# A NASTRAN/TREETOPS Solution to a Flexible, Multi-Body Dynamics and Controls Problem on a UNIX Workstation

Javier E. Benavente and Norris R. Luce
Dynacs Engineering Co., Inc.
Clearwater, Fla.

N89-22945

## SUMMARY

Demands for non-linear time history simulations of large, flexible multi-body dynamic systems has created a need for efficent interfaces between finite-element modeling programs and time-history simulations.

One such interface, TREEFLX, an interface between NASTRAN and TREETOPS, a non-linear dynamics and controls time history simulation for multi-body structures, is presented and demonstrated via example using the proposed Space Station Mobile Remote Manipulator System (MRMS).

The ability to run all three programs (NASTRAN, TREEFLX and TREETOPS), in addition to other programs used for controller design and model reduction (such as DMATLAB and TREESEL, both described in this paper), under a UNIX Workstation environment demonstrates the flexibility engineers now have in designing, developing and testing control systems for dynamically complex systems.

## INTRODUCTION

Traditionally, the "Modern" control design process has begun with a linearized representation of a model. From this point several paths may be taken to derive gains that form the basis of a feedback control system.

Many tools exist today that facilitate this control design process. One such tool, DMATLAB, accepts the model via the (A,B,C,D) matrices defined by;

$$\dot{x}(t) = A\,x(t) + B\,u(t)$$

$$y(t) = C\,x(t) + D\,u(t)$$

| | | | |
|---|---|---|---|
| where: | x(t) | is the state vector | $x \in \mathbb{R}^{nx}$ |
| | u(t) | is the input vector | $u \in \mathbb{R}^{nu}$ |
| | y(t) | is the output vector | $y \in \mathbb{R}^{ny}$ |
| | t | represents time | $t \in \mathbb{R}$ |
| | A | is the state matrix | $A \in \mathbb{R}^{nx \cdot nx}$ |
| | B | is the control matrix | $B \in \mathbb{R}^{nx \cdot nu}$ |
| | C | is the state output matrix | $C \in \mathbb{R}^{ny \cdot nx}$ |
| | D | is the control output matrix | $D \in \mathbb{R}^{ny \cdot nu}$ |

The question arises, where do the (A,B,C,D) system matrices come from? TREETOPS, a non-linear time history simulation for multi-body systems with active control elements, answers this question via a linearization option which produces the (A,B,C,D) matrices as an output.

TREETOPS numerically derives the equations of motion of systems based on a user defined topology. For rigid systems, the process is simple. The mass and inertia properties of each rigid body in the system is specified, along with node point geometry. The relationship between the bodies is specified by defining hinges. Sensors and actuators are easily included, along with controllers and other simulation elements.

For flexible bodies the topology is defined in a similar manner; however, additional modal data is needed for TREETOPS to accurately simulate the flexible system response [ref. 1]. Until recently, this flex data had to be generated off line and in a form compatible with TREETOPS.

The development of TREEFLX has allowed the use of NASTRAN to generate flexible models of the individual bodies represented in the TREETOPS system. TREEFLX utilizes the NASTRAN data to generate all of the terms required by TREETOPS to simulate the time-history response of a flexible, multi-body dynamic system.

This paper demonstrates the general modeling and control design process and the role NASTRAN plays within it. The paper is organized as follows. First, some comments on system observability/controllability and reduced order controller design is presented, along with comments on a general control design procedure. Next, the topology of the system of interest is presented and a rigid model of the system is developed to facilitate controller design. The controller is derived based upon the rigid system. With this analysis complete, the bodies are modeled as flexible via NASTRAN. For computational considerations, component model reduction is performed on the flexible model. The reduced order model is used to evaluate the controller designed with the rigid system.

It should be emphasized that all of the analysis, modeling and design work for this paper was completed on UNIX Workstations, namely, a SUN 3/60 and Silicon Graphics Personal IRIS Workstation. The ability for an engineer to model a complex multi-body flexible system with a complete version of NASTRAN, design a controller for that system and simulate the non-linear closed-loop time history on a relatively inexpensive UNIX Workstation is a major advancement in computer aided engineering analysis and design.

## CONTROL DESIGN CONCEPTS

It is generally acknowledged by control designers that the model and control design processes are inseperable. Indeed, Skelton [ref. 2] refers to this as the Modeling and Control Inseparability Principle. Simply put, the modeling and control designs are necessarily iterative.

Often, simple models of a physical systems are employed to facilitate the control design. As an example, consider a single beam modeled as a flexible body by NASTRAN. Suppose 20 flexible modes are retained for the TREETOPS representation of this beam, and that one rigid rotational degree of freedom is provided by a pinned hinge. Further, suppose that two sensors, one to measure the hinge angle and the other to measure the rate of change of the angle, along with a torque actuator, are co-located at this hinge. The linearized TREETOPS state matrix would be size 42 by 42. Controllability and observability (in the sense of a Linear Quadratic (LQ) control design) is certainly not guaranteed and probably not likely.

Now consider the body as rigid, with a stiff spring placed at the pinned hinge to approximate the body's flexibility. For this model the TREETOPS state matrix is size 2 by 2. In general, observability is guaranteed and controllability is more likely; a solution to the LQ design problem is, in general, easier to obtain with simpler models.

As Anderson and Liu mention [ref. 3], the above process is in reality a crude, yet sometimes successful, approach to controller reduction.

A logical question to be asked here is: How does the performance of a controller based on a simple model of a system change when applied to a more complex representation of the same physical phenomenon?

Figure 1 shows the control design process used in this paper. Below each process block is the name of the computer program(s) utilized in this paper to accomplish the process' objectives. Figure 2 shows the general relationship and interaction between these programs as implemented in a UNIX Workstation environment.

This paper demonstrates the design process of Figure 1 by example. A simplified lumped flexibility model of the MRMS is developed to form the basis of an LQ controller. Once settled upon, this controller is applied to a more complex system derived from NASTRAN models. Performance characteristics are compared between the two models.

## MODEL TOPOLOGY

Figure 3 shows the general topology of the system of interest in this paper, a model of the MRMS. Represented is a 4-body system, the first and fourth bodies both being rigid, the second and third bodies both flexible.

Two sensors each are located at the second and third hinges. The first sensor measures the Euler angle between each hinge's inboard and outboard body, the second sensor measures the rate of change of the angle. A torque motor actuator is co-located with the sensors at both of the hinges. For simplicity, only one rotational degree of freedom is modeled at both the second and third hinge. All other hinges are locked. Physical properties of the individual bodies are summarized in Table 1.

The control design objective is to minimize perturbations from the initial conditions of the Euler angles, as measured by the sensors at the second and third hinges, in the presence of a disturbance. The disturbance is modeled within TREETOPS by a non-periodic pulse acting at the end of the third body.

## CONTROLLER DEVELOPMENT

To facilitate control design, a lumped flexibility model is developed with the aid of TREESET and TREETOPS. This lumped flexibility model treats each body as being rigid; flexibility is modeled by lumping the body flexibility at the hinges with stiff springs.

The lumped flexibility model is entered into TREETOPS via TREESET, an interactive setup program. The linearization option is chosen and the simulation is set to run for one time step. Running the TREETOPS simulation results in a file containing the linear, time-invariant (A,B,C,D) matrices for the lumped flexibility model. This process is equivalent to Step 1 of Figure 1. Table 2 list the numerical values of the (A,B,C,D) matrices as output from TREETOPS' linearization option.

These matrices are entered into DMATLAB. DMATLAB provides many controls analysis and design tools for both the "classical" and "modern" controls designer. DMATLAB is used to design a controller based on the lumped flexibility model. This is equivalent to Step 2 of Figure 1.

Feedback control gains are obtained via an LQ control design based on full-state feedback. The state vector is defined by the two Euler angles and their rates. If we represent the angles by $\theta_2$ and $\theta_3$, their rates by $\dot{\theta}_2$ and $\dot{\theta}_3$ and the controller output (actuator commands) as $u_1$ and $u_2$ then the control law gain is a matrix G such that ;

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} G \end{bmatrix} \begin{bmatrix} \dot{\theta}_2 \\ \dot{\theta}_3 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

The numerical values for G will be found in Table 2.

TREESET is used to define a continuous matrix controller for the TREETOPS simulation. Interconnects are established between the sensors and the actuators, scaled by the gains determined in DMATLAB. This forms the basis of a continuous feed-back control system for the non-linear, time-history TREETOPS simulation. This simulation is the equivalent of Step 3 in Figure 1.

For complicated systems, an iteration for the controller gains will probably be required; indeed, the final controller gains for this paper were selected only after several such iterations.

## NASTRAN MODEL

Since no official configuration for the MRMS has been established, the NASTRAN models are based on Space Shuttle Remote Manipulator System (SRMS) data [ref. 4].

The NASTRAN flex data is generated using CBAR elements to represent the mass and stiffness properties of SRMS body elements. Fixed-free boundary conditions were chosen for each body. Standard Solution 3 (Normal Modes Analysis), with an DMAP alter added for the additional output required by TREEFLX, is utilized. A separate OUTPUT5 file is generated for each flexible body in the topology. The development of the NASTRAN model is equivalent to Step 4 of Figure 4.

TREEFLX, based on the NASTRAN data for each body, calculates all of the required and optional modal data for the TREETOPS simulation. Table 3 summarizes the modal terms presently generated by TREEFLX and used by TREETOPS.

To generate this data, TREEFLX requires the NASTRAN Nodal Mass, Eigenvectors, Modal Mass, Modal Stiffness and, if available, Modal Damping matrices. In addition, a matrix consisting of NASTRAN Grid Point Location vectors, expressed in global coordinates, is required. The process of converting NASTRAN output data to TREETOPS input data with TREEFLX is represented by Step 5 of Figure 1.

A major assumption in TREEFLX is that the TREETOPS and NASTRAN models use the same coordinate system for each individual body. Based on this assumption, it is not necessary to designate the TREETOPS node location with coordinates during the TREESET setup procedure, but rather, the user designates a corresponding NASTRAN grid point ID for each TREETOPS node. TREEFLX uses this node/grid point correspondence to develop the TREETOPS nodal geometry. Not all NASTRAN grid points have to be included in the TREETOPS model. TREETOPS nodes are required only as hinge attach points, sensor and actuator locations and for mass centers.

An important distinction must be made at this point. Notice in Table 3 that several TREETOPS terms are calculated with summations over the total number of nodal bodies in the model. Even though all nodes may not be included in the final TREETOPS data file, the TREEFLX nodal summations are made over the entire set of NASTRAN grid points supplied in the NASTRAN OUTPUT5 file, not just over the sub-set of retained TREETOPS nodes.

## COMPONENT MODE MODEL REDUCTION

An optional step may be inserted between Steps 5 and 6 of Figure 1. Suppose the complex model developed by NASTRAN includes 100 modes for each body, yet it is determined that a model with 47 modes for each body is sufficient for an accurate time-history simulation (this paper does not propose any method for this determination). This implies that a model reduction procedure might be inserted at this point of the design process.

TREEFLX provides for model reduction with a simple mode selection technique. If model reduction is indicated, TREEFLX searches for a file that lists the modes that should be retained for each individual body.

A natural question arises: Which modes should be retained in a reduced order model? This paper does not present a theoretical discussion of component mode model reduction procedures; however, TREESEL, a TREETOPS companion program, can assist in the answering of the above question.

TREESEL uses several methods to rank the relative importance of the system modes. One method, used in this paper to reduce the MRMS NASTRAN model, is the Modified Component Cost method.

The Component Cost method is based on the assumption that each state contributes to a cost function, $\mathcal{V}$, defined by the model designer. By decomposing the cost function into its components, the relative contribution of each system state to the cost function can be ranked.

In TREETOPS, each degree of freedom is a state. A beam with N flexible modes will have at least 2*N states, 2 states for each mode. TREESEL ranks these modal degrees of freedom in a concise form. Once ranked, the number of modes to be retained depends on the open loop performance matching the analyst would like to obtain. An iterative process of selecting modes is usually required to obtain a suitable reduced order model. Table 4 lists the NASTRAN modes as ranked by TREESEL. This ranking represents only a single iteration with TREESEL using simply selected weights.

To demonstrate TREESEL model reduction techniques, the five highest ranked modes for each body (10 system modes) were retained for the "complex" NASTRAN/TREETOPS model.

Step 5 of Figure 1 is accomplished by merging the NASTRAN/TREETOPS model with the continuous matrix controller gains derived earlier. TREETOPS is used to simulate the closed-loop time-history response of the system.

## RESULTS

Figures 4 and 5 show the results of Steps 1, 2 and 3 of Figure 1. Plotted is the time-history of the uncontrolled vs controlled hinge Euler angles and rates for the lumped flexibility model. The results shown were considered adequate to accept the controller design.

Figures 6 and 7 show the results of component mode model reduction using the five highest body modes ranked by TREESEL. Shown are the uncontrolled hinge Euler angles and rates for the full-order (20 modes) and reduced-order (5 modes) NASTRAN model. The TREESEL ranking was obtained with just one run of the program and simply selected weight were used. The results seem to indicate that reduced order models of higher order systems can approximate the higher order system's uncontrolled response.

Figures 8 and 9 show the results of Step 6 of Figure 1. Plotted is the time-history of the uncontrolled vs controlled hinge angles and hinge angle rates for the NASTRAN reduced-order model. Figures 10 and 11 compare the uncontrolled responses of the Lumped Flexibility and NASTRAN reduced order models. Figures 12 and 13 compare the controlled responses of the Lumped Flexibility and NASTRAN reduced order models. The results indicate that, for some systems, controllers designed on the basis of simplified models of complicated systems do perform adequately on higher fidelity models of the same system.

Figure 14 compares the actuator commands (controller outputs) of the Lumped Flexibility and NASTRAN reduced order models.

## CONCLUSIONS

This paper demonstrates the use of NASTRAN and a UNIX Workstation environment in the system modeling/control design process. An automated design environment on a UNIX Workstation applicable to modeling and control theory is presented.

TREEFLX is used to interface flexible body data from NASTRAN with the flexible multi-body non-linear analysis program TREETOPS. Powerful modeling and control design concepts are demonstrated via a non-trivial example. Results support the feasibility of using all of the programs in conjuction with one another to provide viable analysis and designs.

The ease in which the model or the controller can be changed further enhances the analysis turn-around-time and the design process itself, clearly demonstrating the advantages of working within a dedicated UNIX Workstation environment.

## ACKNOWLEDGEMENTS

The authors wish to acknowledge and thank Jayant Ramakrishnan for his advice, suggestions and remarks. Thanks go to Mike VanderVoort for all the help he gave, Anren Hu, for advice on lumped flexibility models and Julie Hendricks, for all the help she gave, without which this paper would never of taken so long to finish.

Thanks go to John Sharkey for his suggestions on the improvement of the TREEFLX program. This paper was supported in part with funds from NASA MSFC contract number NAS8-37287.

## REFERENCES


[1] TREETOPS Users Manual, Dynacs Engineering Company, Inc., 1988

[2] Skelton, Robert E., "Dynamic Systems Control: Linear Systems Analysis and Synthesis", John Wiley & Sons, 1988

[3] Anderson, Brian and Yi Liu, "Controller Reduction: Concepts and Approaches", Department of Systems Engineering, Australian National University

[4] Spar Aerospace Limited Document SPAR-R.775 Issue G, 1981, released by National Research Council of Canada for Unrestricted Use

**TABLE 1. - MASS AND GEOMETRIC PROPERTIES OF MRMS MODEL**

| BODY | MASS (kg) | Ixx | Iyy ( $kg\text{-}m^2$ ) | Izz | LENGTH (m) |
|---|---|---|---|---|---|
| 1 | 63.3 | 0 | 41.04 | 41.04 | 1.06 |
| 2 | 139.2 | 0 | 1877.9 | 1877.9 | 6.38 |
| 3 | 100.0 | 0 | 1429.9 | 1429.9 | 7.06 |
| 4 | 50.0 | 0 | 15.0 | 15.0 | 1.00 |

**TABLE 2. - (A,B,C,D) MATRICES AND CONTROLLER GAIN MATRIX G**

$$A = \begin{bmatrix} -.008816 & .016190 & -9.41087 & 11.2336 \\ .016190 & -.039067 & 17.28277 & -27.1063 \\ 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \end{bmatrix}$$

$$B = \begin{bmatrix} .000176 & -.000324 \\ -.000324 & .000781 \\ 0.0 & 0.0 \\ 0.0 & 0.0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \qquad D = \begin{bmatrix} 0.0 & 0.0 \\ 0.0 & 0.0 \\ 0.0 & 0.0 \\ 0.0 & 0.0 \end{bmatrix}$$

$$G = \begin{bmatrix} -5281.28 & 2243.41 & 1199.95 & -386.81 \\ 3478.13 & -8005.26 & -4361.25 & -1209.26 \end{bmatrix}$$

## TABLE 3. - TREETOPS MODAL TERMS CALCULATED BY TREEFLX

$$\underline{h}_i = b^T\{\sum_{o=1}^{NNB} [m_o \tilde{r}_o \{\phi_{oi}\} - \tilde{r}_o m_o \tilde{\ell}_o \{\phi'_{oi}\} + m_o \tilde{\ell}_o \{\phi'_{oi}\} + J^{bo} \{\phi'_{oi}\}]\}$$

$$\underline{a}_i = 1/_m \; \{\sum_{o=1}^{NNB} m_o \, (\underline{\phi}_{oi} - \underline{\ell}_o \times \underline{\phi}_{oi})\}$$

$$\underline{\underline{M}}_i^b = b^T\{\sum_{O=1}^{NNB} [-m_o \tilde{r}_o \, \tilde{\phi}_{oi} - m_o \, \tilde{\ell}_o \tilde{\phi}_{oi}]\} \, b$$

$$\underline{\underline{P}}_{ki}^b = b^T \{\sum_{o=1}^{NNB} [-m_o \, \tilde{\phi}_{ok} \, \tilde{\phi}_{oi}]\} \, b$$

$$\underline{\gamma}_{ki}^b = b^T\{\sum_{o=1}^{NNB} [m_o \, \tilde{\phi}_{ok} \, \{\phi_{oi}\} - \tilde{\phi}_{ok} \, m_o \, \tilde{\ell}_o \, \{\phi'_{oi}\}$$

WHERE:

$b$ represents the body reference frame

i,k represent the ith,kth modes

$\sum_{o=1}^{NNB}$ is the sum over the number of nodal bodies

$m_o$ is the mass of the $o^{th}$ nodal body

$m$ is the body mass ; $m = \sum_{o=1}^{NNB} m_o$

CONTINUED

TABLE 3. - CONTINUED

$J^{bo}$ is the inertia matrix of the $o^{th}$ nodal body

$\underline{\phi}_{ok}$ is the $k^{th}$ mode shape at the $o^{th}$ nodal body

$\phi'_{ok}$ is the $k^{th}$ mode slope at the $o^{th}$ nodal body

$\underline{\ell}_o$ is the vector location of the $o^{th}$ nodal body mass center

$\underline{r}_o$ is the vector location of the $o^{th}$ nodal body

$\{\cdot\}$ represents a 3x1 column matrix

$\tilde{\cdot}$ represents a skew symmetric matrix, that is, suppose $\underline{r}$ is given by;

$$\underline{r} = r_1 \hat{i} + r_2 \hat{j} + r_3 \hat{k}$$

then $\tilde{r}$ is;

$$\tilde{r} = \begin{bmatrix} o & -r_3 & r_2 \\ r_3 & o & -r_1 \\ -r_2 & r_1 & o \end{bmatrix}$$

**TABLE 4. - TREESEL RANKING OF SYSTEM MODES (BY BODY)**

| RANK | BODY #2 MODES | BODY #3 MODES |
|---|---|---|
| 1 | 14 | 16 |
| 2 | 9 | 4 |
| 3 | 11 | 2 |
| 4 | 4 | 9 |
| 5 | 6 | 11 |
| ------ | ------ | ------ |
| 6 | 2 | 6 |
| 7 | 13 | 18 |
| 8 | 19 | 14 |
| 9 | 8 | 5 |
| 10 | 5 | 3 |
| 11 | 10 | 8 |
| 12 | 3 | 1 |
| 13 | 18 | 10 |
| 14 | 7 | 17 |
| 15 | 12 | 7 |
| 16 | 16 | 13 |
| 17 | 20 | 20 |
| 18 | 15 | 15 |
| 19 | 1 | 19 |
| 20 | 17 | 12 |

This table presents data obtained with only one TREESEL iteration.
Simply selected weights were used.
The five highest ranked modes were retained
for the TREETOPS flexible model.

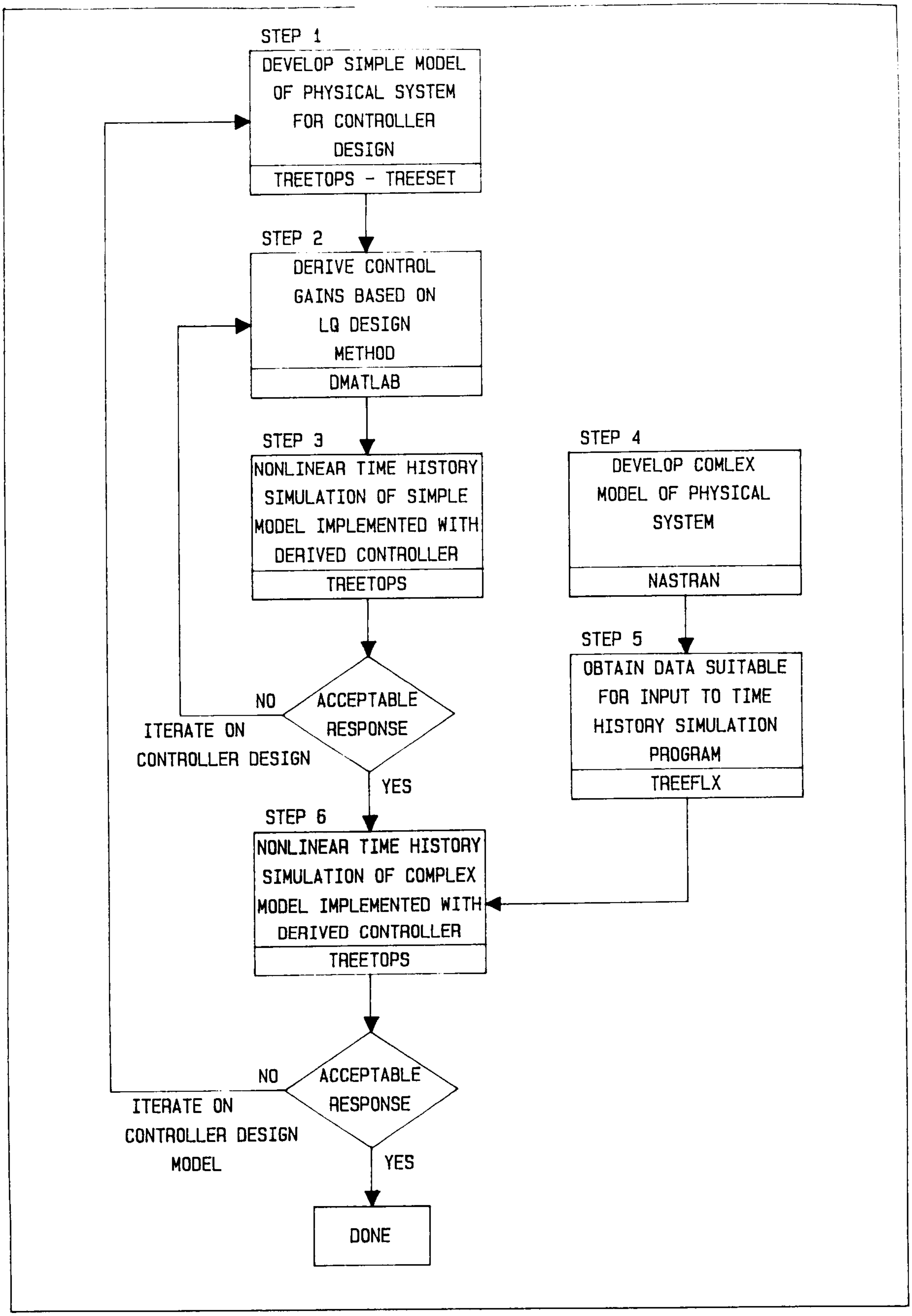

STEP 1
DEVELOP SIMPLE MODEL OF PHYSICAL SYSTEM FOR CONTROLLER DESIGN
TREETOPS - TREESET
STEP 2
DERIVE CONTROL GAINS BASED ON LQ DESIGN METHOD
DMATLAB
STEP 3
NONLINEAR TIME HISTORY SIMULATION OF SIMPLE MODEL IMPLEMENTED WITH DERIVED CONTROLLER
TREETOPS
STEP 4
DEVELOP COMLEX MODEL OF PHYSICAL SYSTEM
NASTRAN
STEP 5
OBTAIN DATA SUITABLE FOR INPUT TO TIME HISTORY SIMULATION PROGRAM
TREEFLX
ACCEPTABLE RESPONSE
NO
ITERATE ON CONTROLLER DESIGN
YES
STEP 6
NONLINEAR TIME HISTORY SIMULATION OF COMPLEX MODEL IMPLEMENTED WITH DERIVED CONTROLLER
TREETOPS
ACCEPTABLE RESPONSE
NO
ITERATE ON CONTROLLER DESIGN MODEL
YES
DONE

Figure 1.

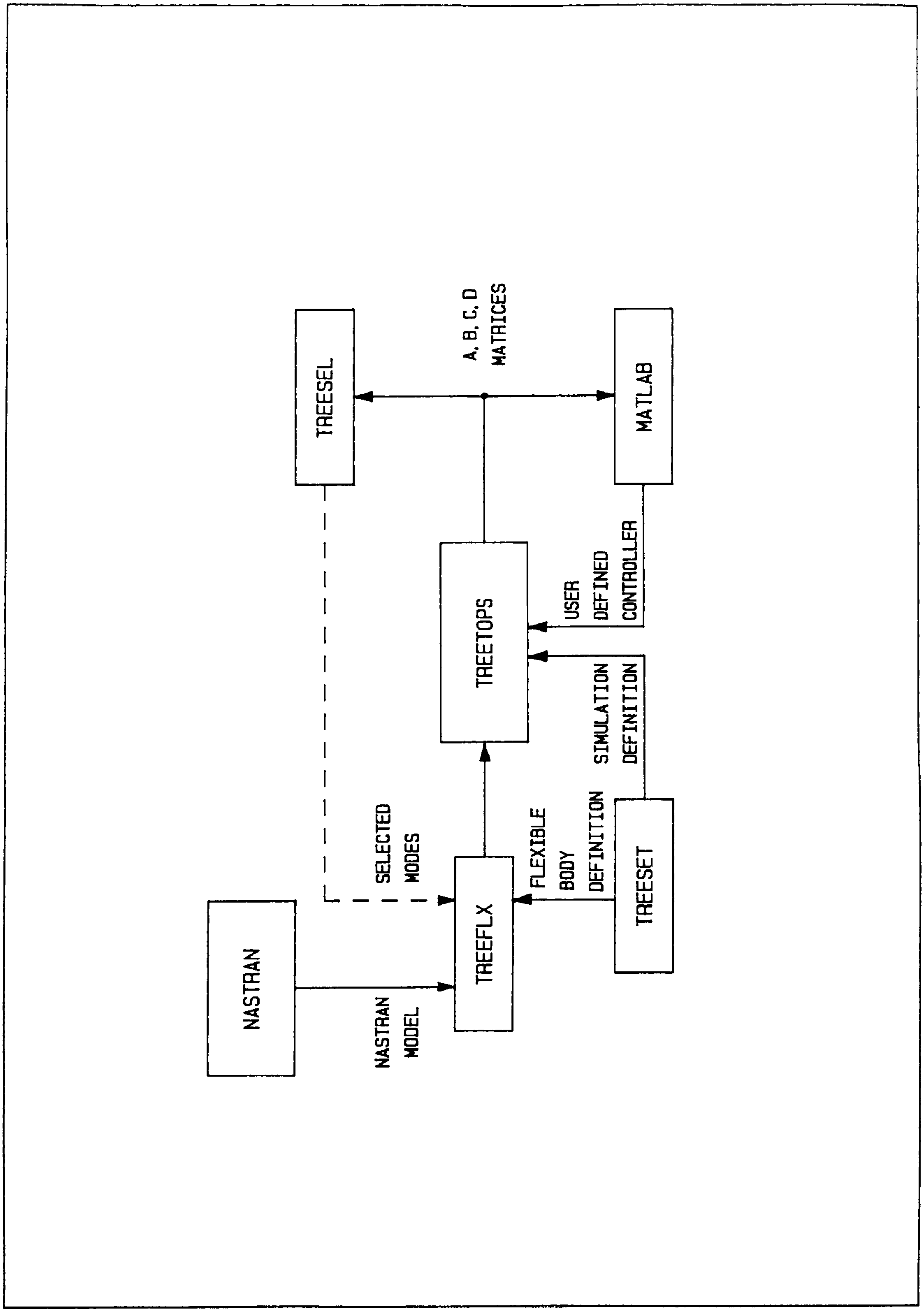

NASTRAN
NASTRAN MODEL
TREEFLX
SELECTED MODES
FLEXIBLE BODY DEFINITION
TREESET
SIMULATION DEFINITION
TREETOPS
USER DEFINED CONTROLLER
TREESEL
A, B, C, D MATRICES
MATLAB

Figure 2.

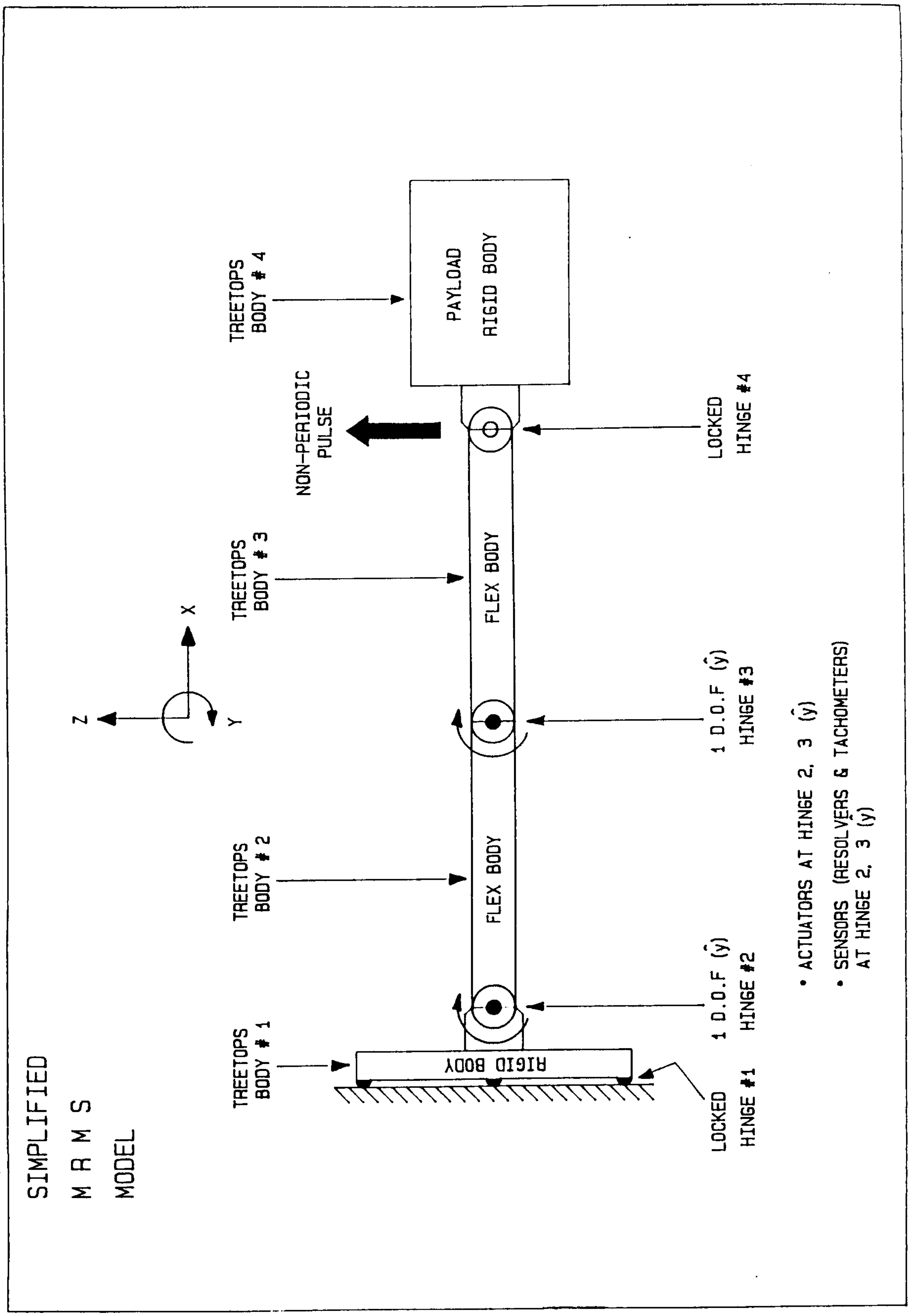

SIMPLIFIED
M R M S
MODEL
TREETOPS BODY # 1
TREETOPS BODY # 2
TREETOPS BODY # 3
TREETOPS BODY # 4
RIGID BODY
FLEX BODY
FLEX BODY
PAYLOAD
RIGID BODY
NON-PERIODIC PULSE
Z
X
Y
LOCKED HINGE #1
1 D.O.F (ŷ) HINGE #2
1 D.O.F (ŷ) HINGE #3
LOCKED HINGE #4
• ACTUATORS AT HINGE 2, 3 (ŷ)
• SENSORS (RESOLVERS & TACHOMETERS) AT HINGE 2, 3 (ŷ)

Figure 3.

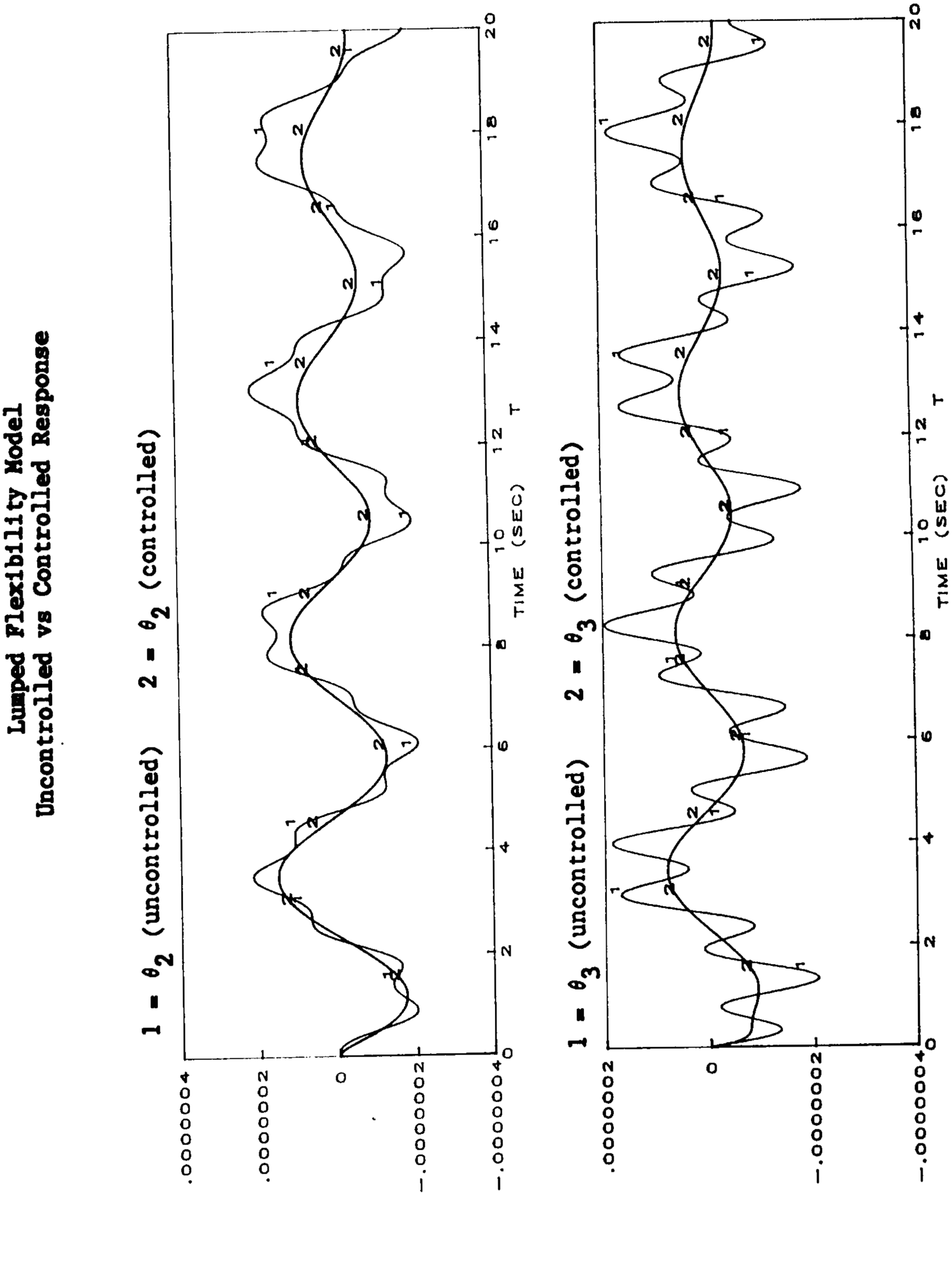
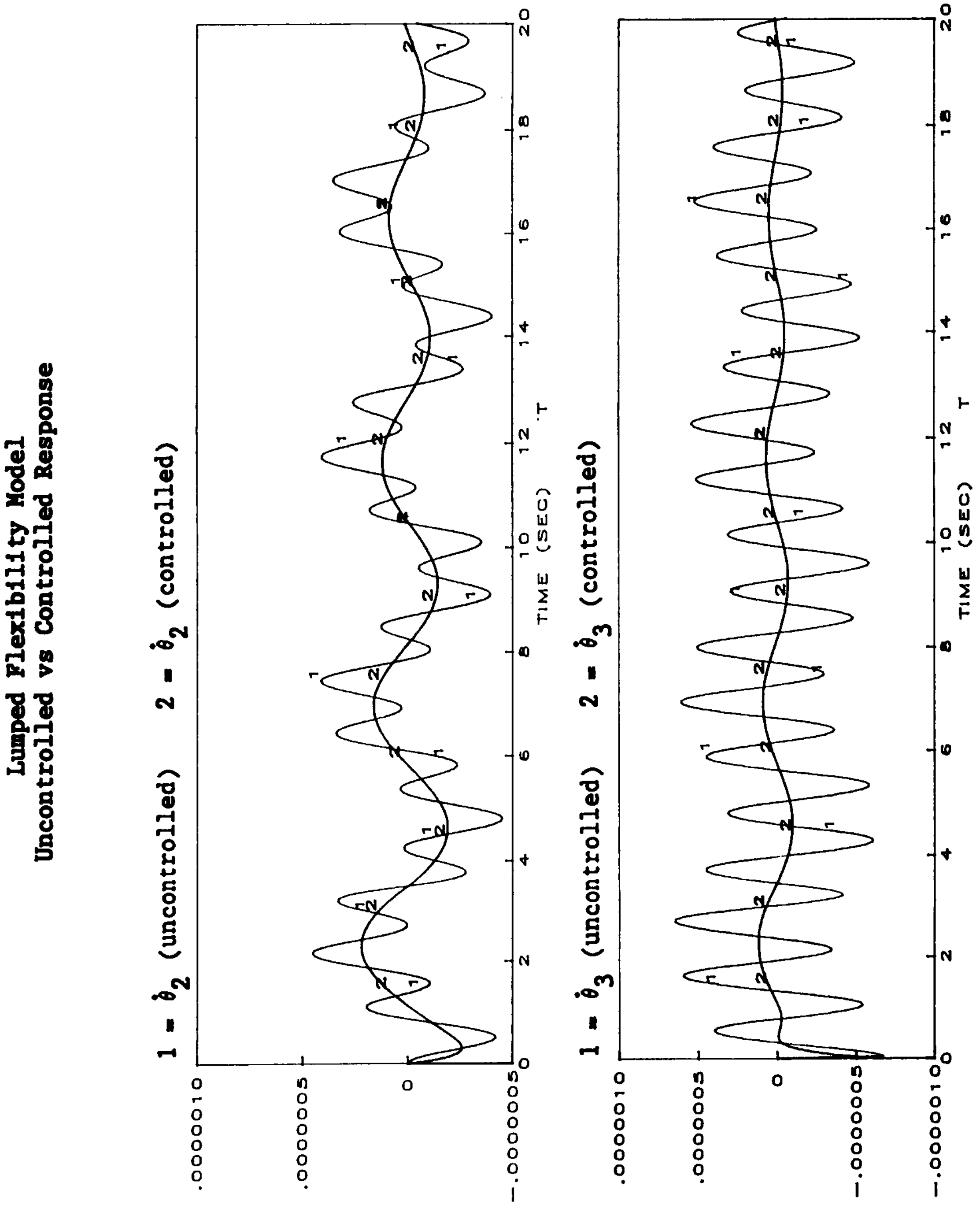

Lumped Flexibility Model
Uncontrolled vs Controlled Response
1 = θ2 (uncontrolled)    2 = θ2 (controlled)
1 = θ3 (uncontrolled)    2 = θ3 (controlled)
TIME (SEC)
T
.0000004
.0000002
0
-.0000002
-.0000004
0
2
4
6
8
10
12
14
16
18
20

Figure 4.



Lumped Flexibility Model
Uncontrolled vs Controlled Response
1 = $\dot{\theta}_2$ (uncontrolled)  2 = $\dot{\theta}_2$ (controlled)
.0000010
.0000005
0
-.0000005
0
2
4
6
8
10
12
14
16
18
20
TIME (SEC)
T
1 = $\dot{\theta}_3$ (uncontrolled)  2 = $\dot{\theta}_3$ (controlled)
.0000010
.0000005
0
-.0000005
-.0000010
TIME (SEC)
T


Figure 5.

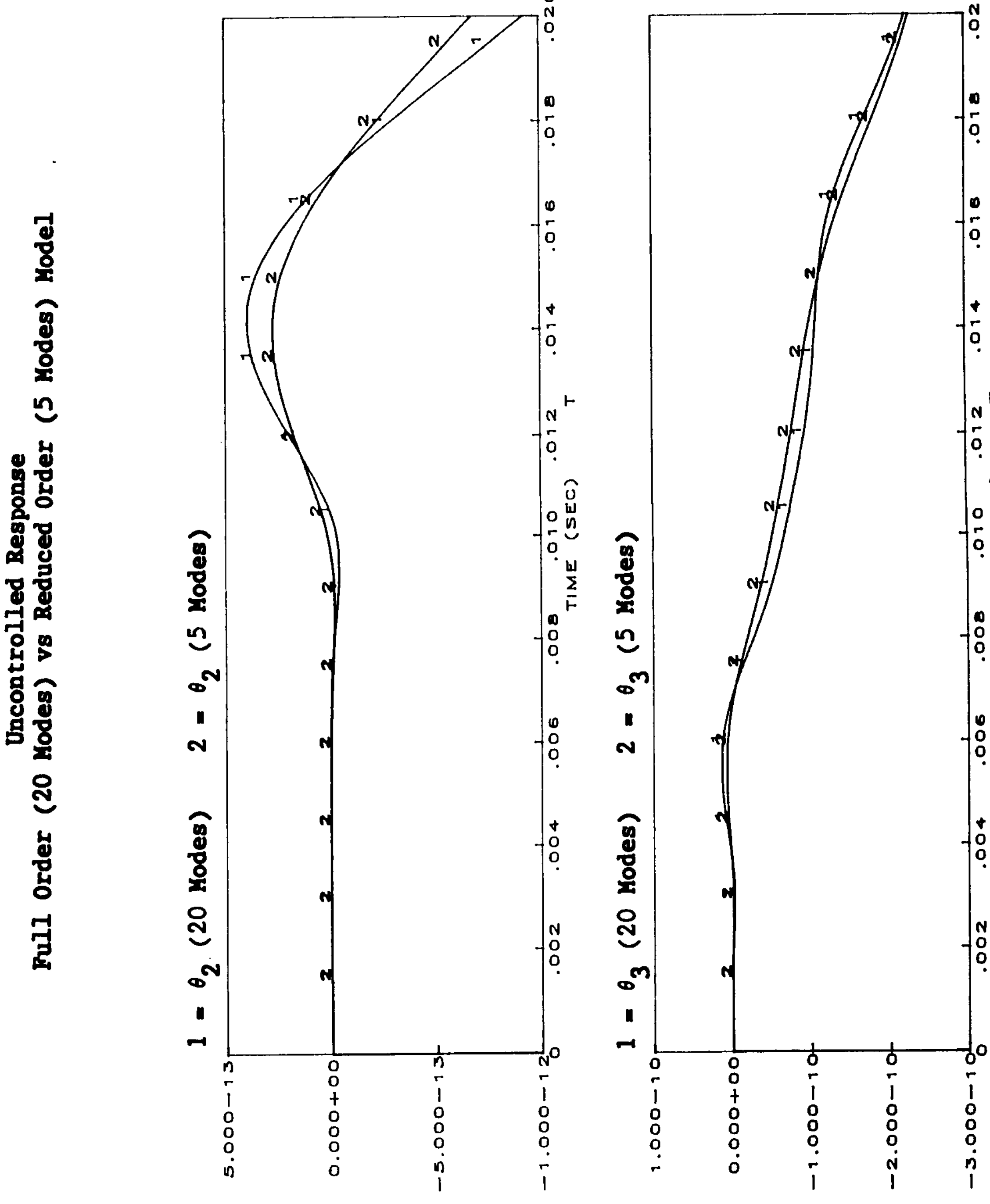
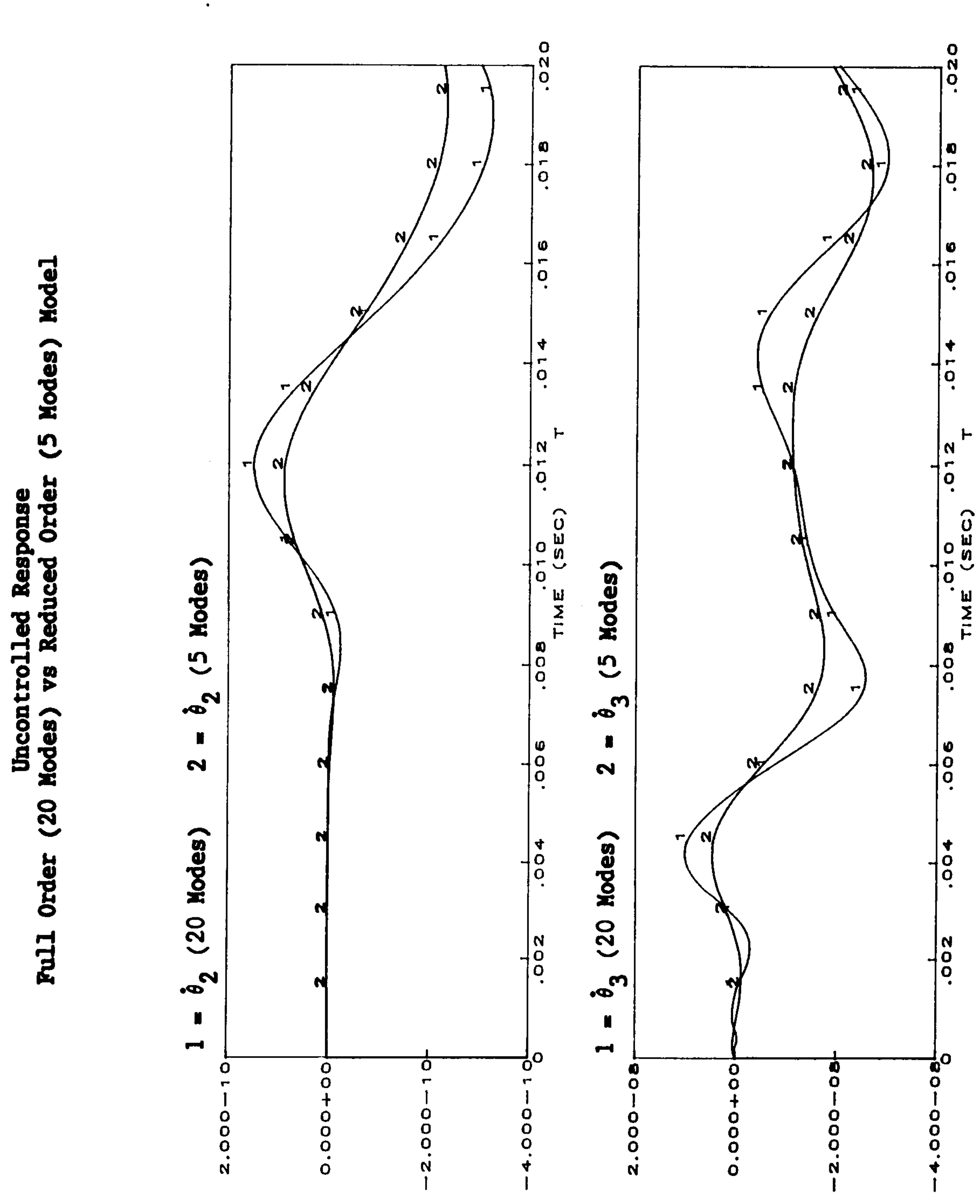

Uncontrolled Response
Full Order (20 Modes) vs Reduced Order (5 Modes) Model
1 = θ2 (20 Modes)   2 = θ2 (5 Modes)
5.000−13
0.000+00
−5.000−13
−1.000−12
0
.002
.004
.006
.008
.010
.012
.014
.016
.018
.020
TIME (SEC)
T
1 = θ3 (20 Modes)   2 = θ3 (5 Modes)
1.000−10
0.000+00
−1.000−10
−2.000−10
−3.000−10

Figure 6.



Uncontrolled Response
Full Order (20 Modes) vs Reduced Order (5 Modes) Model
1 = $\dot{\theta}_2$ (20 Modes) 2 = $\dot{\theta}_2$ (5 Modes)
1 = $\dot{\theta}_3$ (20 Modes) 2 = $\dot{\theta}_3$ (5 Modes)
TIME (SEC)
T


Figure 7.

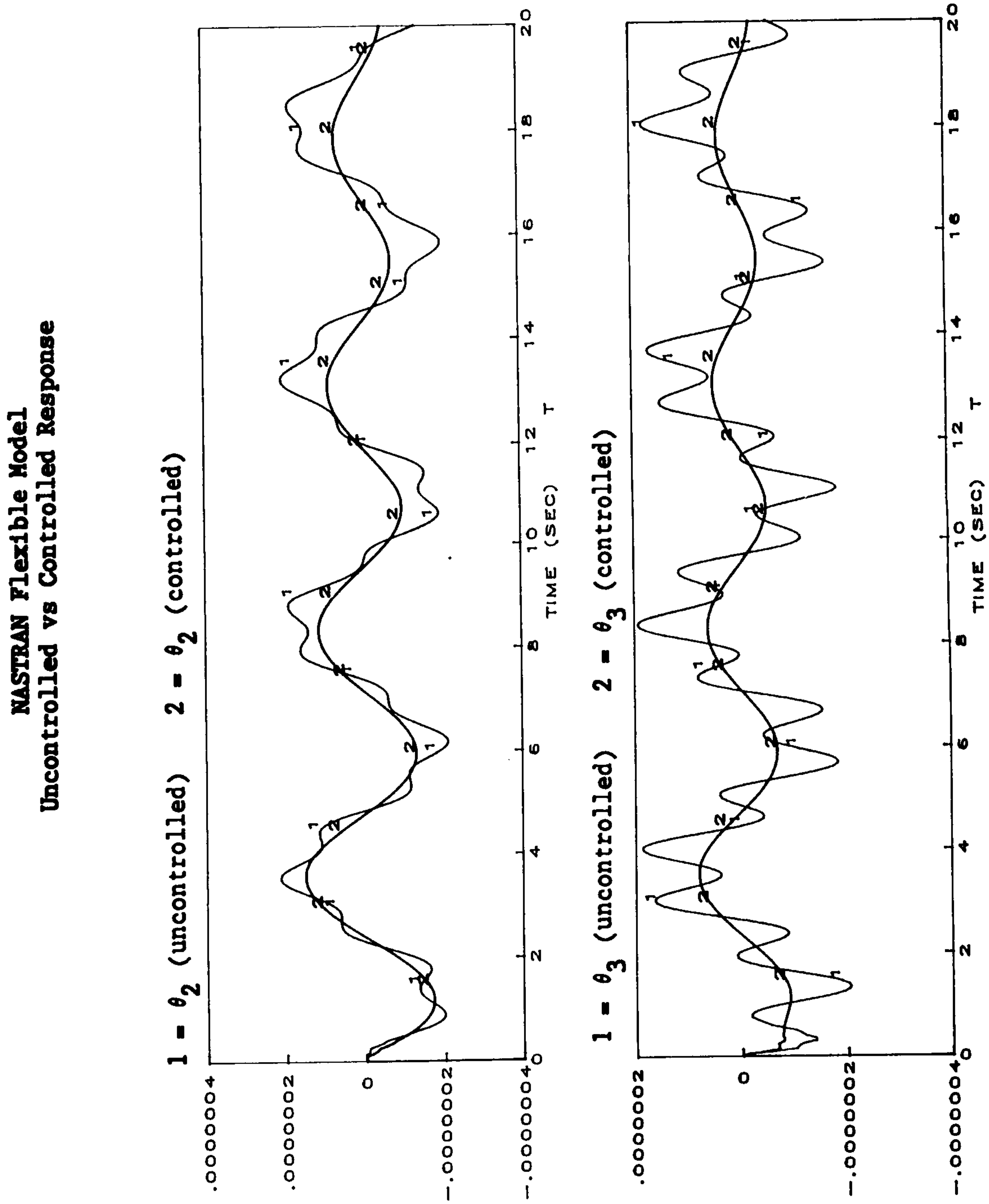

NASTRAN Flexible Model
Uncontrolled vs Controlled Response
1 = $\theta_2$ (uncontrolled)    2 = $\theta_2$ (controlled)
.0000004
.0000002
0
-.0000002
-.0000004
0
2
4
6
8
10
12
14
16
18
20
TIME (SEC)
T
1 = $\theta_3$ (uncontrolled)    2 = $\theta_3$ (controlled)
.0000002
0
-.0000002
-.0000004
TIME (SEC)
T

Figure 8.

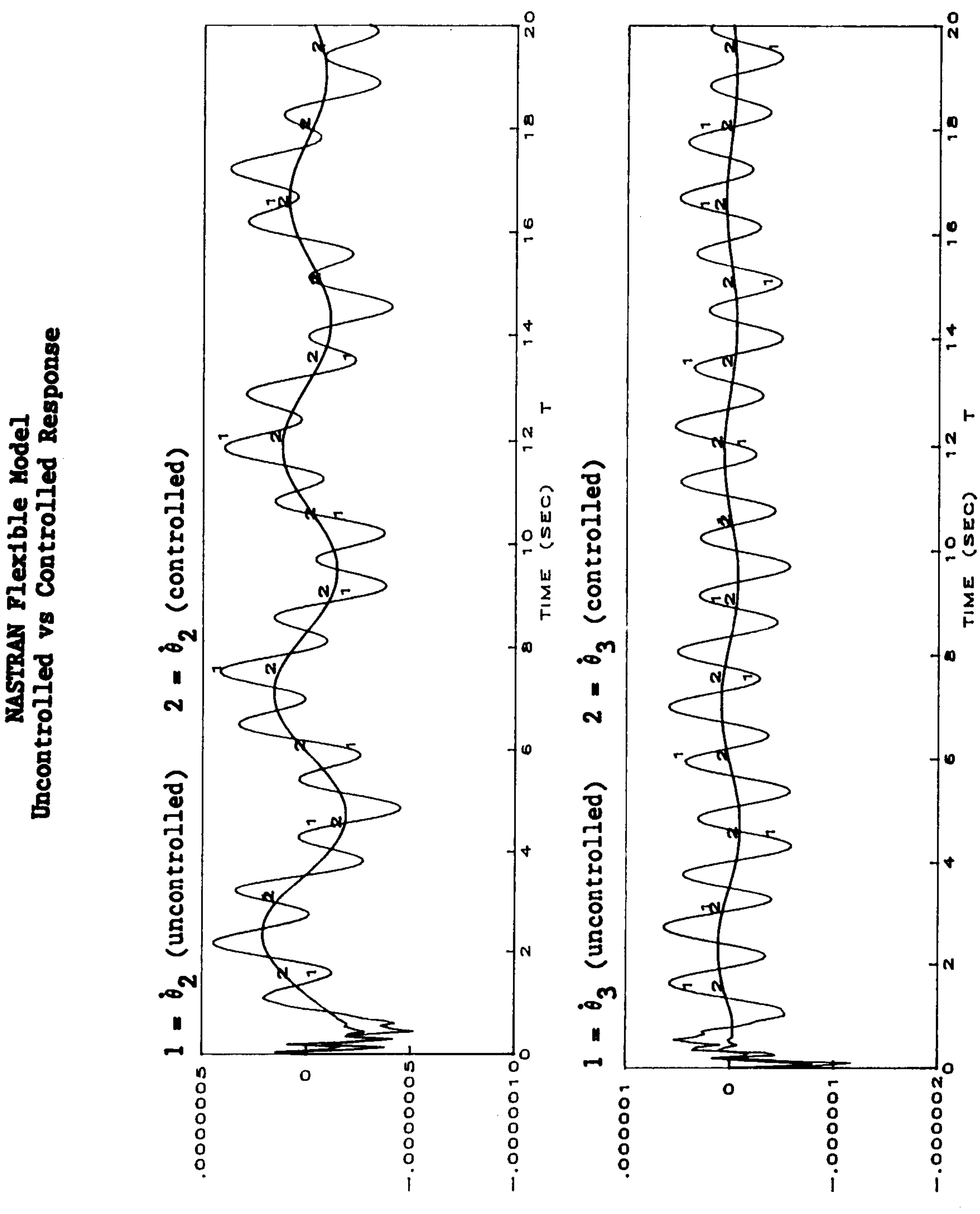

NASTRAN Flexible Model
Uncontrolled vs Controlled Response
1 = $\dot{\theta}_2$ (uncontrolled)  2 = $\dot{\theta}_2$ (controlled)
.0000005
0
-.0000005
-.0000010
TIME (SEC)  T
1 = $\dot{\theta}_3$ (uncontrolled)  2 = $\dot{\theta}_3$ (controlled)
.000001
0
-.000001
-.000002
TIME (SEC)  T

Figure 9.

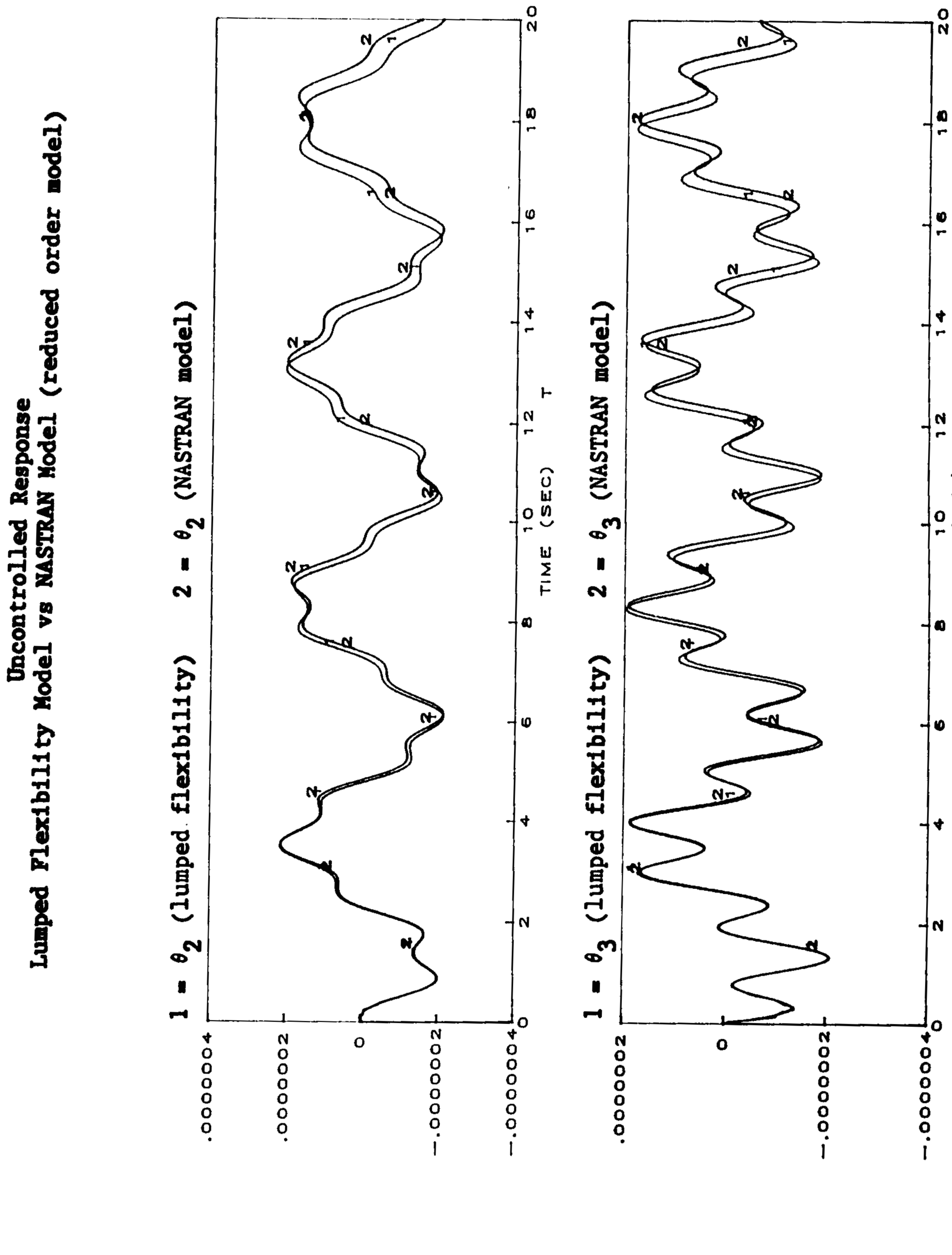

Uncontrolled Response
Lumped Flexibility Model vs NASTRAN Model (reduced order model)
1 = $\theta_2$ (lumped flexibility)    2 = $\theta_2$ (NASTRAN model)
1 = $\theta_3$ (lumped flexibility)    2 = $\theta_3$ (NASTRAN model)
TIME (SEC)
T

Figure 10.

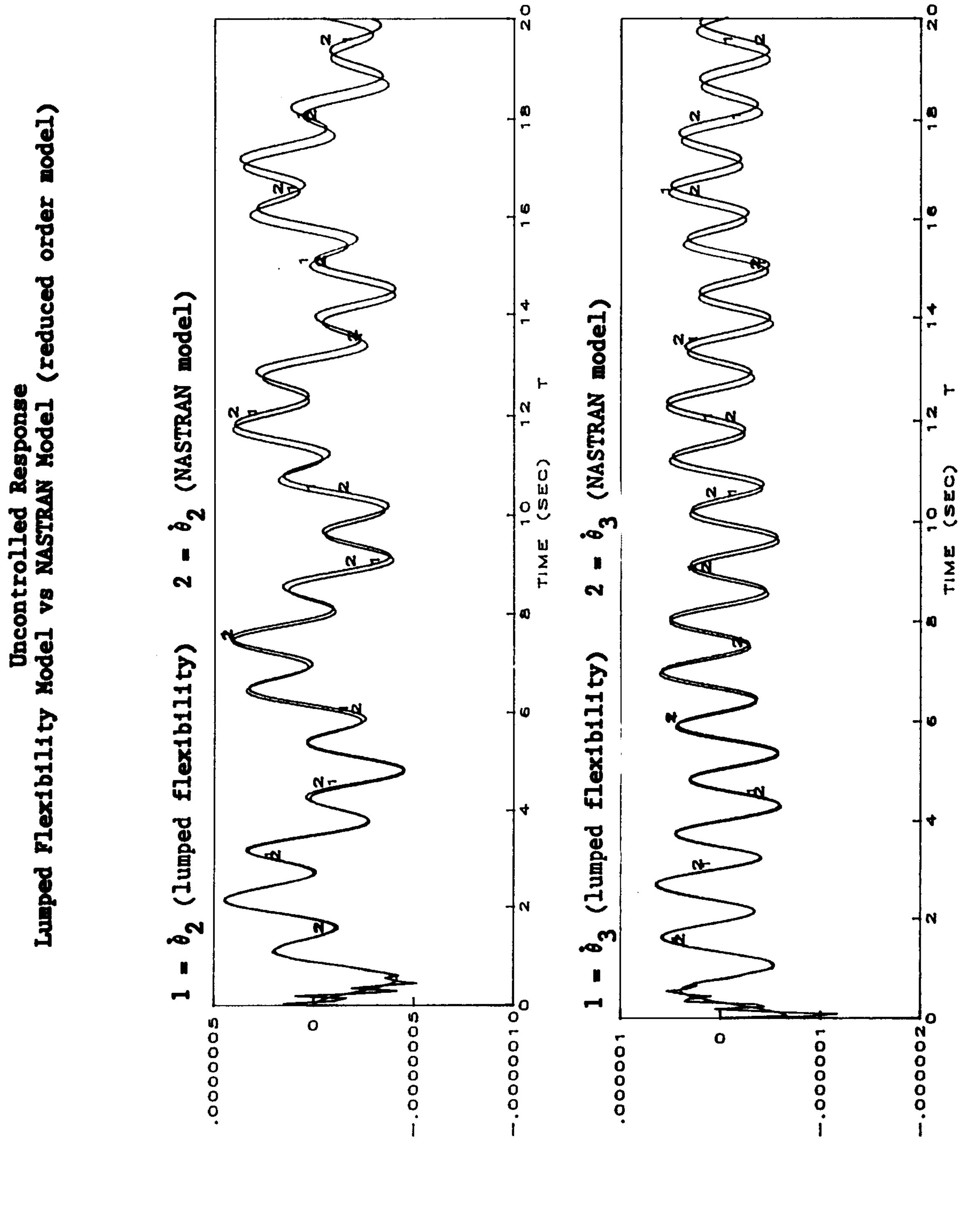

Uncontrolled Response
Lumped Flexibility Model vs NASTRAN Model (reduced order model)
1 = $\dot{\theta}_2$ (lumped flexibility)   2 = $\dot{\theta}_2$ (NASTRAN model)
.0000005
0
-.0000005
-.0000010
1 = $\dot{\theta}_3$ (lumped flexibility)   2 = $\dot{\theta}_3$ (NASTRAN model)
.000001
0
-.000001
-.000002
0
2
4
6
8
10
12
14
16
18
20
TIME (SEC)   T

Figure 11.

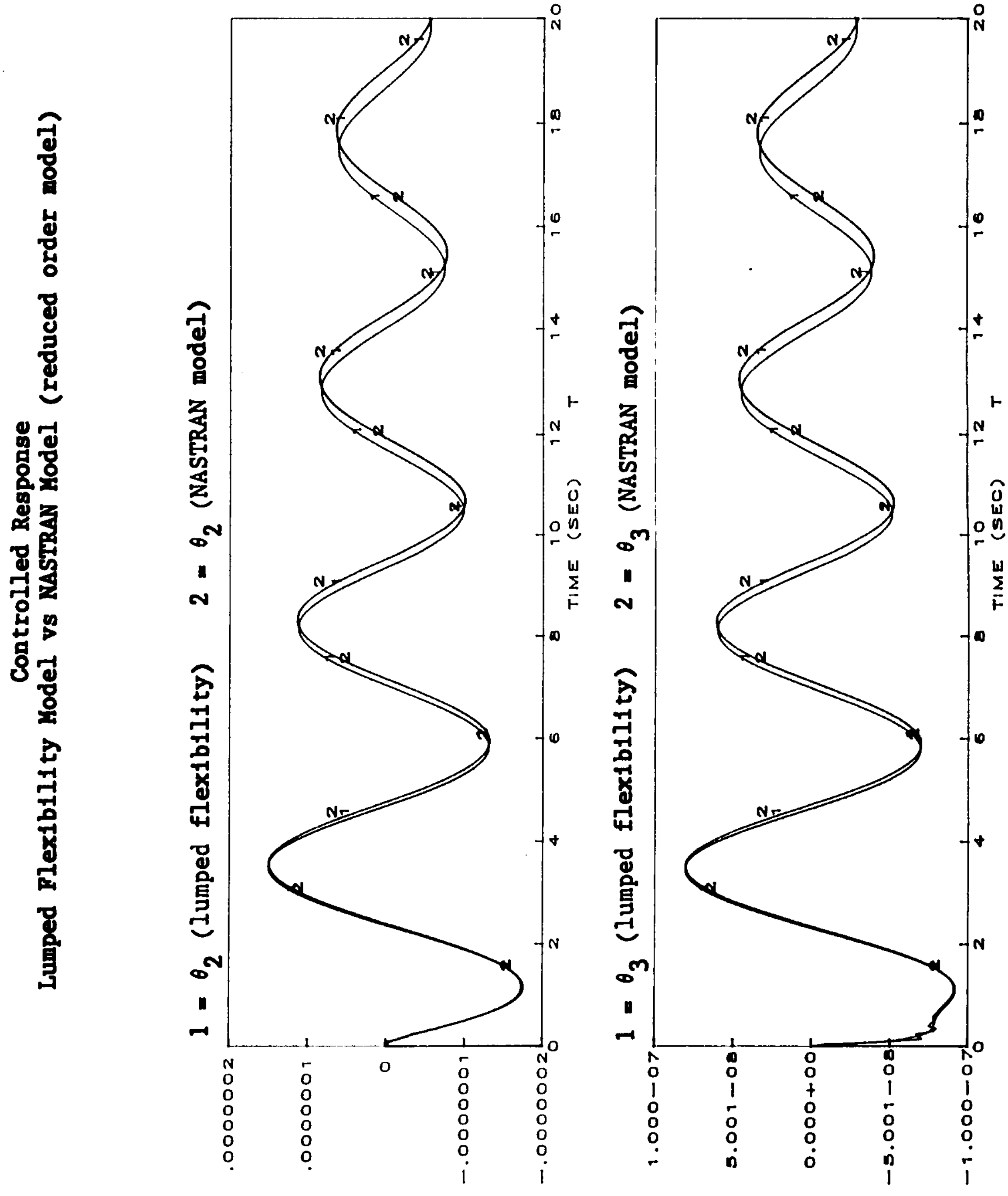
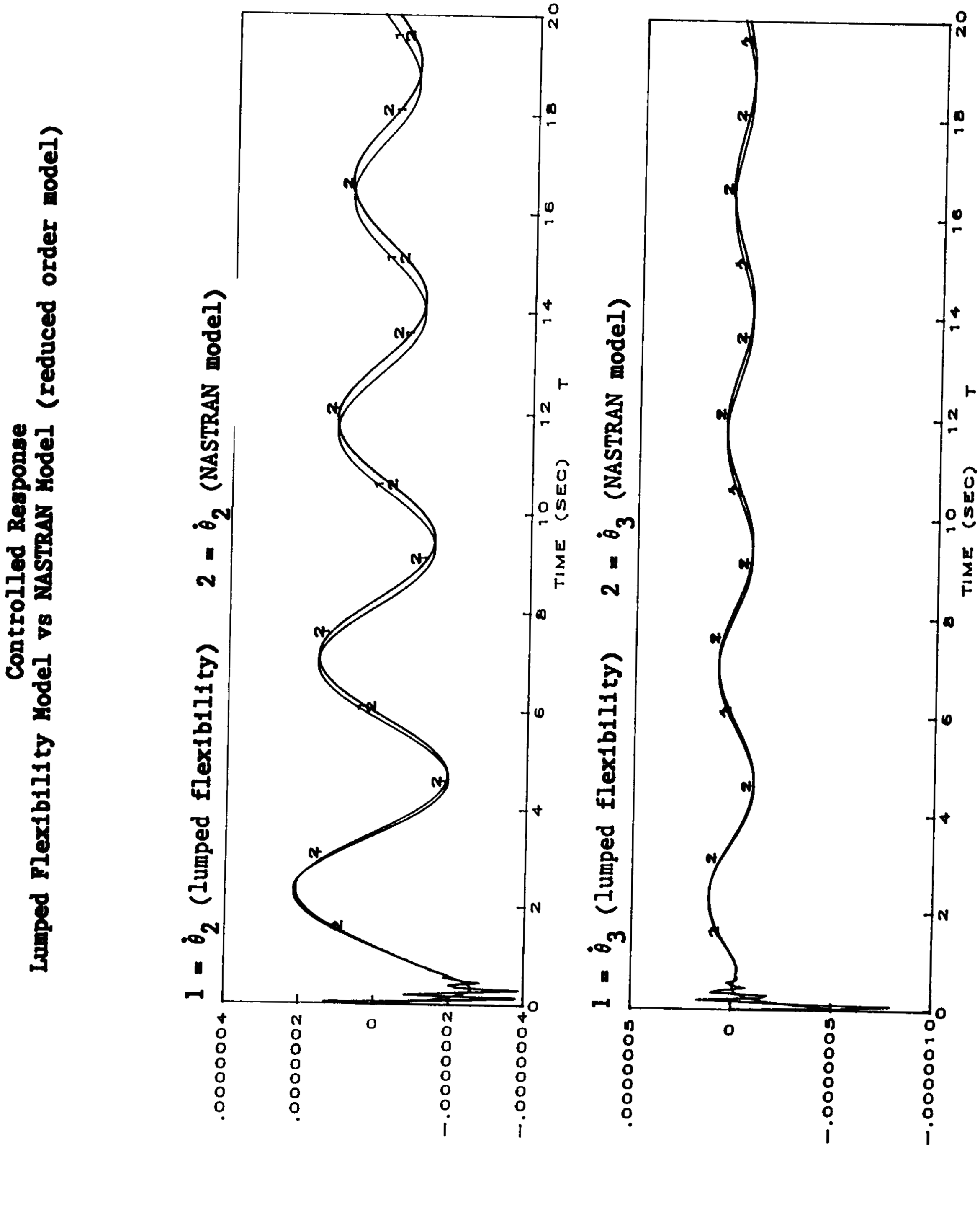

Controlled Response
Lumped Flexibility Model vs NASTRAN Model (reduced order model)
1 = θ2 (lumped flexibility)   2 = θ2 (NASTRAN model)
.0000002
.0000001
0
-.0000001
-.0000002
1 = θ3 (lumped flexibility)   2 = θ3 (NASTRAN model)
1.000-07
5.001-08
0.000+00
-5.001-08
-1.000-07
0 2 4 6 8 10 12 14 16 18 20
TIME (SEC) T

Figure 12.



Controlled Response
Lumped Flexibility Model vs NASTRAN Model (reduced order model)
1 = $\dot{\theta}_2$ (lumped flexibility) 2 = $\dot{\theta}_2$ (NASTRAN model)
.0000004
.0000002
0
-.0000002
-.0000004
0
2
4
6
8
10
12
14
16
18
20
TIME (SEC) T
1 = $\dot{\theta}_3$ (lumped flexibility) 2 = $\dot{\theta}_3$ (NASTRAN model)
.0000005
0
-.0000005
-.0000010
0
2
4
6
8
10
12
14
16
18
20
TIME (SEC) T


Figure 13.

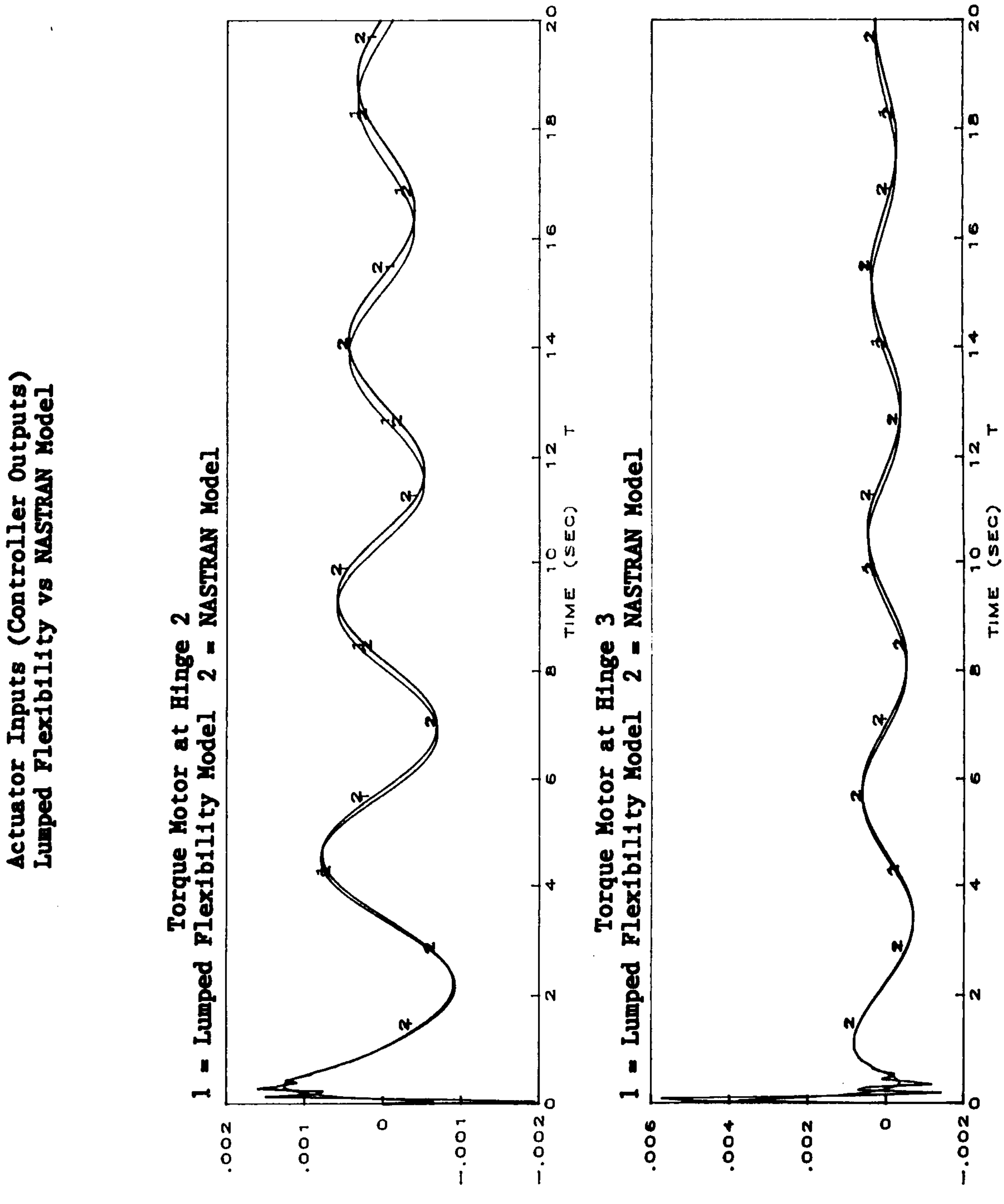

Actuator Inputs (Controller Outputs)
Lumped Flexibility vs NASTRAN Model
Torque Motor at Hinge 2
1 = Lumped Flexibility Model 2 = NASTRAN Model
.002
.001
0
-.001
-.002
0
2
4
6
8
10
12
14
16
18
20
TIME (SEC)
T
Torque Motor at Hinge 3
1 = Lumped Flexibility Model 2 = NASTRAN Model
.006
.004
.002
0
-.002
0
2
4
6
8
10
12
14
16
18
20
TIME (SEC)
T

Figure 14.