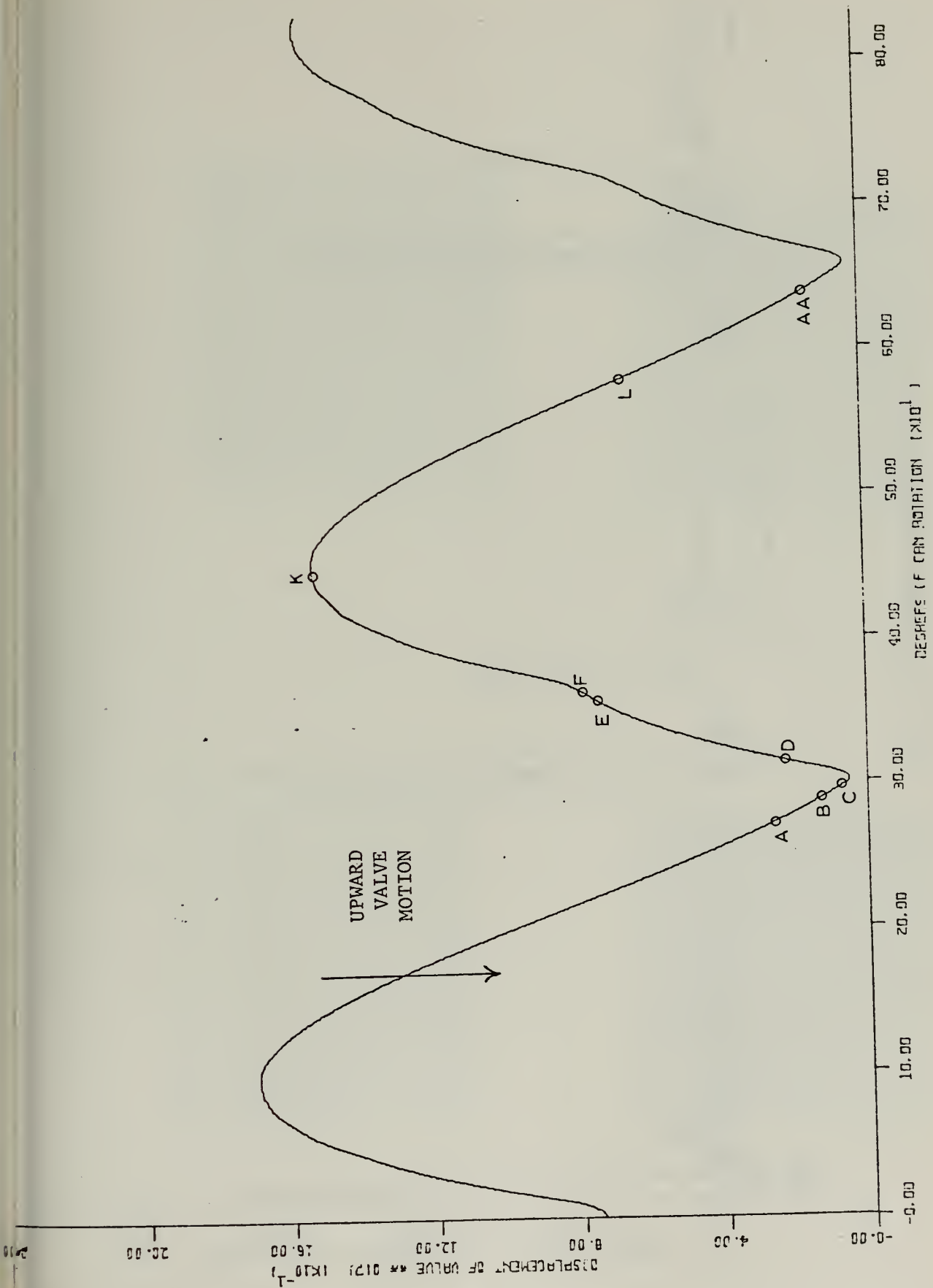


Figure 6.39 THREE CLEARANCES 11,000 RPM NO VALVE SEAT

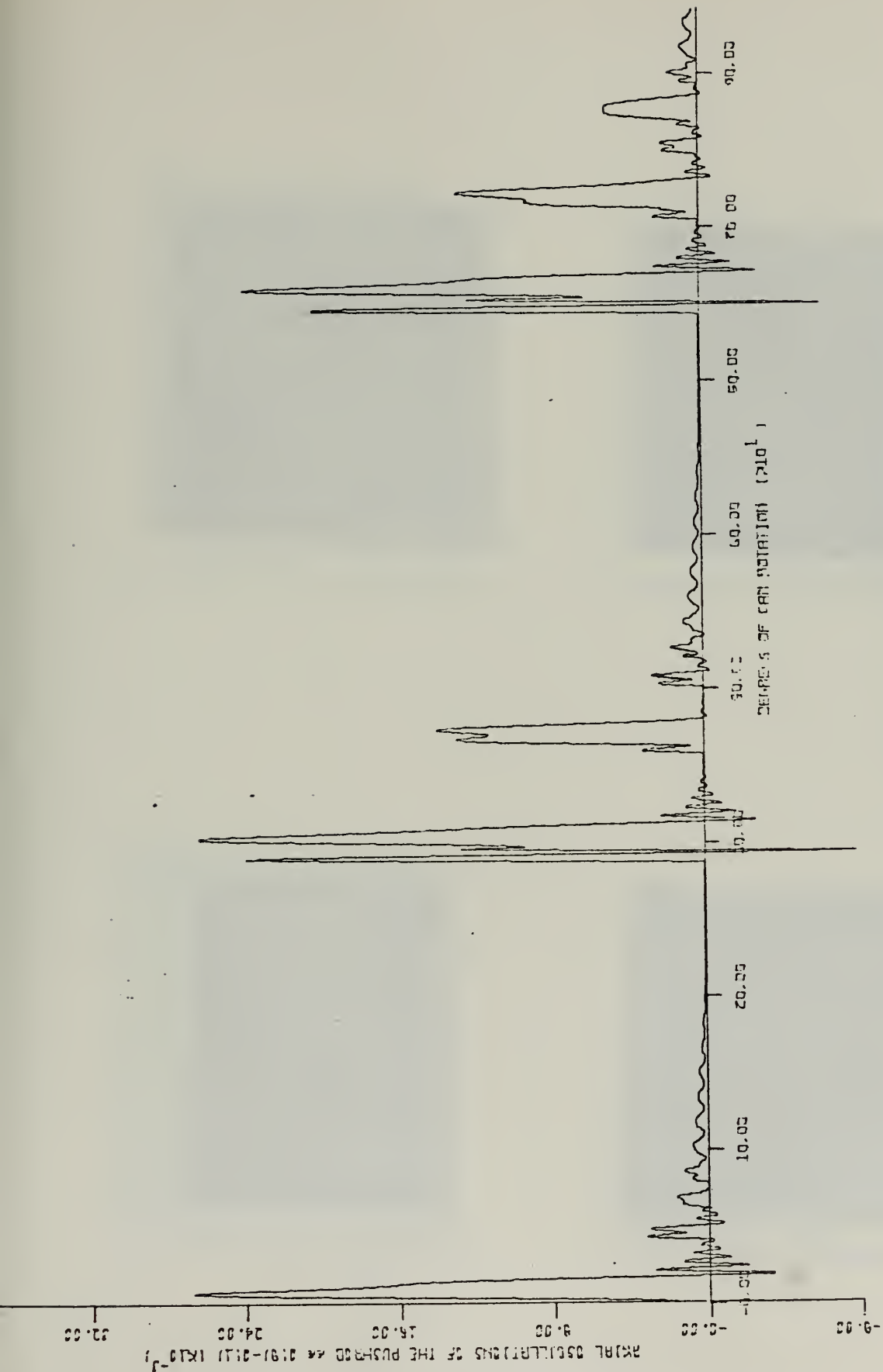


Figure 6.40 THREE CLEARANCES
 11,000 RPM NO VALVE SEAT

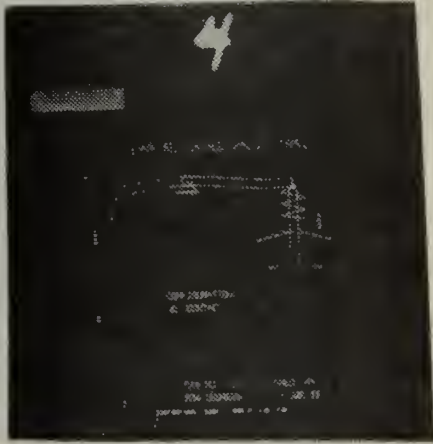


Photo 6.24 - 634 degrees

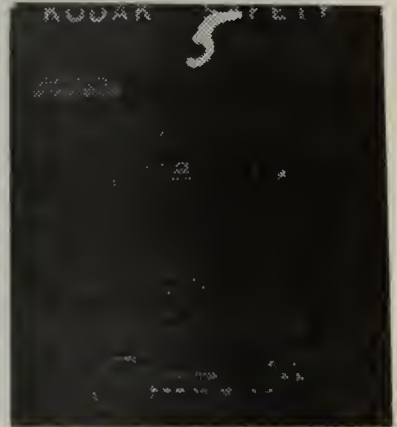


Photo 6.25 - 646 degrees

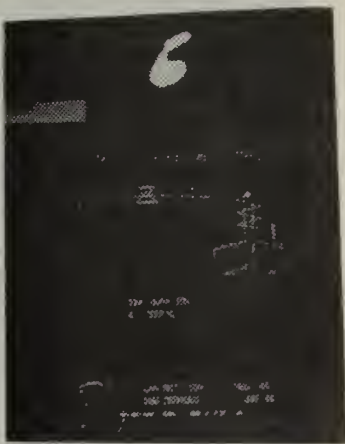


Photo 6.26 - 652 degrees

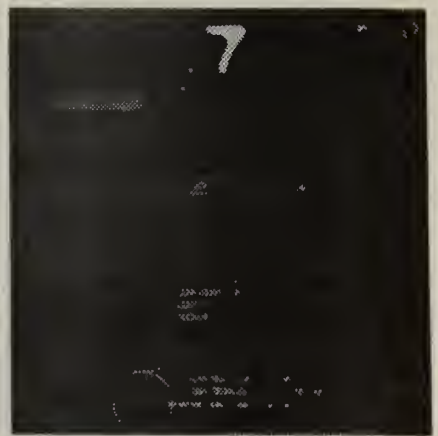


Photo 6.27 - 654 degrees

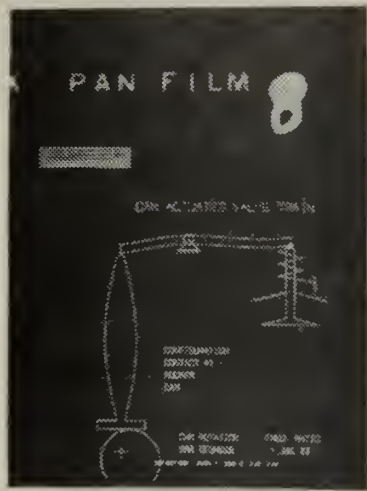


Photo 6.28 - 658 degrees

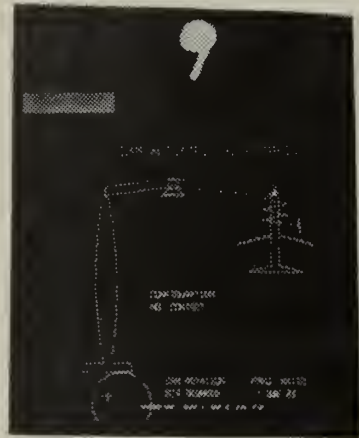


Photo 6.29 - 674 degrees



Photo 6.30 - 708 degrees

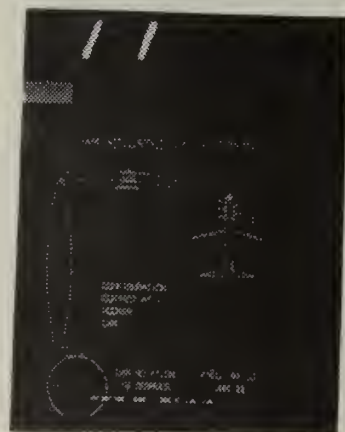


Photo 6.31 - 716 degrees

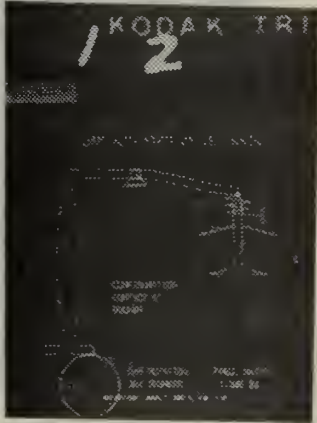


Photo 6.32 - 364 degrees

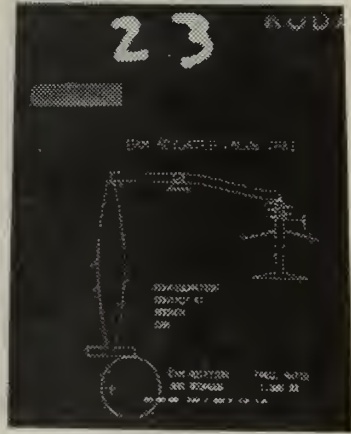


Photo 6.33 - 366 degrees



Photo 6.34 - 382 degrees

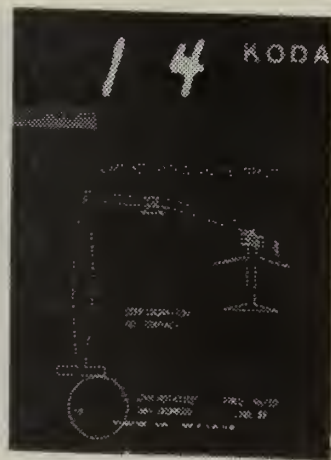


Photo 6.35 - 384 degrees

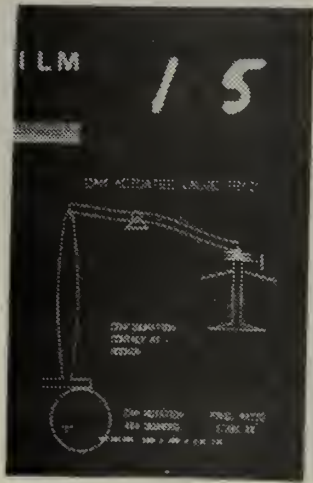


Photo 6.36 - 404 degrees

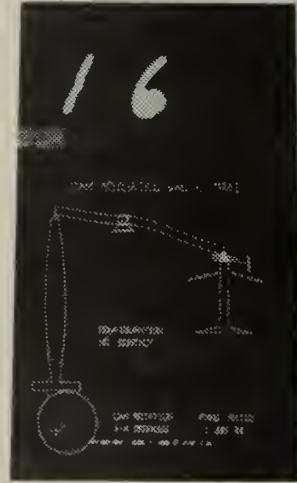


Photo 6.37 - 414 degrees

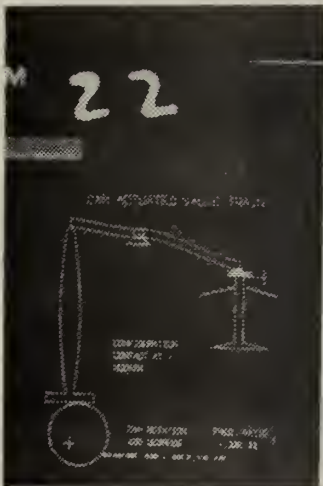


Photo 6.38 - 422 degrees



Photo 6.39 - 450 degrees

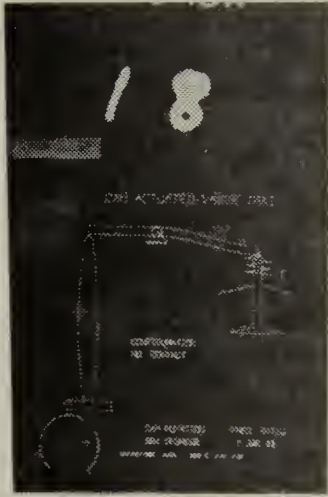


Photo 6.40 - 464 degrees

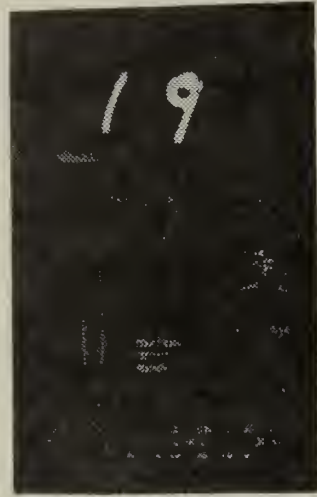


Photo 6.41 - 572 degrees

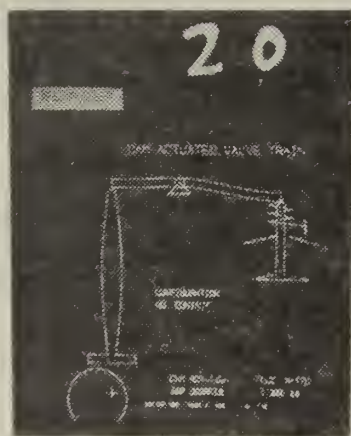


Photo 6.42 - 582 degrees

3. Example 1C, Two Clearances, 11,000 RPM, Valve Seat

This analysis will commence at 333 degrees of cam rotation. One cycle will be discussed, from point (A) to point (AA). The following figures are utilized for the analysis:

- a. Figure 6.41 Cam profile and displacement of pushrod,
 $q_9 \times 10^{-1}$ and $q_1 \times 10^{-1}$
- b. Figure 6.42
 - (1) Cam profile, $q_9 \times 10^{-1}$
 - (2) Displacement difference between cam and pushrod,
 $(q_1 - q_9) \times 10^{-2}$
- c. Figure 6.43 Force on pushrod, $F_1 \times 10^3$
- d. Figure 6.44 Force on the valve, $F_7 \times 10^3$
- e. Figure 6.45 Displacement of valve and valve seat,
 $q_7 \times 10^{-1}$ and $q_{10} \times 10^{-1}$
- f. Figure 6.46 Axial oscillation of the pushrod,
 $(q_8 - q_1) \times 10^{-2}$
- g. Figure 6.47 Axial oscillation of the valve stem,
 $(q_7 - q_6) \times 10^{-3}$

Figures 6.46 and 6.47 have been included for completeness and are not annotated.

The following reference points approximately correspond to the indicated degrees of cam rotation, Table 6.3.

Reference Point	Degrees of Cam Rotation	Reference Point	Degrees of Cam Rotation
A	333	H	600
B	340	I	606
C	379	J	612
D	398	K	681
E	448	L	688
F	467	AA	693
G	483		

Table 6.3 Graph Reference Points and Degrees of Cam Rotation For Figures 6.41 - 6.45

The analysis is similar to that for examples 1A and 1B. For this example the pushrod is pinned to the rocker and thus moves as a part of the system rather than floating free.

At point (A), impact against the pushrod by the cam has just occurred. This has created large system oscillations, first in the pushrod and later in the valve stem. The cam and pushrod are rising. At points (B) and (D) the pushrod is thrown clear of the cam due to its high speed of rotation. At point (E) the cam has reached its highest position and starts to fall. The valve has reached its lowest position and starts its upward travel.

At point (F) the pushrod and rocker come into contact with very little force and travel together for a short distance. Separation again occurs at point (G) when the cam falls away from the pushrod. The valve in its upward travel impacts against the valve seat at point (H). This retards the motion of the pushrod just before contact

at the cam was due to occur (I). The large oscillations created by the valve seating cause the valve to bounce and leave contact at the seat, point (J).

Meanwhile the cam has reached its lowest extreme and begins its upward travel. At point (K) the rising cam impacts against the oscillating pushrod and forces the system to return to its harmonic motion. The cycle then repeats, point (AA).

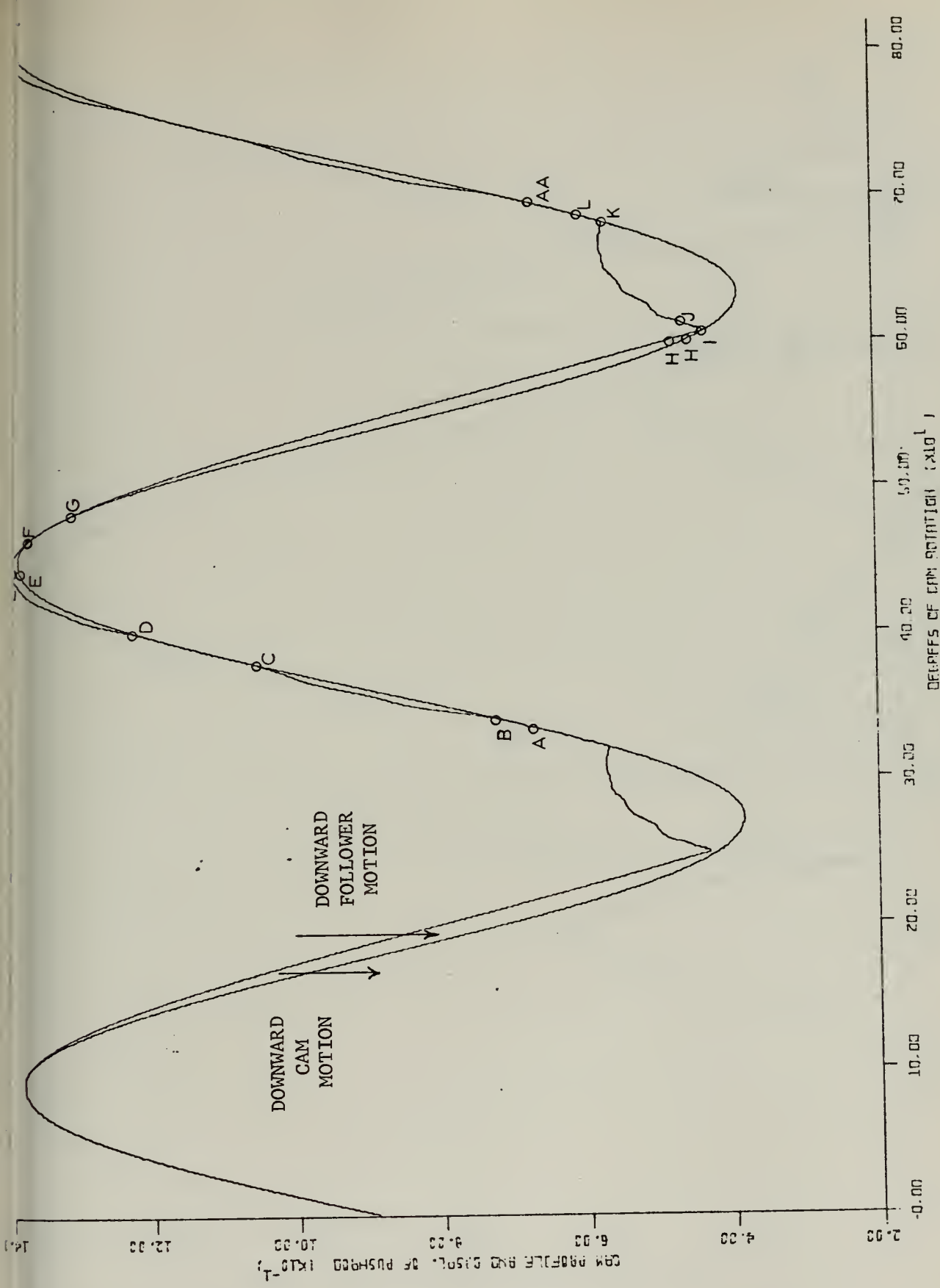


Figure 6.41 TWO CLEARANCES 11,000 RPM VALVE SEAT

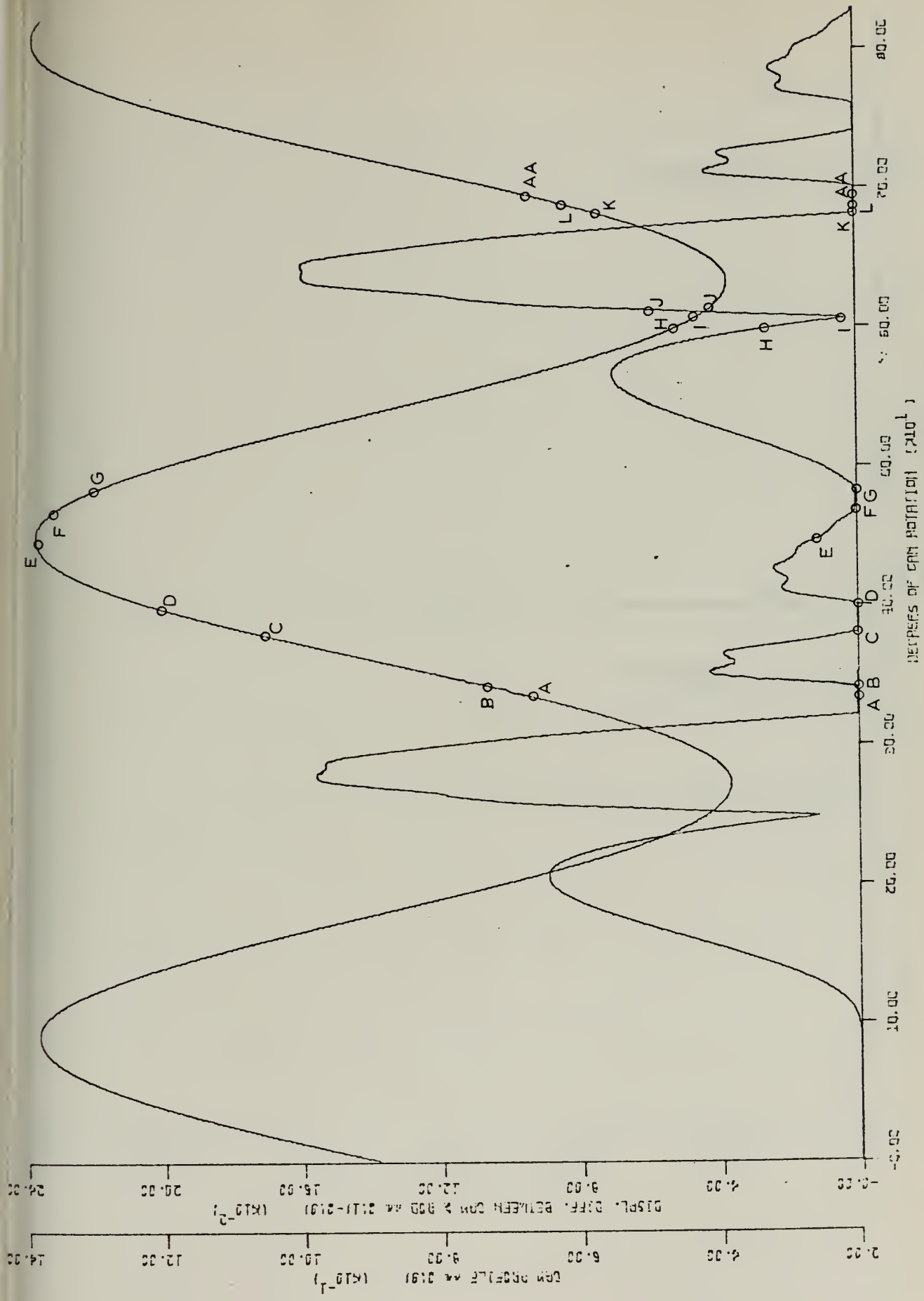


Figure 6.42 TWO CLEARANCES 11,000 RPM VALVE SEAT

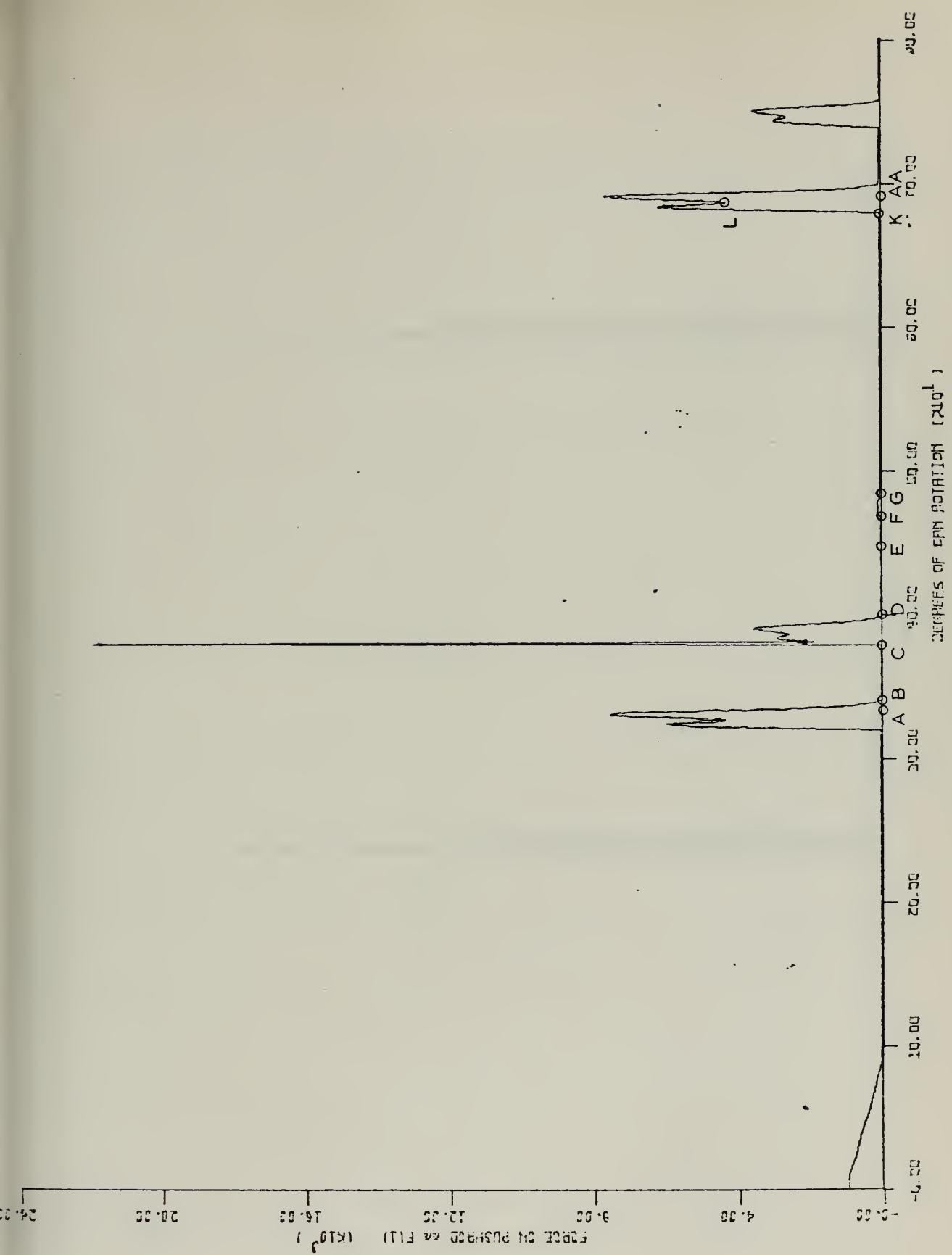


Figure 6.43 TWO CLEARANCES 11,000 RPM VALVE SEAT

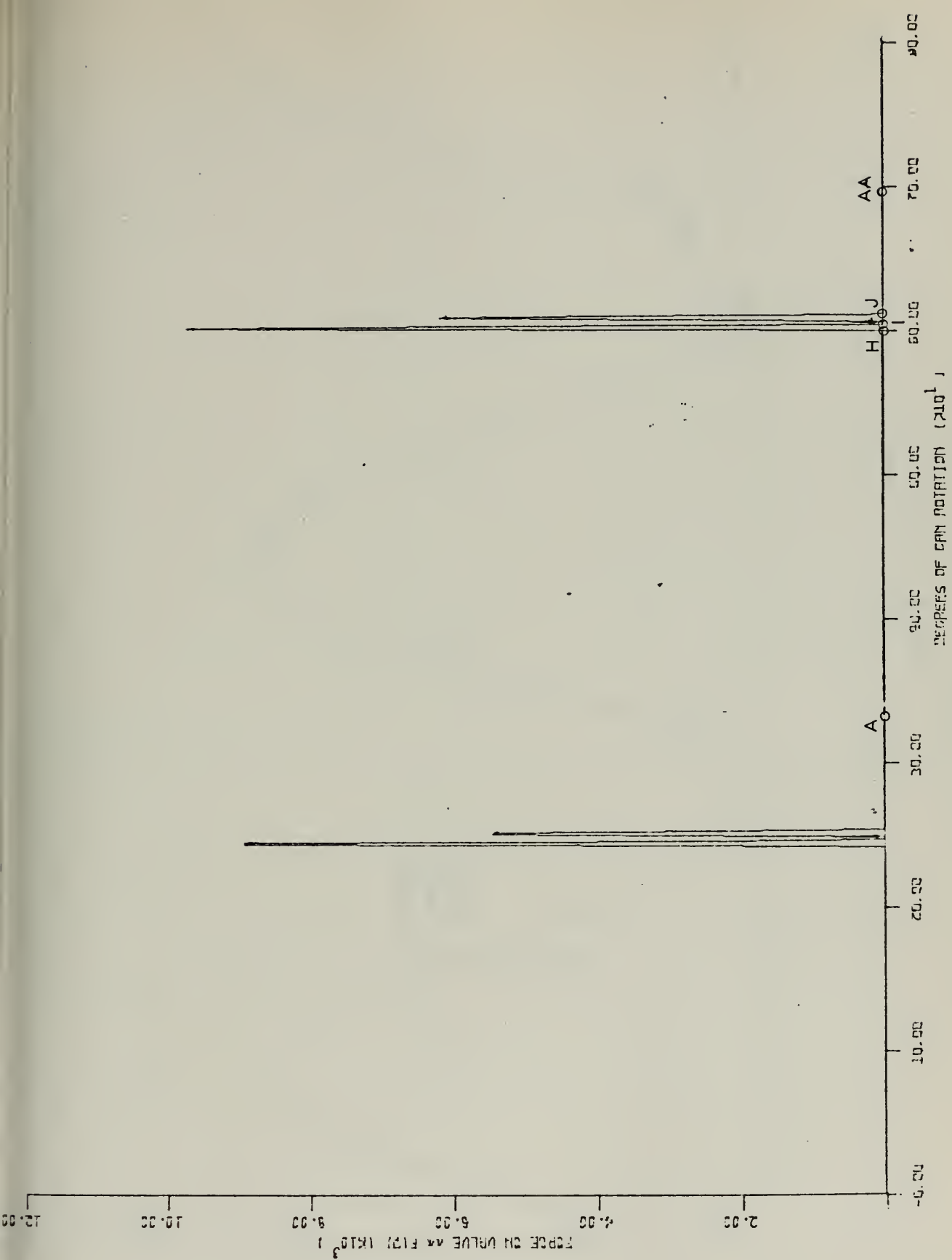


Figure 6.44 TWO CLEARANCES 11,000 RPM VALVE SEAT

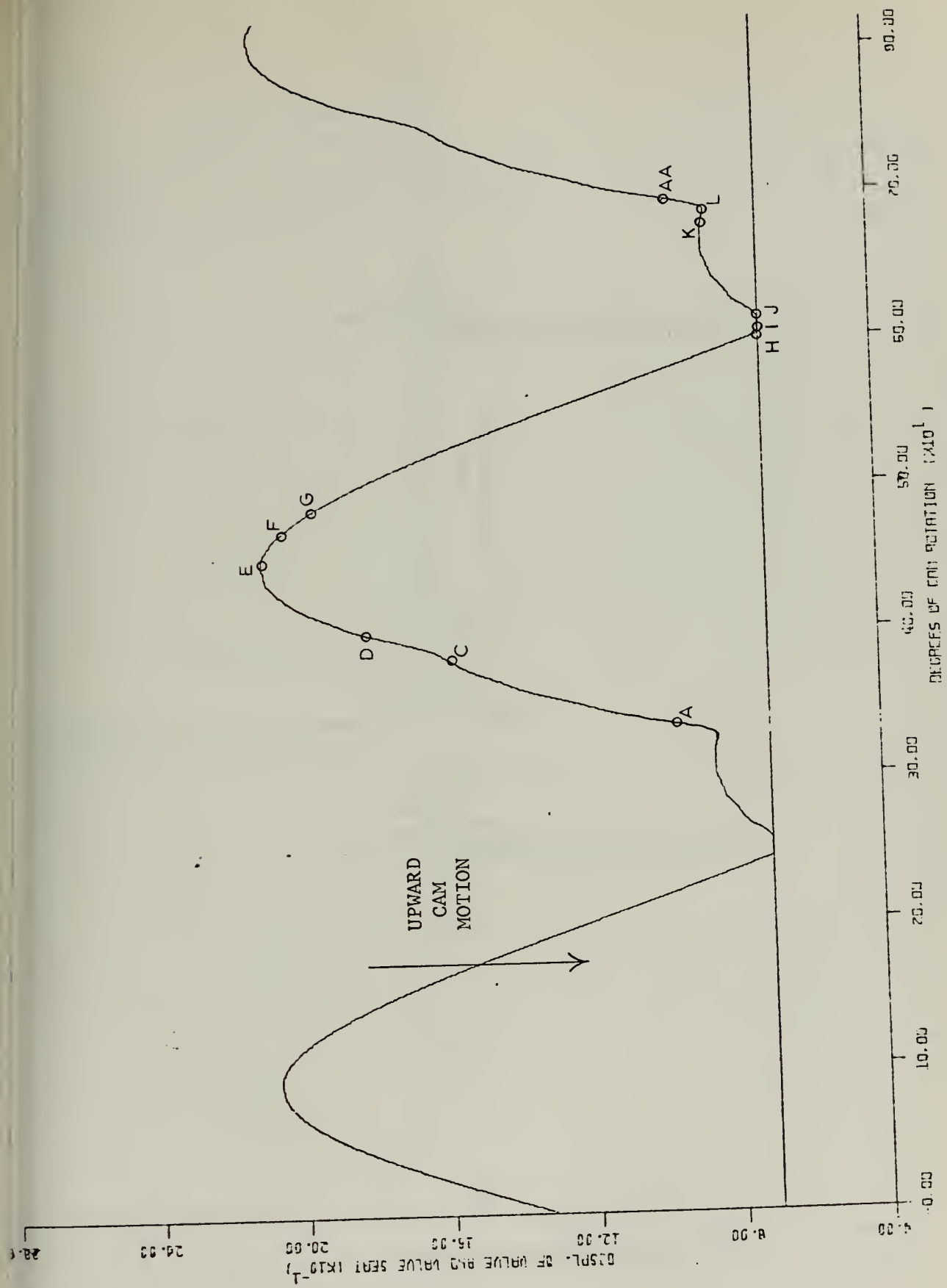


Figure 6.45 TWO CLEARANCES 11,000 RPM VALVE SEAT

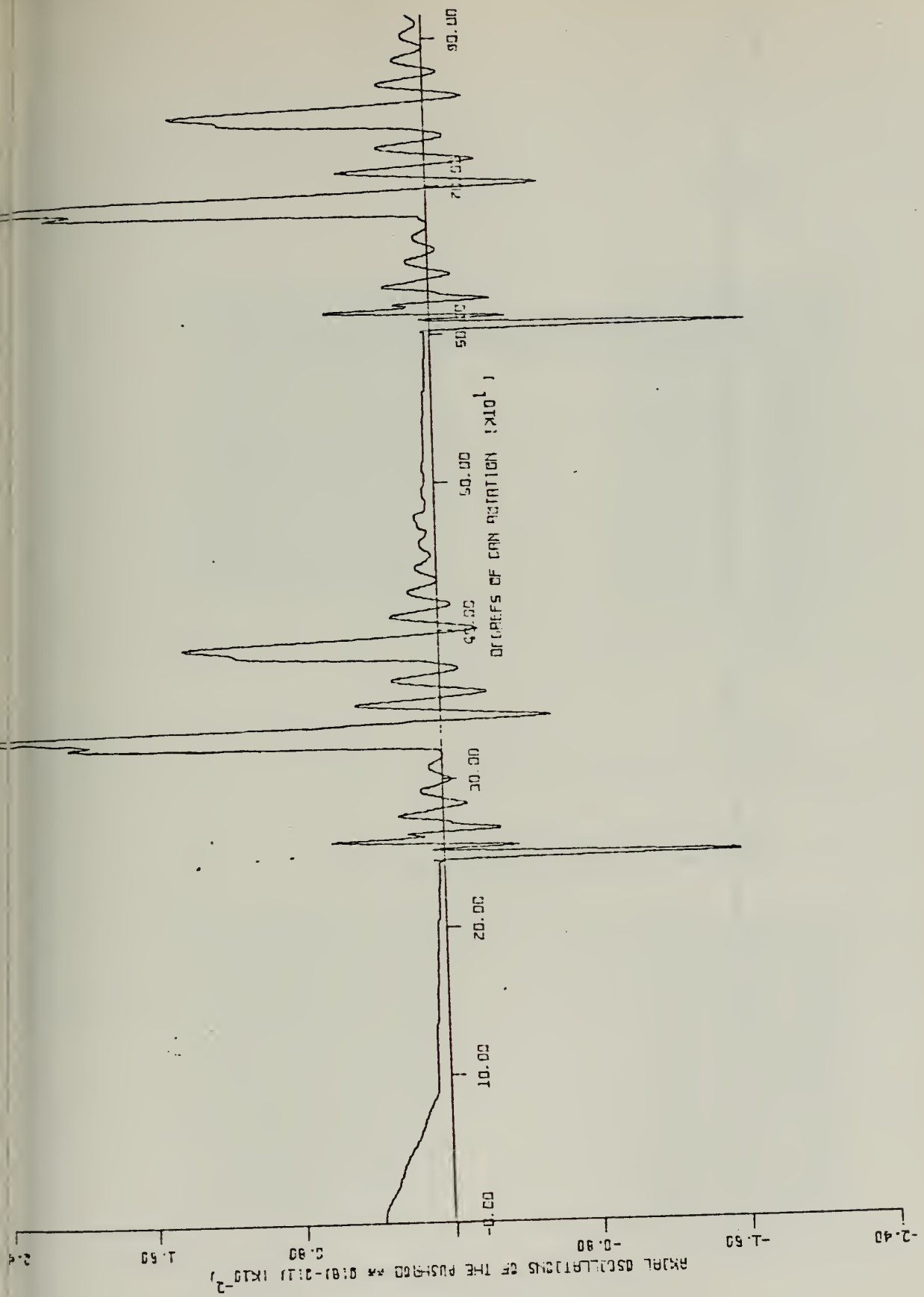


Figure 6.46 TWO CLEARANCES 11,000 RPM VALVE SEAT

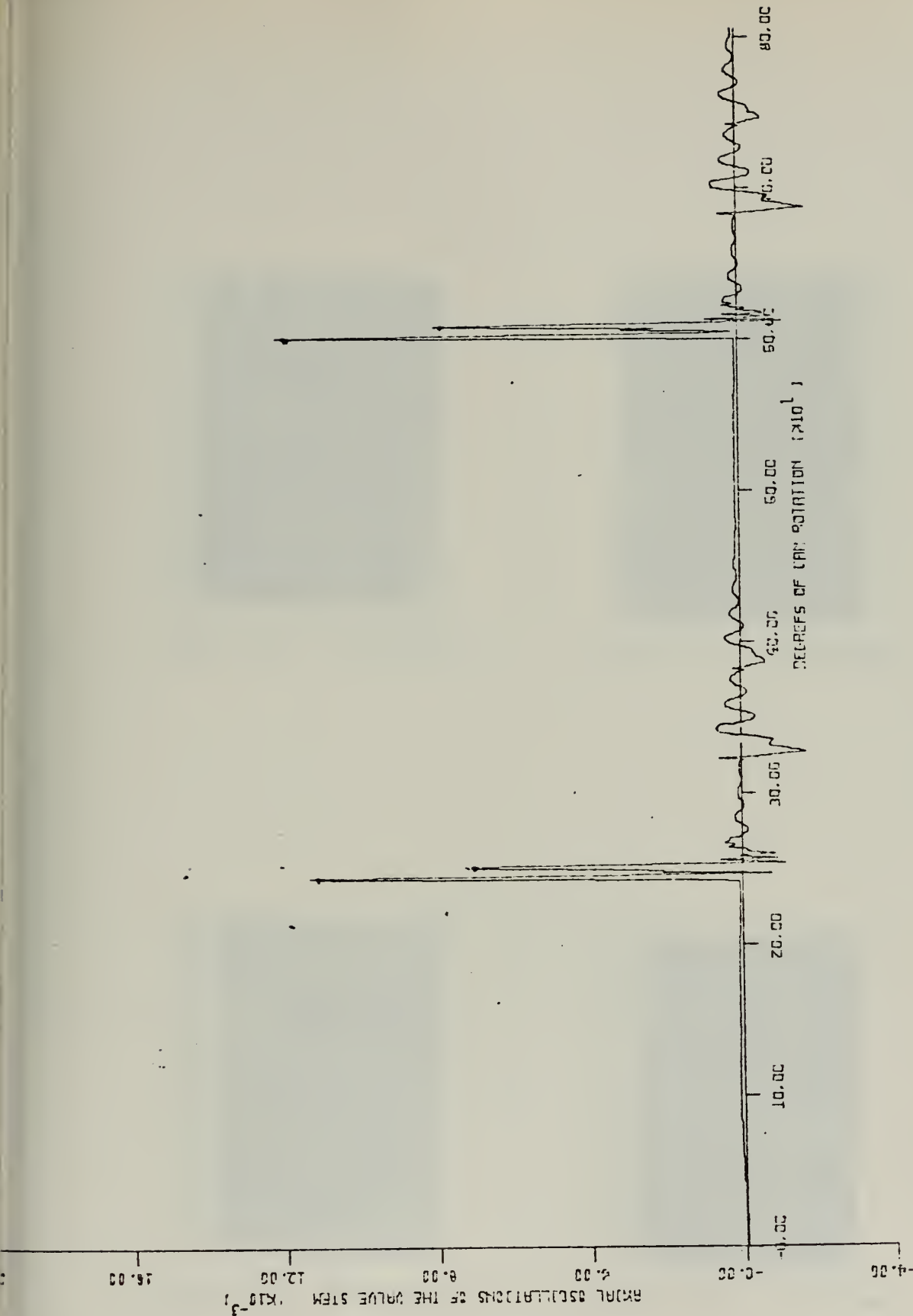


Figure 6.47 TWO CLEARANCES 11,000 RPM VALVE SEAT

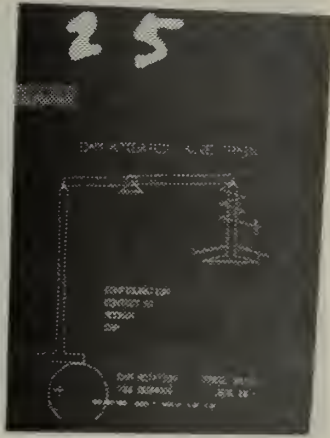


Photo 6.43 - 700 degrees

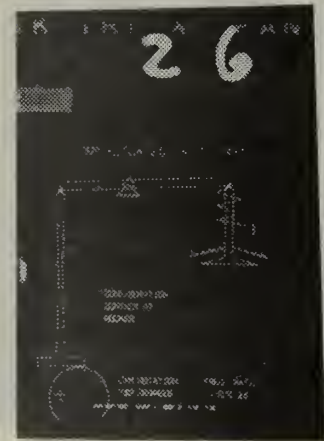


Photo 6.44 - 702 degrees

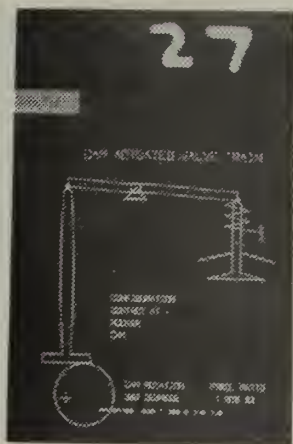


Photo 6.45 - 382 degrees

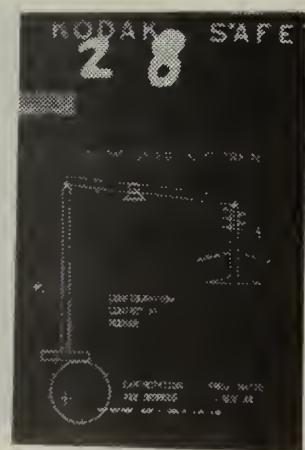


Photo 6.46 - 400 degrees

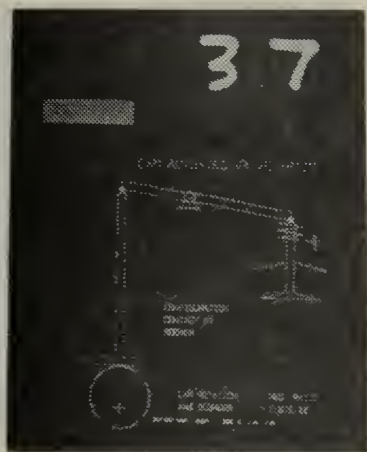


Photo 6.47 - 448 degrees

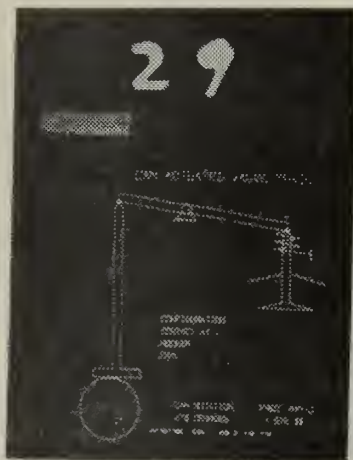


Photo 6.48 - 470 degrees

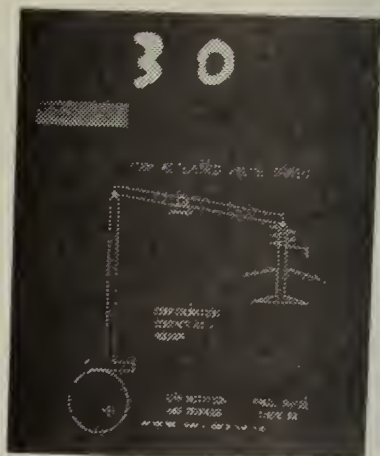


Photo 6.49 - 482 degrees

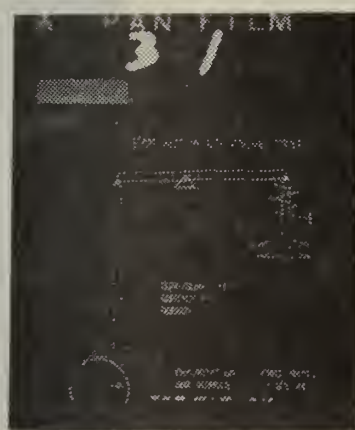


Photo 6.50 - 560 degrees

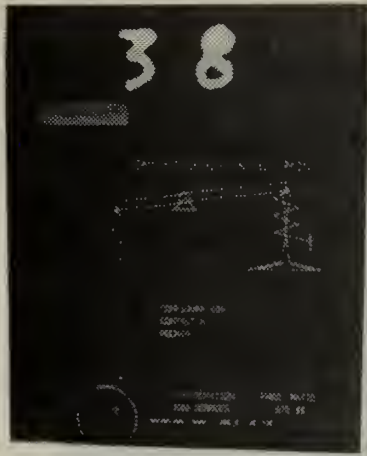


Photo 6.51 - 602 degrees

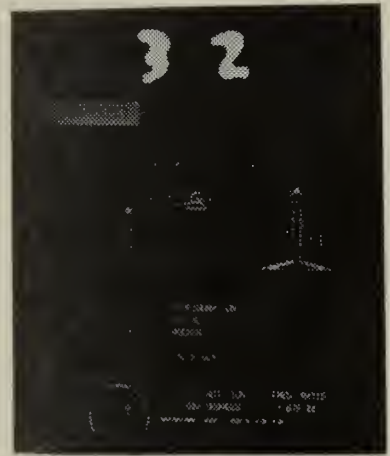


Photo 6.52 - 604 degrees



Photo 6.53 - 606 degrees

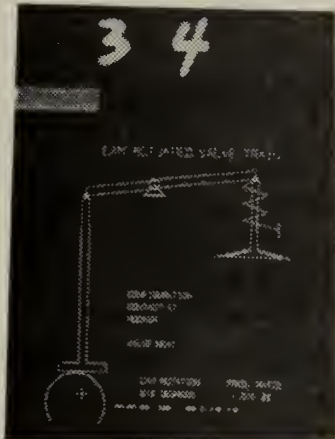


Photo 6.54 - 610 degrees

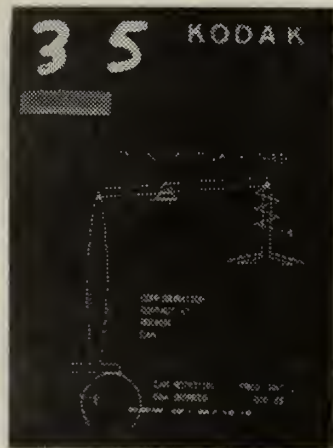


Photo 6.55 - 684 degrees

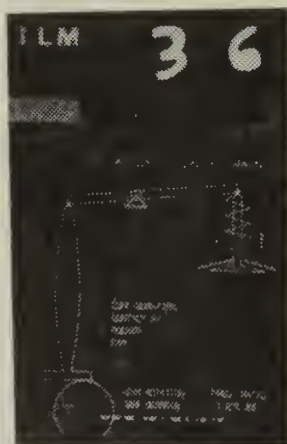


Photo 6.56 - 690 degrees

4. Example 1D, Two Clearances, 11,000 RPM, No Valve Seat

This analysis will commence at 274 degrees of cam rotation, point

(A). The following figures are utilized for the analysis.

a. Figure 6.48

(1) Cam profile, $q_9 \times 10^{-1}$

(2) Displacement of lower end of pushrod,
 $q_1 \times 10^{-1}$

(3) Displacement difference between cam and pushrod,
 $(q_1 - q_9) \times 10^{-2}$

b. Figure 6.49, Force on pushrod, $F_1 \times 10^3$

c. Figure 6.50, Displacement of valve, $q_7 \times 10^{-1}$

d. Figure 6.51, Axial oscillation of the pushrod,
 $(q_9 - qq_1) \times 10^{-2}$

Figure 6.51 has been included for completeness and is not annotated.

The following reference points approximately correspond to the indicated degrees of cam rotation, Table 6.4.

Reference Point	Degrees of Cam Rotation
A	274
B	295
C	314
D	376

Reference Point	Degrees of Cam Rotation
E	398
F	448
AA	634

Table 6.4 Graph Reference Points and Degrees of Cam Rotation For Figures 6.48 - 6.50

The analysis is similar to the above examples. The valve seat position is higher than the maximum excursion of the valve, thus contact does not occur between the seat and valve. The remaining points of interest are then only associated with clearance l , between pushrod and cam.

At point (A) the cam has reached its lowest extreme and begins rising. The pushrod continues its downward harmonic motion, the valve its harmonic motion upward. At point (B), the rising cam impacts against the pushrod. The motion of the pushrod is reversed. The severe oscillations and inertial effects first insure positive contact and then cause the pushrod and cam to separate, point (C). As the upward motion of the pushrod slows, the rising cam again contacts the pushrod, point (D). Since the motion of these two members is similar, the impact force is not as severe.

At point (E), the speed of cam rotation throws the pushrod off. Harmonic motion of the system then results. The cycle repeats at point (AA).

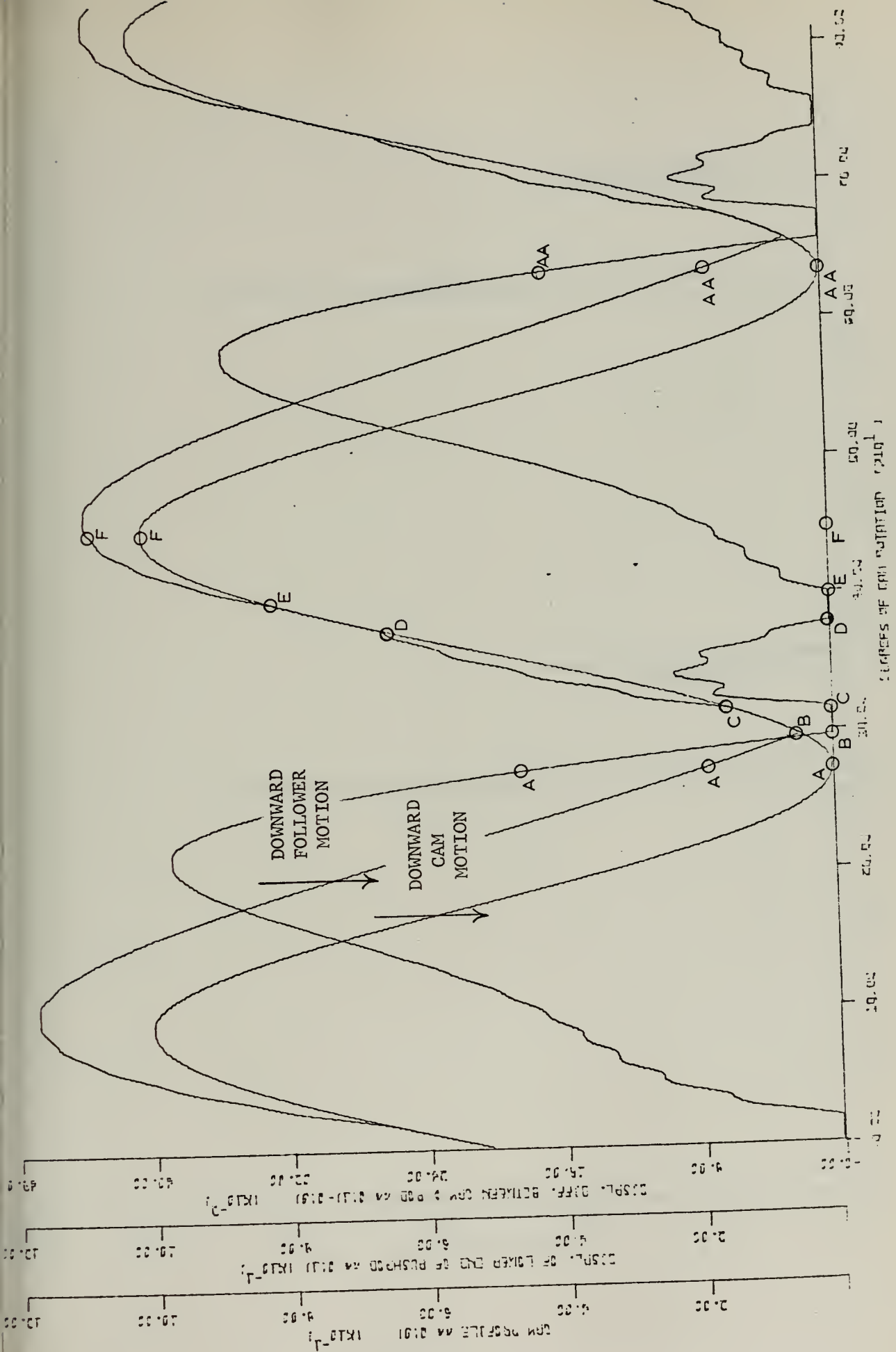
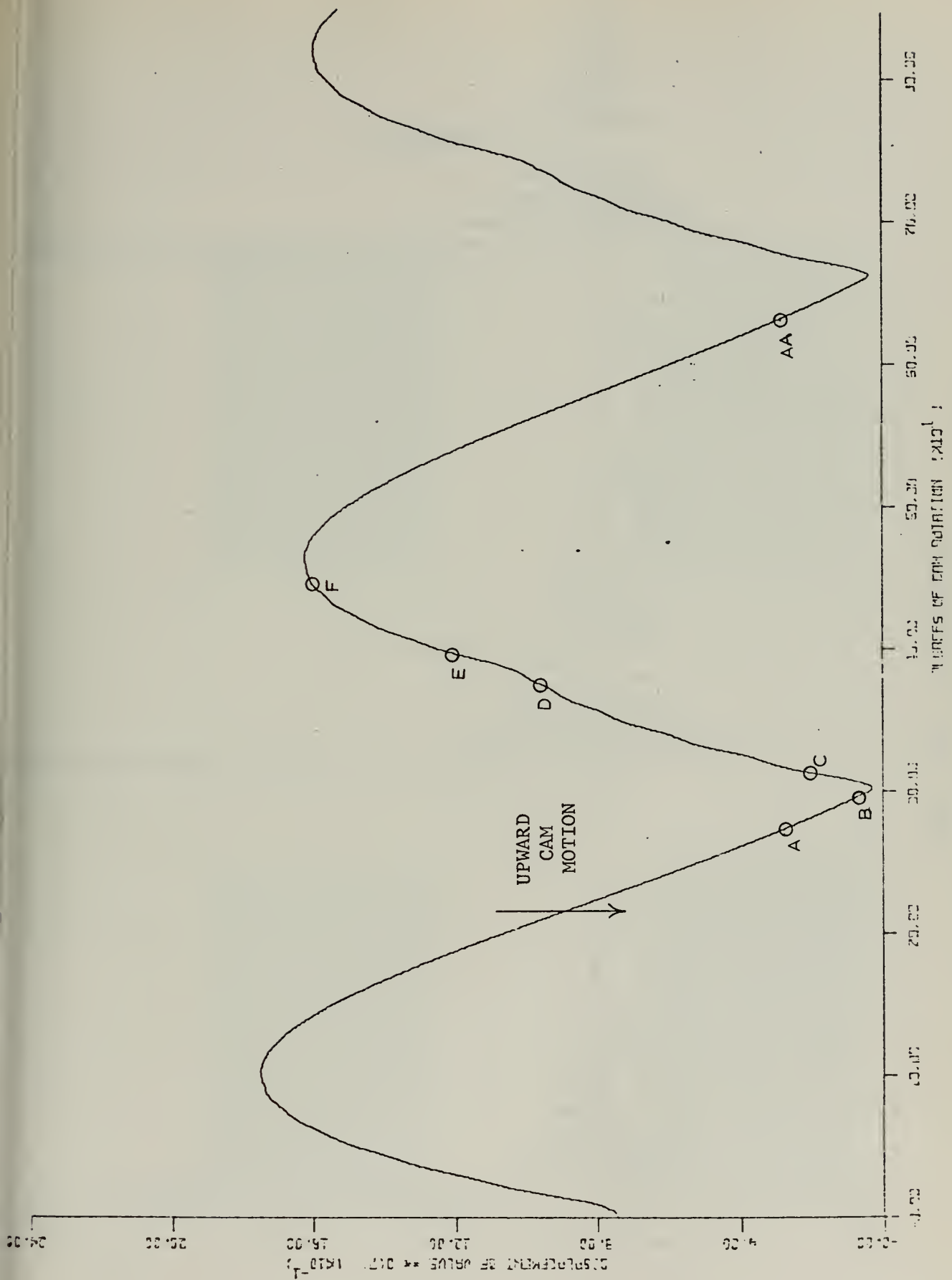


Figure 6.48 TWO CLEARANCES 11,000 RPM NO VALVE SEAT



11,000 RPM NO VALVE SEAT

Figure 6.50 TWO CLEARANCES

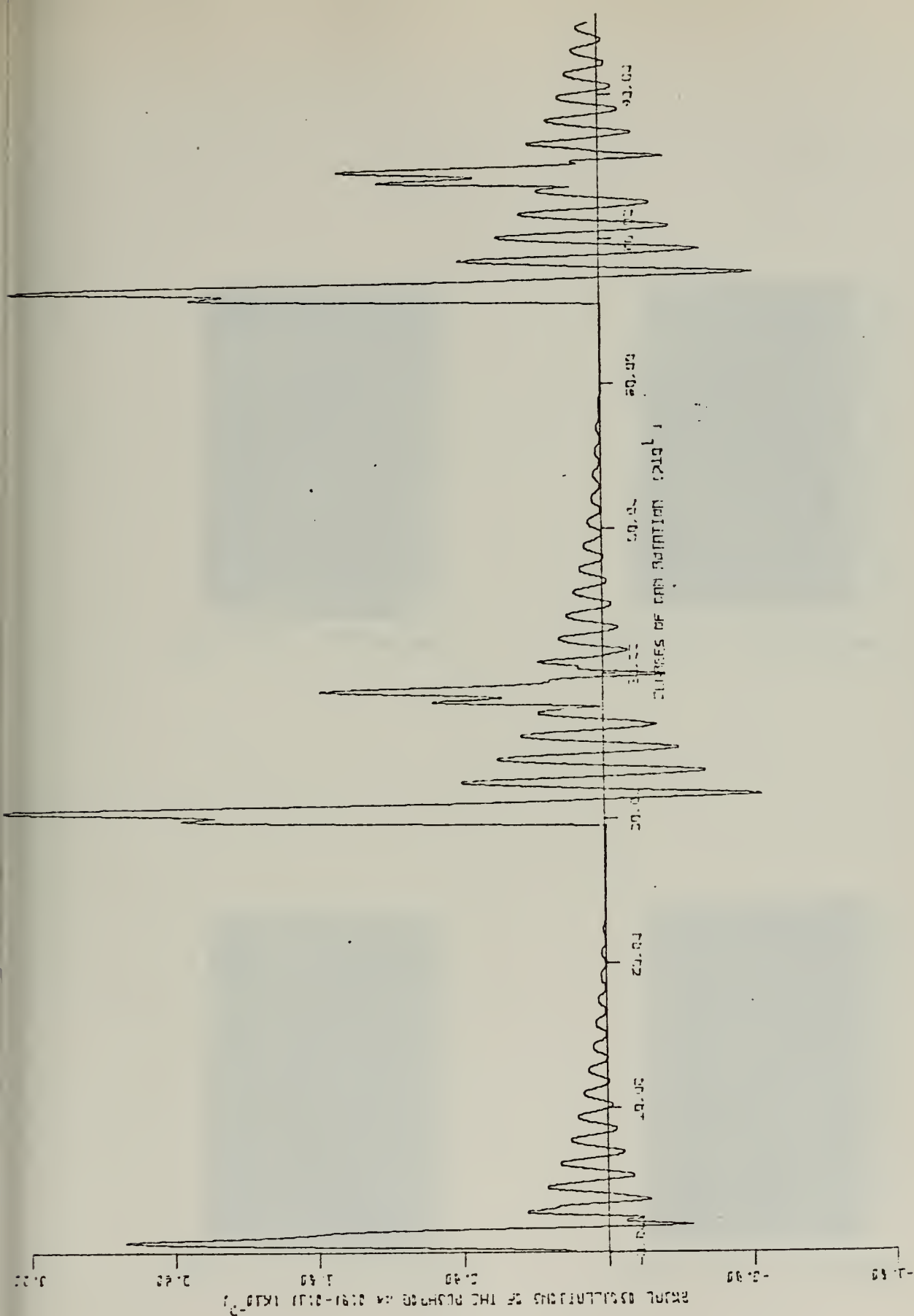


Figure 6.51 TWO CLEARANCES

11,000 RPM NO VALVE SEAT

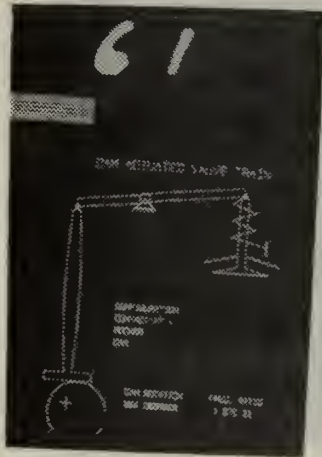


Photo 6.57 - 664 degrees

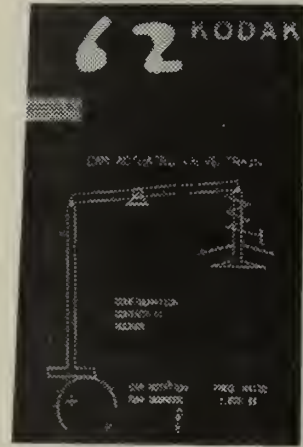


Photo 6.58 - 682 degrees



Photo 6.59 - 694 degrees

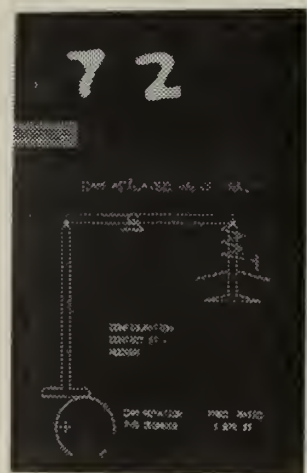


Photo 6.60 - 716 degrees

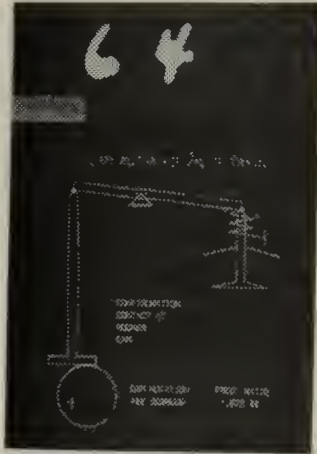


Photo 6.61 - 400 degrees

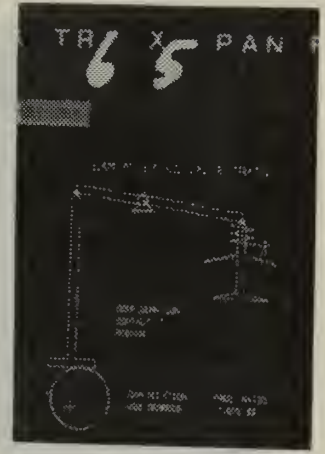


Photo 6.62 - 420 degrees

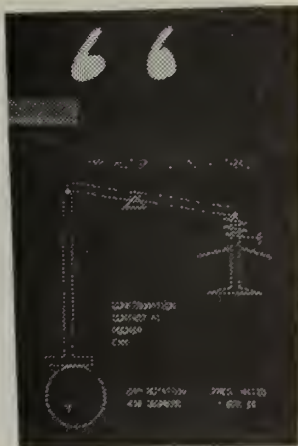


Photo 6.63 - 430 degrees

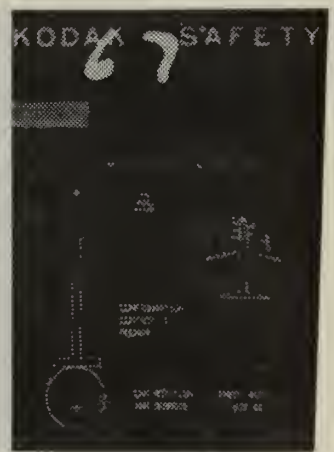


Photo 6.64 - 452 degrees

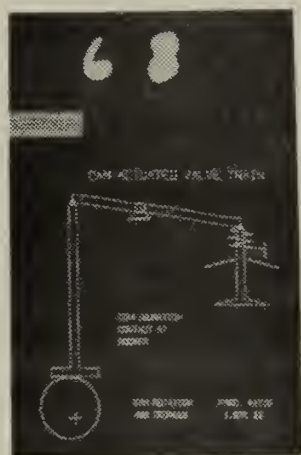


Photo 6.65 - 466 degrees

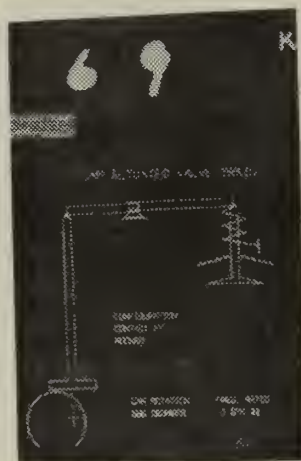


Photo 6.67 - 586 degrees

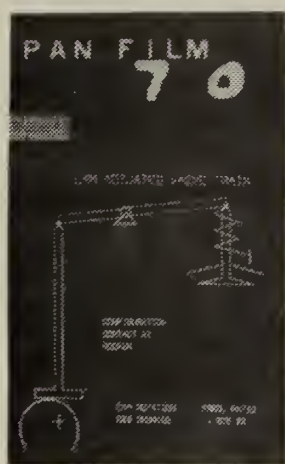


Photo 6.66 - 600 degrees

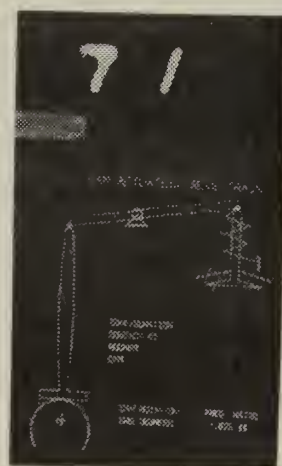


Photo 6.68 - 626 degrees

5. Example two, 9000 RPM and example three, 10,000 RPM were chosen for comparison and completeness of this study. These examples are not annotated and may be found in Appendix F.

VII. CONCLUSION

The simulation for the dynamic response of an elastic link model with three clearances has been described. The algorithm developed is applied to the automobile cam actuated valve train assembly, with external clearances at the valve and cam. The internal clearance at the pushrod-left rocker arm, requires the solution to a second eigenvalue problem in a portion of the calculations. These algorithms are generally coded to allow for application to similar mechanisms. The methods of calculation use address modification arrays to enable application of the program to more complicated models by changing a minimum number of variables and array elements. The numerous subroutines developed have, in general, very specific functions, and are intended to be used as separate calculation packages in a variety of other applications.

While the accuracy of the mathematical model of a mechanism has been greatly increased over the rigid version, the complexity of solution has also increased, even though a number of simplifying assumptions were made. The major assumption made by the theory was that of a linear model so as to permit superposition. Other assumptions, made to simplify the equations but not the theory, involved the choice of coordinates and the use of straight, constant area links.

The purpose of this work was to demonstrate a more accurate model of a mechanism. The extent of this work was to (1) restructure and reprogram an existing simulation to provide for a generalized

method of establishing the physical model configurations; (2) Modify this simulation to allow for more than two clearances; (3) Establish a method for the dynamic graphical display of the model showing the resulting elastic deformations of the model members at high operational speeds. Objectives (1) and (2) are accomplished using the IBM-360 computer for a three clearance simulation. Dynamic dimensioning and program address modifications arrays contribute to the program generalization. Sample problem graphical output is provided. Objective (3) is accomplished with the aid of the ADAGE Graphical Terminal, model 10 interfaced with a Scientific Data System Computer model 9300. Data generated by the IBM computer is transferred to the SDS computer by punched cards. A program then calculates the model's geometry, position, and deformations. The results are then displayed on the AGT. These results are provided on video tape and may be viewed by requesting NPGS Video Tape 12NPS7200042-VR from the Educational Media Department, Naval Postgraduate School.

It is suggested that additional study be done on the calculation of the pushrod bounce at the rocker. The simulation, as written, has difficulty in obtaining a stable system when contact is regained at this point.

An extension of the AGT system would be to allow for the display of force and displacement curves on the AGT. Methods similar to those developed by this author and Lt. Bowden (Ref. 19) could be conveniently applied.

APPENDIX A

MASS AND STIFFNESS MATRICES

The system mass and stiffness matrices are given by equations (3.6) and (3.7), and are repeated below.

$$[M] = \sum_{i=1}^5 [\beta]_i^T [M_e]_i [\beta]_i \quad (3.6)$$

$$[K] = \sum_{i=1}^5 [\beta]_i^T [K_e]_i [\beta]_i \quad (3.7)$$

The transformation matrix $[\beta]$ is given in equation (3.4) and will not be repeated here. The element stiffness and mass matrices are given by reference (1). For the axial members (pushrod, spring, and valve) they are:

$$[K_e]_i = \frac{E_i A_i}{L_i} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad (A.1)$$

$$[M_e]_i = \frac{M_i}{6} \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \quad (i=1,4,5) \quad (A.2)$$

The element stiffness and mass matrices for the bending members (left and right arms) are:

$$[K_e]_i = \frac{E_i I_i}{L_i^3} \begin{bmatrix} 12 & -12 & 6L_i & 6L_i \\ -12 & 12 & -6L_i & -6L_i \\ 6L_i & -6L_i & 4L_i^2 & 2L_i^2 \\ 6L_i & -6L_i & 2L_i^2 & 4L_i^2 \end{bmatrix} \quad (A.3)$$

$$[M_e]_i = \frac{M_i}{420}$$

$$\begin{bmatrix} 156 & 54 & 22L_i & -13L_i \\ 54 & 156 & 13L_i & -22L_i \\ 22L_i & 13L_i & 4L_i^2 & -3L_i^2 \\ -13L_i & -22L_i & -3L_i^2 & 4L_i^2 \end{bmatrix} \quad \begin{matrix} (A.4) \\ (i=2,3) \end{matrix}$$

For equations A.1 through A.4, E_i is the element elastic modulus, A_i is the element cross-sectional area, L_i is the element length, I_i is the element moment of inertia about the bending axis, and M_i is the total element mass of the i^{th} member.

Evaluating equation (3.6) for the five elements of the system yields the full system mass matrix, designated as layer one. The elements of this matrix are given by Figure A.1. Similarly, evaluating equation (3.7) yields the system stiffness matrix. The elements of this matrix are shown in Figure A.2.

$\frac{M_1}{3}$	0	0	0	0	0	0	0	$\frac{M_1}{6}$
0	$\frac{156 M_2}{420}$	$\frac{22M_2L_2}{420}$	0	0	$\frac{-13L_2M_2}{420}$	0	0	0
0	$\frac{22L_2M_2}{420}$	$\frac{4L_2^2M_2}{420}$	0	0	$\frac{-3L_2^2M_2}{420}$	0	0	0
0	0	0	$\frac{156M_3}{420} + \frac{M_4}{3}$	$\frac{-22L_3M_3}{420}$	$\frac{13L_3M_3}{420}$	$\frac{-M_5}{6}$	0	0
			$\frac{+M_5}{3}$					
0	0	0	$\frac{-22L_3M_3}{420}$	$\frac{4L_3^2M_3}{420}$	$\frac{-3L_3^2M_3}{420}$	0	0	0
0	$\frac{-13L_2M_2}{420}$	$\frac{-3L_2^2M_2}{420}$	$\frac{13L_3M_3}{420}$	$\frac{-3L_3^2M_3}{420}$	$\frac{4L_2^2M_2}{420}$	0	0	0
0	0	0	$\frac{-M_5}{6}$	0	0	$\frac{M_5}{3}$	0	0
$\frac{M_1}{6}$	0	0	0	0	0	0	0	$\frac{M_1}{3}$

Figure A.1 Full System Mass Matrix

$\frac{E_1 A_1}{L_1}$	0	0	0	0	0	0	$\frac{-E_1 A_1}{L_1}$
0	$\frac{+12E_2 I_2}{L_2^3}$	$\frac{6E_2 I_2}{L_2^2}$	0	0	$\frac{6E_2 I_2}{L_2^2}$	0	0
0	$\frac{6E_2 I_2}{L_2^2}$	$\frac{4E_2 I_2}{L_2^2}$	0	0	$\frac{2E_2 I_2}{L_2}$	0	0
0	0	0	$\frac{12E_3 I_3}{L_3^3}$	$\frac{-6E_3 I_3}{L_3^2}$	$\frac{-6E_3 I_3}{L_3^2}$	$\frac{E_5 A_5}{L_5}$	0
			$\frac{+E_5 A_5}{L_5}$				
			$\frac{+E_4 A_4}{L_4}$				
0	0	0	$\frac{-6E_3 I_3}{L_3^2}$	$\frac{4E_3 I_3}{L_3}$	$\frac{2E_3 I_3}{L_3}$	0	0
0	$\frac{6E_2 I_2}{L_2^2}$	$\frac{2E_2 I_2}{L_2}$	$\frac{-6E_3 I_3}{L_3^2}$	$\frac{2E_3 I_3}{L_3}$	$\frac{4E_3 I_3}{L_3}$	0	0
					$\frac{+4E_2 I_2}{L_2}$		
0	0	0	$\frac{E_5 A_5}{L_5}$	0	0	$\frac{E_5 A_5}{L_5}$	0
$\frac{-E_1 A_1}{L_1}$	0	0	0	0	0	0	$\frac{E_1 A_1}{L_1}$

Figure A.2 Full System Stiffness Matrix

The mass and stiffness matrices are valid for the eight coordinate system, i.e. configuration I, II, V, and VI. When the rocker is in contact with the pushrod, the seven coordinate system is used and the transformation, mass and stiffness revert to those developed by Anderson, reference (2). The reduced system mass and stiffness matrices are shown in Figures A.3 and A.4, respectfully. These reduced matrices are stored in layer 3 and are used for configurations III, IV, VII, and VIII.

$\frac{M_1}{3}$	$\frac{-M_1}{6}$	0	0	0	0	0
$\frac{-M_1}{6}$	$\frac{M_1}{3} + \frac{156M_2}{420}$	$\frac{22M_2L_2}{420}$	0	0	$\frac{-13L_2M_2}{420}$	0
0	$\frac{22L_2M_2}{420}$	$\frac{4L_2^2M_2}{420}$	0	0	$\frac{-3L_2^2M_2}{420}$	0
0	0	0	$\frac{156M_3}{420} + \frac{M_4}{3}$	$\frac{-22L_3M_3}{420}$	$\frac{13L_3M_3}{420}$	$\frac{-M_5}{6}$
0	0	0	$\frac{+M_5}{3}$	$\frac{-22L_3^2M_3}{420}$	$\frac{4L_3^2M_3}{420}$	$\frac{-3L_3^2M_3}{420}$
0	$\frac{-13L_2M_2}{420}$	$\frac{-3L_2^2M_2}{420}$	$\frac{13L_3M_3}{420}$	$\frac{-3L_3^2M_3}{420}$	$\frac{4L_2^2M_2}{420}$	0
0	0	0	$\frac{-M_5}{6}$	0	0	$\frac{M_5}{3}$

Figure A.3 Reduced System Mass Matrix

$\frac{E_1 A_1}{L_1}$	$\frac{E_1 A_1}{L_1}$	0	0	0	0	0
$\frac{E_1 A_1}{L_1}$	$\frac{E_1 A_1}{L_1}$	$\frac{6E_2 I_2}{L_2^2}$	0	0	$\frac{6E_2 I_2}{L_2^2}$	0
	$\frac{+12E_2 I_2}{L_2^3}$					
0	$\frac{6E_2 I_2}{L_2^2}$	$\frac{4E_2 I_2}{L_2}$	0	0	$\frac{2E_2 I_2}{L_2}$	0
0	0	0	$\frac{12E_3 I_3}{L_3^3}$	$\frac{-6E_3 I_3}{L_3^2}$	$\frac{-6E_3 I_3}{L_3^2}$	$\frac{E_5 A_5}{L_5}$
			$\frac{+E_5 A_5}{L_5}$			
			$\frac{+E_4 A_4}{L_4}$			
0	0	0	$\frac{-6E_3 I_3}{L_3^2}$	$\frac{4E_3 I_3}{L_3}$	$\frac{2E_3 I_3}{L_3}$	0
0	$\frac{6E_2 I_2}{L_2^2}$	$\frac{2E_2 I_2}{L_2}$	$\frac{-6E_3 I_3}{L_3^2}$	$\frac{2E_3 I_3}{L_3}$	$\frac{4E_3 I_3}{L_3}$	0
					$\frac{+4E_2 I_2}{L_2}$	
0	0	0	$\frac{E_5 A_5}{L_5}$	0	0	$\frac{E_5 A_5}{L_5}$

Figure A.4 Reduced System Stiffness Matrix

APPENDIX B

NEWMARK'S β PARAMETER METHOD

The one step algorithm developed by Newmark (Ref. 4), is chosen as the numerical integration technique used in the simulation. The β parameter may be changed for an analysis. This time marching technique solves the governing equations of motion by using the following relations:

$$\{\dot{q}\}^{n+1} = \{\dot{q}\}^n + \frac{\Delta T}{2} \left(\{\ddot{q}\}^n + \{\ddot{q}\}^{n+1} \right) \quad (B.1)$$

and

$$\begin{aligned} \{q\}^{n+1} = \{q\}^n + \Delta T \{\dot{q}\}^n + \frac{(\Delta T)^2}{2} \{\ddot{q}\}^n \\ + \beta (\Delta T)^2 \left(\{\ddot{q}\}^{n+1} - \{\ddot{q}\}^n \right) \end{aligned} \quad (B.2)$$

The net effect of the parameter β is to change the form of the variation of acceleration during the time interval ΔT . For this study, $\beta = 1/6$, assumes that the acceleration changes linearly from $\{\ddot{q}\}^n$ to $\{\ddot{q}\}^{n+1}$, Figure B.1.

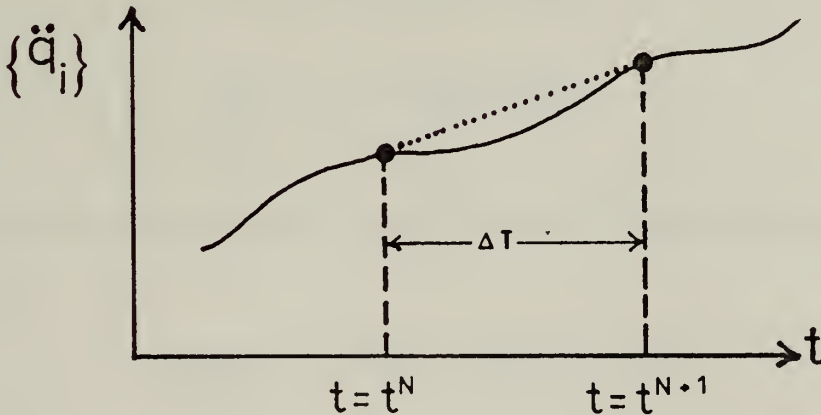


Figure B.1 Time Step Acceleration Curve

In this regard equations B.1 and B.2 are written as:

$$\text{Acceleration} = \ddot{q}_i(t) = \ddot{q}_i^n + \frac{(\ddot{q}_i^{n+1} - \ddot{q}_i^n)}{\Delta T} t \quad (\text{B.3})$$

$$\text{Velocity} = \dot{q}_i(t) = \dot{q}_i^n + \ddot{q}_i^n t + \frac{(\dot{q}_i^{n+1} - \dot{q}_i^n)}{\Delta T} \frac{t^2}{2} \quad (\text{B.4})$$

$$\text{Displacement} = q_i(t) = q_i^n + \dot{q}_i^n t + \ddot{q}_i^n \frac{t^2}{2} + \frac{(\ddot{q}_i^{n+1} - \ddot{q}_i^n)}{\Delta T} \frac{t^3}{6} \quad (\text{B.5})$$

Evaluate $\dot{q}_i(t)$ and $q_i(t)$ at time $t = \Delta T$,

$$\dot{q}_i^{n+1} = \dot{q}_i^n + \ddot{q}_i^n \frac{\Delta T}{2} + \ddot{q}_i^{n+1} \frac{\Delta T}{2} \quad (\text{B.6})$$

$$q_i^{n+1} = q_i^n + \dot{q}_i^n \Delta T + \ddot{q}_i^n \frac{(\Delta T)^2}{3} + \ddot{q}_i^{n+1} \frac{(\Delta T)^2}{6} \quad (\text{B.7})$$

$$\text{Let } \alpha_i^n = \dot{q}_i^n + \ddot{q}_i^n \frac{\Delta T}{2} \quad (\text{B.8})$$

$$\text{and } \psi_i^n = q_i^n + \dot{q}_i^n \Delta T + \ddot{q}_i^n \frac{(\Delta T)^2}{3} \quad (\text{B.9})$$

Now \dot{q}_i^{n+1} and q_i^{n+1} can be rewritten:

$$\dot{q}_i^{n+1} = \alpha_i^n + \ddot{q}_i^{n+1} \frac{\Delta T}{2} \quad (\text{B.10})$$

$$q_i^{n+1} = \psi_i^n + \ddot{q}_i^{n+1} \frac{(\Delta T)^2}{6} \quad (\text{B.11})$$

Substitute the above results for \dot{q}_i^{n+1} and q_i^{n+1} into the equations of motion

$$[M] \{\ddot{q}\} + C \{\dot{q}\} + [K] \{q\} = \{F\} \quad (\text{B.12})$$

$$\begin{aligned}
 [M] \{\ddot{q}\}^{n+1} + [C] \{\alpha\}^n + \frac{\Delta T}{2} [C] \{\ddot{q}\}^{n+1} + [K] \{\psi\}^n + \frac{(\Delta T)^2}{6} [K] \{\ddot{q}\}^{n+1} \\
 = \{P\}^{n+1}
 \end{aligned}
 \tag{B.13}$$

$$\text{Rearranged, } \left[[M] + \frac{\Delta T}{2} [C] + \frac{\Delta T^2}{6} [K] \right] \{\ddot{q}\}^{n+1} = \{P\}^{n+1} - [C] \{\alpha\}^n - [K] \{\psi\}^n$$

(B.14)

$$\text{Rewritten, } \overline{[M]}^{n+1} \{\ddot{q}\}^{n+1} = \overline{\{P\}}^{n+1}$$

(B.15)

where $\{P\}^{n+1} = \{F\}$ evaluated at time $n+1$

$$\overline{[M]}^{n+1} = [M] + \frac{\Delta T}{2} [C] + \frac{\Delta T^2}{6} [K]$$

(B.16)

$$\overline{\{P\}}^{n+1} = \{P\}^{n+1} - [C] \{\alpha\}^n - [K] \{\psi\}^n$$

(B.17)

This results in the accelerations $\{\ddot{q}\}^{n+1}$ expressed in terms of the known system values at the prior time step, $t = t^n$.

$$\{\ddot{q}\}^{n+1} = \left[\overline{[M]}^{n+1} \right]^{-1} \overline{\{P\}}^{n+1}$$

APPENDIX C

FOUR CLEARANCE PROBLEM

During the early stages of this study, limited research was conducted on the four clearance problem to better understand the necessary requirements of program generalization procedures. This appendix will present the unique four clearance number scheme that must be used in conjunction with the shifting algorithm presented in section 3.D. Figure C.1 indicates the configuration number associated with the various combinations of clearances, for a given time, that the model is physically permitted to take. Figure C.2 shows the resulting flow diagram. The shift lines relate the flow for a nonpositive force at the clearance indicated. Non-positive displacement differences are along the same shift lines, but in the direction opposite the force arrow.

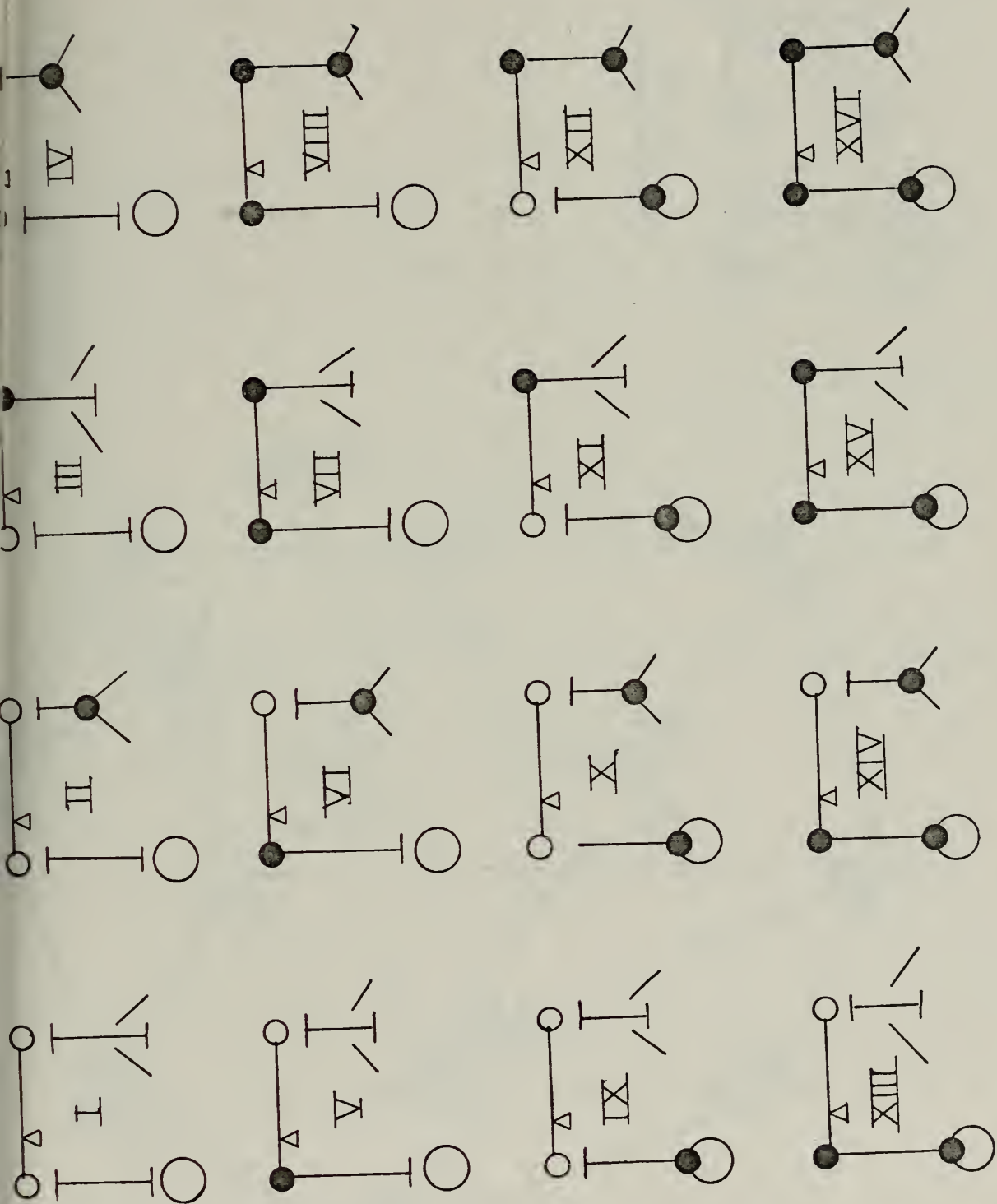
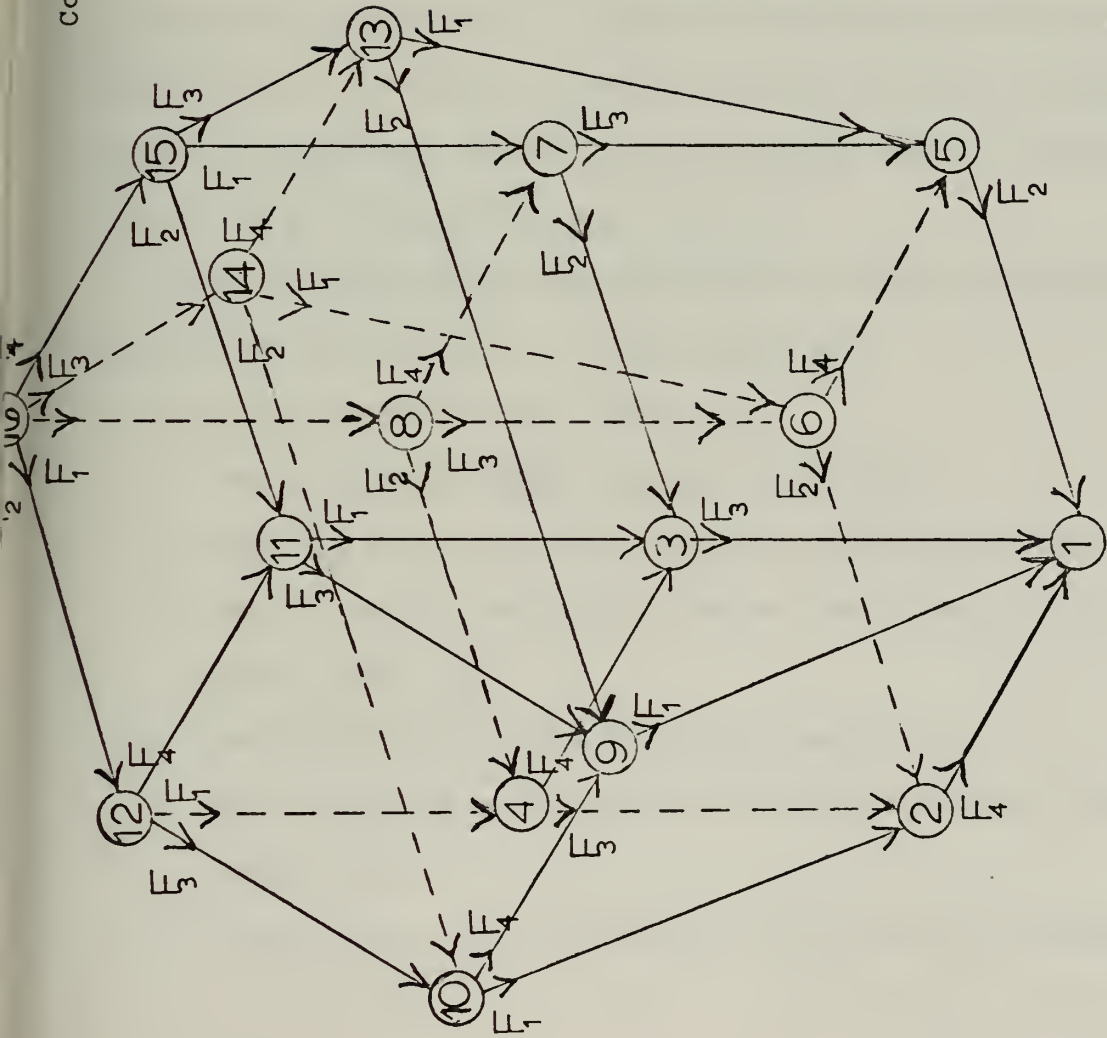


Figure C.1 FOUR CLEARANCE SYSTEM CONFIGURATIONS

Configuration Numbers Indicated



Clearance Numbers Indicated

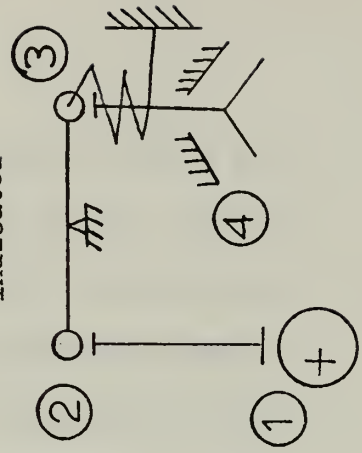


Figure C.2 4 Clearance Configuration Flow Diagram

APPENDIX D

COMPUTER CONTROL CARDS

A. IBM-360 COMPUTER CONTROL CARDS

As previously noted subroutines were compiled and placed on a 2314 Disk Storage Device. This accomplished two goals. First, it eliminated the requirement to physically read into the computer a large number of punched cards. Second, it significantly reduced program execution time by greatly reducing the FORT step. This allowed the program to execute in a higher job class.

The following figures list the control cards required to accomplish various disk manipulations. A complete explanation will not be attempted here, but merely a listing of procedures that have been found to be effective. Section 3.4 of the Users Manual, reference (13), should be reviewed for more information.

In the following example control cards, the data set name is SO274. NILKA and is placed on public disk Mary.

1. Initialize disk space, Figure D.1.
2. Load subroutine NEWMRK onto disk, Figure D.2.
3. List disk data set members and space remaining, Figure D.3.
4. Execute program reading only control cards, main, and data cards, Figure D.4.
5. Load main program, CWG, onto disk, Figure D.5.
6. Execute program reading only control cards and data cards, Figure D.6.
7. Compress disk area when available space has been exceeded.

- a. Read disk members onto scratch disk without intermediate spacing, Figure D.7.
- b. Copy all members to users disk, Figure D.8.
- c. Copy all members except NEWMRK to users disk, Figure D.9.

```
// JOB CARD (GREEN CARD)
// EXEC FORTHCL,PARM.LINK='NCAL,LET'
//FORT.SYSIN DD *
--- SUBROUTINE NEWMRK CARDS ---
STEP DELIMITER /* CARD
//LINK.SYSLMOD DD UNIT=2314,VOL=SER=MARY,DSN=S0274.NILKA,
// DISP=(NEW,KEEP),SPACE=(CYL,(6,1,6)),LABEL=EXPDT=71365
//LINK.SYSIN DD *
NAME NEWMRK(R)
END OF JOB /* ( ORANGE CARD )
```

Figure D.1 Initialize Disk Space

```
// JOB CARD (GREEN CARD)
// EXEC FORTHCL,PARM.LINK='NCAL,LET'
//FORT.SYSIN DD *
--- SUBROUTINE NEWMRK CARDS ---
STEP DELIMITER /* CARD
//LINK.SYSLMOD DD UNIT=2314,VOL=SER=MARY,DSN=S0274.NILKA,
// DISP=OLD,SPACE=(CYL,(6,1,6))
//LINK.SYSIN DD *
NAME NEWMRK(R)
END OF JOB /* ( ORANGE CARD )
```

Figure D.2 Load Subroutine onto Disk

```
// JOB CARD (GREEN CARD)
//LIST EXEC PGM=IEHLIST,REGION=58K
//SYSPRINT DD SYSOUT=A,SPACE=(TRK,(1,1))
//DD1 DD UNIT=2314,VOL=SER=MARY,DISP=OLD
//DD2 DD UNIT=2314,VOL=SER=MARY,DISP=OLD
//SYSIN DD *,DCB=BLKSIZE=80
LISTVTOC FORMAT,VOL=2314=MARY,DSNAME=(S0274.NILKA)
LISTPDS DSNAME=S0274.NILKA,VOL=2314=MARY
END OF JOB /* ( ORANGE CARD )
```

Figure D.3 List Data Set VTOC and PDS


```

// JOB CARD (GREEN CARD)
// EXEC FORTHCLG,REGION.GO=250K
//FORT.SYSIN DD *
----- MAIN PROGRAM CARDS -----
      STEP DELIMINATOR /* CARD
//LINK.PLIB DD DSN=S0274.NILKA,VOL=SER=MARY,DISP=SHR,UNIT=2314
//LINK.SYSIN DD *
INCLUDE PLIB(BRANCH,CAM,DEBUG,DEL,DRUP,DVVS,MCK)
INCLUDE PLIB(INITAL,MATRIX,NEWMRK,OLPLOT,SHIFT)
INCLUDE PLIB(BKSTEP)
INCLUDE PLIB(INPUT)
INCLUDE PLIB(GRAPH)
INCLUDE PLIB(AGT)
INCLUDE PLIB(THESIS)
INCLUDE PLIB(PRINT)
INCLUDE PLIB(ARRAY)
      STEP DELIMINATOR /* CARD
//GO.FT06F001 DD SYSOUT=A,SPACE=(CYL,(4,2))
//GO.SYSIN DD *
--- DATA CARDS ---
END OF JOB /*                ( ORANGE CARD )

```

Figure D.4 Execute Program with Subroutines on Disk

```

// JOB CARD (GREEN CARD)
// EXEC FORTHCL,PARM.FORT='NAME=CWG,LOAD,MAP',
// PARM.LINK='LET,NCAL,MAP,XREF'
//FORT.SYSIN DD *
----- MAIN PROGRAM CARDS -----
      STEP DELIMINATOR /* CARD
//LINK.SYSLMOD DD UNIT=2314,VOL=SER=MARY,DSN=S0274.NILKA,
// DISP=OLD,SPACE=(CYL,(6,1,6))
//LINK.SYSIN DD *
NAME CWG(R)
END OF JOB /*                ( ORANGE CARD )

```

Figure D.5 Load Main Program onto Disk

```

// JOB CARD (GREEN CARD)
// EXEC LGOP,REGION.GO=250K
//LINK.PLIB DD UNIT=2314,VOL=SER=MARY,DSN=S0274,NILKA,DISP=SHR
//LINK.SYSIN DD *
INCLUDE PLIB(ARRAY,BKSTEP,BRANCH,CAM,DEBUG,DEL,DRUP,DVVS,GRAPH,MCK)
INCLUDE PLIB(INITAL,INPUT,MATRIX,NEWMRK,OLPLOT,PRINT,SHIFT,THESIS)
INCLUDE PLIB(CWG)
ENTRY CWG
      STEP DELIMINATOR /* CARD
//GO.SYSIN DD *
--- DATA CARDS ---
END OF JOB /*                ( ORANGE CARD )

```

Figure D.6 Execute Entire Program from Disk


```

// JOB CARD (GREEN CARD)
//STEP1 EXEC PGM=IEHLIST
//SYSPRINT DD SYSOUT=A
//D5 DD UNIT=2314,VOL=(PRIVATE,,SER=MARY),DISP=SHR
//SYSIN DD *
LISTVTOC VOL=2314=MARY
//STEP2 EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//SPOOL3 DD DISP=OLD,UNIT=2314,VOLUME=SER=SPOOL3
//TMPLIB DD DISP=(NEW,KEEP),UNIT=2314,DSNAME=NILKA.TMPLIB,
// SPACE=(CYL,(6,0,17)),DCB=(RECFM=U,BLKSIZE=7294),
// VOLUME=SER=SPOOL3
//SYSIN DD *
SCRATCH DSNAME=S0274.NILKA,VOL=2314=SPOOL3,PURGE
RENAME DSNAME=NILKA.TMPLIB,VOL=2314=SPOOL3,NEWNAME=S0274.NILKA
STEP DELIMINATOR /* CARD
//STEP3 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DISP=OLD,UNIT=2314,DSNAME=S0274.NILKA,VOLUME=SER=MARY
//SYSUT2 DD DISP=OLD,UNIT=2314,DSNAME=S0274.NILKA,VOLUME=SER=SPOOL3
//SYSIN DD *
COPY
END OF JOB /* ( ORANGE CARD )

```

Figure D.7 Read Data Set Members onto Scratch Disk

```

// JOB CARD (GREEN CARD)
//STEP3 EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//MARY DD DISP=OLD,UNIT=2314,VOLUME=SER=MARY
//SYSIN DD *
SCRATCH DSNAME=S0274.NILKA,VOL=2314=MARY,PURGE
STEP DELIMINATOR /* CARD
//STEP4 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DISP=OLD,UNIT=2314,DSNAME=S0274.NILKA,VOLUME=SER=SPOOL3
//SYSUT2 DD DISP=(NEW,KEEP),UNIT=2314,DSNAME=S0274.NILKA,
// SPACE=(CYL,(6,1,6)),DCB=(RECFM=U,BLKSIZE=7294),
// VOLUME=SER=MARY,LABEL=EXPDT=71365
//SYSIN DD *
COPY
END OF JOB /* ( ORANGE CARD )

```

Figure D.8 Copy all Data Set Members

```

// JOB CARD (GREEN CARD)
//STEP3 EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//MARY DD DISP=OLD,UNIT=2314,VOLUME=SER=MARY
//SYSIN DD *
SCRATCH DSNAME=S0274.NILKA,VOL=2314=MARY,PURGE
STEP DELIMINATOR /* CARD
//STEP4 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DISP=OLD,UNIT=2314,DSNAME=S0274.NILKA,VOLUME=SER=SPOOL3
//SYSUT2 DD DISP=(NEW,KEEP),UNIT=2314,DSNAME=S0274.NILKA,
// SPACE=(CYL,(6,1,6)),DCB=(RECFM=U,BLKSIZE=7294),
// VOLUME=SER=MARY,LABEL=EXPDT=71365
//SYSIN DD *
COPY
MEMBER NAME=(NEWMRK)
END OF JOB /* ( ORANGE CARD )

```

Figure D.9 Copy Part of Data Set Members

B. SDS-9300 COMPUTER CONTROL CARDS

Transferring the SDS-9300 program, in Fortran source code, from punched cards to magnetic tape is a relatively easy task. Once accomplished, repeated compiling and execution of the program is made much easier. Figure D.10 indicates the control cards required to read the punched cards onto magnetic tape.

```
SDS-9300 COMPUTER BINARY BOOT CARD*  
-JOB  
-LOAD X, TAPEDT  
$  
+  
--- MAIN PROGRAM CARDS ---  
=EOF  
--- SUBROUTINE BAR1 CARDS ---  
--- SUBROUTINE BAR2 CARDS ---  
--- SUBROUTINE BAR3 CARDS ---  
--- SUBROUTINE BAR4 CARDS ---  
=EOF  
--- SUBROUTINE CAM CARDS ---  
=EOF  
ETC. UNTIL ALL SUBROUTINES ARE READ.  
=EOF
```

*Indicates card supplied by Computer Laboratory

Figure D.10 Transferring FORTRAN Source Code From Cards to Magnetic Tape

The computer laboratory provided routine TAPEDT enables the user to change the program coding with a minimum of cards, while the program remains on the tape.

Figure D.11 shows the required control cards to compile, load, and execute the magnetic tape Fortran source program.


```

SDS-9300 COMPUTER BINARY BOOT CARD*
-ΔJOB
$
-ΔASSIGN SI=MT2A
-ΔREWIND SI
-ΔDATE 17,DEC,1971
-ΔTITLE . CHARLES W. GNILKA - - THESIS - -
-ΔAGT
-ΔASSIGN 4=MT4A
-ΔLABEL MAIN
-ΔFORTRAN GO
-ΔLABEL CAM
-ΔFORTRAN GO
-ΔLABEL MIDBAR
-ΔFORTRAN GO
-ΔLABEL SPRCTR
-ΔFORTRAN GO
-ΔLABEL BAR
-ΔFORTRAN GO
-ΔLABEL DISPLY
-ΔFORTRAN GO
-ΔLABEL INITAL
-ΔFORTRAN GO
-ΔLABEL READ
-ΔFORTRAN GO
-ΔLOAD XR,MAP
-ΔSEG MAIN-BAR-CAM-(MIDBAR,SPRCTR,DISPLY,INITAL,READ)
-ΔDATA

```

*Indicates Card Supplied by Computer Laboratory

Figure D.11 Program Execution From FORTRAN Object on Magnetic Tape

To place the program on magnetic tape as binary executable code, the control cards are as in Figure D.11 with an additional control card immediately after the Δ JOB card. Figure D.12 indicates the required control cards.


```
SDS-9300 COMPUTER BINARY BOOT CARD*  
-JOB  
$  
-ASSIGN GO=MT3A  
( THE REMAINING CONTROL CARDS ARE EXACTLY AS GIVEN ABOVE )
```

Figure D.12 Placing Program as Binary Code on Magnetic Tape

Once the binary code is on magnetic it may be executed approximately 75% faster than in uncompiled FORTRAN source form. The control cards to accomplish this are shown in Figure D.13

```
SDS-9300 COMPUTER BINARY BOOT CARD*  
-JOB  
$  
-AGT  
-ASSIGN BI=MT3A  
-ASSIGN GO=BI  
-ASSIGN SI=MT3A  
-LCAD XR,MAP  
-SEG MAIN-BAR-CAM-(MIDBAR,SPRCTR,DISPLY,INITAL,READ)  
-DATA
```

Figure D.13 Program Execution From Binary Code on Magnetic Tape

APPENDIX E

IBM-360 COMPUTER PROGRAM DATA CARDS

GNILKA, C. W. 17 DECEMBER THREE CLEARANCES NO. 1

1,000 RPM VALVE SEAT
 1000. RPM
 .7 NO CLEARANCE AT VALVE SEAT AND PUSHROD/ROCKER
 ES PRINT OUT MATRICES MPRINT
 0 CALCOMP GRAPHS DESIRED
 0 AGT OUTPUT
 0 NO PUNCHED CARDS; MAGNETIC TAPE
 .052 DEGREES TO PRINT DEBUG OUTPUT OR STOP
 0 10 PRINTED OUTPUT DESIRED; NCOUNT
 10 PRINTED OUTPUT DESIRED FOR DEBUG PURPOSES - IPRINT
 10 OFFLINE PRINTER GRAPH
 1 IDEG - - FORCED TO STAY IN CONFIG 7
 .000001 DELT- - TIME INCREMENT
 DISPL. DIFF. BETWEEN ROD & ROCKER ** Q(2)-Q(8)
 DISPL. DIFF. BETWEEN CAM & ROD ** Q(1)-Q(9)
 CAM PROFILE ** Q(9)
 DISPL. OF LOWER END OF PUSHROD ** Q(1)
 FORCE ON PUSHROD ** F(1)
 AXIAL OSCILLATIONS OF THE PUSHROD ** Q(8)-Q(1)
 DISPLACEMENT OF VALVE ** Q(7)
 POSITION OF VALVE SEAT ** Q(10)
 FORCE ON ROCKER ** F(2)
 AXIAL OSCILLATIONS OF THE VALVE STEM
 FORCE ON VALVE ** F(7)
 CAM PROFILE AND DISPL. OF PUSHROD
 DISPL. OF VALVE AND VALVE SEAT
 DEGREES OF CAM ROTATION

APPENDIX F

This appendix presents two additional examples for comparison with the 11,000 RPM examples presented in chapter 6.C.

A. EXAMPLE 3, THREE CLEARANCES, 9,000 RPM, VALVE SEAT

Approximately 800 degrees of cam rotation is indicated on the associated graphs. These graphs only indicate the major system points of interest and are not otherwise annotated. The graphs, in order of presentation are:

1. Figure F.1

a. Cam Profile, $q_9 \times 10^{-1}$

b. Displacement of lower end of pushrod

$$q_1 \times 10^{-1}$$

c. Displacement difference between cam and pushrod

$$q_1 - q_9 \times 10^{-3}$$

2. Figure F.2, Force on Pushrod

$$F_1 \times 10^2$$

3. Figure F.3, Force on Rocker

$$F_2 \times 10^2$$

4. Figure F.4, Force on Valve

$$F_7 \times 10^2$$

5. Figure F.5, Displacement of Valve

$$q_7 \times 10^{-1}$$

6. Figure F.6, Axial Oscillations of the Pushrod

$$(q_8 - q_1) \times 10^{-3}$$

7. Figure F.7, Axial Oscillation of the Valve Stem

$$(q_7 - q_4) \times 10^{-3}$$

8. Figure F.8

- a. Displacement Difference between Cam and Pushrod

$$(q_1 - q_9) \times 10^{-3}$$

- b. Displacement Difference between Pushrod and Rocker

$$(q_2 - q_8) \times 10^{-2}$$

B. EXAMPLE 3, THREE CLEARANCES 10,000 RPM, VALVE SEAT

This last example referred to in chapter 6.C. also presents approximately 800 degrees of cam rotation. These graphs only indicate the major system points of interest and are not otherwise annotated. The graphs, in order of presentation are:

1. Figure F.9

- a. Cam Profile, $q_9 \times 10^{-1}$

- b. Displacement of lower end of pushrod

$$q_1 \times 10^{-1}$$

- c. Displacement difference between cam and pushrod

$$q_1 - q_9 \times 10^{-3}$$

2. Figure F.10, Force on Pushrod

$$F_1 \times 10^2$$

3. Figure F.11, Force on Rocker

$$F_2 \times 10^2$$

4. Figure F.12, Force on Valve

$$F_7 \times 10^2$$

5. Figure F.13, Displacement of Valve

$$q_7 \times 10^{-1}$$

6. Figure F.14, Axial Oscillations of the Pushrod

$$(q_8 - q_1) \times 10^{-3}$$

7. Figure F.15, Axial Oscillation of the Valve Stem

$$(q_7 - q_4) \times 10^{-3}$$

8. Figure F.16

a. Displacement Difference between cam and pushrod

$$(q_1 - q_9) \times 10^{-3}$$

b. Displacement Difference between Pushrod and Rocker

$$(q_2 - q_8) \times 10^{-2}$$

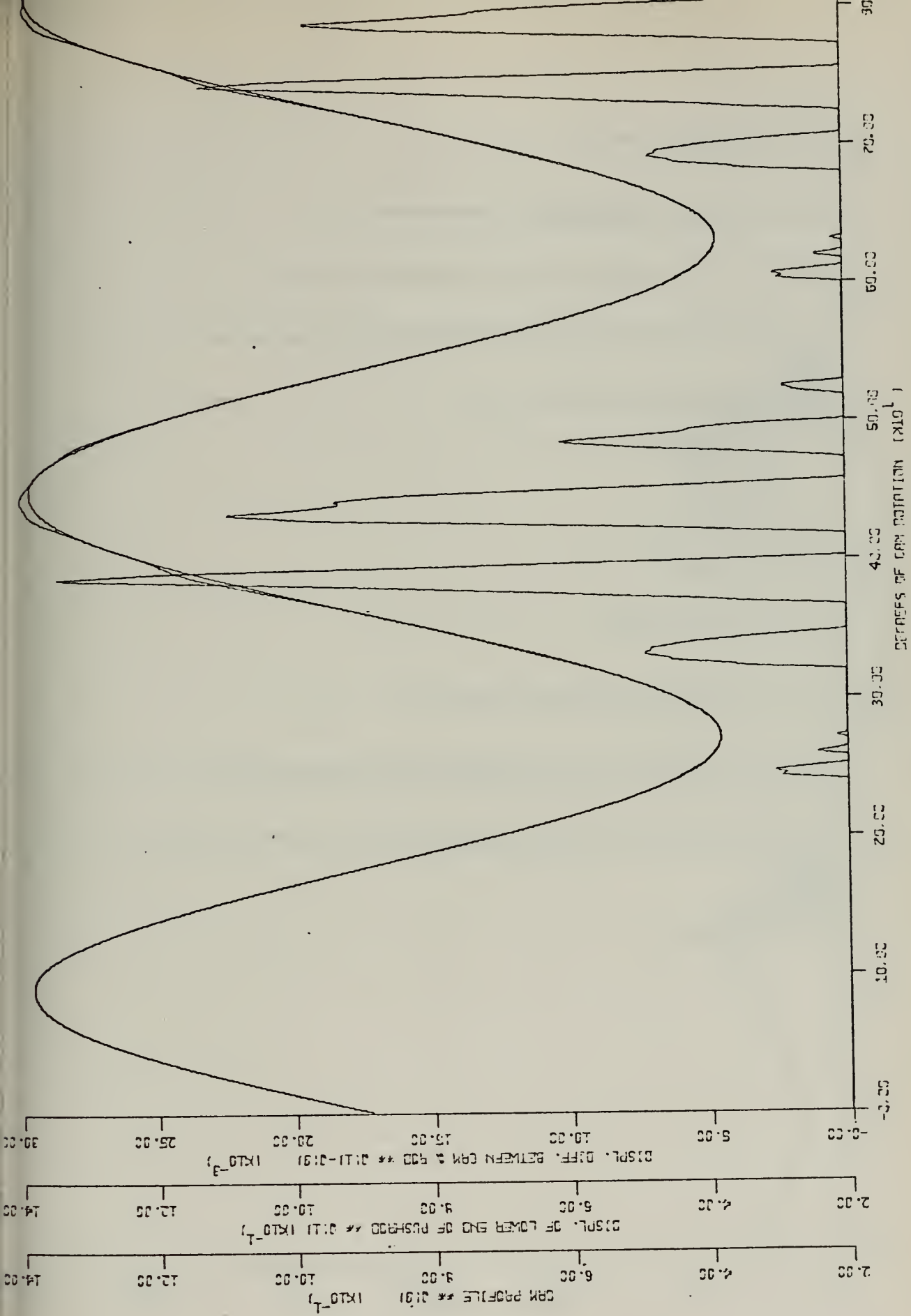


Figure F.1 THREE CLEARANCES 9,000 RPM VALVE SEAT

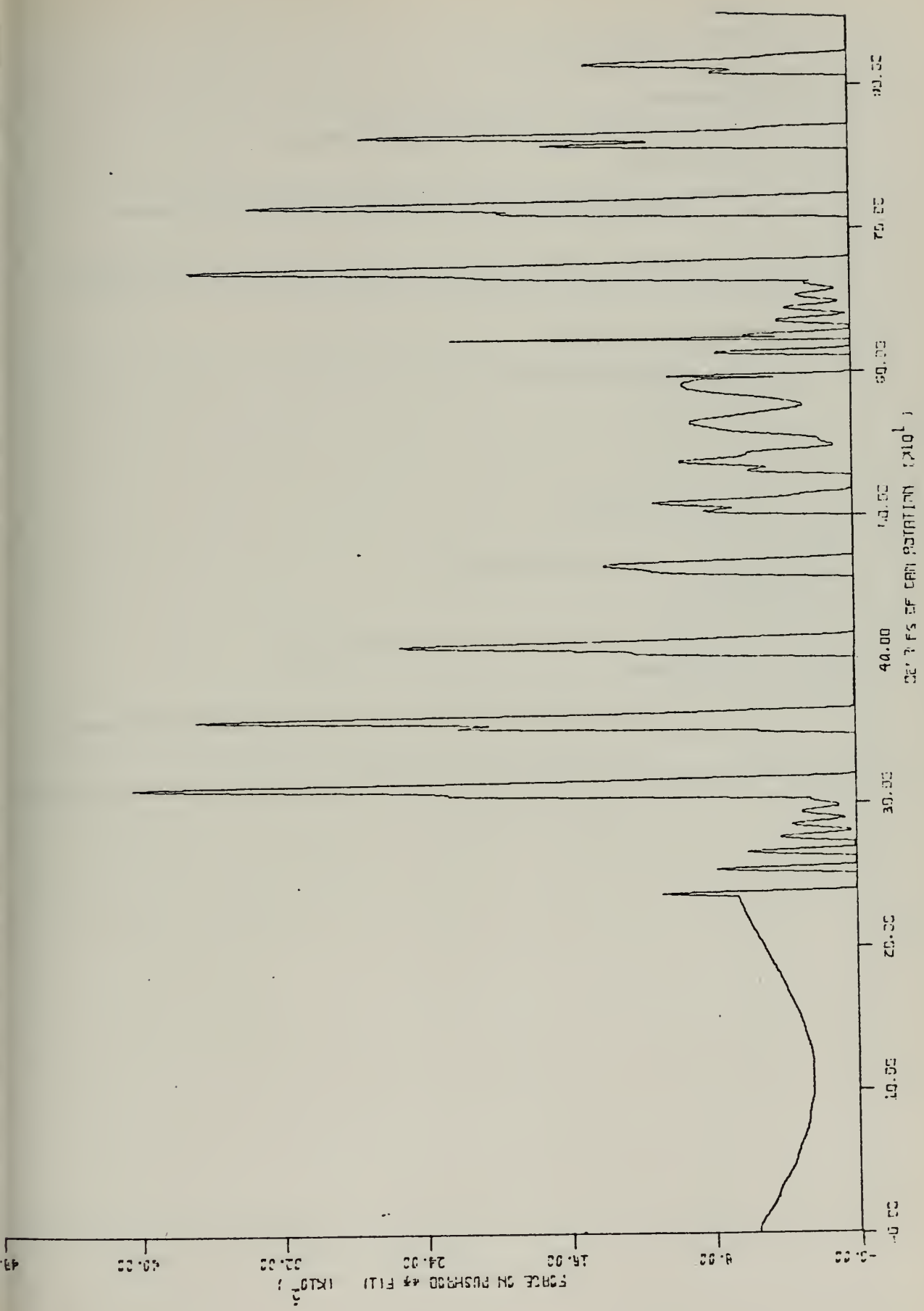


Figure F.2 THREE CLEARANCES 9,000 RPM VALVE SEAT

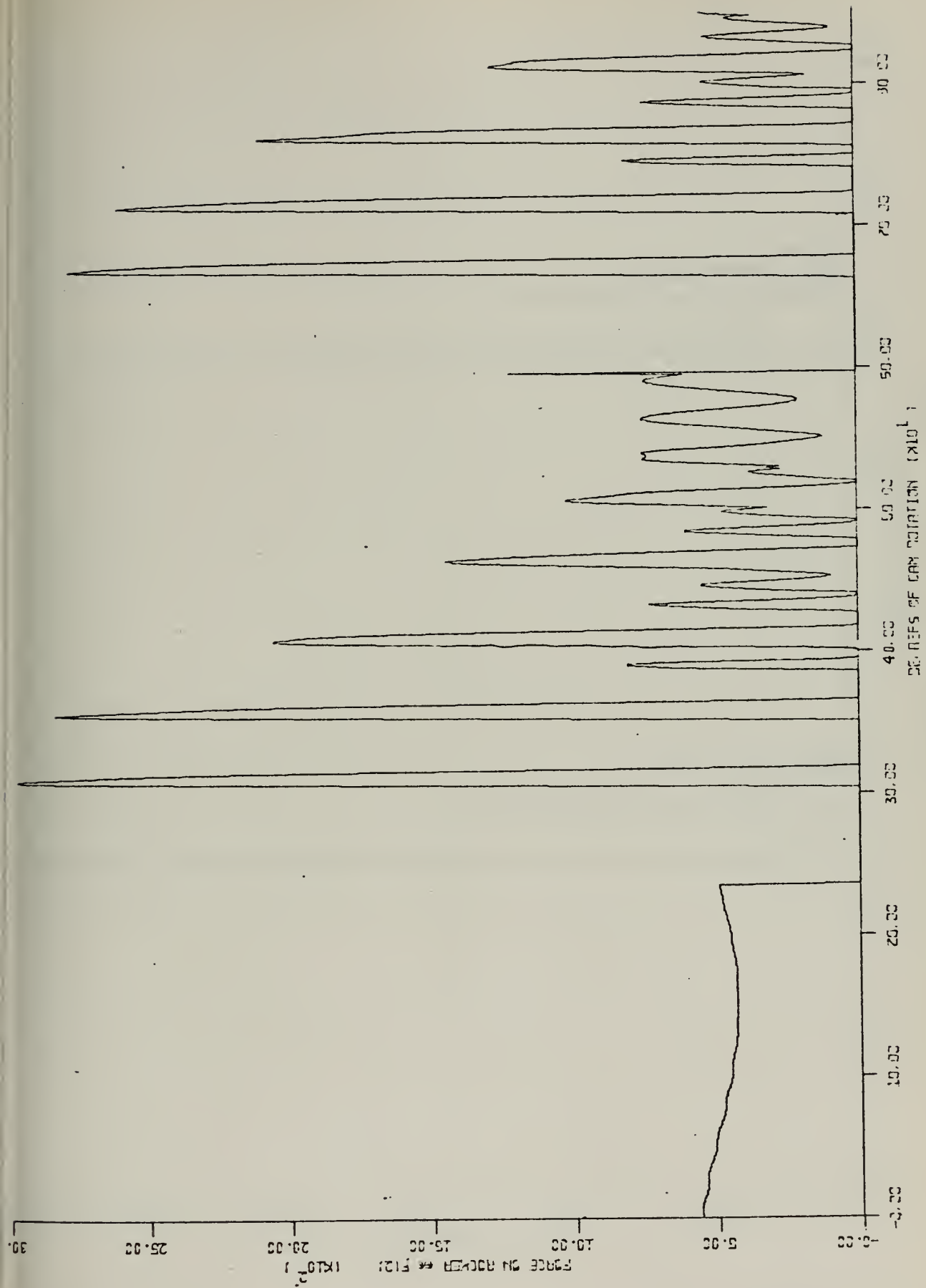
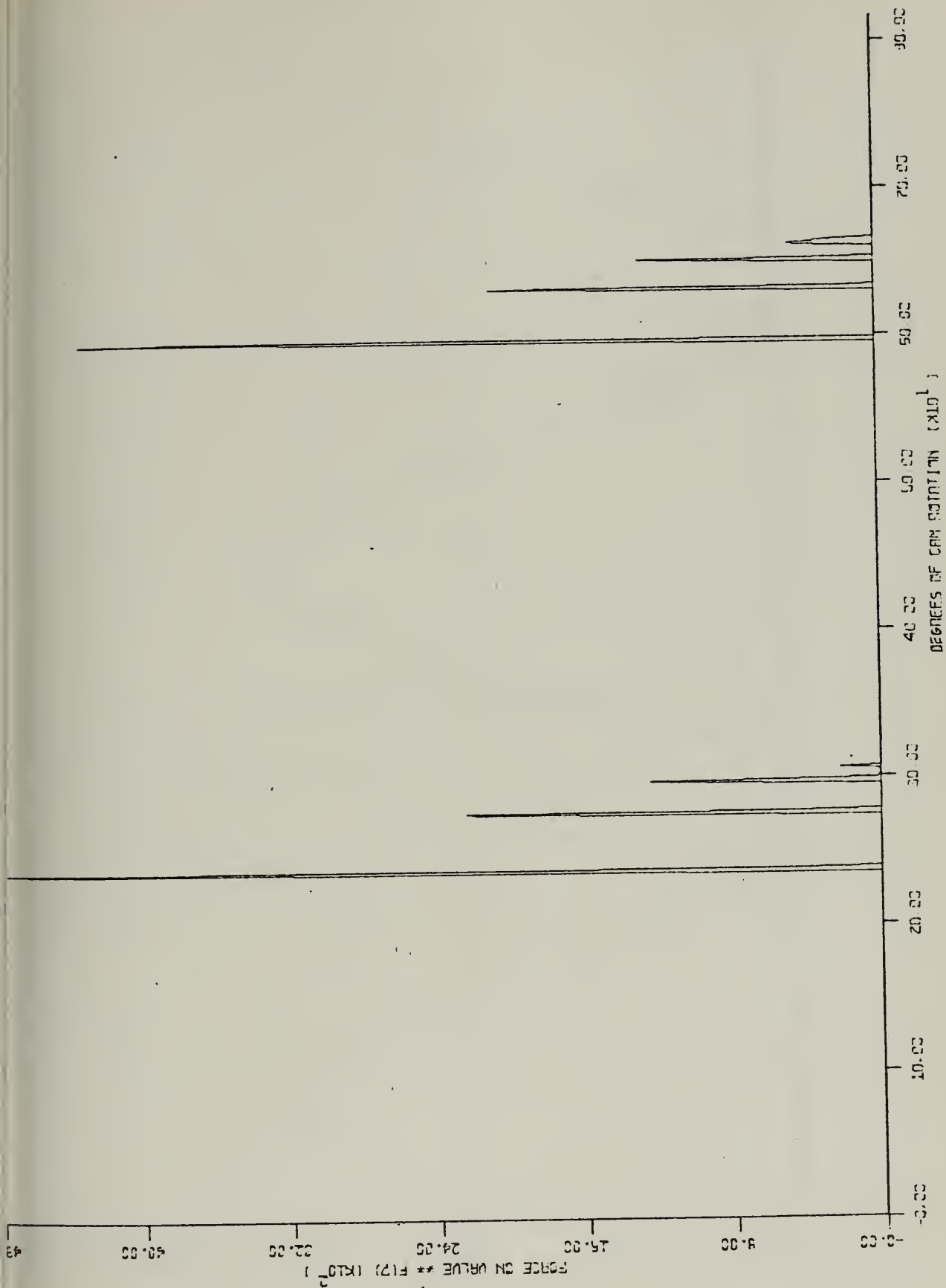


Figure F.3 THREE CLEARANCES 9,000 RPM VALVE SEAT



9.000 RPM VALVE SEAT

Figure F.4 THREE CLEARANCES

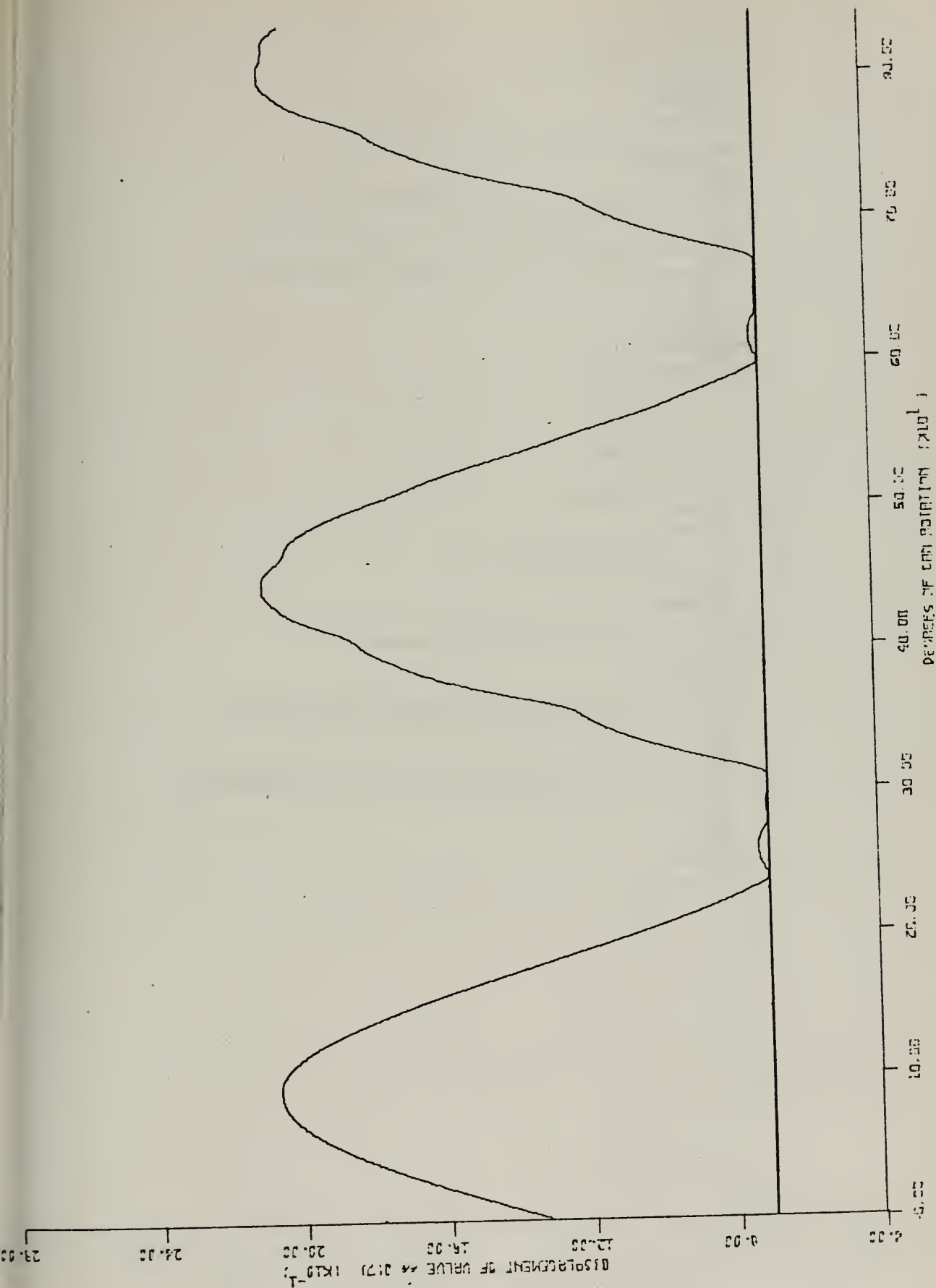
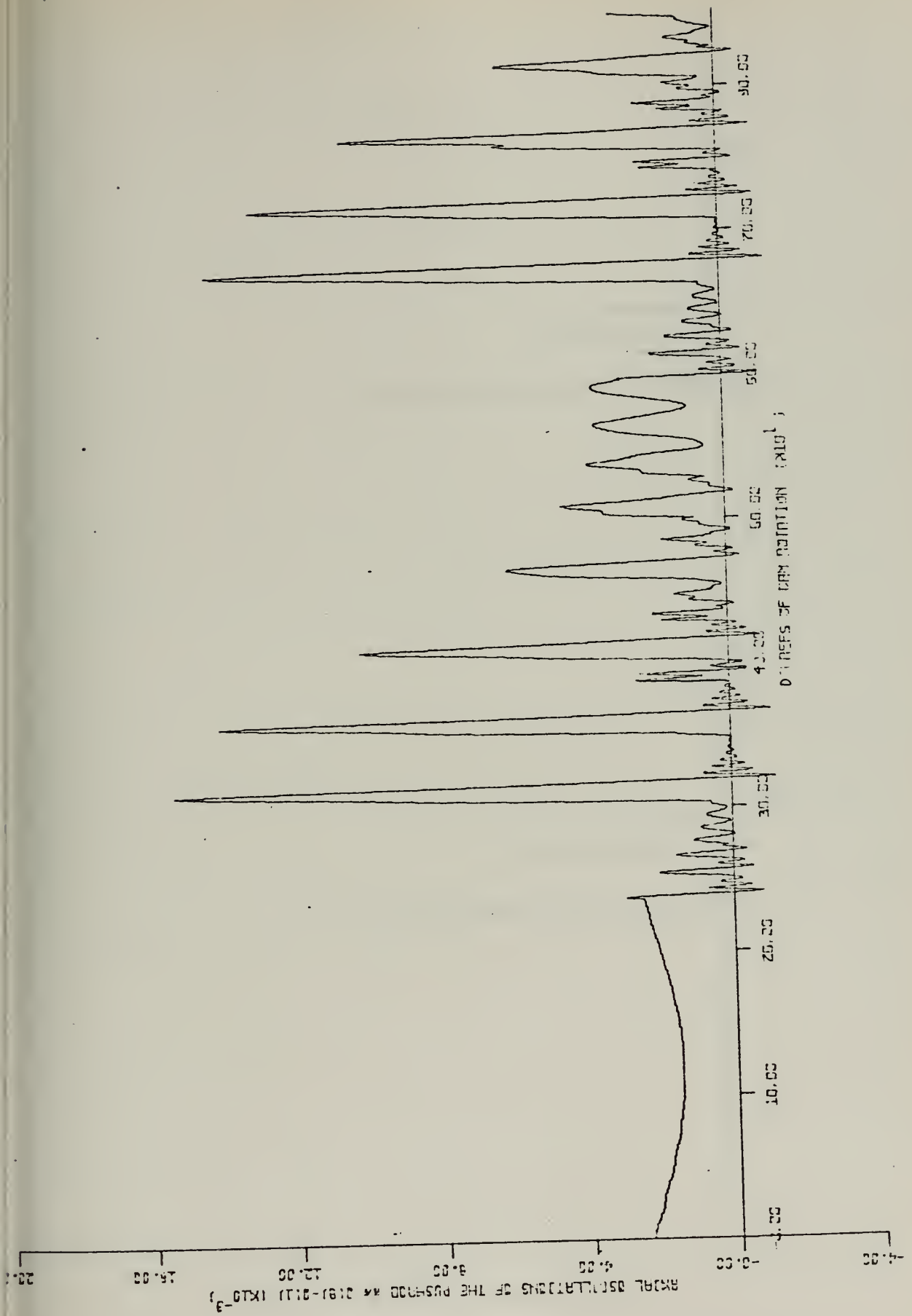


Figure F.5 THREE CLEARANCES

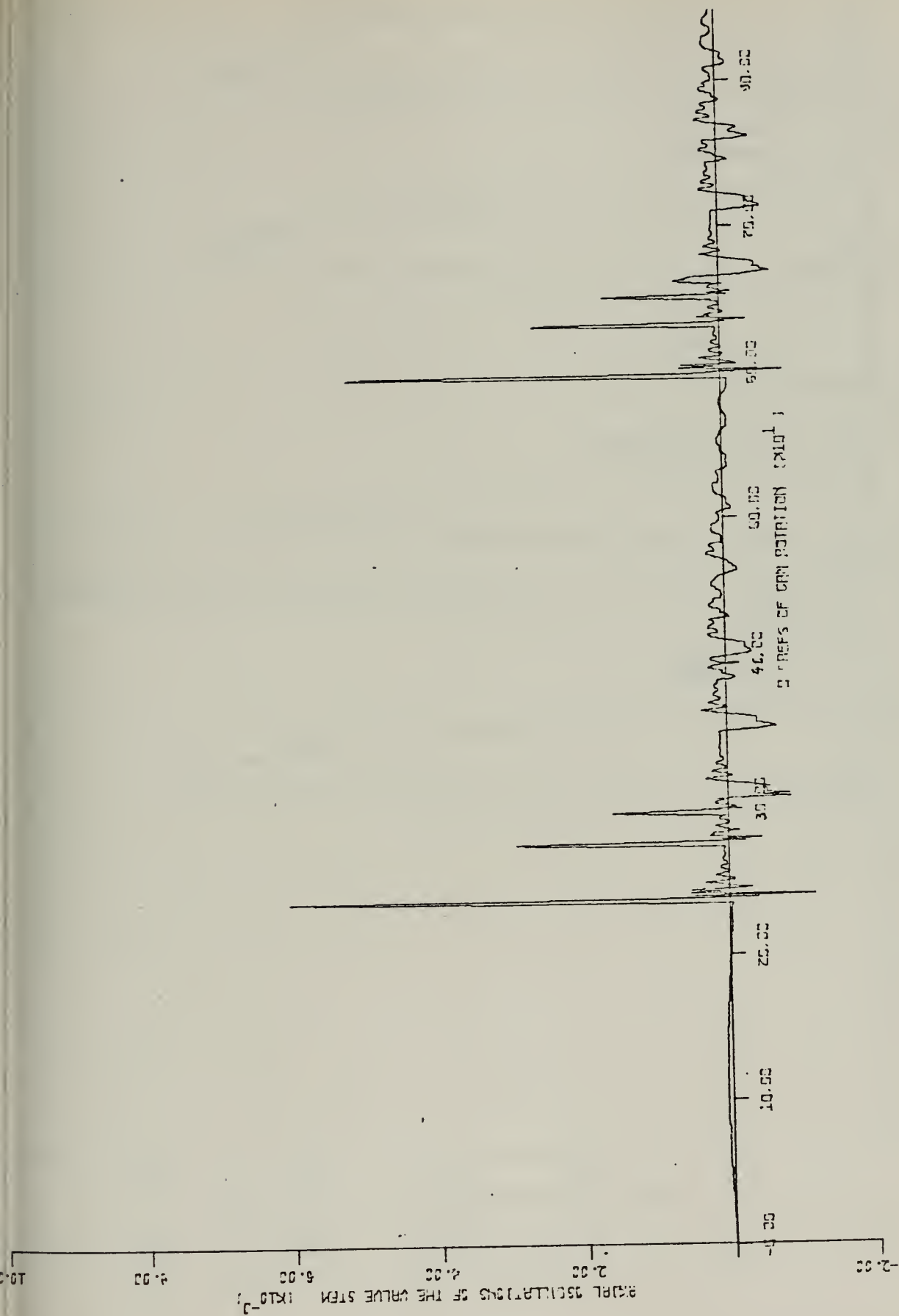
9,000 RPM

VALVE SEAT



9,000 RPM VALVE SEAT

Figure F.6 THREE CLEARANCES



9,000 RPM VALVE SEAT

THREE CLEARANCES

Figure F.7

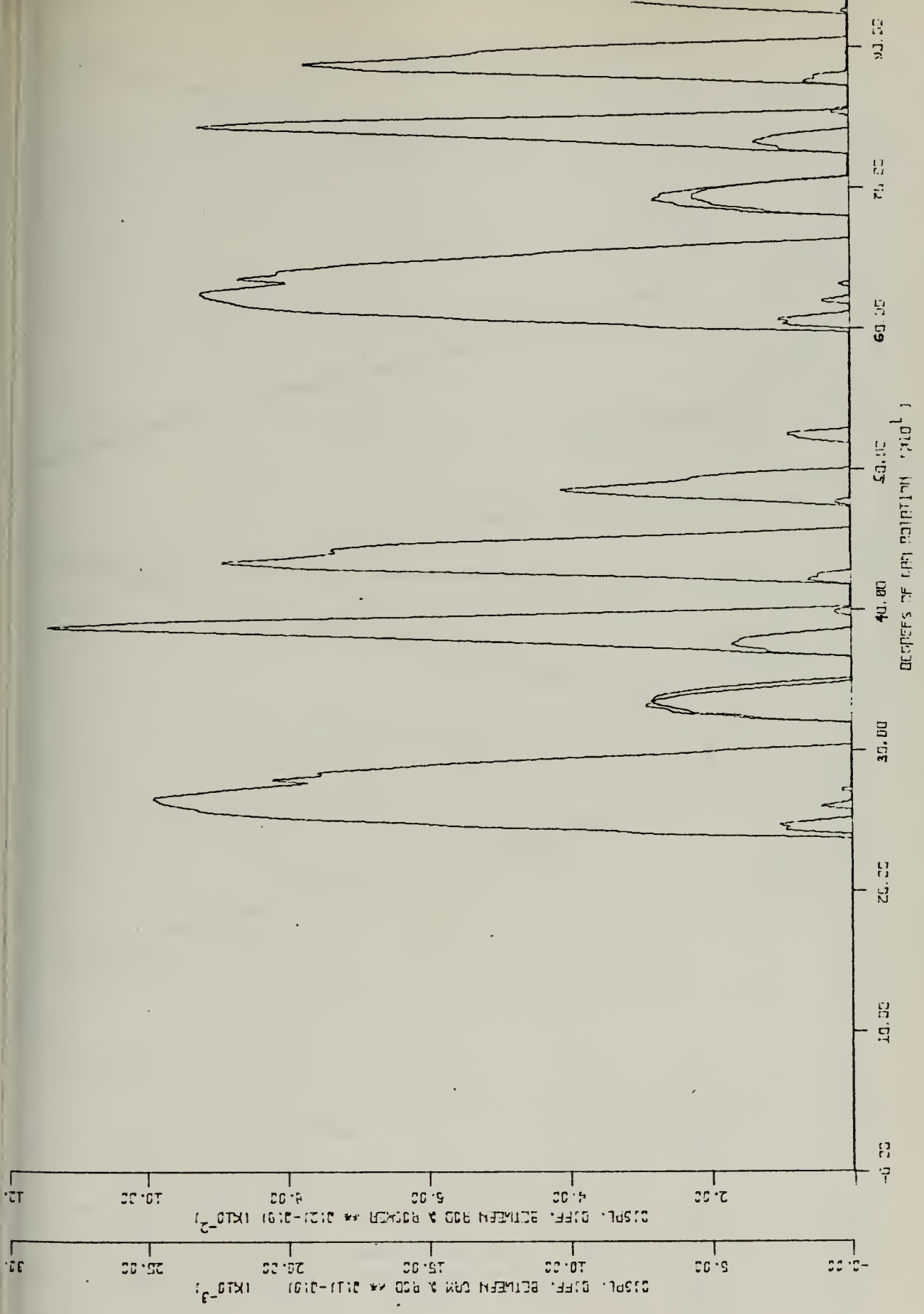


Figure F.8 THREE CLEARANCES 9,000 RPM VALVE SEAT

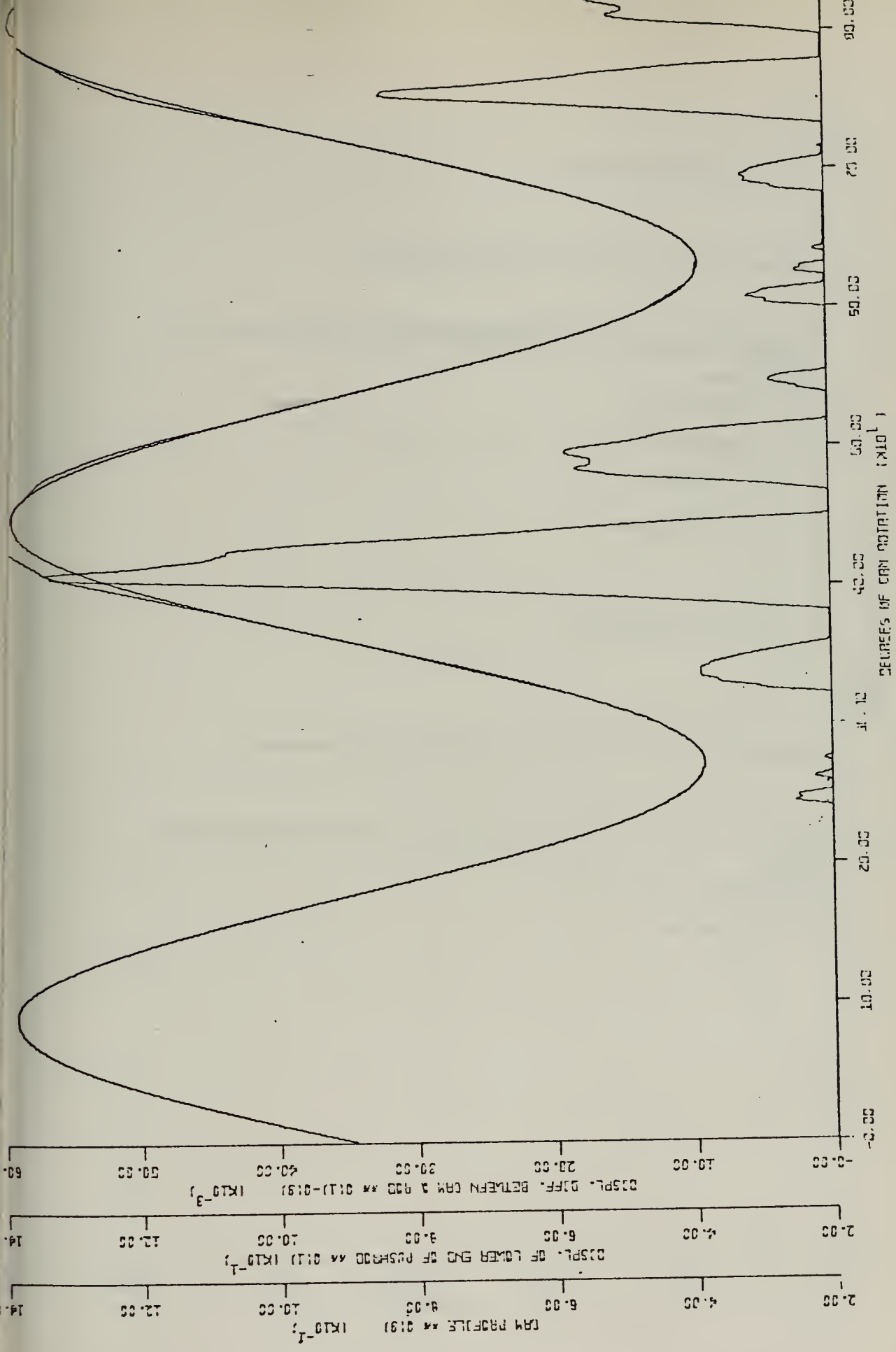


Figure F.9 THREE-CLEARANCES 10,000 RPM VALVE SEAT

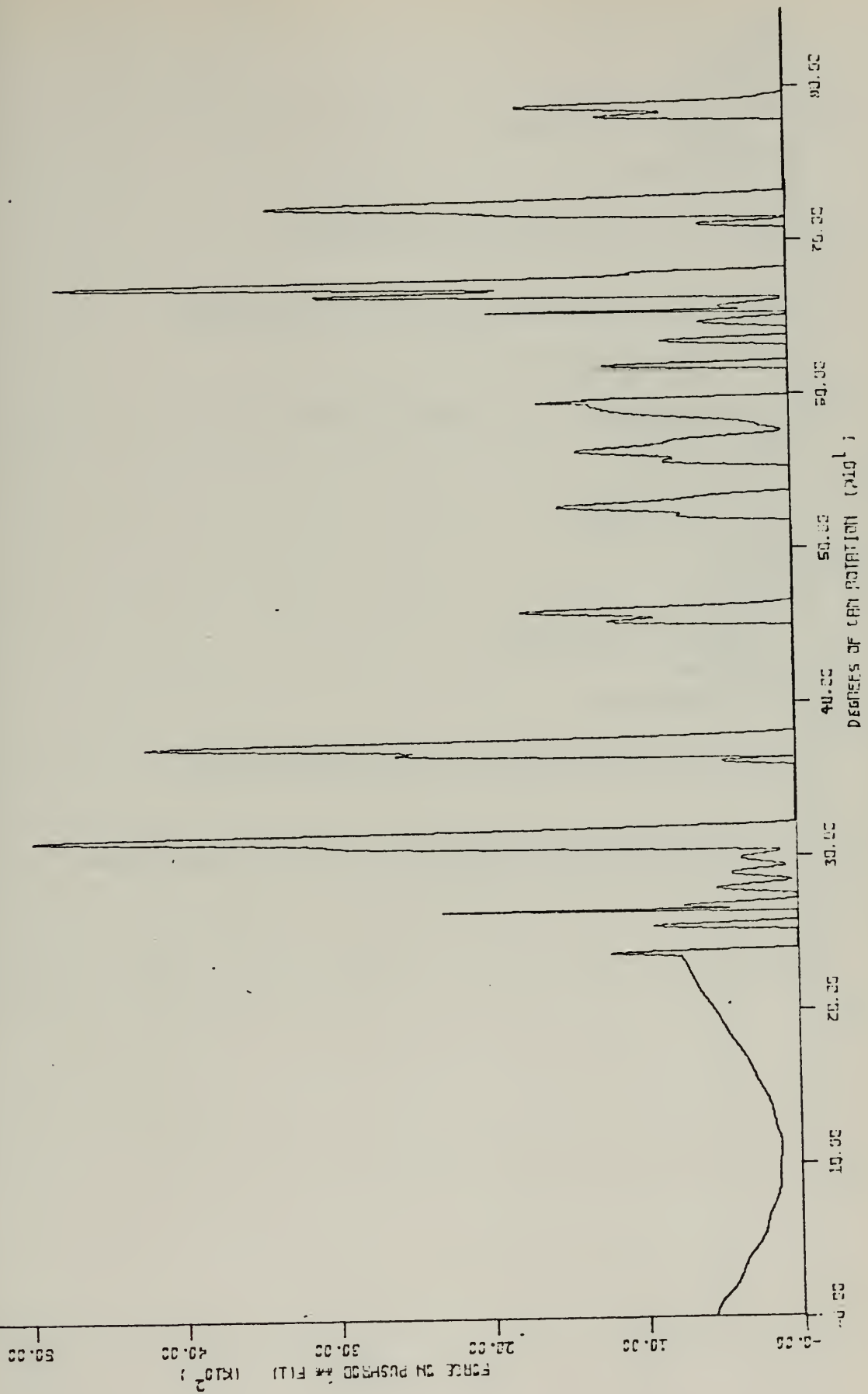


Figure F.10 THREE CLEARANCES 10,000 RPM VALVE SEAT

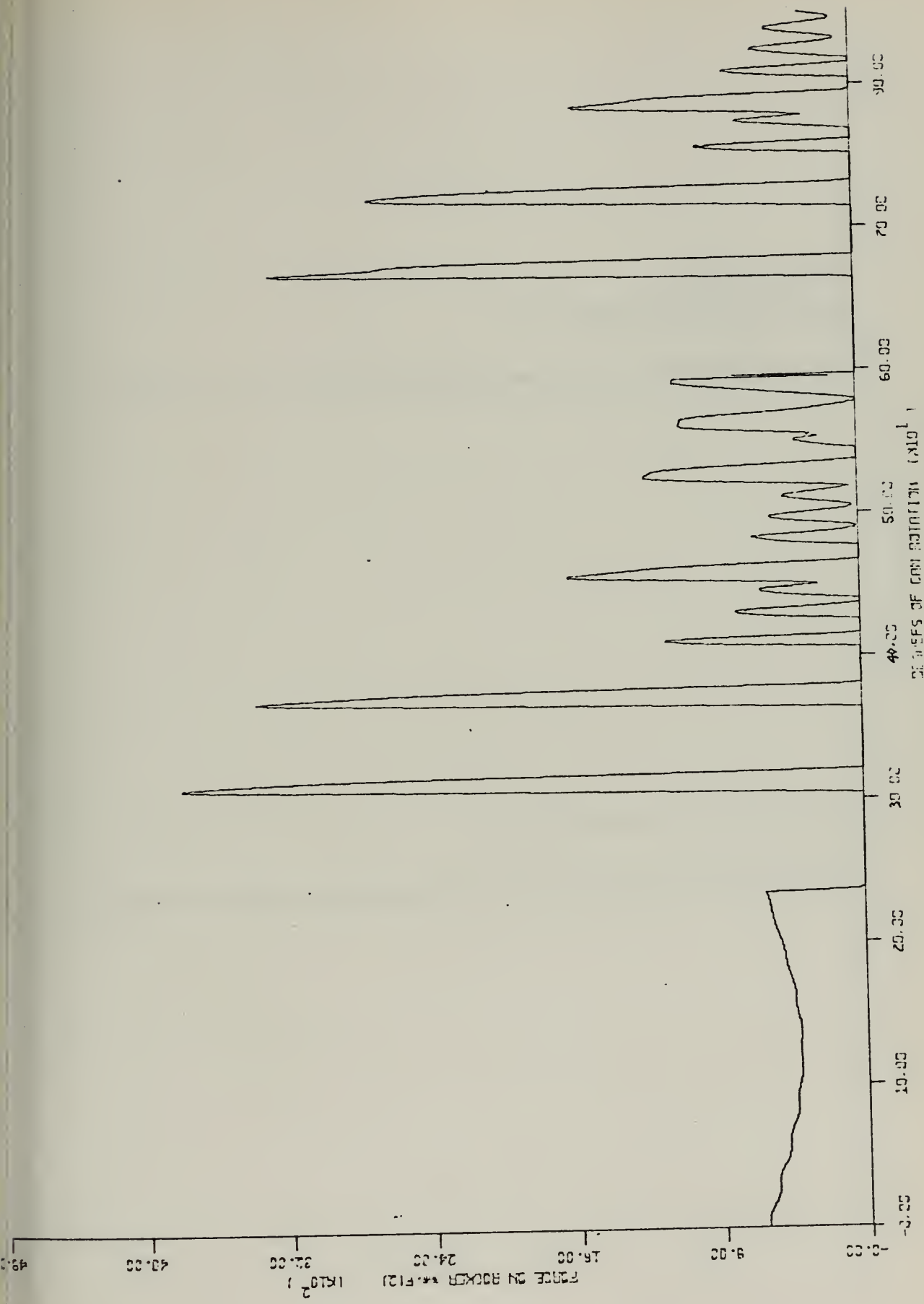


Figure F.11 THREE CLEARANCES 10,000 RPM VALVE SPNT

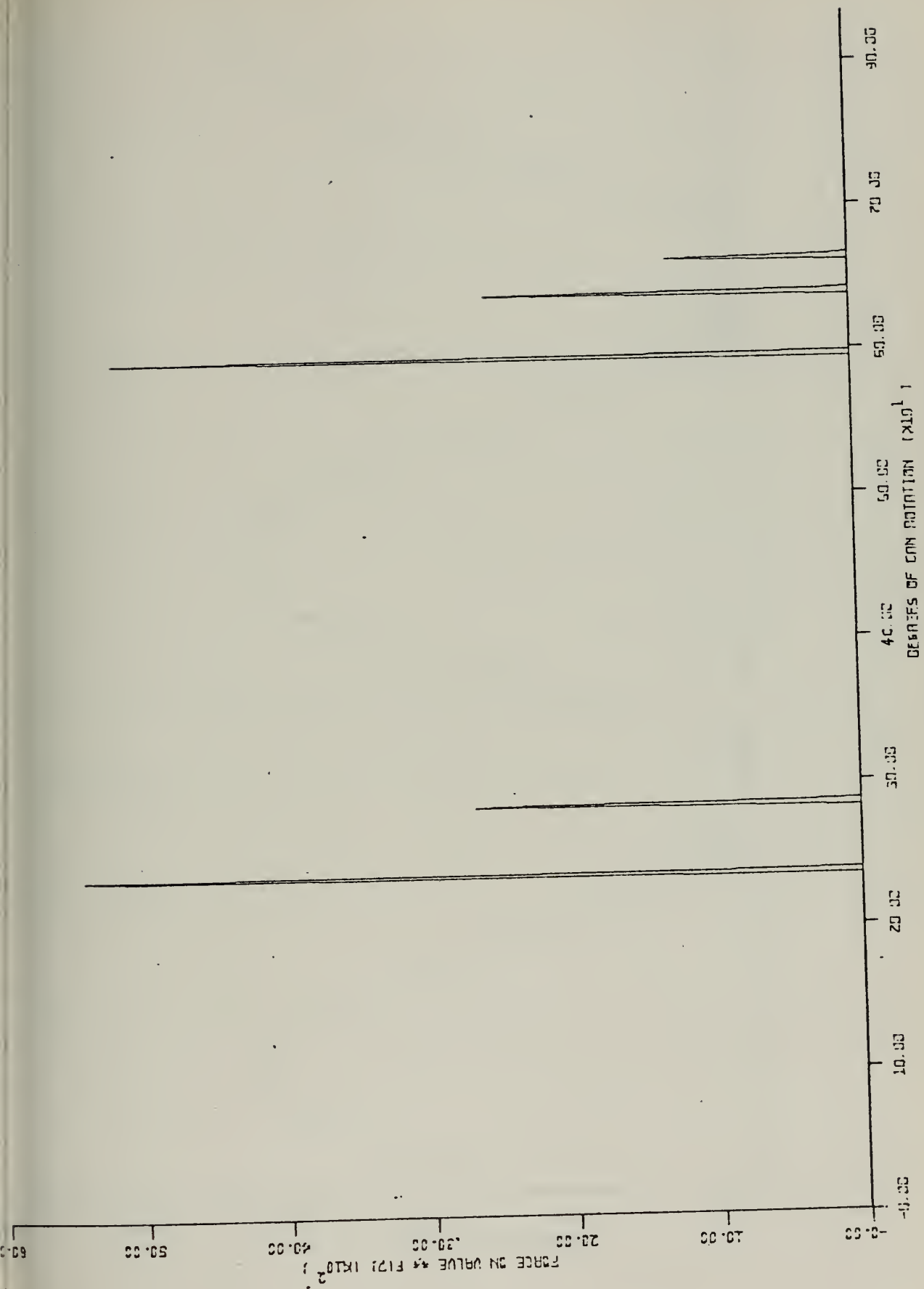


Figure F.12 THREE CLEARANCES 10,000 RPM VALVE SEAT

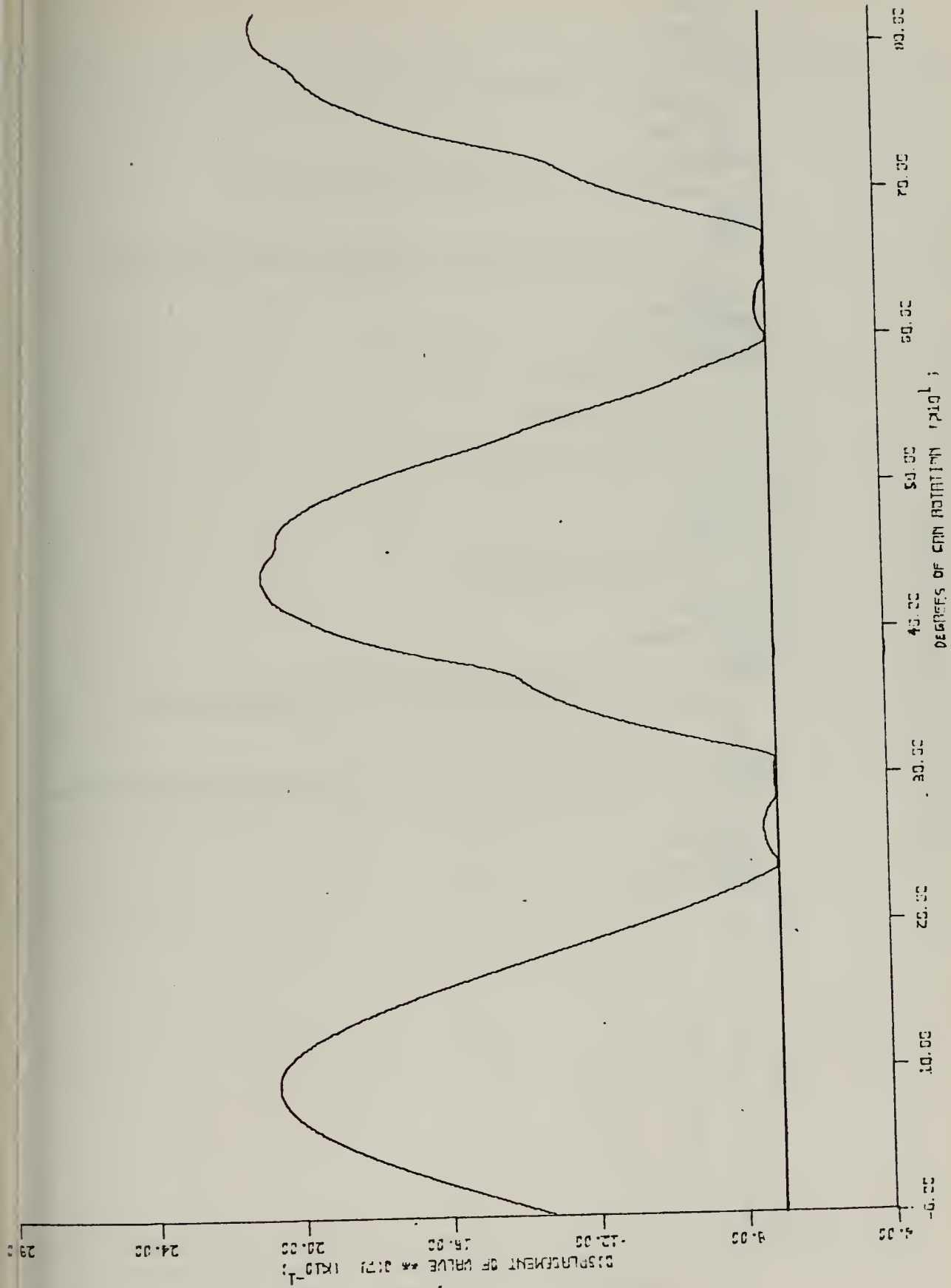


Figure F.13 THREE CLEARANCES 10,000 RPM VALVE SEAT

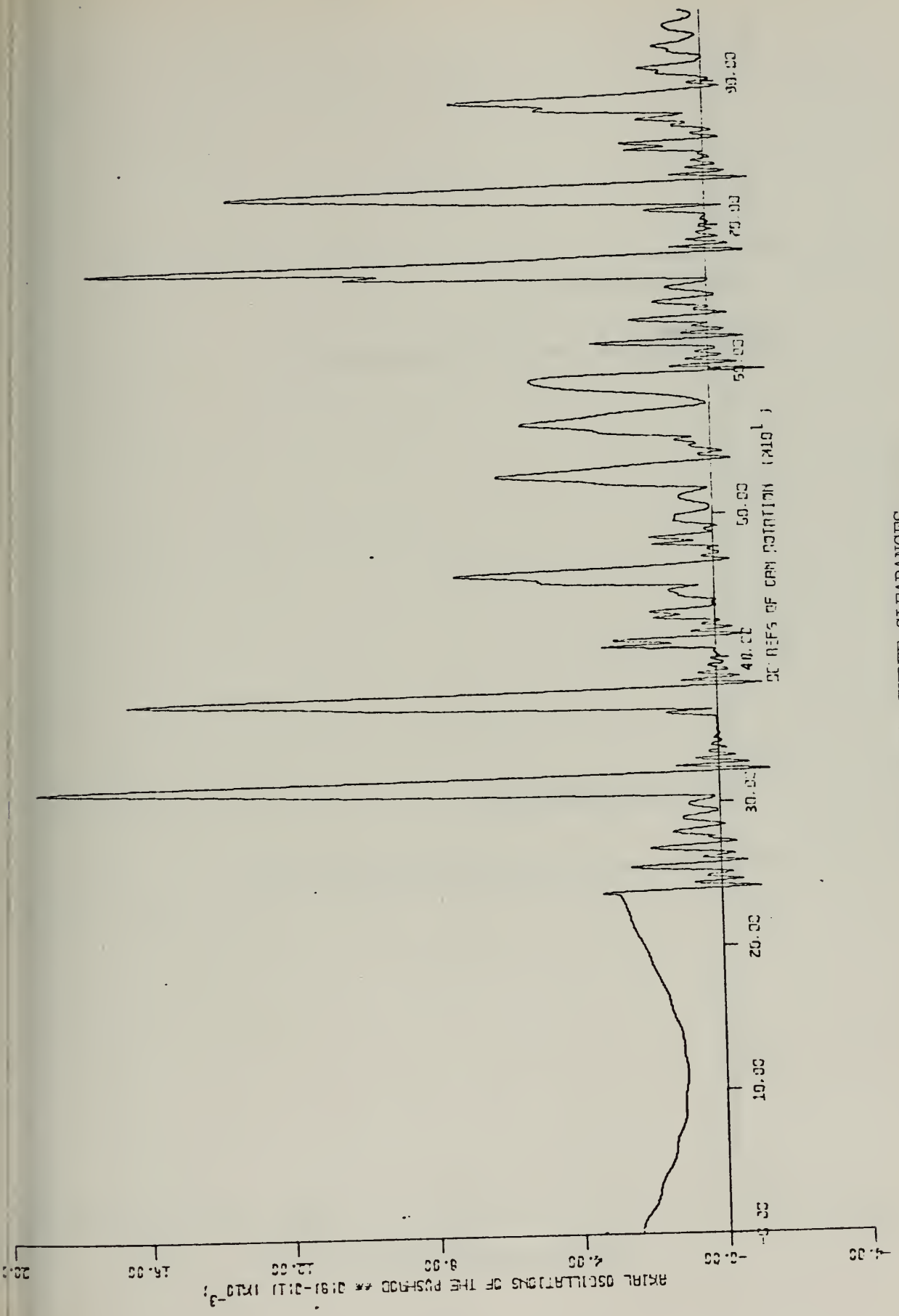


Figure F.14 THREE CLEARANCES

10,000 RPM VALVE SEAT

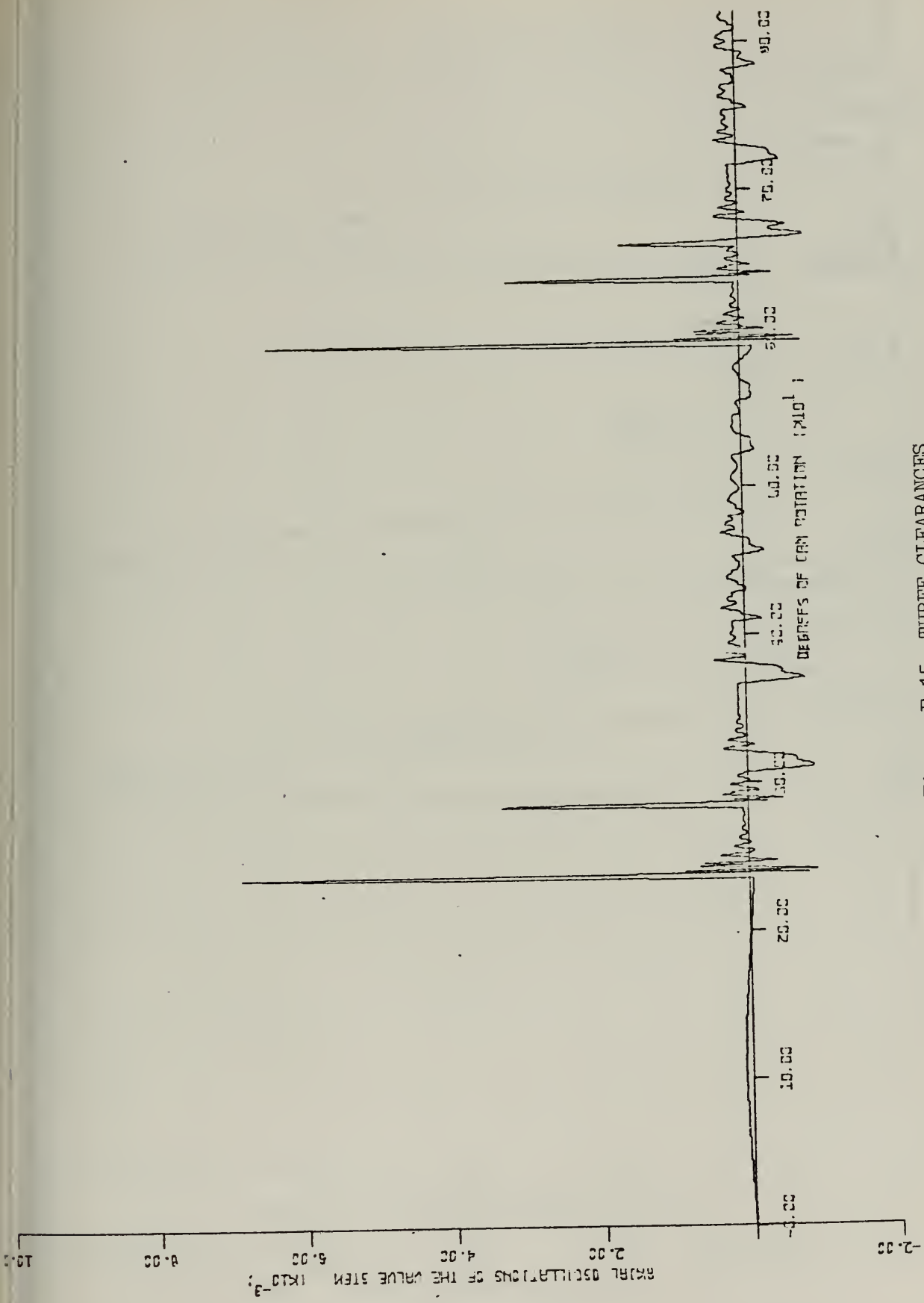


Figure F.15 THREE CLEARANCES

10,000 RPM VALVE SEAT

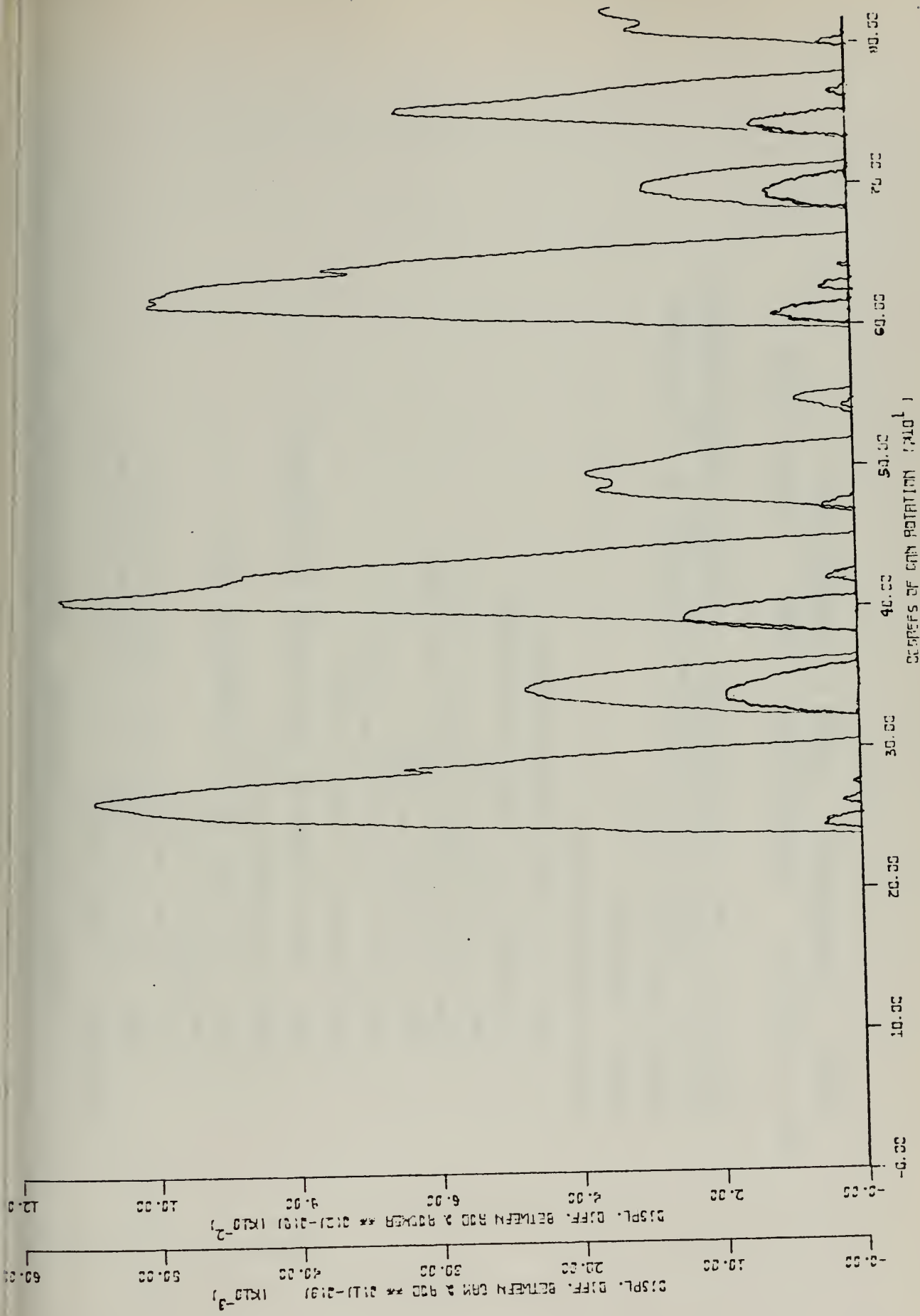


Figure F.16 THREE CLEARANCES ; 10,000 RPM VALVE SEAT

11,000 RPM VALVE SEAT

DISPL. DIFF. BETWEEN ROD & ROCKER ** Q(2)-Q(8)

DISPL. DIFF. BETWEEN CAM & ROD ** Q(1)-Q(9)

CAM PROFILE ** Q(9)

DISPL. OF LOWER END OF PUSHROD ** Q(1)

FORCE ON PUSHROD ** F(1)

AXIAL OSCILLATIONS OF THE PUSHROD ** Q(8)-Q(1)

DISPLACEMENT OF VALVE ** Q(7)

POSITION OF VALVE SEAT ** Q(10)

FORCE ON ROCKER ** F(2)

AXIAL OSCILLATIONS OF THE VALVE STEM

FORCE ON VALVE ** F(7)

CAM PROFILE AND DISPL. OF PUSHROD

DISPL. OF VALVE AND VALVE SEAT

NO AGT OUTPUT REQUESTED

DAMPING REQUESTED

CALCOMP GRAPH NOT DESIRED

VALVE SEAT IMPOSED WITH AN INITIAL CLEARANCE OF 0.70 INCHES
COORDINATES 2 AND 8 ARE NOT PINED -- THREE CLEARANCE PROBLEM

RPM = 11000.00 DAMPING COEF. = 1.000 CLEAR = 0.700

XZERO= 8.8245022D-01 FZERO = 5.0185073D 02

IBM-360 COMPUTER PROGRAM LISTING

This listing is coded in FORTRAN IV, (Refs. 10-13), to run on an IBM-360/67, with OS/360 release 18. The simulation requires 234 K bytes of core storage for 850 degrees of cam rotation, including the CALCOMP graph buffer. The offline printer array (GRID) is not included in this core storage figure. Execution time is a function of the speed of rotation of the cam. For 11,000 RPM with 850 degrees, execution time is about 4.3 minutes. The primary output parameters fill arrays for SYSOUT = B and SYSOUT = C. The B queue is for transferring data, via punched cards, for use in the SDS-9300 computer. The C queue is for plotting graphs via the CALCOMP.

C-----
 C
 C
 C
 C
 C
 C
 C
 C

CHARLES W. GNILKA - - - THESIS - - - DECEMBER 1971
 THE MODELING AND ANALYSIS OF AN ELASTIC MECHANISM WITH CLEARANCES
 THREE CLEARANCES - - - CAM - VALVE SEAT - ROCKER

```

INTEGER*2 GRID( 1, 1)
DIMENSION FPLLOT( 850,11), Y1( 850,2), Y2( 850,2), Y3(850,2),
           Y4(850,2), DEG(1700), DEG1(850), DEG2( 850,2),
           YNAME(20,13), YNAME1(20,2), YNAME2(20,2)
1 2 COMMON NUNIT, IPRINT(1,1), Y1(1,1), ( YNAME(1,1), YNAME1(1,1) ),
EQUIVALENCE ( FPLLOT(1,2), Y2(1,1) ), ( YNAME(1,2), YNAME2(1,1) ),
              ( FPLLOT(1,3), Y3(1,1) ), ( FPLLOT(1,7), Y4(1,1) ),
              ( DEG(1), DEG2(1,1) ), ( DEG(851), DEG1(1) )
1 2 3 DATA YNAME(19,1) //, YNAME(20,1) //, ( YNAME(1,1), YNAME1(1,1) ),
DATA YNAME(19,2) //, YNAME(20,2) //, ( YNAME(1,2), YNAME2(1,1) ),
DATA YNAME(19,3) //, YNAME(20,3) //, ( FPLLOT(1,2), YNAME2(1,1) ),
DATA YNAME(19,4) //, YNAME(20,4) //, ( FPLLOT(1,3), Y4(1,1) ),
DATA YNAME(19,5) //, YNAME(20,5) //, ( DEG(1), DEG2(1,1) ),
DATA YNAME(19,6) //, YNAME(20,6) //, ( DEG(851), DEG1(1) )
DATA YNAME(19,7) //, YNAME(20,7) //, //
DATA YNAME(19,8) //, YNAME(20,8) //, //
DATA YNAME(19,9) //, YNAME(20,9) //, //
DATA YNAME(19,10) //, YNAME(20,10) //, //
DATA YNAME(19,11) //, YNAME(20,11) //, //
DATA YNAME(19,12) //, YNAME(20,12) //, //
DATA YNAME(19,13) //, YNAME(20,13) //, //
IP = 0
LINES = 80*IP + 1
NTIME = 850
IGU = YES
IGU = NO
SCALE = 1.1D0
CALL THESIS(FPLLOT, DEG, DEG1, DEG2, NTIME, LINES, IP, GRID, Y1,
            Y2, Y3, Y4, YNAME, YNAME1, YNAME2, IGU, SCALE)
1 STOP
END

```

00000010
 00000020
 00000030
 00000040
 00000050
 00000060
 00000070
 00000080
 00000090
 00000100
 00000110
 00000120
 00000130
 00000140
 00000150
 00000160
 00000170
 00000180
 00000190
 00000200
 00000210
 00000220
 00000230
 00000240
 00000250
 00000260
 00000270
 00000280
 00000290
 00000300
 00000310
 00000320
 00000330
 00000340
 00000350
 00000360
 00000370
 00000380
 00000390
 00000400
 00000410


```

C          SUBROUTINE ARRAY(F, Q, QD, QDD, NDEG, FPLOTT, NTIME)
C          IMPLICIT REAL*8(A-H,O-Z)
C          REAL*4 FPLOTT
C          INTEGER YES
C          DIMENSION F(1), Q(1),QD(1),QDD(1),FPLOTT(NTIME,1)
C          COMMON NUNIT, IPRINT
C          DATA YES/4HYES /, NO/4HNO /
C          FPLOTT(NDEG,1) = Q(2) - Q(8)
C          FPLOTT(NDEG,2) = Q(1) - Q(9)
C          FPLOTT(NDEG,3) = Q(9)
C          FPLOTT(NDEG,4) = Q(1)
C          FPLOTT(NDEG,5) = F(1) - Q(8)
C          FPLOTT(NDEG,6) = Q(1) - Q(7)
C          FPLOTT(NDEG,8) = Q(10)
C          FPLOTT(NDEG,9) = F(2)
C          FPLOTT(NDEG,10) = Q(7) + Q(4)
C          FPLOTT(NDEG,11) = F(7)
C          RETURN
C          END
00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290

```



```

C      CALL DEL( DTPM, DT02, DT203, DT206, BMIV, DA, DV, SYSM(1,1,C),
C      DAMP(1,1,C), STF(1,1,C), ISYS, C, NPRINT, KSYS, KPRINT)
C      1 IF(NPRINT .EQ. YES)
C      WRITE(6,5009) DTPM, C
C      1 CALL BRANCH(C, FP, KSYS, ISYS, QP, QDP, QDDP, QDP, ICORD, DT,
C      NPRINT, NCLEAR, 0, PIN2$8, IDIM)
C      1 IF(NPRINT .EQ. YES)
C      WRITE(6,5010) C, (FP(JA), JA= 1,KSYS)
C      1 CALL NEWMRK( ISYS, KSYS, C, STF, DAMP, SYSM, DP, DTPM, DT02, ALFA,
C      DT206, DT203, QP, QDP, QDDP, BMIV, DA, DV, NC, ALFA,
C      1 BET A, CONFIG, FP, ICORD, NCLEAR, NPRINT, IQ, COLM,
C      2 COLC, COLK, IDP, PIN2$8, IDIM, LAYER, IGU, QDD$AV,
C      3 QDSAV, Q$AV, 0.0, 0)
C      4 IF(NPRINT .EQ. YES)
C      WRITE(6,5011) (FP(JA), JA= 1,KSYS)
C      1 DISPL1 = QP(1) - QP(9)
C      DISPL2 = QP(2) - QP(8)
C      DISPL7 = QP(7) - QP(10)
C      1 IF(NPRINT .EQ. YES)
C      WRITE(NUNIT,5013) DISPL1, DISPL2, DISPL7
C      INDEX = ICORD(IDT,1)
C      NDEX = ICORD(IDT,2)
C      IF(CONFIG(C, IDT) .EQ. 0) GO TO 25
C      IF(FP(INDEX) .GE. 0.0 ) GO TO 24
C      GO TO 30
C      24 IF(KPRINT .EQ. YES .AND. IK .GT. 1)
C      1 WRITE(NUNIT,5001) IK, INDEX, NDEX, C
C      GO TO 40
C

```

```

00000950
00000960
00000970
00000980
00000990
00001000
00001010
00001020
00001030
00001040
00001050
00001060
00001070
00001080
00001090
00001100
00001110
00001120
00001130
00001140
00001150
00001160
00001170
00001180
00001190
00001200
00001210
00001220
00001230
00001240
00001250
00001260
00001270
00001280
00001290
00001300
00001310
00001320
00001330
00001340

```



```

00001350
00001360
00001370
00001380
00001390
00001400
00001410
00001420
00001430
00001440
00001450
00001460
00001470
00001480
00001490
00001500
00001510
00001520
00001530
00001540
00001550
00001560
00001570
00001580
00001590
00001600
00001610
00001620
00001630
00001640
00001650
00001660
00001670
00001680
00001690
00001700

25 DISPL = QP(INDEX) - QP(NDEX)
C
IF(DISPL.GE.0.0) GO TO 28
IF(DABS(DISPL).LE.EPS2) GO TO 28
GO TO 30
C
28 IF(KPRINT.EQ.YES.AND. IK.GT.1)
1 WRITE(NUNIT,5012) IK, INDEX, NDEX, C
GO TO 40
C
C
30 DTPM = DTPM - DELDT
WT = WT - DELWT
C
IF(WT.LE.WTSAV) RETURN
C
GO TO 4
C
40 CONTINUE
C
DEGR = WT*DEGOPI
NDEG = DEGR
C
WRITE(NUNIT,5014) DEGR,C,ICHNG,DISPL1,DISPL2,DISPL7,
1 FP(1),FP(2),FP(7)
IF(NPRINT.EQ.NO) RETURN
C
C
WRITE(NUNIT,5017)
C
CALL DEBUG( 2, Q, QD, QDD, F, KSYS, ISYS, DP, NDEG, NPRINT, DEGR,
1
C
RETURN
END

```



```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460

C SUBROUTINE BRANCH(C,F,KSYS,ISYS,Q,QD,QDD,QDSAV,ICORD,DT,
1 NPRINT,NCLEAR,IMP,PIN2$8,IDIM)
C IMPLICIT REAL*8(A-H,O-Z)
C INTEGER YES,C,PIN2$8
C DIMENSION Q(1),QD(1),QDD(1),QDSAV(1),F(1),ICORD(NCLEAR,1),
1 IDIM(1)
C COMMON NUNIT,IPRINT
C DATA YES/4HYES/,NO/4HNO/
C 5 ISYS = IDIM(C)
C GO TO ( 10, 20, 30, 40, 50, 60, 70, 80), C
C
C CONFIGURATION - I - NO CONTACT
C 10 F(1) = 0.0D0
C F(2) = 0.0D0
C F(7) = 0.0D0
C GO TO 100
C
C CONFIGURATION -II - CONTACT AT VALVE SEAT
C 20 CONTINUE
C CALL ADJUST(7,10,Q,QD,QDD,Q,QD,QDD)
C
C F(1) = 0.0D0
C F(2) = 0.0D0
C GO TO 100
C
C CONFIGURATION - III - CONTACT AT ROCKER
C 30 F(1) = 0.0D0
C F(7) = 0.0D0
C GO TO 100
C
C CONFIGURATION - IV - CONTACT AT ROCKER AND VALVE SEAT
C 40 CONTINUE
C CALL ADJUST(7,10,Q,QD,QDD,Q,QD,QDD)
C
C F(1) = 0.0D0

```


00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630
00000640
00000650
00000660
00000670
00000680
00000690
00000700
00000710
00000720
00000730
00000740
00000750
00000760
00000770
00000780
00000790
00000800
00000810
00000820
00000830
00000840
00000850
00000860
00000870
00000880
00000890
00000900
00000910
00000920
00000930

```

C      GO TO 100
C
C      CONFIGURATION - V -CONTACT AT CAM
C      CONTINUE
C      50  CALL ADJUST(1,9, Q, QD, QDD, Q, QD, QDD)
C
C      F(2) = 0.0D0
C      F(7) = 0.0D0
C      GO TO 100
C
C      CONFIGURATIGN -VI -CONTACT AT CAM AND VALVE SEAT
C      CONTINUE
C      60  CALL ADJUST(1,9, Q, QD, QDD, Q, QD, QDD)
C      CALL ADJUST(7,10, Q, QD, QDD, Q, QD, QDD)
C
C      F(2) = 0.0D0
C      GO TO 100
C
C      CONFIGURATION - VII - CONTACT AT ROCKER AND CAM
C      CONTINUE
C      70  CALL ADJUST(1,9, Q, QD, QDD, Q, QD, QDD)
C
C      F(7) = 0.0D0
C      GO TO 100
C
C      CONFIGURATION - VIII - CONTACT AT ROCKER, CAM, AND VALVE SEAT
C      CONTINUE
C      80  CALL ADJUST(1,9, Q, QD, QDD, Q, QD, QDD)
C      CALL ADJUST(7,10, Q, QD, QDD, Q, QD, QDD)
C
C      100 CONTINUE
C      IF(IMP .NE. 0) QDD(IMP) = ( QD(IMP)-QDSAV(IMP) ) / DT
C
C      RETURN
C      END

```



```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400

```

```

SUBROUTINE CONMAT(NC, CONFIG, IDP, NPRINT, KSYS)
IMPLICIT REAL*8(A-H,O-Z)
INTEGER CONFIG(16,4), CSAVE, TWON, TWONP1, YES
DIMENSION IDP(KSYS,1)
COMMON NUNIT, IPRINT
DATA YES/4HYES /, NO/4HNO /

```

```

*****
**          ESTABLISH CONFIGURATION
**          MATRIX
**          *****

```

```

100 NC = 4
    TWON = 2**NC
    TWONP1 = TWON + 1
    NCM1 = NC - 1
    NCM2 = NC - 2
    NCM3 = NC - 3

```

```

C
DO 110 I = 2, TWON, 2
  CONFIG(I,NC) = 1
CONTINUE

```

```

C
DO 120 I = 2, TWON, 4
  CONFIG(I+1,NCM1) = 1
  CONFIG(I+2,NCM1) = 1
CONTINUE

```

```

C
DO 130 I = 1, NC
  CONFIG(I+4,NCM2) = 1
  CONFIG(I+12,NCM2) = 1
CONTINUE

```


00000470
 00000480
 00000490
 00000500
 00000510
 00000520
 00000530
 00000540
 00000550
 00000560
 00000570
 00000580
 00000590
 00000600
 00000610
 00000620
 00000630
 00000640
 00000650
 00000660
 00000670
 00000680
 00000690
 00000700
 00000710
 00000720
 00000730

```

2 CONTINUE
IF(NDEG .GT. 200)
WRITE(NUNIT,5061)
WRITE(NUNIT,5007)
WRITE(NUNIT,5063)
WRITE(NUNIT,5002)
WRITE(NUNIT,5003)
WRITE(NUNIT,5004)
WRITE(NUNIT,5037)
RETURN
C
C
3 CONTINUE
RETURN
C
C
4 CONTINUE
RETURN
C
C
5 CONTINUE
RETURN
END
C

```



```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460

SUBROUTINE DEL( DT, DT02, DT203, DT206, BMIV, DA, DV, EM,
1 SE, CA, KS, C, NPRINT, KSYS, KPRINT)
C
C IMPLICIT REAL*8(A-H,O-Z)
C
C INTEGER YES, C
C
C DIMENSION BMIV( KSYS,KSYS), DA( KSYS,KSYS), DV( KSYS,KSYS)
C DIMENSION EM( KSYS,KSYS), SE( KSYS,KSYS), CA( KSYS,KSYS)
C
C COMMON NUNIT, IPRINT
C DATA YES/4HYES /, NO/4HNO /
C
1000 1. FORMAT('0',5X,'DT =',1PD12.5,' DT/2 =',D12.3,
C DT**2/3 =',D12.3,' DT**2/6 =',D12.3}
C
C DT02 = DT*0.5D0
C DT203 = DT**2/3.0D0
C DT206 = DT203*0.5D0
C
C IF(NPRINT.EQ.NO) GO TO 10
C WRITE(NUNIT,1000) DT, DT02, DT203, DT206
C
C 10 DO 20 JA = 1, KS
C
C DO 20 JB = 1, KS
C BMIV(JA,JB) = EM(JA,JB) + DT02*SE(JA,JB) + DT206*CA(JA,JB)
C
C 20 CONTINUE
C
C CALL DINVS( BMIV, KS,KSYS)
C
C DO 30 JA = 1, KS
C
C DO 30 JB = 1, KS
C DA( JA, JB) = 0.0D0
C DV( JA, JB) = 0.0D0
C
C DO 30 JC = 1, KS
C DA( JA, JB) = DA( JA, JB) + BMIV( JA, JC)*SE( JC, JB)
C DV( JA, JB) = DV( JA, JB) + BMIV( JA, JC)*CA( JC, JB)
C
C 30 CONTINUE
C
C RETURN
C END

```


00000010
00000020
00000030
00000040
00000050
00000060
00000070

THE FOLLOWING PLOT SUBROUTINES CODED BY DR. R. C. WINFREY

SUBROUTINE DRUP(X, Y, K, N, XSIZE, YSIZE, NPT, LABEL, TITLE, XNAME, YNAME)
DIMENSION TITLE(1), XNAME(1), YNAME(20, 1), LABEL(1)
DIMENSION X(1), Y(N, 1)
CALL DRUBL(X, Y, K, N, XSIZE, YSIZE, NPT, LABEL, TITLE, XNAME, YNAME)

RETURN
END

00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190

SUBROUTINE DRUPB(X, Y, K, N, XSIZE, YSIZE, NPT, LABEL, TITLE, XNAME, YNAME)
DIMENSION TITLE(1), XNAME(1), YNAME(20, 1), LABEL(1)
DIMENSION X(1), Y(N, 1)
CALL DRUB(X, Y, K, N, XSIZE, YSIZE, NPT, LABEL, TITLE, XNAME, YNAME)

RETURN
END

00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310

SUBROUTINE DRUBL(X, Y, K, N, XSIZE, YSIZE, NPT, LABEL, TITLE, XNAME, YNAME)
DIMENSION TITLE(1), XNAME(1), YNAME(20, 1), LABEL(1)
DIMENSION X(1), Y(N, 1), YMIN(20), DY(20)

PAPER=11.
CALL PLOTS
CALL PLOT(-11, 0, -3)
CALL SYMBOL(0, 0, 14, LABEL, 0, 80)
CALL PLOT(0, 4, -3)
NSIGN=1
IF(NPT.LT.0) NSIGN=-1
NPA=NPT*NSIGN

00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420

C
C
C
C
C
C

C
C
C
C
C

C
C
C
C
C


```

XST=FLOAT(K)*.65
XSTP=XST+(PAPER-XSIZE)*.5
IF(XSIZE.GT.8.01) XSTP=XST+1.
XAXIS=XSIZE-XST
CALL SCALE(X,N,1,XAXIS,0.,XMIN,DX)
DO 100 JA=1,K
CALL SCALE(Y(1,JA),N,1,YSIZE,0.,YMIN(JA),DY(JA))
CONTINUE
100
DO 130 JA=1,N
X(JA)=X(JA)+XSTP
YMAX=YMIN(1)+DY(1)*YSIZE
YS=ABS(YMIN(1))/DY(1)
IF(YMAX.LT.0.) YS=0.
IF(YMIN(1).GT.0.) YS=0.
IF(XSIZE.GT.8.01) GO TO 200
CALL CENT(XNAME,NTT)
NLT=-NTT
CALL EXISA(XSTP,YS,XNAME,NLT,XAXIS,0.,XMIN,DX)
CALL EXISA(XSTP,YS,XNAME,NLT,XAXIS,0.,XMIN,DX)
JB=1
DO 160 JA=1,K
JB=JB+1
IF(JB.EQ.8) JB=2
YST=XSTP-.65*FLOAT(JA-1)
CALL CENT(YNAME(1,JA),NTT)
CALL EXISA(YST,0.,YNAME(1,JA),NTT,YSIZE,90.,YMIN(JA),DY(JA))
JC=NPT
IF(NPA.EQ.10) JC=JB*NSIGN
CALL LIND(X,Y(1,JA),N,1,JC)
CALL LIND(X,Y(1,JA),N,1,JC)
CALL LIND(X,Y(1,JA),N,1,JC)
CONTINUE
160
YL=-.8
CALL CENT(TITLE,NTT)
XL=PAPER*.5-FLOAT(NTT/2)*.18
CALL SYMBOL(XL, YL,.21,TITLE,0.,NTT)
YAX=YSIZE+3.
GO TO 300
200
XFAR=(PAPER+YSIZE)*.5
YS=XFAP-YS
CALL CENT(XNAME,NTT)
NLT=-NTT
CALL EXISA(YS,XSTP,XNAME,NLT,XAXIS,90.,XMIN,DX)
JB=1
DO 260 JA=1,K
JB=JB+1
IF(JB.EQ.8) JB=2
YST=XSTP-.65*FLOAT(JA-1)

```

```

00000430
00000440
00000450
00000460
00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630
00000640
00000650
00000660
00000670
00000680
00000690
00000700
00000710
00000720
00000730
00000740
00000750
00000760
00000770
00000780
00000790
00000800
00000810
00000820
00000830
00000840
00000850
00000860
00000870
00000880
00000890
00000900

```



```

00000910
00000920
00000930
00000940
00000950
00000960
00000970
00000980
00000990
00001000
00001010
00001020
00001030
00001040
00001050
00001060
00001070
00001080
00001090
00001100
00001110
00001120
00001130
00001140
00001150
00001160
00001170
00001180
00001190
00001200
00001210
00001220

CALL CENT(YNAME(1,JA),NTT)
CALL EXISA(XFAR,YST,YNAME(1,JA),NTT,YSIZE,180.,YMIN(JA),DY(JA))
DO 230 JC=1,N
Y(JC,JA)=XFAR-Y(JC,JA)
JC=NPT
IF(NPA.EQ.10) JC=JB*NSIGN
CALL LIND(Y(1,JA),X,N,1,JC)
CALL LIND(Y(1,JA),X,N,1,JC)
CALL LIND(Y(1,JA),X,N,1,JC)
DO 260 JC=1,N
Y(JC,JA)=XFAR-Y(JC,JA)
CONTINUE
XL=XFAR+.8
IF(YS.LT.01) XL=XL+.55
CALL CENT(TITLE,NTT)
YL=XSIZE*.5+1.-FLOAT(NTT/2)*.18
CALL SYMBOL(XL,YL,.21,TITLE,90.,NTT)
YAX=XSIZE+3.
CALL PLOT(0.,YAX,-3)
CALL SYMBOL(0.,0.,.14,LABEL,0.,80)
CALL PLOT(1.25,4.,-3)
CALL PLOTE
WRITE(6,400)(TITLE(J),J=1,20)
FORMAT(///20X,75HA PLOT HAS BEEN DRAWN BY SUBROUTINE DRUP CONTAINING THE FOLLOWING TITLE - ,///20X,20A4,///)
C RESTORE ORIGINAL VALUES OF X AND Y
DO 420 JA=1,N
X(JA)=(X(JA)-XSTP)*DX+XMIN
DO 420 JB=1,K
Y(JA,JB)=Y(JA,JB)*DY+(JB)+YMIN(JB)
END

```

```

00001230
00001240
00001250
00001260
00001270
00001280
00001290
00001300
00001310
00001320
00001330
00001340
00001350
00001360

SUBROUTINE EXISA(X,Y,BCD,NC,SIZE,THETA,YMIN,DY)
DIMENSION BCD(1)
DATA EPS/1.E-6/
ZING=1.
IF(NC) 1,2,2
ZING = -1.
NAC=IABS(NC)
TH=THETA*.0174533
N=SIZE+.5
CTH=COS(TH)
STH=SIN(TH)
TN=N
XB=X
YB=Y

```


00001370
 00001380
 00001390
 00001400
 00001410
 00001420
 00001430
 00001440
 00001450
 00001460
 00001470
 00001480
 00001490
 00001500
 00001510
 00001520
 00001530
 00001540
 00001550
 00001560
 00001570
 00001580
 00001590
 00001600
 00001610
 00001620
 00001630
 00001640
 00001650
 00001660
 00001670
 00001680
 00001690
 00001700
 00001710
 00001720
 00001730
 00001740
 00001750
 00001760
 00001770
 00001780
 00001790
 00001800
 00001810
 00001820
 00001830
 00001840

C DRAW AXIS WITH TIC MARKS

```

XA=X-.06*ZING*STH
YA=Y+.06*ZING*CTH
CALL PLOT(XA,YA,3)
DO 20 I=1,N
CALL PLOT(XB,YB,2)
XC=XB+CTH
YC=YB+STH
CALL PLOT(XC,YC,2)
XA=XA+CTH
YA=YA+STH
CALL PLOT(XA,YA,2)
XB=XC
YB=YC
  
```

20 NUMBER THE TIC MARKS

```

IF(DY) 22,31,22
CHAR=ABS(YMIN)
ABSV=ABS(YMIN+DY)
IF(ABSV-CHAR) 5,6,6
ABSV=CHAR
POW=0.
CALL ANN(ABSV,BBSV,KPOW)
POW=KPOW
ADY=DY*10.**(-POW)
ABSV=YMIN*10.**(-POW)+TN*ADY
XA=XB- (.18*ZING-.07)*STH-.1714*CTH
YA=YB+ (.18*ZING-.07)*CTH-.1714*STH
N=N+1
  
```

DO 30 I=1,N

```

IF(ABS(ABSV)-EPS) 101,101,100
CALL NUMBER(XA,YA,.14,ABSV,THETA,2)
ABSV=ABSV-ADY
XA=XA-CTH
YA=YA-STH
  
```

30 TNC=NAC+7

```

WRITE AXIS LABEL
XA=X+(SIZE*.5-.06*TNC)*CTH-(-.14+ZING*.46)*STH
YA=Y+(SIZE*.5-.06*TNC)*STH+(-.14+ZING*.46)*CTH
CALL SYMBOL(XA,YA,.14,BCD,THETA,NAC)
XA=XA+((TNC-6.)*.12)*CTH
YA=YA+((TNC-6.)*.12)*STH
  
```

801 IF(DY) 801,50,801

801 IF(POW) 35,50,35

35 CALL SYMBOL(XA,YA,.14,7H(X10),THETA,7)

XA=XA+.48*CTH-.14*STH

YA=YA+.48*STH+.14*CTH

IF(POW) 40,50,40

40 CALL NUMBER(XA,YA,.14,POW,THETA,-1)

00001850
00001860

00001870
00001880
00001890
00001900
00001910
00001920
00001930
00001940
00001950
00001960
00001970
00001980
00001990
00002000
00002010
00002020
00002030
00002040
00002050
00002060
00002070
00002080
00002090
00002100
00002110
00002120
00002130
00002140
00002150
00002160
00002170
00002180
00002190
00002200
00002210
00002220
00002230
00002240
00002250
00002260
00002270
00002280
00002290
00002300

50 RETURN
END

```

SUBROUTINE LIND(X,Y,N,K,L)
DIMENSION X(1),Y(1)
DATA IPLUS,ISQ,IDM,ISTAR,ICROSS,IDOT
C /,+, ZCA000000, ZCB000000, '#', ZCC000000, '.'/
I3=3
KK=K
C...DATA IN EVERY KTH CELL
IF(K) 5,55,6
55 WRITE(6,1000)
1000 FORMAT(5X,'PLT CODE 01 K=0 IN ARG LIST FOR SUBROUT. LINE')
5 I3=2
KK=-KK
6 NP=N*KK
C...TEST BRANCH VALUE
LL=IABS(L)
IF(LL)65,65,64
64 IF(LL-7) 7,7,65
65 LL=1
7 DO 10 I=1,NP,KK
CALL PLOT(X(I),Y(I),I3)
C...PEN CONDITION BETWEEN POINTS
IF(L)71,71,73
71 I3=3
73 I3=2
C...BRANCH TO SPECIFIED MARKER
199 GO TO(10,200,300,400,500,600,700),LL
200 IBCD=IPLUS
300 GO TO 9
300 IBCD=ISQ
400 GO TO 91
400 IBCD=IDM
500 GO TO 9
500 IBCD=ISTAR
600 GO TO 9
600 IBCD=ICROSS
700 GO TO 91
700 IBCD=IDOT
XP=X(I)-.03
YP=Y(I)-.03
9 GO TO 94
XP=X(I)-.04
YP=Y(I)-.06

```


00002310
 00002320
 00002330
 00002340
 00002350
 00002360
 00002370
 00002380
 00002390

91 GO TO 94
 XP=X(I)-.03
 YP=Y(I)-.05
 GO TO 94
 94 CALL SYMBOL(XP,YP;.14,IBCD,0.,1)
 95 CALL PLOT(X(I),Y(I),3)
 10 CONTINUE
 11 RETURN
 END

00002400
 00002410
 00002420
 00002430
 00002440
 00002450
 00002460
 00002470
 00002480
 00002490
 00002500
 00002510
 00002520
 00002530
 00002540
 00002550
 00002560
 00002570
 00002580
 00002590
 00002600
 00002610
 00002620
 00002630
 00002640
 00002650

SUBROUTINE CENT(BCD,NTT)
 DIMENSION BCD(1)
 DATA BLNK,' /
 NTT=0
 DO 10 JA=1,20
 JB=JA
 IF(BCD(JA).NE.BLNK) GO TO 20
 CONTINUE
 RETURN
 10 JA=20
 21 IF(BCD(JA).NE.BLNK) GO TO 30
 JA=JA-1
 IF(JA.LT.0) RETURN
 GO TO 21
 30 JC=JA
 40 NTTA=JC-JB+1
 NTT=4*NTTA
 DO 50 JA=1,NTTA
 JD=JA+JB-1
 BCD(JA)=BCD(JD)
 NTT=NTTA+1
 IF(NTT.GT.20) RETURN
 DO 60 JA=NTT,20
 BCD(JA)=BLNK
 RETURN
 END

00002660
 00002670
 00002680
 00002690
 00002700
 00002710
 00002720
 00002730
 00002740

SUBROUTINE DRUA(X,Y,K,N,XSIZE,YSIZE,NPT)
 DIMENSION X(1),Y(N,1),YMIN(20),DY(20)
 DIMENSION TITLE(20),XNAME(20),YNAME(20,20)
 PAPER=11.
 50 FORMAT(20A4)
 READ(5,50,END=400) (TITLE(JA),JA=1,20)
 READ(5,50,END=400) (XNAME(JA),JA=1,20)
 DO 60 JA=1,K
 READ(5,50,END=400) (YNAME(JB,JA),JB=1,20)


```

60 CONT INUE
CALL PLOT(-11.,0.,-3)
CALL SYMBOL(0.,0.,.21,'BOX 81 WINFREY',0.,15)
CALL PLOT(0.,4.,-3)
NSIGN=1
IF(NPT.LT.0) NSIGN=-1
NPA=NPT*NSIGN
XST=FLOAT(K)*.5
XSTP=XST+(PAPER-XSIZE)*.5
IF(XSIZE.GT.8.01) XSTP=XST+1.
XAXIS=XSIZE-XST
CALL SCALE(X,N,1,XAXIS,0.,XMIN,DX)
DO 100 JA=1,K
CALL SCALE(Y(1,JA),N,1,YSIZE,0.,YMIN(JA),DY(JA))
CONT INUE
DO 130 JA=1,N
X(JA)=X(JA)+XSTP
YMAX=YMIN(1)+DY(1)*YSIZE
YS=ABS(YMIN(1))/DY(1)
IF(YMAX.LT.0.) YS=0.
IF(YMIN(1).GT.0.) YS=0.
IF(XSIZE.GT.8.01) GO TO 200
CALL CENT(XNAME,NTT)
NLT=-NTT
CALL EXIS(XSTP,YS,XNAME,NLT,XAXIS,0.,XMIN,DX)
JB=1
DO 160 JA=1,K
JB=JB+1
IF(JB.EQ.8) JB=2
YST=XSTP-.5*FLOAT(JA-1)
CALL CENT(YNAME(1,JA),NTT)
CALL EXIS(YST,0.,YNAME(1,JA),NTT,YSIZE,90.,YMIN(JA),DY(JA))
JC=NPT
IF(NPA.EQ.10) JC=JB*NSIGN
CALL LINE(X,Y(1,JA),N,1,JC)
CONT INUE
YL=-.8
CALL CENT(TITLE,NTT)
XL=PAPER*.5-FLOAT(NTT/2)*.12
CALL SYMBOL(XL, YL,.14,TITLE,0.,NTT)
YAX=YSIZE+3.
GO TO 300
200 XFAR=(PAPER+YSIZE)*.5
YS=XFAR-YS
CALL CENT(XNAME,NTT)
NLT=-NTT
CALL EXIS(YS,XSTP,XNAME,NLT,XAXIS,90.,XMIN,DX)

```

```

00002750
00002760
00002770
00002780
00002790
00002800
00002810
00002820
00002830
00002840
00002850
00002860
00002870
00002880
00002890
00002900
00002910
00002920
00002930
00002940
00002950
00002960
00002970
00002980
00002990
00003000
00003010
00003020
00003030
00003040
00003050
00003060
00003070
00003080
00003090
00003100
00003110
00003120
00003130
00003140
00003150
00003160
00003170
00003180
00003190
00003200
00003210
00003220

```



```

00003230
00003240
00003250
00003260
00003270
00003280
00003290
00003300
00003310
00003320
00003330
00003340
00003350
00003360
00003370
00003380
00003390
00003400
00003410
00003420
00003430
00003440
00003450
00003460
00003470
00003480
00003490
00003500
00003510
00003520
00003530
00003540
00003550
00003560
00003570
00003580
00003590
00003600
00003610
00003620
00003630

00003640
00003650
00003660
00003670
00003680

JB=1
DO 260 JA=1,K
JB=JB+1
IF(JB.EQ.8) JB=2
YST=XSTP-.5*FLOAT(JA-1)
CALL CENT(YNAME(1,JA),NTT)
CALL EXIS(XFAR,YST,YNAME(1,JA),NTT,YSIZE,180.,YMIN(JA),DY(JA))
DO 230 JC=1,N
YC(JC,JA)=XFAR-Y(JC,JA)
JC=NPT
IF(NPA.EQ.10) JC=JB*NSIGN
CALL LINE(Y(1,JA),X,N,1,JC)
DO 260 JC=1,N
Y(JC,JA)=XFAR-Y(JC,JA)
CONTINUE
XL=XFAR+.8
CALL CENT(TITLE,NTT)
YL=XSIZE*.5+1.-FLOAT(NTT/2)*.12
CALL SYMBOL(XL,YL,.14,TITLE,90.,NTT)
YAX=XSIZE+3.
CALL PLOT(0.,YAX,-3)
CALL SYMBOL(0.,0.,.21,'BOX 81 WINFREY',0.,15)
CALL PLOT(1.25,4.,-3)
CALL PLOT
WRITE(6,380)(TITLE(J),J=1,20)
FORMAT(///20X,75HA PLOT HAS BEEN DRAWN BY SUBROUTINE DRUA CONTAINING THE FOLLOWING TITLE -
C RESTORE ORIGINAL VALUES TO X AND Y
DO 390 JA=1,N
X(JA)=(X(JA)-XSTP)*DX+XMIN
DO 390 JB=1,K
Y(JA,JB)=Y(JA,JB)*DY(JB)+YMIN(JB)
RETURN
JA=2+K
WRITE(6,450) JA
FORMAT(1H1///20X,43HNOT ENOUGH DATA CARDS FOR SUBROUTINE DRUA.
1 ///122,52H CARDS ARE READ EACH TIME SUBROUTINE DRUA IS CALLED.
2 ///20X54HRETURNED TO CALLING ROUTINE WITHOUT STARTING A PLOT.
3 //1H1///)
RETURN
END
SUBROUTINE DRUB(X,Y,K,N,XSIZE,YSIZE,NPT,LABEL,TITLE,XNAME,YNAME)
DIMENSION X(1),Y(N,1),YMIN(20),DY(20)
DIMENSION TITLE(1),XNAME(1),YNAME(20,1),LABEL(1)
PAPER=11.
CALL PLOTS

```



```

00003690
00003700
00003710
00003720
00003730
00003740
00003750
00003760
00003770
00003780
00003790
00003800
00003810
00003820
00003830
00003840
00003850
00003860
00003870
00003880
00003890
00003900
00003910
00003920
00003930
00003940
00003950
00003960
00003970
00003980
00003990
00004000
00004010
00004020
00004030
00004040
00004050
00004060
00004070
00004080
00004090
00004100
00004110
00004120
00004130
00004140
00004150
00004160

CALL PLOT(-11.,0.,-3)
CALL SYMBOL(0.,0.,14,LABEL,0.,80)
CALL PLOT(0.,4.,-3)
NSIGN=1
IF(NPT.LT.0) NSIGN=-1
NPA=NPT*NSIGN
XST=FLOAT(K)*.5
IF(XSIZE.GT.8.01) XSTP=XST+1.
XAXIS=XSIZE-XST
CALL SCALE(X,N,1,XAXIS,0.,XMIN,DX)
DO 100 JA=1,K
CALL SCALE(Y(1,JA),N,1,YSIZE,0.,YMIN(JA),DY(JA))
CONTINUE
DO 130 JA=1,N
X(JA)=X(JA)+XSTP
YMAX=YMIN(1)+DY(1)*YSIZE
YS=ABS(YMIN(1))/DY(1)
IF(YMAX.LT.0.) YS=0.
IF(YMIN(1).GT.0.) YS=0.
IF(XSIZE.GT.8.01) GO TO 200
CALL CENT(XNAME,NTT)
NLT=-NTT
CALL EXIS(XSTP,YS,XNAME,NLT,XAXIS,0.,XMIN,DX)
JB=1
DO 160 JA=1,K
JB=JB+1
JB=2
IF(JB.EQ.8) JB=2
YF=XSTP-.5*FLOAT(JA-1)
CALL CENT(YNAME(1,JA),NTT)
CALL EXIS(YST,0.,YNAME(1,JA),NTT,YSIZE,90.,YMIN(JA),DY(JA))
JC=NPT
IF(NPA.EQ.10) JC=JB*NSIGN
CALL LINE(X,Y(1,JA),N,1,JC)
CONTINUE
YL=-.8
CALL CENT(TITLE,NTT)
XL=PAPER*.5-FLOAT(NTT/2)*.12
CALL SYMBOL(XL, YL,.14,TITLE,0.,NTT)
YAX=YSIZE+3.
GO TO 300
200 XFAR=(PAPER+YSIZE)*.5
YS=XFAR-YS
CALL CENT(XNAME,NTT)
NLT=-NTT
CALL EXIS(YS,XSTP,XNAME,NLT,XAXIS,90.,XMIN,DX)
JB=1
DO 260 JA=1,K

```



```

00004170
00004180
00004190
00004200
00004210
00004220
00004230
00004240
00004250
00004260
00004270
00004280
00004290
00004300
00004310
00004320
00004330
00004340
00004350
00004360
00004370
00004380
00004390
00004400
00004410
00004420
00004430
00004440
00004450
00004460
00004470
00004480
00004490

JB=JB+1
IF(JB.EQ.8) JB=2
YST=XSTP-.50*FLOAT(JA-1)
CALL CENT(YNAME(1,JA),NTT)
CALL EXIS(XFAR,YST,YNAME(1,JA),NTT,YSIZE,180.,YMIN(JA),DY(JA))
DO 230 JC=1,N
Y(JC,JA)=XFAR-Y(JC,JA)
230 JC=NPT
IF(NPA.EQ.10) JC=JB*NSIGN
CALL LINE(Y(1,JA),X,N,1,JC)
DO 260 JC=1,N
Y(JC,JA)=XFAR-Y(JC,JA)
260 CONTINUE
XL=XFAR+.8
IF(YS.LT.01) XL=XL+.55
CALL CENT(TITLE,NTT)
YL=XSIZE*.5+1.-FLOAT(NTT/2)*.12
CALL SYMBOL(XL,YL,.14,TITLE,90.,NTT)
YAX=XSIZE+3
CALL PLOT(0.,YAX,-3)
CALL SYMPO(0.,0.,.14,LABEL,0.,80)
CALL PLOT(1.25,4.,-3)
CALL PLOTE
WRITE(6,400)(TITLE(J),J=1,20)
400 FORMAT(///20X,75HA PLOT HAS BEEN DRAWN BY SUBROUTINE DRUB CONTAINING THE FOLLOWING TITLE -
C RESTORE ORIGINAL VALUES OF X AND Y,///20X,20A4,////)
DO 420 JA=1,N
X(JA)=(X(JA)-XSTP)*DX+XMIN
DO 420 JB=1,K
Y(JA,JB)=Y(JA,JB)*DY(JB)+YMIN(JB)
420 RETURN
END

```

```

SUBROUTINE DRUC(X,Y,K,N,XSIZE,YSIZE,NPT,LABEL,TITLE,XNAME,YNAME)
DIMENSION TITLE(1),XNAME(1),YNAME(1),LABEL(1),X(N,1),Y(N,1)
PAPER=11.
CALL PLOTS
CALL PLOT(-11.,0.,-3)
CALL SYMBOL(0.,0.,.14,LABEL,0.,80)
CALL PLOT(0.,4.,-3)
NSIGN=1
IF(NPT.LT.0) NSIGN=-1
NPA=NPT*NSIGN
XST=1.
XSTP=(PAPER-XSIZE)*.5
IF(XSIZE.GT.8.01) XSTP=XST+1.
00004500
00004510
00004520
00004530
00004540
00004550
00004560
00004570
00004580
00004590
00004600
00004610
00004620

```



```

XAXIS=XSIZE-XST
NK=N*K
CALL SCALE(X,NK,1,XAXIS,0.,XMIN,DX)
CALL SCALE(Y,NK,1,YSIZE,0.,YMIN,DY)
DO 130 JB=1,K
DO 130 JA=1,N
X(JA,JB)=X(JA,JB)+XSTP
YMAX=YMIN+DY*YSIZE
YS=ABS(YMIN/DY)
IF(YMAX.LT.0.) YS=0.
IF(YMIN.GT.0.) YS=0.
IF(XSIZE.GT.8.01) GO TO 200
CALL CENT(XNAME,NTT)
NLT=-NTT
CALL EXIS(XSTP,YS,XNAME,NLT,XAXIS,0.,XMIN,DX)
CALL CENT(YNAME,NTT)
CALL EXIS(XSTP,0.,YNAME,NTT,YSIZE,90.,YMIN,DY)
JB=1
DO 160 JA=1,K
JB=JB+1
IF(JB.EQ.8) JB=2
JC=NPT
IF(NPA.EQ.10) JC=JB*NSIGN
CALL LINE(X(1,JA),Y(1,JA),N,1,JC)
CONTINUE
YL=-.8
CALL CENT(TITLE,NTT)
XL=PAPER*.5-FLOAT(NTT/2)*.12
CALL SYMBOL(XL, YL,.14,TITLE,0.,NTT)
YAX=YSIZE+3.
GO TO 300
XFAR=(PAPER+YSIZE)*.5
YS=XFAR-YS
CALL CENT(XNAME,NTT)
NLT=-NTT
CALL EXIS(YS,XSTP,XNAME,NLT,XAXIS,90.,XMIN,DX)
CALL CENT(YNAME,NTT)
CALL EXIS(XFAR,XSTP,YNAME, NLT,YSIZE,180.,YMIN,DY)
JB=1
DO 260 JA=1,K
JB=JB+1
IF(JB.EQ.8) JB=2
DO 230 JC=1,N
Y(JC,JA)=XFAR-Y(JC,JA)
JC=NPT
IF(NPA.EQ.10) JC=JB*NSIGN
CALL LINE(Y(1,JA),X(1,JA),N,1,JC)
DO 260 JC=1,N

```

```

00004630
00004640
00004650
00004660
00004670
00004680
00004690
00004700
00004710
00004720
00004730
00004740
00004750
00004750
00004770
00004780
00004790
00004800
00004810
00004820
00004830
00004840
00004850
00004860
00004870
00004880
00004890
00004900
00004910
00004920
00004930
00004940
00004950
00004960
00004970
00004980
00004990
00005000
00005010
00005020
00005030
00005040
00005050
00005060
00005070
00005080
00005090
00005100

```



```

Y(JC,JA)=XFAR-Y(JC,JA)
CONTINUE
XL=XFAR+.8
CALL CENT(TITLE,NTT)
YL=X SIZE*.5+1.-FLOAT(NTT/2)*.12
CALL SYMBOL(XL,YL,.14,TITLE,90.,NTT)
YAX=X SIZE+3.
CALL PLOT(0.,YAX,-3)
CALL SYMBOL(0.,0.,.14,LABEL,0.,80)
CALL PLOT(1.25,4.,-3)
CALL PLOTE
WRITE(6,400)(TITLE(J),J=1,20)
FORMAT(///20X,75HA PLOT HAS BEEN DRAWN BY SUBROUTINE DRUC CONTAINING THE FOLLOWING TITLE - ,///20X,20A4,////)
1 RESTORE ORIGINAL VALUES TO X AND Y
DO 420 JA=1,N
DO 420 JB=1,K
X(JA,JB)=(X(JA,JB)-XSTP)*DX+XMIN
Y(JA,JB)=Y(JA,JB)*DY+YMIN
420 RETURN
END

```

```

00005110
00005120
00005130
00005140
00005150
00005160
00005170
00005180
00005190
00005200
00005210
00005220
00005230
00005240
00005250
00005260
00005270
00005280
00005290
00005300
00005310

```

```

SUBROUTINE EXIS (X,Y,BCD,NC,SIZE,THETA,THETA,YMIN,DY)

```

```

DIMENSION BCD(1)
DATA EPS/1.E-6/
ZING=1.
IF(NC) 1,2,2
ZING=-1.
TH=THETA*.0174533
N=SIZE+.5
CTH=COS(TH)
STH=SIN(TH)
TN=N
XB=X
YB=Y
XA=X-.1*ZING*STH
YA=Y+.1*ZING*CTH
CALL PLOT(XA,YA,3)
DO 20 I=1,N
CALL PLOT(XB,YB,2)
XC=XB+CTH
YC=YB+STH
CALL PLOT(XC,YC,2)
XA=XA+CTH
YA=YA+STH
CALL PLOT(XA,YA,2)

```

```

00005320
00005330
00005340
00005350
00005360
00005370
00005380
00005390
00005400
00005410
00005420
00005430
00005440
00005450
00005460
00005470
00005480
00005490
00005500
00005510
00005520
00005530
00005540
00005550
00005560

```



```

20 XB=XC
20 YB=YC
22 IF(DY) 22,31,22
22 CHAR=ABS(YMIN)
22 ABSV=ABS(YMIN+DY)
5 IF(ABSV-CHAR) 5,6,6
6 ABSV=CHAR
6 POW=0.
CALL ANN(ABSV,BBSV,KPOW)
POW=KPOW
93 ADY=DY*10.**(-POW)
ABSV=YMIN*10.**(-POW)+TN*ADY
XA=XB- (.2*ZING-.05)*STH-.0857*CTH
YA=YB+ (.2*ZING-.05)*CTH-.0857*STH
N=N+1
DO 30 I=1,N
IF(ABS(ABSV)-EPS) 101,101,100
CALL NUMBER(XA,YA,.1,ABSV,THETA,2)
100 ABSV=ABSV-ADY
101 XA=XA-CTH
30 YA=YA-STH
31 TNC=NAC+7
XA=X+(SIZE*.5-.03*TNC)*CTH-(-.07+ZING*.36)*STH
YA=Y+(SIZE*.5-.03*TNC)*STH+(-.07+ZING*.36)*CTH
CALL SYMBOL(XA,YA,.07,BCD,THETA,NAC)
XA=XA+((TNC-6.)*.06)*CTH
YA=YA+((TNC-6.)*.06)*STH
802 IF(DY) 801,50,801
801 IF(POW) 35,50,35
35 CALL SYMBOL(XA,YA,.07,7H(X10),THETA,7)
XA=XA+.24*CTH-.07*STH
YA=YA+.24*STH+.07*CTH
IF(POW) 40,50,40
40 CALL NUMBER(XA,YA,.1,POW,THETA,-1)
50 RETURN
END

```

```

SUBROUTINE DRUQ(X,Y,K,N,XSIZE,YSIZE,NPT,LABEL,TITLE,XNAME,YNAME)
DIMENSION TITLE(1),XNAME(1),YNAME(1),LABEL(1),X(N,1),Y(N,1)
CALL DRUCL(X,Y,K,N,XSIZE,YSIZE,NPT,LABEL,TITLE,XNAME,YNAME)
END
00005570
00005580
00005590
00005600
00005610
00005620
00005630
00005640
00005650
00005660
00005670
00005680
00005690
00005700
00005710
00005720
00005730
00005740
00005750
00005760
00005770
00005780
00005790
00005800
00005810
00005820
00005830
00005840
00005850
00005860
00005870
00005880
00005890
00005900
00005910
00005920

00005930
00005940
00005950
00005960
00005970

```


00005980
 00005990
 00006000
 00006010
 00006020
 00006030
 00006040
 00006050
 00006060
 00006070
 00006080
 00006090
 00006100
 00006110
 00006120
 00006130
 00006140
 00006150
 00006160
 00006170
 00006180
 00006190
 00006200
 00006210
 00006220
 00006230
 00006240
 00006250
 00006260
 00006270
 00006280
 00006290
 00006300
 00006310
 00006320
 00006330
 00006340
 00006350
 00006360
 00006370
 00006380
 00006390
 00006400
 00006410
 00006420
 00006430
 00006440
 00006450

```

SUBROUTINE DRUCL(X,Y,K,N,XSIZE,YSIZE,NPT,LABEL,TITLE,XNAME,YNAME)
DIMENSION TITLE(1),XNAME(1),YNAME(1),LABEL(1),X(N,1),Y(N,1)
PAPER=11.
CALL PLOTS
CALL PLOT(-11.,0.,-3)
CALL SYMBOL(0.,0.,.14,LABEL,0.,80)
CALL PLOT(0.,4.,-3)
NSIGN=1
IF(NPT.LT.0) NSIGN=-1
NPA=NPT*NSIGN
XST=1.
XSTP=XST+(PAPER-XSIZE)*.5
IF(XSIZE.GT.8.01) XSTP=XST+1.
XAXIS=XSIZE-XST
NK=N*K
CALL SCALE(X,NK,1,XAXIS,0.,XMIN,DX)
CALL SCALE(Y,NK,1,YSIZE,0.,YMIN,DY)
DO 130 JB=1,K
DO 130 JA=1,N
X(JA,JB)=X(JA,JB)+XSTP
YMAX=YMIN+DY*YSIZE
YS=ABS(YMIN/DY)
IF(YMAX.LT.0.) YS=0.
IF(YMIN.GT.0.) YS=0.
IF(XSIZE.GT.8.01) GO TO 200
CALL CENT(XNAME,NTT)
NLT=-NTT
CALL EXISA(XSTP,YS,XNAME,NLT,XAXIS,0.,XMIN,DX)
CALL CENT(YNAME,NTT)
CALL EXISA(XSTP,0.,YNAME,NTT,YSIZE,90.,YMIN,DY)
JB=1
DO 160 JA=1,K
JB=JB+1
IF(JB.EQ.8) JB=2
JC=NPT
IF(NPA.EQ.10) JC=JB*NSIGN
CALL LIND(X(1,JA),Y(1,JA),N,1,JC)
CALL LIND(X(1,JA),Y(1,JA),N,1,JC)
CALL LIND(X(1,JA),Y(1,JA),N,1,JC)
CONTINUE
YL=-.8
CALL CENT(TITLE,NTT)
XL=PAPER*.5-FLOAT(NTT/2)*.18
CALL SYMBOL(XL, YL,.21,TITLE,0.,NTT)
YAX=YSIZE+3.
GO TO 300
XFAR=(PAPER+YSIZE)*.5
YS=XFAR-YS
  
```

130

160

200


```

00006460
00006470
00006480
00006490
00006500
00006510
00006520
00006530
00006540
00006550
00006560
00006570
00006580
00006590
00006600
00006610
00006620
00006630
00006640
00006650
00006660
00006670
00006680
00006690
00006700
00006710
00006720
00006730
00006740
00006750
00006760
00006770
00006780
00006790
00006800
00006810
00006820
00006830

CALL CENT(XNAME,NTT)
NLT=-NTT
CALL EXISA(YS,XSTP,XNAME,NLT,XAXIS,90.,XMIN,DX)
CALL CENT(YNAME,NTT)
CALL EXISA(XFAR,XSTP,YNAME,NTT,YSIZE,180.,YMIN,DY)
JB=1
DO 260 JA=1,K
JB=JB+1
IF(JB.EQ.8) JB=2
DO 230 JC=1,N
Y(JC,JA)=XFAR-Y(JC,JA)
JC=NPT
IF(NPA.EQ.10) JC=JB*NSIGN
CALL LIND(Y(1,JA),X(1,JA),N,1,JC)
CALL LIND(Y(1,JA),X(1,JA),N,1,JC)
CALL LIND(Y(1,JA),X(1,JA),N,1,JC)
DO 260 JC=1,N
Y(JC,JA)=XFAR-Y(JC,JA)
CONTINUE
XL=XFAR+.8
CALL CENT(TITLE,NTT)
YL=XSIZE*.5+1.-FLOAT(NTT/2)*.18
CALL SYMBOL(XL,YL,.21,TITLE,90.,NTT)
YAX=XSIZE+3.
CALL PLOT(0.,YAX,-3)
CALL SYMBOL(0.,0.,14,LABEL,0.,80)
CALL PLOT(1.25,4.,-3)
CALL PLOTE
WRITE(6,400)(TITLE(J),J=1,20)
FORMAT(///20X,75HA PLOT HAS BEEN DRAWN BY SUBROUTINE DRUQ CONTAINING THE FOLLOWING TITLE -
1 RESTORE ORIGINAL VALUES TO X AND Y
DO 420 JA=1,N
DO 420 JB=1,K
X(JA,JB)=(X(JA,JB)-XSTP)*DX+XMIN
Y(JA,JB)=(Y(JA,JB)-DY)+YMIN
RETURN
END

```



```

C          SUBROUTINE DTPRIM( RECENT, KK, OLD, DT, DTP, NPRINT )
C          IMPLICIT REAL*8(A-H,O-Z)
C          INTEGER YES
C          DIMENSION DTP(1)
C          COMMON NUNIT, IPRINT
C          DATA YES/4HYES /, NO/4HNO /
C          5001_1 FORMAT(' ', 10X, 'FOR INTERACTION ', I1, ' RECENT = ', OLD - RECENT = ', D15.7)
C          SMALL = 1.0D-10
C          DIFF = OLD - RECENT
C          IF(DIFF .LT. SMALL) DIFF = SMALL
C          DTP(KK) = ( OLD/ DIFF) * DT
C          IF(NPRINT .EQ. NO) RETURN
C          WRITE(NUNIT,5001) KK, RECENT, OLD, DIFF
C          RETURN
C          END
00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290

```


00000470
 00000480
 00000490
 00000500
 00000510
 00000520
 00000530
 00000540
 00000550
 00000560
 00000570
 00000580
 00000590
 00000600
 00000610
 00000620
 00000630
 00000640
 00000650
 00000660
 00000670
 00000680
 00000690
 00000700
 00000710
 00000720
 00000730
 00000740
 00000750
 00000760
 00000770
 00000780
 00000790
 00000800
 00000810
 00000820
 00000830
 00000840
 00000850
 00000860
 00000870
 00000880
 00000890
 00000900
 00000910
 00000920
 00000930
 00000940

```

C MATRIX REDUCTION---FIND MATRIX M SUCH THAT M(T)BM=I
C THE ACTUAL COMPUTATION OF M(UPPER TRIANGULAR) IS ONLY
C DONE WHEN EIGENVECTORS ARE REQUESTED IT IS STORED AT X
C REDUCE B TO I(UPPER TRIANGULAR)
C REDUCE A TO AI=M(T)AM(LOWER TRIANGULAR)
C THE EIGENVALUES OF (A-ELAM*B)X=0 EQUAL THOSE OF (AI-ELAM*I)Y=0
C THE EIGENVECTORS HAVE THE RELATION X=YM
C
C DO 150 I=1,NN
C TBI=B(I,I)
C IF(TBI.GT.0.0) GO TO 100
C NERR=I
C RETURN
C TLI=1.0/TBI
C TAI=A(I,I)
C IM1=I-1
C IF(I.EQ.NN) GO TO 130
C IP1=I+1
C DO 125 J=IP1,NN
C TBJ=B(I,J)
C ELAM(J)=-TBJ*T1
C TLJ=ELAM(J)
C IF(IM1.EQ.0) GO TO 110
C DO 105 K=1,IM1
C A(J,K)=TLJ*A(I,K)+A(J,K)
C T2=TLJ*TAI+A(J,I)
C DO 115 K=IP1,J
C B(K,J)=B(K,J)+ELAM(K)*TBJ
C A(J,K)=ELAM(K)*T2+TLJ*A(K,I)+A(J,K)
C IF (NNX.EQ.0)GO TO 125
C IF NO VECTORS ARE REQUIRED M IS NOT COMPUTED
C
C DO 120 K=1,I
C X(K,J)=X(K,I)*TLJ+X(K,J)
C CONTINUE
C DO 127 J=IP1,NN
C A(J,I)=TLI*(ELAM(J)*TAI+A(J,I))
C IF(IM1.EQ.0) GO TO 140
C DO 135 K=1,IM1
C A(I,K)=TLI*A(I,K)
C IF (NNX.EQ.0)GO TO 150
C DO 145 K=1,I
C X(K,I)=X(K,I)*TLI
C A(I,I)=TAI
C
C 120
C 125
C 127
C 130
C 135
C 140
C 145
C 150
C

```



```

C C 200 SEQUENTIAL THRESHOLD JACOBI METHOD
C C 201 NMI=NN-1
C C 202 INITIALIZE THRESHOLD--FIND MAXIMUM OFF DIAGONAL ELEMENT
C C 203
C C 204 IS=6
C C 205 EMAX=0.0
C C 206 DO 215 J=1,NM1
C C 207 JP1=J+1
C C 208 DO 215 I=JP1,NN
C C 209 T1=DABS(A(I,J))
C C 210 IF(T1.GT.EMAX) EMAX=T1
C C 211 CONTINUE
C C 212 EMAX=EMAX/4.0
C C 213
C C 214
C C 215
C C 216
C C 217
C C 218
C C 219
C C 220
C C 221
C C 222
C C 223
C C 224
C C 225
C C 226
C C 227
C C 228
C C 229
C C 230
C C 231
C C 232
C C 233
C C 234
C C 235
C C 236
C C 237
C C 238
C C 239
C C 240
C C 241
C C 242
C C 243
C C 244
C C 245
C C 246
C C 247
C C 248
C C 249
C C 250
C C 251
C C 252
C C 253
C C 254
C C 255
C C 256
C C 257
C C 258
C C 259
C C 260
C C 261
C C 262
C C 263
C C 264
C C 265
C C 266
C C 267
C C 268
C C 269
C C 270
C C 271
C C 272
C C 273
C C 274
C C 275
C C 276
C C 277
C C 278
C C 279
C C 280
C C 281
C C 282
C C 283
C C 284
C C 285
C C 286
C C 287
C C 288
C C 289
C C 290
C C 291
C C 292
C C 293
C C 294
C C 295
C C 296
C C 297
C C 298
C C 299
C C 300
C C 301
C C 302
C C 303
C C 304
C C 305
C C 306
C C 307
C C 308
C C 309
C C 310
C C 311
C C 312
C C 313
C C 314
C C 315
C C 316
C C 317
C C 318
C C 319
C C 320
C C 321
C C 322
C C 323
C C 324
C C 325
C C 326
C C 327
C C 328
C C 329
C C 330
C C 331
C C 332
C C 333
C C 334
C C 335
C C 336
C C 337
C C 338
C C 339
C C 340
C C 341
C C 342
C C 343
C C 344
C C 345
C C 346
C C 347
C C 348
C C 349
C C 350
C C 351
C C 352
C C 353
C C 354
C C 355
C C 356
C C 357
C C 358
C C 359
C C 360
C C 361
C C 362
C C 363
C C 364
C C 365
C C 366
C C 367
C C 368
C C 369
C C 370
C C 371
C C 372
C C 373
C C 374
C C 375
C C 376
C C 377
C C 378
C C 379
C C 380
C C 381
C C 382
C C 383
C C 384
C C 385
C C 386
C C 387
C C 388
C C 389
C C 390
C C 391
C C 392
C C 393
C C 394
C C 395
C C 396
C C 397
C C 398
C C 399
C C 400
C C 401
C C 402
C C 403
C C 404
C C 405
C C 406
C C 407
C C 408
C C 409
C C 410
C C 411
C C 412
C C 413
C C 414
C C 415
C C 416
C C 417
C C 418
C C 419
C C 420
C C 421
C C 422
C C 423
C C 424
C C 425
C C 426
C C 427
C C 428
C C 429
C C 430
C C 431
C C 432
C C 433
C C 434
C C 435
C C 436
C C 437
C C 438
C C 439
C C 440
C C 441
C C 442
C C 443
C C 444
C C 445
C C 446
C C 447
C C 448
C C 449
C C 450
C C 451
C C 452
C C 453
C C 454
C C 455
C C 456
C C 457
C C 458
C C 459
C C 460
C C 461
C C 462
C C 463
C C 464
C C 465
C C 466
C C 467
C C 468
C C 469
C C 470
C C 471
C C 472
C C 473
C C 474
C C 475
C C 476
C C 477
C C 478
C C 479
C C 480
C C 481
C C 482
C C 483
C C 484
C C 485
C C 486
C C 487
C C 488
C C 489
C C 490
C C 491
C C 492
C C 493
C C 494
C C 495
C C 496
C C 497
C C 498
C C 499
C C 500

```



```

00001430
00001440
00001450
00001460
00001470
00001480
00001490
00001500
00001510
00001520
00001530
00001540
00001550
00001560
00001570
00001580
00001590
00001600
00001610
00001620
00001630
00001640
00001650
00001660
00001670
00001680
00001690
00001700
00001710
00001720
00001730
00001740
00001750
00001760
00001770
00001780
00001790
00001800
00001810
00001820
00001830
00001840
00001850
00001860
00001870
00001880
00001890

DO 325 K=JPI, IM1
T1=A(K, J)
T2=A(I, K)
A(K, J)=C*T1+S*T2
A(I, K)=C*T2-S*T1
DO 330 K=I, NN
T1=A(K, J)
T2=A(K, I)
A(K, J)=C*T1+S*T2
A(K, I)=-S*T1+C*T2
DO 335 K=1, J
T1=A(J, K)
T2=A(I, K)
A(J, K)=C*T1+S*T2
A(I, K)=-S*T1+C*T2
A(I, I)=-S*T3+C*A(I, I)
C
C
FORM EIGENVECTORS IF REQUESTED
IF (NNX.EQ.0)GO TO 350
DO 345 K=1, NN
T1=X(K, J)
T2=X(K, I)
X(K, J)=C*T1+S*T2
X(K, I)=-S*T1+C*T2
CONTINUE
C
C
TEST FOR CONVERGENCE AND MODIFY THRESHOLD
IS=IS-1
IF(IS.LE.0)GO TO 400
K=-(2**(5-IS))
EMAX=EMAX*(2.0**K)
GO TO 300
IF(IST.EQ.0)GO TO 500
IF(IS.GE.(-33))GO TO 300
C
C
METHOD HAS FAILED TO CONVERGE
NERR=-1
C
C
TRANSFER EIGENVALUES AND EXIT
DO 510 I=1, NN
ELAM(I)=A(I, I)
RETURN
END
500
510

```



```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460

```

```

C      SUBROUTINE FIXFOR(C, F)
C      IMPLICIT REAL*8(A-H,O-Z)
C      INTEGER C, YES
C      DIMENSION F(1)
C      COMMON NUNIT, IPRINT
C      DATA YES/4HYES /, NO/4HNO /
C
C      GO TO (10, 20, 30, 40, 50, 60, 70, 80), C
C
C      10 CONTINUE
C         F(1) = 0.0D0
C         F(2) = 0.0D0
C         F(7) = 0.0D0
C         F(8) = -F(2)
C         RETURN
C
C      20 CONTINUE
C         F(1) = 0.0D0
C         F(2) = 0.0D0
C         RETURN
C
C      30 CONTINUE
C         F(1) = 0.0D0
C         F(7) = 0.0D0
C         F(8) = -F(2)
C         RETURN
C
C      40 CONTINUE
C         F(1) = 0.0D0
C         F(8) = -F(2)
C         RETURN
C
C      50 CONTINUE
C         F(7) = 0.0D0
C         F(2) = 0.0D0
C         F(8) = 0.0D0
C         RETURN
C
C      60 CONTINUE
C         F(2) = 0.0D0

```


00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580

```
C      F(8) = 0.0D0  
      RETURN  
  
      70 CONTINUE  
      F(7) = 0.0D0  
      F(8) = -F(2)  
      RETURN  
  
      80 CONTINUE  
      F(8) = -F(2)  
      RETURN  
      END
```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180

```
C      SUBROUTINE ADJUST(I1,I2,QR,QRD,QRDD,Q,QD,QDD)  
C      IMPLICIT REAL*8(A-H,O-Z)  
C      INTEGER YES  
C      DIMENSION QR(1), QRD(1), QRDD(1), Q(1), QD(1), QDD(1)  
C      COMMON NUNIT, IPRINT  
C      DATA YES/4HYES /, NO/4HNO /  
C      QR (I1) = Q (I2)  
C      QRD (I1) = QD (I2)  
C      QRDD(I1) = QDD(I2)  
C      RETURN  
C      END
```



```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460

SUBROUTINE GPRINT(FPLOT, DEG, KTIME, YNAME, XSCALE, IP, NPRINT,
C      GRID, LINES, TITLE, LABEL)
C
C      INTEGER*2 GRID(LINES,1)
C      INTEGER YES
C
C      DIMENSION YNAME(20) , FPLOT(KTIME) , DEG(1) , XSCALE(1),
C      TITLE(1) , LABEL(1)
C      COMMON NUNIT, IPRINT
C      DATA YES/4HYES /, NO/4HNO /
C      FORMAT('0',5X,4(1PE15.7,5X) )
5000  FORMAT(' ',5X,15,5X,1P2E15.7)
5001  FORMAT(' ',5X,15,5X,1P2E15.7)
5026  FORMAT('1',18A4)
5027  FORMAT('+',72X,15A4)
5028  FORMAT('0',40X,18A4)
C      XMIN = DEG(1)
C      XMAX = DEG(KTIME)
C      YMIN = 1.0E40
C      YMAX=-1.E40
C
C      DO 10 K = 1, KTIME
C      IF(FPLOT(K) .GT. YMAX) YMAX = FPLOT(K)
C      IF(FPLOT(K) .LT. YMIN) YMIN = FPLOT(K)
10  CONTINUE
C      WRITE(NUNIT,5026) ( LABEL(JA), JA = 1,18)
C      WRITE(NUNIT,5027) ( TITLE(JA), JA = 1,15)
C      WRITE(NUNIT,5028) ( YNAME(JA), JA = 1,18)
C
C      IRYP1 = 80*IP + 1
C      CALL OLFPLOT(DEG,FPLOT,KTIME,XMAX,XMIN,YMAX,YMIN,IRYP1,GRID,XSCALE,
C      0, NPRINT, LINES)
C
C      IF(NPRINT .EQ. NO) RETURN
C
C      WRITE(6,5000) XMIN,XMAX,YMIN,YMAX
C
C      DO 51 JA = 1, KTIME
C      WRITE(6,5001) JA, DEG(JA), FPLOT(JA)
51  CONTINUE
C      RETURN
C      END

```



```

SUBROUTINE GRAPH(FPLOT, YNAME, XNAME, DEG1, DEG2, Y1, Y2, Y3, Y4,
1 YNAME1, YNAME2, PIN2$8, CLEAR, XSCALE,
2 NTIME, LABEL, TITLE, IDRAW, IPLOT, XSCALE,
3 IP, NPRINT, GRID, LINES)
C
INTEGER*2 GRID(LINES,1)
INTEGER YES, PIN2$8
C
REAL*8 CLEAR
REAL*4 LABEL
C
DIMENSION XNAME(1), DEG1(1), DEG2(NTIME,2), XSCALE(1), TITLE(1),
1 FPLOT(NTIME,1), YNAME(20,1), Y1(NTIME,1), Y2(NTIME,1),
2 Y3(NTIME,1), Y4(NTIME,1),
3 YNAME1(20,1), YNAME2(20,1), LABEL(1)
C
COMMON NUNIT, IPRINT
C
DATA YES/4HYES /, NO/4HNO /
C
DO 5 I = 1,10
IF(FPLOT(1,I) .EQ. 0.0) FPLOT(1,I) = 1.0E-10
CONTINUE
5
C
IF(IDRAW .EQ. NO) GO TO 2501
C
4201 CALL DRUB (DEG1, Y1, 2, NTIME, 10., 6., 1, LABEL, TITLE, XNAME, YNAME1)
4202 CALL DRUB (DEG1, Y2, 2, NTIME, 10., 6., 1, LABEL, TITLE, XNAME, YNAME2)
4203 CALL DRUB (DEG2, Y3, 2, NTIME, 10., 6., 1, LABEL, TITLE, XNAME, YNAME(1, 12))
4203 CALL DRUB (DEG1, FPLOT(1, 5), 1, NTIME, 10., 6., 1, LABEL, TITLE, XNAME,
1 YNAME(1, 5))
4204 CALL DRUB (DEG1, FPLOT(1, 6), 1, NTIME, 10., 6., 1, LABEL, TITLE, XNAME,
1 YNAME(1, 6))
4205 CALL DRUB (DEG1, FPLOT(1, 7), 1, NTIME, 10., 6., 1, LABEL, TITLE, XNAME,
1 YNAME(1, 7))
C
IF(PIN2$8 .EQ. YES) GO TO 4207
C
4206 CALL DRUB (DEG1, FPLOT(1, 9), 1, NTIME, 10., 6., 1, LABEL, TITLE, XNAME,
1 YNAME(1, 9))
C
4207 IF(CLEAR .LE. -10.0) GO TO 2501
C
4208 CALL DRUB (DEG1, FPLOT(1, 10), 1, NTIME, 10., 6., 1, LABEL, TITLE, XNAME,
1 YNAME(1, 10))

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460

```



```

00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630
00000640
00000650
00000660
00000670
00000680
00000690
00000700
00000710
00000720
00000730
00000740
00000750
00000760
00000770
00000780
00000790
00000800
00000810
00000820
00000830
00000840
00000850
00000860
00000870
00000880
00000890
00000900
00000910
00000920
00000930

4209 CALL DRUPB(DEG1,FPLOT(1,11),1,NTIME,10.,6.,1,LABEL,TITLE,XNAME,
1 YNAME(1,11) )
C CALL DRUC (DEG2,Y4,2,NTIME,10.,6.,1,LABEL,TITLE,XNAME,YNAME(1,13))
C
C
C 2501 CONTINUE
C
C IF(I PLOT .EQ. NO) GO TO 4199
C
C IF(PIN2$8 .EQ. YES) GO TO 4102
4101 CALL GPRINT(FPLOT(1, 1), DEG1, NTIME, YNAME(1,
1 NPRINT, GRID, LINES, TITLE, LABEL)
C
C 4102 CALL GPRINT(FPLOT(1, 2), DEG1, NTIME, YNAME(1,
1 NPRINT, GRID, LINES, TITLE, LABEL)
C 4103 CALL GPRINT(FPLOT(1, 3), DEG1, NTIME, YNAME(1,
1 NPRINT, GRID, LINES, TITLE, LABEL)
C 4104 CALL GPRINT(FPLOT(1, 4), DEG1, NTIME, YNAME(1,
1 NPRINT, GRID, LINES, TITLE, LABEL)
C 4105 CALL GPRINT(FPLOT(1, 5), DEG1, NTIME, YNAME(1,
1 NPRINT, GRID, LINES, TITLE, LABEL)
C 4106 CALL GPRINT(FPLOT(1, 6), DEG1, NTIME, YNAME(1,
1 NPRINT, GRID, LINES, TITLE, LABEL)
C
C IF(CLEAR .LE. -10.0) GO TO 4200
4107 CALL GPRINT(FPLOT(1, 7), DEG1, NTIME, YNAME(1,
1 NPRINT, GRID, LINES, TITLE, LABEL)
C
C 4200 IF(PIN2$8 .EQ. YES) GO TO 4109
4108 CALL GPRINT(FPLOT(1, 8), DEG1, NTIME, YNAME(1,
1 NPRINT, GRID, LINES, TITLE, LABEL)
C
C 4109 CALL GPRINT(FPLOT(1, 9), DEG1, NTIME, YNAME(1,
1 NPRINT, GRID, LINES, TITLE, LABEL)
C
C IF(CLEAR .LE. -10.0) GO TO 4199
4110 CALL GPRINT(FPLOT(1,10), DEG1, NTIME, YNAME(1,10),
1 NPRINT, GRID, LINES, TITLE, LABEL)
C
C 4199 CONTINUE
C RETURN
C END

```


00000470
 00000480
 00000490
 00000500
 00000510
 00000520
 00000530
 00000540
 00000550
 00000560
 00000570
 00000580
 00000590
 00000600
 00000610
 00000620
 00000630
 00000640
 00000650
 00000660
 00000670
 00000680
 00000690
 00000700
 00000710
 00000720
 00000730
 00000740
 00000750
 00000760
 00000770

```

C 350 CONTINUE
C      DO 8 I = 1, KSYS
C      F(I) = 0.0D0
C      8 CONTINUE
C
C      *****
C      **
C      ** CALL SUBROUTINE DINVS
C      ** DINVS CALCULATES THE INVERSE STIFFNESS
C      ** MATRIX
C      ** SEND STF - - RETURN STFRI
C      ** *****
C      **
C      **
C      CALL DINVS(STFRI,ISYS,ISYS)
C
C      DO 360 JA = 1,ISYS
C      Q(JA+1) = 0.
C
C      DO 360 JB = 1,ISYS
C      Q(JA+1) = Q(JA+1)+STFRI(JA,JB)*DP(JB)
C      360 CONTINUE
C
C      CALL ADJUST(1,9, Q, QD, QDD, Q, QD, QDD)
C      CALL ADJUST(8,2, Q, QD, QDD, Q, QD, QDD)
C
C      RETURN
C      END
  
```



```

SUBROUTINE INPUT(LABEL, TITLE, XNAME, YNAME, YNAME, TAPE, IPLOT, IDRAW,
1 KPRINT, DAMPNG, RPM, CLEAR, WIDE, AMP, DELT,
2 ROCKER, AXILTH, REALTH,
3 NTIME, KSYS, NCLEAR, NC,
4 NCOUNT, IDEG, PIN2$8, MPRINT)
5
C IMPLICIT REAL*8(A-H,O-Z)
C
C REAL*4 LABEL, TITLE, XNAME, YNAME,
1 VLV, TW01, TW02
C
C INTEGER YES, AGTOUT, DAMPNG, TAPE, CARDS, SEAT, PIN2$8
C
C DIMENSION YNAME(20,1), AXILTH(1), REALTH(1),
1 LABEL(1), TITLE(1), XNAME(1)
C
C COMMON NUNIT, IPRINT
DATA YES/4HYES /, NO/4HNO /
DATA TW01/, T/, TW02/, WO C/, VLV/, NO V./
C
5001 FORMAT(18A4)
5002 FORMAT(1A4)
5003 FORMAT(D10.3)
5004 FORMAT(14)
5005 FORMAT(1A4,2I4)
5006 FORMAT(4D10.3)
5007 FORMAT('0',10(14,8X) )
5008 FORMAT('0',6(D10.3,5X) )
5009 FORMAT('0',8(A4,10X) )
5010 FORMAT(D10.3, 2X, 1A4)
C
REALTH(1) = 9.0D0
REALTH(2) = 1.0D0
REALTH(3) = 1.5D0
REALTH(4) = 36.0D0
AXILTH(1) = 1.0D0
AXILTH(2) = 3.4D0
AXILTH(3) = 5.1D0
AXILTH(4) = 1.0D0
AMP = 0.5D0
DAMPNG = YES
KSYS = 8
NC = 8
NCLEAR = 3
NUNIT = 6

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460

```


00000950
00000960
00000970
00000980
00000990
00001000
00001010
00001020
00001030
00001040
00001050
00001060
00001070
00001080
00001090
00001100

```
C RETURN  
C 50 IF( IUNIT .EQ. 4 ) END FILE 4  
C IF( IUNIT .EQ. 4 ) END FILE 4  
C IF( IUNIT .EQ. 4 ) REWIND 4  
C STOP  
C  
C  
C  
C  
C  
C  
C-----  
C END
```



```

C      KCOUNT = KCOUNT + 1
C      JCIQ = JC
C      IFOR = 1
C      IF(C .LE. 4) IFOR = 7
C      IF(C .EQ. 1) JCIQ = IQ(JA,6)
C      IF(C .EQ. 3) JCIQ = IQ(JA,8)
C      COLM(JA,C) = SYSM(JCIQ,IFOR,LC)
C      COLC(JA,C) = DAMP(JCIQ,IFOR,LC)
C      COLK(JA,C) = STF (JCIQ,IFOR,LC)
C      200 CONTINUE
C      300 IF(MPRINT .EQ. NO) GO TO 350
C      1 CALL PRNTRM( COLM(1,C), COLC(1,C), COLK(1,C), SYSM(1,1,C),
C      1 DAMP(1,1,C), STF(1,1,C), KSYS, C, IDIM(C) )
C      350 IF(C .GT. 1) RETURN
C      DO 400 JA = 1,KSYSM1
C      DO 400 JB = 1,KSYSM1
C      STFI(JA,JB) = STF(JA,JB,3)
C      400 CONTINUE
C      *****
C      ** CALL SUBROUTINE DINVS
C      ** DINVS CALCULATES THE INVERSE STIFFNESS
C      ** MATRIX
C      ** SEND STF - - RETURN STFI
C      *****
C      CALL DINVS(STFI,KSYSM1,KSYSM1)
C      RETURN
C      END
00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630
00000640
00000650
00000660
00000670
00000680
00000690
00000700
00000710
00000720
00000730
00000740
00000750
00000760
00000770
00000780
00000790
00000800
00000810
00000820
00000830
00000840
00000850
00000860
00000870
00000880
00000890

```

SUBROUTINE CODED BY DR. R. C. WINFREY AND R. ANDERSON
MODIFIED BY C. W. GNILKA

```

1 SUBROUTINE MCK( SYSM, DAMP, STF, KSYS, JC, N, NPRINT, COFD, PHI,
C ELAM, C)
C
C IMPLICIT REAL*8(A-H,O-Z)
C
C INTEGER YES, C
C
C DIMENSION PHINV(8,8), PHITM(8,8), PHITK(8,8), DIAGM(8,8),
C E(5), SEE(8,8), PHINV(8,8), PHI(KSYS,KSYS,1),
C SYSM(KSYS,KSYS,1), STF(KSYS,KSYS,1), DAMP(KSYS,KSYS,1),
C DIAGK(8,8), DIAGC(8,8), ZTA(8,8), BETA(8,8), L(8), Y(5),
C AM(5), AMM(5), DIA(5), ELAM(KSYS,1), PHTIDC(8,8)
C
C COMMON NUNIT, IPRINT
C
C DATA YES/4HYES /, NO/4HNO /
C
C LENGTH
C DATA Y/9.0D0, 1.0D0, 1.5D0, 36.0D0, 3.0D0/
C
C DATA AMM/5*0.283D0/, E/5*3.0D7/
C
C DATA DIA/0.3D0, 0.6D0, 0.6D0, 0.07D0, 0.3D0/
C
C DATA PI/3.14159265359D0/, GRAV/386.0D0/
C
C330 FORMAT('0',//,10X,'ERROR RETURN FROM DVVS. NERR = ', I3,/// )
C700 FORMAT(' ',4X, 8(1PD15.8, 1X) )
C
C IF(NPRINT .EQ. YES) WRITE(NUNIT,700) (ZTA(KA), KA = 1,JC)
C RAD = PI/180.0D0
C N IS THE NUMBER OF ELEMENTS
C JC = SIZE OF K (OR M) AND NUMBER OF COL OF B
C
00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380

```



```

C      DO 5 I = 1, N
C      AM(I) = AMM(I) * PI * DIA(I)**2 * Y(I) / (4.0D0 * GRAV)
5      CONTINUE
C      DO 10 I = 1, JC
C      ZTA(I) = 0.1D0
C      IF(C.EQ. 1) ZTA(4) = 0.5D0
C      IF(C.EQ. 3) ZTA(1) = 0.5D0
C      IF(C.EQ. 3) ZTA(4) = 0.1D0
C      SET SYSTEM K (OR M) TO 0
C      DO 10 J = 1, JC
C      SYM(I,J,C) = 0.0D0
C      STF(I,J,C) = 0.0D0
10     CONTINUE
C      CALCULATE EACH ELEMENT K (OR M)
C      APYX = PI * E(3) * DIA(3)**4 / 64.0D0
C      APYY = PI * E(2) * DIA(2)**4 / 64.0D0
C      ONE = E(1) * PI * DIA(1)**2 / ( Y(1)**4.0D0 )
C      TWO = 12.0D0 * APYY / ( Y(2)**3 )
C      THREE = 6.0D0 * APYY / ( Y(2)**2 )
C      FOUR = 4.0D0 * APYY / Y(2)
C      FIVE = 2.0D0 * APYY / Y(2)
C      SIX = 12.0D0 * APYX / ( Y(3)**3 )
C      SEVEN = E(5) * PI * DIA(5)**2 / ( Y(5)**4.0D0 )
C      EIGHT = 6.0D0 * APYX / ( Y(3)**2 )
C      NINE = 4.0D0 * APYX / Y(3)
C      TEN = E(4) * PI * DIA(4)**2 / ( Y(4)**4.0D0 )
C      ELVEN = 2.0D0 * APYX / Y(3)
C      STF(1,1,C) = ONE
C      STF(1,8,C) = -ONE
C      STF(2,2,C) = TWO
C      STF(2,3,C) = THREE
C      STF(2,6,C) = THREE
C      STF(3,3,C) = FOUR
C      STF(3,6,C) = FIVE
C      STF(4,4,C) = SIX
C      STF(4,5,C) = SIX + SEVEN + TEN
C      STF(4,7,C) = -EIGHT
C      STF(4,8,C) = -EIGHT
C      STF(5,5,C) = NINE
C      STF(5,6,C) = ELVEN
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460
00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630
00000640
00000650
00000660
00000670
00000680
00000690
00000700
00000710
00000720
00000730
00000740
00000750
00000760
00000770
00000780
00000790
00000800
00000810
00000820
00000830
00000840
00000850
00000860

```



```

STF(6,6,C) = ANINE + FOUR
STF(7,7,C) = SEVEN
STF(8,8,C) = ONE

SYSM(1,1,C) = AM(1) / 3.0D0
SYSM(1,8,C) = AM(1) / 6.0D0
SYSM(2,2,C) = 156.0D0 * AM(2) / 420.0D0
SYSM(2,3,C) = AM(2) * 22.0D0 * Y(2) / 420.0D0
SYSM(2,6,C) = -AM(2) * 13.0D0 * Y(2) / 420.0D0
SYSM(3,3,C) = -AM(2) * 4.0D0 * Y(2)**2 / 420.0D0
SYSM(3,6,C) = -AM(2) * 3.0D0 * Y(2)**2 / 420.0D0
SYSM(4,4,C) = -AM(3) * 156.0D0 / 420.0D0 + AM(5)
SYSM(4,5,C) = -AM(3) * 22.0D0 * Y(3) / 420.0D0
SYSM(4,6,C) = -AM(3) * 13.0D0 * Y(3) / 420.0D0
SYSM(4,7,C) = -AM(5) / 6.0D0
SYSM(5,5,C) = AM(3) * 4.0D0 * Y(3)**2 / 420.0D0
SYSM(5,6,C) = -AM(3) * 3.0D0 * Y(3)**2 / 420.0D0
SYSM(6,6,C) = SYSM(3,3,C) + SYSM(5,5,C)
SYSM(7,7,C) = AM(5) / 3.0D0
SYSM(8,8,C) = AM(1) / 3.0D0

IF(C.EQ.1) GO TO 12
STF(1,2,3) = -ONE
STF(2,2,3) = ONE + TWO
SYSM(1,2,3) = AM(1) / 6.0D0
SYSM(2,2,3) = AM(1) / 3.0D0 + 156.0D0*AM(2)/420.0D0

ZTA(1) = 0.1D0
ZTA(4) = 0.5D0
CONTINUE
DO 20 JA = 1, JC
  DO 20 JB = 1, JC
    STF(JB,JA,C) = STF(JA,JB,C)
    SYSM(JB,JA,C) = SYSM(JA,JB,C)
    DIAGC(JA,JB) = SYSM(JA,JB,C)
    PHINV(JA,JB) = STF(JA,JB,C)
  CONTINUE
CONTINUE

24 CALL DVVS( PHINV, DIAGC, ELAM(1,C), PHI(1,1,C), JC, JC, JC, JC,
1
C IF( NERR .NE. 0) WRITE(NUNIT,330) NERR
C
C DO 37 JA = 1, JC
  ELAM(JA,C) = ELAM(JA,C)**.5
  CONTINUE
37
C

```

```

00000870
00000880
00000890
00000900
00000910
00000920
00000930
00000940
00000950
00000960
00000970
00000980
00000990
00001000
00001010
00001020
00001030
00001040
00001050
00001060
00001070
00001080
00001090
00001100
00001110
00001120
00001130
00001140
00001150
00001160
00001170
00001180
00001190
00001200
00001210
00001220
00001230
00001240
00001250
00001260
00001270
00001280
00001290
00001300
00001310
00001320
00001330
00001340

```



```

C      DO 1310 JA = 1, JC
C      35 DO 1310 JB = 1, JC
C      PHTINV(JA, JB) = PHI(JB, JA, C)
C      PHINV(JA, JB) = PHI(JA, JB, C)
C      DIAGC(JA, JB) = 0.000
C      1310 CONTINUE
C      DO 1312 KA = 1, JC
C      1314 DO 1315 KB = 1, JC
C      PHITM(KA, KB) = 0.000
C      PHITK(KA, KB) = 0.000
C      DO 1315 KC = 1, JC
C      PHITM(KA, KB) = PHTINV(KA, KC)*SYSM(KC, KB, C) + PHITM(KA, KB)
C      PHITK(KA, KB) = PHTINV(KA, KC)* STF(KC, KB, C) + PHITK(KA, KB)
C      1315 CONTINUE
C      DO 1318 KA = 1, JC
C      DO 1318 KB = 1, JC
C      DIAGK(KA, KB) = 0.000
C      DIAGM(KA, KB) = 0.000
C      1316 DO 1318 KC = 1, JC
C      PHITM(KA, KB) = PHITM(KA, KC)*PHI(KC, KB, C) + DIAGM(KA, KB)
C      DIAGK(KA, KB) = PHITK(KA, KC)*PHI(KC, KB, C) + DIAGK(KA, KB)
C      1318 CONTINUE
C      DO 1328 KA = 1, JC
C      DIAGC(KA, KA) = 2.000*ZTA(KA)*( DIAGM(KA, KA)*DIAGK(KA, KA) )**0.500
C      1328 CONTINUE
C      1326 CALL DINVS(PHTINV, JC, KSYS)

```

```

00001350
00001360
00001370
00001380
00001390
00001400
00001410
00001420
00001430
00001440
00001450
00001460
00001470
00001480
00001490
00001500
00001510
00001520
00001530
00001540
00001550
00001560
00001570
00001580
00001590
00001600
00001610
00001620
00001630
00001640
00001650
00001660
00001670
00001680
00001690
00001700
00001710
00001720
00001730
00001740

```



```

C 1327 CALL DINVS(PHINV,JC,KSYS)
C
C DO 1322 KA = 1,JC
C
C DO 1322 KB = 1,JC
PHTIDC(KA,KB) = 0.0D0
C
C 1323 DO 1322 KC = 1,JC
PHTIDC(KA,KB) = PHTINV(KA,KC)*DIAGC(KC,KB) + PHTIDC(KA,KB)
C
C 1322 CONTINUE
C
C DO 1324 KA = 1,JC
C
C DO 1324 KB = 1,JC
SEE(KA,KB) = 0.0D0
C
C 1321 DO 1324 KC = 1,JC
SEE(KA,KB) = PHTIDC(KA,KC)*PHINV(KC,KB) + SEE(KA,KB)
C
C 1324 CONTINUE
C
C DO 1325 KA = 1, JC
DO 1325 KB = 1, JC
DAMP(KA,KB,C) = SEE(KA,KB)*COFD
C 1325 CONTINUE
C
C RETURN
C END

```

```

00001750
00001760
00001770
00001780
00001790
00001800
00001810
00001820
00001830
00001840
00001850
00001860
00001870
00001880
00001890
00001900
00001910
00001920
00001930
00001940
00001950
00001960
00001970
00001980
00001990
00002000
00002010
00002020
00002030
00002040
00002050

```



```

00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630
00000640
00000650
00000660
00000670
00000680
00000690
00000700
00000710
00000720
00000730
00000740
00000750
00000760
00000770
00000780
00000790
00000800
00000810
00000820
00000830
00000840
00000850
00000860
00000870
00000880
00000890
00000900
00000910
00000920
00000930
00000940

C      FSAVE = FSAV( INDEX )
C      FSEND = F( INDEX )
C      CALL DTPRIM( FSEND, KK, FSAVE, DT, DTP, NPRINT )
C      IDT = IDT + 1
C      GO TO 23
C
C      *****
C      **          DISPLACEMENT MONITOR          **
C      *****
C
C      12  DISPL = Q( INDEX ) - Q( NDEX )
C          IF( DISPL .GE. 0.0 ) GO TO 23
C          IF( DABS( DISPL ) .LE. EPS2 ) GO TO 23
C
C          IMPACT( KK ) = KK
C          DOLD = QSAV( INDEX ) - QSAV( NDEX )
C          IF( DOLD .LE. 0.0 ) DOLD = 1.0D-10
C
C          CALL DTPRIM( DISPL, KK, DOLD, DT, DTP, NPRINT )
C          IDT = IDT + 1
C      23  CONTINUE
C
C      25  DO 25 I = 1, NCLEAR
C          IF( DTP( I ) .LT. 0.1D0 ) GO TO 50
C          CONTINUE
C          IDT = -1
C      RETURN
C
C      50  DTPMIN = +1.0D6
C          DO 60 JA = 1, NCLEAR
C              DTPMIN = DMINI( DTPMIN, DTP( JA ) )
C          IF( NPRINT .EQ. YES ) WRITE( NUNIT, 5002 ) JA, DTP( JA ), DTPMIN
C          CONTINUE
C      60
C
C      IF( IDT .GT. 1 .AND. NPRINT .EQ. YES ) WRITE( NUNIT, 5000 )
C
C      70  DO 70 KK = 1, NCLEAR
C          IF( DTPMIN .EQ. DTP( KK ) ) GO TO 80
C          CONTINUE
C      WRITE( NUNIT, 5001 )

```


00000950
00000960
00000970
00000980
00000990
00001000
00001010
00001020
00001030
00001040
00001050
00001060
00001070
00001080
00001090

**
**
**
**
**

CHANGE CONFIGURATION

80 CALL SHIFT(C, KK, CONFIG, NPRINT)

IDT = KK

IF(NPRINT.EQ.NO) RETURN
WRITE(NUNIT,5003) KK, IMPACT(KK)

RETURN
END

C
C
C
C
C
C
C
C

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190

SUBROUTINE CAM(WT, AMP, AMPW, XZERO, W2, Q, QD, QDD)

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

INTEGER YES

DIMENSION Q(1), QD(1), QDD(1)

COMMON NUNIT, IPRINT

DATA YES/4HYES /, NO/4HNO /

AMPSIN = AMP*DSIN(WT)
Q(9) = AMPSIN + XZERO
QD(9) = AMPW*DCOS(WT)
QDD(9) = AMPSIN*W2

RETURN
END

C
C
C
C
C
C
C


```

00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630
00000640
00000650
00000660
00000670
00000680
00000690
00000700
00000710
00000720
00000730
00000740
00000750
00000760
00000770
00000780
00000790
00000800
00000810
00000820
00000830
00000840
00000850
00000860
00000870
00000880
00000890
00000900
00000910
00000920
00000930
00000940

DP(I) = 0.0D0
50 CONTINUE
C
C IF(C .EQ. 1 .OR. C .EQ. 3) GO TO 500
DO 300 JA = 1, NCLEAR
KDP = IDP(C, JA)
IF(KDP .NE. 0) GO TO 325
C 300 CONTINUE
C
C 325 KC = C
C 350 ACCEL = QDD(KDP)
VEL = QD(KDP)
DIST = Q(KDP)
C
C 400 DO 400 JA = 1, ISYS
CSYSM = COLM(JA, KC)
CDAMP = COLC(JA, KC)
CSTFF = COLK(JA, KC)
DP(JA) = CSYSM*ACCEL + CDAMP*VEL + CSTFF*DIST + DP(JA)
CONTINUE
C
C 425 IF(KC .NE. C) GO TO 500
IF(C .EQ. 6) KC = 1
IF(C .EQ. 8) KC = 3
KDP = 7
IF(KC .NE. C) GO TO 350
500 CONTINUE
C
C 600 DO 700 JA = 1, ISYS
IA = IQ(JA, C)
ACCEL = QDDSAV(IA)
VEL = QDSAV(IA)
C
C 700 ALFA(JA) = ACCEL*DT02 + VEL
BETA(JA) = ACCEL*DT203 + VEL*DT + QSAV(IA)
CONTINUE
C
C IF(NPRINT .EQ. NO) GO TO 750
C
C WRITE(NUNIT, 5008) ( ALFA(JA), JA = 1, ISYS )
WRITE(NUNIT, 5009) ( BETA(JA), JA = 1, ISYS )
WRITE(NUNIT, 5011)
C
C 750 DO 900 JA = 1, ISYS
C

```



```

00000950
00000960
00000970
00000980
00000990
00001000
00001010
00001020
00001030
00001040
00001050
00001060
00001070
00001080
00001090
00001100
00001110
00001120
00001130
00001140
00001150
00001160
00001170
00001180
00001190
00001200
00001210
00001220
00001230
00001240
00001250
00001260
00001270
00001280
00001290
00001300
00001310
00001320
00001330
00001340
00001350
00001360
00001370
00001380
00001390
00001400
00001410
00001420

IA = IQ(JA,C)
QDD(IA) = 0.000

DO 800 JB = 1, ISYS
QDD(IA) = QDD(IA) - BMIV(JA,JB)*DP(JB) - DA(JA,JB)*ALFA(JB)
1 - DV(JA,JB)*BETA(JB)
1

IF(NPRINT.EQ.NO) GO TO 800
WRITE(NUNIT,5010) BMIV(JA,JB),DP(JB),DA(JA,JB),ALFA(JB),DV(JA,JB),
BETA(JB), QDD(IA)
1

C 800 CONTINUE
C
C 900 IF(NPRINT.EQ.YES) WRITE(NUNIT,5015)
CONTINUE
C
IF(IMP.NE.2) GO TO 950
QDD(2) = (SCALE - 1.000) * QDSAV(2) / DT
WRITE(NUNIT,5045) QDD(2), SCALE, QDSAV(2)
IMP = -10
950 DO 1000 JA = 1, ISYS
IA = IQ(JA,C)
QD(IA) = ALFA(JA) + QDD(IA)*DT02
Q(IA) = BETA(JA) + QDD(IA)*DT206
C 1000 CONTINUE
IF(IMP.NE.-10) GO TO 1050
IF(IMP.NE.5046) QD(2)
WRITE(NUNIT,5046) QD(2)
FORMAT(' ',10X,'NEW CALCULATED VELOCITY OF COORD. TWO IS ',
1 IPD15.7)
1
IMP = 0
1050 CONTINUE
C
LC = LAYER(C)
IF(LC.EQ.3) GO TO 1170
IF(PIN2$8.EQ.NO) GO TO 1175
C 1150
C 1170 CALL ADJUST(8,2, Q, QD, QDD, Q, QD, QDD)
LC = 3
1175 CONTINUE
IF(IGU.EQ.YES) LC = 1
ILC = IDIM(LC)
C
DO 1300 IA = 1, NCLEAR
KK = ICORD(IA,1)
F(KK) = 0.000
C
DO 1200 JA = 1, ILC

```



```

F(KK) = F(KK) + SYSM(KK,JA,LC)*QDD(JA)
+ DAMP(KK,JA,LC)*QD(JA) + STF(KK,JA,LC)*Q(JA)
1 CONTINUE
C 1200 CONTINUE
C 1300 CONTINUE
C 1350 IF(PIN2$8.EQ.NO) GO TO 1375
C F(2) = 0.000
F(8) = 0.000
C 1375 CONTINUE
IF(NPRINT.EQ.NO) RETURN
WRITE(NUNIT,5037) ( F(JQ), JQ = 1, KSYS )
C WRITE(NUNIT,5042)
C RETURN
END

```

```

00001430
00001440
00001450
00001460
00001470
00001480
00001490
00001500
00001510
00001520
00001530
00001540
00001550
00001560
00001570
00001580
00001590
00001600

```



```

30 CONTINUE
C
C PLOTTING Y AND X AXIS , IF NECESSARY
C
YRANGE=YMAX-YMIN
XRANGE=XMAX-XMIN
IF (XRANGE.NE.0.) GO TO 298
XMIN=YMIN
XMAX=YMAX
XRANGE=YRANGE
GO TO 299
298 IF (YRANGE.NE.0.) GO TO 299
YMIN=XMIN
YMAX=XMAX
YRANGE=XRANGE
C
C BLANKING OUT MATRIX-(GRID)
C
299 DO 300 I=1,IRYP1
DO 301 JJ=1,81
301 GRID(I, JJ)=BLANK
300 CONTINUE
XTEST=XMAX*XMIN
YTEST=YMAX*YMIN
222 IF(XTEST)1,222,222
1 IXAXIS=80.*(-YMIN)/YRANGE+1.5
40 DO 40 I=1,IRYP1
GRID(I, IXAXIS)=SLINE
GO TO 222
333 IYAXIS=RY *XMAX/XRANGE+1.5
60 GRID(IYAXIS, I)=DASH
C
C PLACING POINTS IN THEIR PROPER GRID POSITIONS
C
444 IF(MODCUR.EQ.0.OR.MODCUR.EQ.1)JSET=0
JSET=JSET+1
IF(JSET.GT.4) JSET=1
DO 700 I=1,NDATA,KKZ
IPTX=RY *(X(I)-XMIN)/XRANGE+1.5
IPTX=80.*(Y(I)-YMIN)/YRANGE+1.5 GO TO 70
IF(IPTY.GT.IRYP1.OR.IPTX.GT.81) GO TO 70
IF(IPTY.LE.0.OR.IPTX.LE.0)GO TO 70
GRID(IPTY, IPTX) = YCHAR(JSET)
GO TO 700
70 IERR=IERR+1
700 CONTINUE
C

```

```

00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630
00000640
00000650
00000660
00000670
00000680
00000690
00000700
00000710
00000720
00000730
00000740
00000750
00000760
00000770
00000780
00000790
00000800
00000810
00000820
00000830
00000840
00000850
00000860
00000870
00000880
00000890
00000900
00000910
00000920
00000930
00000940

```



```

SUBROUTINE PRINT(KSYS, ISYS, C, SYSM, DAMP, STF, BMIV, DA, DV,
1  CONFIG, PHI, ELAM, STFI, STFRI, WTSTOP, WTPRNT, LABEL,
2  YNAME, XNAME, TITLE,
3  AGTOUT, CARDS, TAPE, IPLOT, IDRAW, DAMPNG, RPM, CLEAR,
4  AMP, DELT, ROCKER, AXILTH, REALTH, NIIME, NC,
5  NCLEAR, KSYSM1, COLM, COLC, COLK, COFD, ICORD,
6  LAYER, IDP, IQ, IDIM, KSYSM2, PIN2$8, XZERO, FZERO,
7  MPRINT)
00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420

C      IMPLICIT REAL*8(A-H,O-Z)
C      REAL*4 LABEL, TITLE, XNAME, YNAME
C      INTEGER YES, AGTOUT, DAMPNG, TAPE, CARDS, SEAT, C, CONFIG(16,1), PIN2$8
C
1  DIMENSION PHI(KSYS, KSYS), ELAM(KSYS), SYSM(KSYS, KSYS),
2  DAMP(KSYS, KSYS), STF(KSYS, KSYS), BMIV(KSYS, KSYS),
3  DA(KSYS, KSYS), DV(KSYS, KSYS),
4  STFI(KSYSM1, 1), STFRI(KSYSM2, 1), LABEL(1), TITLE(1),
5  YNAME(20, 1), AXILTH(1), REALTH(1),
6  XNAME(1), COLM(KSYS, 1), COLC(KSYS, 1), COLK(KSYS, 1),
7  ICORD(NCLEAR, 1)
C      DIMENSION LAYER(1), IDP( KSYS, 1), IQ(KSYS, 1), IDIM(1)
C
C      COMMON NUNIT, IPRINT
C
C      DATA YES/4HYES /, NO/4HNO /
620  FORMAT('0', 10X, ' MODE SHAPES-(PHI)', //)
660  FORMAT('0', 10X, ' NATURAL FREQ-(WN)', //)
1020  FORMAT('0', 50X, ' -( BMIV ) -', //)
1030  FORMAT('0', 20X, ' 1P8D10.2) -', //)
1040  FORMAT('0', 50X, ' -( DA ) -', //)
1049  FORMAT('1', //)
1050  FORMAT('0', 40X, ' CONFIGURATION AND MATRIX LAYER =', I3, //, 50X,
1  ' MATRICES ARE', I3, 'X', I1, //)
1060  FORMAT('0', 50X, ' -( DV ) -', //)
1070  FORMAT('0', //, 50X, ' --- SYSM ---', //)
1071  FORMAT('0', //, 50X, ' --- DAMP ---', //)
1072  FORMAT('0', //, 50X, ' --- STF ---', //)
5000  FORMAT('0', 2(10X, ICORD( , I1, , I1, ' )=' , I3) )
5001  FORMAT('1', //)

```



```

5002 FORMAT('0', 20X, 'COORDINATES 2 AND 8 ARE PINED -- TWO ',
5003 'CLEARANCE PROBLEM')
5004 1 FORMAT('0', 20X, 'COORDINATES 2 AND 8 ARE NOT PINED -- THREE ',
5010 1 'CLEARANCE PROBLEM')
5017 1 '5X15DAMPING =,OPF9.2,5X,7HCLEAR =,OPF6.3)
5020 1 'XZERO =,IPD15.7, FZERO =, D15.7)
5021 'AGT OUTPUT REQUESTED')
5022 'AGT OUTPUT ON MAGNETIC TAPE')
5023 'AGT OUTPUT ON PUNCHED CARDS')
5024 'DAMPING REQUESTED')
5025 'NO DAMPING REQUESTED')
5026 'NO VALVE SEAT')
5027 'CONFIGURATION MATRIX')
5028 'VALVE SEAT IMPOSED WITH AN INITIAL CLEARANCE OF ',
5041 1 'F5.2, INCHES')
5042 'NO VALVE SEAT')
5043 'COL-M-C-K',//,6(10X,1P3D18.9,//) )
5044 'IP7D18.9)
5045 'STFI -- -- -- ',//)
5046 'STFRI -- -- -- ',//)
5047 'CALCOMP GRAPH DESIRED')
5048 'CALCOMP GRAPH NOT DESIRED')
5050 'MATRIX -- LAYER --')
5051 'MATRIX -- IDP --')
5052 'MATRIX -- IQ --')
5053 'MATRIX -- IDIM --')
5054 'MATRIX -- ICORD --')
5055 'WTSTOP =,IPD18.8, 5X, 'WTPRNT =,D18.8)
5056 'MATRIX -- ICORD --')
5057 'WTSTOP =,IPD18.8, 5X, 'WTPRNT =,D18.8)
5058 'MATRIX -- ICORD --')
5312 'WTSTOP =,IPD18.8, 5X, 'WTPRNT =,D18.8)
C
C
IF(C .EQ. 3) GO TO 358
IF(C .EQ. 2 .OR. C .GE. 4) GO TO 359
C
WRITE(NUNIT,5001)
C

```

```

00000430
00000440
00000450
00000460
00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630
00000640
00000650
00000660
00000670
00000680
00000690
00000700
00000710
00000720
00000730
00000740
00000750
00000760
00000770
00000780
00000790
00000800
00000810
00000820
00000830
00000840
00000850

```


00000860
 00000870
 00000880
 00000890
 00000900
 00000910
 00000920
 00000930
 00000940
 00000950
 00000960
 00000970
 00000980
 00000990
 00001000
 00001010
 00001020
 00001030
 00001040
 00001050
 00001060
 00001070
 00001080
 00001090
 00001100
 00001110
 00001120
 00001130
 00001140
 00001150
 00001160
 00001170
 00001180
 00001190
 00001200
 00001210
 00001220
 00001230
 00001240
 00001250
 00001260
 00001270
 00001280
 00001290

```

C      WRITE(NUNIT,5026) ( LABEL(JA), JA = 1,18 )
C      WRITE(NUNIT,5026) ( TITLE(JA), JA = 1,18 )
C      WRITE(NUNIT,5026) ( YNAME(JA,1), JA = 1,18 )
C      WRITE(NUNIT,5026) ( YNAME(JA,2), JA = 1,18 )
C      WRITE(NUNIT,5026) ( YNAME(JA,3), JA = 1,18 )
C      WRITE(NUNIT,5026) ( YNAME(JA,4), JA = 1,18 )
C      WRITE(NUNIT,5026) ( YNAME(JA,5), JA = 1,18 )
C      WRITE(NUNIT,5026) ( YNAME(JA,6), JA = 1,18 )
C      WRITE(NUNIT,5026) ( YNAME(JA,7), JA = 1,18 )
C      WRITE(NUNIT,5026) ( YNAME(JA,8), JA = 1,18 )
C      WRITE(NUNIT,5026) ( YNAME(JA,9), JA = 1,18 )
C      WRITE(NUNIT,5026) ( YNAME(JA,10), JA = 1,18 )
C      WRITE(NUNIT,5026) ( YNAME(JA,11), JA = 1,18 )
C      WRITE(NUNIT,5026) ( YNAME(JA,12), JA = 1,18 )
C      WRITE(NUNIT,5026) ( YNAME(JA,13), JA = 1,18 )
C
C      IF( AGTOUT .EQ. YES ) WRITE(NUNIT,5020)
C      IF( AGTOUT .EQ. NO ) WRITE(NUNIT,5021)
C
C      IF( TAPE .EQ. YES ) WRITE(NUNIT,5022)
C
C      IF( CARDS .EQ. YES ) WRITE(NUNIT,5023)
C
C      IF( DAMPING .EQ. NO ) WRITE(NUNIT,5025)
C      IF( DAMPING .EQ. YES ) WRITE(NUNIT,5024)
C
C      IF( IDRAW .EQ. YES ) WRITE(NUNIT,5047)
C      IF( IDRAW .EQ. NO ) WRITE(NUNIT,5048)
C
C      IF( CLEAR .LE. -10.0 ) WRITE(NUNIT,5042)
C      IF( CLEAR .GT. -10.0 ) WRITE(NUNIT,5041) CLEAR
C
C      IF( PIN2$8 .EQ. YES ) WRITE(NUNIT,5003)
C      IF( PIN2$8 .EQ. NO ) WRITE(NUNIT,5004)
C
C      WRITE(NUNIT,5010) RPM, COFD , CLEAR
C      WRITE(NUNIT,5017) XZERO, FZERO
C      IF( MPRINT .EQ. NO ) RETURN
C
C      WRITE(NUNIT,5027)
C      DO 80 JA = 1, NC
C      WRITE(NUNIT,5028) ( CONFIG(JA,JB), JB = 1, NCLEAR)

```



```

80 CONTINUE
C WRITE(NUNIT,5050)
DO 700 JA = 1, KSYS
WRITE(NUNIT,5051) LAYER(JA)
CONTINUE
C
700
C WRITE(NUNIT,5002)
WRITE(NUNIT,5052)
DO 710 JA = 1, KSYS
WRITE(NUNIT,5053) (IDP(JA,JB), JB = 1, NCLEAR)
CONTINUE
C
710
C WRITE(NUNIT,5054)
DO 720 JA = 1, KSYS
WRITE(NUNIT,5055) (IQ(JA,JB), JB = 1, KSYS)
CONTINUE
C
720
C WRITE(NUNIT,5002)
WRITE(NUNIT,5056)
DO 730 JA = 1, KSYS
WRITE(NUNIT,5057) IDIM(JA)
CONTINUE
C
730
C WRITE(NUNIT,5058)
DO 740 JA = 1, NCLEAR
WRITE(NUNIT,5053) (ICORD(JA,JB), JB = 1,2)
CONTINUE
C
740
C WRITE(NUNIT,5001)
C WRITE(NUNIT,5045)
C
221 JA = 1, KSYSM1
WRITE(NUNIT,5044) (STFI(JA,JB), JB = 1, KSYSM1)
CONTINUE
C
C WRITE(NUNIT,5046)
C
7 JA = 1, KSYSM2
WRITE(NUNIT,5044) (STFRI(JA,JB), JB = 1, KSYSM2)
CONTINUE
C
358 CONTINUE
C WRITE(NUNIT,620)
C
DO 640 JA = 1, ISYS

```

```

00001300
00001310
00001320
00001330
00001340
00001350
00001360
00001370
00001380
00001390
00001400
00001410
00001420
00001430
00001440
00001450
00001460
00001470
00001480
00001490
00001500
00001510
00001520
00001530
00001540
00001550
00001560
00001570
00001580
00001590
00001600
00001610
00001620
00001630
00001640
00001650
00001660
00001670
00001680
00001690
00001700
00001710
00001720
00001730
00001740
00001750
00001760
00001770

```



```

00001780
00001790
00001800
00001810
00001820
00001830
00001840
00001850
00001860
00001870
00001880
00001890
00001900
00001910
00001920
00001930
00001940
00001950
00001960
00001970
00001980
00001990
00002000
00002010
00002020
00002030
00002040
00002050
00002060
00002070
00002080
00002090
00002100
00002110
00002120

WRITE(NUNIT,1030) ( PHI(JA,JB), JB = 1, ISYS)
640 CONTINUE
C
WRITE(NUNIT,660)
WRITE(NUNIT,1030) ( ELAM(JA), JA = 1, ISYS)
C
C
359 CONTINUE
C
IF(MPRINT .EQ. NO) RETURN
C
2 WRITE(6,1049)
WRITE(NUNIT,1050) C, ISYS, ISYS
C
C
WRITE(NUNIT,1020)
DO 40 JA = 1, ISYS
40 WRITE(NUNIT,1030) ( BMIV(JA,JB), JB = 1, ISYS)
C
WRITE(NUNIT,5002)
WRITE(NUNIT,1040)
DO 50 JA = 1, ISYS
50 WRITE(NUNIT,1030) ( DA(JA,JB), JB = 1, ISYS)
CONTINUE
C
WRITE(NUNIT,5002)
WRITE(NUNIT,1060)
DO 60 JA = 1, ISYS
60 WRITE(NUNIT,1030) ( DV(JA,JB), JB = 1, ISYS)
CONTINUE
C
70 CONTINUE
C
RETURN
END

```



```

C
3 DO 3 JA = 1, ISYS
WRITE(NUNIT,1030) ( SYSM(JA,JB) , JB = 1, ISYS)
CONTINUE
C
4 WRITE(NUNIT,1071)
DO 4 JA = 1, ISYS
WRITE(NUNIT,1030) ( DAMP(JA,JB) , JB = 1, ISYS)
CONTINUE
C
5 WRITE(NUNIT,1072)
DO 5 JA = 1, ISYS
WRITE(NUNIT,1030) ( STF(JA,JB) , JB = 1, ISYS)
CONTINUE
C
WRITE(6,1049)
WRITE(6,1048)
WRITE(NUNIT,1051) C, ISYS
WRITE(NUNIT,1073)
C
6 DO 6 JA = 1, ISYS
WRITE(NUNIT,1031) COLM(JA)
WRITE(NUNIT,1032) COLC(JA)
WRITE(NUNIT,1033) COLK(JA)
CONTINUE
C
RETURN
END

```

```

00000440
00000450
00000460
00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630
00000640
00000650
00000660
00000670
00000680
00000690
00000700
00000710

```



```

C      SUBROUTINE REDUCE( SYSM, DAMP, STF, KSYS)
C      IMPLICIT REAL*8(A-H,O-Z)
C      DIMENSION SYSM(KSYS,KSYS,1), DAMP(KSYS,KSYS,1), STF(KSYS,KSYS,1)
      DO 10 JA = 1, KSYS
      DO 10 JB = 1, KSYS
      SYSM(JA,JB,3) = SYSM(JA,JB,1)
      DAMP(JA,JB,3) = DAMP(JA,JB,1)
      STF(JA,JB,3) = STF(JA,JB,1)
10 CONTINUE
C      DO 20 JC = 1, KSYS
      SYSM(2,JC,3) = SYSM(8,JC,3)
      DAMP(2,JC,3) = DAMP(8,JC,3)
      STF(2,JC,3) = STF(8,JC,3)
20 CONTINUE
C      DO 30 JD = 1, KSYS
      SYSM(JD,2,3) = SYSM(JD,8,3)
      DAMP(JD,2,3) = DAMP(JD,8,3)
      STF(JD,2,3) = STF(JD,8,3)
30 CONTINUE
C      RETURN
      END
00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270

```



```

C      SUBROUTINE SHIFT( C, J, CONFIG, NPRINT)
C      INTEGER CONSAV, CSAVE, C, CONFIG, BINARY, YES
C      DIMENSION CONFIG(16,1)
C      COMMON NUNIT, IPRINT
C      DATA YES/4HYES /, NO/4HNO /
C      BINARY(I3,I2,I1) = I3*4 + I2*2 + I1
C      10 FORMAT('0', 10X, 'FROM SHIFT, C=', I1, ' CONFIG(', I1, ',', I1,
C      11 FORMAT(' ', 10X, 'NEW C = ', I1)
C      IF(NPRINT .EQ. NO) GO TO 12
C      WRITE(NUNIT,10) C,C,J,CONFIG(C,J)
C      12 CONSAV = CONFIG( C, J)
C      CSAVE = C
C      CONFIG(C,J) = CONSAV + 1
C      IF( CONFIG(C,J) .EQ. 2 ) CONFIG(C,J) = 0
C      C = BINARY( CONFIG(C,1), CONFIG(C,2), CONFIG(C,3) ) + 1
C      IF(NPRINT .EQ. NO) GO TO 20
C      WRITE(NUNIT,11) C
C      20 CONFIG(CSAVE,J) = CONSAV
C      RETURN
C      END
00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370

```


00000470
 00000480
 00000490
 00000500
 00000510
 00000520
 00000530
 00000540
 00000550
 00000560
 00000570
 00000580
 00000590
 00000600
 00000610
 00000620
 00000630
 00000640
 00000650
 00000660
 00000670
 00000680
 00000690
 00000700
 00000710
 00000720
 00000730
 00000740
 00000750
 00000760
 00000770
 00000780
 00000790
 00000800
 00000810
 00000820
 00000830
 00000840
 00000850
 00000860
 00000870
 00000880
 00000890
 00000900
 00000910
 00000920
 00000930
 00000940

```

C 5000 FORMAT('0','CONFIGURATION =',I3,5X,'DEGREES =',F9.3,10X,'WT =',
      1 IPD15.7)
5001 FORMAT('1')
5002 FORMAT('0','IOX, BACKSTEP ROUTINE CALLED DUE TO CHANGE IN ',
      1 CONFIGURATIONS,')
5003 FORMAT(' ','IOX, RETURNED TO MAIN FROM BACKSTEP ROUTINE')
5004 FORMAT(' ','IOX, IN MAIN - LOCATION ',I4,' - - - CSAVE=', I2,
      1 AND C =', I2)
5005 FORMAT('0','AMPW=', IPD15.7, ' AMPW=', D15.7, ' W2=', D15.7,
      1 AMPW2=', D15.7)
5007 FORMAT('0','IOX, VALUES CALCULATED AFTER BACKSTEP TAKEN:')
5015 FORMAT(5E10.4,2A4)
5016 FORMAT('0','IOX, IMPACT FORCE AT COORDINATE ',I2, '= ',1PD15.7,
      1 - - - DEGREES ', OPF9.3)
5026 FORMAT('0','IOX,20A4)
5038 FORMAT(20A4)
5039 FORMAT('0',20X,20A4)
5040 FORMAT('0',30X,20A4)
5041 FORMAT('0',5X,I5,5(10X,1PE10.3) )
5050 FORMAT('0',40X,20A4)
5051 FORMAT('0',50X,20A4)
5062 FORMAT('0',23X,'DEGREES =',F9.3,10X,'WT =',1PD15.7)
5063 FORMAT('0','CAMQD =',1PD18.8, 10X,'CAMQD =', D18.8, 10X,'CAMQDD =',
      1 D18.8)
5310 FORMAT('0',5(1PD15.7, 5X) )

```

```

C C C C C C C C C C
      *****
      **
      **
      **
      *****
      READ INPUT VARIABLES
      *****
      **
      **
      **
      *****

```

```

50 CALL INPUT(LABEL, TITLE, XNAME, YNAME, IDRAW, AMP, DELT,
      1 AGTOUT, CARDS, TAPE, IPLOT, WIDEG,
      2 KPRINT, DAMPNG, RPM, CLEAR,
      3 ROCKER, AXILTH,
      4 REALTH, NTIME, KSYS, NCLEAR, NC,
      5 NCOUNT, IDEG, PIN2$8, MPRINT)

```

```

FS = +145.000
PI = 3.14159265359000
DEGR = 0.
WT = 0.
NDEG = 0
NDEGKP = 0

```



```

ICOUNT = 0
KAGT = 0
IAGT = 1
IDT = 0
IMP = 0
DTPM = 0.000
COFD = 1.000
IUNIT = 6
ICORD(1,1) = 1
ICORD(1,2) = 9
ICORD(2,1) = 2
ICORD(2,2) = 8
ICORD(3,1) = 7
ICORD(3,2) = 10
DT = DELT
NPRINT = IPRINT
SEAT = YES

NSYS = KSYS + 2
DEGOPI = 180.000/PI
KSYSM1 = KSYS-1
KSYSM2 = KSYS-2
KSYSM3 = KSYS - 3
NCONF = NC
NCPI = NC + 1

Q(10) = CLEAR
QD(10) = 0.000
QDD(10) = 0.000
QS(10) = CLEAR
QDS(10) = 0.000
QDD(10) = 0.000
W = RPM*PI/30.000
AMPW = AMP*W
W2 = -W*W
AMPW2 = -AMP*W
DTW = DT*W
WTSTOP = WTDEG
WTPRINT = WTDEG

WRITE(NUNIT,5005) W, AMPW, W2, AMPW2
WRITE(NUNIT,5310) WTSTOP, WTPRINT, WTDEG, DTW, DEGOPI
WRITE(NUNIT,5039) (LABEL(JA), JA=1,18)
WRITE(NUNIT,5039) (TITLE(JA), JA=1,18)

IF(TAPE.EQ.YES) IUNIT = 4
IF(CARDS.EQ.YES) IUNIT = 7
IF(CLEAR.LE.-10.0) SEAT = NO

```

C

C

C

C

00001910
 00001920
 00001930
 00001940
 00001950
 00001960
 00001970
 00001980
 00001990
 00002000
 00002010
 00002020
 00002030
 00002040
 00002050
 00002060
 00002070
 00002080
 00002090
 00002100
 00002110
 00002120
 00002130
 00002140
 00002150
 00002160
 00002170
 00002180
 00002190
 00002200
 00002210
 00002220
 00002230
 00002240
 00002250
 00002260
 00002270
 00002280
 00002290
 00002300
 00002310
 00002320
 00002330
 00002340
 00002350
 00002360
 00002370
 00002380

 ** COMPUTE INITIAL PROGRAM PARAMETERS FOR CONTACT AT CAM AND ROCKER **

120 CALL INITIAL(KSYS,Q,QD,QDD,AMPW,FS,FZERO,XZERO,ROCKER,KSYSM2,
 1 COLM(1,7),COLC(1,7),COLK(1,7),DP,F,DAMP(1,1,7),
 2 STF(1,1,7),STFRI,NPRINT)

 ** CALL SUBROUTINE DEL ESTABLISHES MATRICES USED IN THE DEL INTEGRATION ROUTINE. - DEL SETS : DELT/2, DELT*2/3, DELT*2/6 BMIVR, DAV, DVR *****

DO 180 C = 1,NCONF

ISYS = IDIM(C)

183 CALL DEL(DT,DT02,DT203,DT206,BMIV(1,1,C),DA(1,1,C),DV(1,1,C),
 1 SYSM(1,1,C),DAMP(1,1,C),STF(1,1,C),ISYS,C,NPRINT,KSYS,
 2 KPRINT)

IF(MPRINT.EQ.NO.AND.C.NE.1) GO TO 180

IF(C.EQ.1) CALL
 3 .AND.IGU.EQ.NO) IC = 3
 PRINT(KSYS,ISYS,C,SYSM(1,1,C),DAMP(1,1,C),CONFIG
 STF(1,1,C),BMIV(1,1,C),DA(1,1,C),DV(1,1,C),WTSTOP,
 (1,2),PHI(1,1,IC),ELAM(1,1,IC),STFI,STFRI,WTSTOP,
 WTPRINT,LABEL,YNAME,IPLLOT,ITLE,DAMPNG,RPM,CLEAR,
 AGTOUT,CARDS,TAPE,AXILTH,REALTH,NTIME,NC,
 AMP,DELT,ROCKER,AXILTH,COLK,COFD,ICORD,
 NCLEAR,KSYSM1,COLM,COLK,COFD,ICORD,
 LAYER,IDP,IQ,IDIM,KSYSM2,PIN2\$8,XZERO,FZERO,
 MPRINT)

CCCCCCCCC CCCCCCCCCCCCCC CC CC CC C


```

00002390
00002400
00002410
00002420
00002430
00002440
00002450
00002460
00002470
00002480
00002490
00002500
00002510
00002520
00002530
00002540
00002550
00002560
00002570
00002580
00002590
00002600
00002610
00002620
00002630
00002640
00002650
00002660
00002670
00002680
00002690
00002700
00002710
00002720
00002730
00002740
00002750
00002760
00002770
00002780
00002790
00002800
00002810
00002820
00002830
00002840
00002850
00002860

```

```

C 180 CONTINUE
C WRITE(NUNIT,5001)
C INITIAL CONTACT AT CAM AND ROCKER
C 190 C = 7
C CSAVE = C + 1
C *****
C ** CONVERT TO REAL * 4
C ** OUTPUT FOR AGT
C ** *****
C IF( AGTOUT .EQ. NO ) GO TO 265
C RXZERO = XZERO
C RAMP = AMP
C RRRPM = RPM
C RRCLEAR = CLEAR
C WNR4 = WN
C IF(IUNIT .EQ. 6) GO TO 260
C WRITE(IUNIT, 5038) (LABEL(JA), JA=1,18)
C WRITE(IUNIT, 5038) (TITLE(JA), JA=1,18)
C 260 WRITE(IUNIT,5015) RXZERO,RAMP,RRPM, WNR4, RRCLEAR, SEAT, PIN2$8
C 265 CONTINUE
C *****
C ***** CONFIGURATION CALCULATIONS *****
C *****
C ** INITIALIZING ROUTINE
C **
C ** *****
C 360 WT = WT + DTW
C DO 361 JA = 1, NSYS
C CALL ADJUST(JA,JA,QSAV,QDSAV,QDDSAV,Q,QD,QDD)
C 361 CONTINUE
C DO 364 I = 1, KSYS

```



```

00003340
00003350
00003360
00003370
00003380
00003390
00003400
00003410
00003420
00003430
00003440
00003450
00003460
00003470
00003480
00003490
00003500
00003510
00003520
00003530
00003540
00003550
00003560
00003570
00003580
00003590
00003600
00003610
00003620
00003630
00003640
00003650
00003660
00003670
00003680
00003690
00003700
00003710
00003720
00003730
00003740
00003750
00003760
00003770
00003780
00003790
00003800

*****
** DETERMINE IF IMPACT AND AT WHAT COORD. **
*****
DO 415 IMP = 1, NCLEAR
IF(IDT.EQ. IMPACT(IMP) ) GO TO 418
CONTINUE
IMP = 0
GO TO 420
C 415

418 IMP = ICORD(IMP,1) GO TO 450
420 IF(CSAVE .EQ. C ) GO TO 450
C 420

IF(NPRINT .EQ. YES) WRITE(NUNIT,5002)
C 440 CALL BKSTEP( BMIV(1,1,NCPI), DA(1,1,NCPI), DV(1,1,NCPI), W, WT,
1 SYSM, DAMP, QDD, QDD, STF, CSAVE, NPRINT, KSYS, KPRINT,
2 DP, Q, QD, QDD, QDD, ALFA, BETA, CONFIG(1,2), F,
3 ICORD, NCLEAR, AMP, XZERO, DTPM, DT, ISYS, NC, AMPW,
4 W2, KSYSM1, KSYSM2, IDP, QSAV, QDSAV, QDDSAV, IDIM, IQ,
5 COLM, COLC, IGU, SCALE)
6 LAYER, C, WRITE(NUNIT,5003)
IF(NPRINT .EQ. YES) WRITE(NUNIT,5003)
*****
** PROGRAM PARAMETERS ARE NOW CALCULATED **
** ESTABLISH PLOT PACKAGE **
*****
C 450 CALL FIXFOR(CSAVE, F)
GO TO 465

463 DO 464 I = 1, KSYS
FSAV(I) = DABS(F(I))
464 CONTINUE
465 IF(DEGR .EQ. 0.0) CALL DEBUG(2,Q,QD,QDD,F, KSYS, ISYS, DP, NDEG,
1 NPRINT, DEGR, C)
DEGR = WT*DEGOPI
NDEGKP = NDEG
NDEG = DEGR
C IF( NDEG .EQ. NDEGKP ) GO TO 360

```


SDS-9300 COMPUTER PROGRAM LISTING

This listing is coded in SDS FORTRAN IV, (Ref. 5-7), to run on a Scientific Data System Mode 9300 computer interfaced with an ADAGE Graphics Display Terminal, Model 10 (AGT-10). Due to the limited core storage of this computer, overlaying techniques have been employed.

000000490
 000000500
 000000510
 000000520
 000000530
 000000540
 000000550
 000000560
 000000570
 000000580
 000000590
 000000600
 000000610
 000000620
 000000630
 000000640
 000000650
 000000660
 000000670
 000000680
 000000690
 000000700
 000000710
 000000720
 000000730
 000000740
 000000750
 000000760
 000000770
 000000780
 000000790
 000000800
 000000810
 000000820
 000000830
 000000840
 000000850
 000000860
 000000870
 000000880
 000000890
 000000900
 000000910
 000000920
 000000930
 000000940
 000000950
 000000960

```

RAD = PI/180.0
SF1 = 1.0
SF2 = 1.0
SF3 = 1.0
SF4 = 1.0
SF1A = 15.0
SF4A = 15.0
WTH1=WTH2=WTH3=WTH4=0.4
AXLTH1 = 7.9
AXLTH2 = 3.4
AXLTH3 = 5.1
AXLTH4 = 3.3
XA = 0.5
XB = 3.9
XC = 10.2
XD = 0.5
XE = 1.0
XF = 0.5
XG = 3.9
XH = 9.0
XP = 11.2
XS = 16.8
CALL INITIAL(CONFIG, IRRAY1, IRRAY2, IRRAY3, IRRAY4, IRRAY5, IRRAY6, IRRAY7, IRRAY9,
1 IRRAY10, ISPRING)
2 CALL READ(NTIME, I104, I105, XZERO, AMP, RPM, WN, ISEAT, PIN28,
1 CONFIG, DIFF)
1 THETA = 0.0
RADIUS = 1.5
DTHETA = 4.0*RAD
IF(SENSE SWITCH 2) 2 CHANGES ; INPUT(101)
2 OUTPUT(101) ; INPUT(101)
2 CONTINUE
OUTPUT(6) ; SPRCTR CALLED ;
CALL SPRCTR(IRRAY3, ISEAT, DIFF)
CALL MIDBAR(IRRAY2, IRRAY9)
DO 1400 I = 6, 97
DTHETA = THETA + DTHETA
RADCCOS(I) = RADIUS*COS(THETA)
RADSIN(I) = RADIUS*SIN(THETA)
CONTINUE
AMPO2 = AMP/2.0
FREQ = (RPM * PI) / (WN * 30.0)
ENCODE(40, 102, I116)
ENCODE(32, 103, I113)
ENCODE(20, 1450, I111)
ENCODE(8, 1455, I110)
ENCODE(28, 1430, I109)

```


00000970
 00000980
 00000990
 00001000
 00001010
 00001020
 00001030
 00001040
 00001050
 00001060
 00001070
 00001080
 00001090
 00001100
 00001110
 00001120
 00001130
 00001140
 00001150
 00001160
 00001170
 00001180
 00001190
 00001200
 00001210
 00001220
 00001230
 00001240
 00001250
 00001260
 00001270
 00001280
 00001290
 00001300
 00001310
 00001320
 00001330
 00001340
 00001350
 00001360
 00001370
 00001380
 00001390
 00001400
 00001410
 00001420
 00001430
 00001440

```

ENCODE(32, 1431, I104)
ENCODE(28, 1432, I101)
ENCODE(12, 1433, I105)
ENCODE(20, 1434, I117)
ENCODE(28, 1460, I108)
CALL TEXTO(IDEV, I108, 7, 24, 30, 2, 3, IER)
IF(IER.NE.0) OUTPUT(101) IER
CALL TEXTO(IDEV, I113, 8, 38, 71, 2, 3, IER)
IF(IER.NE.0) OUTPUT(101) IER
CALL TEXTO(IDEV, I116, 10, 40, 65, 2, 3, IER)
IF(IER.NE.0) OUTPUT(101) IER
CALL TEXTO(IDEV, I110, 2, 40, 44, 2, 3, IER)
IF(IER.NE.0) OUTPUT(101) IER
CALL TEXTO(IDEV, I111, 5, 38, 36, 2, 3, IER)
IF(IER.NE.0) OUTPUT(101) IER
N = 2000; I = 0
DO 1500 K = 1, NTIME
  I = I + 1
  IF( K .EQ. NTIME) OUTPUT(6)
  N = N + 1
  CALL DISPLAY(CONFIG, K, I105, I109, I117, I101, I104)
  CALL OUTPUT(6), CAM CALLED
  IF(SENSE SWITCH 3) 1551
  CALL CAM( IRRAY1, RADIUS, AMP, I, RADCOS, RAD SIN)
  ROTATE = IFIX( I115) ROTATE
  ENCODE(4, I1550, NULL, 1, 40, 36, 2, 3, IER)
  CALL TEXTO(IDEV, I101, IER)
  IF(IER.NE.0) OUTPUT(101) IER
  CALL TEXTO(IDEV, I115, 1, 40, 36, 2, 3, IER)
  IF(IER.NE.0) OUTPUT(101) IER
  CALL BAR1( I, IRRAY4, SF1, SF1A, PIN28)
  CALL BAR2( I, IRRAY5, SF2, IRRAY9, PIN28)
  CALL BAR3( I, IRRAY6, SF3, IRRAY10)
  CALL BAR4( I, IRRAY7, SF4, SF4A)
  IF( I .NE. 1 ) GO TO 175
  X(1) = Y(1) = 0.0
  X(2) = Y(2) = YC
  X(3) = X(5) = XH - 0.6
  X(4) = X(6) = XH + 0.6
  X(8) = XH
  DO 50 J = 1, 5
    Y(J+2) = FLOAT(J)*(YH-YC)/6.0 + YC
  CONTINUE
  Y(8) = YH
  X(9) = Y(9) = 0.0
  ISPRNG(1) = IHEAD(0, 10)
  DO 100 J = 1, 9
    YZERO(J) = Y(J)
  
```


00001450
 00001460
 00001470
 00001480
 00001490
 00001500
 00001510
 00001520
 00001530
 00001540
 00001550
 00001560
 00001570
 00001580
 00001590
 00001600
 00001610
 00001620
 00001630
 00001640
 00001650
 00001660
 00001670
 00001680
 00001690
 00001700
 00001710
 00001720
 00001730
 00001740
 00001750
 00001760

```

X(J) = GDTOCM( X(J) ) ; Y(J) = GDTOCM( Y(J) )
CONTINUE
ISPRNG(2) = IPACK( X(2), Y(2), 0)
DO 150 J = 3,8
ISPRNG(J) = IPACK( X(J), Y(J), 1 )
CONTINUE
ISPRNG(9) = 0
CALL GRAPHQ(IDEV, ISPRNG, 9, 8, IER)
IF( IER.NE.0) OUTPUT(101) IER
IF( I .EQ. 1 ) GO TO 1990
CONTINUE
DO 200 J = 3, 8
Y(J) = CMTGDI( Y(J) )
Y(J) = YZERO(J) + FLOAT(J-2) * Q4( I)*SF4/ 6.0
Y(J) = GDTOCM( Y(J) )
ISPRNG(J) = IPACK( X(J), Y(J), 1 )
CONTINUE
CALL GRAPHQ(IDEV, ISPRNG, 9, 8, IER)
IF( IER.NE.0) OUTPUT(101) IER
CONTINUE
IF( SENSE SWITCH 1 ) 1, 1991
CONTINUE
IF( SENSE SWITCH 2 ) 2, 1992
CONTINUE
IF( SENSE SWITCH 6 ) 6, 2000
CALL TEXTIR(IDEV, NULL, 1, 39, 46, 2, 3, IER)
IF( IER.NE.0) OUTPUT(101) IER GO TO 1998
IF(MOD(ITDIR(1), 8).EQ.0)
CONTINUE
I = 1 ; N = 1
GO TO 1500
END
  
```



```

550 CONTINUE
IF(PIN28.EQ.YES) Y(3) = Y(30) = YZER03 + Q2(JA)*SF1 + YP
IF(PIN28.EQ.NO) Y(3) = Y(30) = YZER03 + Q8(JA)*SF1 + YP
Y(2) = Y(3) + 0.2
Y(31) = Y(2)
Y(14) = Y(15) = Y(18) = Y(19) = YZER14 + Q1(JA)*SF1
Y(16) = Y(17) = Y(14) - WTH1
DIFF = ABS( Y(3) - Y(14) ) - AXLTH1
DO 560 I = 4, 13
Y(I) = Y(33-I) = Y(3) - ( FLOAT(I-3) * ( Y(3) - Y(14) ) / 11.0 )
CONTINUE
DO 575 I = 4, 8
AMP(I) = DIFF*SIN( FLOAT(I-3) * PI/11.0 ) * SF1A
IF( AMP(I) .GT. WTH1/2.0 ) AMP(I) = WTH1/2.0
FORMAT( 'I, 3X, 'I, 13, 'AMP=', 'E15.5' )
X(I) = X(17-I) = X(3) + AMP(I)
X(33-I) = X(I+16) = X(I) + WTH1 - 2.0*AMP(I)
CONTINUE
DO 600 I = 2, 31
X(I) = GDTOCM( X(I) ) ; Y(I) = GDTOCM( Y(I) )
LINK1(I) = IPACK( X(I), Y(I), 1 )
CONTINUE
LINK1(2) = IPACK( X(2), Y(2), 0 )
LINK1(32) = 0
CALL GRAPHO( IDEV, LINK1, 32, 4, IER )
IF( IER.NE.0 ) OUTPUT(101), IER, 'BAR1-2,
RETURN
END

```

```

00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630
00000640
00000650
00000660
00000670
00000680
00000690
00000700
00000710
00000720
00000730
00000740

```



```

SUBROUTINE BAR2(JA, LINK2, SF2, IRRAY9, PIN28)
INTEGER PIN28, YES
DIMENSION LINK2(24), X(24), Y(24), PHI(10,3), YZERO(24)
COMMON AXLTH1, AXLTH2, AXLTH3, AXLTH4
COMMON WTH1, WTH2, WTH3, WTH4
COMMON PI, RAD, DEG, IDEV
COMMON IGD, IR(20), Y, NO/4HNO /
COMMON XA, YA, XB, YB, XC, YC, XD, YD, XE, YE, XF, YF, XG, YG, XH, YH, XS, YS, XP, YP
COMMON Q1(180), Q2(180), Q3(180), Q4(180), Q5(180), Q6(180), Q7(180)
COMMON Q8(180), Q9(180), OMEGA(180)
COMMON XCAM, YCAM
DATA YES/4HYES /, NO/4HNO /
GDTOCM(A) = 0.2000 * A - 1.0000
CMTOGD(A) = A / 0.2000 + 5.000
NAMELIST IER
L = 1
K = 2
IF( JA .NE. 1 ) GO TO 200
X(1) = Y(1) = 0.0
X(24) = XF ; Y(24) = YF
FACT2 = Y(24) - WTH2/2.0
FACT3 = Y(24) + WTH2/2.0
DO 10 I = 2, 12
Y(I) = YZERO(I) = FACT2
Y(I+11) = FACT3
10 CONTINUE
I = 1, 11
X(13-I) = X(12+I) = X(24) + FLOAT(I-1) * AXLTH2 / 10.0
25 CONTINUE
XA = X(12) ; YA = Y(12)
IF(PIN28 .EQ. NO) ; GO TO 26
IF( SENSE SWITCH 5 ) 5, 26
5 CALL JOINT( 37, IRRAY9, 0.1, 10.0, K, JA)
26 CONTINUE
AL22 = AXLTH2**2
AL23 = AXLTH2**3
DO 30 I = 3, 12
XX = X(I) - X(24)
XX2 = XX**2
XX3 = XX**3
PHI(I-2, 1) = (AL23 - 3.0*XX2*AXLTH2 + 2.0*XX3) * SF2/AL23
PHI(I-2, 3-L) = (XX*AL22 - 2.0*XX2*AXLTH2 + XX3) * SF2/AL22
PHI(I-2, 4-L) = -(XX2*AXLTH2 - XX3) * SF2/AL22
30 CONTINUE
I = 1, 23
DO 50 I = 1, 3X, I = 1, I3, 2E15.4)
99 FORMAT(

```



```

X(I) = GDTOCM( X(I) ) ; Y(I) = GDTOCM( Y(I) )
CONTINUE
LINK2(1) = IHEAD(0,10)
LINK2(2) = IPACK( X(2), Y(2), 0 )
DO 100 I = 3, 23
LINK2(I) = IPACK( X(I), Y(I), 1 )
CONTINUE
LINK2(24) = 0
CALL GRAPHO(IDEV, LINK2,24,5,IER)
IF( IER.NE.0 ) OUTPUT(101) IER, 'BAR2-1'
IF( JA.EQ. 1 ) RETURN
CONTINUE
DO 300 I = 3, 12
Y(I) = CMTOGD( Y(I) )
Y(25-I) = CMTOGD( Y(25-I) )
W = PHI(I-2,1)*Q2(JA) + PHI(I-2,3-L)*Q3(JA) + PHI(I-2,4-L)*Q6(JA)
Y(I) = W + YZERO(I)
Y(25-I) = WTH2 + Y(I)
CONTINUE
X(12) = CMTOGD( X(12) )
XA = X(12)
IF( PIN28.EQ. NO) YA = Y(12)
IF( SENSE SWITCH 5 ) GO TO 326
CALL JOINT( 37, IARRAY, 0.1, 10.0, K, JA)
CONTINUE
X(12) = GDTOCM( X(12) )
DO 350 I = 3, 12
Y(I) = GDTOCM( Y(I) )
Y(25-I) = GDTOCM( Y(25-I) )
CONTINUE
DO 400 I = 3, 23
LINK2(I) = IPACK( X(I), Y(I), 1 )
CONTINUE
CALL GRAPHO(IDEV, LINK2,24,5,IER)
IF( IER.NE.0 ) OUTPUT(101) IER, 'BAR2-2'
RETURN
END

```

```

00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630
00000640
00000650
00000660
00000670
00000680
00000690
00000700
00000710
00000720
00000730
00000740
00000750
00000760
00000770
00000780
00000790
00000800
00000810
00000820
00000830

```



```

SUBROUTINE RAR3(JA, LINK3, SF3, IRRAY10)
DIMENSION LINK3(36), X(35), Y(35), PHI(16,3), YZERO(35)
COMMON AXLTH1, AXLTH2, AXLTH3, AXLTH4
COMMON WTH1, WTH2, WTH3, WTH4
COMMON PI, RAD, DEG, IDEV
COMMON IGDIR(20), ITDIR(40)
COMMON XA, YA, XB, YB, XC, YC, XD, YD, XE, YE, XF, YF, XG, YG, XH, YH, XS, YS, XP, YP
COMMON Q1(180), Q2(180), Q3(180), Q4(180), Q5(180), Q6(180), Q7(180)
COMMON Q8(180), Q9(180), OMEGA(180)
GDTOCM(A) = 0.2000*A - 1.0000
CMTOGD(A) = A / 0.2000 + 5.000
NAMELIST IER
L = 1
K = 3
IF( JA .NE. 1 ) GO TO 200
X(1) = Y(1) = 0.0
X(2) = XG ; Y(2) = YG
DO 10 I = 2, 18
Y(I) = YZERO(I) = Y(2)
Y(I+17) = Y(2) + WTH3
CONTINUE I = 2, 18
DO 25 X(I) = X(37-I) = X(2) + FLOAT(I-2) * AXLTH3 / 16.0
CONTINUE
XA = X(18) ; YA = Y(18)
IF( SENSE SWITCH 5 ) 5, 26
CALL JOINT( 37, IRRAY10, 0.1, 10.0, K, JA )
CONTINUE
AL32 = AXLTH3**2
AL33 = AXLTH3**3
DO 30 I = 3, 18
XX = X(I) - X(2)
XX2 = XX**2
XX3 = XX**3
PHI(I-2, 2-L) = ( + 3.0*XX2*AXLTH3 - 2.0*XX3 ) * SF3/AL33
PHI(I-2, 3-L) = ( XX*AL32 - 2.0*XX2*AXLTH3 + XX3 ) * SF3/AL32
PHI(I-2, 4-L) = -( XX2*AXLTH3 - XX3 ) * SF3/AL32
CONTINUE I = 1, 35
DO 50 X(I) = GTOCM( X(I) ) ; Y(I) = GTOCM( Y(I) )
CONTINUE
LINK3(1) = IHEAD(0, 10)
LINK3(2) = IPACK( X(2), Y(2), 0 )
DO 100 I = 3, 35
LINK3(I) = IPACK( X(I), Y(I), 1 )

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460

```



```

100 CONTINUE(36) = 0
LINK3(36) = 0
CALL GRAPHO(IDEV, LINK3,36,6, IER)
IF(IER.NE.0) OUTPUT(101) IER, 'BAR3-1'
IF(JA.EQ.1) RETURN
200 CONTINUE
DO 300 I = 3, 18
Y(I) = CMTOGD(Y(I), )
Y(37-I) = CMTOGD(Y(37-I), )
W = PHI(I-2,2-L)*Q4(JA) + PHI(I-2,3-L)*Q6(JA) + PHI(I-2,4-L)*Q5(JA)
Y(I) = W + YZERO(I)
Y(37-I) = WTH3 + Y(I)
300 CONTINUE
X(18) = CMTOGD(X(18), )
XA = X(18) ; YA = Y(18)
IF(SENSE SWITCH 5) 325,326
325 CALL JOINT(37, IRRAY10,0.1,10.0, K, JA)
326 CONTINUE
X(18) = GDTOCM(X(18), )
DO 350 I = 3, 18
Y(I) = GDTOCM(Y(I), )
Y(37-I) = GDTOCM(Y(37-I), )
350 CONTINUE
DO 400 I = 3, 35
LINK3(I) = IPACK(X(I), Y(I), 1)
400 CONTINUE
CALL GRAPHO(IDEV, LINK3,36,6, IER)
IF(IER.NE.0) OUTPUT(101) IER, 'BAR3-2'
RETURN
END

```

```

00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630
00000640
00000650
00000660
00000670
00000680
00000690
00000700
00000710
00000720
00000730
00000740
00000750
00000760

```



```

SUBROUTINE BAR4(JA,LINK4,SF4,SF4A)
DIMENSION LINK4(24),X(23),Y(23),AMP(7)
COMMON AXLTH1,AXLTH2,AXLTH3,AXLTH4
COMMON WTH1,WTH2,WTH3,WTH4
COMMON PI,RAD,DEG,IDEV
COMMON IDIR(20),IDIR(40)
COMMON XA,YA,XB,YB,XC,YC,XD,YD,XE,YE,XF,YF,XG,YG,XH,YH,XS,YS,XP,YP
COMMON Q1(180),Q2(180),Q3(180),Q4(180),Q5(180),Q6(180),Q7(180)
COMMON Q8(180),Q9(180),OMEGA(180)
COMMON XCAM,YCAM
GDTOCM(A) = 0.2000*A - 1.0000
CMTOGD(A) = A / 0.2000 + 5.000
NAMELIST IER
LINK4(1) = IHEAD(0,10)
IF( JA.NE.1 ) GO TO 500
X(1) = Y(1) ; Y(2) = YH
X(2) = X(2) ; Y(23) = Y(2)
X(23) = X(2) - WTH4/2.0
X(3) = X(2) - WTH4/2.0
Y(3) = Y(2) - WTH4/2.0
YZER03 = Y(3)
DO 25 I = 3, 11
X(I) = X(3)
X(25-I) = X(3) + WTH4
Y(I) = Y(25-I) = Y(3) - AXLTH4 * FLOAT(I-3) / 8.0
CONTINUE
X(12) = X(11) - 1.2
X(13) = X(14) + 1.2
Y(12) = Y(13) = Y(11) - 0.3
YZER11 = Y(11)
DO 100 I = 2, 23
X(I) = GDTOCM( X(I) ) ; Y(I) = GDTOCM( Y(I) )
CONTINUE
DO 150 I = 2, 23
IF( I.EQ.2 ) LINK4(2) = IPACK( X(2), Y(2), 0 ) ; GO TO 150
LINK4(I) = IPACK( X(I), Y(I), 1 )
CONTINUE
LINK4(24) = 0
CALL GRAPHO(IDEV,LINK4,24,7,IER)
IF( IER.NE.0 ) OUTPUT(101) IER,'BAR4-1'
DO 275 I = 1,22
CONTINUE
IF( JA.EQ.1 ) RETURN
DO 550 I = 2,23
X(I) = CMTOGD( X(I) ) ; Y(I) = CMTOGD( Y(I) )
CONTINUE

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460

```


00000470
 00000480
 00000490
 00000500
 00000510
 00000520
 00000530
 00000540
 00000550
 00000560
 00000570
 00000580
 00000590
 00000600
 00000610
 00000620
 00000630
 00000640
 00000650
 00000660
 00000670
 00000680
 00000690
 00000700
 00000710

```

Y(3) = Y(22) = YZER03 + Q4(JA)*SF4
Y(2) = Y(3) + 0.2
Y(23) = Y(2)
Y(11) = Y(14) = YZER11 - Q7(JA)*SF4
Y(12) = Y(13) = Y(11) - 0.3
DIFF = ABS( Y(3) - Y(11) ) - AXLTH4
DO 560 I = 4, 10
  Y(I) = Y(25-I) = Y(3) - ( FLOAT(I-3) * ( Y(3)-Y(11) ) / 8.0 )
CONTINUE
DO 575 I = 4, 7
  AMP(I) = DIFF*SIN( FLOAT(I-3) * PI/8.0 ) * SF4A
  IF( AMP(I) .GT. WTH4/2.0 ) AMP(I) = WTH4/2.0
  X(I) = X(14-I) = X(3) + AMP(I)
  X(25-I) = X(I+11) = X(I) + WTH4 - 2.0 * AMP(I)
CONTINUE
DO 600 I = 2, 23
  X(I) = GDTOCM( X(I) ) ; Y(I) = GDTOCM( Y(I) )
  LINK4(I) = IPACK( X(I), Y(I), 1 )
CONTINUE
LINK4(2) = IPACK( X(2), Y(2), 0 )
LINK4(24) = 0
CALL GRAPHO(IDEV, LINK4, 24, 7, IER)
IF( IER.NE.0 ) OUTPUT(101) IER, 'BAR4-2'
RETURN
END
560
575
600

```



```

SUBROUTINE CAM(NCAM, ICAM, RADIUS, AMP, JA, RADCOS, RADSIN)
DIMENSION ICAM(NCAM), X(97), Y(97), RADCOS(97), RADSIN(97)
COMMON AXLTH1, AXLTH2, AXLTH3, AXLTH4
COMMON WTH1, WTH2, WTH3, WTH4
COMMON PI, RAD, DEG, IDEV
COMMON IGDIR(20), ITDIR(40)
COMMON XA, YA, XB, YB, XC, YC, XD, YD, XE, YE, XF, YF, XG, YG, XH, YH, XS, YS, XP, YP
COMMON Q1(180), Q2(180), Q3(180), Q4(180), Q5(180), Q6(180), Q7(180)
COMMON Q8(180), Q9(180), OMEGA(180)
COMMON XCAM, YCAM
CMTOGD(A) = A / 0.2000 + 5.000
GDTOCM(A) = 0.2000*A - 1.0000
X(1) = 0.0 ; GO TO 55
IF( JA .NE. 1 ) GO TO 55
ICAM(1) = IHEAD(0,10)
CROSS = 0.3
XCAM = XE - AMP CROSS
X(2) = XCAM + CROSS
X(3) = XCAM - CROSS
XCAM = YD - 0.6 - AXLTH1 - RADIUS + YS
YC(1) = YD + CROSS ; Y(3) = YCAM - CROSS
X(4) = X(5) = Y(4) = Y(5) = 0.0
DO I = 1, 3
  X(I) = GDTOCM( X(I) ) ; Y(I) = GDTOCM( Y(I) )
CONTINUE
XCAM = GDTOCM( XCAM ) ; YCAM = GDTOCM( YCAM )
ICAM(2) = IPACK( X(2), YCAM, 0 )
ICAM(3) = IPACK( X(3), YCAM, 1 )
ICAM(4) = IPACK( XCAM, Y(2), 0 )
ICAM(5) = IPACK( XCAM, Y(3), 1 )
CONTINUE
IF( JA .EQ. 1 ) XCAM = CMTOGD( XCAM ) ; YCAM = CMTOGD( YCAM )
XE = XCAM + COS(OMEGA(JA))
YE = Q9(JA) + YCAM
X(6) = RADIUS + XE ; Y(6) = YE
DO 200 I = 6, NCAM-1
  X(I+1) = RADCOS(I+1) + XE
  Y(I+1) = RADSIN(I+1) + YE
  X(I) = GDTOCM( X(I) ) ; Y(I) = GDTOCM( Y(I) )
  IF( I .EQ. 6 ) ICAM(6) = IPACK( X(6), Y(6), 0 ) ; GO TO 200
ICAM(1) = IPACK( X(1), Y(1), 1 )
CONTINUE
ICAM(NCAM) = 0
CALL GRAPHO(IDEV, ICAM, NCAM, 1, IER)
RETURN
END

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460

```

30

55

200


```

SUBROUTINE DISPLY(CONFIG, K, I105, I109, I117, I101, I104) 00000010
DIMENSION I101(7), I104(8), I105(3), I109(7), I117(5); CONFIG(180) 00000020
INTEGER ROTATE, CONFIG, YES, BLNK 00000030
COMMON AXLTH1, AXLTH2, AXLTH3, AXLTH4 00000040
COMMON WTH1, WTH2, WTH3, WTH4 00000050
COMMON PI, RAD, DEG, IDEV 00000060
COMMON IGDIR(20), IDIR(40) 00000070
COMMON XA, YA, XB, YB, XC, YC, XD, YD, XE, YE, XF, YF, XG, YG, XH, YH, XS, YS, XP, YP 00000080
COMMON Q1(180), Q2(180), Q3(180), Q4(180), Q5(180), Q6(180), Q7(180) 00000090
COMMON Q8(180), Q9(180), OMEGA(180) 00000100
COMMON XCAM, YCAM 00000110
DATA NULL/7777777B/, BLNK/4H /, YES/4HYES /, NO/4HNO / 00000120
NAMELIST IER 00000130
GO TO(1501,1502,1503,1504,1505,1506,1507,1508) CONFIG(K) 00000140
CONTINUE 00000150
CALL TEXTTO(IDEV, NULL, 1,26,30,2,3,IER) 00000160
IF(IER.NE.0) OUTPUT(101) IER 00000170
CALL TEXTTO(IDEV, I105, 3,26,30,2,3,IER) 00000180
IF(IER.NE.0) OUTPUT(101) IER 00000190
CALL TEXTTO(IDEV, NULL, 1,28,30,2,3,IER) 00000200
IF(IER.NE.0) OUTPUT(101) IER 00000210
CALL TEXTTO(IDEV, NULL, 1,30,30,2,3,IER) 00000220
IF(IER.NE.0) OUTPUT(101) IER 00000230
CALL TEXTTO(IDEV, NULL, 1,32,30,2,3,IER) 00000240
IF(IER.NE.0) OUTPUT(101) IER 00000250
GO TO 1509 00000260
CONTINUE 00000270
CALL TEXTTO(IDEV, NULL, 1,26,30,2,3,IER) 00000280
IF(IER.NE.0) OUTPUT(101) IER 00000290
CALL TEXTTO(IDEV, I109, 3,26,30,2,3,IER) 00000300
IF(IER.NE.0) OUTPUT(101) IER 00000310
CALL TEXTTO(IDEV, NULL, 1,28,30,2,3,IER) 00000320
IF(IER.NE.0) OUTPUT(101) IER 00000330
CALL TEXTTO(IDEV, NULL, 1,30,30,2,3,IER) 00000340
IF(IER.NE.0) OUTPUT(101) IER 00000350
CALL TEXTTO(IDEV, I117, 5,32,30,2,3,IER) 00000360
IF(IER.NE.0) OUTPUT(101) IER 00000370
GO TO 1509 00000380
CONTINUE 00000390
CALL TEXTTO(IDEV, NULL, 1,26,30,2,3,IER) 00000400
IF(IER.NE.0) OUTPUT(101) IER 00000410
CALL TEXTTO(IDEV, I109, 3,26,30,2,3,IER) 00000420
IF(IER.NE.0) OUTPUT(101) IER 00000430
CALL TEXTTO(IDEV, I101, 7,28,30,2,3,IER) 00000440
IF(IER.NE.0) OUTPUT(101) IER 00000450
CALL TEXTTO(IDEV, NULL, 1,30,30,2,3,IER) 00000460

```

1501

1502

1503

00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630
00000640
00000650
00000660
00000670
00000680
00000690
00000700
00000710
00000720
00000730
00000740
00000750
00000760
00000770
00000780
00000790
00000800
00000810
00000820
00000830
00000840
00000850
00000860
00000870
00000880
00000890
00000900
00000910
00000920
00000930
00000940

```
IF(IER.NE.0) OUTPUT(101) IER  
CALL TEXTO(IDEV, NULL, 1, 32, 30, 2, 3, IER)  
GO TO 1509  
CONTINUE  
CALL TEXTO(IDEV, NULL, 1, 26, 30, 2, 3, IER)  
IF(IER.NE.0) OUTPUT(101) IER  
CALL TEXTO(IDEV, I109, 3, 26, 30, 2, 3, IER)  
IF(IER.NE.0) OUTPUT(101) IER  
CALL TEXTO(IDEV, I101, 7, 28, 30, 2, 3, IER)  
IF(IER.NE.0) OUTPUT(101) IER  
CALL TEXTO(IDEV, NULL, 1, 30, 30, 2, 3, IER)  
IF(IER.NE.0) OUTPUT(101) IER  
CALL TEXTO(IDEV, I117, 5, 32, 30, 2, 3, IER)  
IF(IER.NE.0) OUTPUT(101) IER  
GO TO 1509
```

1504

```
CONTINUE  
CALL TEXTO(IDEV, NULL, 1, 26, 30, 2, 3, IER)  
IF(IER.NE.0) OUTPUT(101) IER  
CALL TEXTO(IDEV, I109, 3, 26, 30, 2, 3, IER)  
IF(IER.NE.0) OUTPUT(101) IER  
CALL TEXTO(IDEV, NULL, 1, 28, 30, 2, 3, IER)  
IF(IER.NE.0) OUTPUT(101) IER  
CALL TEXTO(IDEV, I104, 8, 30, 30, 2, 3, IER)  
IF(IER.NE.0) OUTPUT(101) IER  
CALL TEXTO(IDEV, NULL, 1, 32, 30, 2, 3, IER)  
IF(IER.NE.0) OUTPUT(101) IER  
GO TO 1509
```

1505

```
CONTINUE  
CALL TEXTO(IDEV, NULL, 1, 26, 30, 2, 3, IER)  
IF(IER.NE.0) OUTPUT(101) IER  
CALL TEXTO(IDEV, I109, 3, 26, 30, 2, 3, IER)  
IF(IER.NE.0) OUTPUT(101) IER  
CALL TEXTO(IDEV, NULL, 1, 28, 30, 2, 3, IER)  
IF(IER.NE.0) OUTPUT(101) IER  
CALL TEXTO(IDEV, I104, 8, 30, 30, 2, 3, IER)  
IF(IER.NE.0) OUTPUT(101) IER  
CALL TEXTO(IDEV, I117, 5, 32, 30, 2, 3, IER)  
IF(IER.NE.0) OUTPUT(101) IER  
GO TO 1509
```

1506

```
CONTINUE  
CALL TEXTO(IDEV, NULL, 1, 26, 30, 2, 3, IER)  
IF(IER.NE.0) OUTPUT(101) IER  
CALL TEXTO(IDEV, I109, 3, 26, 30, 2, 3, IER)  
IF(IER.NE.0) OUTPUT(101) IER  
CALL TEXTO(IDEV, I101, 7, 28, 30, 2, 3, IER)  
IF(IER.NE.0) OUTPUT(101) IER  
CALL TEXTO(IDEV, I104, 8, 30, 30, 2, 3, IER)  
IF(IER.NE.0) OUTPUT(101) IER  
CALL TEXTO(IDEV, I117, 5, 32, 30, 2, 3, IER)  
IF(IER.NE.0) OUTPUT(101) IER  
GO TO 1509
```

1507


```

1508 CALL TEXTTO(IDEV, NULL, 1, 32, 30, 2, 3, IER)
      IF(IER.NE.0) OUTPUT(101) IER
      GO TO 1509
      CONTINUE
      CALL TEXTTO(IDEV, NULL, 1, 26, 30, 2, 3, IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTTO(IDEV, I109, 3, 26, 30, 2, 3, IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTTO(IDEV, I101, 7, 28, 30, 2, 3, IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTTO(IDEV, I104, 8, 30, 30, 2, 3, IER)
      IF(IER.NE.0) OUTPUT(101) IER
      CALL TEXTTO(IDEV, I117, 5, 32, 30, 2, 3, IER)
      IF(IER.NE.0) OUTPUT(101) IER
1509 CONTINUE
      RETURN
      END

```

```

00000950
00000960
00000970
00000980
00000990
00001000
00001010
00001020
00001030
00001040
00001050
00001060
00001070
00001080
00001090
00001100
00001110

```


00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630
00000640
00000650
00000660
00000670
00000680
00000690
00000700
00000710
00000720
00000730
00000740
00000750
00000760
00000770
00000780
00000790
00000800
00000810
00000820
00000830
00000840
00000850
00000860
00000870
00000880
00000890
00000900
00000910
00000920
00000930
00000940

```
ENCODE(48,106,1106)  
ENCODE(44,107,1107)  
ENCODE(28,108,1108)  
ENCODE(28,109,1109)  
ENCODE( 8,110,1110)  
ENCODE(20,111,1111)  
ENCODE(36,112,1112)  
ENCODE(32,113,1113)  
ENCODE(44,114,1114)  
ENCODE( 4,115,1115)  
ENCODE(40,116,1116)  
ENCODE(20,117,1117)  
CALL TEXTO(IDEV,1101,7,1,13,3,3,IER)  
IF( IER.NE.0) OUTPUT(101)  
CALL TEXTO(IDEV,1106,12,5,1,2,3,IER)  
IF( IER.NE.0) OUTPUT(101)  
CALL TEXTO(IDEV,1107,11,8,6,2,3,IER)  
IF( IER.NE.0) OUTPUT(101)  
CALL TEXTO(IDEV,1115,1,14,45,2,3,IER)  
IF( IER.NE.0) OUTPUT(101)  
CALL TEXTO(IDEV,1108,7,17,25,2,3,IER)  
IF( IER.NE.0) OUTPUT(101)  
CALL TEXTO(IDEV,1109,7,19,37,1,3,IER)  
IF( IER.NE.0) OUTPUT(101)  
CALL TEXTO(IDEV,1110,2,24,43,2,3,IER)  
IF( IER.NE.0) OUTPUT(101)  
CALL TEXTO(IDEV,1111,5,27,29,2,3,IER)  
IF( IER.NE.0) OUTPUT(101)  
CALL TEXTO(IDEV,1112,9,29,31,1,3,IER)  
IF( IER.NE.0) OUTPUT(101)  
CALL TEXTO(IDEV,1113,8,30,33,1,3,IER)  
IF( IER.NE.0) OUTPUT(101)  
CALL TEXTO(IDEV,1114,11,33,30,1,3,IER)  
IF( IER.NE.0) OUTPUT(101)  
CALL TEXTO(IDEV,1116,10,34,33,1,3,IER)  
IF( IER.NE.0) OUTPUT(101)  
CALL TEXTO(IDEV,1117,5,11,27,2,3,IER)  
IF( IER.NE.0) OUTPUT(101)  
PAUSE TO READ TITLE PAGE  
CALL TEXTR(IDEV,NULL,1,39,46,2,3,IER)  
IF( IER.NE.0) OUTPUT(101)  
IF(MOD(ITDIR(1),8).EQ.0) GO TO 12  
CALL TEXTO(IDEV,NULL,1,39,46,2,3,IER)  
IF( IER.NE.0) OUTPUT(101)  
DO 20 JA = 1,97  
IRRAY1(JA) = 0.0  
DO 21 JA = 1,18  
IRRAY2(JA) = 0.0
```

12
20
21

C


```

SUBROUTINE JOINT(NCIR, ICIR, RADIUS, ANGLE, K, JA)
DIMENSION ICIR(NCIR), X(75), Y(75), RADCOS(75), RADSIN(75)
COMMON AXLTH1, AXLTH2, AXLTH3, AXLTH4
COMMON WTH1, WTH2, WTH3, WTH4
COMMON PI, RAD, DEG, IDEV
COMMON IGDIR(20), ITDIR(40)
COMMON XA, YA, XB, YB, XC, YC, XD, YD, XE, YE, XF, YF, XG, YG, XH, YH, XS, YS, XP, YP
COMMON Q1(180), Q2(180), Q3(180), Q4(180), Q5(180), Q6(180), Q7(180)
COMMON Q8(180), Q9(180), OMEGA(180)
COMMON XCAM, YCAM
GDTOCM(A) = 0.2000*A - 1.0000
NAMELIST IER
IF( JA .NE. 1 ) GO TO 35
ICIR(1) = IHEAD(1,10)
X(1)=Y(1)=0.0
THETA = 0.0
DTHETA = ANGLE * RAD
DO 25 I=3, NCIR
THETA = THETA + DTHETA
RADCOS(I) = RADIUS*COS(THETA)
RADSIN(I) = RADIUS*SIN(THETA)
CONTINUE
X(2) = RADIUS + XA ; Y(2) = YA
DO 100 I = 3, NCIR
X(I) = RADCOS(I) + XA
Y(I) = RADSIN(I) + YA
CONTINUE
OUTPUT(6) , JOINT CALLED
DO 200 I = 2, NCIR
WRITE(6, 99) I, X(I), Y(I)
FORMAT( , , 3X, 'I=', I3, 2E15.4)
X(I) = GDTOCM( X(I) )
Y(I) = GDTOCM( Y(I) )
IF(I.EQ.2) ICIR(2)=IPACK(X(2),Y(2),0); GO TO 200
ICIR(I)=IPACK(X(I),Y(I),1)
CONTINUE
CALL GRAPHO(IDEV, ICIR, NCIR, 7+K, IER)
IF(IER.NE.0) OUTPUT(101) IER
RETURN
END

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400

```

```

25
35
100
X
X
X
99
200

```



```

SUBROUTINE MIDBAR(MIDCIR, IRRAY9)
DIMENSION MIDCIR(18), X(18), Y(18), IRRAY9(49)
COMMON AXLTH1, AXLTH2, AXLTH3, AXLTH4
COMMON WTH1, WTH2, WTH3, WTH4
COMMON PI, RAD, DEG, IDEV
COMMON IGDIR(20), IDIR(40)
COMMON XA, YA, XB, YB, XC, YC, XD, YD, XE, YE, XF, YF, XG, YG, XH, YH, XS, YS, XP, YP
COMMON Q1(180), Q2(180), Q3(180), Q4(180), Q5(180), Q6(180), Q7(180)
COMMON Q8(180), Q9(180), OMEGA(180)
COMMON XCAM, YCAM
NAMELIST IER
GDTOCM(A) = 0.2000*A - 1.0000
MIDCIR(1) = IHEAD(0,10)
X(1)=Y(1)=0.0
ANGLE = 9.0 ; K = 6 ; JA = 1
RADIUS = 0.25 ; YA = YB
XA = XB ; YC = YD ; Y(3) = YB - .5
CALL JOINT(49, IRRAY9, RADIUS, ANGLE, K, JA)
DO 100 I = 1, 49
  IRRAY9(I) = 0
CONTINUE
X( 2) = XB + .5 ; Y( 2) = YB
X( 3) = XB ; Y( 3) = Y( 4) = Y( 3)
X( 4) = X( 3) - 1.0 ; Y( 4) = Y( 2)
X( 5) = X( 2) ; Y( 5) = Y( 2)
X( 6) = X( 3) ; Y( 6) = Y( 3)
DO 150 I = 1, 11
  X( 6 + I) = 1, 11
  X( 6 + I) = X( 5 + I) - .1
CONTINUE
DO 160 I = 7, 17, 2
  Y(I) = Y( 6) - .1
  Y(I+1) = Y( 6)
CONTINUE
DO 200 I = 2, 17
  X(I) = GDTOCM( X(I) ) ; Y(I) = GDTOCM( Y(I) )
  IF(I.EQ.2) MIDCIR(I)=IPACK(X(I),Y(I),0); GO TO 200
MIDCIR(I) = IPACK(X(I),Y(I),1)
CONTINUE
DO 250 I = 8, 18, 2
  MIDCIR(I) = IPACK(X(I),Y(I),0)
CONTINUE
MIDCIR( 18) = 0
CALL GRAPHO(IDEV, MIDCIR, 18, 2, IER)
RETURN
END

```

X

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460


```

00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630
00000640
00000650
00000660
00000670
00000680
00000690
00000700
00000710
00000720
00000730
00000740
00000750
00000760
00000770
00000780
00000790
00000800
00000810
00000820
00000830
00000840
00000850
00000860
00000870
00000880
00000890
00000900
00000910
00000920
00000930
00000940

```

```

15 IF(IER.NE.0) OUTPUT(101) IER
IF(MOD(I,8).EQ.0) GO TO 15
CALL TEXT(I,IDEV,IMODA,1,39,46,IER)
IF(IER.NE.0) OUTPUT(101) IER
IPRINT = NO
IF(IMODA.EQ. YES ) IPRINT = YES
ERASE QUESTION
CALL TEXT(IDEV, NULL, 1, 39, 46, 2, 3, IER)
IF(IER.NE.0) OUTPUT(101) IER
CALL TEXT(IDEV, NULL, 1, 39, 23, 2, 3, IER)
IF(IER.NE.0) OUTPUT(101) IER
CALL TEXT(IDEV, NULL, 1, 37, 17, 2, 3, IER)
IF(IER.NE.0) OUTPUT(101) IER
READ(4,30) DIFF
WRITE(6,5017) DIFF
DO 1000 I = 1, NTIME
READ(4,30) Q1(I), Q2(I), Q3(I), Q4(I), Q5(I)
READ(4,31) Q6(I), Q7(I), Q8(I), Q9(I), OMEGA(I), CONFIG(I)
CONTINUE
IF(IPRINT.EQ. NO ) GO TO 1250
WRITE(6,45)
DO 1200 I = 1, NTIME
NDEG = IFIX( OMEGA(I)*DEG )
WRITE(6,40) Q8(I), OMEGA(I), CONFIG(I)
1 CONTINUE
CONTINUE
CALL TITLE PAGE, NULL, 1, 5, 1, 2, 3, IER)
IF(IER.NE.0) OUTPUT(101) IER
CALL TEXT(IDEV, NULL, 1, 8, 6, 2, 3, IER)
IF(IER.NE.0) OUTPUT(101) IER
CALL TEXT(IDEV, NULL, 1, 4, 45, 2, 3, IER)
IF(IER.NE.0) OUTPUT(101) IER
CALL TEXT(IDEV, NULL, 1, 17, 25, 2, 3, IER)
IF(IER.NE.0) OUTPUT(101) IER
CALL TEXT(IDEV, NULL, 1, 19, 37, 1, 3, IER)
IF(IER.NE.0) OUTPUT(101) IER
CALL TEXT(IDEV, NULL, 1, 24, 43, 2, 3, IER)
IF(IER.NE.0) OUTPUT(101) IER
CALL TEXT(IDEV, NULL, 1, 27, 29, 2, 3, IER)
IF(IER.NE.0) OUTPUT(101) IER
CALL TEXT(IDEV, NULL, 1, 29, 31, 1, 3, IER)
IF(IER.NE.0) OUTPUT(101) IER
CALL TEXT(IDEV, NULL, 1, 30, 33, 1, 3, IER)
IF(IER.NE.0) OUTPUT(101) IER
CALL TEXT(IDEV, NULL, 1, 33, 30, 1, 3, IER)
IF(IER.NE.0) OUTPUT(101) IER

```


00000950
00000960
00000970
00000980
00000990
00001000

```
CALL TEXTIO(IDEV, NULL, 1, 34, 33, 1, 3, IER)  
IF(IER.NE.0) OUTPUT(101) IER  
CALL TEXTIO(IDEV, NULL, 1, 11, 27, 2, 3, IER)  
IF(IER.NE.0) OUTPUT(101) IER  
RETURN  
END
```



```

SUBROUTINE SPRCTR( ISPRNG, ISEAT, DIFF)
DIMENSION X(13), Y(13), ISPRNG(13)
INTEGER YES
COMMON AXLTH1, AXLTH2, AXLTH3, AXLTH4
COMMON WTH1, WTH2, WTH3, WTH4
COMMON PI, RAD, DEG, IDEV
COMMON IGD, DIR(20), IDIR(40)
COMMON XA, YA, XB, YB, XC, YC, XD, YD, XE, YE, XF, YF, XG, YG, XH, YH, XS, YS, XP, YP
COMMON Q1(180), Q2(180), Q3(180), Q4(180), Q5(180), Q6(180), Q7(180)
COMMON Q8(180), Q9(180), OMEGA(180)
COMMON XCAM, YCAM
DATA YES/4HYES /, NO/4HND /
NAMELIST IER
GDTOCM(A) = 0.2000*A - 1.0000
ISPRNG(1) = IHEAD(0,10)
X(1) = Y(1) = 0.0
X(2) = XC ; Y(2) = YC -.3
X(3) = XC ; Y(3) = YC +.3
DO 100 I = 1, 7
Y(3+I) = Y(2+I) - .1*(-1.0)**(I+2)
X(I+2) = X(I+1)
CONTINUE
DO 175 I = 1, 10
X(I) = GDTOCM( X(I) ) ; Y(I) = GDTOCM( Y(I) )
CONTINUE
DO 200 I = 2, 10
IF( I .EQ. 2) ISPRNG(I) = IPACK( X(I), Y(I), 0) ; GO TO 200
ISPRNG(I) = IPACK( X(I), Y(I), 1)
CONTINUE
DO 250 I = 5, 9, 2
ISPRNG(I) = IPACK( X(I), Y(I), 0 )
CONTINUE
ISPRNG(11) = 0
CALL GRAPHO( IDEV, ISPRNG, 11, 3, IER)
IF( IER.NE.0) OUTPUT(101) IER, SPRCTR,
-IF( ISEAT .EQ. NO) DIFF = 1.2
K = 0
SIGN = +1.0
X(12) = XS
Y(2) = YH - WTH4/2.0 - AXLTH4 + DIFF + YP
Y(3) = Y(2) - 0.4
Y(4) = Y(2)
DO 275 I = 4, 10, 2
Y(I+1) = Y(I+2) = Y(I) - .1
CONTINUE
K = K + 1

```


00000470
 00000480
 00000490
 00000500
 00000510
 00000520
 00000530
 00000540
 00000550
 00000560
 00000570
 00000580
 00000590
 00000600
 00000610
 00000620
 00000630
 00000640
 00000650
 00000660
 00000670
 00000680
 00000690
 00000700
 00000710
 00000720
 00000730
 00000740
 00000750
 00000760
 00000770
 00000780
 00000790
 00000800
 00000810

```

DO 350 I = 1,7,2 + SIGN*0.3
X(I2-I) = X(I3-I) + SIGN*0.1
X(I1-I) = X(I2-I) + SIGN*0.1
CONTINUE
X(3) = X(11) + 0.3*SIGN
X(2) = X(4) + 0.5
X(I3) = Y(I3) = 0.5
DO 375 I = 1,13
X(I) = GDTOCM( X(I) )
IF( K .EQ. 1 ) Y(I) = GDTOCM( Y(I) )
CONTINUE
ISPRNG(2) = IPACK( X(2), Y(2), 0 )
ISPRNG(3) = IPACK( X(3), Y(3), 1 )
ISPRNG(4) = IPACK( X(4), Y(4), 0 )
ISPRNG(5) = IPACK( X(2), Y(2), 1 )
DO 400 I = 6,12,2
ISPRNG(I) = IPACK( X(I-1), Y(I-1), 0 )
ISPRNG(I+1) = IPACK( X(I), Y(I), 1 )
CONTINUE
ISPRNG(13) = 0
CALL GRAPHO(IDEV, ISPRNG, 13,10+K, IER)
IF(IER.NE.0) OUTPUT(101) IER
X(I2) = XP
SIGN = -1.0
IF( K .EQ. 1 ) GO TO 300
RETURN
DO 600 I = 2, 13
ISPRNG(I) = 0
CONTINUE
CALL GRAPHO(IDEV, ISPRNG, 13,11, IER)
IF(IER.NE.0) OUTPUT(101) IER
CALL GRAPHO(IDEV, ISPRNG, 13,12, IER)
IF(IER.NE.0) OUTPUT(101) IER
RETURN
END

```

350

375

400

500

600

X

X

X

LIST OF REFERENCES

1. Winfrey, R. C., Dynamics of Mechanism With Elastic Links, Research and Development Division, Aerospace Group, Hughes Aircraft Co., Report No. JT 5180, March 1969.
2. Anderson, R. V., Dynamic Analysis of Cams, M. S., ME Thesis U.S. Naval Postgraduate School, 1971.
3. Thomson, W. T., Vibration Theory and Applications, 1st ed., Prentice-Hall, Inc., New Jersey 1965.
4. Newmark, N. M., "A Method of Computation for Structural Dynamics," ASCE, Journal of Engineering Mechanics Division, v. 85, No. EM3, p. 67-94, July 1959.
5. Scientific Data Systems, SDS- FORTRAN IV REFERENCE MANUAL, SDS, Santa Monica, California, 1966.
6. Scientific Data Systems, SDS-9300 Computer Reference Manual, SDS, Santa Monica, California, 1966.
7. Scientific Data Systems, SDS Real Time Monitor Reference Manual, SDS, Santa Monica, California, 1966.
8. Rubinstein, M. F., Matrix Computer Analysis of Structures, 1st ed., Prentice-Hall, Inc., 1965.
9. Hurty, W. C., and Rubinstein, M. F., Dynamics of Structures, 1st ed., Prentice-Hall, Inc., 1966.
10. International Business Machines Corporation Form C28-6515-7, IBM System/360 FORTRAN IV Language, 1968.
11. International Business Machines Corporation Form H20, 0205-3, IBM System/360 Scientific Subroutine Package (360-CM-03X) Version III Programers Manual, 1968.
12. IBM System/360 Operating System, FORTRAN IV (H) Programmer's Guide, Form C-28-6602-1, IBM Corp., Poughkeepsie, N.Y.
13. W. R. Church Computer Center, Users Manual, 1st ed., Rev. 6, March 1970.
14. Mitchell, D. B., "Cam Follower Dynamics", Machine Design, v. 22, p. 151-154, June 1950.
15. Rothbart, H. A., Cams, 1st ed., John Wiley & Sons, Inc., 1956.

16. Stoddart, D. A., "Polydyne Cam Design", Machine Design, v. 25, p. 121-135, January 1953.
17. Molian, S., "Use of Algebraic Polynomials in Design of Cams", The Engineer, v. 215, p. 352-354, 22 February, 1963.
18. Barkan, P., and McGarrity, R. V., "A Spring-Actuated, Cam-Follower System: Design Theory and Experimental Results", Journal of Engineering for Industry, v. 87, p. 279-286, August 1965.
19. Gniska, C. W., and Bowden, P. K., Dynamic Analysis of a Four Bar Mechanism Utilizing Interactive Computer Graphics, paper presented for M.E. Laboratory Project (ME-3400), Dept. of Mech. Eng., U.S. Naval Postgraduate School, June 1971.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3. Director, Computer Facility, Code 0211 Naval Postgraduate School Monterey, California 93940	1
4. Chairman, Department of Mechanical Engineering Code 59 Naval Postgraduate School Monterey, California 93940	1
5. Professor David Salinas, Code 59Zc Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	3
6. Dr. Richard C. Winfrey Naval Civil Engineering Laboratory Port Hueneme, California	3
7. Supervisor, Computer Laboratory Code Ec Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
8. Charles W. Gnilka, LT, USN Philadelphia Naval Shipyard Philadelphia, Pennsylvania 19112	5

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

ORIGINATING ACTIVITY (Corporate author)

Naval Postgraduate School
Monterey, California

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

REPORT TITLE

Modeling and Analysis of an Elastic Mechanism With Clearances

DESCRIPTIVE NOTES (Type of report and, inclusive dates)

Master's Thesis; December 1971

AUTHOR(S) (First name, middle initial, last name)

Charles W. Gnilka

REPORT DATE

December 1971

7a. TOTAL NO. OF PAGES

342

7b. NO. OF REFS

19

CONTRACT OR GRANT NO.

9a. ORIGINATOR'S REPORT NUMBER(S)

SUBJECT NO.

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

Naval Postgraduate School
Monterey, California 93940

ABSTRACT

The classical approach for the analysis of linkages has been to assume rigid elements during operation. As the operational speeds of mechanisms increase, this assumption creates larger errors in the analysis. Thus a better model of a machine that considers elastic deformations will aid in more efficient design as well as provide improved accuracy and performance.

This thesis presents a simulation for the dynamic response of an elastic link model with three clearances. The simulation has been developed for an automobile, cam actuated valve train. The computer program is coded in FORTRAN IV, for an IBM-360 computer and is included as a portion of this work. The capability to visually observe the dynamic action of the model is included by a graphic display routine. This routine is implemented in SDS FORTRAN IV for a SDS-9300 computer interfaced with an ADAGE Graphics Display Terminal, Model 10.

Sample problems for various cam speeds of interest are included utilizing graphs and photos from the graphic display.

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Elastic Mechanism Hydraulic Actuated Valve Train Computer Graphics Clearance Dynamic Modeling Computer Simulation						

BINDERY

Thesis
G522 Gnilka
c.1

133468

The modeling and ana-
lysis of an elastic mec-
hanism with clearances.

BINDERY

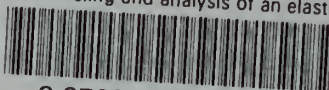
Thesis
G522 Gnilka
c.1

133468

The modeling and ana-
lysis of an elastic mec-
hanism with clearances.

thesG522

The modeling and analysis of an elastic



3 2768 002 02960 5

DUDLEY KNOX LIBRARY