A TRIDENT SCHOLAR PROJECT REPORT

NO. 207

Design of a State-Space Controller for an Advanced Gas Turbine Engine

UNITED STATES NAVAL ACADEMY
ANNAPOLIS, MARYLAND

U.S.N.A. - Trident Scholar project report; no. 207 (1993)

Design of a State-Space Controller for an Advanced Gas Turbine Engine

by
Midshipman Karl F. Prigge, Class of 1993
U.S. Naval Academy
Annapolis, Maryland

Jerry W. Watts
Adviser: Associate Professor Jerry W. Watts
Weapons and Systems Engineering Department

Terry E. Dwan
Adviser: Professor Terry E. Dwan
Weapons and Systems Engineering Department

Accepted for Trident Scholar Committee

Francis D Correll
Chair

May 17, 1993
Date

USNA-1531-2

REPORT DOCUMENTATION PAGE

Form Approved
OMB no. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE May 17, 1993	3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE Design of a state-space controller for an advanced gas turbine engine	5. FUNDING NUMBERS
6. AUTHOR(S) Karl Frederick Prigge	
7. PERFORMING ORGANIZATIONS NAME(S) AND ADDRESS(ES) U.S. Naval Academy, Annapolis, MD	8. PERFORMING ORGANIZATION REPORT NUMBER U.S.N.A. - Trident scholar project report ; no. 207
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

Accepted by the U.S. Trident Scholar Committee

12a. DISTRIBUTION/AVAILABILITY STATEMENT This document has been approved for public release; its distribution is UNLIMITED.	12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

A multi-input, multi-output (MIMO) controller for an advanced gas turbine has been developed and tested using a computer simulation. The engine modeled is a two-and-one-half spool gas turbine with both an intercooler and a regenerator. In addition, variable stator vanes are present in the free-power turbine. This advanced engine is proposed for future naval propulsion for both mechanical drive and electrical drive. The designed controller controls free-power turbine speed and turbine inlet temperature using fuel flow and angle of the stator vanes. The controller will also have four modes of operation to deal with over temperature and over speed conditions. An eight state reduced order controller was used with pole placement and LQR to arrive at control gains. Both these methods required considerable insight into the problem. This insight was provided by previous experience with controller design for a less complicated engine, and also by use of a polyhedral search model of the gas turbine engine. The difficulty with a MIMO controller was that both inputs affect both of the control variables. The classical resolution of this problem is to have one input control one variable at a fast time constant and the other input control the other variable at a slow time constant. The "optimal" resolution of this problem is analyzed using the transient curves and basic control theory.

14. SUBJECT TERMS gas turbine engine, intercooled, regenerated, controller design, MIMO controller design	15. NUMBER OF PAGES 48
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED

NSN 7540-01-280-5500

Standard Form 298 (Rev.2-89)

Abstract

A multi-input, multi-output (MIMO) controller for an advanced gas turbine has been developed and tested using a computer simulation. The engine modeled is a two-and-one-half spool gas turbine with both an intercooler and a regenerator. In addition, variable stator vanes are present in the free-power turbine. This advanced engine is proposed for future naval propulsion for both mechanical drive and electrical drive. The designed controller controls free-power turbine speed and turbine inlet temperature using fuel flow and angle of the stator vanes. The controller will also have four modes of operation to deal with over temperature and over speed conditions. An eight state reduced order controller was used with pole placement and LQR to arrive at control gains. Both these methods required considerable insight into the problem. This insight was provided by previous experience with controller design for a less complicated engine, and also by use of a polyhedral search model of the gas turbine engine. The difficulty with a MIMO controller was that both inputs affect both of the control variables. The classical resolution of this problem is to have one input control one variable at a fast time constant and the other input control the other variable at a slow time constant. The "optimal" resolution of this problem is analyzed using the transient curves and basic control theory.

Keywords: gas turbine engine, intercooled, regenerated, controller design, MIMO controller design.

Table of Contents

List of Tables

List of Figures

List of Appendices

1.0 INTRODUCTION

The development of a computer model for the advanced gas turbine engine under investigation has been through many stages and steps. The development of a controller depended totally on a working computer model capable of running load transients. At the time this Trident Scholar project started, a working dynamic model was available but the controller consisted only of fuel controlling free power turbine speed. The variable turbine nozzles were not yet modeled. The control gains for controlling free power turbine speed had been explored thoroughly[1,2,3]. This was a big help in finding the correct gains for the multi-input, multi-output (MIMO) system.

2.0 GOALS OF THIS PROJECT

The goals for this project were: 1) Learn the theory involved with state-space controls and matrix manipulation; 2) Become familiar with the ACSL language and commands in order to be able to program the controller and add any other changes necessary to the existing model; 3) Learn the advanced commands in Matlab, to be used in development and optimization of the four mode controller; 4) Acquire an understanding of the basic design and operation of a gas turbine engine, as this is necessary for the correct interpretation of results acquired in the process of controller development; and 5) Develop, implement and optimize a four mode controller to provide tightly controlled speed and temperature, and highly efficient operation for the gas turbine over severe load transients. All these goals have been accomplished. In addition to the stated goals, the stator vane map, which defines for the model how the vanes will affect the turbine engine operation, was refined and extended to include a more satisfactory and reasonable range of operation for the turbine. This was necessary to produce the expected results from the MIMO controller.

3.0 MODEL DESCRIPTION

The computer model is programmed in the Advanced Continuous Simulation Language (ACSL)[4], which is based on FORTRAN, but gives commands specific to system modeling, adding convenience and ease to operations such as integration and first order lags. The program also communicates directly with the powerful engineering software package, Matlab[5]. The matrix manipulation abilities of Matlab are useful in the analysis of the state matrices of the model and in the design of system controls. ACSL produces the A,B,C and D matrices from the model and sends them directly to Matlab for analysis and manipulation. Control methods such as state-space pole placement and linear quadratic regulator (LQR) could be performed, producing gains for the optimal control of the ICR gas turbine.

The development of a complete and accurate model of an intercooled, regenerative (ICR) gas turbine engine is a long and complicated task. Figure 3.1 is the flow chart of the turbine model as programmed in ACSL. The creation of this model began many years ago and this project is only concerned with a small addition to it. This model was developed by Richard Garman of David Taylor Research Center and Associate Professor Jerry Watts of the United States Naval Academy. In May of 1992 the model consisted of a gas turbine engine with intercooler and regenerator heat exchangers. The code for variable angle stator vanes had been written, but the map, which defined the effects of the vanes on the system, was not yet fully completed. The map needed to be expanded to properly replicate the actual operation of the vanes in a true turbine. Also, the control at that time was Single-Input, Single-Output (SISO), varying only fuel flow to control speed error.

Work was also done by Midshipman First Class Mark Olson in the area of developing a state-space control using a reduced order system[3]. Using the SISO controller described above, Midshipman Olsen developed a three (3) state and a five (5) state controller. Figure 3.2 shows the results of a three state SISO controller. The most critical problem with this controller is its inability to rapidly control speed because of the resulting temperature increase. Many questions such as the importance of particular states in a controller, and how to determine the states to use in a controller, were investigated by Midshipman Olsen in his work.

The turbine used in this model is a two-and-one-half spool, intercooled, regenerative (ICR) gas turbine with variable angle stator vanes. Figure 3.3 is a diagram of the turbine modeled for this project. The ICR engine has both low and high pressure compressors linked to low and high pressure turbines, respectively. These are on separate concentric shafts, allowing them to rotate at different speeds. This independence gives smoother, more efficient compression and produces less stress on the spools, as they will be able to rotate at their optimal speeds. The turbine which provides the work to the outside system, i.e. drives the electric generator or shaft, is known as the free power turbine. It is also on a separate spool, again allowing rotation independent from the compressor/turbine spools, thereby reducing compressor stalls at heavy loads.

A means to increase efficiency is to add heat exchangers to the system in order to recuperate some of the energy normally lost to the surroundings. The regenerator accomplishes this by running the air exiting the high pressure compressors past the exhaust gasses prior to entering the combustion chamber. This has two positive effects. First, the regenerator cools the exhaust gas, thereby reducing heat signatures of the engine. This is much more critical on military vessels then merchant ships. Second, the heat exchanger pre-heats the air entering the combustion chamber, which reduces the amount of fuel needed to bring the same volume of gas to the high temperature desired for efficient operation.

The second heat exchanger is an intercooler, which removes heat from the partially compressed air just prior to entering the high pressure compressor. This allows the same amount of compression of the air with less energy spent. Again, the same efficient temperature and pressure levels can be reached at a lower energy cost.

The final and most innovative addition to the ICR engine is the variable angle stator vanes. These vanes are added at the inlet of the free power turbine and provide a means of rapid control of the temperatures in the engine. The angle of the vanes perpendicular to the air flow can be increased to slow the flow of air through the turbine, thereby increasing the temperature in the combustion chamber. When the angle is decreased, the volume of airflow is rapidly increased, quickly bringing down the temperature in the combustion chamber. This unique, innovative control over temperature allows more rapid acceleration of the engine when load is added as well as continued high efficiency at part power, a case most commonly experienced by naval vessels on long transits or during station keeping exercises. The increases in efficiency at these periods would be substantial.

4.0 CONTROLLER DESCRIPTION

The controller designed for this project is a multi-input, multi-output (MIMO) controller, with fuel flow (WFUEL) and variable stator vane angle (VSAPT) as the inputs, and low pressure turbine inlet temperature (T4) error and free power turbine speed (XNPT) error as the outputs. A MIMO controller is different from, as well as more complicated than, a single-input, single-output (SISO) controller because the effect of one input cannot easily be directly connected to the response of one output. WFUEL will affect both T4 error and XNPT error, often in opposition to the control being applied by the VSAPT. The classical method of solving this problem is to have one input react very rapidly, while the second input acts more slowly. This allows the fast variable to exercise its control without the slower variable countering its action. The result of two variables being too similar in reaction speed is an unstable system characterized by 'ringing' as shown in Figure 4.1. This figure graphically shows the interconnection between the inputs and the outputs. The instability in one variable is seen in all the others. The system response is to a standard load profile, Figure 4.2, which is used in all plots to produce the transients shown.

In controls, a system is described by its states. The ICR engine has 36 states which uniquely describe it. These 36 states are:

1. WGLC low speed spool shaft speed, rad/sec
2. WGHC high speed spool shaft speed, rad/sec

3. WGPT free power turbine shaft speed, rad/sec
4. WGPTI integral of FPT shaft speed, radians
5. WGPTD derivative of FPT shaft speed, rad/sec**2
6. T22 exit temperature of control volume 1, deg R
7. W22 mass in control volume 1, lbs
8. T24 exit temperature of control volume 2, deg R
9. W24 mass in control volume 2, lbs
10. T3 exit temperature of control volume 3, deg R
11. W3 mass in control volume 3, lbs
12. T35 exit temperature of control volume 4, deg R
13. W35 mass in control volume 3, lbs
14. T4 exit temperature of control volume 5, deg R
15. W4 mass in control volume 5, lbs
16. T42 exit temperature of control volume 6, deg R
17. W42 mass in control volume 6, lbs
18. T44 exit temperature of control volume 7, deg R
19. W44 mass in control volume 7, lbs
20. T5 exit temperature of control volume 8, deg R
21. W5 mass in control volume 8, lbs
22. THC temperature contribution (hot in to cold out) of the regenerator, deg R
23. TCC temperature contribution (cold in to cold out) of the regenerator, deg R
24. THH temperature contribution (hot in to hot out) of the regenerator, deg R
25. TCH temperature contribution (cold in to hot out) of the regenerator, deg R
26. TCHI temperature contribution (cold in to hot out) of the intercooler, deg R
27. THHI temperature contribution (hot in to hot out) of the intercooler, deg R
28. WA22 flow rate into control volume 1, lbs/sec (necessary added lag between intercooler pressure drop and flow to permit balancing)
29. WA3 flow rate into control volume 4, lbs/sec (necessary added lag between intercooler pressure drop and flow to permit balancing)
30. QLOAD load torque, in-lbf (changes in load are modeled with a slight lag which produces this state)
31. WFUEL fuel flow, lbs/sec (actuator lag produces this state)
32. WFUELI integral of fuel flow, lbs
33. WFUELD derivative of fuel flow, lbs/sec**2
34. VSAPT FPT stator vane angle, deg
35. VSAPTD derivative of FPT stator vane angle, deg/sec
36. T4INTG integral of exit temperature of control volume 5, deg R

These states determine the A matrix, which describes the ICR turbine. Development of a controller using all 36 states would not be practical due to the complexity of the mathematics and the processing power that would be necessary to manipulate a 36 x 36 matrix. For this reason, a reduced order system is needed. A smaller number of states is chosen, based on probable significance in the control of the output variables. For this control attempt six variables were chosen. In earlier models as few as three and five variables were use with mixed success. There is no firm number that must be used. The six chosen were numbers 31 to 36 in the list above. These states were chosen because of their direct relationship to the magnitude, rate of change, and amount of error in the output variables, T4 and XNPT. The control used is a simple Proportional-Integral-Derivative (PID) controller. The PID is able to provide fast response to transients while still having a steady state error of zero. Both of these characteristics are critical in this system. The equations used in the controller are:

$$\begin{aligned} WFUEL &= K_1 * SPDERROR + K_2 * WFUELI + K_3 * WFUELD + \\ &\quad K_4 * DELT4 + K_5 * T4INTG + K_6 * VSAPTD + WFUELEQ \\ VSAPT &= KV_1 * SPDERROR + KV_2 * WFUELI + KV_3 * WFUELD + \\ &\quad KV_4 * DELT4 + KV_5 * T4INTG + KV_6 * VSAPTD + VSAPTEQ \end{aligned} \tag{4.1}$$

where K and KV are the gain matrices for the PID controller and VSAPTEQ and WFUELEQ are the equilibrium set points for the stator vanes and the fuel flow, respectively.

5.0 CONTROL DESIGN METHOD

To produce the four control modes and their respective gains, each mode was separated, analyzed using three methods of control design, and then linked together using programmed logic for entering and exiting the modes. There were three methods used for determining controller gains: 1) State-space pole placement; 2) Linear quadratic regulator; and 3) polyhedral search routine.

5.1 STATE FEEDBACK

While investigating the different modes, it was determined that the gain matrix K could be partitioned into four 3 x 3 submatrices, each linking an input variable with an output variable. Thus,

$$K = \begin{bmatrix} K_{1-3} & K_{4-6} \\ KV_{1-3} & KV_{4-6} \end{bmatrix} \tag{5.1}$$

where K_{1-3} is WFUEL control of XNPT, K_{4-6} is WFUEL control of T4, KV_{1-3} is VSAPT control of XNPT and KV_{4-6} is VSAPT control of T4. Foı the different modes, the control links not desired are simply zeroed out or given very small values.

State feedback follows the form[6]

$$\underline{u} = -\underline{g}^T\underline{x} \tag{5.2}$$

where $\underline{g}$ is an (n x 2) array of gains and $\underline{u}$ is a vector of system inputs, WFUEL and VSAPT. The gains will move the system poles to create a stable system. The PLACE command in Matlab is a useful tool for solving equation 5.2.

5.2 LINEAR QUADRATIC REGULATOR

Linear Quadratic Regulator[7] is a minimizing function which calculates the gain matrix in accordance with the cost function

$$J = \int_{0}^{t_f} (\underline{x}^T Q \underline{x} + \underline{u}^T R \underline{u})\, dt \qquad 5.3$$

where $\underline{u}$ is the input matrix and $\underline{x}$ is the state matrix. Q and R are square matrices, used for the weighting of elements, both states and inputs, to obtaining optimal gains.

5.3 POLYHEDRAL SEARCH

The polyhedral search routine[8] is a method of optimization feedback gains where n number of states are perturbed and 'plotted' in n+1 space. The program evaluates the individual iterations, determines the worst set of gains based on a cost function, and reflects this set of gains across the polyhedron created in n+1 space. This allows the program to walk its way towards the area of least cost. For example, to optimize a two variable system, the first three runs of the simulation would make a triangle on a two dimensional contour map of cost. This cost is determined by an arbitrary cost function based on factors which are to be optimized. The worst value, or corner of the triangle, would be reflected across the triangle to a point of increased optimization. This procedure is repeated numerous times.

5.4 DETERMINING CONTROL GAINS IN EACH MODE

5.4.1 MODE 1 DESIGN

5.4.1.1 STATE-SPACE POLE PLACEMENT

For Mode 1, initial attempts were made to produce a controller using state-space pole placement. The procedure for this is to freeze the desired states using the ACSL FREEZE command then produce reduced order A and B matrices in ACSL, using the JACOBIAN command. These are sent directly to Matlab for analysis. Tables 5.1 and 5.2 contain the A and B matrices obtained for Mode 1. On the first effort it was noted that the B and C matrices were all zeros. This was due to the lack of direct connection between the control states and the input variables. With a zero B matrix the system may be controllable, but any effort to determine the poles, or to perform other manipulations, produces zeros. For this reason the states left unfrozen by the FREEZE command in ACSL were extended to include the setpoints for the stator vanes and the fuel flow (VSAPTEQ and WFUELEQ). In the analysis of the state matrices, these values are included but they are not part of the control effort and these columns and rows, in the gain and eigenvalue matrices, are partitioned off.

Once the A and B matrices are in Matlab, the PLACE command is executed, to determine the values of the gain matrix elements needed to place the poles of the system in the position specified by the user. These gains are then inserted into the model and the response plotted. The eigenvalues obtained from the unmodified system are shown in Table 5.3. Classical state-space theory states that all system poles must be moved to the left half of the complex plane in order for the system to be stable. A number of the states created by ACSL are not true states of the model but are created mathematically as side effects of other calculations. These states are not covered by the theorem stated above. An example of this is the integral of shaft speed, shaft position. This is used in the control equation but it is an ever increasing number and no control effort can, or should, try to drive this value to zero. It is 'unstable' for this reason, but its eigenvalue is not moved because it does not truly affect the stability of the system. This pole is the one seen at 170.33.

Once the eigenvalues of the system are determined, the placement of the system poles must be decided upon. This requires some trial and error as well as practical experience. Figure 5.1 shows the results of the only stable system to be determined strictly by this method. The gains obtained from this attempt were then modified individually, based on knowledge of the effect each had on the system, to develop a stable response with the desired percent overshoot, time to peak, and steady state error.

Figure 5.2 shows the result of leaving the unstable pole at 170.33. This response is slightly better, but the cost of increased gains to move the pole is a consideration for leaving it alone. Figures 5.1 and 5.2 both have excellent temperature control but the speed control is unsatisfactory. Due to the use of a reduced order system, complicated mathematical methods of determining pole placement become ineffective because the reduced order system is an estimation of the true system, so the precision in these methods would be lost in the error of the estimation. The gains for the controller must be handpicked, generally based on some insight into the system from some other source. The polyhedral search routine provided that insight.

5.4.1.2 POLYHEDRAL SEARCH

The polyhedral search routine provided the most useful approach to designing of a controller for a reduced-order system. Because of the inexact nature of a reduced-order system and the consequent inability to use strictly classical design methods, the trial and error processing performed by the polyhedral search is an ideal approach to optimizing this control effort. Typically, a near approximation is derived from pole placement or LQR. These values are then used as the starting gains in the search routine. As many as 50 to 100 iterations of the model will be done with intermediate results displayed after every 5 to 10 iterations. These results include the value of the lowest cost function up to that point, and the gains which give that cost. Table 5.4 lists a set of intermediate gains obtained for one of the runs. Figures 5.3 and 5.4 are the plots speed error and temperature error obtained with these gains. Progress can be seen, compared to Figures 5.1 and 5.2, as the search minimizes steady state error and ringing in the response to the standard load profile.

5.4.1.3 LINEAR QUADRATIC REGULATOR

The final method of control design used for Mode 1 was the linear quadratic regulator (LQR). This method was primarily attempted to prove its applicability to a reduced order system. Figure 5.5 shows the best response obtained by LQR. The response is not acceptable for a controller, but it does demonstrate that the LQR method produced gains which increased stability, proving that the method is viable. The weighting matrices, Q and R, used in the minimizing function were initially identity matrices. The results from this initial effort were very unstable. Examination of the cost function showed that when using just the identity matrix for Q and R, the state matrix (8 x 8) overpowered the control matrix (2 x 2) when applied to a least square minimizing function. The solution to this was to scale the Q matrix by various powers of 10 and examine these results. This effort produced gains, shown in Table 5.5, very similar to those obtained by the polyhedral search and gave the response in Figure 5.5. This result proves the applicability of LQR but extensive time was not spent in examining the effects of individual perturbation of all elements in the weighting matrices. This would be a considerable study in and of itself.

5.4.2 MODE 2 DESIGN

Based on the attempts and mistakes made in the design of Mode 1, the polyhedral search routine was used to obtain the feedback gains for Mode 2. The set of gains needed for this mode are K_{4-6} and KV_{4-6}, which are the gains related to the control of temperature. Arbitrary values were chosen as the starting point for the search. The important criterion for the values is the relationship between the magnitude of the fuel flow values and the stator vane values. To avoid conflicting control efforts, they need to be of differing magnitude. The vane values were started at half the size of the fuel flow. In control terms this makes the vanes roughly twice as quick in their response. The results of the search, shown in Figure 5.6 and 5.7, produced an excellent response. The Mode 2 controller overcame the add-load speed error in less than 0.2 seconds and never exceeded a temperature error of 100°R. This is approximately one-fifth the time and onetenth the temperature error of the Mode 1 controller's response to an identical add-load. During the critical time when temperature is going to do permanent damage to the turbine, shaft speed error is no longer considered. This can be seen in the XNPT plot in Figure 5.7. The values on the speed error deviate substantially from the setpoint due to the fuel flow drop to control temperature. In this situation, the protection of the turbine is much more important that maintaining a steady speed. Mode 2 has priority over all others due to the damaging nature of excessive temperatures. The controller will exit any other mode when an over temperature situation is detected.

5.4.3 MODE 3 DESIGN

As with the design of Mode 2, the polyhedral search was used on this mode. Mode 3 is primarily concerned with correcting excessive speed error. Both inputs will be used to control the speed so the important quadrants in the gain matrix are K_{1-3} and KV_{1-3}. Because of its increased effectiveness over speed error, fuel flow was chosen to be the faster variable. For this Mode a guideline was used to help determine the effectiveness of the controller. The American Society of Mechanical Engineers has determined that any turbine supplying load to electrical circuits, in response to a transient, should be able to stabilize to within one percent speed error in less than one second. The Mode 1 controller was unable to achieve that specification, but with both inputs attempting to control the speed error, the specification is easily met. Figures 5.8 and 5.9 show the temperature and speed error response in Mode 3. The second and third modes especially show the advantages of having two input variables and the addition of the stator vanes to the model. Previously there was nothing to counteract the temperature spikes and drops during load changes. Substantial increases in efficiency as well as controllability have come with the addition of the variable angle stator vanes.

5.4.4 MODE 4 DESIGN

The design of Mode 4 was the simplest, but this mode has room for creative variations that have not been attempted. The gains for the fuel flow are identical to Mode 1, but the stator vanes are set wide open to maximize air flow through the turbine. The purpose of this mode is to increase spool speed as well as increase air flow as a preemptive build up of potential energy in the rotors. Figure 5.10 shows increased speed in the low pressure turbine (FXNL), in response to the changing vane angle. This energy can be instantly transferred to power turbine torque when the load increases. This figure also shows the slight increase in power turbine speed. A substantial first order lag (delay time constant of 3 seconds) is needed in the vane actuation to prevent instability in the system due to drastic change in air flow. A time lag of this magnitude will not be noticed, even by the operator.

5.5 COMPLETE 4 MODE CONTROLLER

The final step in the four-mode controller design is the integration of the four individual modes into a unified controller with stable, smooth transition and steady operation. Good results were not obtained on the initial attempt. The greatest difficulty is the implementation of logic needed to determine the mode of operation. Table 5.6 is a set of criteria used to determine the mode to be selected as well as exit conditions and mode priorities. Appendix A is an excerpt from the ICR turbine model fuel and vane control section showing the logic code for entering and exiting each mode. The priority can be seen in the code as well as Table 5.6, where all states will be exited upon any over temperature situation and Modes 1 and 4 will be exited upon any over speed situation.

A prime case for the usefulness of modeling is seen in the next example. In Mode 2, the amount of speed error is not considered, as it is secondary in importance to an excessive temperature condition, yet the controller is still calculating the integral of speed error. At the exit of Mode 2 the integral has grown to an overpowering size and creates instability when it again is considered in the control equation. The same effect is true in Mode 3 for under temperature error (over temperature will activate Mode 2 due to priority). To alleviate this, these integrals are simply set to zero when not being considered in the control equations. This is an important step that was missed in paper analysis, but quickly spotted in simulated runs of the model.

Figure 5.11 is the final result of the integration of the four modes into a smooth control effort. At time equal one second the load is dropped to 50%. The speed error is corrected within one second. The switching between Mode 1 and 3 is clean and smooth. The entire system is stable in just over one second. When the load is increased to 100%, Mode 2 is activated to protect against the over temperature common during add loads. The vanes snap open quickly to provide as much cooling as possible. Because of this, fuel can be added quite rapidly. Steady state is achieved in approximately six seconds. From this plot, the four modes appear to work as expected and provide smooth, rapid response to critical situations to protect the turbine while providing efficient power.

6.0 CONCLUSION

The results of this project are very positive. The final model of the turbine is a complete, accurate representation of the actual turbine it is emulating. The controller is very effective, working at least as well as hoped for, and meeting many of the specification which were used as guidelines for the optimization. The four modes are able to quickly control temperature and speed, providing an efficient standard operating setpoint for the turbine. Overall, the design of a four mode, MIMO, state-space controller for a reduced order system was ultimately very successful.

Table 5.1

System A Matrix

-6.613	0	0	0	0	0	1.4309	-1.325
-0.999	0	-200.0	0	0	0	0	0
4.00E6	0	0	0	0	0	0	0
0	0	0	170.32	0	0	3.59E4	0
0	0	0	-0.999	0	0	0	0
0	0	0	3.99E6	0	-200.1	0	0
0	0	0	0	0	0	-19.99	0
0	0	0	0	0	0	0	-100.0

Table 5.2

System B Matrix

0	0
0	0
0	0
0	0
0	0
0	0
19.999	0
0	100.04

Table 5.3

System Eigenvalues

-200.06
0
170.33
-200.01
-6.6132
-100.04
-19.999

Table 5.4

Polyhedral Search Gains - Modes 1-3

Mode 1		Mode 2		Mode 3	
0.19719	0	0	0	0.2984	0.1979
0.90237	0	0	0	0.3970	0.8995
0.97E-5	0	0	0	-1.4E-3	-3.7E-4
0	9.72E-4	0.1731	0.0875	0	0
0	1.24E-3	0.2195	0.1108	0	0
0	0	7.07E-5	-6.1E-5	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Table 5.5

LQR Gains - Mode 1

0.14	0.21
0.45	-1.3
7.0E-6	0
0.08	9.0E-5
0	1.0E-3
0	0
0	0
0	0

Table 5.6

Mode Switching Guidelines

Mode	Priority	Enter	Exit
1	4th	Default Mode	On activation of other modes.
2	1st	T4 error > 50°	T4 error < 1°
3	2nd	Not in Mode 2 XNPT error > 0.2%	Mode 2 activated XNPT error < 0.2% d/dt(XNPT) < 10
4	3rd	Not Mode 2 Not Mode 3 Operator Selected	At add-load control switches to Mode 3

Table 6.1

Explanation of Figures

FVSAPT	=	(VSAPT-LOWER LIM)/(UPPER LIMIT-LOWER LIMIT)
FT4	=	-T4 ERROR/100
FXNPT	=	(XNPT ERROR*3600)/100
FXNL	=	XNL ERROR/100
FQLD	=	LOAD/507704(100% RATED LOAD)

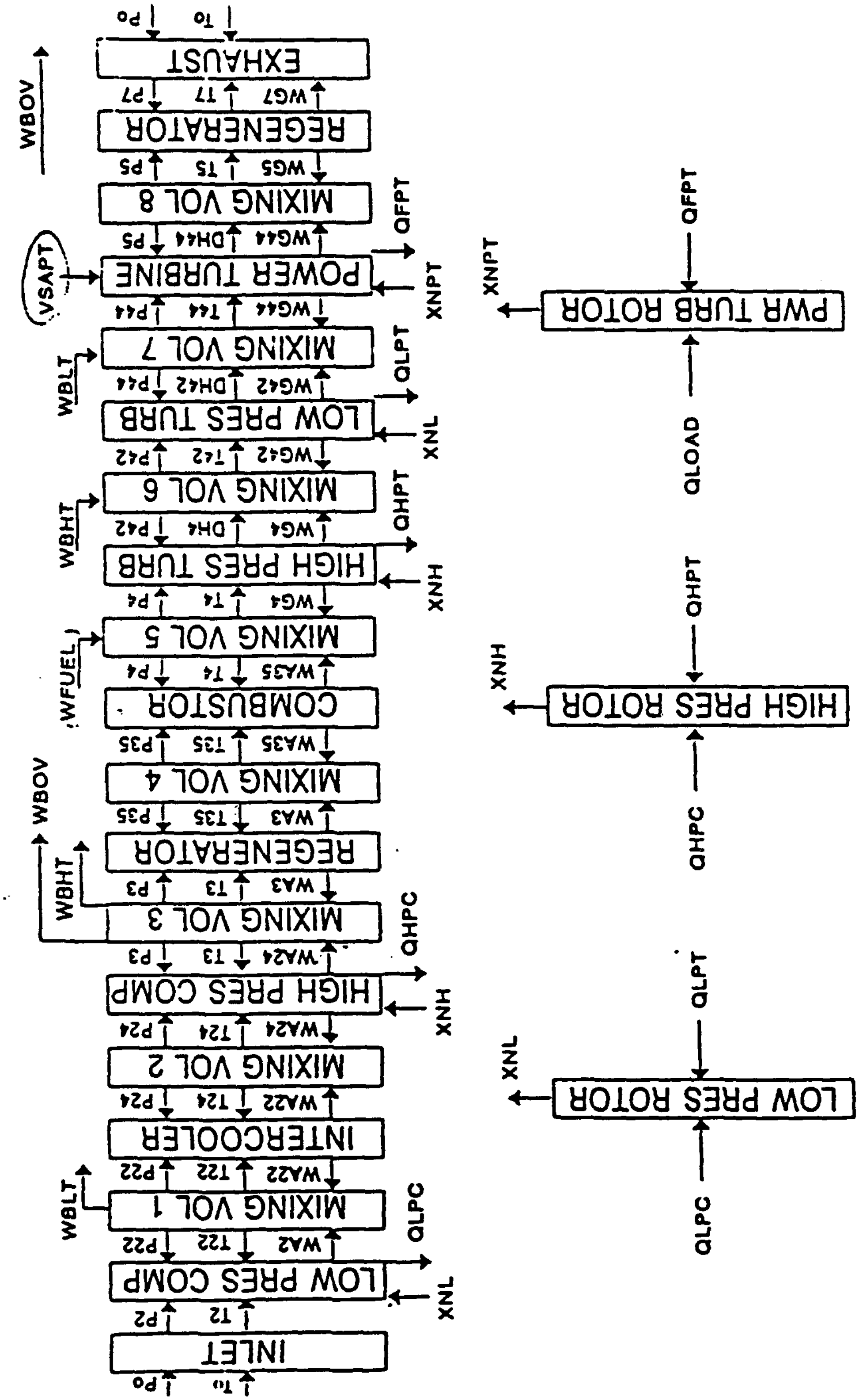

Figure 3.1: ICR Gas Turbine Simulation Flow Chart

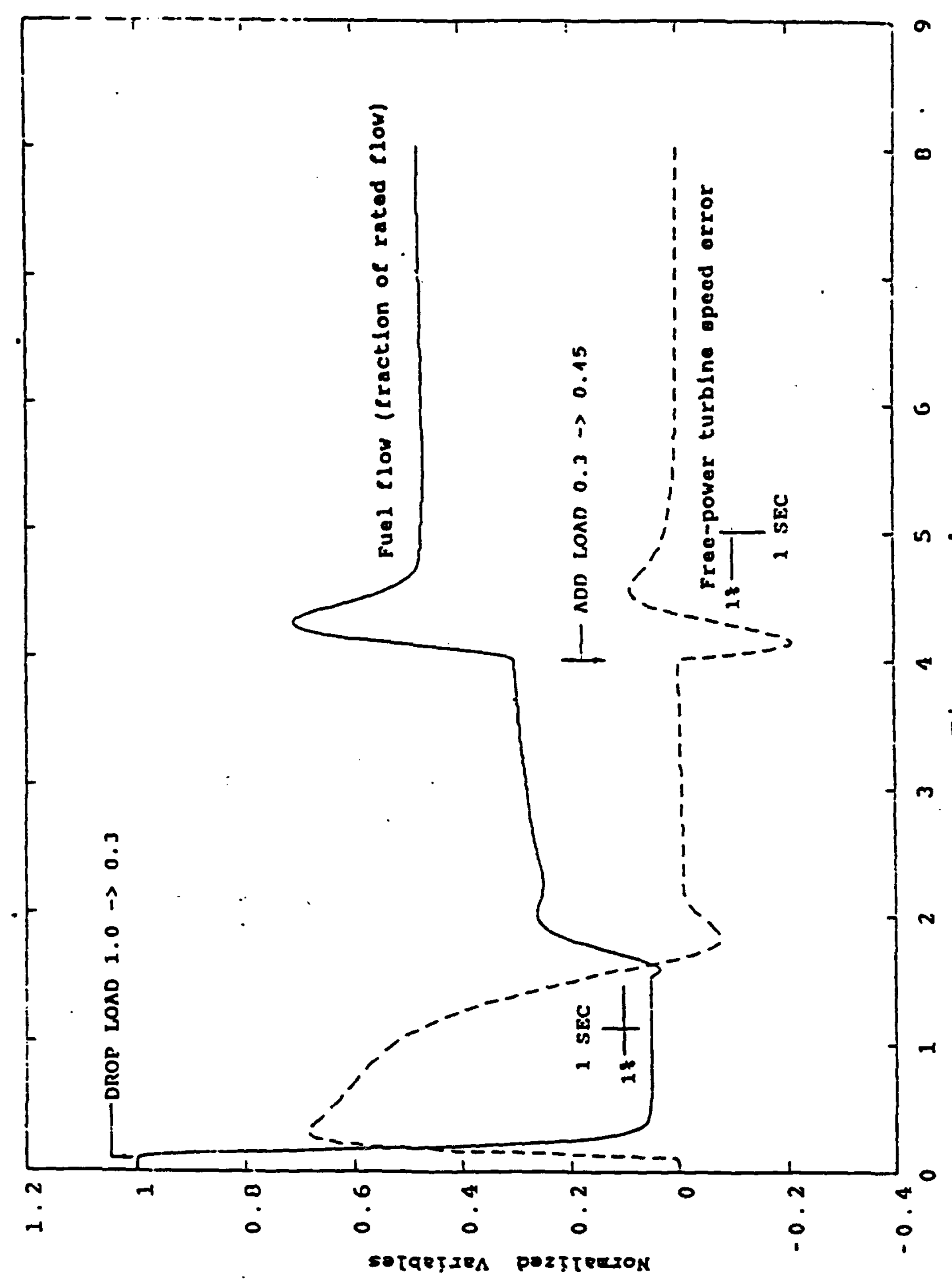

Figure 3.2: Three-State SISO Controller Response

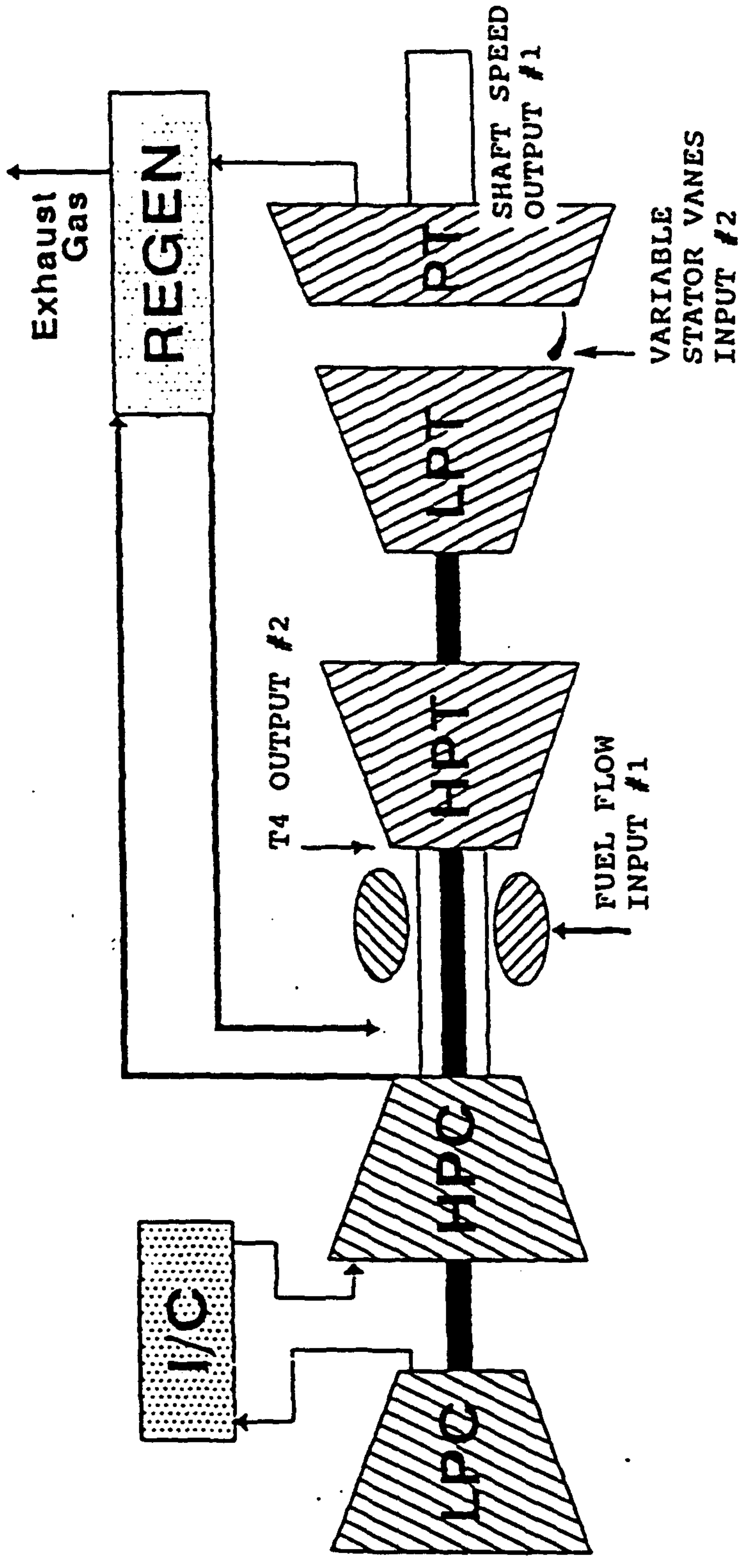

Figure 3.3: ICR Gas Turbine Model Diagram

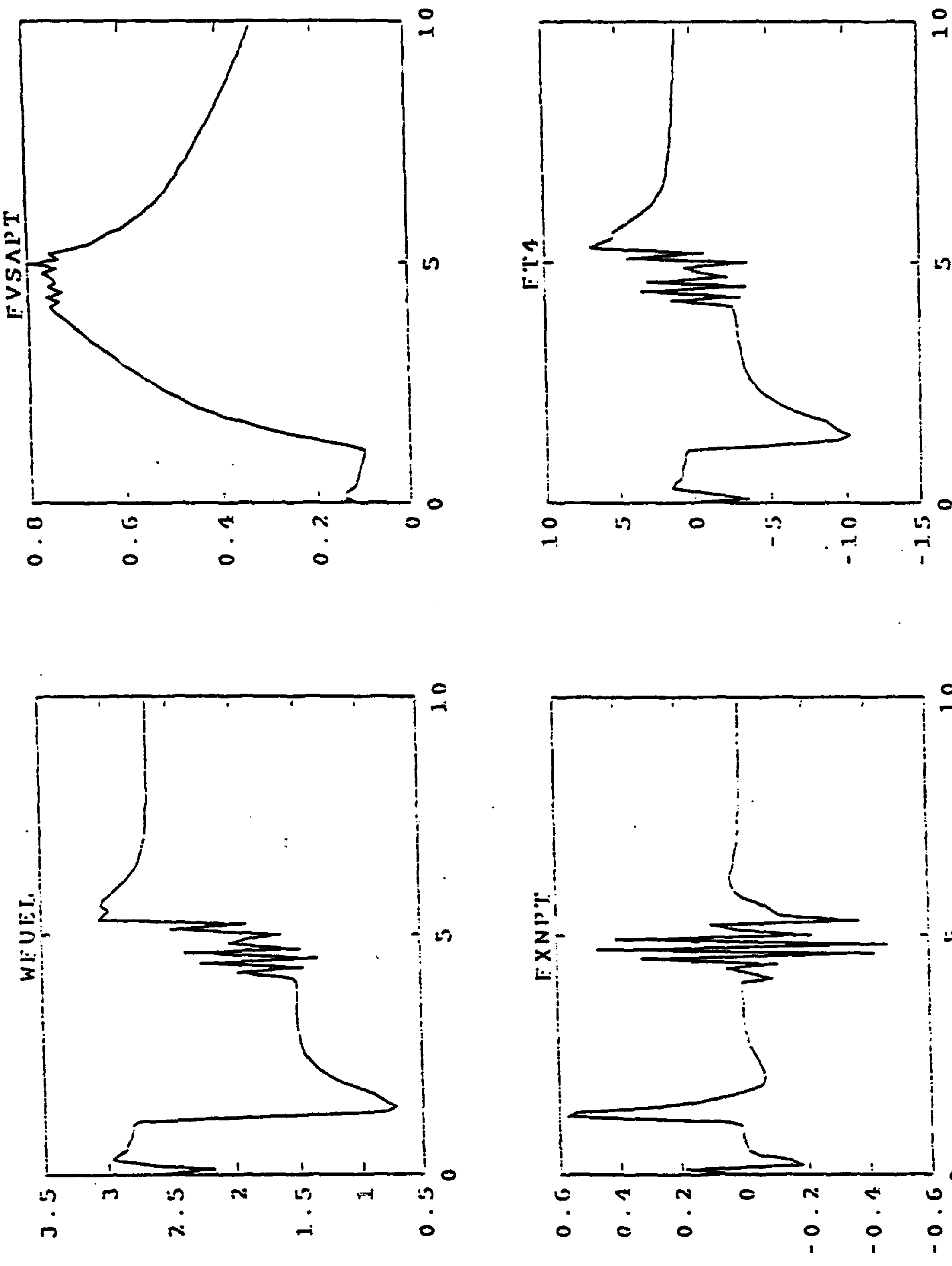

Figure 4.1: State Space Design, Ringing Response

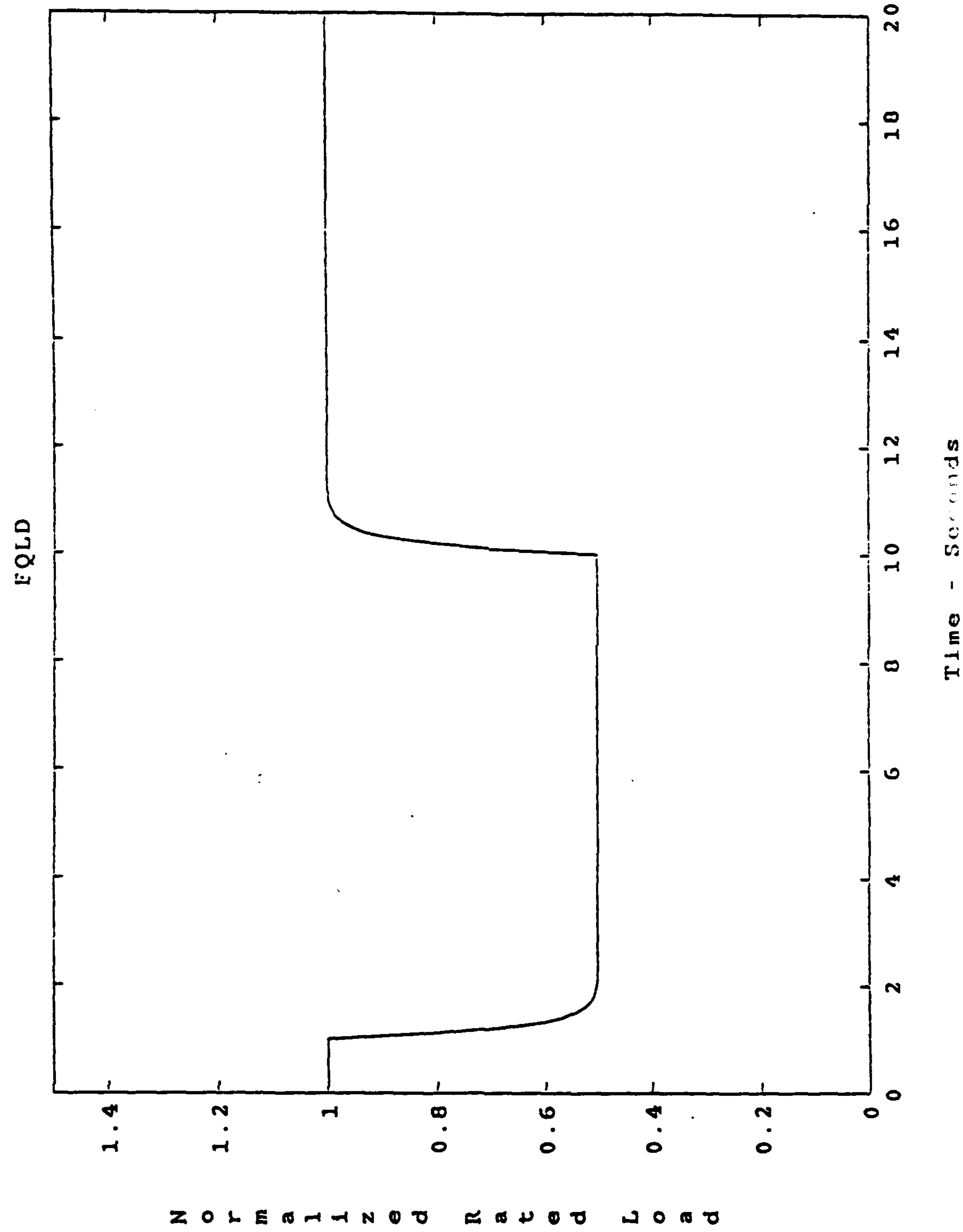

Figure 4.2: Standard Load Profile

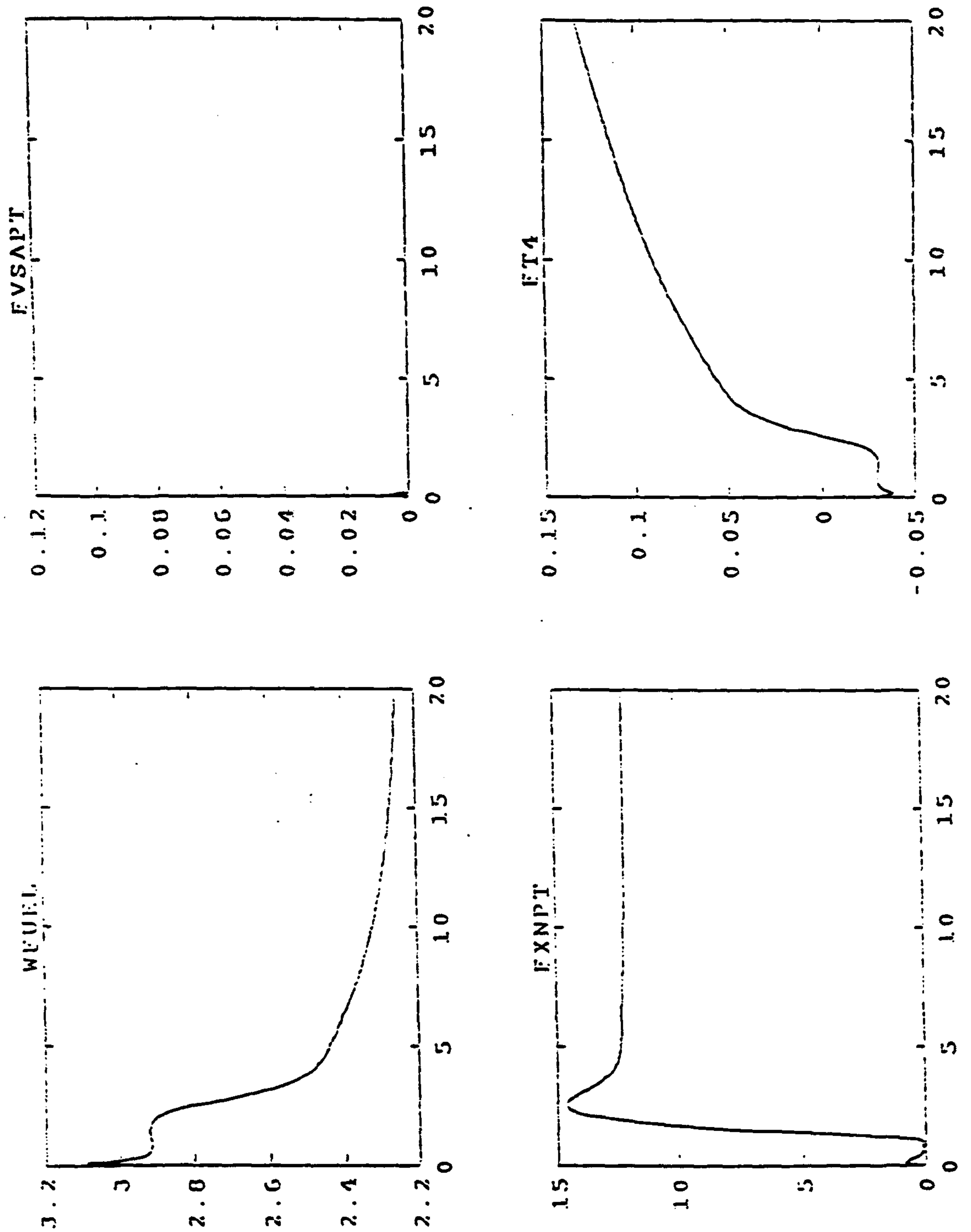

Figure 5.1: State Feedback Pole Placement Design

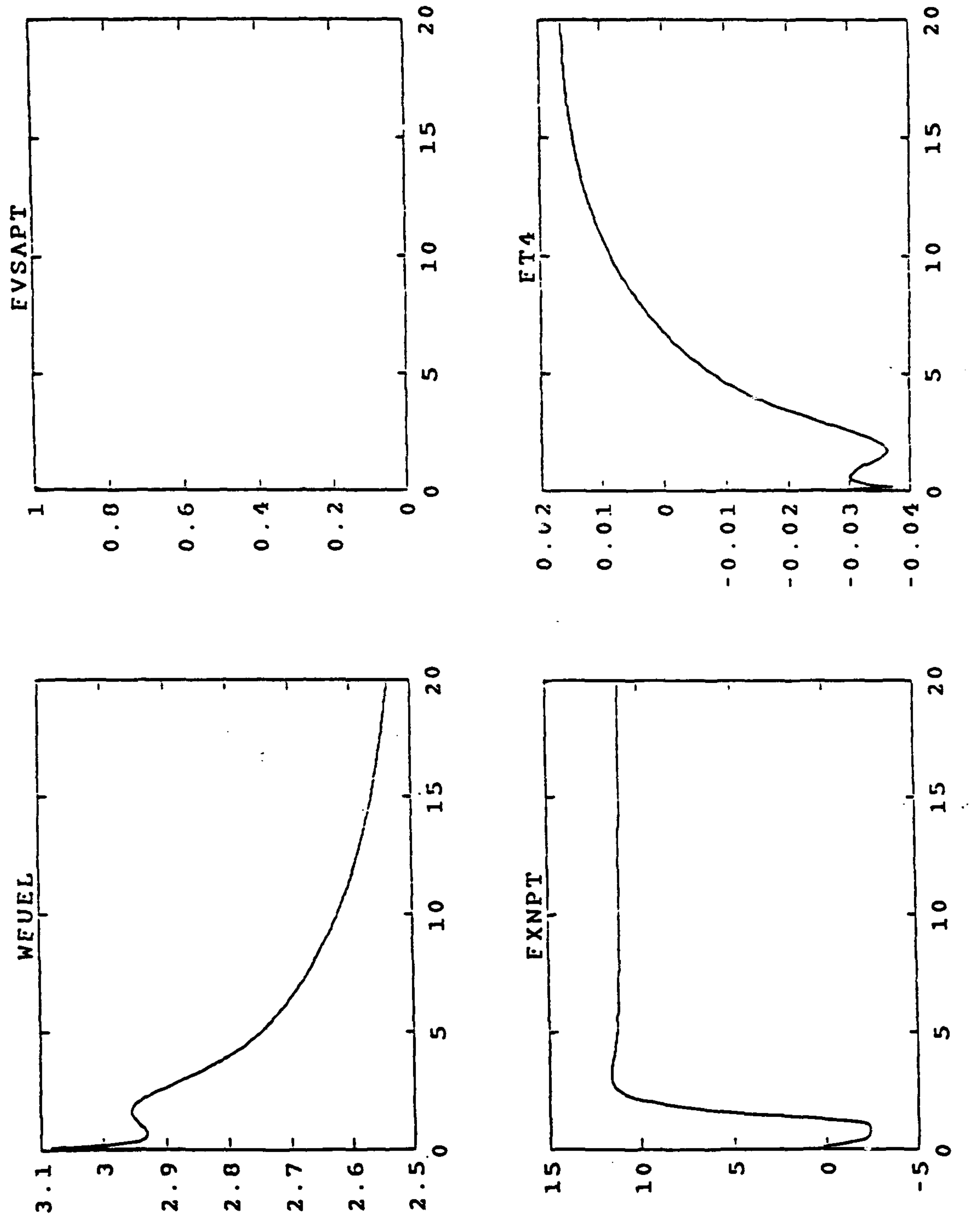

Figure 5.2: State Feedback Design With Pole at 170.33

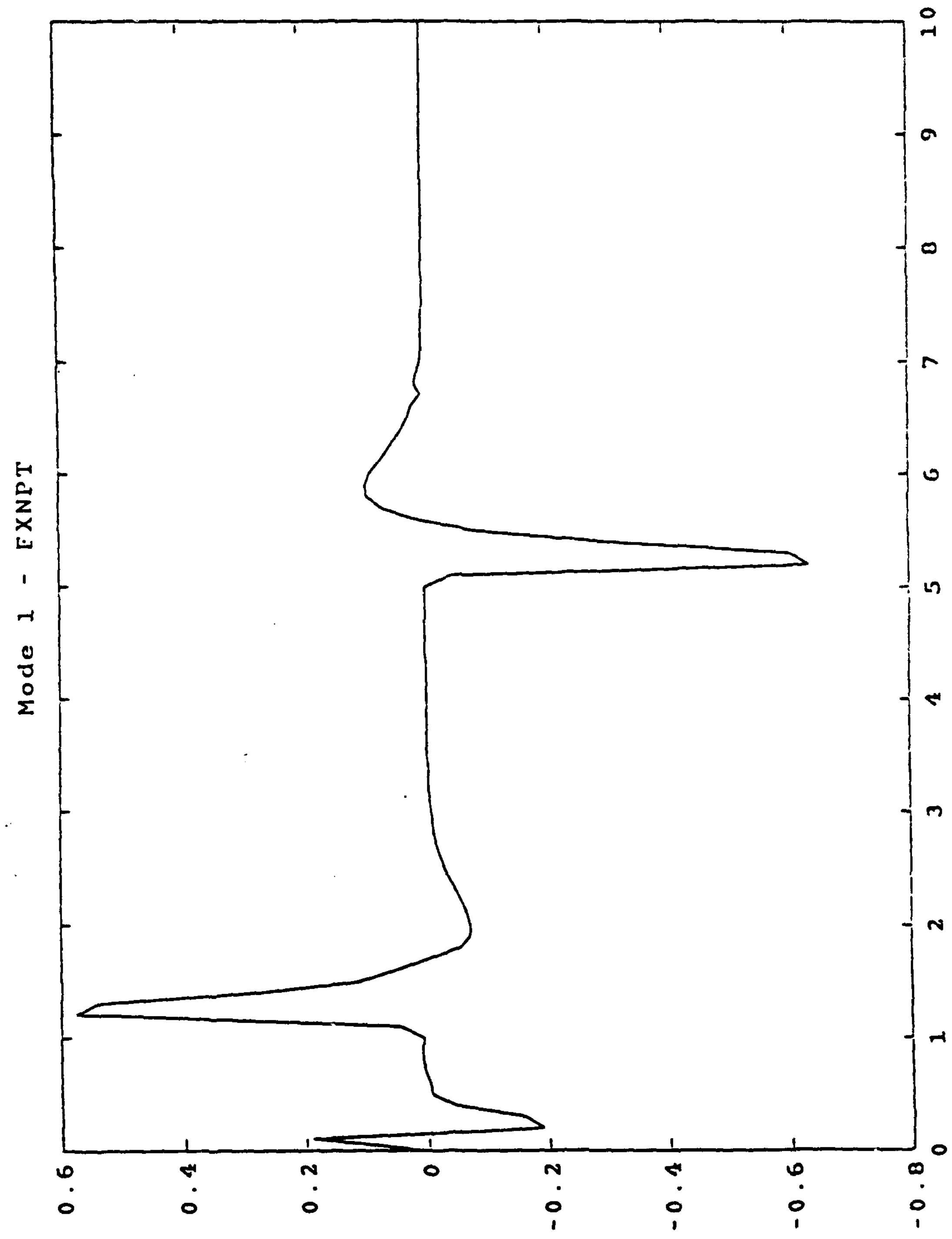

Figure 5.3: Mode 1, Polyhedral Search, FXNPT

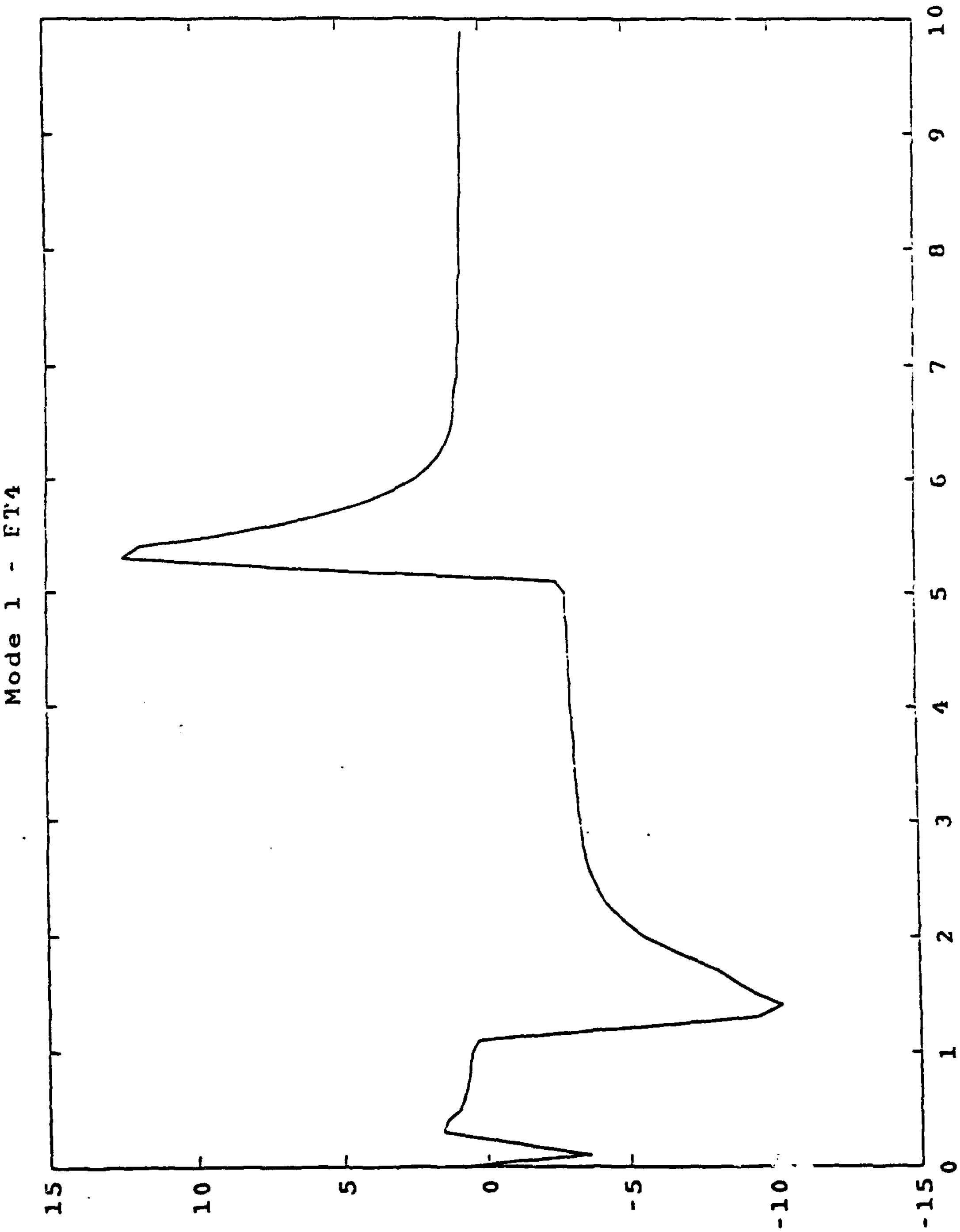

Figure 5.4: Mode 1, Polyhedral Search, FT4

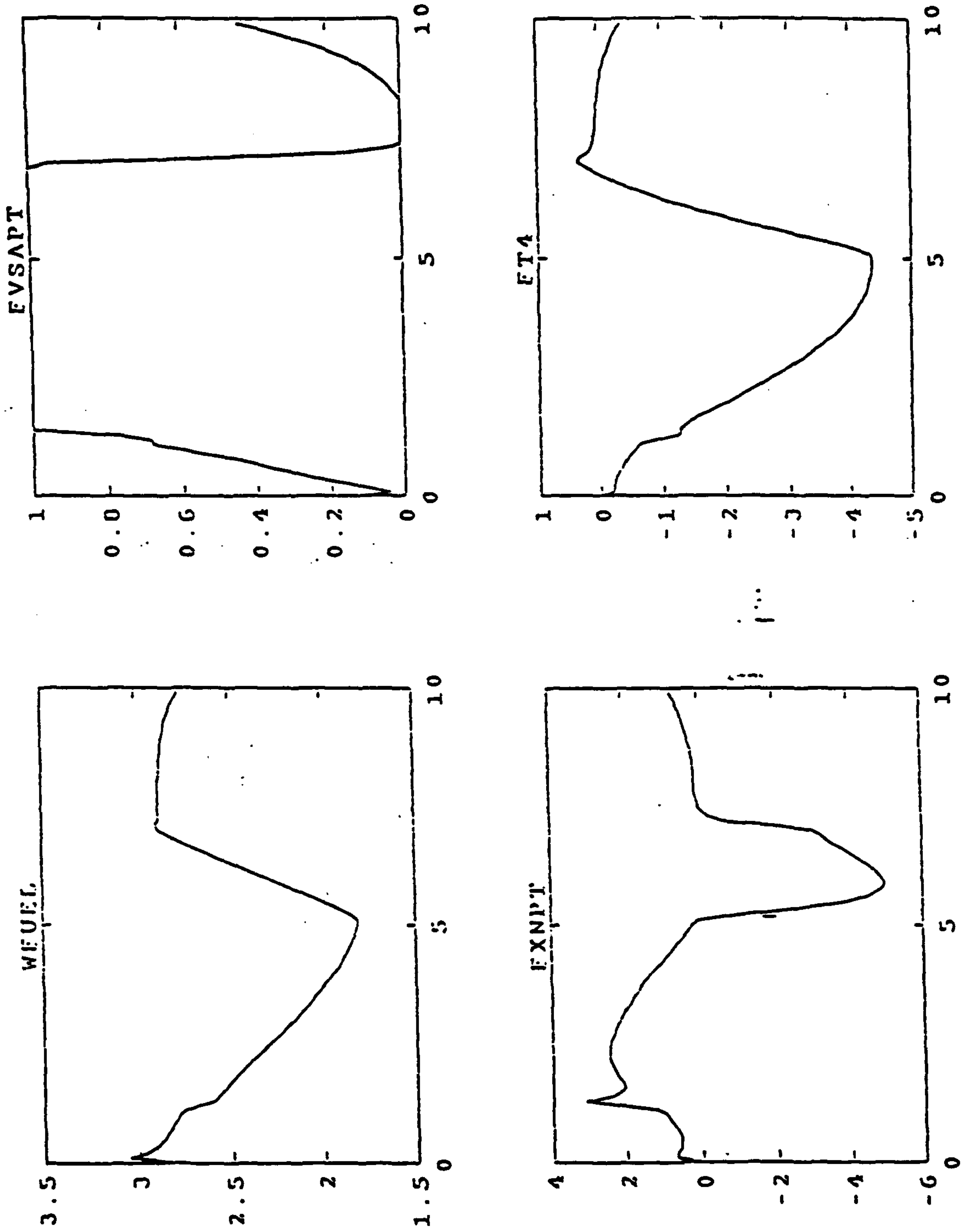

Figure 5.5: LQR Design Response

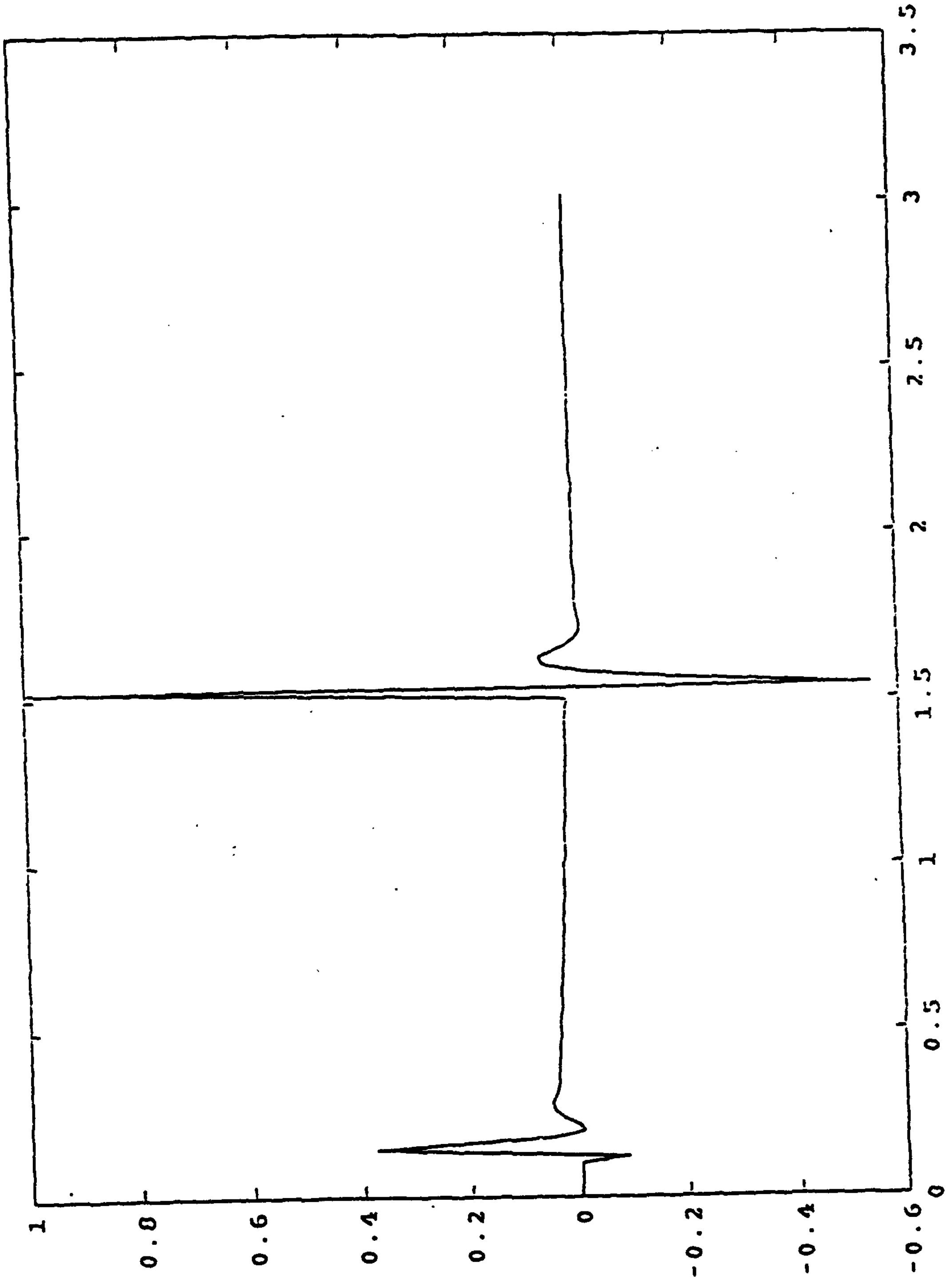

Figure 5.6: Mode 2, FT4

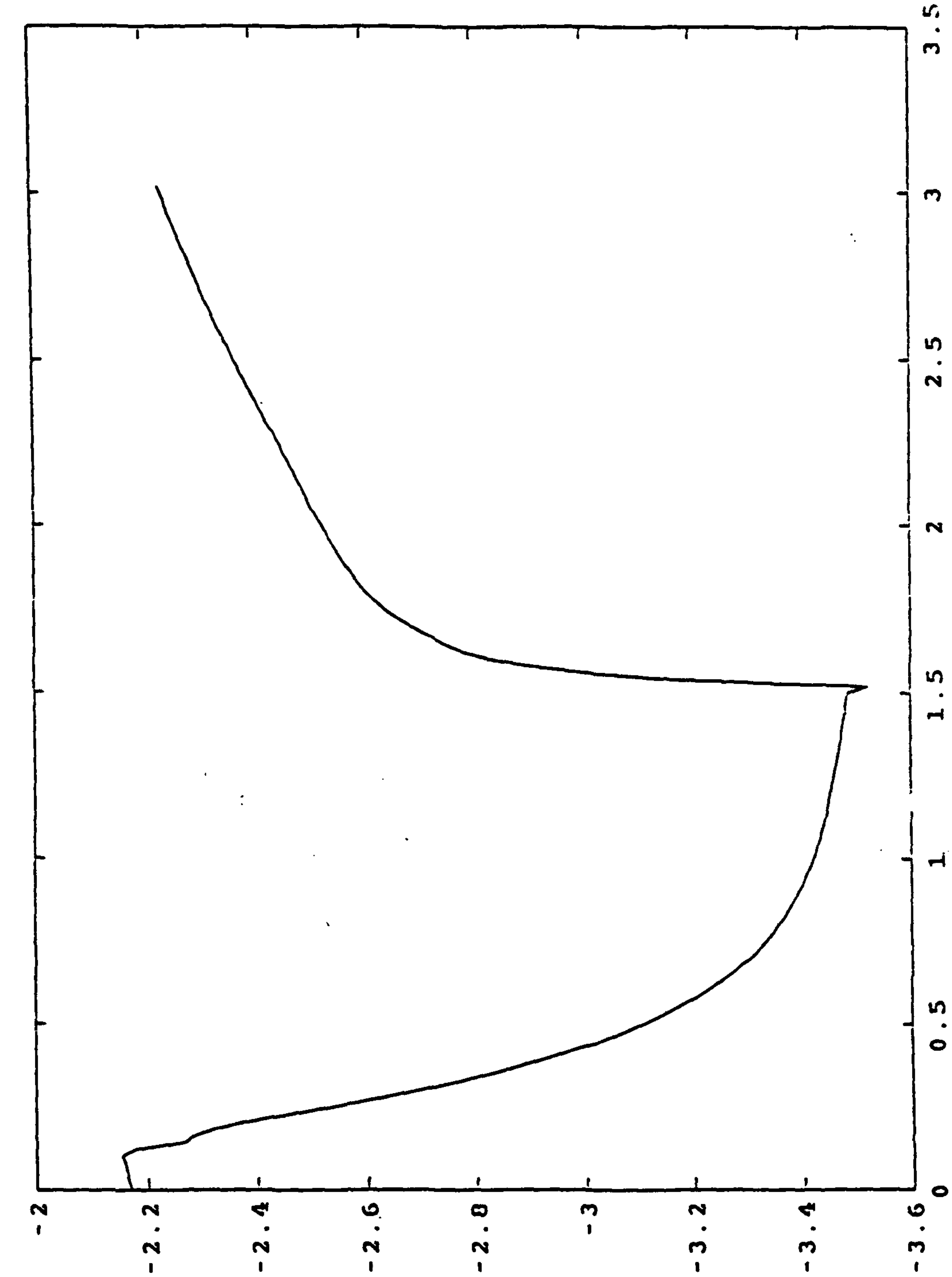

Figure 5.7: Mode 2, FXNPT

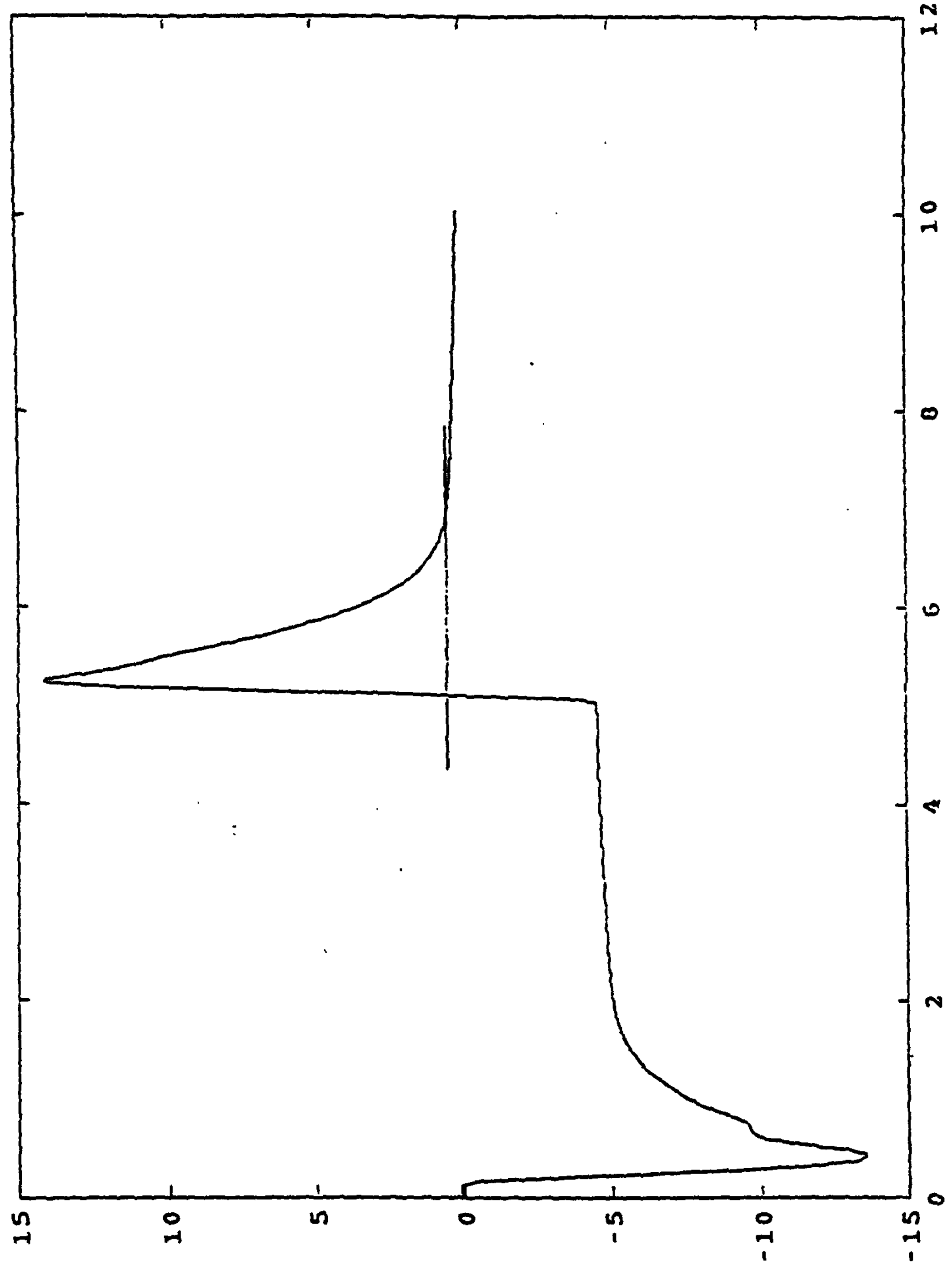

Figure 5.8: Mode 3, FT4

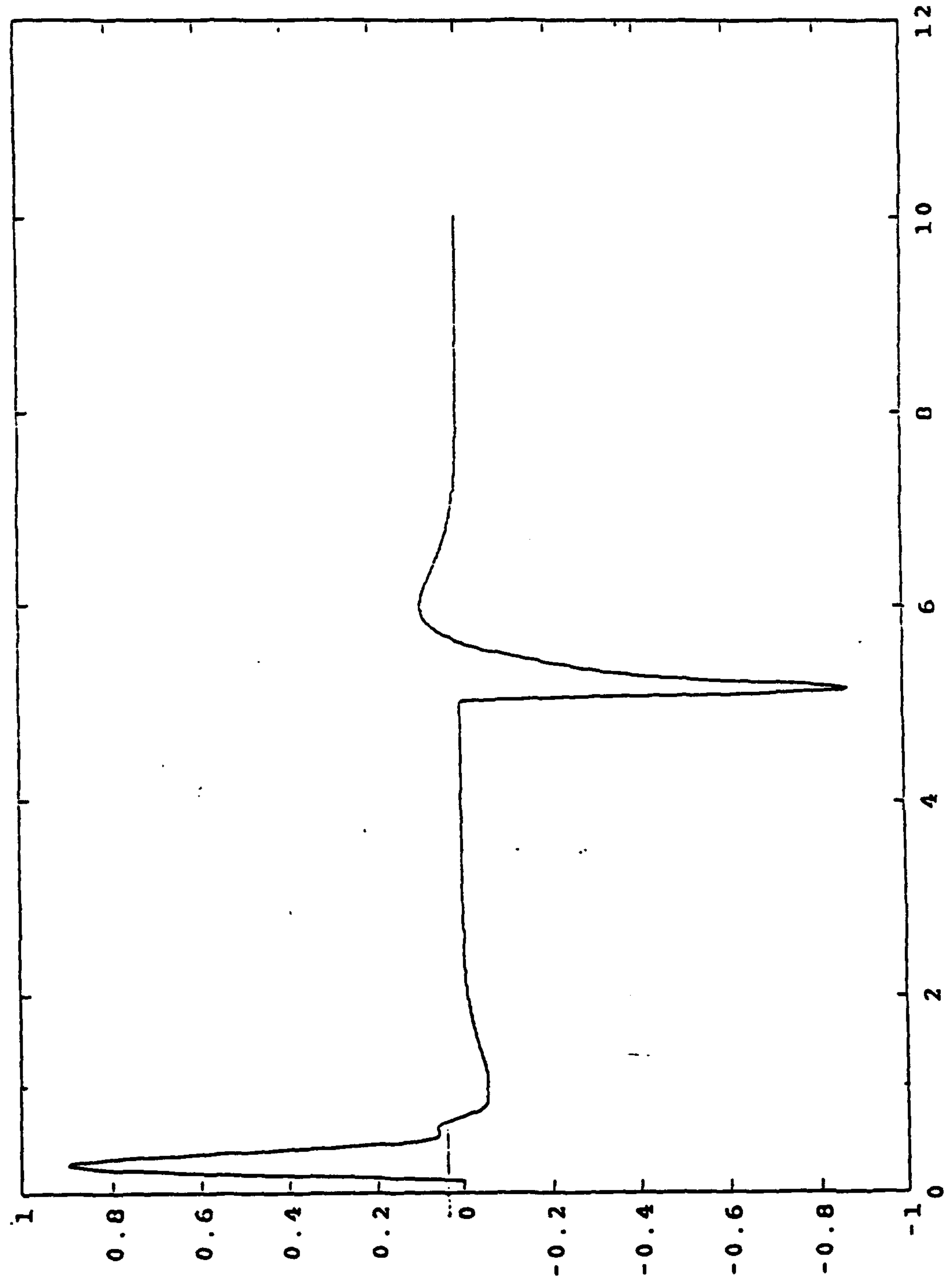

Figure 5.9: Mode 3, FXNPT

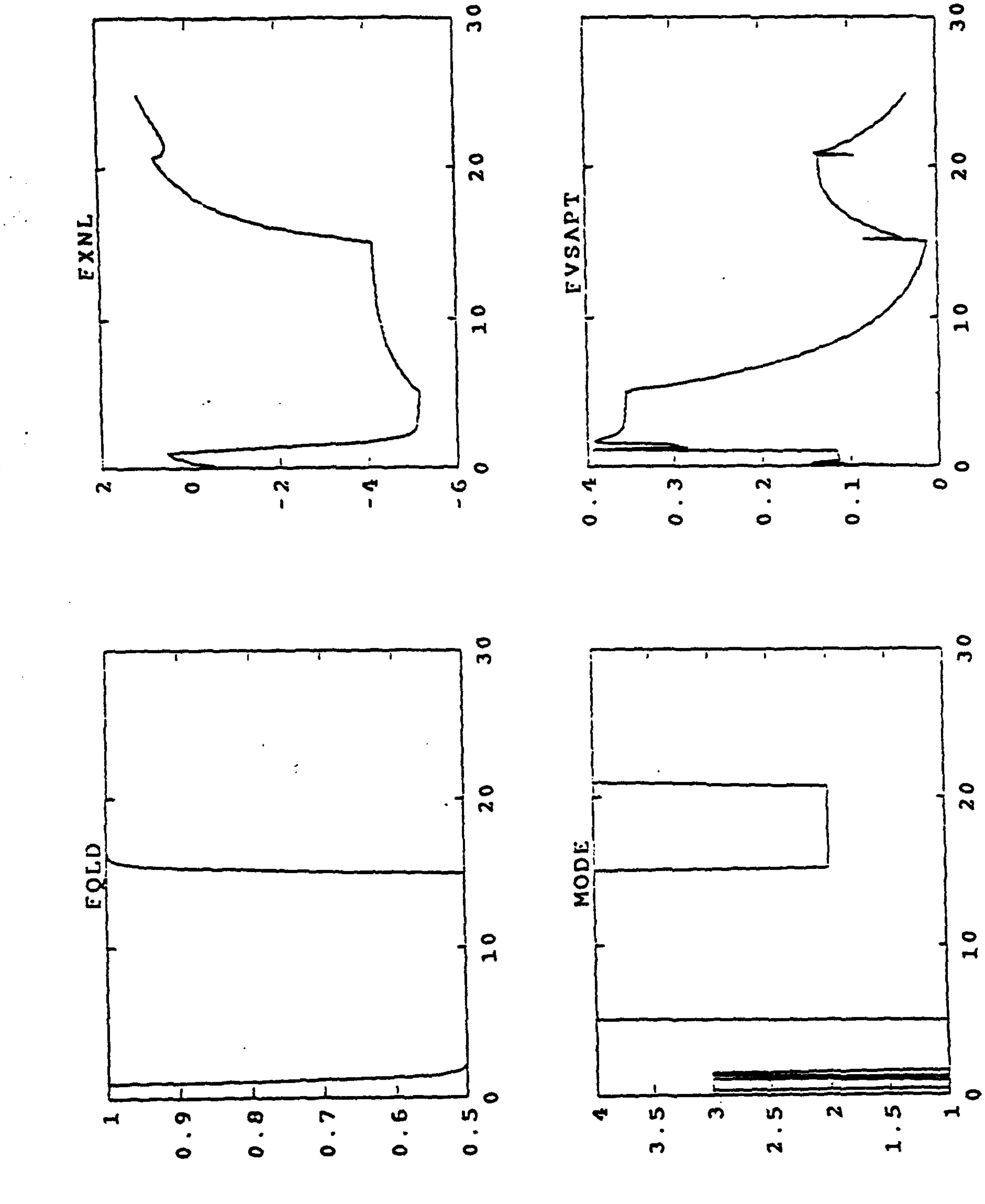

Figure 5.10: Mode 4, FQLD, FXNL, FVSAPT, MODE

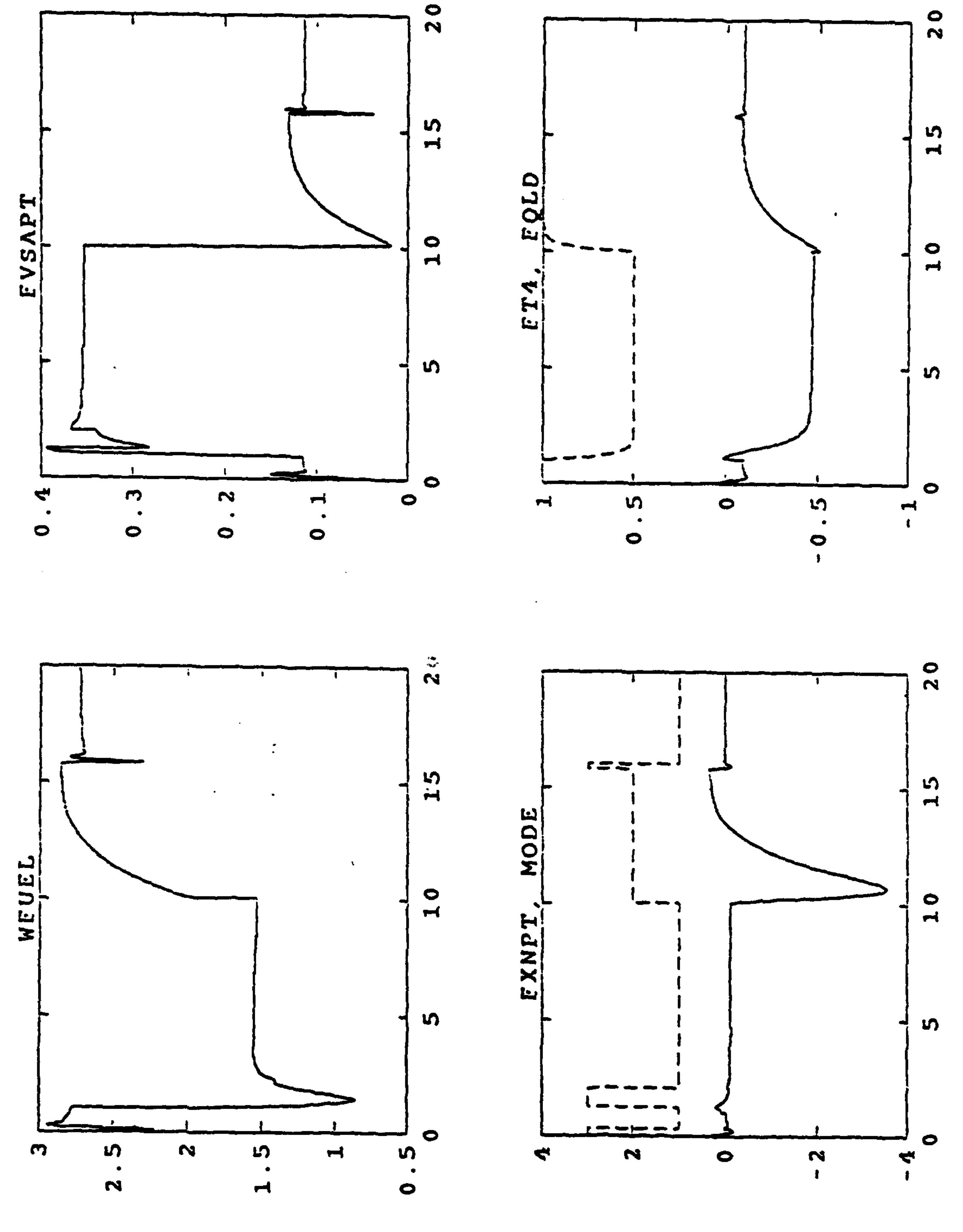

Figure 5.11: Integrated Controller

References

1. Watts, J.W., Dwan, T.E., and Brockus, C.G., "Optimal State Space Control of a Gas Turbine Engine," Transactions of the ASME, J. of Engineering for Gas Turbines and Power, Vol. 114, pp763-767, Oct 1992.

2. Watts, J.W., Dwan, T.E., and Garman, R.W., "A Model and State-Space Controllers for an Intercooled, Regenerated (ICR) Gas Turbine Engine," ASME Paper No. 92-GT-43, 1992.

3. Watts, J.W., Dwan, T.E., Olson, M.A., and Garman, R.W., "A Reduced Order State-Space Controller for an Intercooled, Regenerated Gas Turbine Engine," The Proceedings of the 1992 Summer Computer Simulation Conference, pp 632-636, The Society for Computer Simulations, San Diego, CA, 1992.

4. Mitchell and Gauthier Associates, Advanced Continuous Simulation Language Reference Manual, 1986.

5. PRO-MATLAB, the Mathworks, 1987.

6. Phillips, C.L. & Harbor, R.D., Feedback Control Systems, 2nd Ed., Prentice-Hall, 1991.

7. D'Azzo, J.J & Houpis, C.H., Linear Control System Analysis and Design, McGraw-Hill, 3rd Ed., 1988.

8. Himmelblau, David M., Applied Nonlinear Programming, McGraw-Hill, 1972.

Appendix A

_ZIPPY$DKA300:[93.PRIGGE]CONTROL.CSL;2

```
! ********** Fuel & Vane Control **********

! Inputs:
!   XNPT,T4
! Outputs:
!   WFUEL,VSAPT

CONSTANT  WFUELIIC=0.0,      WFUELDDIC=0.,...
          TAUWFD=.005,       TAUXNPTSET=.2,   XNPTSETIC=1.0,...
          TAUWF=.05,         WFUELIC=2.72562,...
          PDF=100.,          QDF=.005,...
          WFLLIM=.16667,     WFULIM=6.,       SPDI=0.,...
          XNPTSET=1.,        T4SETV=2800.,    VSAPTLL=75,...
          VSAPTUL=88.4,      VSAPTIC=76.5,    TAUVS=.01,...
          TAUVS1=1.,...
          VSAPTDIC=0.,       VANEI=0.,        TAUVS4=.1,...
          T4TRIPM2=2850.,    T4SETM4=2820.,   TESTOVER=5.,...
          OFFSPD=.7,         OFFSPD1=2.,      SPDSETM4=1.002,...
          T4SETBIC=2800.,    T4TRIPM4=2900.,  VSM4IC=76.5

  CONSTANT T4SCALE=100.,     XNPTSCALE=100.,  WGSCALE=100.,...
          WGFL=150.,         QLDREFIC=507704.

QLOADREFA=REALPL(TAUXNPTSET,QLOADREF,QLDREFIC)

PROCEDURAL(WFUEL,VSAPT,MODE,SPDERROR,SPDERROR1,DELT4 = ...
          T4,K,KV,VSAPTEQ,WFUELEQ)

XNPTSETA = REALPL(TAUXNPTSET,XNPTSETB,XNPTSETIC)
SPDERROR = (3600*XNPTSETA - XNPT)/RADRPM  !multiply xnpt based gains
                                          !by RADRPM this pertains
                                          !to K(1),K(2),K(3)

SPDERROR1= SPDERROR*RADRPM                !units are rpm
                                          !use for plotting

!MODE ENTER LOGIC
IF(.NOT.NEWMODE)GOTO M
IF(T4.GE.T4TRIPM2)GOTO SG2
IF(ABS(SPDERROR).GE.OFFSPD1)GOTO SG3
IF(MODE4)GOTO SG4

!MODE 1 - NORMAL OPERATION WFUEL C. XNPT AND VSAPT C. T4
  K(1)=.197191      $ K(2)=.902375          $ K(3)=.97E-5
  K(4)=0.           $ K(5)=0.               $ K(6)=0.
  KV(1)=0.          $ KV(2)=0.              $ KV(3)=0.
  KV(4)=9.7214E-4   $ KV(5)=1.2472E-3       $ KV(6)=0.
MODE=1
T4SET=T4SETV
XNPTSETB=XNPTSET
WFUELI=0.  $ T4INTG=0.     !reset integrators
GOTO M

 !MODE 2 - OVERTEMP RECOVERY MODE
SG2..  K(1)=0.      $ K(2)=.0               $ K(3)=.0
       K(4)=0.08746 $ K(5)=0.11083          $ K(6)=0.
```

_ZIPPY$DKA300:[93.PRIGGE]CONTROL.CSL;2

```
          KV(1)=0.          $ KV(2)=0.              $ KV(3)=0.
          KV(4)=0.1731      $ KV(5)=.21946          $ KV(6)=0.
        MODE=2
        T4SET=T4SETM4
        XNPTSETB=SPDSETM4
        WFUELI=0. $ T4INTG=0.     !reset integrators
        NEWMODE=.FALSE.
        GOTO M

        !MODE 3 - OVERSPEED RECOVERY MODE
SG3..     K(1)=.19798       $ K(2)=.89952           $ K(3)=.0
          K(4)=0.           $ K(5)=0.               $ K(6)=0.
          KV(1)=0.2984      $ KV(2)=0.39696         $ KV(3)=0.
          KV(4)=0.          $ KV(5)=0.              $ KV(6)=0.
        MODE=3
        T4SET=T4SETV
        XNPTSETB=XNPTSET
        WFUELI=0.  $ T4INTG=0.    !reset integrators
        NEWMODE=.FALSE.
        GOTO M

        !MODE 4 - HIGH IDLE PREPARATORY TO ADDING LOAD
SG4..     K(1)=.197191      $ K(2)=.902375          $ K(3)=.97E-5
          K(4)=0.           $ K(5)=0.               $ K(6)=0.
          KV(1)=0.          $ KV(2)=0.              $ KV(3)=0.
          KV(4)=0.          $ KV(5)=0.              $ KV(6)=0.
        MODE=4
        XNPTSETB = SPDSETM4
        NEWMODE=.FALSE.

M..     CONTINUE

        T4SETB = REALPL(TAUVS4,T4SET,T4SETBIC)
          DELT4   = T4SETB - T4

        WFUELD    = LEDLAG(PDF,QDF,SPDERROR,WFUELDDIC)
        VSAPTD    = LEDLAG(PDF,QDF,DELT4,VSAPTDIC)


        WFUEL2  = K(1)*SPDERROR + K(2)*WFUELI + K(3)*WFUELD +...
                  K(4)*DELT4 + K(5)*T4INTG + K(6)*VSAPTD + WFUELEQ

        IF(WFUEL2.LT.WFLLIM)THEN
                  WFUEL2=WFLLIM
                  SPDERRORI=SPDI
        ELSEIF(WFUEL2.GT.WFULIM)THEN
                  WFUEL2=WFULIM
                  SPDERRORI=-SPDI
        ELSE
                  SPDERRORI=SPDERROR
        ENDIF
        IF(MODE.EQ.2)SPDERRORI=SPDI        !turn off spd integrator
                                           !in mode 2

        WFUELI = INTEG(SPDERRORI,WFUELIIC)
        WFUEL  = REALPL(TAUWF,WFUEL2,WFUELIC)
```

```
_ZIPPY$DKA300:[93.PRIGGE]CONTROL.CSL;2

      VSAPT2   = KV(1)*SPDERROR + KV(2)*WFUELI + KV(3)*WFUELD +...
                 KV(4)*DELT4 + KV(5)*T4INTG + KV(6)*VSAPTD + VSAPTEQ

      IF(VSAPT2.LT.VSAPTLL)THEN            !turn off integration at
                 VSAPT2=VSAPTLL            !upper and lower limits
                 DELT4I=VANEI
      ELSEIF(VSAPT2.GT.VSAPTUL)THEN
                 VSAPT2=VSAPTUL
                 DELT4I=-VANEI
      ELSE
                 DELT4I=DELT4
      ENDIF
      IF(MODE.EQ.3)DELT4I=VANEI
        !turn off t4 integ when in mode 3

      T4INTG   = INTEG(DELT4I, 0.0)
      IF(VSAPT2.GT.VSAPTUL)VSAPT2=VSAPTUL
      IF(VSAPT2.LT.VSAPTLL)VSAPT2=VSAPTLL
      IF(MODE.EQ.4.OR.MODE.EQ.5)THEN
        VSAPT2=VSAPTLL
        TAUVS2=TAUVS1
      ELSE
        TAUVS2=TAUVS
      ENDIF
      VSAPT    = REALPL(TAUVS2,VSAPT2,VSAPTIC)

      !EXIT MODE LOGIC
      GOTO(EG1,EG2,EG3,EG4,EG5),MODE
      GOTO EG1

EG2.. IF(ABS(DELT4).LT.TESTOVER.AND.ABS(VSAPTD).LT.TESTOVER...
        .AND.SPDERROR.LT.0.)THEN
        NEWMODE=.TRUE.
        T4SET=T4SETV
        XNPTSETB=XNPTSET
      ENDIF
        GOTO EG1

EG3.. IF(ABS(SPDERROR).LT.OFFSPD.AND.WFUELD.LT.OFFSPD...
        .OR.T4.GT.T4TRIPM2)NEWMODE=.TRUE.
      GOTO EG1

EG4.. IF(ABS(SPDERROR).LT.OFFSPD.AND.ABS(WFUELD).LT.OFFSPD1...
        .AND.ABS(VSAPT-VSAPTLL).LT.0.01)THEN
        MODE=5
        MODE4=.FALSE.
      ENDIF
      IF(T4.GE.T4TRIPM4)THEN
        NEWMODE=.TRUE.
        MODE4=.TRUE.
      ENDIF
      GOTO EG1

EG5.. IF(SPDERROR.GT.OFFSPD.OR.T4.GE.T4TRIPM2)THEN
        NEWMODE=.TRUE.         !GO TO MODE 2 DURING THE ONLOAD
      ENDIF

EG1.. CONTINUE !EXIT MODE 1 FROM THE ENTRANCE LOGIC
```