

UNCLASSIFIED

AD NUMBER

ADB043376

LIMITATION CHANGES

TO:

Approved for public release; distribution is unlimited.

FROM:

Distribution authorized to U.S. Gov't. agencies only; Test and Evaluation; JUN 1979. Other requests shall be referred to Space and Missile Systems Organization, Los Angeles, CA.

AUTHORITY

SAMSO ltr 31 Mar 1980

THIS PAGE IS UNCLASSIFIED

THIS REPORT HAS BEEN DELIMITED
AND CLEARED FOR PUBLIC RELEASE
UNDER DOD DIRECTIVE 5200.20 AND
NO RESTRICTIONS ARE IMPOSED UPON
ITS USE AND DISCLOSURE.

DISTRIBUTION STATEMENT A

APPROVED FOR PUBLIC RELEASE,
DISTRIBUTION UNLIMITED.

**BEST
AVAILABLE COPY**

LEVEL

PH

ADB043376

R-1267

FINAL TECHNICAL REPORT
ILS³ SIMULATION OVERVIEW AND
PRELIMINARY USERS GUIDE

by

S. Serben
F. Satlow

June 1979

Sponsored by
Advanced Research Projects Agency
ARPA Order No. 3223

DD
RECEIVED
JAN 18
1980

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Projects Agency of the U.S. Government.

DDC FILE COPY



The Charles Stark Draper Laboratory, Inc.

Cambridge, Massachusetts 02139

Distribution limited to U.S. Government agencies only. Test and Evaluation (June 1979). Other requests for this document must be referred to SAMSO/YAD.

P.O. Box 92960, World
Postal Center, L.A., CA

80

1

14

022

UNCLASSIFIED

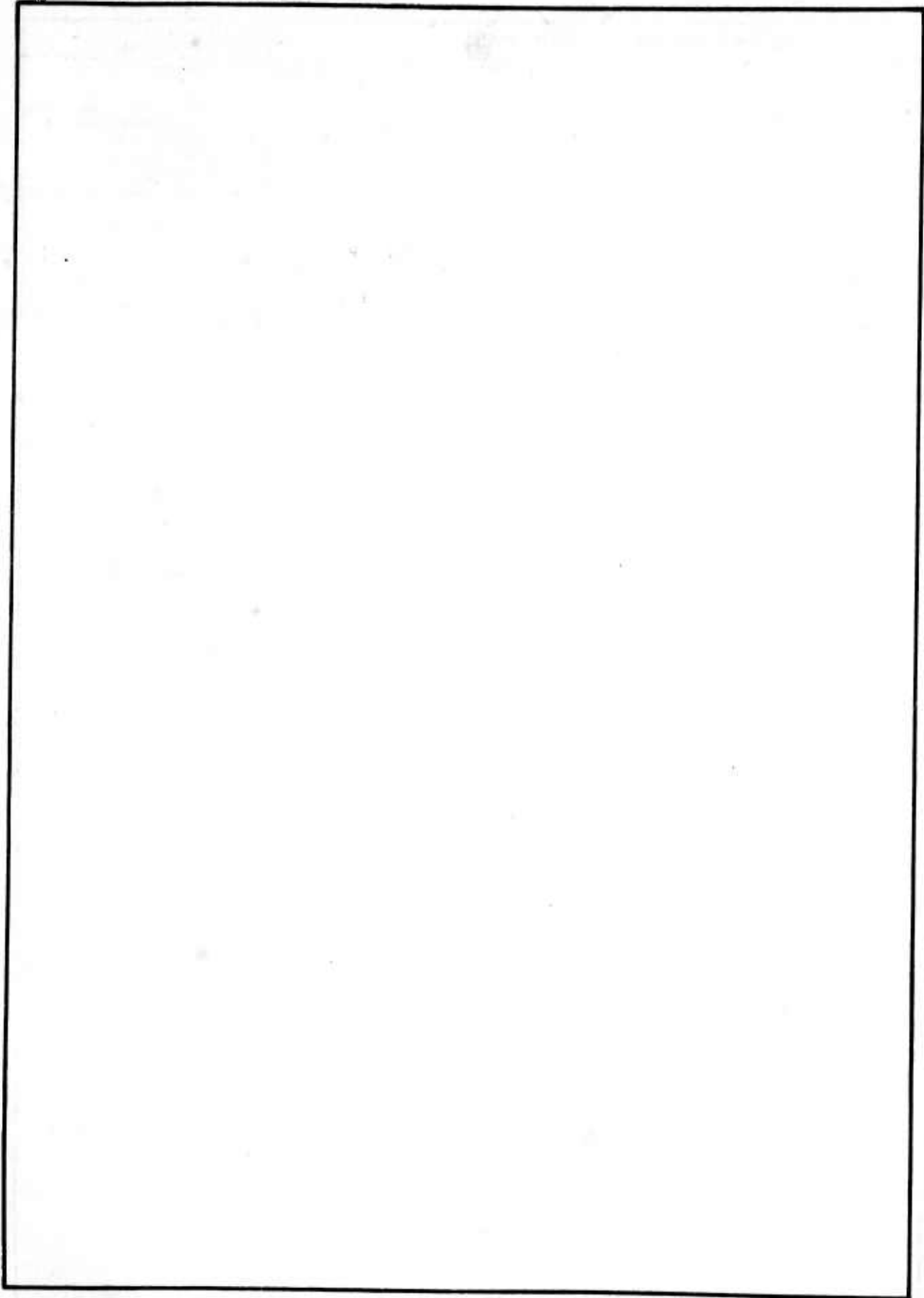
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) <u>ILS</u> Simulation Overview and Preliminary Users Guide.		5. TYPE OF REPORT & PERIOD COVERED Final Jan - Dec 78.
7. AUTHOR(S) Serben, F. / Satlow		6. PERFORMING ORG. REPORT NUMBER R-1267
9. PERFORMING ORGANIZATION NAME AND ADDRESS The Charles Stark Draper Laboratory, Inc. Cambridge, Massachusetts 02139		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS ARPA Proj. 3223 Program Code No. 7E20
11. CONTROLLING OFFICE NAME AND ADDRESS Advanced Research Projects Agency Arlington, Virginia 22209		12. REPORT DATE Jun 79
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) HQ SAMSO (YAD) P.O. Box 92960 Worldway Postal Center Los Angeles, California 90009		13. NUMBER OF PAGES 107
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
16. DISTRIBUTION STATEMENT (of this Report) Distribution limited to U.S. Government agencies only. Test and Evaluation (June 1979). Other requests for this document must be referred to SAMSO/YAD.		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) 12 115		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) aberrations modes disturbance actuators state-vector truncation deformations sensors FFT data base nodes PSF		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A system software architecture and accompanying algorithms are developed to aid in the analysis of large flexible satellites and their sensors, System features and constraints as well as data base definition are included. A sample problem to illustrate the use of the system for a simplified model of a typical large space structure is presented.		

7812F 376

alt

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)



SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

R-1267

FINAL TECHNICAL REPORT
ILS³ SIMULATION Overview and
Preliminary Users Guide

Approved: *D. G. Hoag* 12/26/79
D. G. Hoag
Head, Advanced Systems Department

ARPA Order No.-3223
Program Code No.-7E20
Name of Contractor-The Charles Stark Draper Laboratory, Inc.
Contract No.-F04701-76-C-0178
Date of Contract-22 October 1976
Contract Expiration Date-31 December 1978
Principal Investigator-Dr. Keto Soosaar(617)258-2575
Short Title of Work-LAS Evaluation

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Projects Agency of the U.S. Government.

80 1 14 022

ACKNOWLEDGMENT

The work reported herein was performed by the Charles Stark Draper Laboratory, Inc. (CSDL). This research was supported by the Advanced Research Projects Agency of the Department of Defense, and was monitored by Deputy for Advanced Space Programs, Space and Missile Systems Organization, under Contract No. F04701-76-C-0178. This report covers the time period from January, 1978 to December 31, 1978. The various technical advisors of the program at DARPA have been Lt. Col. W. Cuneo, Dr. A. Pike, Dr. J. Jenney; and Capt. T. Ernst of SAMSO.

The simulation was developed within the Large Space Structures Division of the CSDL Advanced Systems Department headed by Mr. David Hoag. Dr. Keto Soosaar, Division Leader of the Large Space Structures Division, was the Program Manager. Mr. Saul Serben has been responsible for the architecture and software development.

Key organizers to the simulation formulation were Dr. V. Mahajan in optics and the late Dr. J. Canavin in the dynamics and control. Other contributors were the following:

Ms. F. Satlow	Software Architecture, Code Generation
Mr. S. Anagnostis	Structure, Quasi-Static, Steady State
Mr. F. Ayer	Quasi-Static
Mr. J. Govignon	Optics
Mr. T. Henderson	Structure, Control, Plotting
Dr. L. Matson	System Consultant

The authors would also like to thank Dr. M. Balas, control analyst, Mr. R. Grin, structure analyst, Mr. R. Therrien, project administrator, Dr. R. Wilkinson, attitude control analyst, and Dr. W. Koenigsberg, Fast Fourier Transform consultant.

We also gratefully acknowledge the assistance of the following students: R. Bickford and C. Hogan.

Publication of this report does not constitute approval by ARPA or the U.S. Air Force, of the findings and conclusions contained herein. It is published for the exchange and stimulation of ideas.

TABLE OF CONTENTS

<u>Section</u>	<u>Page</u>
1. Introduction.....	1
2. System Overview.....	2
3. Source Code Editor.....	6
3.1 Purpose.....	6
3.2 Input Definitions For Each Subsystem.....	6
3.2.1 DYNAMICS.....	6
3.2.2 STEADY STATE.....	8
3.2.3 QUASI-STATIC.....	9
3.2.4 INTERFACE.....	10
3.2.5 OPTICS.....	11
4. DYNAMICS.....	13
4.1 Purpose.....	13
4.2 Technical Summary.....	13
4.2.1 Plant Equations of Motion.....	15
4.2.2 Structure Control System.....	17
4.2.3 Attitude Control.....	18
4.3 Features And Constraints.....	18
4.4 Subsystem Architecture.....	19
4.5 Subroutine Nesting Hierarchy.....	23
4.6 Subroutine Summaries.....	25
4.7 User Input Specifications.....	27
4.8 External Stored Data Bases.....	32
4.9 Internal Stored Data Bases.....	34
4.10 Glossary.....	35
5. STEADY STATE.....	39
5.1 Purpose.....	39
5.2 Technical Summary.....	39
5.3 Features And Constraints.....	40
5.4 Subsystem Architecture.....	40
5.5 Subroutine Nesting Hierarchy.....	43
5.6 Subroutine Summaries.....	43
5.7 User Input Specifications.....	43
5.8 External Stored Data Bases.....	46
5.9 Internal Stored Data Bases.....	48
5.10 Glossary.....	49
6. QUASI-STATIC.....	51
6.1 Purpose.....	51
6.2 Technical Summary.....	51
6.3 Features And Constraints.....	52
6.4 Subsystem Architecture.....	52
6.5 Subroutine Nesting Hierarchy.....	55
6.6 Subroutine Summaries.....	55
6.7 User Input Specifications.....	56
6.8 External Stored Data Bases.....	57
6.9 Internal Stored Data Bases.....	59
6.10 Glossary.....	60

Accession For	
NTIS ..	
DDC TAB	<input checked="" type="checkbox"/>
Unannounced	
Justification	
By.....	
Distribution	
Availability Codes	
Dist	Avail and/or special
B	

TABLE OF CONTENTS - CONT'D

<u>Section</u>	<u>Page</u>
7. INTERFACE.....	62
7.1 Purpose.....	62
7.2 Technical Summary.....	62
7.2.1 Dynamic.....	62
7.2.2 Quasi-Static.....	62
7.2.3 Net Wave Aberrations.....	63
7.3 Features And Constraints.....	63
7.4 Subsystem Architecture.....	64
7.5 Subroutine Nesting Hierarchy.....	67
7.6 Subroutine Summaries.....	68
7.7 User Input Specifications.....	69
7.8 External Stored Data Bases.....	70
7.9 Internal Stored Data Bases.....	72
7.10 Plotting Outputs.....	76
7.11 Glossary.....	76
8. OPTICS.....	80
8.1 Purpose.....	80
8.2 Technical Summary.....	80
8.3 Features And Constraints.....	82
8.4 Subsystem Architecture.....	82
8.5 Subroutine Nesting Hierarchy.....	85
8.6 Subroutine Summaries.....	86
8.7 User Input Specifications.....	87
8.8 Internal Stored Data Bases.....	89
8.9 Plotting Outputs.....	90
8.10 Glossary.....	91
9. Sample Test Case.....	93
9.1 Model Description.....	93

LIST OF ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
2.1	Simulation architecture flow.....	3
2.2	Simulation data flow.....	4
4.1	DYNAMIC technical overview.....	14
4.2	DYNAMIC subsystem architecture.....	21
5.1	STEADY STATE subsystem architecture.....	41
6.1	QUASI-STATIC subsystem architecture.....	53
7.1	INTERFACE subsystem architecture.....	65
8.1	OPTICS subsystem architecture.....	83
9.1	Sample structure model.....	94
9.2	Sample plots.....	100

LIST OF TABLES

<u>Table</u>		<u>Page</u>
2.1	Data base index.....	5
9.1	Sample natural frequencies and mode shapes.....	95
9.2	Sample DYNAMIC local input.....	96
9.3	Sample DYNAMIC local input (continued).....	97
9.4	Sample INTERFACE local input.....	98
9.5	Sample OPTICS local input.....	99
9.6	Sample CPU time and storage requirements.....	101

1. INTRODUCTION

This manual is a preliminary guide to the Integrated Large Space Sensors Simulation (ILS³) computer program. It is intended to illustrate the software methodology of the system as well as to supply information to those persons involved in maintaining and using the system. ILS³ is a tool to aid in the analysis of large flexible satellites and their sensors with special emphasis on the structures, dynamics, optics and control disciplines associated with the performance. It couples the appropriate disciplines shown in Figure 2.1 into an interdisciplinary integrated analysis to reveal the critical interactive effects of design features. The program is modular in design, with the major elements written in FORTRAN IV for use on an IBM 370 or compatible computer. Double precision computation is used throughout the system. Defaults are provided for key parameters.

2. SYSTEM OVERVIEW

The ILS³ system currently consists of five major modules and a source code editor, which may be run either together or decoupled.

One subsystem is called the DYNAMICS module. This module consists of the dynamic formulations, the dynamic disturbances, and the control of both structure and attitude of the satellite. The output of this subsystem is the transient dynamic deformations. Another module, called the STEADY STATE, determines the steady state of the system in response to a sinusoidal input. These two subsystems are mutually exclusive. A third module, known as the QUASI-STATIC accounts for slowly varying loads, resulting in residual thermal deformations. The data essential to the execution of all of the above mentioned modules comes from a finite element mathematical model of the entire structure. The fourth subsystem, the INTERFACE module is used for merging and interpolating the output of the above modules with optical data from the raytrace program and performing coordinate system transformations. The fifth subsystem, known as the Deformed Optics Evaluator (OPTICS) uses the outputted results from the INTERFACE module to evaluate the optical performance by computing line-of-sight and power on the detector.

The configuration of the system is shown in Figure 2.1, its associated data flow in Figure 2.2, and a description of the data bases in Table 2.1. The configuration may be started and/or stopped at any subsystem along the path.

The source code editor, written in SNOBOL, is used to modify variables having complicated functional source code dimensions prior to compilation time, thereby increasing efficiency and decreasing turnaround time.

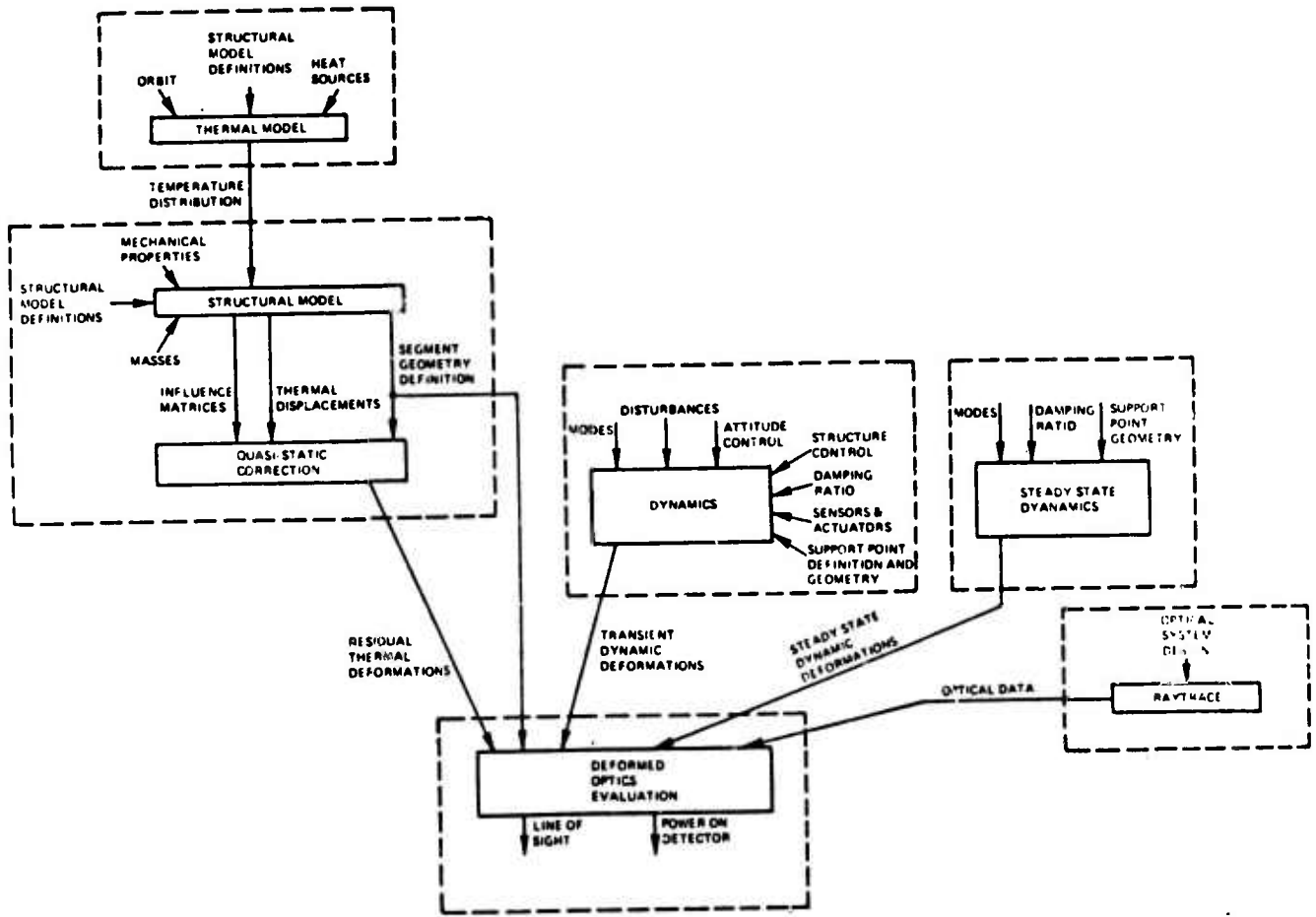


Figure 2.1 Simulation architecture flow

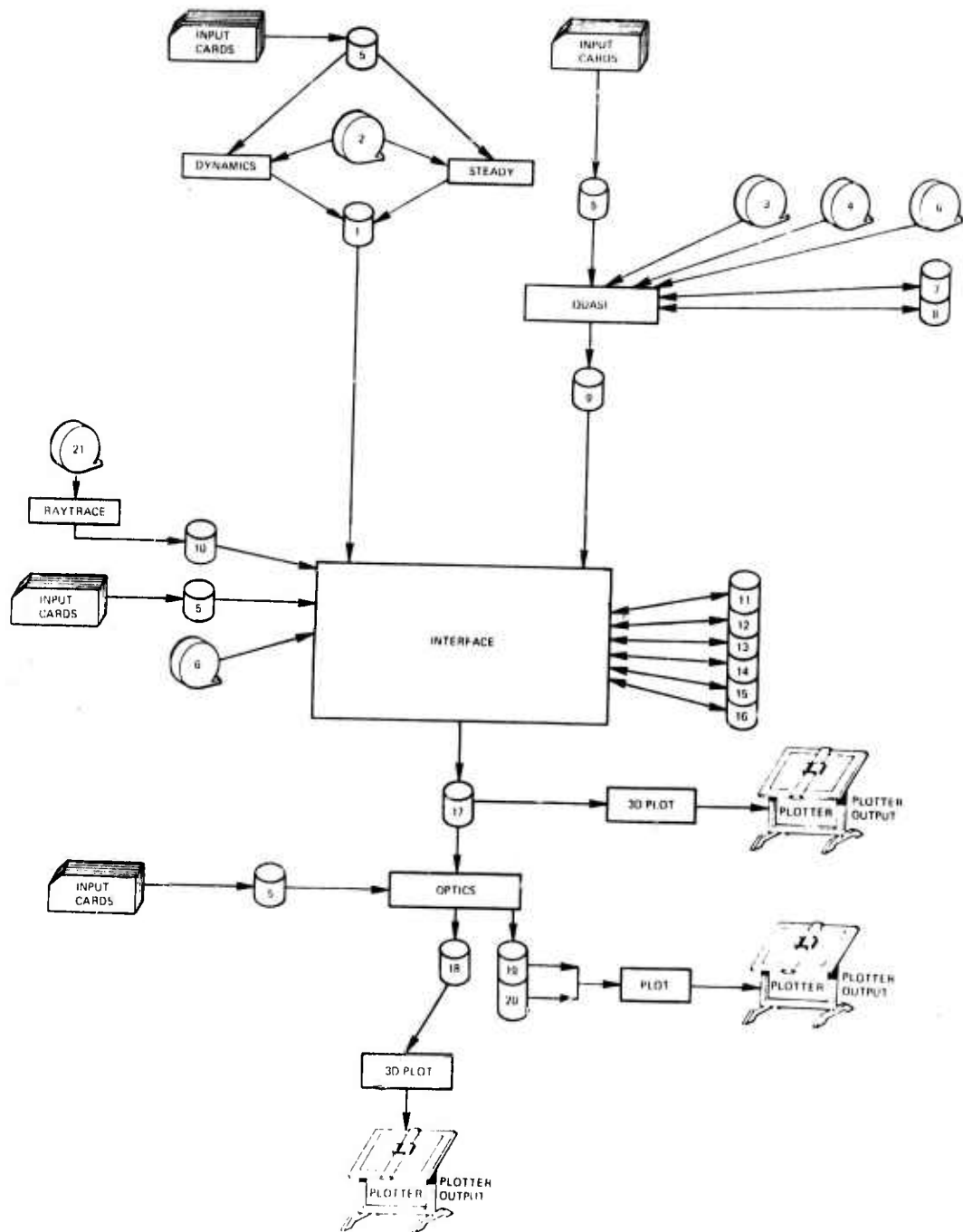


Figure 2.2 Simulation data flow

Table 2.1 Data base index

No.	Contents	Section	Name
1	Dynamic or steady state transient deformations	4.9	IQTAP
		5.9	IQTAP
		7.9	FIL3
2	Mode shapes and frequencies	4.8	ITAPE
		5.8	ITAPE
3	Initial thermal displacements	6.8	FIL1
4	Actuator influence matrices	6.8	FIL5
5	Local input	4.7	IN
		5.7	IN
		6.7	IN
		7.7	IN
		8.7	IN
6	Nodal coordinates on surfaces	6.8	FIL7
		7.8	FIL5
7	Normal vectors	6.9	FIL2
8	Figure control matrices	6.9	FIL6
9	Thermal deformations	6.9	UFIL
		7.9	FIL1
10	Raytrace data of undeformed system	7.8	FIL7
11	Interpolated quasi-static deformations	7.9	FIL2
12	Interpolated dynamic or steady-state deformations	7.9	FIL4
13	Nodal polar Coordinates	7.9	FIL6
14	Polar coordinates of rays	7.9	FIL8
15	Coefficients of interpolation	7.9	FIL9
16	Coefficients of extrapolation	7.9	FIL11
17	Net wave aberrations	7.9	FIL10
		8.8	FIL1
18	Point-spread-function	8.8	FIL2
19	Line-of-sight errors	8.8	FIL3
20	Energy on detector	8.8	FIL4
21	Raytrace input		

3. SOURCE CODE EDITOR

3.1 PURPOSE

This program modifies array sizes and data constants for all subsystems in order to minimize the storage requirements of the system. This is done by editing source code to replace parameters by their desired values. The parameters and their values are read from FT09F001. The first record is a symbol (defined in these programs as an ampersand(&)). The remaining inputs define the parameters and their values, one per record. The value of a parameter may be defined by a constant or a sequence of arithmetic operations (or a function which computes the maximum of two variables, MAX) applied to a constant or previously defined parameter(s). Comment records are allowed in the input stream and are defined by an asterisk in column 1. The 'pseudo source code' to be edited is read on FT08F001. The resulting output on FT10F001 is compilable FORTRAN source code.

The parameters for each of the subsystems and the subroutines where they are used are defined below.

3.2 INPUT DEFINITIONS FOR EACH SUBSYSTEM

3.2.1 DYNAMICS

A. Modules Where Parameters Are Used

<u>MODULE</u>	<u>M</u>	<u>N26</u>	<u>MD2</u>	<u>MD122</u>	<u>N16</u>	<u>MD1</u>	<u>N1</u>	<u>N2</u>	<u>CMAX</u>	<u>N1MD1</u>	<u>N2MD2</u>	<u>P</u>	<u>MD12</u>
ASN	x	x	x	x									
BLOCK	x	x	x	x	x	x	x	x		x	x		
CAINIT	x	x	x	x									
CHEKER	x												
CNTL	x	x	x	x	x	x	x	x	x			x	x
COPA	x	x	x	x									x
COPAC	x		x	x									x
COPO	x	x	x	x	x	x							x
DERIVP	x		x	x									x
ETAQ		x	x	x	x	x							
FTAB	x	x	x	x	x	x							x
INPUT	x	x	x	x	x	x							
PSINWT	x	x	x	x	x	x							x
RKETA	x			x									x
RKSC				x									

<u>MODULE</u>	<u>N2C</u>	<u>NSEGS</u>	<u>MN</u>	<u>NT</u>	<u>NDIS1</u>	<u>NDIS2</u>	<u>N1C</u>	<u>NNORM</u>	<u>CMAXN</u>
ASN									
BLOCK									
CAINIT									
CHEKER									
CNTL	x						x		x
COPA									
COPAC									
COPO						x			
DERIVP									
ETAQ	x	x						x	
FTAB			x	x					
INPUT	x						x		x
PSINWT					x				
RKETA									
RKSC									

B. Parameter Definitions

CMAX: Twice the number of modes to be used for structure control
(2 * EN1C + 2 * EN2C)

CMAXN: The total number of modes to be used for structure control
(EN1C + EN2C)

M: Number of force actuators

MD1: Number of System 1 modes to be used

MD12: The total number of modes in the system
(EMD1 + EMD2)

MD122: Twice the number of modes in the system
(2 * EMD1 + 2 * EMD2)

MD2: Number of System 2 modes to be used

MN: The maximum number of nodes in one system
(MAX(EN16, EN26))

NDIS1: Number of System 1 disturbances

NDIS2: Number of System 2 disturbances

NNORM: The number of normal vectors for each segment
(1 or 3)

NSEGS: Number of segments in the system

NT: Number of entries in table of

N1: System 1 forcing functions
 Number of System 1 degrees-of-freedom
 (6 * ϵ N16)
 N1C: Number of truncated states to be used in precision base
 N1MD1: Number of System 1 degrees-of-freedom times the number of System 1 modes
 (ϵ N1 * ϵ MD1)
 N16: Number of System 1 nodes to be used
 N2: Number of System 2 degrees-of-freedom
 (6 * ϵ N26)
 N2C: Number of truncated states to be used in navigation base
 N2MD2: Number of System 2 degrees-of-freedom times the number of System 2 modes
 (ϵ N2 * ϵ MD2)
 N26: Number of System 2 nodes to be used
 P: Number of sensors

3.2.2 STEADY STATE

A. Modules Where Parameters Are Used

<u>MODULE</u>	<u>NW</u>	<u>N1</u>	<u>MD1</u>	<u>N16</u>	<u>N1MD1</u>	<u>NDIS1</u>	<u>NNORM</u>	<u>NSEGS</u>
MAIN	x							
BLOCK		x	x	x	x			
COO			x	x		x		
COW		x	x					
FUNC		x	x	x			x	x
INPUT	x		x	x		x	x	x

B. Parameter Definitions

MD1: Number of System 1 modes to be used
 NORM3: Flag which indicates if three normal vectors will be inputted for each segment
 NDIS1: Number of System 1 disturbances

NNAM1: Node numbers to be included in System 1
 NNORM: Number of normal vectors per segment (1 or 3)
 NSEGS: Number of segments in the system
 NW: Number of frequencies
 N1: Number of System 1 degrees-of-freedom (6 * &N16)
 N1MD1: Number of System 1 degrees-of-freedom times the number of System 1 modes (&N1 * &MD1)
 N16: Number of System 1 nodes to be used

3.2.3 QUASI-STATIC

A. Modules Where Parameters Are Used

<u>MODULE</u>	<u>NSEGS</u>	<u>NM</u>	<u>NODES</u>	<u>NODES3</u>	<u>FMAX</u>	<u>FMAX2D</u>	<u>NDUM</u>	<u>NT</u>	<u>NI</u>
BLOCK	x	x							
INPUT	x	x							
MAIN	x	x	x	x					
MATCOM			x	x	x	x			
PREPRO	x	x	x				x		
SCALE	x	x	x					x	x
UOUT		x	x					x	x

B. Parameter Definitions

FMAX: Maximum number of force actuators
 FMAX2D: Dimension of an upper triangular matrix of force actuators
 (&FMAX * (&FMAX + 1) / 2)
 NDUM: The amount of space in a common block left over after scratch storage
 (&NODES3 * &FMAX + &FMAX2D - &NNODES3)

NI: Number of time steps or thermal load configurations
 NM: Number of mirrors
 NODES: Maximum number of nodes in a segment
 NODES3: Triple the maximum number of nodes in a segment
 NSEGS: Maximum number of segments on each mirror
 NT: Maximum number of times for scaling

3.2.4 INTERFACE

A. Modules Where Parameters Are Used

<u>MODULE</u>	NM	NSEGS	DEGR	DEGRTH	MAXRT	MAXRTNM	MTMM	NR	NTHETA
BLOCK	x	x	x	x	x	x			
COMP	x	x			x				
DBLIN			x	x					
DYPREP	x	x					x		
GLOBC		x						x	x
GLOBR		x						x	x
INPUT	x	x			x				
INTERP		x						x	x
MAIN	x	x			x		x	x	x
OUTPUT	x				x				
SEGFND		x						x	x

<u>MODULE</u>	NODES	DEGRTH
BLOCK		
COMP		
DBLIN		
DYPREP		
GLOBC		
GLOBR		
INPUT		
INTERP	x	x
MAIN	x	
OUTPUT		
SEGFND		

B. Parameter Definitions

DEGR:	Number of points needed for interpolation in R direction
DEGRTH:	Number of points needed for interpolation in both R and THETA directions
DEGTH:	Number of points needed for interpolation in THETA direction
MAXRT:	Maximum number of rays that will be allowed
MAXRTNM:	Total number of points where each ray hits each mirror ($\&MAXRT * \&NM$)
MTMM:	Scaling factor
NM:	Number of mirrors
NODES:	Maximum number of nodes in a segment on STARDYNE tape
NR:	Maximum Number of different radii in a segment
NSEGS:	Maximum number of segments in each mirror
NTHETA:	Maximum number of different angles in a segment
NX:	Maximum Number of different X-coordinates in a segment
NY:	Maximum Number of different Y-coordinates in a segment

3.2.5 OPTICS

A. Modules Where Parameters Are Used

<u>MODULE</u>	<u>NRT</u>	<u>NR</u>	<u>N</u>	<u>N1</u>	<u>N12</u>	<u>N21</u>
BFOC	x	x				
BLOCK	x	x	x			
FILL	x	x		x	x	
INPUTA	x	x				
MAIN	x	x	x			x
PROC	x	x	x			x

B. Parameter Definitions

N: Size of the detector (must be a power of 2)
NR: Number of radii to be used for integration
NRT: Number of rays
N1: One more than the size of the detector
($\&N + 1$)
N12: Number of grid points in the detector
($\&N1 * \&N1$)
N21: Number of grid spaces in half a row (or
column) of the detector
($\&N / 2 + 1$)

4. DYNAMICS

4.1 PURPOSE

This program computes the dynamic equations of motion open loop. It also provides the capability of closing the loop by having a dynamic attitude control and/or structure control. The output at each time-step is a vector of displacements for each support point having six degrees-of-freedom (three translational and three rotational components). This is converted into a deformation for each support point.

4.2 TECHNICAL SUMMARY

A set of truncated second-order equations of motion in modal coordinates, as defined in Section 4.2.1, are integrated from a given initial time, T_0 , to a given final time, T_F , using a given time-step, DTP . A state vector is produced for each mode at each time-step, which may be transformed to discrete coordinates, if desired, and used to produce a set of transient dynamic deformations.

An option may be exercised to utilize structure and/or attitude control models, defined in Sections 4.2.2 and 4.2.3 respectively, to generate feedback control vectors. This is performed by passing modal state-vectors from the plant dynamics to the control models at desired times, in order to compute sensor information needed by the control estimators.

A technical overview of the program is illustrated in Figure 4.1

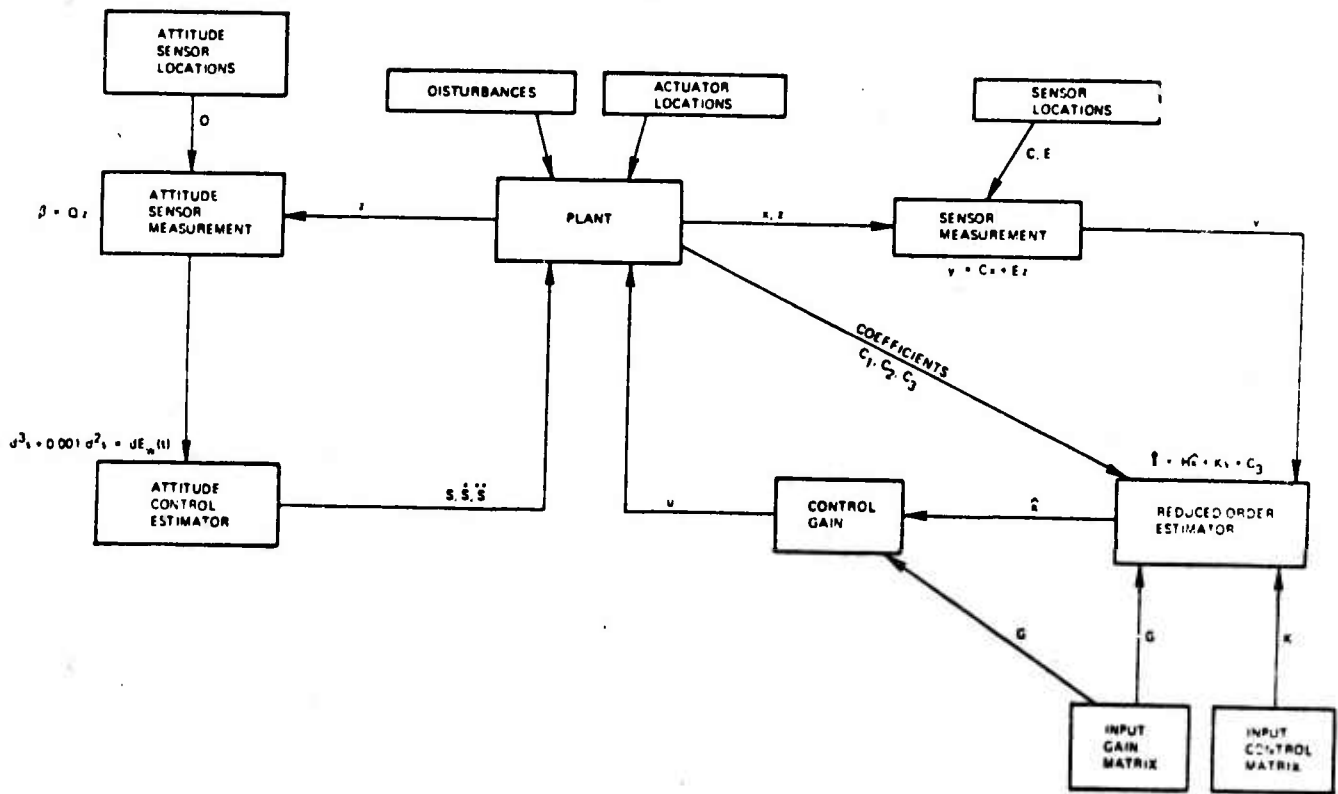


Figure 4.1 Overview

4.2.1 Plant Equations of Motion

$$\begin{bmatrix} \eta_1'' \\ \eta_2'' \end{bmatrix} = \begin{bmatrix} A' & B_1 D \\ B_2 H & F' \end{bmatrix} \begin{bmatrix} \eta_1' \\ \eta_1 \\ \eta_2' \\ \eta_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \begin{bmatrix} U \\ U \end{bmatrix} + \begin{bmatrix} B_1 U_X \\ B_2 U_Z \end{bmatrix}$$

$$\begin{matrix} [(MD1+MD2) \times 1] & [(2MD1+2MD2) \times 1] & [MX1] & [(MD1+MD2) \times 1] \\ & [(MD1+MD2) \times (2MD1+2MD2)] & [(MD1+MD2) \times M] & [(MD1+MD2) \times 1] \end{matrix}$$

where the coefficient matrices and their dimensions are defined by:

$$A' = [-2 \xi_1 \omega_1 \mid -\omega_1^2] \quad (MD1 \times 2MD1)$$

$$B_1 = [-\phi_1 \mid b_1 \text{ Ka}]^T \quad (MD1 \times M)$$

$$D = [0 \mid -d \mid b_2 \mid \phi_2]^T \quad (M \times 2MD2)$$

$$B_2 = [-\phi_2 \mid b_2 \text{ Ka}]^T \quad (MD2 \times M)$$

$$H = [0 \mid d \mid b_1 \mid \phi_1]^T \quad (M \times 2MD1)$$

$$F' = [-2 \xi_2 \omega_2 \mid -\phi_2 \mid g \phi_2 \mid -\omega_2^2]^T \quad (MD2 \times 2MD2)$$

$$B_1 U_X = [\phi_1 \mid F D_1]^T \quad (MD1 \times 1)$$

$$B_2 U_Z = [\phi_2 \mid F D_2]^T + [\phi_2 \mid b_R \text{ [IR] } \{s''\}]^T \quad (MD2 \times 1)$$

b_1 is a $N1 \times 3M$ System 1 force actuator influence matrix denoted by a 6×3 matrix of zeros for each actuator at each node. For each actuator an identity matrix is inserted at the translational coordinates of the node to which it is connected. Each actuator may be connected to only one System 1 node. The activated nodes are defined by the vector $A1$.

b_2 is a $N2 \times 3M$ System 2 force actuator influence matrix denoted by a 6×3 matrix of zeros for each actuator at each node. For each actuator an identity matrix is

inserted at the translational coordinates of the node to which it is connected. Each actuator may be connected to only one System 2 node. The activated nodes are defined by the vector A2.

d is a Mx3M block diagonal force actuator nominal distance matrix defined as follows:

$$\begin{bmatrix} DJ_{1,1} & DJ_{2,1} & DJ_{3,1} & 0 & 0 & 0 \\ 0 & 0 & 0 & DJ_{1,m} & DJ_{2,m} & DJ_{3,m} \end{bmatrix}$$

Ka is a 3MxM block diagonal force actuator matrix defined as follows:

$$\begin{bmatrix} K_1 [DJ_{1,1} \quad DJ_{2,1} \quad DJ_{3,1}] & 0 & 0 & 0 \\ 0 & 0 & 0 & K_m [DJ_{1,m} \quad DJ_{2,m} \quad DJ_{3,m}] \end{bmatrix}$$

br is a N2x3 rotor influence matrix of zeros except for a 3x3 identity matrix inserted at the rotational coordinates of the node specified by NR.

$$g = -br [IRs']^T \quad (N2 \times N2) \text{ Gyroscopic coupling matrix (block diagonal)}$$

IR is a 3x3 resultant rotor inertia matrix composed as follows:

$$\begin{bmatrix} IRIN_{1,1} * CI_1 & IRIN_{1,2} * CI_2 & IRIN_{1,3} * CI_3 \\ IRIN_{1,1} * CI_1 & IRIN_{1,2} * CI_2 & IRIN_{1,3} * CI_3 \\ IRIN_{1,1} * CI_1 & IRIN_{1,2} * CI_2 & IRIN_{1,3} * CI_3 \end{bmatrix}$$

[IRs'] is a 3x3 skew symmetric rotor body coupling matrix.

$$[IR] \{s'\} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad [IRs'] = \begin{bmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{bmatrix}$$

the state vectors outputted from the plant can be truncated for use by the sensor computation.

Estimator (Low-Order System Model):

$$H = C1' + \frac{C2' G}{[(2N1C+2N2C)X(2N1C+2N2C)] [(2N1C+2N2C)XM]} - \frac{K [C | E]}{[(2N1C+2N2C)XP]}$$

$$\frac{[(2N1C+2N2C)X(2N1C+2N2C)] [MX(2N1C+2N2C)]}{[(2N1C+2N2C)X(2N1C+2N2C)] [(2N1C+2N2C)XP]}$$

$$X' = H X + \frac{K y}{[(2N1C+2N2C)X1]} \frac{C3'}{[(2N1C+2N2C)X1]}$$

$$\frac{[(2N1C+2N2C)X(2N1C+2N2C)] [(2N1C+2N2C)XP]}{[(2N1C+2N2C)X(2N1C+2N2C)] [(2N1C+2N2C)XP]}$$

This is a set of first-order differential equations, the initial conditions of which are all zero.

Control Gain:

$$U = \frac{G}{[(2N1C+2N2C)X1]} X$$

$$[(2N1C+2N2C)X1] [MX(2N1C+2N2C)] [(2N1C+2N2C)X1]$$

4.2.3 Attitude Control

IMU output:

$$\begin{bmatrix} \beta_{IMU}' \\ \beta_{IMU} \end{bmatrix} = \begin{bmatrix} bR & 0 \\ 0 & bR \end{bmatrix} \begin{bmatrix} \phi_2 & 0 \\ 0 & \phi_2 \end{bmatrix} \begin{bmatrix} \eta_2' \\ \eta_2 \end{bmatrix}$$

$$[6X1] \quad [6X2N2] \quad [2N2X2N2MD2] \quad [2MD2X1]$$

$$.0033 \, dEg(t)/dt + Eg(t) = \beta_{IMU}$$

$$d^4 Ei(t)/dt^4 + 2.613 \, d^3 Ei(t)/dt^3 + 3.414 \, d^2 Ei(t)/dt^2 + 2.613 \, Ei(t)/dt + Ei(t)/dt = Eg(t)$$

$$dEw(t)/dt = .15 [d^2 Ei(t)/dt^2 + .0051 \, dEi(t)/dt + .00005 \, Ei(t)]$$

$$d^3 s/dt^3 + .001 \, d^2 s/dt^2 = .5 \, dEw(t)/dt$$

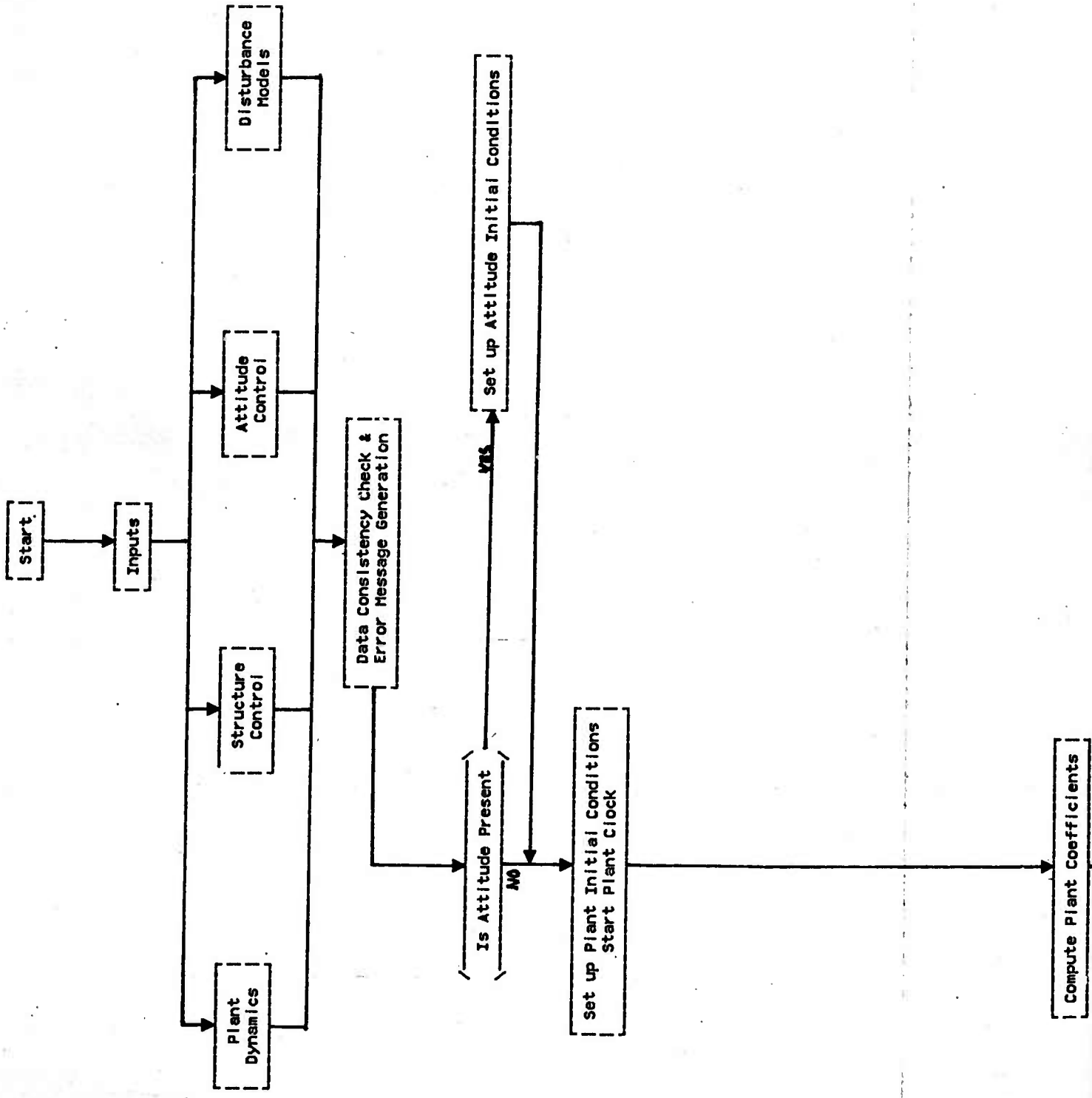
4.3 FEATURES AND CONSTRAINTS

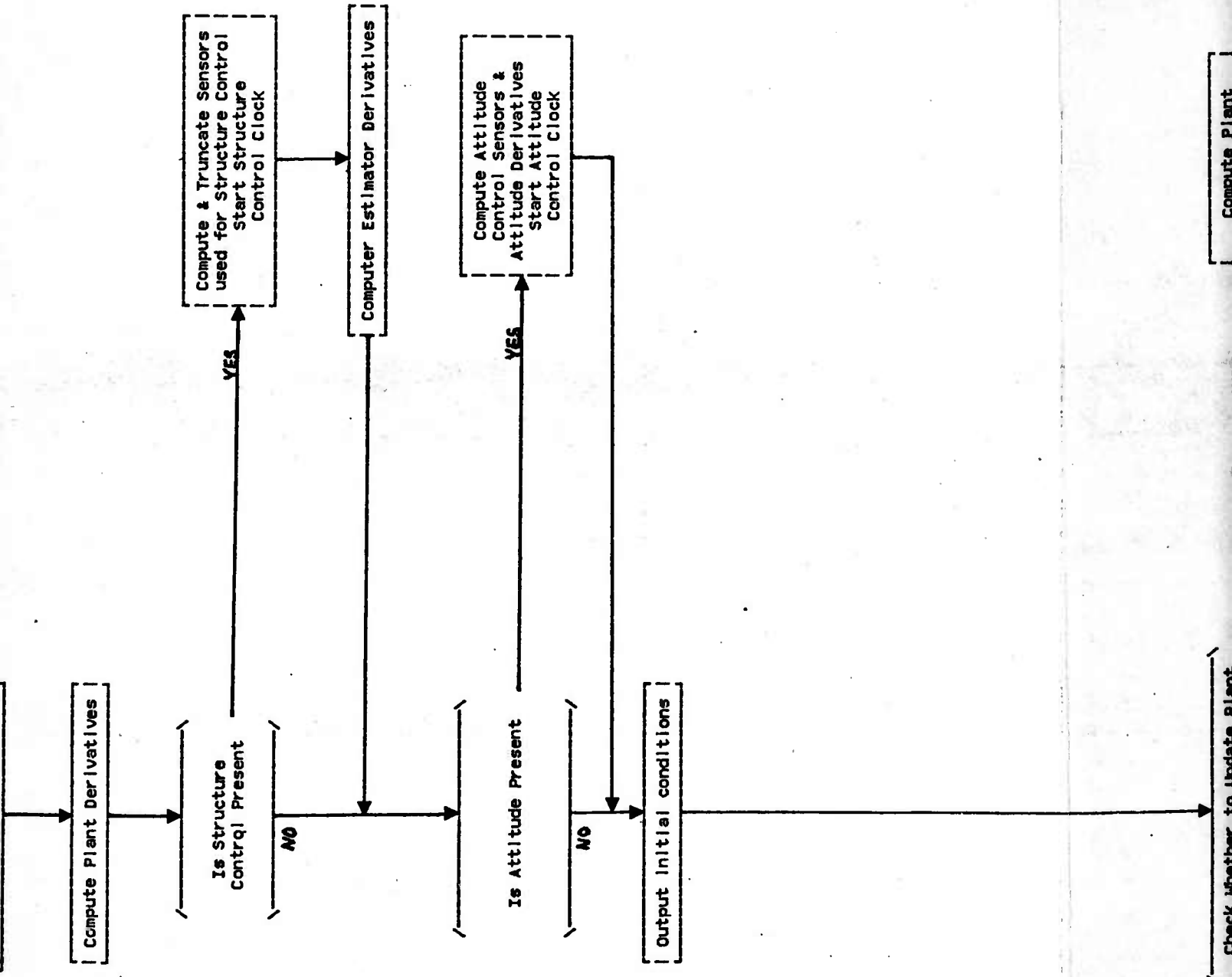
- A. Independent integration time-steps for plant, and structure and attitude control models.
- B. Truncation capability to choose desired nodes and modes for a particular set of disturbances.

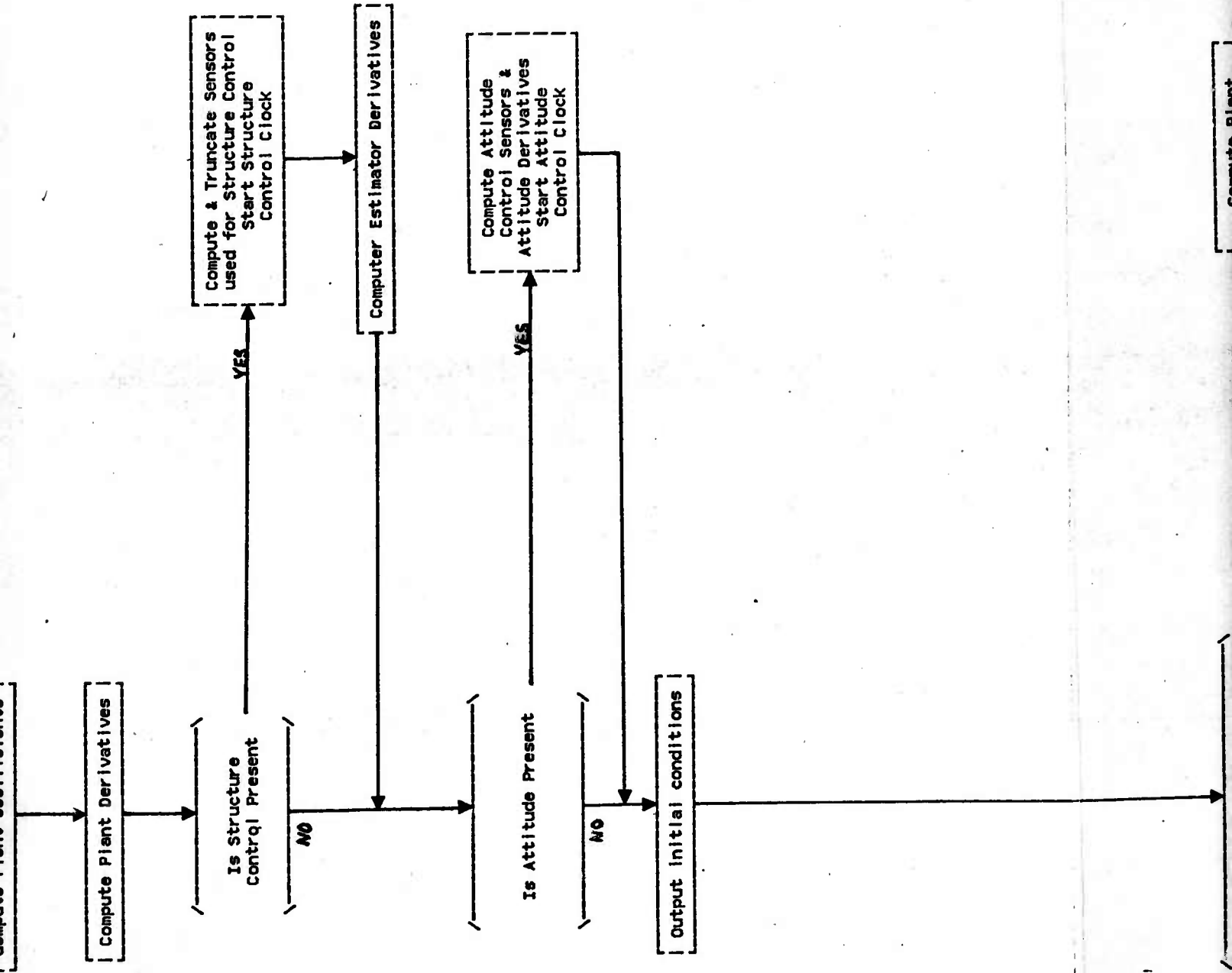
- C. Disturbances may be modeled either analytically or in tabular form.
- D. Capability to run the DYNAMIC subsystem four ways:
 - 1 - Open-loop plant dynamics
 - 2 - Closed-loop structure control
 - 3 - Closed-loop attitude control
 - 4 - Closed-loop structure and attitude control
- E. Time-histories are printed in both modal and discrete coordinates at a frequency defined by input.
- F. Files containing time-histories of selected nodes may be outputted for further analysis at a frequency determined by input.
- G. Fixed time-step for each model.
- H. Structure and attitude control models must be run at a frequency which is either equal to or an even multiple of the plant time-step.

4.4 SUBSYSTEM ARCHITECTURE

The flow of the DYNAMICS subsystem is shown in Figure 4.2







Compute & Truncate Sensors used for Structure Control Start Structure Control Clock

Computer Estimator Derivatives

Compute Attitude Control Sensors & Attitude Derivatives Start Attitude Control Clock

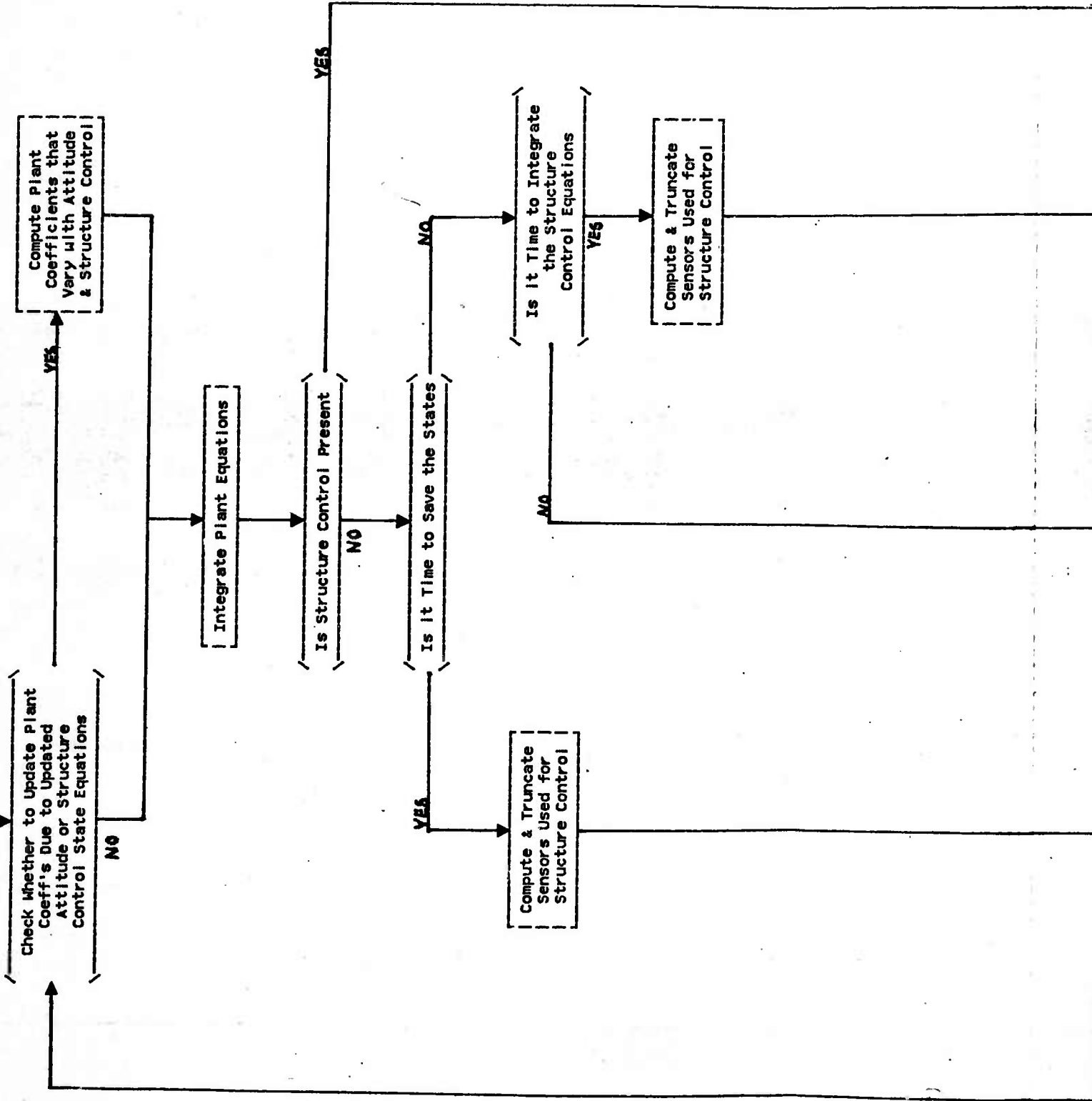
Compute Plant Derivatives

Is Structure Control Present

Is Attitude Present

Output Initial conditions

Compute Plant



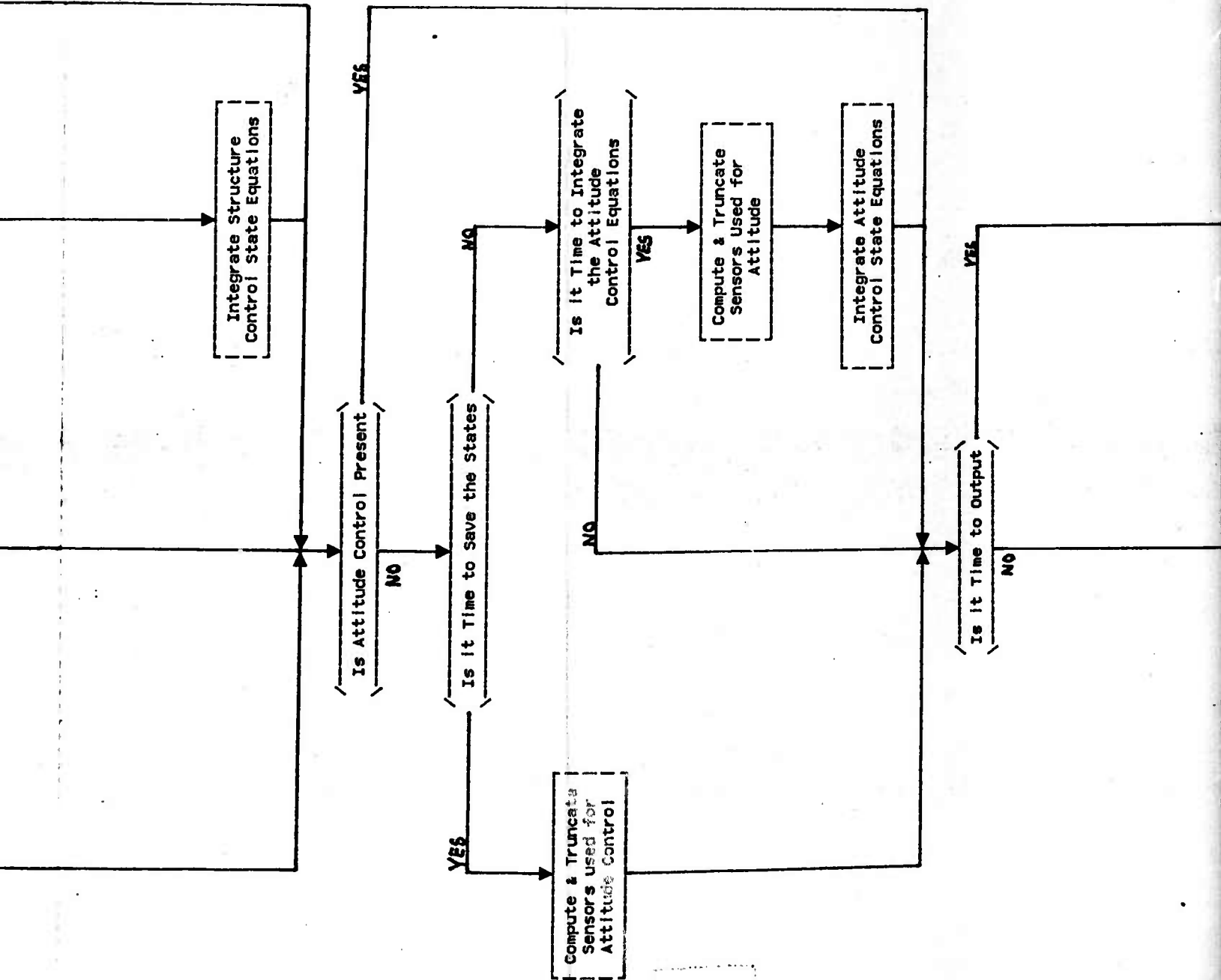
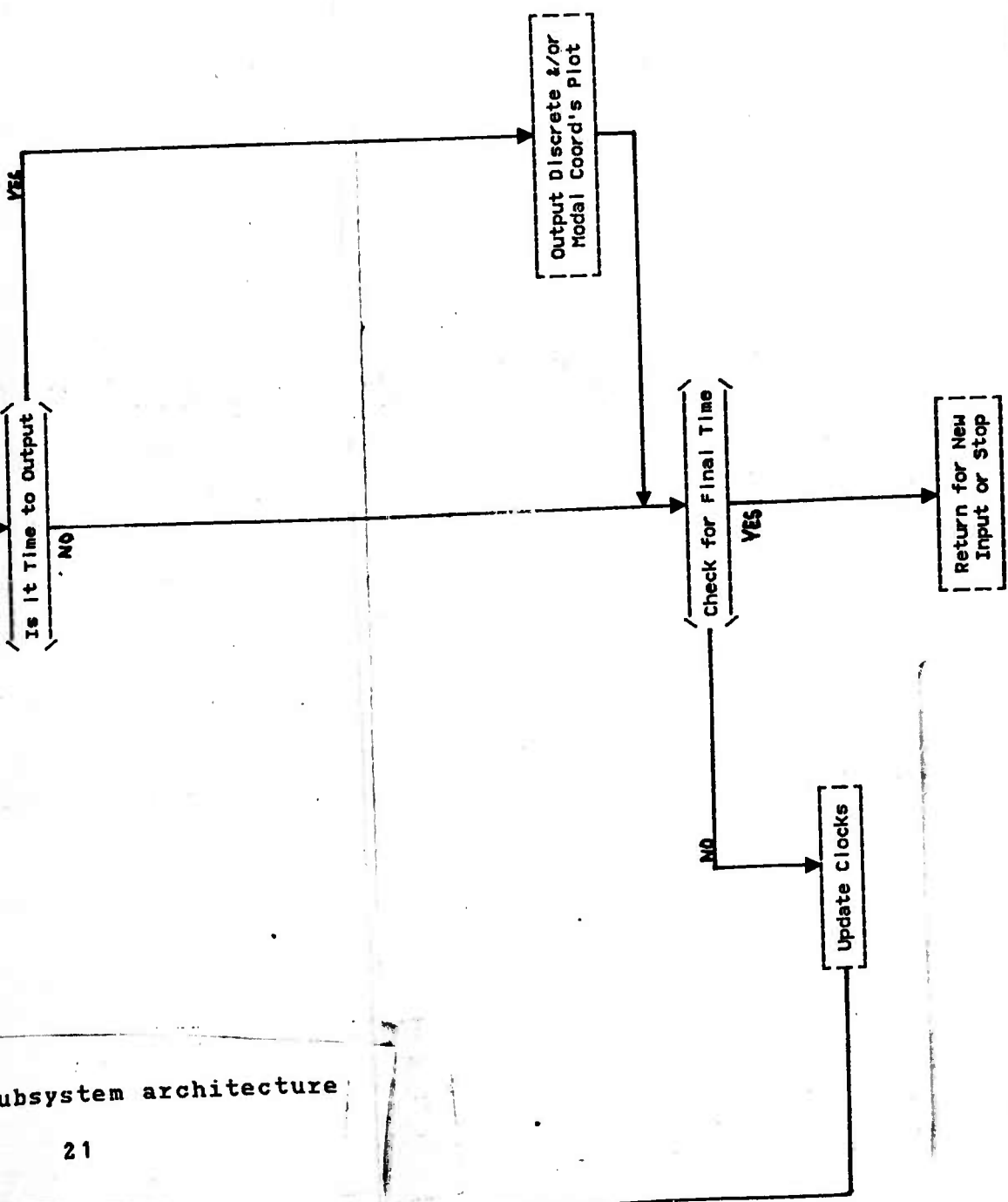


Figure 4.2 Subsystem architecture



Subsystem architecture

4.5 SUBROUTINE NESTING HIERARCHY

```
MAIN
|
|--- INPUT
|   |
|   |--- TABLES
|   |
|   |--- MOVER
|   |
|   |--- ERROR
|   |
|   |--- ABVAL
|   |
|   |--- FORCE (FORCO)
|       |
|       |--- TABLES
|       |
|       |--- ERROR
|
|--- CNTL (CNTLIN)
|   |
|   |--- TABLES
|   |
|   |--- ERROR
|
|--- CHEKER
|   |
|   |--- ERROR
|   |
|   |--- NEARZ
|
|--- AINIT
|   |
|   |--- MOVER
|
|--- COFO
|   |
|   |--- MOVER
|   |
|   |--- DOT
|   |
|   |--- MTXV
```

```

|--- COPA
|   |--- MXV
|   |--- CROSS
|   |--- DOT
|
|--- COPAC
|   |--- MXV
|   |--- MOVER
|
|--- DERIVP
|   |--- MXV
|   |--- FORCE
|       |--- TABLES
|       |--- MOVER
|       |--- LGRANG
|
|--- CNTL (DERCO)
|   |--- DOT
|   |--- MXM
|   |--- MOVER
|   |--- MXV
|
|--- CAINIT
|
|--- ETAQ
|   |--- NEARZ
|

```

```

      |--- MXV
      |
      |--- DOT
      |
|--- RKETA
      |
      |--- DERIVP
            |
            |--- MXV
            |
            |--- FORCE
                  |
                  |--- TABLES
                  |
                  |--- MOVER
                  |
                  |--- LGRANG

|--- CNTL (COCA)

|--- CNTL (COCS)

|--- CNTL (DERIVC)
      |
      |--- RKSC
      |
      |--- MXV

|--- ASN

|--- RKATT
      |
      |--- RKSC

|--- NEARZ

```

4.6 SUBROUTINE SUMMARIES

ASD(N) computes attitude sensor information as a function of plant state variables, either when the state is to be saved (N=1) or when integration is to be performed

(N=2).

Cainit initializes those attitude variables which change as a function of structure control.

Cheker checks for compatibility of size and timing between plant and control models.

Cntl contains all code related to structure control. Derivc, Coca, Cocs, and Cntlin are all entries.

Derivc(CODE) computes the first derivative of the structure control equations and calls a first-order four step integrator. When the control clock is slower than the plant CODE=1 and the saved sensor information is used along with current information. Otherwise CODE=2 and only current sensor information is used.

Derc0 initializes all structural control variables.

Coca computes the attitude-dependent portions of the control coefficients.

Cocs(CODE) computes sensor information as a function of structure control coefficients and plant state variables. If the control clock is running slower than plant clock this entry is called twice per control time-step, once when the state is to be saved (CODE=1) and once when the control equations are to be integrated (CODE=2). If the clocks are running at the same speed, it is only called once with CODE=2.

Cntlin reads input local to structure control equations.

Copa computes the attitude-dependent portions of the plant coefficients.

Copac computes the portions of the plant coefficients that vary with attitude and/or structure control.

Cop0 initializes the time-invariant portions of the plant coefficients.

Derivp computes the second derivative of the plant equations.

Etag outputs in modal and discrete coordinates.

Force(T) computes the forcing function at each time(T).
Forc0 is an entry.

Forc0 reads input local to the forcing computation. We currently have two versions of this subroutine. One obtains the function by linear interpolation from a table which was input. The other computes them using a sinusoidal function.

Input reads in input

Rketa second-order Runge-Kutta integrator.

Rksc first-order Runge-Kutta integrator.

Error(N) prints error messages and stops if serious. The parameter, N, defines the message to be printed.

Rkatt performs attitude control updates.

4.7 USER INPUT SPECIFICATIONS

A. FILE NAME: IN
SOURCE: LOCAL

RECORD 1:

FORMAT: NAMELIST 'SYSTEM'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	L*4	SYS1
2	L*4	SYS2

RECORD 2:

FORMAT: NAMELIST 'DP1'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	MD1
2	I*4	N16

RECORD 3:

FORMAT: NAMELIST 'TRUN1'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
--------------	-------------	-------------

1	I*4	NNAM1
2	I*4	MNAM1
3	L*4	N1ALL
4	L*4	M1ALL

RECORD 4:
 FORMAT: NAMELIST 'NFLAG'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	L*4	NORM3

RECORD 5:
 FORMAT: (5I10)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	NODES(k)
2	I*4	NODNOS(1,k)
3	I*4	NODNOS(2,k)
4	I*4	NODNOS(3,k)
5	I*4	NODNOS(4,k)

RECORD 6:
 FORMAT: (3E20.7)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	NORM(1,j,k)
2	R*4	NORM(2,j,k)
3	R*4	NORM(3,j,k)

RECORD 7:
 FORMAT: (I10)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	999

RECORD 8:
 FORMAT: NAMELIST 'TRUP1'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	PHI1P
2	R*4	W1P
3	R*4	PSI1P

RECORD 9:
FORMAT: NAMELIST 'DP2'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	ND2
2	I*4	N26

RECORD 10:
FORMAT: NAMELIST 'TRUN2'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	NNAM2
2	I*4	MNAM2
	L*4	N2ALL
4	L*4	M2ALL

RECORD 11:
FORMAT: NAMELIST 'TRUP2'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	PHI2P
2	R*4	W2P
3	R*4	PSI2P

RECORD 12:
FORMAT: NAMELIST 'ETAS'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	ETA

RECORD 13:
FORMAT: NAMELIST 'INR1'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	TO
2	R*4	TOPT
3	R*4	DTOPT
4	R*4	TF
5	R*4	TPRINT
6	R*4	DTPRNT
7	L*4	\$CNTL
8	L*4	\$ATT

RECORD 14:
FORMAT: NAMELIST 'INP3'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	K
2	R*4	DJ
3	R*4	IRIN
4	R*4	CI
5	R*4	DTP
6	I*4	M
7	I*4	NR
8	I*4	A1
9	I*4	A2

RECORD 15:
FORMAT: NAMELIST 'INP4'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	FD2

RECORD 16A:
FORMAT: NAMELIST 'FORC'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	P01
2	R*4	OMEGA1
3	I*4	NDIS1
4	I*4	IDIS1
5	I*4	ITUP1

RECORD 16B:
FORMAT: NAMELIST 'FORC'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	P01
2	R*4	TTAB
3	R*4	FTAB
4	I*4	NT
5	I*4	NDIS1
6	I*4	IDIS1
7	I*4	ITUP1

RECORD 17:
FORMAT: NAMELIST 'DC1'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	U
2	R*4	XHAT
3	R*4	DXHAT

RECORD 18:
 FORMAT: NAMELIST 'INC1'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	G
2	R*4	CK
3	R*4	LI
4	R*4	BET1
5	R*4	BET2
6	R*4	DTC
7	I*4	N1C
8	I*4	N2C
9	I*4	P
10	I*4	COPT
11	I*4	TRC1
12	I*4	TRC2

RECORD 19:
 FORMAT: NAMELIST 'DA1'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	S
2	R*4	EG
3	R*4	EI

RECORD 20:
 FORMAT: NAMELIST 'INA1'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	DTA

There is one record of type 1. Records 2 through 4 are required if SYS1=T. There is one of each type. One Record 5 is required for each segment if SYS1=T. Each Record 5 must be followed by 3 records of type 6 if SYS1=T and NNORN3=T. One each of Records 7 and 8 are required if SYS1=T. One record each of types 9 through 11 are required if SYS2=T. One record each of types 12 through 15 is always required. Record 16A is required if the System 1 forcing function is

of the form: $FD1 = P01 * SIN(OMEGA * TP)$. Record 16B is required if the System 1 forcing function is given in a table. One record each of types 17 and 18 are required if \$CNTL= TRUE. One record each of types 19 and 20 is always required.

4.8 EXTERNAL STORED DATA BASES

A. FILE NAME: ITAPE
SOURCE: CDC NASTRAN OR STARDYNE PROGRAMS

RECORD 1:
FORMAT: (3I10)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	NIT
2	I*4	NELT
3	I*4	MDIT

RECORD 2:
FORMAT: (I12,3E20.7)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	Node Number (Temporary)
2	R*4	COORD(1,i)
3	R*4	COORD(2,i)
4	R*4	COORD(3,i)

RECORD 3:
FORMAT: (I10)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	IDUM

RECORD 4:
FORMAT: (I12,E20.7)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	Mode Number (Temporary)
2	R*4	W1P(j)

RECORD 5:
FORMAT: (I12,3E20.7)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	Node Number (Temporary)
2	R*4	PHI1P(1,i,j)
3	R*4	PHI1P(2,i,j)
4	R*4	PHI1P(3,i,j)

RECORD 6:
FORMAT: (12X,3E20.7)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	PHI1P(4,i,j)
2	R*4	PHI1P(5,i,j)
3	R*4	PHI1P(6,i,j)

RECORD 7:
FORMAT: (2I10)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	N2T
2	I*4	MD2T

RECORD 8:
FORMAT: (I12,E20.7)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	Mode Number (Temporary)
2	R*4	W2P(j)

RECORD 9:
FORMAT: (I12,3E20.7)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	Node Number (Temporary)
2	R*4	PHI2P(1,i,j)
3	R*4	PHI2P(2,i,j)
4	R*4	PHI2P(3,i,j)

RECORD 10:
FORMAT: (12X,3E20.7)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	PHI2P(4,i,j)
2	R*4	PHI2P(5,i,j)
3	R*4	PHI2P(6,i,j)

Records 1 through 6 are required when SYS1=T. One record of type 1 is required. The number of records of type 2 is N1T. The number of records of type 3 is NELT. The number of records of type 4 is MD1T. Each record of type 4 is followed by N1T sets of records where each set consists of one record of type 5 followed by one record of type 6. Records 7 through 10 are required when SYS2=T. One record of type 7 is needed. The number of records of type 8 is MD2T. Each record of type 8 is followed by N2T sets of records where each set consists of one record of type 9 followed by one record of type 10.

4.9 INTERNAL STORED DATA BASES

A. FILE NAME: IQTAP (unformatted)
SOURCE: DYNAMICS

RECORD 1:

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	COORD(1,NODNOS(1,k))
2	R*4	COORD(2,NODNOS(1,k))
3	R*4	COORD(3,NODNOS(1,k))

RECORD 2:

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	COORD(1,NODNOS(2,k))
2	R*4	COORD(2,NODNOS(2,k))
3	R*4	COORD(3,NODNOS(2,k))

RECORD 3:

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	COORD(1,NODNOS(3,k))
2	R*4	COORD(2,NODNOS(3,k))
3	R*4	COORD(3,NODNOS(3,k))

RECORD 4:

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
--------------	-------------	-------------

1	R*4	DEL(1)
2	R*4	DEL(2)
3	R*4	DEL(3)

One record each of types 1 through 3 is produced for each segment in the system. For each time-step, NSEGS records of type 4 are produced.

4.10 GLOSSARY

A1: Node number to be activated as a function of actuator number in the System 1 force actuator influence matrix

A2: Node number to be activated as a function of actuator number in the System 2 force actuator influence matrix

BET1: Sets which node of precision base has a sensor and whether position or velocity is measured

BET2: Sets which node of navigation base has a sensor and whether position or velocity is measured

CI: Rotor spin axis direction cosines

CK: Estimator gain matrix

COORD(3,node #): X, Y, and Z coordinates for each node

COPT: Control option flag
 1 = Untruncated sensor computation
 Untruncated coefficients
 2 = Untruncated sensor computation
 Truncated coefficients
 3 = Truncated sensor computation
 Truncated coefficients

DEL(3): Deformations of the three points defining a segment

DJ: Force actuator axis (nominal distance matrix)

DTA: Attitude time-step

DTC: Control time-step size

DTOPT: Time between file outputs

DTP: Time-step size for plant equations

DTPRNT: Time between prints

DXHAT: Current velocity for estimator

EG: Attitude inputs

EI:	Attitude inputs
ETA:	Vector of modal velocities and coordinates ordered as follows:
	1 - all System 1 velocities
	2 - all System 1 coordinates
	3 - all System 2 velocities
	4 - all System 2 coordinates
FD2(dof):	System 2 disturbance
FTAB(time #,dist. #):	Table of System 1 forcing functions
G:	Control gain matrix
IDIS1:	Node numbers in System 1 where disturbances occur
IDUM:	Dummy integer
ITUP1:	Degree-of-freedom within the System 1 node where disturbances occur (must be 1 through 6)
IRIN:	Rotor inertia matrix in attached body vector
K:	Force actuator spring constant
LI:	Sets the sensor axis
M:	Number of force actuators
MD1:	Number of System 1 modes to be used
MD1T:	Total number of System 1 modes in the model
NORM3:	Flag which indicates if three normal vectors will be inputted for each segment
MD2:	Number of System 2 modes to be used
MD2T:	Total number of System 2 modes in the model
MNAM1:	Mode numbers to be included in System 1
MNAM2:	Mode numbers to be included in System 2
M1ALL:	Flag which is set when all System 1 modes are to be used
M2ALL:	Flag which is set when all System 2 modes are to be used
NDIS1:	Number of System 1 disturbances
NELT:	Number of dummy records on file ITAPE
NNAM1:	Node numbers to be included in System 1
NNAM2:	Node numbers to be included in System 2

NODES(segment #): Number of nodes in each segment
 (either 3 or 4)
 NODNOS(node #,segment #): Node numbers for the segment
 NR: Determines which node in the rotor
 influence matrix has an activated
 wheel
 NSEGS: Number of segments in the system
 NT: Number of entries in table of
 System 1 forcing functions
 N1ALL: Flag which is set when all System
 1 nodes are to be used
 N1C: Number of truncated states to be
 used in precision base
 N1T: Total number of System 1 nodes in
 the model
 N16: Number of System 1 nodes to be
 used
 N2ALL: Flag which is set when all System
 2 nodes are to be used
 N2C: Number of truncated states to be
 used in navigation base
 N2T: Total number of System 2 nodes in
 the model
 N26: Number of System 2 nodes to be
 used
 OMEGA1: Forcing frequency
 P: Number of sensors
 PHI1P(dof,node #,mode #): Set of normalized mode shapes
 relative to the mass matrix for
 System 1
 PHI2P(dof,node #,mode #): Set of normalized mode shapes
 relative to the mass matrix for
 System 2
 PSI1P(mode #): Damping ratio per mode
 PSI2P(mode #): Damping ratio per mode
 P01: Amplitude of forcing function
 S: Attitude inputs
 SYS1: Logical flag which is true if
 System 1 is present
 SYS2: Logical flag which is true if
 System 2 is present
 TF: Final time of dynamic simulation
 TOPT: Time of next file output
 TPRINT: Time of next print
 TRC1: Precision base states to be used
 by estimator
 TRC2: Navigation base states to be used
 by estimator

TTAB(time #):	Table of times for System 1 forcing functions
T0:	Initial time of dynamic simulation
U:	Current control vector
W1P(mode #):	Natural frequency associated with each mode (rad/sec)
W2P(mode #):	Natural frequency associated with each mode (rad/sec)
XHAT.	Current position for estimator
\$ATT:	Flag which indicates if attitude control is present
\$CNTL:	Flag which indicates if structure control is present

5. STEADY STATE

5.1 PURPOSE

This program computes the steady-state dynamic equations of motion open-loop. The output at each time step is a vector of displacements for each support point having six degrees-of-freedom (three translational and three rotational components). This is converted into a deformation for each support point.

5.2 TECHNICAL SUMMARY

A set of algebraic equations as defined below are computed for a number of equally spaced times between $t=0$ and $t=2\pi/\omega_D$ to produce a set of steady-state dynamic deformation time-histories for each specified frequency of disturbance (ω_D). For each mode:

$$GF_J = 1/\omega_J^2 (F_1 \phi_{D1,J} + F_2 \phi_{D2,J} + \dots + F_N \phi_{DN,J})$$

where N is the number of disturbances and DI is the degree-of-freedom at which disturbance I is located

$$\gamma_J = \omega_D / \omega_J$$

$$A_J = 1 - \gamma_J^2$$

$$B_J = -2 \zeta_J \gamma_J$$

$$C_J = GF_J / (A_J^2 + B_J^2)$$

If $\omega_J = 0$,

$$GF_J = F_1 \phi_{D1,J} + F_2 \phi_{D2,J} + \dots + F_N \phi_{DN,J}$$

$$A_J = -1 / \omega_D^2$$

$$B_J = 0$$

$$C_J = GF_J$$

For each degree-of-freedom:

$$\alpha_I = A_1 C_1 \phi_{I,1} + A_2 C_2 \phi_{I,2} + \dots + A_M C_M \phi_{I,M}$$

$$\beta I = B_1 C_1 \phi_{I,1} + B_2 C_2 \phi_{I,2} + \dots + B_M C_M \phi_{I,M}$$

where M is the number of modes.

For p equally spaced times between $t=0$ and $t=2\pi/\omega$:

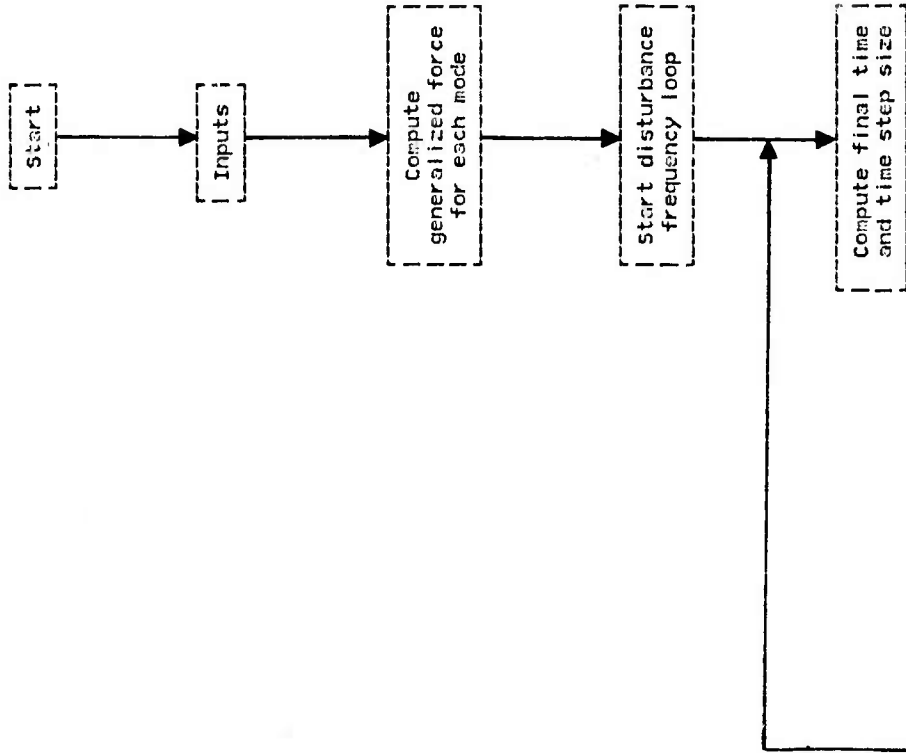
$$q_I(t) = \alpha_I \sin(\omega t) + \beta_I \cos(\omega t)$$

5.3 FEATURES AND CONSTRAINTS

- A. Truncation capability to choose desired nodes and modes for a particular set of disturbances.
- B. The STEADY STATE subsystem may only be run open loop. There is currently no provision for any control.
- C. Only one disturbance frequency may be used at a time.
- D. Files containing time-histories of selected nodes may be output for further analysis at a frequency determined by input.

5.4 SUBSYSTEM ARCHITECTURE

The flow of the STEADY STATE subsystem is shown in Figure 5.1



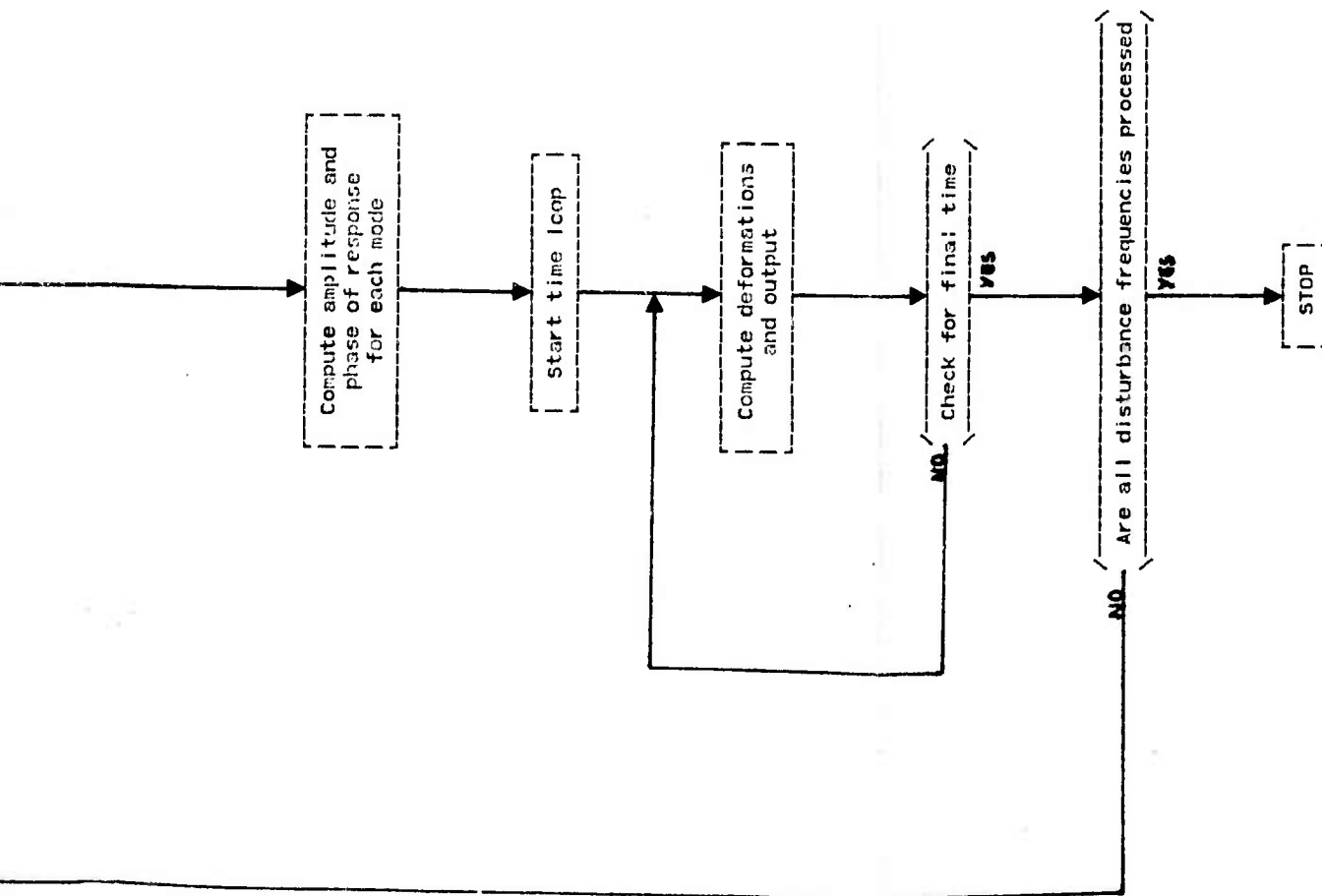
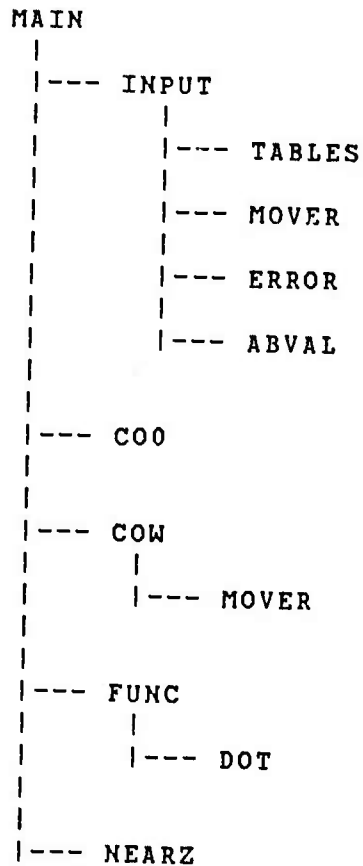


Figure 5.1 Subsystem architecture

5.5 SUBROUTINE NESTING HIERARCHY



5.6 SUBROUTINE SUMMARIES

Co0 computes portions of the coefficients that are independent of frequency.

Cow computes the coefficients for each frequency.

Func computes velocities and coordinates for all degrees-of-freedom and outputs deformations at each time for each frequency.

Input reads all input.

5.7 USER INPUT SPECIFICATIONS

A. FILE NAME: IN
SOURCE: LOCAL

RECORD 1:
FORMAT: NAMELIST 'DP1'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	MD1
2	I*4	N16

RECORD 2:
FORMAT: NAMELIST 'TRUN1'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	NNAM1
2	I*4	MNAM1
3	L*4	N1ALL
4	L*4	M1ALL

RECORD 3:
FORMAT: NAMELIST 'NFLAG'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	L*4	NORM3

RECORD 4:
FORMAT: (5I10)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	NODES(k)
2	I*4	NODNOS(1,k)
3	I*4	NODNOS(2,k)
4	I*4	NODNOS(3,k)
5	I*4	NODNOS(4,k)

RECORD 5:
FORMAT: (3E20.7)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	NORM(1,j,k)
2	R*4	NORM(2,j,k)
3	R*4	NORM(3,j,k)

RECORD 6:
FORMAT: (I10)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	999

RECORD 7:
 FORMAT: NAMELIST 'TRUP1'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	PHI1P
2	R*4	W1P
3	R*4	PSI1P

RECORD 8:
 FORMAT: NAMELIST 'INP1'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	W
2	I*4	P
2	I*4	NW

RECORD 9:
 FORMAT: NAMELIST 'INP2'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	F
2	I*4	NDIS1
3	I*4	IDIS1
4	I*4	ITUP1

RECORD 10:
 FORMAT: NAMELIST 'INP3'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	FREQ
2	I*4	NEXT0

There is one record each of types 1 through 3. One Record 4 is required for each segment. Each Record 4 must be followed by 3 records of type 5 if NNORM3=T. One each of Records 6 through 9 are required.

5.8 EXTERNAL STORED DATA BASES

A. FILE NAME: ITAPE
SOURCE: CDC NASTRAN OR STARDYNE PROGRAMS

RECORD 1:
FORMAT: (3I10)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	N1T
2	I*4	NELT
3	I*4	MD1T

RECORD 2:
FORMAT: (I12,3E20.7)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	Node Number (Temporary)
2	R*4	COORD(1,i)
3	R*4	COORD(2,i)
4	R*4	COORD(3,i)

RECORD 3:
FORMAT: (I10)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	IDUM

RECORD 4:
FORMAT: (I12,E20.7)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	Mode Number (Temporary)
2	R*4	W1P(j)

RECORD 5:
FORMAT: (I12,3E20.7)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	Node Number (Temporary)
2	R*4	PHI1P(1,i,j)
3	R*4	PHI1P(2,i,j)

4 R*4 PHI1P(3,i,j)

RECORD 6:
FORMAT: (12X,3E20.7)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	PHI1P(4,i,j)
2	R*4	PHI1P(5,i,j)
3	R*4	PHI1P(6,i,j)

RECORD 7:
FORMAT: (2I10)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	N2T
2	I*4	MD2T

RECORD 8:
FORMAT: (I12,E20.7)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	Mode Number (Temporary)
2	R*4	W2P(j)

RECORD 9:
FORMAT: (I12,3E20.7)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	Node Number (Temporary)
2	R*4	PHI2P(1,i,j)
3	R*4	PHI2P(2,i,j)
4	R*4	PHI2P(3,i,j)

RECORD 10:
FORMAT: (12X,3E20.7)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	PHI2P(4,i,j)
2	R*4	PHI2P(5,i,j)
3	R*4	PHI2P(6,i,j)

Records 1 through 6 are required when SYS1=T. One record of

type 1 is required. The number of records of type 2 is N1T. The number of records of type 3 is NELT. The number of records of type 4 is MD1T. Each record of type 4 is followed by N1T sets of records where each set consists of one record of type 5 followed by one record of type 6. Records 7 through 10 are required when SYS2=T. One record of type 7 is needed. The number of records of type 8 is MD2T. Each record of type 8 is followed by N2T sets of records where each set consists of one record of type 9 followed by one record of type 10.

5.9 INTERNAL STORED DATA BASES

A. FILE NAME: IQTAP(unformatted)

RECORD 1:

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	COORD(1,NODNOS(1,k))
2	R*4	COORD(2,NODNOS(1,k))
3	R*4	COORD(3,NODNOS(1,k))

RECORD 2:

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	COORD(1,NODNOS(2,k))
2	R*4	COORD(2,NODNOS(2,k))
3	R*4	COORD(3,NODNOS(2,k))

RECORD 3:

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	COORD(1,NODNOS(3,k))
2	R*4	COORD(2,NODNOS(3,k))
3	R*4	COORD(3,NODNOS(3,k))

RECORD 4:

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	DEL(1)
2	R*4	DEL(2)
3	R*4	DEL(3)

One record each of types 1 through 3 is produced for each segment in the system. For each time step, NSEGS records of type 4 are produced.

5.10 GLOSSARY

COORD(3,node #):	X, Y, and Z coordinates for each node
DEL(3):	Deformations of the three points defining a segment
F(frequency #):	Amplitude of disturbance
FREQ:	Number of time-steps between printouts
IDIS1:	Node numbers in System 1 where disturbances occur
IDUM:	Dummy integer
ITUP1:	Degree-of-freedom within the System 1 node where disturbances occur (must be 1 through 6)
MD1:	Number of System 1 modes to be used
MD1T:	Total number of System 1 modes in the model
NORM3:	Flag which indicates if three normal vectors will be input for each segment
MNAM1:	Mode numbers to be included in System 1
M1ALL:	Flag which is set when all System 1 modes are to be used
NDIS1:	Number of System 1 disturbances
NELT:	Number of dummy records on file ITAPE
NEXT0:	First printout time-step
NNAM1:	Node numbers to be included in System 1
NODES(segment #):	Number of nodes in each segment (either 3 or 4)
NODNOS(node #,segment #):	Node numbers for the segment
NORM(3,n,segment #):	Normal vector for the segment n=1 if NNORM3=F n=3 if NNORM3=T
NSEGS:	Number of segments in the system
NW:	Number of frequencies
N1ALL:	Flag which is set when all System 1 nodes are to be used
N1T:	Total number of System 1 nodes in the model
N16:	Number of System 1 nodes to be used
P(frequency #):	Number of time-steps

PHI1P(dof,node #,mode #): Set of normalized mode shapes
relative to the mass matrix for
System 1
PSI1P(mode #): Damping ratio per mode (ζ_j)
W(frequency #): Frequencies (ω)
W1P(mode #): Natural frequency associated with
each mode (ω_j)
(rad/sec)

6. QUASI-STATIC

6.1 PURPOSE

This program corrects the thermally induced displacements of the optical surfaces by the use of position and figure control actuators.

6.2 TECHNICAL SUMMARY

For each node n in segment s of mirror m in the system, the component of deflection normal to the mirror surface is:

$$E0(n,s,m) = \bar{N}(3,n,s,m) \cdot U0(3,n,s,m)$$

where \bar{N} is the normal vector at the node and $U0$ is the total deflection of the node.

The initial surface deflection of each mirror is:

$$E0(3,m) = -\sum_{S,N} \bar{N}(3,n,s,m) E0(n,s,m)$$

Each element of the 3×3 rigid body sensitivity matrix for each mirror is:

$$E(j,k,m) = \sum_{S,N} \bar{N}(j,n,m) \bar{N}(k,n,m)$$

The rigid-body movement of each mirror due to the thermal load is:

$$R(3,m) = E(3,3,m)^{-1} E0(3,m)$$

The deflection due to the optical train <--> backup truss actuators is:

$$U1(3,n,m,s) = U0(3,n,m,s) - R(3,m)$$

The initial surface deflection of each segment is:

$$E0(3,s,m) = -\sum_N \bar{N}(3,n,s,m) E0(n,s,m)$$

Each element of the 3x3 rigid-body sensitivity matrix for each segment is:

$$E(j,k,s,m) = \sum_N \bar{N}(j,n,s,m) \bar{N}(k,n,s,m)$$

The rigid-body movement of each segment due to the thermal load is:

$$R(3,s,m) = E(3,3,s,m)^{-1} E0(3,s,m)$$

The deflection due to the backup truss <--> segment actuators is:

$$U2(3,n,m,s) = U1(3,n,m,s) - R(3,s,m)$$

The residual deformation of each segment due to the thermal load after implementation of the quasi-static control system is:

$$U3(3,n,m,s) = U2(3,n,m,s) (I - A(AA)^{-1}A) + UA^*$$

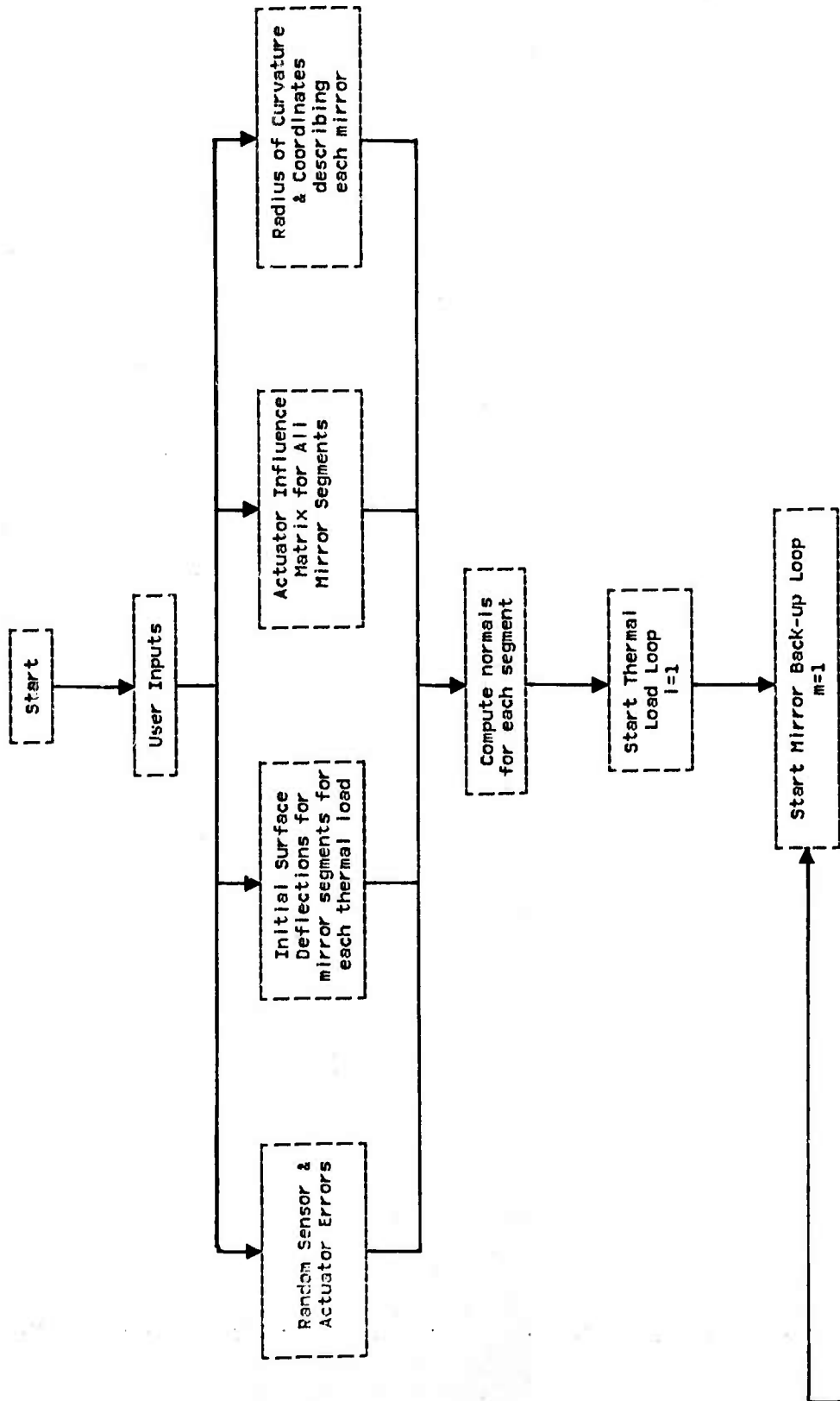
where A is the actuator influence matrix of the segment and UA* is the deflection due to random actuator errors.

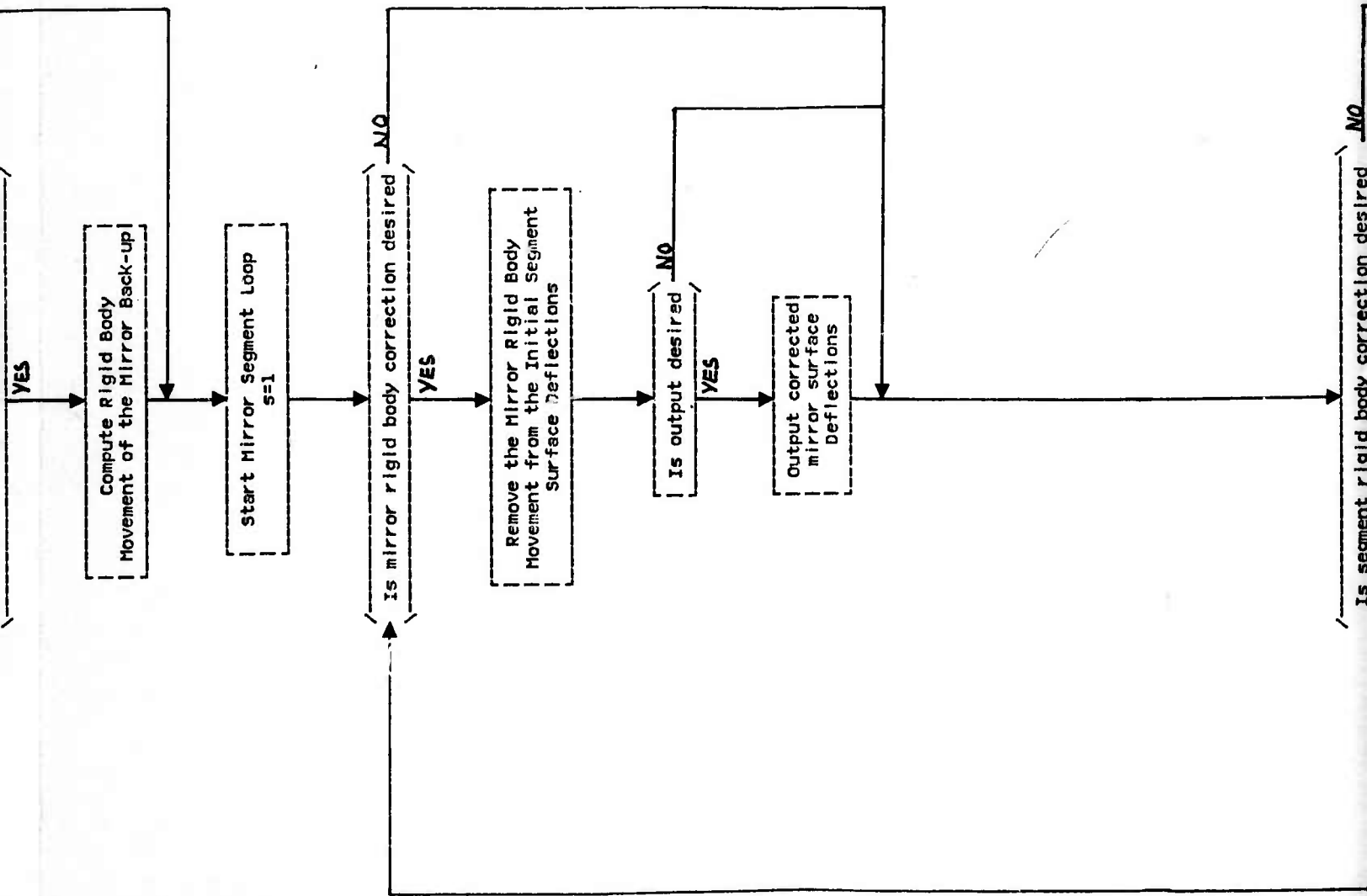
6.3 FEATURES AND CONSTRAINTS

- A. Rigid-body actuators can be used to remove piston and/or tilt motions.
- B. Surface actuators correct surface deformations.
- C. Any combination of corrections may be performed.
- D. Output may be produced before corrections and/or after any corrections.

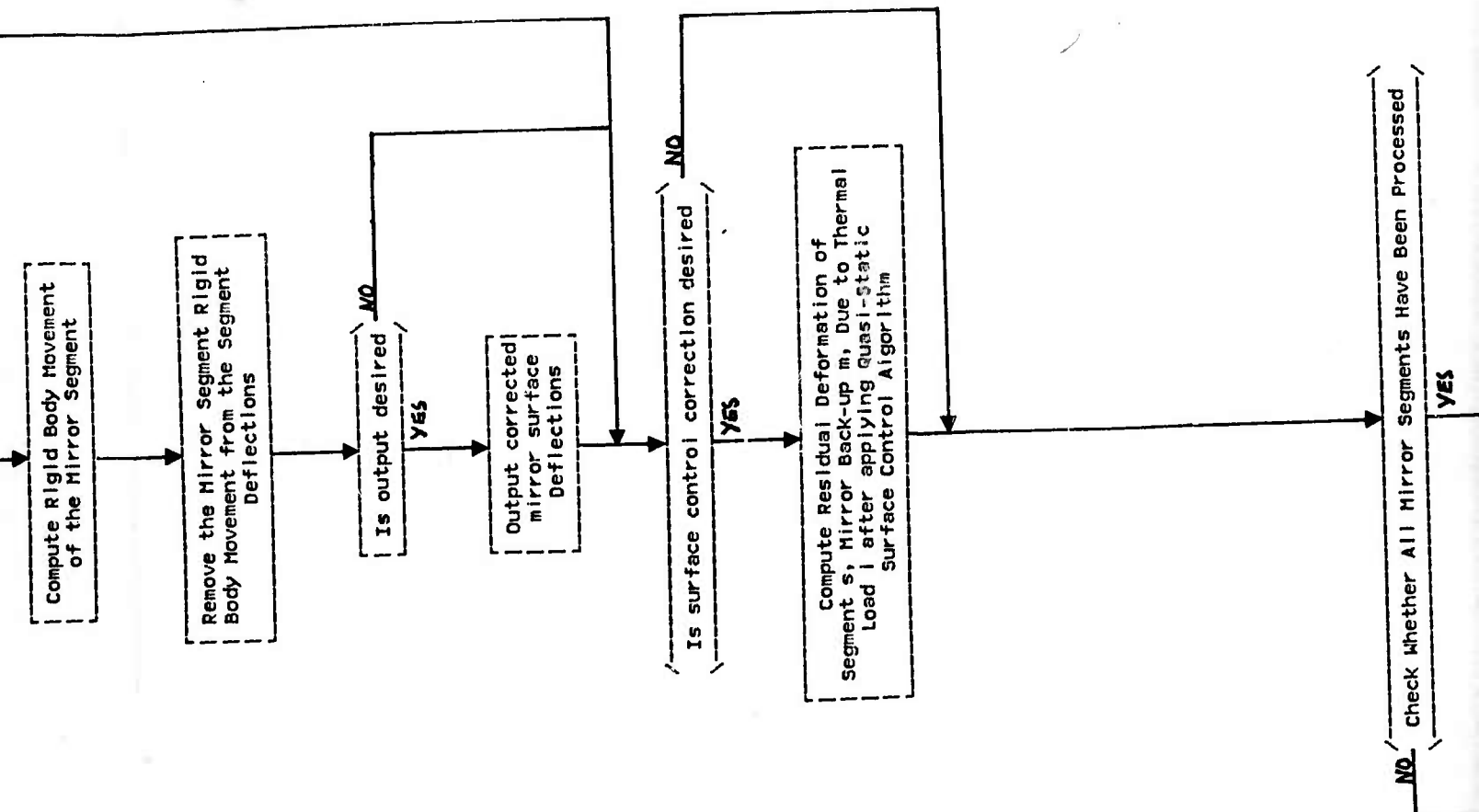
6.4 SUBSYSTEM ARCHITECTURE

The flow of the QUASI-STATIC subsystem is shown in Figure 6.1





Is segment rigid body correction desired NO



Compute Rigid Body Movement of the Mirror Segment

Remove the Mirror Segment Rigid Body Movement from the Segment Deflections

Is output desired?

Output corrected mirror surface Deflections

Is surface control correction desired?

Compute Residual Deformation of Segment s, Mirror Back-up m, Due to Thermal Load l after applying Quasi-Static Surface Control Algorithm

Check Whether All Mirror Segments Have Been Processed?

lv

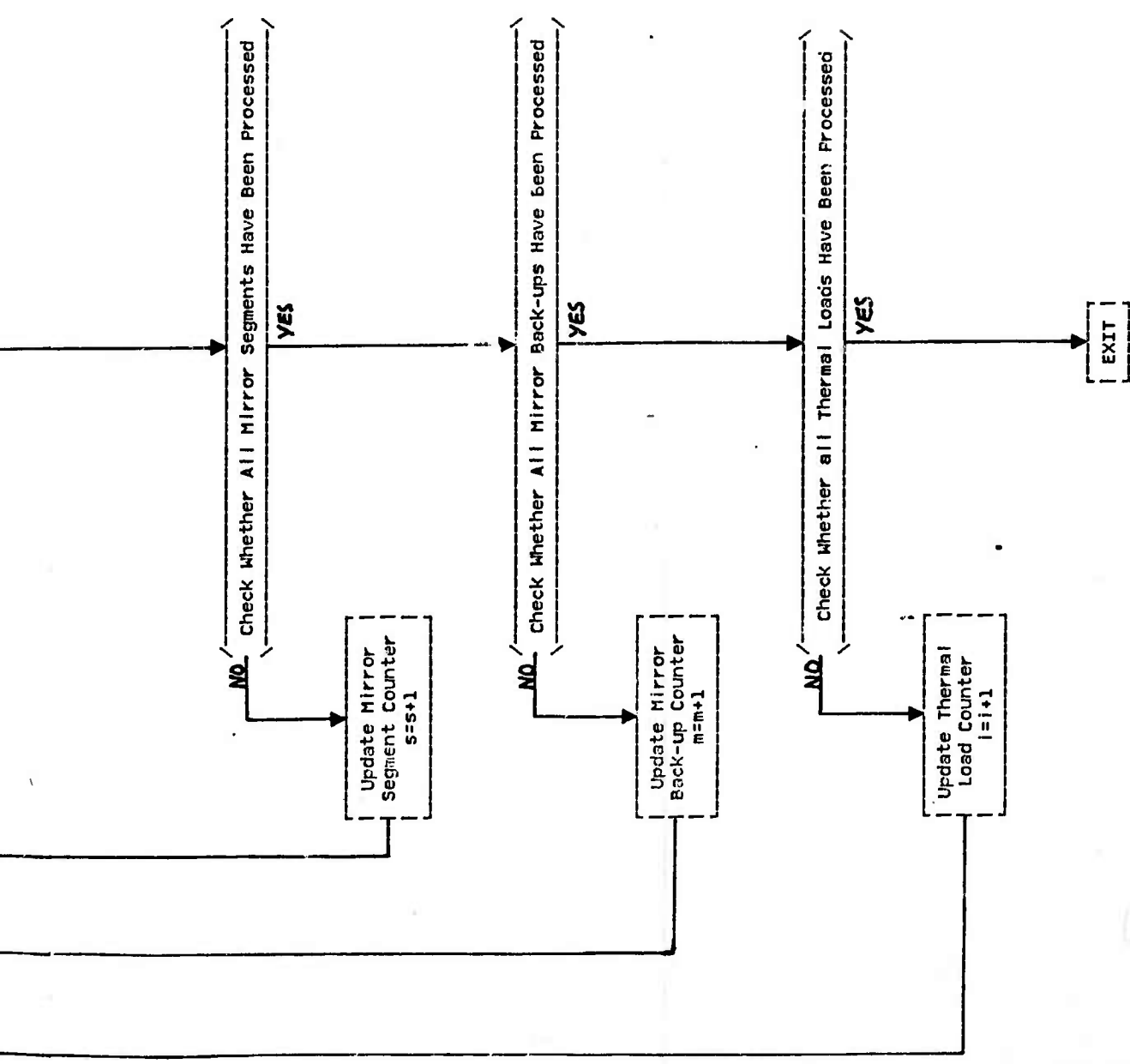


Figure 6.1 Subsystem architecture

Uout(U,N,NO,NVEC,RMSTOT,NTOT,TOTAL,I,M) computes the deformation for each node in the normal direction and prints them with the root-mean-square. If TOTAL=TRUE, the total root-mean-square for the mirror is printed. The deformations are also written onto one or more files, scaled if required.

Erstop prints error messages and stops if severe.

Random(VEC,N) currently inserts N zeros into a vector.

Input Reads and echoes local input and sets up files for scaling results if required.

Matcom(ROW,U2,U3,NVEC,N) Computes:

$$U3 = U2 (I - A(A A)^{-1}A) + U3$$

using ROW as a holding vector.

Prepro computes normal vectors and rigid-body sensitivity matrices.

6.7 USER INPUT SPECIFICATIONS

A. FILE NAME: IN
SOURCE: LOCAL

RECORD 1:
FORMAT: NAMELIST INPT

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	NI
2	I*4	NM
3	I*4	NSEGS
4	I*4	NODES
5	I*4	EXROWS
6	L*4	\$MAT
7	L*4	\$WRIT1
8	L*4	\$WRIT2
9	L*4	\$1
10	L*4	\$2
11	L*4	\$3
12	L*4	PUSH
13	L*4	TILT

6.5 SUBROUTINE NESTING HIERARCHY

```
MAIN
|
|--- INPUT
|
|--- PREPRO
|   |
|   |--- MOVER
|   |
|   |--- SI
|
|--- MOVER
|
|--- DOT
|
|--- MXVSYM
|
|--- UOUT
|   |
|   |--- DOT
|
|--- RANDOM
|
|--- MATCOM
|   |
|   |--- DOT
|   |
|   |--- SI
|   |
|   |--- MOVER
|
|--- SCALE
```

6.6 SUBROUTINE SUMMARIES

Scale consolidates scaled deformations into one file for each type of correction. This routine is only invoked when \$SCALE=TRUE.

Uout(U,N,NO,NVEC, RMSTOT, NTOT, TOTAL, I, M) computes the deformation for each node in the normal direction and prints them with the root-mean-square. If TOTAL=TRUE, the total root-mean-square for the mirror is printed. The deformations are also written onto one or more files, scaled if required.

Erstop prints error messages and stops if severe.

Random(VEC, N) currently inserts N zeros into a vector.

Input Reads and echoes local input and sets up files for scaling results if required.

Matcom(ROW, U2, U3, NVEC, N) Computes:

$$U3 = U2 (I - A(A A)^{-1} A) + U3$$

using ROW as a holding vector.

Prepro computes normal vectors and rigid-body sensitivity matrices.

6.7 USER INPUT SPECIFICATIONS

A. FILE NAME: IN
SOURCE: LOCAL

RECORD 1:
FORMAT: NAMELIST INPT

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	NI
2	I*4	NM
3	I*4	NSEGS
4	I*4	NODES
5	I*4	EXROWS
6	L*4	\$MAT
7	L*4	\$WRIT1
8	L*4	\$WRIT2
9	L*4	\$1
10	L*4	\$2
11	L*4	\$3
12	L*4	PUSH
13	L*4	TILT

RECORD 2:
FORMAT: NAMELIST IN2

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*8	FACT
2	I*4	NT
3	L*4	\$SCALE

There is one record each of types 1 and 2.

6.8 EXTERNAL STORED DATA BASES

A. FILE NAME: FIL1
SOURCE: CDC NASTRAN OR STARDYNE PPROGRAMS

RECORD 1:
FORMAT: (3E20.7)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	U(1,J)
2	R*4	U(2,J)
3	R*4	U(3,J)

There is one record of type 1 for each node and each thermal load.

B. FILE NAME: FIL5
SOURCE: CDC NASTRAN OR STARDYNE PROGRAMS

RECORD 1:
FORMAT: (I10)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	FIN

RECORD 2:
FORMAT: (5E16.6)¹

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	A(1,1,1)
2	R*4	A(2,1,1)
3	R*4	A(3,1,1)
4	R*4	A(1,2,1)
5	R*4	A(2,2,1)

There is one record of type 1 for each segment of each mirror. The number of records of type 2 needed for each segment is $3 * \text{NODES}(S,M) * (\text{FIN} + \text{EXROWS}) / 5$. The number of sets of Record 1 followed by the proper number of Records 2 is determined by the number of segments in the system.

C. FILE NAME: FIL7
SOURCE: CDC NASTRAN OR STARDYNE PROGRAMS

RECORD 1:
FORMAT: (F10.5)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	RC

RECORD 2:
FORMAT: (2E20.10)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	X
2	R*4	Y

There is one record of type 1 for each mirror. The number of records of type 2 is determined by the total number of nodes on the mirror. The number of sets of Record 1 followed by the proper number of Records 2 is determined by the number of mirrors in the system.

¹ This format was changed from (8F10.7) on September 19, 1978.

6.9 INTERNAL STORED DATA BASES

A. FILE NAME: FIL2(unformatted)

RECORD 1:

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	NVEC(1,1)
2	R*4	NVEC(2,1)
3	R*4	NVEC(3,1)
.	R*4	.
.	R*4	.
.	R*4	.
.	R*4	NVEC(1,J)
.	R*4	NVEC(2,J)
.	R*4	NVEC(3,J)

One record of type 1 is produced for each segment of each mirror. Each record has 3*NODES(S,M) fields.

B. FILE NAME: FIL6(unformatted)

RECORD 1:

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	ROW(1,1)
2	R*4	ROW(2,1)
3	R*4	ROW(3,1)
.	R*4	.
.	R*4	.
.	R*4	.
.	R*4	ROW(1,NODES(S,M))
.	R*4	ROW(2,NODES(S,M))
.	R*4	ROW(3,NODES(S,M))

Both the number of fields per record and the number of records per segment is 3*NODES(S,M).

C. FILE NAME: UFIL(unformatted)

RECORD 1:

1	R*4	DELT(1)
---	-----	---------

```

.      R*4      .
.      R*4      .
.      R*4      .
J      R*4      DELT(J)

```

One record of type 1 is produced for each segment of each mirror. Each record has NODES(S,M) fields. There are four of these files.

6.10 GLOSSARY

A(3,node #,actuator #+EXROWS):	Actuator influence matrix for a segment
DELT(node #):	Deformation of a node
EXROWS:	Number of rows added to A for dependent push and tilt removal
FACT(mirror #,time #,thermal #):	Scale factor
FIN:	Number of figure actuators for a segment
J:	Node number
M:	Mirror number
NI:	Number of time-steps or thermal loads
NM:	Number of mirrors
NT(thermal #):	Number of times for scaling
NODES(segment #,mirror #):	Number of nodes in each segment
NSEGS(mirror #):	Number of segments in each mirror
NVEC(3,node#):	Normal vector for each node
PUSH:	Flag which indicates if mirror rigid body actuators should remove push motion in mirror rigid-body computation
RC:	Radius of curvature
ROW(3,node #):	Row of the matrix used for figure control
S:	Segment number
TILT:	Flag which indicates if mirror rigid-body actuators should remove tilt motion
U(3,node #):	Initial displacements for each node
U1(3,node #):	Displacement for each node

	after mirror rigid-body correction
U2(3,node #):	Displacement for each node after segment rigid-body correction
U3(3,node #):	Displacement for each node after random actuator correction
X:	X coordinate of node
Y:	Y coordinate of node
\$MAT:	Flag which indicates whether matrix used for figure control will be computed
\$SCALE:	Flag which indicates whether scaling is to be done T - scale F - no scaling
\$WRIT1:	Flag which indicates whether output is desired after mirror rigid-body computation
\$WRIT2:	Flag which indicates whether output is desired after segment rigid-body computation
\$1:	Flag which indicates whether computation of mirror rigid-body deformations is desired
\$2:	Flag which indicates whether computation of segment rigid-body deformations is desired
\$3:	Flag which indicates whether computation of figure control deformations is desired

7. INTERFACE

7.1 PURPOSE

This program interpolates residual deformations produced by the DYNAMICS or STEADY STATE and/or the QUASI-STATIC programs at the structural points, to obtain deformations at the optical points. The wave aberrations produced by the combined deformations are calculated using the raytrace data of the undeformed system and are added to the disturbance-free aberrations of the system to obtain the total wave aberrations of the deformed system.

7.2 TECHNICAL SUMMARY

The INTERFACE subsystem can presently utilize deformations from two sources, dynamic and quasi-static, to produce total net wave aberrations at the raytrace points.

7.2.1 Dynamic

The deformation, δ_{DM} , at the i th raytrace point on mirror M is computed as follows:

$$\begin{aligned} \delta_{DM} = & \{ [y_I(x_C - x_B) + y_B(x_I - x_C) + y_C(x_B - x_I)] \delta_{MA} \\ & + [y_I(x_A - x_C) + y_A(x_C - x_I) + y_C(x_I - x_A)] \delta_{NB} \\ & + [y_I(x_B - x_A) + y_A(x_I - x_B) + y_B(x_A - x_I)] \delta_{NC} \} / \\ & [y_A(x_C - x_B) + y_B(x_A - x_C) + y_C(x_B - x_A)] \end{aligned}$$

where it is assumed that the segment of the mirror touched by the ray is a plane defined by support points A, B, and C, having associated deformations, δ_{MA} , δ_{NB} and δ_{NC} .

7.2.2 Quasi-static

The deformation, δ_{QM} , at the i th raytrace point on mirror M is computed as follows:

$$\delta_{QM} = \sum_{N=1}^{S+1} \sum_{L=1}^{R+1} \left[\prod_{K=1}^{S+1} \left[\frac{(y_I - y_K)}{(y_N - y_K)} \right] \prod_{J=1}^{R+1} \left[\frac{(x_I - x_J)}{(x_L - x_J)} \right] \right] \delta_{LN}$$

where R and s are the orders of interpolation desired in each direction respectively and the x's and y's are the nodal coordinates for interpolation.

7.2.3 Net Wave Aberrations

The deformations obtained by the methods in Sections 7.2.1 and 7.2.2 are now combined on each mirror and summed over N mirrors to produce the total aberration W of the ith ray by the following expression:

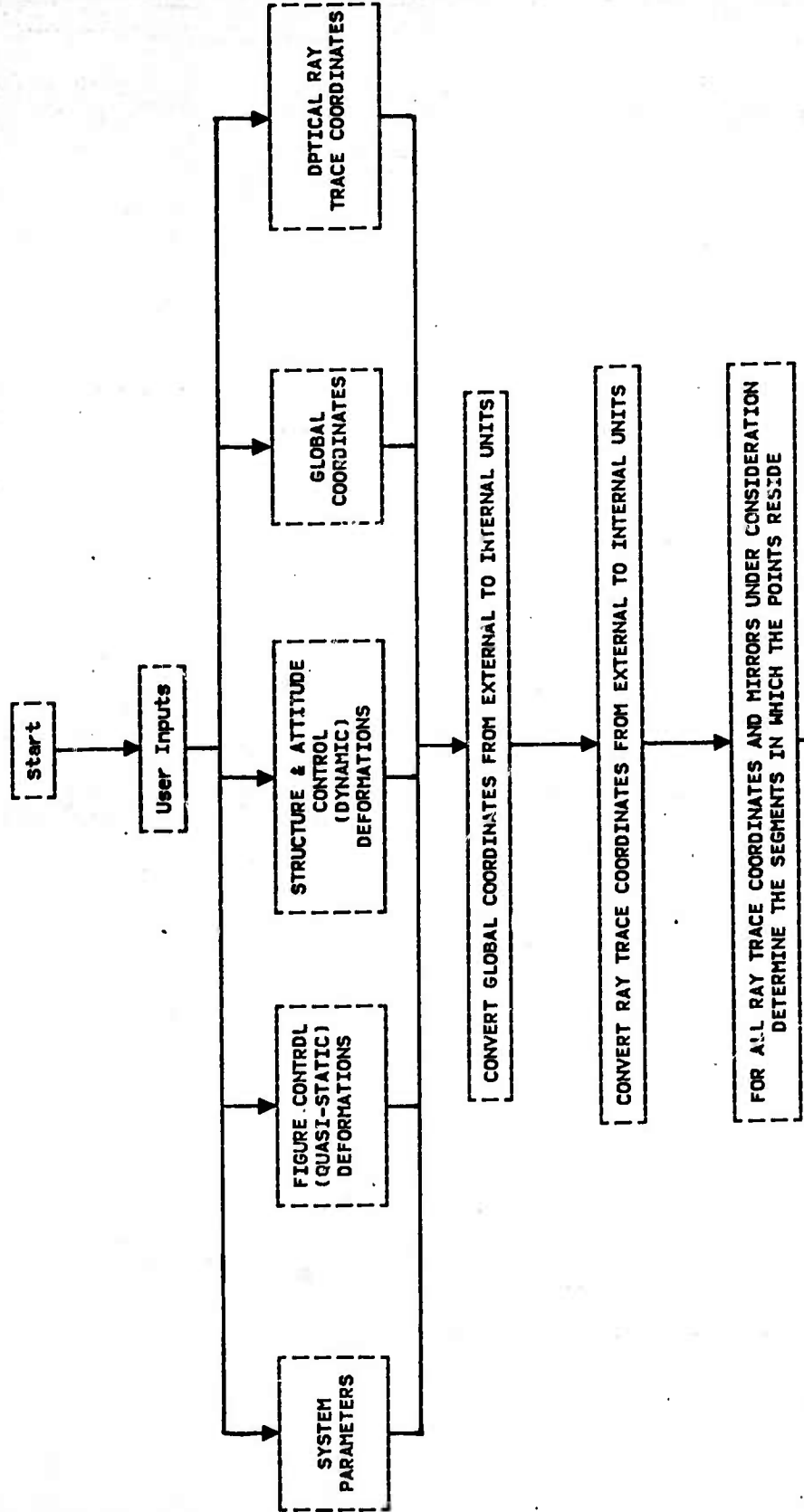
$$W_i = W^0 + \sum_{M=1}^N \pm 2 \text{COSIM} (\delta_{QM} + \delta_{DM})$$

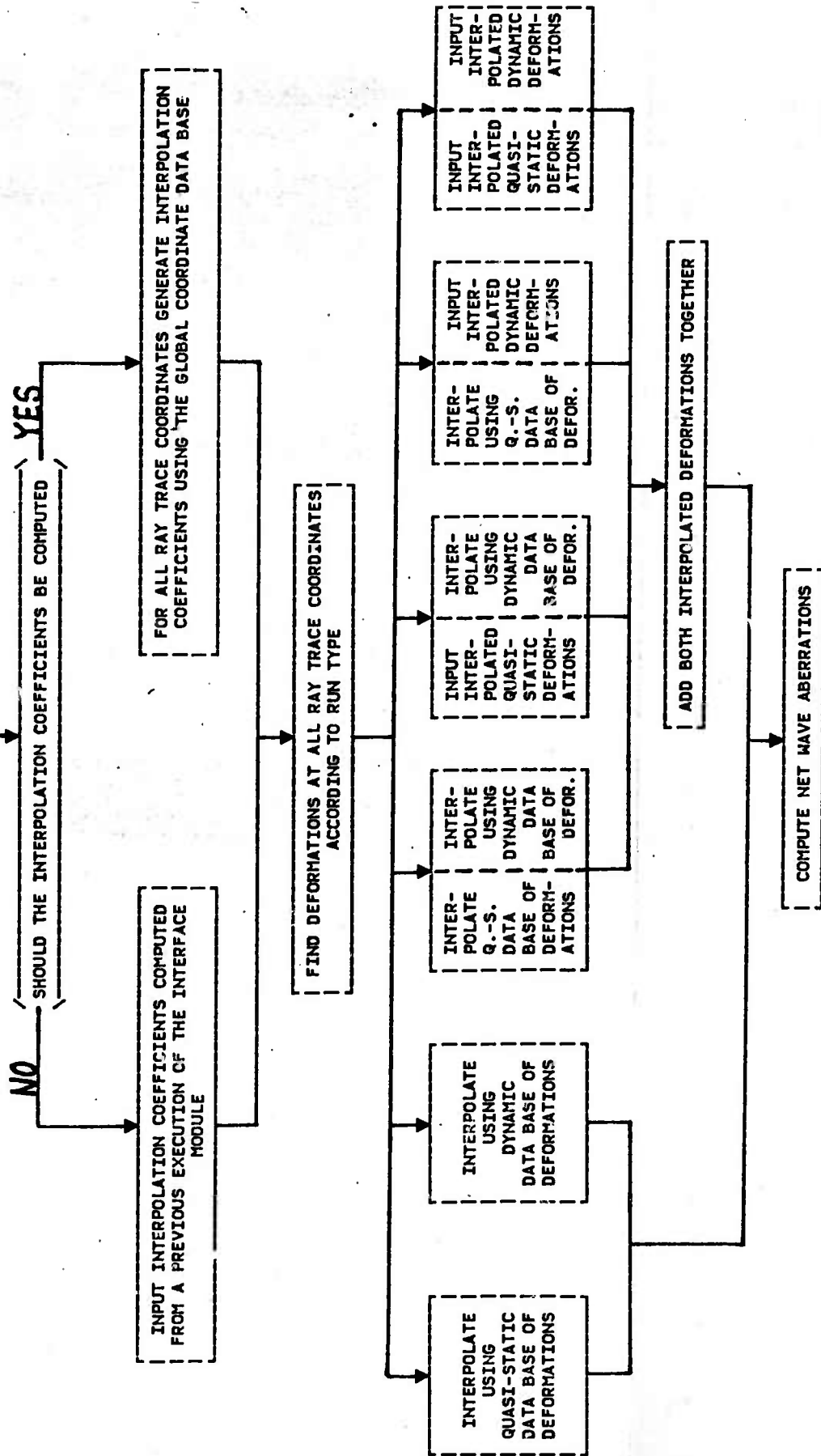
7.3 FEATURES AND CONSTRAINTS

- A. Coefficients of interpolation may be saved and reused.
- B. Interpolated deformations may be saved and reused.
- C. Cartesian or polar coordinates may be specified.
- D. Interpolation orders may be specified in both directions.
- E. There are six running options:
 - 1 - Use deformation data from the QUASI-STATIC subsystem only.
 - 2 - Use deformation data from the DYNAMIC subsystem only.
 - 3 - Use deformation data from both the QUASI-STATIC and the DYNAMIC subsystems.
 - 4 - Use deformation data from the DYNAMIC subsystem with saved interpolated QUASI-STATIC data.
 - 5 - Use deformation data from the QUASI-STATIC subsystem with saved interpolated DYNAMIC data.
 - 6 - Use saved interpolated QUASI-STATIC and DYNAMIC data.

7.4 SUBSYSTEM ARCHITECTURE

The flow of the INTERFACE subsystem is shown in Figure 7.1.

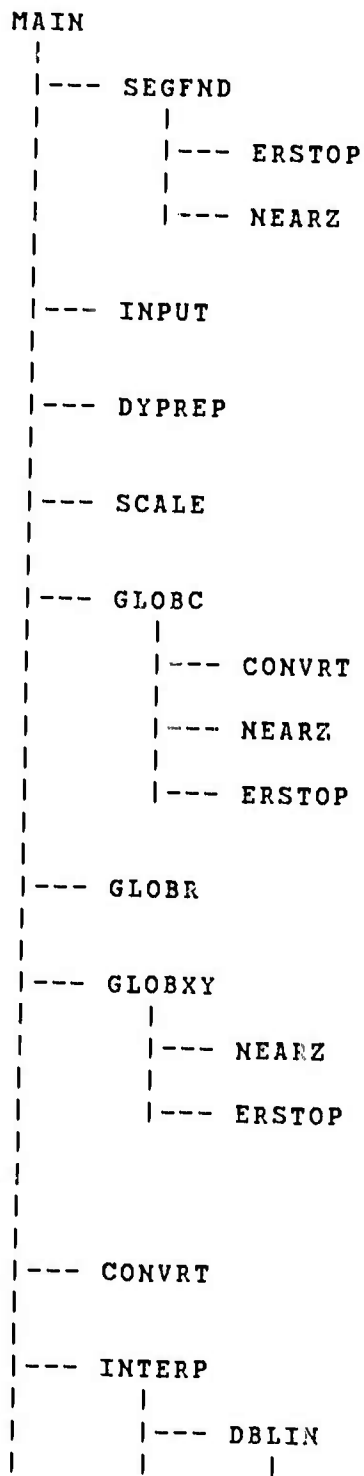




OUTPUT OPTICAL POINTS OF INCIDENCE, FOCUS POSITIONS,
WAVE ABERRATIONS AND NET WAVE ABERRATIONS

Figure 7.1 Subsystem architecture

7.5 SUBROUTINE NESTING HIERARCHY



Globr reads in the polar coordinates for one mirror.

Input reads data in namelist form. Variables which have default values may optionally be overridden by input. Variables which do not have default values must be inputted.

Interp interpolates the quasi-static displacements to get the displacement at a point on a mirror. If the coefficients of interpolation are also to be computed and filed, Dblin is used. If the coefficients are to be read in, Cint is used.

Lgrang performs Lagrangian interpolation of any order and also returns the coefficients of interpolation together with a pointer for them.

Scale computes scale factors for data on each mirror if required

Output writes the updated raytrace file for the OPTICS program.

Segfnd finds the segment of the mirror which contains a certain point.

7.7 USER INPUT SPECIFICATIONS

A. FILE NAME: IN
SOURCE: LOCAL

RECORD 1:
FORMAT: NAMELIST 'IN1'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	NM
2	I*4	MAXRT
3	I*4	NODES
4	I*4	NSEGS
5	I*4	GLFLAG
6	I*4	RTCFLG
7	I*4	RUNTYP
8	I*4	NT

RECORD 2:

FORMAT: NAMELIST 'RT'
 1 R*4 ZRI

RECORD 3:
 FORMAT: NAMELIST 'IN2'
 1 I*4 DEGR
 2 I*4 DEGTH
 3 I*4 COFLAG

RECORD 4:
 FORMAT: NAMELIST 'REW'
 1 L*4 \$REWD2

RECORD 5:
 FORMAT: NAMELIST 'DYN'
 1 L*4 DYCFLG

RECORD 6:
 FORMAT: NAMELIST 'IN3'
 1 R*4 SGN
 2 I*4 FREQ
 3 I*4 NEXT

There is one record each of types 1 and 6. There is one record each of types 2 and 5 required for dynamic runs only. There is one record of type 3 required for quasi-static runs only. There is one record of type 4 required for runs reading interpolated quasi-static data from file.

7.8 EXTERNAL STORED DATA BASES

- A. FILE NAME: FILE5
 SOURCE: CDC NASTRAN OR STARDYNE PROGRAMS

RECORD 1:
FORMAT: (F10.5)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	RC

RECORD 2:
FORMAT: (2E20.10)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	X
2	R*4	Y

There is one record of type 1 for each mirror. It is followed by NODES(S,M) records of type 2.

B. FILE NAME: FIL7
SOURCE: RAYTRACE FORMATTER

RECORD 1:
FORMAT: (3I5,F15.6)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	IND
2	I*4	IC(NRT)
3	I*4	JC(NRT)
4	R*4	T

RECORD 2:
FORMAT: (4F20.6)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	XRT(NRT,1)
2	R*4	YRT(NRT,1)
3	R*4	COSI(NRT,1)
4	R*4	XRT(NRT,2)

RECORD 3:
FORMAT: (4F20.6)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	YRT(NRT,2)

2	R*4	COSI(NRT,2)
3	R*4	XRT(NRT,3)
4	R*4	YRT(NRT,3)

RECORD 4:
 FORMAT: (4F20.6)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	COSI(NRT,3)
2	R*4	WIO(NRT)
3	R*4	XR(NRT)
4	R 4	YR(NRT)

RECORD 5:
 FORMAT: (F20.6)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	ZR(NRT)

There is one record each of types 1 through 5 for each ray.

7.9 INTERNAL STORED DATA BASES

A. FILE NAME: FIL1(unformatted)
 SOURCE: QUASI-STATIC

RECORD 1:

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	DELQ1(1)
.	R*4	.
.	R*4	.
.	R*4	.
J	R*4	DELQ1(J)

One record of type 1 is required for each segment of each mirror. Each record has NODES(S,M) fields.

B. FILE NAME: FIL3(unformatted)

SOURCE: DYNAMICS

RECORD 1:

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	XDYN(1,S,M)
2	R*4	YDYN(1,S,M)
3	R*4	ZDYN(1,S,M)

RECORD 2:

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	XDYN(2,S,M)
2	R*4	YDYN(2,S,M)
3	R*4	ZDYN(2,S,M)

RECORD 3:

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	XDYN(3,S,M)
2	R*4	YDYN(3,S,M)
3	R*4	ZDYN(3,S,M)

RECORD 4:

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	DELD1(1,S)
2	R*4	DELD1(2,S)
3	R*4	DELD1(3,S)

One record each of types 1 through 3 is required for each segment in the system. These are followed by NSEGS records of type 4 for each time-step.

C. FILE NAME: FIL2(unformatted)

RECORD 1:

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	DELQ2

NRT records of type 1 are produced for each time-step or thermal load whenever quasi-static interpolation is performed.

D. FILE NAME: FIL4(unformatted)

RECORD 1:

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	DELD2

NRT records of type 1 are produced for each time step or thermal load whenever dynamic interpolation is performed.

E. FILE NAME: FIL6(unformatted)

RECORD 1:

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	NRS
2	R*4	R(1,S)
.	R*4	.
.	R*4	.
.	R*4	.
.	R*4	R(NRS,S)
.	I*4	NTHS
.	R*4	THETA(1,S)
.	R*4	.
.	R*4	.
.	R*4	.
.	R*4	THETA(NTHS,S)

One record of type 1 is produced for each segment in the system.

F. FILE NAME: FIL8(unformatted)

RECORD 1:

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	RTR
2	R*4	RTHH
3	I*4	RTSEG

One record of type 1 is produced for each ray in the system.

G. FILE NAME: FIL9(unformatted)

RECORD 1:

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	CRTH(1)
2	R*4	CRTH(2)
.	R*4	.
.	R*4	.
.	R*4	.
.	I*4	RPT
.	I*4	THPT

One record of type 1 is produced for each ray for each mirror in the system whenever quasi-static interpolation is performed.

H. FILE NAME: FIL11(unformatted)

RECORD 1:

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	C(1)
2	R*4	C(2)
3	R*4	C(3)

One record of type 1 is produced for each ray for each mirror in the system whenever dynamic extrapolation is performed.

I. FILE NAME: FIL10(unformatted)

RECORD 1:

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	NRT
2	I*4	IC(1)
3	I*4	JC(1)
4	R*8	WIO(1)
5	R*8	XR(1)
6	R*8	YR(1)
7	R*8	ZR(1)
.	R*8	.
.	R*8	.
.	R*8	.
.	I*4	IC(NRT)
.	I*4	JC(NRT)

.	R*8	WIO(NRT)
.	R*8	XR(NRT)
.	R*8	YR(NRT)
.	R*8	ZR(NRT)

RECORD 2:

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*8	TOTW(1)
.	R*8	.
.	R*8	.
.	R*8	.
NRT	R*8	TOTW(NRT)

One record of type 1 is produced. One record of type 2 is produced for each time-step or thermal load

7.10 PLOTTING OUTPUTS

- A. Nominal wave aberrations
- B. Time-history of net wave aberrations

7.11 GLOSSARY

C(3):	Coefficients of DYNAMIC extrapolation
COFLAG:	Flag which determines whether interpolation coefficients for QUASI-STATIC interpolation will be computed or read 1 = Computed 2 = Read
COSI(ray #,mirror #):	Cosine of the angle of incidence
CRTH(DEGR*DEGTH):	Coefficients of QUASI-STATIC interpolation
DELD1(3,segment #):	Displacements of the three points defining a segment
DEGR:	Number of points needed for interpolation in R direction
DEGTH:	Number of points needed for interpolation in THETA direction
DELD1:	Displacement at the DYNAMIC support points
DELD2(ray #):	Displacements from the DYNAMICS

DELQ1:	program interpolated to the raytrace points
DELQ2(ray #):	Displacement at the QUASI-STATIC nodes
DELTH:	Displacements from the QUASI-STATIC program interpolated to the raytrace points
DYCFLG:	Displacement of a node
	Flag which determines whether extrapolation coefficients for DYNAMIC extrapolation will be computed or read
	F = Computed
	T = Read
DYFLAG:	Flag which determines formula for DYNAMIC computation
FREQ:	Frequency of print output
GLFLAG:	Global coordinate flag
	1 = Convert to polar
	2 = Read in polar
	3 = Not needed in polar (RUNTYP = 6)
	4 = Used in rectangular
IC(ray #):	I coordinate in the grid
IND:	Ray identifier
JC(ray #):	J coordinate in the grid
M:	Mirror number
MAXRT:	Maximum number of rays that will be allowed
NEXT:	Time or thermal load number of next print output
NM:	Number of mirrors
NODES(segment #, mirror #):	Number of nodes in each segment on STARDYNE tape
NRS:	Number of different radii in a segment
NRT:	Number of rays
NSEGS(mirror #):	Number of segments in each mirror
NT:	Number of time-steps or thermal loads
NTMS:	Number of different angles in a segment
R(radius # segment #):	Radius
RC:	Radius of curvature
RPT:	Pointer to array of radii for QUASI-STATIC interpolation
RCFLG:	Raytrace coordinate flag

	1 = Convert to polar
	2 = Read in polar
	3 = Not needed in polar (RUNTYP = 6)
	4 = Used in rectangular
RTR:	Radius of the raytrace point
RTSEG:	Segment number of the raytrace point
RTTH:	Angle of the raytrace point
RUNTYP:	Deformation data from
	1 = Quasi-static
	2 = Dynamic
	3 = Quasi-static and Dynamic
	4 = Quasi-static (from file) and Dynamic
	5 = Quasi-static and Dynamic (from file)
	6 = Quasi-static and Dynamic (both from file)
S:	Segment number
SGN(mirror #):	Sign of the factor to be used in computing the total wave aberration
T:	Transmission constant (may only be 1.0 or 0.0)
THETA(angle #,segment #):	Angle
THPT:	Pointer to array of angles for QUASI-STATIC interpolation
TOTW(ray #):	Wave aberration (W ₁)
WIO(ray #):	Nominal wave aberration (W ⁰)
X:	X coordinate of node
XDYN(3,segment #,mirror #):	X coordinates for the three points which define a segment in the DYNAMICS program
XR(ray #):	X-coordinate of the incidence point of the ray on the original reference sphere
XRT(ray #,mirror #):	X coordinate of ray
Y:	Y coordinate of node
YDYN(3,segment #,mirror #):	Y coordinates for the three points which define a segment in the DYNAMICS program
YR(ray #):	Y-coordinate of the incidence point of the ray on the original reference sphere
YRT(ray #,mirror #):	Y coordinate of ray
ZDYN(3,segment #,mirror #):	Z coordinates for the three points which define a segment in

ZRT(ray #, mirror #):
ZR(ray #):

\$REWD2:

the DYNAMICS program
Z coordinate of ray
Z-coordinate of the incidence
point of the ray on the original
reference sphere
Flag which is true if the
interpolated quasi-static file
is to be rewound and reread

8. OPTICS

8.1 PURPOSE

This program computes the point-spread-function of the deformed system and generates the line-of-sight and amount of signal energy reaching a detector element as a function of time in order to evaluate the optical performance.

8.2 TECHNICAL SUMMARY

The OPTICS subsystem computes the following:

- Pupil function, F
- Irradiance point-spread-function, I
- Ensquared power, P

for every α (field angle) in the system. These computations are performed for the:

- Nominal or disturbance free aberrations, W^0 with respect to nominal focus.
- Net wave aberrations, W_I (including time-history thermal and/or dynamic deformations) with respect to nominal focus.
- Net wave aberrations, W_F with respect to a new reference sphere centered at a best focus for which the variance of the wavefront error is minimum.

For each time and field angle the following are also computed:

- Transverse and longitudinal defocus
- LOS error
- Average variation in the ensquared power for both W_I and W_F .

The pupil function is computed as follows for $W=W^0(\alpha)$,

$W_I(\alpha, t)$, and $W_F(\alpha, t)$ to obtain $F^0(\alpha)$, $F_I(\alpha, t)$ and $F_F(\alpha, t)$ respectively.

$$F(u, v) = \begin{cases} \text{EXP}[(2\pi i/\lambda) W(u, v)] & \text{Inside Exit Pupil} \\ 0 & \text{Outside Exit Pupil} \end{cases}$$

where λ is the optical wavelength.

The irradiance point-spread-function is computed as follows for $F = F^0(\alpha)$, $F_I(\alpha, t)$ and $F_F(\alpha, t)$ to obtain $I^0(\alpha)$, $I_I(\alpha, t)$ and $I_F(\alpha, t)$ respectively.

$$I(x, y) = (1/\lambda^2 R^2) \left| \iint F(u, v) \text{EXP}[-2\pi i/\lambda R (xu + yv)] du dv \right|^2$$

where (x, y) is in the Image Plane
and (u, v) is in the Exit Pupil Plane.

The above integral is a Fourier transform of the pupil function, and R is the radius of the reference sphere.

The ensquared power of the PSF is computed as follows for $I = I^0(\alpha)$, $I_I(\alpha, t)$ and $I_F(\alpha, t)$ to obtain $P^0(\alpha)$, $P_I(\alpha, t)$ and $P_F(\alpha, t)$ respectively.

$$P = f \iint I(x, y) dx dy$$

where the integration is performed over the desired detector, and then normalized by a factor f .

The average variation in the ensquared power is computed as follows for $P = P_I(\alpha, t)$ and $P_F(\alpha, t)$ to obtain $\Delta P_I(\alpha)$ and $\Delta P_F(\alpha)$ respectively.

$$P = (1/L) \sum_{t=1}^L P - P^0$$

The transverse and longitudinal defocus is computed as follows for $r_F(\alpha, t)$ to obtain $\Delta r(\alpha, t)$

$$\begin{cases} \Delta r_X = r_{FX} - r_X^0 \\ \Delta r_Y = r_{FY} - r_Y^0 \end{cases} \text{ transverse}$$

$$\Delta r_Z = r_{FZ} - r_Z^0 \text{ longitudinal}$$

The line-of-sight error ($LOS(\alpha, t)$) is computed from $\bar{\Delta r}(\alpha, t)$ as follows:

$$LOS = (\Delta r_x^2 + \Delta r_y^2)^{.5}/f$$

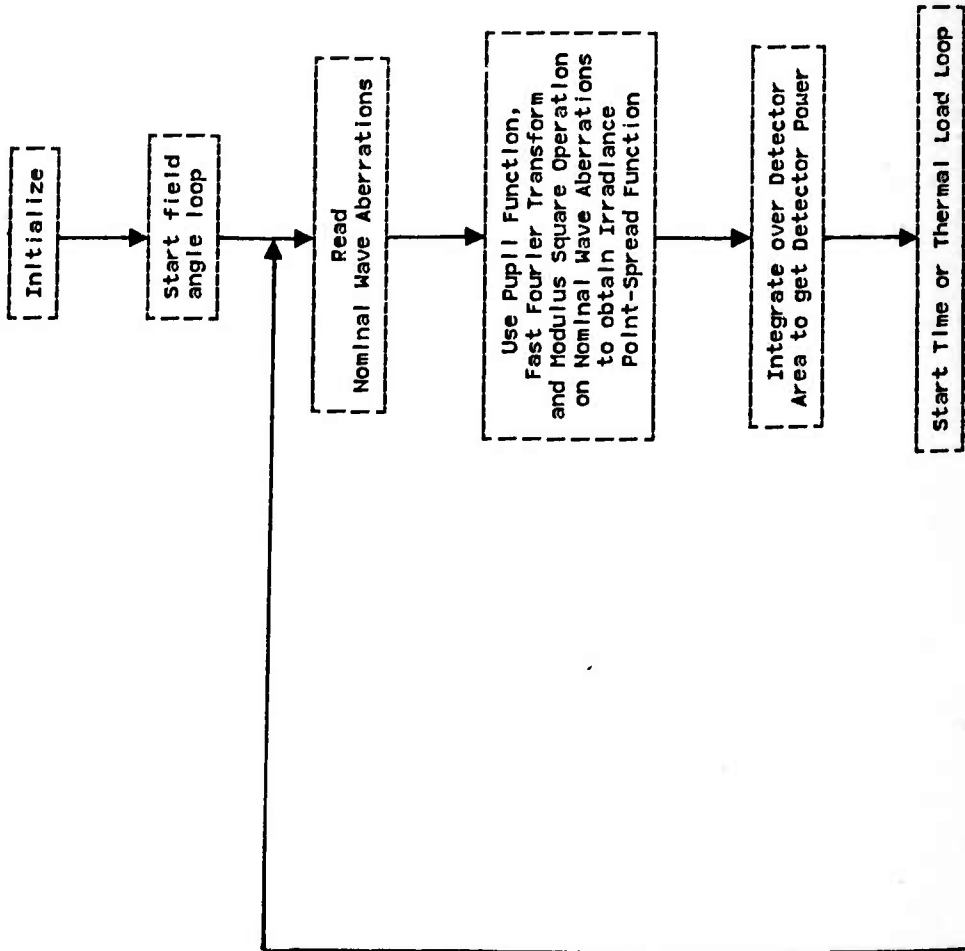
where f is the system focal length.

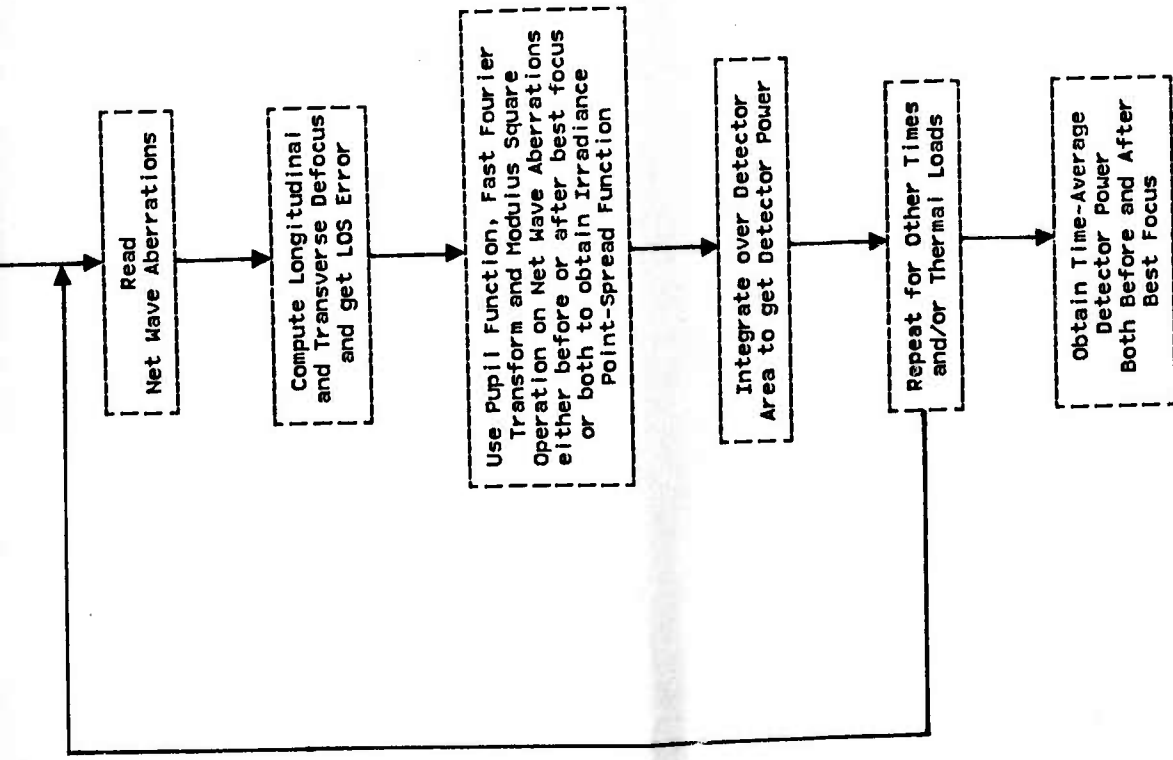
8.3 FEATURES AND CONSTRAINTS

- A. Multiple field angles
- B. Dynamic point-spread-function
- C. Matrix of data points representing the pupil function must be a rectangle with dimensions equal to powers of two
- D. Clear region of pupil may be any size or shape and be placed anywhere in the exit pupil grid.
- E. The detector must be square. It may be of any size and positioned anywhere in the image plane grid.
- F. Multiple detectors may be employed, one at a time, to analyze the data.

8.4 SUBSYSTEM ARCHITECTURE

The flow of the OPTICS subsystem is shown in Figure 8.1.





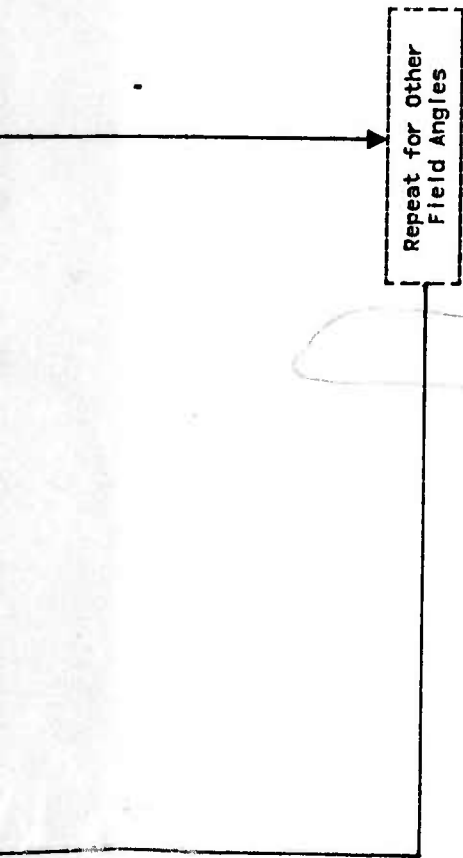


Figure 8.1 Subsystem architecture

2

3

8.5 SUBROUTINE NESTING HIERARCHY

MAIN

```
|--- INPUT
|
|--- INPUTA
|
|--- FILL
|   |
|   |--- FUNC
|   |
|   |--- CMOVER
|
|--- FFT
|   |
|   |--- REVBIT
|
|--- MAPR
|
|--- OUT
|   |
|   |--- SOUT
|
|--- INTSET
|   |
|   |--- INTEG
|
|--- OUT1
|
|--- DMOVER
|
|--- BFOC
|   |
|   |--- DMOVER
|   |
|   |--- CROUT
|   |
|   |--- ERSTOP
```


Func(ARG) computes and returns:
(COS(ARG),i SIN(ARG))

Input reads invariant input.

Integ(DATA,ANS,NINT,N,XCTR,YCTR) integrates a two-dimensional array (DATA) in the square extending NINT units in both directions from the point (XCTR,YCTR).

Intset(AIN,ANS,RADIUS,XCTR,YCTR,C) computes the radius of the square-of-integration and scales the results of integration by C.

Mapr(CIN,ROUT,N,NPT) maps an N×N complex array into alternating elements of a 2×N×N real array by squaring the absolute value of each element and inserting it into the same position of a different quadrant as follows:

I <--> IV
II <--> III

Out(ANS,N) Outputs alternate elements of a 2×N×N real array.

Out1(P,N) outputs a linear array of length N.

Out2(ANS,N,XCTR,YCTR,R) outputs the portion of an N×N array(ANS) lying within the square extending R units from (XCTR,YCTR).

Sout(ANS,N) outputs the real single-precision portion of a complex double-precision vector(ANS).

8.7 USER INPUT SPECIFICATIONS

A. FILE NAME: IN
SOURCE: LOCAL

RECORD 1:
FORMAT: NAMELIST 'IN1'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*8	F
2	R*8	LAMBDA
3	I*4	N
4	I*4	NALPHA

RECORD 2:
FORMAT: NAMELIST 'IN1A'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	FREQ
2	I*4	FIRST
3	I*4	LEVEL

RECORD 3:
FORMAT: NAMELIST 'IN2'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	NT
2	I*4	XOS
3	I*4	YOS
4	I*4	R
5	I*4	OPT
6	I*4	NIN
7	I*4	NR
8	L*4	W0

RECORD 4
FORMAT: NAMELIST 'IN3'

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*8	NP
2	R*8	R1
3	R*8	XP
4	R*8	YP
5	R*8	ZP
6	L*4	\$DELP

There is one each of Records 1 and 2. One record each of types 3 and 4 are required for each field angle.

B. FILE NAME: FILE5
SOURCE: LOCAL

RECORD 1:
FORMAT: (3D20.12)

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*8	DELP(1)
2	R*8	DELP(2)
3	R*8	DELP(3)

There is one Records 1 for each time at each field angle.

8.8 INTERNAL STORED DATA BASES

A. FILE NAME: FIL1(unformatted)
SOURCE: INTERFACE

RECORD 1:

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	I*4	NRT
2	I*4	IR(1)
3	I*4	JC(1)
4	R*8	WI0(1)
5	R*8	XR(1)
6	R*8	YR(1)
7	R*8	ZR(1)
.	R*8	.
.	R*8	.
.	R*8	.
.	I*4	IR(NRT)
.	I*4	JC(NRT)
.	R*8	WI0(NRT)
.	R*8	XR(NRT)
.	R*8	YR(NRT)
.	R*8	ZR(NRT)

RECORD 2:

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*8	WI(1)
.	R*8	.
.	R*8	.
.	R*8	.
NRT	R*8	WI(NRT)

One record of type 1 is required for each field angle. This must be followed by NT records of type 2.

B. FILE NAME: FIL2(unformatted)

RECORD 1:

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*4	DATASQ(1,1)
.	R*4	.

.	R*4	.
.	R*4	.
N	R*4	DATASQ(N,1)
.	R*4	.
.	R*4	.
.	R*4	.
.	R*4	DATASQ(N,N)

If OPT is 1 or 2, then the number of records produced is $NT*NALPHA+1$. If OPT = 3, then the number of records produced is $2*NT*NALPHA+1$.

C. FILE NAME: FIL3(unformatted)

RECORD 1:

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*8	ELOS

The number of records produced is $NT*NALPHA$.

D. FILE NAME: FIL4(unformatted)

RECORD 1:

<u>FIELD</u>	<u>TYPE</u>	<u>NAME</u>
1	R*8	P(1)
.	R*8	.
.	R*8	.
.	R*8	.
N	R*8	P(N)

If OPT is 1 or 2, then the number of records produced is $NR*(NALPHA*NT+NALPHA+1)$. If OPT = 3, then the number of records produced is $2*NR*(NALPHA*NT+NALPHA+1)$.

8.9 PLOTTING OUTPUTS

- A. A Point-spread-function for each time step.
- B. Time history of line-of-sight errors.
- C. Time history of ensquared power for each detector.

8.10 GLOSSARY

DATASQ(N,N):	Point-spread-function
DELP(3):	Displacement of best focus point from the old reference point
F:	System focal length
FIRST:	Time-step of first printout
FREQ:	Frequency of print output
IR(ray #):	I coordinate in the exit pupil grid
JC(ray #):	J coordinate in the exit pupil grid
LAMBDA:	Optical wavelength
LEVEL:	Amount of printout desired 1 - Minimum 2 - All except mod ² data 3 - All
N:	Size of the exit pupil grid (must be a power of 2)
NALPHA:	Number of field angles
NIN:	Grid size of the portion of the exit pupil containing rays
NP:	Index of refraction in the image space
NR:	Number of radii to be used for integration
NRT:	Number of rays
NT:	Number of time-steps or thermal loads
OPT:	Run type option 1 - Use wave aberrations as read 2 - Use wave aberrations after best focus 3 - Use both wave aberrations 4 - Use neither; compute best focus only
P:	Energy on the detector
R(radius #):	Radius of detector
R1:	Radius of the original reference sphere
WI(ray #):	Wave aberration at each time-step
WIO(ray #):	Nominal wave aberration
W0:	Flag which determines whether to bypass reading wave aberrations (use only nominal when true)
XOS(radius #):	X-direction offset of the center of integration
XP:	X-coordinate of the old reference ray point
XR(ray #):	X-coordinate of the incidence point of the ray on the original reference sphere
YOS(radius #):	Y-direction offset of the center of integration
YP:	Y-coordinate of the old reference ray point
YR(ray #):	Y-coordinate of the incidence point of the ray on the original reference sphere

ZP: Z-coordinate of the old reference ray point
ZR(ray #): Z-coordinate of the incidence point of the
ray on the original reference sphere
\$DELP: Flag which determines whether the
displacement of the best focus point from
the old reference point will be computed or
read
F - computed
T - read

9. SAMPLE TEST CASE

9.1 MODEL DESCRIPTION

A sample problem is presented to illustrate the flow of data through the different modules of the simulation. A simplified model of a typical large space structure is used. This system, shown in Figure 9.1, contains three mirrors. The primary and tertiary mirrors each have one segment and the secondary mirror has three segments. The mirror support structure is made up of graphite epoxy hollow tubes. A finite element model was created and analyzed using NASTRAN in order to determine the natural frequencies and mode shapes of the structure. These are listed in Table 9.1. The system was subjected to a dynamic disturbance on the secondary support truss and analyzed using the DYNAMICS, INTERFACE, and OPTICS modules. The local input to the DYNAMICS module is shown in Tables 9.2 and 9.3. The local input to the INTERFACE and OPTICS modules are shown in Tables 9.4 and 9.5 respectively.

The output of the system illustrates the degradation of the image due to the system disturbances. Plots of the nominal point-spread-function and the point-spread-function at 0.0040 seconds are shown in Figure 9.2.

The CPU time and storage requirements of each module are listed in Table 9.6.

ROTATIONS X= 0.0 Y= 0.0 Z= 0.0
 UIEU DIR X= 0.0 Y= -1.00 Z= 0.0
 UP DIR X= 0.0 Y= 0.0 Z= 1.00
 UIEU DIST = 100.00
 SCALE FACTOR= 0.79692E+00
 CENTER AT X= 0.0 Y= 0.0 Z= 2.14
 FRAME SIZE X= 8.00 Y= 8.00

OPTIONS: SCALE, RESET, INCLUDE, EXCLUDE, ROTATE, UIEU, READ,
 PERSPECTIVE, DISTORT, DASHED, CENTER, AXES, END

NEXT

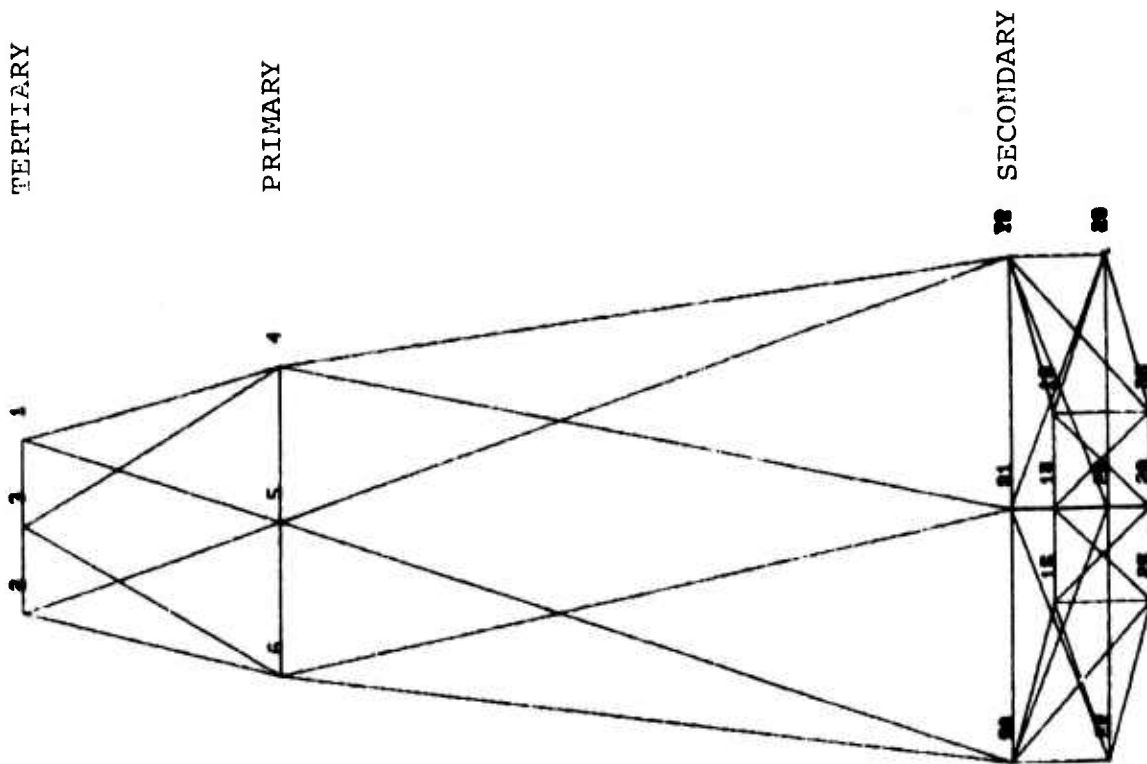


Figure 9.1 Structure Model

Table 9.1 Natural Frequencies and Mode Shapes

Mode	Freq(Hz)	Description
1 - 6	0.0	Rigid Body
7	43.96	} Bending of Secondary Mirror
8	44.08	
9	65.45	
10	73.90	
11	73.92	
12	98.13	Torsion

Table 9.2 Dynamic local inputs

Computer Input	Value	Definition	
MD1	6	No. of nodes	
N16	13	No. of nodes	
MNAM1	7,8,9,10,11, 12	Mode numbers to be used in analysis	
NNAM1	1,2,3,4,5,6, 7,8,9,10,11, 12,22	Node numbers to be used in analysis	
NORM3	TRUE	Flag to indicate that normals at mirror support points follow	
3	4	5 6	
	-0.4152	0.7193	-0.5573
	0.0	0.8303	-0.5573
	-0.4152	-0.7193	-0.5573
3	7	12 8	
	-0.1687	-0.2922	0.9413
	0.1687	-0.2922	0.9413
	-0.3375	0.0	0.9413
3	10	11 12	
	0.1687	0.2922	0.9413
	0.3375	0.0	0.9413
	0.1687	-0.2922	0.9413
3	9	8 10	
	-0.1687	0.2922	0.9413
	-0.3375	0.0	0.9413
	0.1687	0.2922	0.9413
3	1	2 3	
	0.0	0.0	-1.0
	0.0	0.0	-1.0
	0.0	0.0	-1.0
999			

Table 9.3 Dynamic local inputs (continued)

Computer Input	Value	Definition
PSI1P	6*.005	Damping Ratio per mode
TO	0	Initial time
TOPT	0	Initial time to output to file
DTOPT	0.0016	Time interval for file ouput
TF	0.024	Final time
TPRINT	0.0	Initial printout time
DTPRNT	0.0016	Time interval for printout
CNTL	FALSE	No structure control present
ATT	FALSE	No attitude control present
DTP	0.0008	Plant time-step
PO1	0.5	Amplitude of forcing function
OMEGA1	464.32	Frequency of forcing function
NDIS1	1	No. of disturbances present
IDIS1	22	Node number indicating where disturbance should be located
ITUP1	3	Degree-of-freedom where disturbance occurs

Table 9.4 Interface local inputs

Computer Input	Value	Definition
NM	3	No. of mirrors
MAXRT	12	No. of raytrace points
NODES	5*4	No. of nodes used to define each segment
NSEGS	1,3,1	No. of segments in each mirror
GLFLAG	1	Convert segment definitions to polar coordinates
RTCFLG	1	Convert rayTrace data to polar coordinates
RUNTYP	2	Type of input: 2 = Dynamics
NT	15	No. of time-steps on input file
FREQ	1	Frequency of printout
NEXT	1	First time-step to be printed
SGN	3*-1.0	Sign of factor for total wave aberrations

Table 9.5 Optics local inputs

Computer Input	Value	Definition
F	-10.0	System focal length
LAMBDA	1.0E-6	System wavelength
N	64	Size of Detector
NALPHA	1	No. of field angles
FREQ	1	Frequency of printout
NEXT	1	First time-step for printout
LEVEL	3	Level of output: Level 3 = PSF for input wave aberrations and best focus
NT	5	No. of time-steps
XOS, YOS	0.0, 0.0	Offsets for center of integration
R	1	Integration radius
NR	1	Number of radii
NIN	5	Raytrace grid size in each direction
WO	FALSE	Read nominal wave aberrations
R1	-5.33	Radius of the reference sphere
XP, YP, ZP	0.0, 0.0, -5.33	Coordinates of old reference ray point

NOMINAL

DEFORMED

$t = 0.004 \text{ s}$

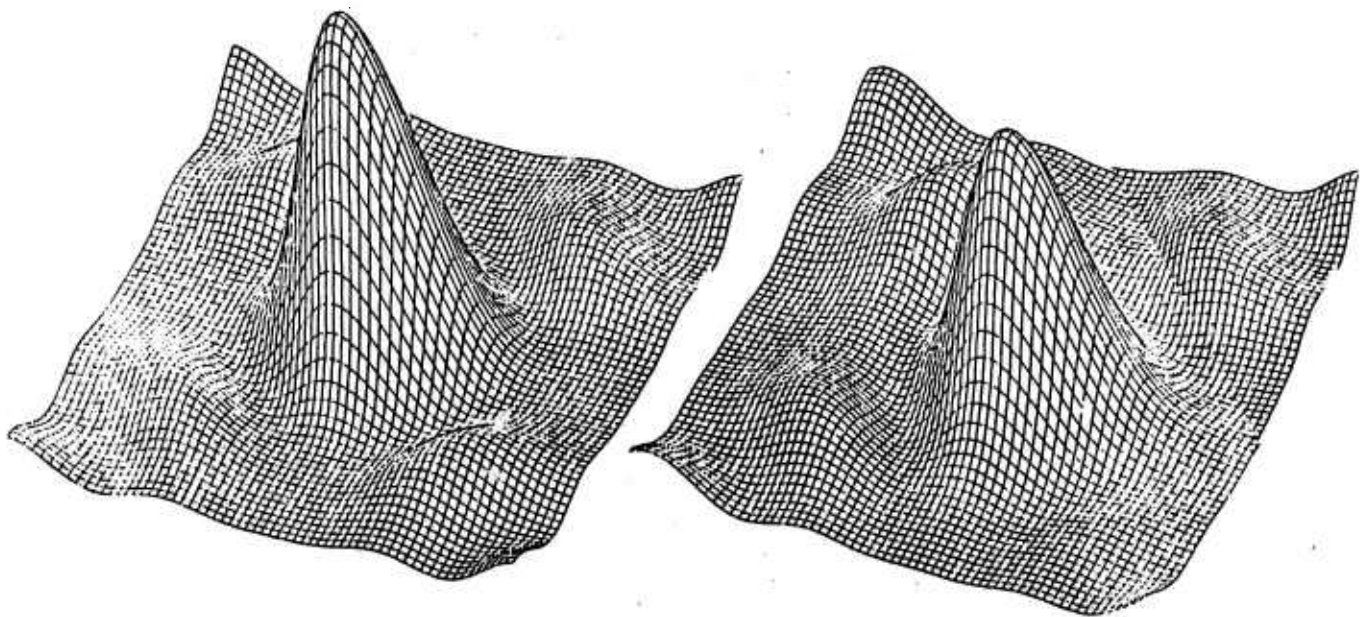


Figure 9.2 Plots

Table 9.6 CPU time and storage requirements

	Time (sec)	Storage (K bytes)
DYNAMICS	1.40	112
INTERFACE	2.31	144
OPTICS	36.29 ¹	172

¹ The PSF was calculated for nominal and best focus at each time-step.