

AD-A956 420



lum

DTIC

(1)

FUZZY SETS AND OTHER METHODS  
FOR TELLING A COMPUTER  
HOW TO DECIDE

AHMET MITHAT BUHARALI

FEBRUARY 1982

DTIC  
ELECTE  
JUN 18 1993  
S A D

Contract MDA-903-30-C-0109  
System Sciences Division  
Defense Advanced Research Projects Agency  
1400 Wilson Boulevard  
Arlington, VA 22209

Approved for public release;  
distribution unlimited. Reproduction  
in whole or in part is permitted for  
any purpose of the United States  
Government

98 8 17 05 3

93-13768



REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Fuzzy Sets and Other Methods for Telling a Computer How to Decide		5. TYPE OF REPORT & PERIOD COVERED Technical Report Nov. 1980 - Feb. 1982
7. AUTHOR(s) Ahmet Mithat Buharali		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Massachusetts Institute of Technology 77 Massachusetts Avenue Cambridge, MA 02139		8. CONTRACT OR GRANT NUMBER(s) MDA-903-80-C-0109
11. CONTROLLING OFFICE NAME AND ADDRESS System Sciences Division Defense Advanced Research Projects Agency 1400 Wilson Blvd., Arlington, VA 22209		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE February 1982
		13. NUMBER OF PAGES 193
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)  Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES  None		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) man-computer interface                      search fuzzy sets                                        decision aids utility theory                                    linguistic controller controls    multi-attribute data base		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)  This report describes the use of two different methods for telling a computer how to make decisions in a dynamic task. Fuzzy set theory and an "interpolation - between - corners" method are explained. An in-trail vehicles		

following task is used for purposes of comparing the methods experimentally. Simulations, inputs and displays are generated by means of a computer.

The effectiveness of the methods is measured by means of time history graphs of the in-trail following simulation. The latter are compared for various modes of controlling the simulation. The two methods are compared with each other in terms of programming and software considerations as well as performance and applicability to the system.

Recommendations on how to improve both the human-computer interface and the evaluation algorithms are discussed. A section describing previous work on fuzzy sets is included. A program which was designed for the multi-attribute database search is described and future development possibilities of these techniques are given.

ACKNOWLEDGEMENTS

This research was performed under the sponsorship of the System Sciences Division , Defence Advanced Research Projects Agency, Arlington, Va. Drs. Clinton Kelly and Judith Daly were contract monitors. Their support is gratefully acknowledged.

I wish to thank Professor Thomas B. Sheridan for his exceptional patience, understanding and advise throughout the long period in which this thesis was written. I am grateful for his ideas and guidance in starting up this work.

DTIC QUALITY INSPECTED 3

Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DHC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution	
Availability Codes	
Dist	Avail and/or Special
A-1	

UNANNOUNCED

## TABLE OF CONTENTS

	ABSTRACT.....	2
	AKNOWLEDGEMENTS.....	3
1:	INTRODUCTION AND PROBLEM STATEMENT .....	7
1.1 -	INTRODUCTION.....	7
1.2 -	PROBLEM STATEMENT.....	11
2:	FUZZY SETS.....	13
2.1 -	FUZZY SET THEORY.....	13
2.2 -	LITERATURE ON FUZZY SETS.....	31
3:	- THE INTERPOLATION-BETWEEN-CORNERS METHOD.....	35
4:	EXPERIMENTAL TASK.....	44
4.1 -	DYNAMIC CONTROL OF A CAR FOLLOWING TWO CARS IN TRAIL.....	45
4.2 -	APPLICATION OF FUZZY SETS.....	47
4.3 -	APPLICATION OF THE INTERPOLATION- BETWEEN-CORNERS METHOD.....	55
5:	EQUIPMENT.....	57
5.1 -	HARDWARE.....	57
5.2 -	SOFTWARE.....	58
6:	APPLICATION OF THE FUZZY STATE MATRIX.....	65
6.1 -	OBTAINING CONTROL VALUES FROM THE FUZZY STATE MATRIX .....	65
6.2 -	DETERMINATION OF THE CONTINUOUS CONTROL	

	FUNCTION FROM THE DISCRETE STATE ACTION	
	MATRIX.....	67
7:	EXPERIMENTS.....	72
7.1 -	GENERAL RULES FOR THE TESTS.....	72
7.2 -	TESTS WITH FUZZY INPUT RULES.....	80
7.3 -	TESTS WITH INTERPOLATION-	
	BETWEEN-CORNERS.....	85
7.4 -	OTHER MODES OF INPUT.....	88
8:	IMPLICATION OF THE METHODS FOR DATABASE	
	SEARCHES.....	90
8.1 -	MULTI-ATTRIBUTE DATABASE SEARCH.....	90
8.2 -	FUZZY SET ALGORITHM.....	91
8.3 -	INTERPOLATION-BETWEEN-CORNERS.....	94
9:	COMPARISON OF FUZZY SETS WITH THE	
	INTERPOLATION BETWEEN CORNERS	
	METHOD.....	98
9.1 -	SOFTWARE CONSIDERATIONS: FUZZY SETS.....	98
9.2 -	SOFTWARE CONSIDERATIONS: INTERPOLATION-	
	BETWEEN CORNERS.....	99
9.3 -	APPLICATION CONSIDERATIONS.....	104
9.4 -	HUMAN INTERFACE CONSIDERATIONS.....	108
10:	RECOMMENDATIONS.....	110
10.1 -	IMPROVED INPUT ALGORITHMS.....	110
10.2 -	ON-LINE MONITORING OF THE OPERATOR'S	
	DECISIONS DURING CONTROL.....	114
10.3 -	MONITORING FOR INTELLIGENT SEARCHES.....	117

APPENDIX  
REFERENCES

## Chapter 1

## INTRODUCTION AND PROBLEM STATEMENT

## 1.1 - INTRODUCTION

Most systems have distinct input and output attributes which are used by the computer to carry out the necessary calculations. These attributes may be temperature, distance, flow or similar variables which are measurable. Usually, the computer uses predefined functional relations between these attributes, often including or based upon a mathematical model of the controlled process, to perform its control function.

Control is straightforward when the mathematical model of the process can be formulated clearly. But if there is significant deviation between the model and the actual process, or if a model cannot be defined mathematically, computer control can become very difficult. Unfortunately, the latter situation is very common.

One straightforward method to handle such a situation is to use human operators. The human operator has the ability to use logic, intuition and experience in order to make decisions in the execution of a task and is usually more or less correct in his decisions if he is a competent and experienced operator.

For many complex systems (vehicles or plants), the process model and parameters are very hard to determine. Even if modelling is possible, it usually ends up to be

extremely nonlinear and complex. One method to automate such complex systems would be to make use of decisions made by an experienced human operator in order to tell a computer how to operate a system. The computer algorithm used in the control of such complex systems would include a decision model of the human operator. This is in lieu of a mathematical approximation of the process and an optimal controller (which is not available for nonlinear processes). Instead the computer would generate control rules based upon the recommendations of the human operator.

Decisions which one would like to be made by computers, but which have to be made by humans because of the reasons given above, are encountered very frequently in many operations. Such tasks do not necessarily consist of controlling a clearly identifiable device. Examples can be given in business, military or other strategic applications.

Usually the important attributes in the above type of situation cannot be formulated mathematically. The decisions are based upon concepts which only exist in the human mind and which can only be described in "fuzzy" terms.

If the problem requires decisions where the different attributes have uncertain characteristics there is very little chance that a computer can aid the operator in the decisions. Related work has been done in our laboratory where the computer is used as an aid ( (1), (2), (3) ).

The major difficulty in telling a computer how to

decide, when controlling systems with uncertain properties, lies in the basic difference between a computer and the human. A computer uses distinct logic. Something is either true, or it is false. The computer executes the orders given to it without interpreting the meaning or without thinking about consequences or uncertainties. The human operator does not use distinct values and terms. We assert that he uses terms and statements of various degrees of fuzziness in order to justify his decisions. When he carries out a task, he knows what he is doing, knows about the consequences or can guess what the future will be only imperfectly. We can say that he must adjust his response according to his intuition.

There are many uncertainties inherent in the human operator for which no reasoning can be given in computer terms. Mathematicians and behavioural scientists have developed different mathematical approaches in order to model human behaviour and decisionmaking. Utility theory is one of the approaches. Another is a method based on fuzzy sets which allows mathematical operations on sets described in linguistic terms. Both have been under development for the last fifteen years.

Implementation of both of these methods requires extensive man - computer and computer - machine interaction. Color graphic systems and interactive manual control devices have improved the effectiveness of human computer

communication. The present research effort investigates the problems occurring in the application of fuzzy set and utility theory to man - computer - machine interactive control.

## 1.2 - PROBLEM STATEMENT

One method to evaluate a multi-attribute situation within a task is to quantize the attributes at discrete levels and determine a matrix which has the same dimensions as the attributes of the situation. The values of the elements of this matrix represent the action to be taken under any particular combination of attributes and levels. This matrix is called the State Action Matrix. In some of the literatures it is referred as the Relation Matrix. The approach of this work is to find methods of determining the values of the elements of the state action matrix for different situations within a task.

The simplest method, assuming that the human operator knows what the correct solution is, would be to enter all individual values of the state action matrix directly. But since a state action matrix requires one individual dimension for each attribute, with many levels per attribute, it can grow up to sizes on the level of megabytes on actual tasks. Therefore, different methods have been developed in order to determine this matrix.

Two methods have been emphasized in this research. The first one is learning extreme points and then interpolating between those points to calculate the intermediate values (as in utility theory). The second method works by means of defining a smaller fuzzy state matrix which is mapped to the distinct state action matrix by means of membership

functions.

## Chapter 2

## FUZZY SETS

## 2.1 - FUZZY SET THEORY

The notion of fuzzy sets was first introduced by Zadeh in 1965. The need he saw was for some mathematical description of variables as they are used by humans, as compared to what might always be representable directly in mathematical distinct terms. This is the main idea which lies behind fuzzy sets. On many attributes used by human operators, the exact value is not significant or cannot be defined numerically. There might not be a need to use the exact numerical value because precision might be insignificant for the task. Sometimes the operator judges according to values given by linguistic statements to define an amount or value. Imprecise values like "very", "somehow" or "approximately" are used. Most of these statements associated with the attributes of a task are used as major factors affecting human decision making. For example in a task where a pilot has to make decisions about landing, the following might be his perceived guidelines for making the decision.

"If the visibility is low or very low and if the fuel reserves are high, do not land; otherwise, if the fuel reserves are very low, do the landing."

or if a person wants to decide on buying a car:

"If the car is fuel efficient and expensive, or if the car

is sport, and cheap, I like it a lot."

In these statements, the variables are used in linguistic terms and are fuzzy statements. The attributes are visibility and fuel reserves in one case, fuel efficiency, price and sportiness in the other.

The computer uses exact values in order to take appropriate unique actions. It takes a given numerical value, makes the additions and subtractions it is told to take and ends up with a decision. However the linguistic statements given above cannot in themselves be represented numerically. For example appearance or sportiness could be represented as a mathematical function of the wind friction factor, color, number of doors and height, but it is very unlikely that such a function is easy to determine, and even if it can be, there is a definite possibility that the function will be different for different persons. An attribute like "danger" might be defined exactly as a function of the necessary variables (e.g. speed, height, acceleration, presence of a certain object) if it could be. The factor of human intuition cannot be included in the function.

Fuzzy set theory is a tool which directly uses terms like those given above. Using this theory, the computer does not need to know what "dangerous" or "beautiful" means exactly in terms of known physical variables. The fuzzy variable combinations produce a fuzzy state matrix which is

related to the multi-attribute database by means of membership functions. The membership function defines the degree to which a physical value belongs to a fuzzy value of the same attribute.

Figure (2.1) shows the way fuzzy sets are used in dynamic decisionmaking in this report.

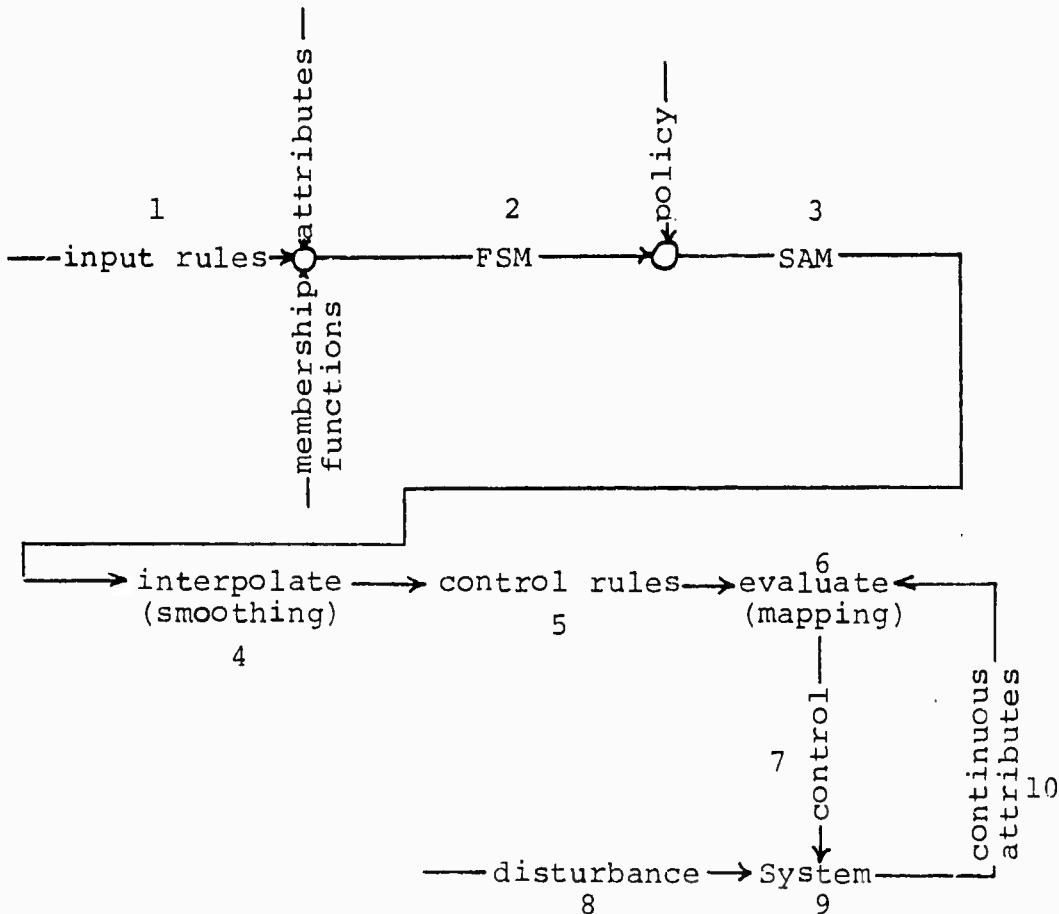


FIGURE 2.1  
Fuzzy input algorithm

Figure 2.1 explanations:

- 1 - Entering of the input rules using the fuzzy description of the attributes
- 2 - The attributes, input rules and membership functions produce the fuzzy state matrix (FSM) .
- 3 - According to the entered policy, The FSM is converted to the state action matrix (SAM).
- 4 - The state action matrix is fed into the interpolator which smooths it out.
- 5 - The control function is the output of the interpolator.
- 6 - The evaluator maps the control function to the attributes to produce the control.
- 7 - The control is fed to the system
- 8 - Disturbance as the leadcar control input is fed to the system.
- 9 - The system responds to the inputs
- 10 - The attributes, which are the output of the system, are fed into the evaluator.

The following is an example showing the application of fuzzy sets to a decisionmaking process. Two attributes have been selected to be important for a task in which a pilot has to make a decision about landing at an airport.

The fuzzy linguistic descriptions of the variables and their abbreviations are fuel reserve and visibility and are described with the following fuzzy descriptions.

TABLE (2.1):

fuel:	Very high	high	low	very low
	VH	H	L	VL
visib:	low	medium	high	
	L	M	H	

Fuzzy set theory requires discrete values of transformed input variables in order to end up with a discrete relation matrix which shows the relation between the input and output variables. Membership functions are usually continuous variables of their attributes but because of the reason given above, the assignment of the membership values will be done on the discretized attributes. We will assume the following discrete quantization of input variables for convenience.

TABLE (2.2):

Fuel reserve (gal) = 50, 100, 150, 200, 250

visib: (miles) = 1, 2, 3, 4, 5

The fuzzy membership function of the variables is given in tables (2.3) and (2.4) and figure (2.2)

TABLE (2.3):

Fuel reserves	VH	H	L	VL
50 gal =	0	0	4	10
100 gal =	0	5	10	5
150 gal =	5	10	2	0
200 gal =	10	5	0	0
250 gal =	10	2	0	0

TABLE (2.4):

visibility	L	M	H
1 mile=	10	3	0
2 mile=	5	10	0
3 mile=	3	10	3
4 mile=	1	10	5
5 mile=	0	3	10

In order to minimize the loss of information due to

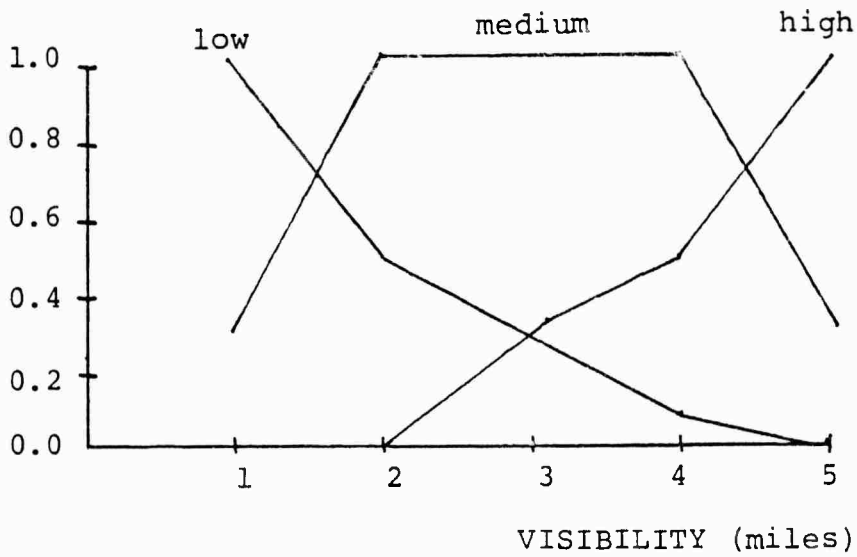
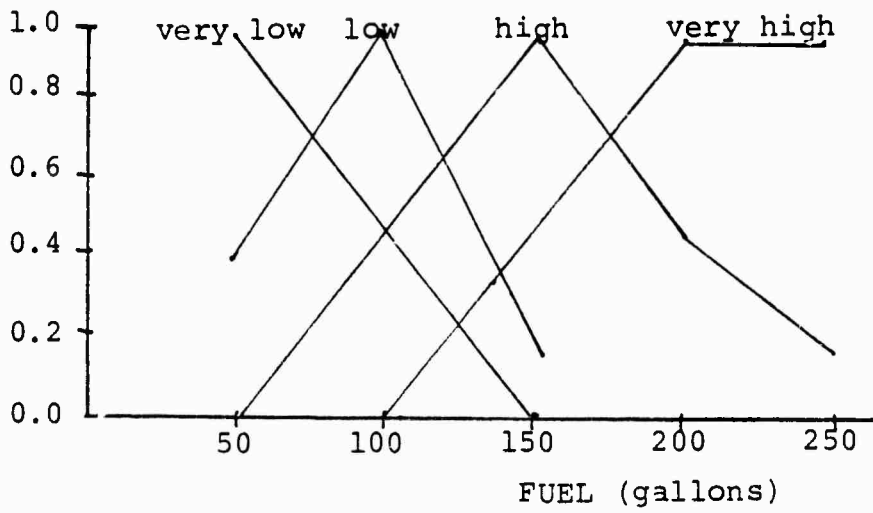


FIGURE 2.2 Graphical representation of membership functions

quantizing of variables, in this example quantization is done by using the values of the variables directly. It should be realized that on actual large databases the versatility of the individual values will be so large that such a procedure has to be done by grouping the variables into groups with similar values of the attributes which will end in no distinction between the different values within the same quantized group which in turn results in a considerable amount of loss in information.

The output variable has three values. They are "LAND", "WAIT", "PASS" There is no fuzzy membership assignment to them since we assume them to be linguistic but distinct variables. They could have been fuzzy variables but this would increase the amount of calculation necessary in this example. Since distinct (regular) sets are actually fuzzy sets which have a membership function of 100% for a single value and zero for any other value, they can be used anywhere in operations done with fuzzy sets. The introduction of a single fuzzy set within operations on regular sets results in a fuzzy output. This can be compared with the introduction of a single noise source which would result in noisy output on a multiple input single output system.

The conditions, which will be given by the operator, and the necessary calculations will show how the knowledge of the computer improves with additional information. The

result will be a (2,2,3) relation matrix which will show us the membership of any combination of the attributes to the three output variables. Initially, the computer has no knowledge about any operator preference so that the relation matrix is blank. We will invent linguistic criteria of the operator in order to fill the relation matrix.

The first instruction entered to the computer is as follows:

RULE 1: "IF FUEL RESERVE IS VERY LOW OR LOW AND VISIBILITY IS MEDIUM OR HIGH ; LAND"

The "OR" operation with fuzzy sets is the mathematical "MAX" operation while the "AND" operation with fuzzy sets is the mathematical "MIN" operation.(4) The computer must go through all the possible combinations of the attributes in the database in order to find out their relative truth.

The relation matrix consists of a (5,5) matrix with three elements each. Each of the elements shows the truth of each of the output variables for that corresponding input element. Table (2.5) shows the initially empty relation matrix.

TABLE (2.5):

visib.	5			4			3			2			1		
gal	L	W	P	L	W	P	L	W	P	L	W	P	L	W	P
50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
150	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
250	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table (2.5) is a three dimensional matrix which will display the memberships of each of the three output actions, L, W and P for all combinations of the attributes. All combinations of the above table have to be evaluated according to rule 1. For a 50 gal 1 mile input data point, the following operations are applied; based on the individual membership functions given in the tables(2.3) and (2.4).

$$\text{MAX}(\text{fuel=L, fuel=VL}) = \text{MAX}(4,10) = 10$$

which is the truth of fuel reserve to be "low or very low", in conjunction with 50 gallons. Likewise:

$$\text{MAX}(\text{visibility=M, visibility=H}) = \text{MAX}(3,0) = 3$$

which is the truth of visibility being "medium or high" in conjunction with 1 mile. The overall membership of the fuel reserve to be low or very low and the visibility to be medium or high is:

$$\text{MIN}( \text{MAX}(4,10) ; \text{MAX}(3,0) ) = \text{MIN}(10;3) = 3$$

in conjunction with 1 mile. In the same manner we can determine the result for the other points of the relation matrix.

for 14 gallons, 2 miles we have

$$\text{MIN}( \text{MAX}(4,10) ; \text{MAX}(10,0) ) = \text{MIN}(10;10) = 10$$

for 14 gallons, 3 miles we have

$$\text{MIN}( \text{MAX}(4,10) ; \text{MAX}(10,3) ) = \text{MIN}(10;10) = 10$$

When the input for all combinations of the attributes has been given, the relation matrix ends up with nonzero terms as it's first element which corresponds to the output value of "LAND". Table (2.6) shows the partial relation matrix of "LAND" and table (2.7) shows the relation matrix after the first input. It can be noted that there are still some zeroes in the last two rows. The reason for those zeroes is that 200 gallon and 250 gallon have zero

membership of having low fuel reserve and the "AND" operation with their memberships produces the zeros.

TABLE (2.6):

visib.	5	4	3	2	1
gal	L W P	L W P	L W P	L W P	L W P
50	10	10	10	10	3
100	10	10	10	10	3
150	2	2	2	2	2
200	0	0	0	0	0
250	0	0	0	0	0

TABLE (2.8):

visib.	5	4	3	2	1
gal	L W P	L W P	L W P	L W P	L W P
50	10 0 0	10 0 0	10 0 0	10 0 0	3 0 0
100	10 0 0	10 0 0	10 0 0	10 0 0	3 0 0
150	2 0 0	2 0 0	2 0 0	2 0 0	2 0 0
200	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0
250	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0

The next input condition is as follows:

RULE 2: "IF FUEL RESERVE IS HIGH OR VERY HIGH AND VISIBILITY IS LOW; PASS"

The same operations as done above are repeated to determine

what the membership function of passing is for each combination. The cumulative result is shown in tables (2.8) and (2.9).

TABLE (2.8):

visib.	5			4			3			2			1		
gal	L	W	P	L	W	P	L	W	P	L	W	P	L	W	P
50			0			0			0			0			0
100			0			0			3			5			5
150			0			0			3			5			10
200			0			0			3			5			10
250			0			0			3			5			10

TABLE (2.9)

visib.	5			4			3			2			1		
gal	L	W	P	L	W	P	L	W	P	L	W	P	L	W	P
50	10	0	0	10	0	0	10	0	0	10	0	0	3	0	0
100	10	0	0	10	0	0	10	0	3	10	0	5	3	0	5
150	2	0	0	2	0	0	2	0	3	2	0	5	2	0	10
200	0	0	0	0	0	0	0	0	3	0	0	5	0	0	10
250	0	0	0	0	0	0	0	0	3	0	0	5	0	0	10

The next condition is:

RULE 3: "IF VISIBILITY IS MEDIUM; WAIT".

Note that no condition for fuel reserve is given. In this

example, as well as throughout the rest of this paper, if an attribute is not referenced, it means that all combinations of that attribute are "OR"ed. It should be also noted that there is no "AND" statement within the same attribute since, for example, visibility can be medium OR low, but not medium AND low

Table (2.10) shows the partial relation matrix after the third input condition and table(2.11) the full relation matrix.

TABLE (2.10):

visib.	5	4	3	2	1
gal	L W P	L W P	L W P	L W P	L W P
50	3	10	10	10	3
100	3	10	10	10	3
150	3	10	10	10	3
200	3	10	10	10	3
250	3	10	10	10	3

TABLE(2.11)

visib.	5			4			3			2			1		
qal	L	W	P	L	W	P	L	W	P	L	W	P	L	W	P
50	10	3	0	10	10	0	10	10	0	10	10	0	3	3	0
100	10	3	0	10	10	0	10	10	3	10	10	5	3	3	5
150	2	3	0	2	10	0	2	10	3	2	10	5	2	3	10
200	0	3	0	0	10	0	0	10	3	0	10	5	0	3	10
250	0	3	0	0	10	0	0	10	3	0	10	5	0	3	10

Now all three input variables (LAND, WAIT, PASS) have been referenced once. It is natural for humans that there is more than one combination of attributes to produce a pass. An example is a condition like:

RULE 4: "IF FUEL RESERVE IS LOW OR HIGH AND VISIBILITY IS HIGH; LAND"

This means that the operator is ready to land if the fuel reserves are medium and the visibility is high. This condition by itself leads to the following table (2.12):

TABLE (2.12):

visib.	5	4	3	2	1
gal	L W P	L W P	L W P	L W P	L W P
50	4	4	3	1	0
100	10	5	3	1	0
150	10	5	3	1	0
200	5	5	3	1	0
250	2	2	2	1	0

Since this table has to be merged with the actual relation matrix, the following reasoning has to be made: Conditions (1) and (4) both lead to the output "LAND". So it can be considered that the overall condition is as follows:

"IF RULE(1) OR RULE(4); LAND"

This shows that if several different conditions lead to the same output, the different results have to be "OR"ed (i.e. their maximum has to be used). The result is table (2.13) which is the maximums of table (2.8) and table (2.12).

TABLE (2.13):

visib.	5	4	3	2	1
gal	L W P	L W P	L W P	L W P	L W P
50	10	10	10	10	3
100	10	10	10	10	5
150	10	5	3	5	10
200	5	5	3	5	10
250	2	2	3	5	10

TABLE (2.14): FINAL RESULT

visib.	5	4	3	2	1
gal	L W P	L W P	L W P	L W P	L W P
50	10 3 0	10 10 0	10 10 0	10 10 0	3 3 0
100	10 3 0	10 10 0	10 10 3	10 10 5	3 3 5
150	10 3 0	5 10 0	3 10 3	5 10 5	2 3 10
200	5 3 0	5 10 0	3 10 3	5 10 5	0 3 10
250	2 3 0	2 10 0	3 10 3	5 10 5	0 3 10

If all the inputs are considered to be complete, table (2.14) is the final fuzzy relation matrix which shows the reaction to take with any combination of attributes provided they are quantized according to our inputs.

The corner of 50 gallons and 5 miles of visibility says that there is a 100% membership to land, 30% membership to

wait and 0% membership to pass. On the other hand, the combination of 250 gallons and 5 miles of visibility has a maximum of only 30% to wait. The inputs were not explicit enough to define a definite action for that point. Another type of uncertainty can be seen in other points, for example both visibility 2 and 100 gallons of fuel reserve say that there is a 100% membership for both landing and waiting. If the rules are traced back, it can be seen that condition 3 says to wait if the visibility is medium but condition 1 describes combinations with the visibility being medium where it can be landed. This table includes the common properties of a fuzzy state matrix. Since the results are not unique, it can be nonfuzzified in several different ways. The action to be taken under these circumstances depends very much on the specific problem. New rules can be introduced to override the old ones, or the computer can be asked to use either extreme or the average of the contradicting outputs. If there is a sensitive situation, the computer is usually instructed to ask the operator what to do under those circumstances.

Fuzzy sets are not sophisticated enough to make interpolations or extrapolations between different arguments. They accept whatever is given and use that as the only information. The more widespread the membership functions are, the more effect will be distributed within the database. Naturally, if the membership of any variable

is described for only a few quantized steps, the information given will be much more localized. This can be interpreted as follows:

Fuzzy sets allow us to determine a response for a region of an attribute space where the membership functions have values. The fuzzier the attribute, the more the membership function will be spread out, but in return, because of the high level of fuzziness, less will be known about what the variable implies.

## 2.2 - LITERATURE ON FUZZY SETS

Since the first introduction of fuzzy sets by Zadeh in 1965 various implementations on a variety of applications has been done. Various publications dealing with fuzzy sets can be found. A good discussion about fuzzy sets written in 1976 can be found in ref.(5).

Freeling(6) did some research comparing decisionmaking analysis by means of fuzzy sets. His results were not very satisfactory. He pointed out some computational anomalies which severely limit the analysis. He suggests that, for individual decisionmaking, fuzzy sets should be viewed as an automatic sensitivity analysis, but that fuzzy sets may be useful with another interpretation for group decisionmaking. The conceptual difficulties in the use of fuzzy sets which require a skilled analyst, seem to make this method less attractive for an everyday commercial application.

In a paper entitled Interactive Multicriteria Optimization, Baptisella and Ollero (7) combine large-scale dynamic optimization with techniques based on fuzzy set theory in order to consider inaccuracies inherent in the decisionmakers judgements, but without a significant increase in the computational effort. They used different methodologies according to their assumptions on the decisionmakers estimation capabilities. They applied the methodologies to a problem of optimal scheduling in a hydrothermal power system with water resources constraints.

In their conclusions they state that fuzzy approaches provide a useful way to deal with the above mentioned inaccuracies.

Saridis and Stephanou(8) used fuzzy sets in their paper called: "A Hierarchical Approach to the Control of a Prosthetic Arm". They used fuzzy sets in their learning schemes of the control of the arm.

Cheng and McInnis(9) did develop an algorithm for multi attribute, multiple alternative decisionmaking problems based on fuzzy set theory. The computer program handles problems involving a large number of alternatives and imprecise relationships between input information and possible alternatives. An application of the algorithm to the medical diagnostic problem of ranking possible diseases on the basis of chemical laboratory test data is given. They conclude that the method they used might be particularly useful for handling a large number of alternatives.

R.R. Yaqer (10) published a paper on: "A logical on-line bibliographic searcher: An application of fuzzy sets". Tribus (11) compared this fuzzy methodology with Bayesian methods and concluded that there was no significant difference between the two methods.

C. V. Negoita(12) did some research on fuzziness in information retrieval and concluded that the concept of fuzzy sets has theoretical and practical significance in the

information retrieval field.

Mamdani and his collaborators have done significant work in the field of applications of fuzzy sets to control(13).

The variables measured are converted into fuzzy variables by quantizing them and using a look-up table with discrete membership functions. The variable and its rate of change are the only input variables. The conclusion was that fuzzy sets might be useful in the control of complex plants where nonlinear effects are predominant. Fuzzy control algorithms are being used in some processes which range from a warm water process(15) sinter plant heat exchanger and other similar processors. Mamdani has given a complete survey of work in this area(14) where he also discusses the question of stability of fuzzy control systems.

The work of "Application of a Fuzzy Controller in a Warm Water Plant" by Kickert and Lemke (15) is also very closely related to this area. They used continuous membership functions and converted the variables by calculating the values at finite quantized intervals. They used three different assumptions in how the operator used the error and rate of error information. They compared their results with a classical PI controller and achieved comparable results. According to the performance criterion chosen, they achieved better results with the fuzzy controller than with the PI controller.

Some analytical work on analysing the relation matrix of the fuzzy control algorithms has been done by Tong(16). He used the relation matrix in order to examine the structure of fuzzy control algorithms. In his paper examples are given on how the structure of the relation matrix, which is the product of the rules and which has been determined by the min-max operations, can give hints about the control algorithms. Tong uses numerical examples in order to show his points and does not go into very detailed or advanced mathematics.

The paper by P.J. MacVicar-Whelan called: "Fuzzy Sets for Man-Machine Interaction"(17) gives some very good ideas on how to better understand the fuzzy behaviour of a human operator in a man-machine system. The paper points out some basic problems which have also been encountered in this thesis and suggests some solutions which have been partially used in this thesis and which will be more deeply investigated as a continuation of this work.

References (18) and (19) are in two issues which are completely devoted to fuzzy sets and their applications. The author specially recommends (18) for a general understanding on fuzzy sets and their applications. Gaines discusses fuzzy sets with all its aspects including the moral standpoint of "thinking machines" which might be one of the future consequences of fuzzy set theory.

## Chapter 3

## THE INTERPOLATION-BETWEEN-CORNERS METHOD

There are other methods which can be used in dealing with linguistic information. Some investigators take tasks that can be executed with fuzzy sets but solve them in a nonfuzzy way. A method closely related to utility theory was developed in 1963 by Yntema and coworkers(21). Yntema notes that this method may not satisfy all the formal requirements of utility theory.

Yntema's approach is based on the following monotonicity assumption: If the human judgement of worth for one multi attribute state is  $r_1$ , and for another state is  $r_2$ , then his judgement of worth for any state between those two extremes will be between  $r_1$  and  $r_2$ .

Figure (3.1) shows how the interpolation between corners method is used for dynamic decisionmaking in this report.

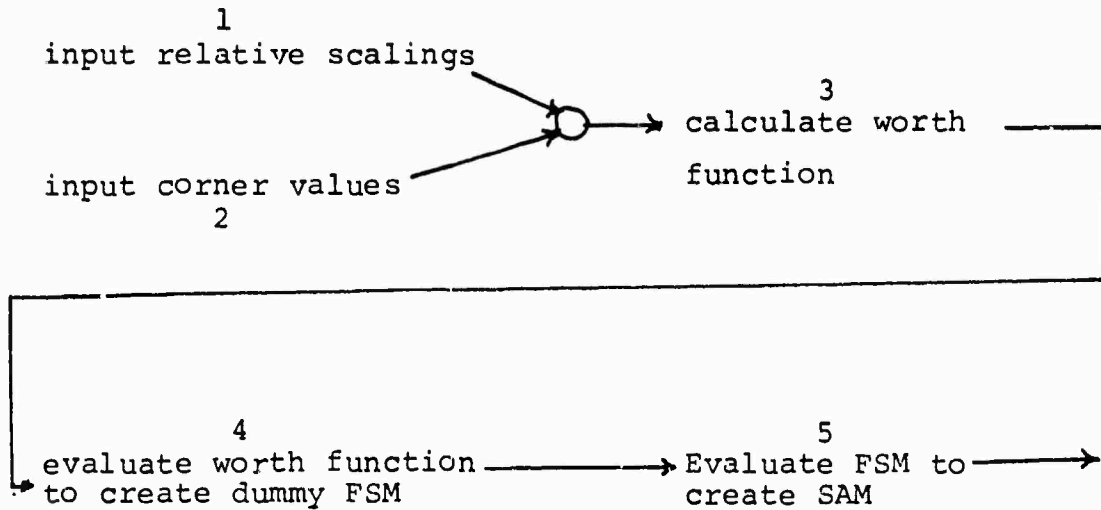


FIGURE 3.1  
Interpolation between corners method

The Fuzzy State Matrix FSM is created to make the procedure compatible with the fuzzy input algorithm. The output of node 5 is fed into node 4 of figure (2.1)

The best way to explain Yntema's algorithm is to use an example.

Example (2):

Let us assume that the same pilot of example (1) (which actually was used by Yntema in his experiments) is asked to enter worth judgements using the interpolation between corners method. The first step would be to ask him to scale the individual attributes according to their relative utilities, other attributes being constant. The first attribute is the fuel reserves. The subject is told to give the most favorable fuel reserve the utility rating of 1.0 and to the  $c$  which is least favorable a rating of 0. In order to explain this further, the instructions given to the test subject are given below:

"Assume that later you will have to enter judgements involving 50 gallons and 250 gallons. The following table shows intermediate values of 100, 150, 200 gallons. Knowing that your judgements involving these intermittent values will be in between your judgements involving the extremes, describe, on the following scale, to what extent you would like these values to be relatively affected by your corner judgements" Let us assume that the test subject enters the following rating:

TABLE (3.1)

fuel reserve

gallons	rating
50	0.0
100	0.1
150	0.5
200	0.7
250	1.0

Even though the numerical difference between 50 gallons and 100 gallons is the same as for 100 gallons and 150 gallons, i.e 50 gallons, the utility difference between the first two is only 0.1 whereas the utility difference between 100 and 150 gallons is 0.4. This means that for the pilot, the difference between 100 gallons and 150 gallons is much more important than the difference between 50 gallons and 100 gallons. It might be that he needs at least 12 gallons to make a successful landing and therefore does not care much if he has 50 or 100 gallons in his tank.

The subject is then asked to enter the utility of the second variable the same way the first variable was entered. Again assume:

TABLE (3.2)

Visibility	rating
5 miles	1.0
4 miles	0.7
3 miles	0.5
2 miles	0.3
1 mile	0.0

The corner points of the rectangular attribute space are the following:

TABLE (3.3)

Corner 1: 5 miles, 50 gallons  
 Corner 2: 5 miles, 250 gallons  
 Corner 3: 1 mile, 50 gallons  
 Corner 4: 1 mile, 250 gallons

If there were three attributes, there would be eight corners, and so on. These are the combinations of the attributes which have relative ratings of 0 and 1 on the individual attribute scales. Now the worths of these corner points are entered to take into account the interrelation between the attributes and their relative weights. The worth in this case is described as the tendency of the pilot to land. In other terms, if the worth is 0 the pilots

wants definitely to land, if it is 0.5, he wants to wait and if it is 0.0, the pilot wants to pass. This time there is only one transformed input variable (i.e. a scalar function of the attribute values) which is directly proportional to the probability of the pilot to do a landing.

It is reasonable in this case to assume that the corner consisting of the attributes with the highest utility will have the highest worth and the corner with the lowest utilities will have the lowest worth. But this is not true in a general case which might be a control problem or the classification of some objects. Let us assume that the worth judgements for the pilot are as follows:

TABLE (3.4)

corner 1:	5 miles,	50 gallons	= 0.5
corner 2:	5 miles,	250 gallons	= 1.0
corner 3:	1 mile,	50 gallons	= 0.0
corner 4:	1 mile,	250 gallons	= 0.7

It is assumed that the overall worth function has to connect all these corner points and should be a surface in between. Since there are four corner points, at least four parameters are needed to find a surface which connects them. The worth function  $w(x,y)$  is in the very general form of:

$$\text{Eq. (3.1)} \quad w = A + Bx + Cy + Dxy + Ex^2 + Fy^2 + Gx^2y + Hxy^2 + Kx^2y^2 + \dots$$

which is the Taylor expansion of a two dimensional function. In order to keep the function as simple as possible the minimum amount of parameters, namely four, is used and all terms which have higher order than 2 are neglected. This ends up with:

$$\text{Eq. (3.2)} \quad w = A + Bx + Cy + Dxy$$

This function, called a regulus, is linear in the sense that if one of the attributes is constant, the worth of the other attribute varies linearly. For example:

$$\text{Eq. (3.3)} \quad \text{at } y=y_0$$

$$w = (A + Cy_0) + (B + Dy_0) * x$$

The four parameters are easily determined since they have to pass through the corner points:

$$\text{Eq. (3.4)}$$

$$A + B*1 + C*0 + D*1*0 = 0.5$$

$$A + B*1 + C*1 + D*1*1 = 1.0$$

$$A + B*0 + C*0 + D*0*0 = 0.0$$

$$A + B*0 + C*1 + D*0*1 = 0.7$$

The solution of these equations gives us the worth function

which tells how much the pilot would have the tendency to land:

$$\text{Eq. (3.5)} \quad w = 0.5x + 0.7y - 0.2xy$$

The value of the parameters of this function gives us some information about the operator's criteria. Since  $B=0.5$  and  $C=0.7$  we can say that he is more sensitive to changes in the rating of fuel reserve than for changes in visibility.  $D=-0.2$  indicates that there is interdependency between the variables.

The worth of any combination of the attributes can be calculated by evaluating the worth function at any combination of the given ratings. It is important to understand what  $x$  and  $y$  in this example mean. They are the utilities of the individual attributes entered in the first step and range between 0 and 1. Therefore, to evaluate for example the worth of 150 gallons and 5 mile visibility, we have to look up from tables (3.1) and (3.2) the individual utilities  $x$  and  $y$ . We find them to be:

$$x(150 \text{ gal}) = .5 \quad y(5 \text{ miles}) = 1.0$$

Then, the worth function can be evaluated at these  $x$  and  $y$  values.

$$w = 0.5 * 0.5 + 0.7 * 1 - 0.2 * 1 * 0.5 = .85$$

If an attribute point is encountered which is not directly given in the preceding tables, it has to be assumed to vary linearly between two adjacent points. Assuming that the worth of a 1.5 mile 150 gallons situation has to be evaluated, and since the utility rating ( $y$ ) of \$1.5 is not given, the following calculation has to be done:

$$y(1.5) = y(1) + (y(2) - y(1)) * (1.5 - 1) / (2 - 1)$$

$$y(1.5) = 0 + 0.3 * 0.5 = 0.15$$

This can be seen more easily from table(3.2) that the pilot entered the utility of 1 mile as 0 and the utility of 2 miles as 0.3. Therefore the utility of 1.5 miles must be around 0.15. Therefore, the worth is:

$$x(150) = 0.4 \quad y(1.5) = 0.15$$

$$w = 0.5 * 0.4 + 0.7 * 0.15 - 0.2 * 0.4 * 0.15 = 0.29$$

If this interpolation between the individual attributes proves to introduce errors, the solution is to go back to step 1 and ask the human to enter more information to the individual utilities scale. The more entries there are in tables (3.1) and (3.2), the more accurate the results will be. Notice that as long as the corner entries of table (3.3) are not changed, the worth function has not to be recalculated.

## Chapter 4

## EXPERIMENTAL TASK

We assumed the computer accepts rules from the human operator and must execute control according to those rules. We assumed a task which is of a continuous dynamic nature and where data could be sampled for future analysis. Complications in which the system characteristics change with time and nonlinearities are useful properties to test the power of the methods. We assumed that the task can be executed by a human operator for comparison reasons. The in-trail following of cars fulfills these assumptions and was chosen to be the experimental task in this research.

#### 4.1 - DYNAMIC CONTROL OF A CAR FOLLOWING TWO CARS IN TRAIL

The basic in-trail following task performed on a simulator with computer graphic display was keep a suitable distance from a car which follows another car. The following convention was used in order to differentiate between the three cars involved in the simulation. "Leadcar" is the foremost car which has a random disturbance input. It was followed by "Midcar" which had a PD (proportional plus derivative) controller and kept a safe distance from Leadcar. Midcar was followed by "Controlcar" which was the vehicle which the operator had to control using a joystick as he observed Leadcar and Midcar in perspective on the computer display analogous to observing through the windshield.

The following attributes were chosen to be significant in this control task:

Distance error: There should be hardly any doubt that the distance error is the most significant variable in a trailing task.

Brake lights: In order to follow as closely as possible without causing an accident the brake lights of Leadcar proved to be a very important hint in what Midcar was going to do.

Velocity error: In order to have a stable damped system with reasonable damped response, feedback on the velocity error was a very important variable.

Speed of Controlcar: Since the characteristics of the system changed with changing speed and more headway was required with increasing speed, the control proved to be a variable of the amount of speed with which Controlcar was travelling.

#### 4.2 - APPLICATION OF FUZZY SETS

As mentioned earlier, the attributes of the control task were:

- 1) Distance error between midcar and controlcar
- 2) Brake lights of leadcar
- 3) Velocity error between midcar and controlcar
- 4) Speed of controlcar

It can be seen that attribute 3 is actually the rate of change of attribute 1. This attribute set included are not only the error and the rate of change of the error, as in most of the previous fuzzy control tasks, but we also had two additional attributes of the system state.

The attributes were quantized into the following regions:

TABLE (4.1a)

Attribute 1: SEPARATION

region 1= 0 - 16  
 region 2= 16 - 19  
 region 3= 19 - 21  
 region 4= 21 - 23  
 region 5= 23 - 25  
 region 6= 25 - 28  
 region 7= 28 - infinity

TABLE (4.1b)

Attribute 2: brake lights

region 1= Brake lights on ( - power)  
 region 2= Brake lights off( + power)

TABLE (4.1c)

Attribute 3: velocity error

region 1= -inf. - -6  
 region 2= -6 - 3  
 region 3= -3 - -1  
 region 4= -1 - 1  
 region 5= 1 - 3  
 region 6= 3 - 6  
 region 7= 6 - infinity

TABLE (4.1d)

Attribute 4: speed

region 1= 0-20

region 2= 20-40

region 3= 40-70

region 4= 70-90

region 5= 90- infinity

These quantized variables were mapped into the fuzzy (linguistic) attribute space with two different sets of membership functions:

Table (4.2a)

SET 1:

Attribute 1: separation (numbers in cells are degrees of membership)

	attribute region						
linguistic variable	1	2	3	4	5	6	7
very close	10	8	2	1	0	0	0
close	2	8	10	8	5	2	0
normal	0	2	6	10	6	2	0
far	0	0	1	5	10	5	1
very far	0	0	0	0	2	5	10

TABLE (4.2a)

Attribute 2: brake lights

	1	2
Brake lights on	10	0
Brake lights off	0	10

TABLE (4.2c)

Attribute 3: velocity error

	attribute region						
linguistic variable	1	2	3	<u>4</u>	5	6	7
closing fast	10	8	2	1	0	0	0
closing	2	8	10	8	5	2	0
same speed	0	2	6	10	6	2	0
opening	0	0	1	5	10	5	1
opening fast	0	0	0	0	2	5	10

TABLE (4.2d)

Attribute 4: speed

	attribute region				
linguistic variable	1	2	3	4	5
slow	10	2	0	0	0
normal	1	4	10	4	0
fast	0	0	2	7	10

TABLE (4.3a)

SET 2:

Attribute 1: separation (numbers in cells are degrees of membership)

	attribute region						
linguistic variable	1	2	3	4	5	6	7
close	10	9	7	3	1	0	0
normal	0	3	7	10	7	3	0
far	0	0	1	3	7	9	10

TABLE (4.3b)

Attribute 2: brake lights

	1	2
Brake lights on	10	0
Brake lights off	0	10

TABLE (4.3c)

Attribute 3: velocity error

	attribute region						
linguistic variable	1	2	3	4	5	6	7
closing	10	9	7	3	1	0	0
same speed	0	1	3	10	3	1	0
opening	0	0	1	3	7	9	10

TABLE (4.3d)

Attribute 4: speed

attribute region					
variable	1	2	3	4	5
slow	10	9	8	2	0
fast	0	2	8	9	10

The output which was not fuzzy consisted of control input values ranging from full brake (-10) up to full power (+5).

Table (4.4) : Output values

$$C = [-10 \quad -5 \quad -2 \quad 0 \quad 2 \quad 5]$$

## Example (3)

Even though a general algorithm for the calculation of fuzzy rules was given earlier, specific use with this task seems useful to explain here.

Let us assume that one of the conditions given is:

"IF DISTANCE ERROR IS SMALL OR VERY SMALL AND BRAKE LIGHTS ARE ON THEN INPUT=-10".

The computer interprets all the ungiven conditions for other attributes as OR's:

"IF DISTANCE ERROR IS SMALL OR VERY SMALL AND BRAKE LIGHTS ARE ON AND VELOCITY ERROR IS CLOSING FAST OR CLOSING OR NORMAL OR OPENING OR OPENING FAST AND SPEED IS SLOW OR NORMAL OR FAST THEN INPUT =-10"

The attribute space in the computer consisted of a (7,2,7,5) quantized array and a (5,2,5,3) fuzzy array. (Fuzzy set 1) The maximums of all the membership values of distance error being close or distance error being very close are taken to produce a (1,7) vector which are the maximums of rows 1 and 2 of table (4.2a).

[10 8 10 8 5 2 1]

The same procedure was applied to the other four attributes. This way we end up with a (1,7) ; (1,2) ; (1,7) and a (1,5) vector which spans the four dimensional (7,2,7,5) space. The possible vectors are:

TABLE (4.5)

i= [10 8 10 8 5 2 1]

j= [10 0]

k= [10 8 10 10 10 5 10]

l= [10 4 10 7 10]

This (7,2,7,5) space has six blocks on each element, one for each control. This way we end up with a (7,2,7,5,6) space. Each of these blocks specified one of the controls where the first element represents full brake (-10) and the sixth element represents full acceleration (+5). Now the corresponding block for the control being -10, which is

block 1, is filled up with the minimum of the four vectors corresponding to that index. This number becomes the trueness of the -10 control value for that particular combination of the quantized attributes.

Some examples are given in the following table(4.6):

TABLE (4.6):

ijkl	Membership of C=-10
1111 = MIN(10,10,10,10)=	10
1211 = MIN(10,0,10,10)=	0
1161 = MIN(10,10,5,10)=	5

The above values are the minimums of the maximums given in table (4.5). Table (4.6) means that the trueness of the control to be -10 is 100% for the quantized attribute space of (1,1,1,1), 0% for (1,2,1,1) and 50% for (1,1,6,1).

## 4.3 - APPLICATION OF THE INTERPOLATION-BETWEEN-CORNERS METHOD

As explained earlier, the operator first scaled his relative worth judgements on the single variables. Table (4.7) shows the attribute points at which the worths were scaled.

Table (4.7):

DISTANCE ERROR (DE)
16 feet
18 feet
20 feet
22 feet
24 feet
26 feet
28 feet

BRAKE LIGHTS (BL)
on (1)
off (0)

VELOCITY ERROR (VE)
-6 mph
-4 mph
-2 mph
-0 mph
2 mph
4 mph
6 mph

SPEED (SP)
20 mph
30 mph
55 mph
80 mph
90 mph

Obviously, in this application the corner points were well defined since the attributes were all physical variables which were well ordered with the output variable. The brakes were not scaled since the values of this attribute consisted only of two extremes.

After scaling the worths of the above listed points in table (4.7) the relative worths for the combinations of the corner points were entered. The control values were the same as were used in the fuzzy sets. For convenience, they are shown again:

Table (4.4) : Output values

$$C = [-10 \quad -5 \quad -2 \quad 0 \quad 2 \quad 5]$$

The resulting control rule was of the following form:

Eq.(4.1)

$$\begin{aligned} C = & A_0 + A_1 * X + A_2 * Y + A_3 * Z + A_4 * T + \\ & A_5 * X * Y + A_6 * X * Z + A_7 * X * T + A_8 * Y * Z + A_9 * Y * T + A_{10} * Z * T + \\ & A_{11} * X * Y * Z + A_{12} * X * Y * T + A_{13} * Y * Z * T + A_{14} * X * Z * T + \\ & A_{15} * X * Y * Z * T \end{aligned}$$

Here, X, Y, Z and T are the scaled variables and assumed values between 0 and 1. C is the control value which ranges between -10 and +5. A0...A15 are the solution of a linear matrix equation set up through the inputs. The matrix is the extended version of equation (3.4)

## Chapter 5

## EQUIPMENT

## 5.1 - HARDWARE

In the man-machine laboratory of the Mechanical engineering department at MIT, a PDP 11/34 minicomputer and the following peripherals attached to it were used for the experiments:

- 1) LEXIDATA color raster graphics display with (512\*640\*12) pixel memory
- 2) MEGATEK vector graphics display with a resolution of (4096 \* 4096)
- 3) A ELOGRAPHICS touchpanel attached on front of the color monitor
- 4) A ANALOGICS AN5400 A/D D/A converter with various joysticks attached to it.

The color terminal was used together with the touchpanel in order to enter the human response. Information was entered to the computer by touching the appropriate areas displayed on the color graphics monitor.

## 5.2 - SOFTWARE

The following are the primary programs which were used:

- 1- Fuzzy set input program
- 2- Interpolation between corners input program
- 3- Fuzzy State Matrix display
- 4- In-Trail simulation program

### 1- Fuzzy sets input:

The color terminal displays areas which represent all fuzzy sets involved in the experiment and the control values to be implemented in the system. The lower part of the screen consists of areas to control the termination, the storage of the rules and verifications. The procedure is such that the operator touches the fuzzy description buttons which are displayed, uses the AND or OR buttons displayed on the screen in order to make combinations of the fuzzy conditions, and finally enters his judgement of control by pushing the appropriate button. One typical sequence would be as follows:

```
IF DISTANCE ERROR IS (SMALL OR VERY SMALL) AND BRAKE LIGHTS  
ARE ON; THEN INPUT=-10
```

The operator enters similar sequences until he thinks that enough information has been given or that the limit in the amount of instructions set by the designer has been reached. At any time during the inputting process the operator can

test the rules by entering the quantized real variables and checking what the computer understands from his inputs. This is done by means of a display in which he may enter any combination of attributes which might occur in the simulation. The membership or truth of each of the control rules for that combination of attributes is displayed as thick horizontal red lines on the control input area. The calculation of the truth for each control rule for a given attribute combination takes about two seconds.

#### 2- Interpolation between corners input program:

This display first shows the individual attributes and asks whether a data file created from the previous run should be used or whether a new input is wanted. If the user wants to use old values, he pushes the appropriate button. This enables the operator to change the control rules without having to enter all the individual scalings again. If new scaling is wanted, the computer displays on a vertical scale the maximum and minimum values of the attribute and asks the operator to enter the conceptual differences of the intermediate points.

After the four attributes have been finished, the display asks whether new corner points are necessary or whether inputs from previous runs are going to be used. This enables the operator to make minor changes in the relative scaling of the individual attributes without having





equation developed on the relative weighting points to create a "dummy" fuzzy state matrix with memberships of one for the control rule evaluated from the worth function for the corresponding attributes and a zero membership for the rest.

### 3- Fuzzy state matrix display:

This display shows the fuzzy truenesses of the state matrix. The control rules, which are calculated in advance, result in a (7,2,7,5) state action matrix. One way to display this fuzzy state matrix is the color graphics terminal. It is a suitable method for the user to determine on-line what the control rules are, how dominant any control rule is, what portions of the space would need more rules, or on which action of the control space bad rules have been implied.

The method is a simple display of a four attribute database. Since one of the attributes, namely the brake lights, consists of two values, the screen has been divided into two parts, one with brakelights on, the other with brake lights off. Each of these parts is again divided into seven by five boxes. The seven horizontal boxes show different values of the quantized distance errors whereas the five vertical boxes represent the five different quantized speeds. Each of the boxes is again divided into seven different regions where each region represent the

different velocity errors. This makes all together 490 regions. Each of these 490 quantized locations is a unique combination of the attributes and holds six bars with a different color for each control action. The length of each individual bar indicates the amount of trueness of the corresponding control action. If the results are not fuzzy, or if the fuzzy state matrix has been generated by a distinct algorithm such as the interpolation between corners method, the length of the bars will be full and there will be only one color seen. Each point in the database has a unique value with full membership and the other values have a membership of zero. The display then becomes the state action matrix display. Figs (5.1), (5.2)

#### 4- The In-Trail following simulation:

The In-trail following simulation takes place on the vector graphics terminal where an out of the windshield view of the simulation is displayed. The vehicle in the middle from which the distance margin has to be held, and the leading vehicle are shown. The subject driver's view is of a a straight road ahead where poles on the roadside which have constant separation are displayed and their velocity moving by on both sides is updated continuously according to the speed of the following vehicle so that a realistic perception of movement is available. In order to minimize computation time in the simulation, the vehicles are

SEPARATION

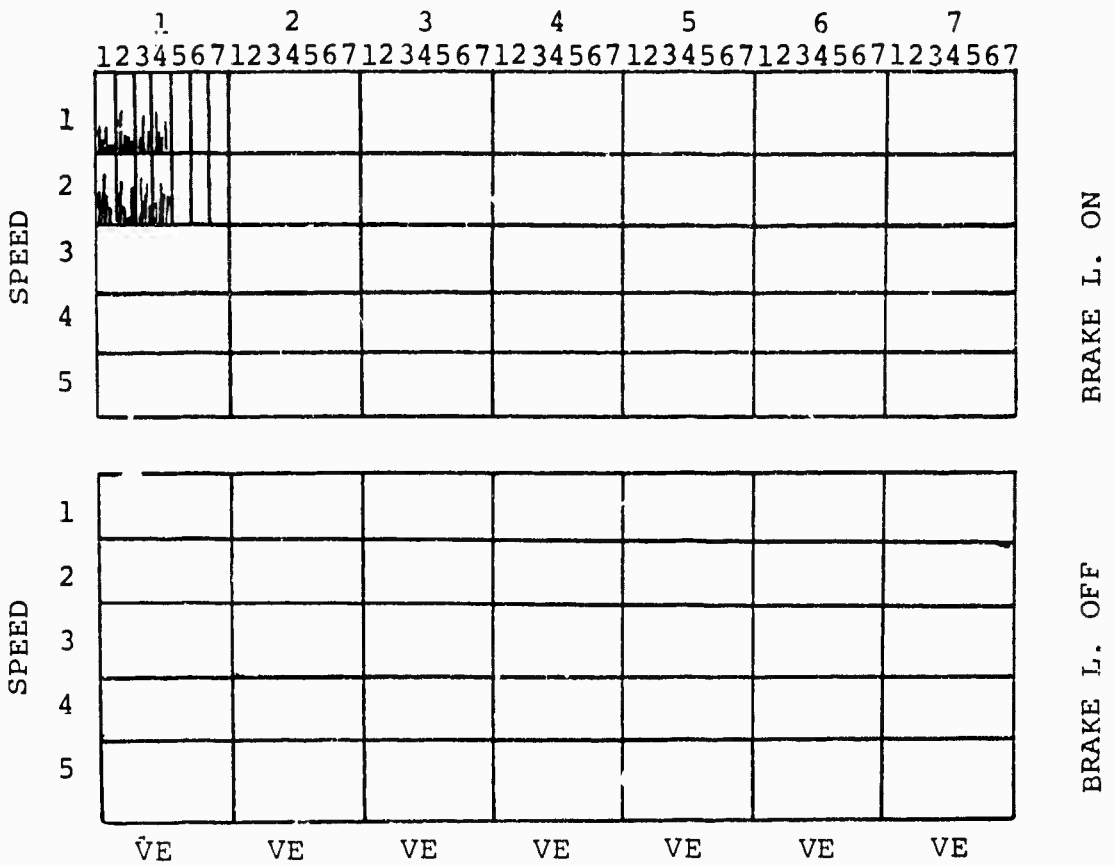


FIGURE 5.1 STATE ACTION MATRIX DISPLAY  
(VE= Velocity error)

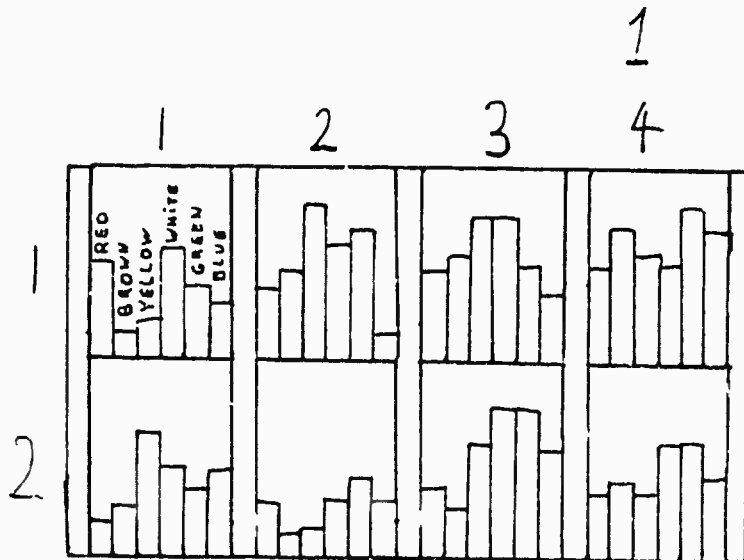


FIGURE 5.2  
Enlarged representation of the State Action  
Matrix display

designed like trucks seen from the back. The brake lights are represented by a cross appearing at their back. If the cross appears, it means that the vehicle is braking. If the cross cannot be seen, they are either accelerating or have constant speed. The perspective has one convergence point on the horizon and uses simple hidden line algorithms to make the display more realistic. However, the point of view chosen is like that from a high truck cab so that both vehicles can be seen most of the time except when the middlecar gets very close to the leadcar so that part of the back of leadcar and its wheels are hidden behind middlecar. The length of the vehicles is approximately 15 feet and the horizon is about 1000 feet away. If a collision occurs, the simulation is not interrupted. The distance error is kept constant until the leading vehicle attains a velocity to pull away from the collision.

Manual input to the in-trail following model was given in the experiment by means of a simple displacement joystick which is interfaced to the computer with the A/D converter to directly determine vehicle acceleration.

The automatic control to the in-trail following model is obtained through a mapping of the attributes to the state action matrix. In order to be able to use the continuous attributes directly, the modifications explained in chapter 6 have been made.

## chapter 6

## APPLICATION OF THE FUZZY STATE MATRIX

## 6.1 - OBTAINING CONTROL VALUES FROM THE FUZZY STATE MATRIX

All the different methods and algorithms described before result in a fuzzy state matrix which is to be applied to the model in order to achieve the results. It was shown in example (1), where an output variable with three different values was used, that the results can be contradictory. Several values of the variable can have a full membership of being true, or none of the variables can have a satisfactory membership value. There is no unique method to cope with these kinds of circumstances. These contradictions are results of bad attribute selections, human uncertainties and other factors. Several methods can be used to define a single-valued output from these contradictory results.

a) Using the output value which is kept based on direction of skew.

b) Taking the average of multiple peaks

c) Letting the computer prompt each time there is a contradiction in the control rules to be implemented.

Option (a) is useful during the control task, where, under uncertainty it might be more advisable to decelerate than to accelerate. Option (c) is used when a very accurate result is wanted and the number of uncertain points are few. The height of the peak is also taken into consideration. The

computer can be instructed to prompt the operator whenever the maximum membership is below a certain value.

Once the fuzzyness of output has been removed, a single valued state action matrix created in which each combination of attributes has a definite output value. If this state action matrix is displayed on the color graphics display, all bars have a full length and a single color is seen.

## 6.2 - DETERMINATION OF THE CONTINUOUS CONTROL FUNCTION FROM THE DISCRETE STATE ACTION MATRIX

Application of the control rules using coarsely quantized values sometimes led to problems. Since the control values are discrete points with step changes between different values of the attributes, wild oscillations occurred in the system simulations. These oscillations resulted in poor control, and therefore the necessity to smooth out the control was apparent. The smoothing was done as a linear interpolation between the combination of attributes. This smoothing enabled the computer to use continuous variables instead of the discrete variables. Since there is a four dimensional space, there were sixteen corners to be connected. The assumptions made were that the values in the corners had to be fixed and that between two corners the change of the value should be linear if only one attribute was changed and the rest were held fixed. An interpolation was made in such a way that the control value of that quantized attribute was attained when the continuous counterpart of that attribute was in the middle of the quantized region. At the end points of the variable scale, where values reach to infinity, the beginning point of the quantized variable was taken as the corner point. A two dimensional case is shown in the following example:

Example (3)

Assume that each attribute in a two dimensional space has been quantized into three discrete levels shown in tables (6.1a) and (6.1b) and the state action matrix shown in table (6.2) has been found:

TABLE (6.1) quantization:

TABLE (6.1a)

X=	1	2	3
x=	0--0.3	0.3--0.7	0.7--1.0

TABLE (6.1b)

Y=	1	2	3
y=	0--0.3	0.3--0.7	0.7--1.0

TABLE (6.2) State action matrix

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

There are  $(3*3)$  9 discrete values and  $(2*2)=4$  corners in this state action matrix. If a smoothing has to be done between corners 11 (X=1,Y=1); 12 (X=1,Y=2); 21 (X=2,Y=1) and 22 (X=2,Y=2), it is done in the way shown below. Center values are the points in continuous space which are assumed

to match exactly with the quantized attribute space and are the centers of the region quantized unless the attribute is single ended (i.e.  $x = -\infty$ ) in which case the closest point to the edge of the attribute is taken as the matching value (i.e. 0.5). Figure (6.1) shows a one-dimensional example on how the smoothing is done. It can be seen that the first and last points which match with the discrete variable are at the corner whereas the intermediate points are at the centers. If the edge points were also two-valued such as separation which cannot go below zero, the matching points at the edges would also be on the corners.

TABLE (6.3)

corner	output	Center values	
		x	y
11	0	0.3	0.15
12	1	0.3	0.5
21	3	0.5	0.15
22	4	0.5	0.5

The linear equation between these corners is in the form:

Eq. (6.1)

$$c=A+Bx+Cy+Dxy$$

$$x= -inf---0.5$$

$$y= 0--- 0.5$$

where the coefficients A,B,C and D satisfy the following linear equations:

$$A+0.3*B+0.15*C+0.3*0.15*D=0$$

$$A+0.3*B+0.5*C+0.3*0.5*D=1$$

$$A+0.5*B+0.15*C+0.5*0.15*D=3$$

$$A+0.5*B+0.5*C+0.5*0.5*D=4$$

These simultaneous equations lead to the following equation:

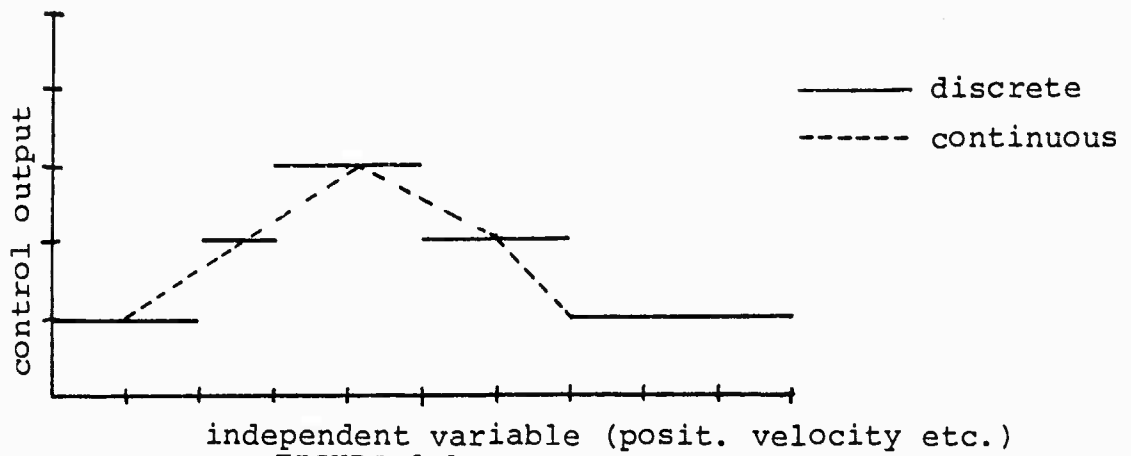
Eq. (6.2)

$$c=-4.928+15x+2.857y$$

$$x= -inf---0.5$$

$$y= 0--- 0.5$$

The rest of the equations connecting the quantized attributes are found in the same way.



independent variable (posit. velocity etc.)

FIGURE 6.1

smoothing of discrete values

## chapter 7

## EXPERIMENTS

## 7.1 - GENERAL RULES FOR THE TESTS

For the experiments done with the dynamic simulation the aim was to achieve a state action matrix which was collision free and also avoided excessive separations. The overall shape of the state action matrix and the time history of the simulations were the major factors in determining the effectiveness of the rules. Even though it was possible to enter the rules without considering the state action matrix, it was found that a set of guidelines about the rough shape of the matrix increases the efficiency of the inputs statements. The term "guidelines" describes basic simple statements about the system to be used as individual rules used to start teaching the computer.

Some of these guidelines are straightforward and some require more conceptual insight to the system. The guidelines have been developed throughout the experiments and are based on intuition. Because of the very trial and error nature of the process, not all different simulations and state action matrices are included in this work. There have been occasions where changes in the commands did not result in any changes in the state action matrix. Sometimes totally unacceptable results were produced. Table (7.1) shows the flowchart of the conducted experiments.

## Explanation of table (7.1)

- 1 - This node gets a new fuzzy input rule from the human operator.
- 2 - The input rule is evaluated mathematically
- 3 - If the operator thinks that the number of input rules is insufficient, he enters another rule
- 4 - The fuzzy state matrix is displayed on the color graphics monitor.
- 5 - If there are parts of the FSM which are not sufficient, (i.e maximum membership is very low ) new rules are entered.
- 6 - The state action matrix is displayed according to the policy chosen
- 7 - The controls are created using the state action matrix
- 8 - The simulation is executed and the time history of the simulation is displayed.
- 9 - If the results are not satisfactory, new rules are entered.
- 10 - The results are stored
- 11 - This node starts a new set of rules for the interpolation between corners method.
- 12 - The relative scalings and corners are entered.
- 13 - The corresponding worth function is evaluated.
- 14 - The dummy fuzzy state matrix with a single full membership of 1 is created
- 15 - This is the node where The two methods join.
- 16 - According to the method used this node branches either

to the left or to the right.

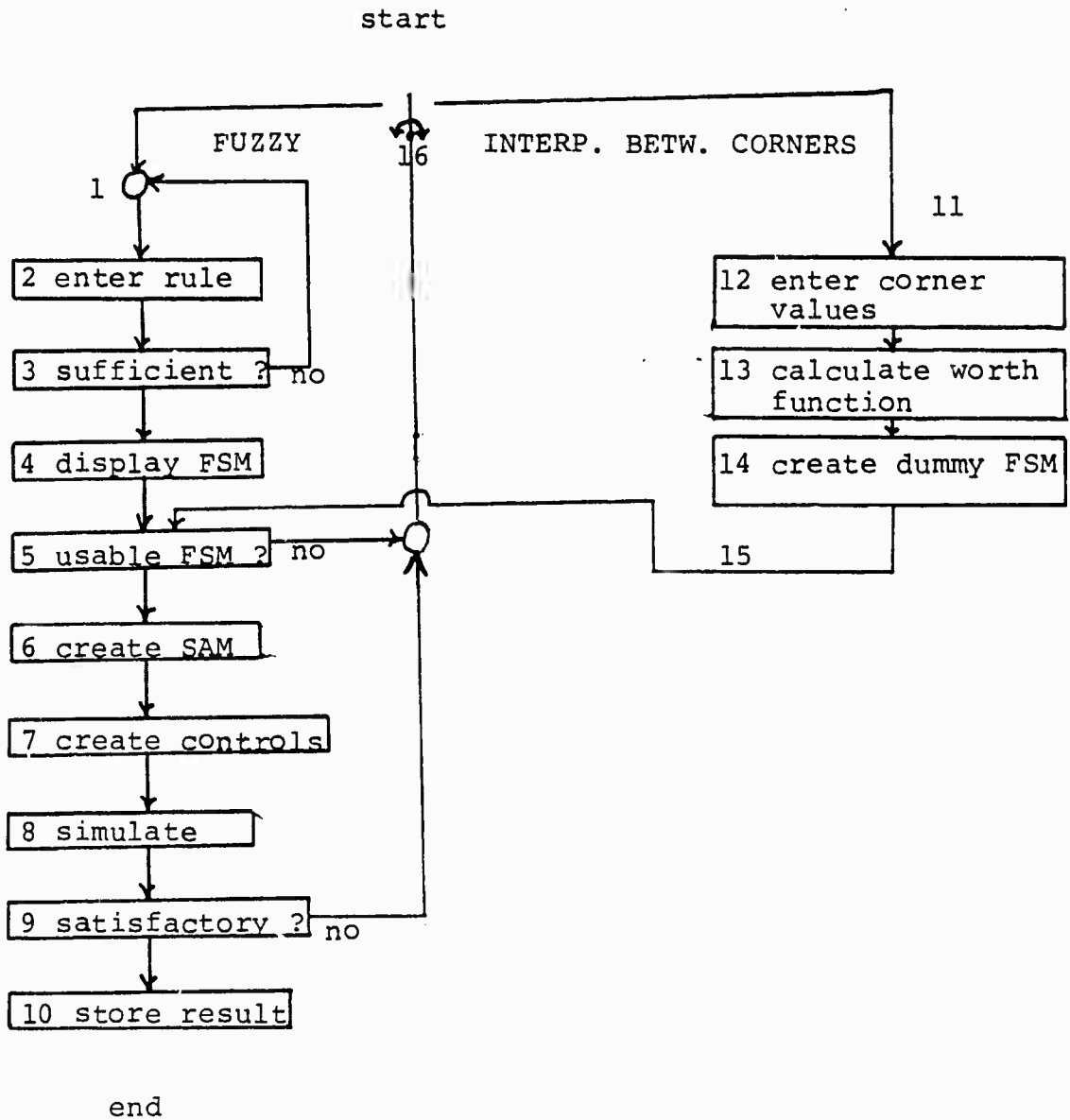


TABLE 7.1  
Flowchart of conducted experiments

Some of the basic guidelines in entering the control statements are the following:

- 1) If the brake lights are on, all the reactions should be based towards deceleration.
- 2) unless the separation is very high, a closing velocity error should induce a deceleration.
- 3) If the separation is large and increasing, acceleration should occur.

The above are some of the trivial guidelines. There are some less trivial ones which require more insight into the nature of nonlinear control. Likewise, some problems occurring from the mathematical structure of the system required certain rules to be used which might not have been necessary otherwise.

Even though the purpose of this project is not the analysis of the in-line following of cars, some consideration to the system characteristics has to be made in order to be able to set up useful rules for the computer.

The experiments started with the basic guidelines, and then, according to the performance of the simulation, the input rules were modified until a satisfactory response was obtained.

A standart test pattern is shown in figure (7.1). It was used as the input distubance to Leadcar. It is divided into eight regions. The following rationale has been made for these regions.

Region 1: The initial separation in this region is 100 ft and the Leadcar input is 3. There is a slight acceleration in the Leadcar where the basic "test" in this region is to see how well the initial separation is closed.

Region 2: The Leadcar goes up to full acceleration. This region has been used to see how well the system behaves in the transition from medium speed to full speed.

Region 3: In this region Leadcar decelerates fully without braking. This region investigates the system behaviour to negative velocity errors without the brake lights on.

Region 4: Leadcar gives full acceleration while the cars have fully stopped. The gap occurring during this acceleration is one of the most important factors in the design of the system.

Region 5: While moving with full speed, Leadcar applies full brake. This is the most critical region. If Controlcar brakes fully immediately when it sees the brake lights, a very large gap may occur. On the other hand, a slight hesitation might result in a crash.

Region 6: While at stillstand, Leadcar applies 60% full

acceleration to investigate the behaviour of the system under moderate accelerations.

Region 7: The full acceleration is applied in this region in order to bring the system to full speed for the next region.

Region 8: This region is important because it tests the system behaviour for slight braking, where the brake lights are on but there is no full deceleration. Usually, simple rules result in high gaps in this region because the system panics when it sees the brake lights.

The input rules in the IBC method resulted in tables such as table (7.2). It contains all the necessary information about the given input. The first three rows show the relative scalings of the separation, velocity error and speed. After this, a table showing the corner points, the attribute values of those corner points, the control and the index of that control as used in the state action matrix is given. The coefficients of the worth function are also given. The last part is the printout of the state action matrix in index form. The indices are as follows: 1=fullbrake 2=brake 3=decelerate 4=no power 5=power 6=full power. These are the indices of table (4.4). Table (7.3) shows the corner numbers of the state action matrix with the

order in which they have been entered in the IBC method.

Table (7.4) gives an example of the Fuzzy set input. It gives the linguistic rules which are separated with IF, AND, OR and ELSE statements. Table (7.5) shows an example of the resulting state action matrix. The form of this state action matrix is the same as the one in table (7.2). But it has been generated from a fuzzy state matrix. It is not convenient to represent a fuzzy state matrix on paper if there are four attributes since each element has six different values with percentages accompanied on each variable. The fuzzy rules should be entered in such a way that the minimum amount of inconsistency occurs.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

TABLE (7.3)  
Corners of the State Action Matrix

TABLE (7.4) Example of linguistic input rules

IF :	SEPARATION	IS VERY CLOSE
	OR SEPARATION	IS CLOSE
	OR SEPARATION	IS NORMAL.
AND	BRAKE LIGHT	IS ON
THEN FULL BRAKE		
ELSE,		
IF :	SEPARATION	IS FAR
	OR SEPARATION	IS VERY FAR
AND	BRAKE LIGHT	IS ON
THEN BRAKE		
ELSE,		
IF :	SEPARATION	IS VERY CLOSE
	OR SEPARATION	IS CLOSE
	OR SEPARATION	IS NORMAL.
	OR SEPARATION	IS FAR
AND	BRAKE LIGHT	IS OFF
AND	VELOCITY ERROR	IS CLOSING FAST
	OR VELOCITY ERROR	IS CLOSING
THEN DECELARATE		
ELSE,		
IF :	SEPARATION	IS FAR
	OR SEPARATION	IS VERY FAR
AND	BRAKE LIGHT	IS OFF
AND	VELOCITY ERROR	IS OPENING
	OR VELOCITY ERROR	IS OPENING FAST
THEN FULL POWER		
ELSE,		
DO NOTHING		

SEPERATION RELATIVE SCALINGS  
 1.00 0.80 0.65 0.50 0.30 0.10 0.00

VELOCITY ERROR RELATIVE SCALINGS  
 1.00 0.80 0.65 0.50 0.30 0.10 0.00

SPEED RELATIVE SCALING  
 1.00 0.75 0.50 0.25 0.00

#	SEPERATION	BRAKE LIGHTS	VELOCITY ERROR	SPEED	INPUT
1	16	ON	-6	20	FULLBRAKE 1
2	16	ON	6	20	FULLBRAKE 1
3	30	ON	-6	20	FULLBRAKE 1
4	30	ON	6	20	NO POWER 4
5	16	ON	-6	90	FULLBRAKE 1
6	16	ON	6	90	BRAKE 2
7	30	ON	-6	90	FULLBRAKE 1
8	30	ON	6	90	BRAKE 2
9	16	OFF	-6	20	FULLBRAKE 1
10	16	OFF	6	20	DECELERATE 3
11	30	OFF	-6	20	DECELERATE 3
12	30	OFF	6	20	FULLPOWER 6
13	16	OFF	-6	90	FULLBRAKE 1
14	16	OFF	6	90	BRAKE 2
15	30	OFF	-6	90	BRAKE 2
16	30	OFF	6	90	FULLPOWER 6

## COEFFICIENTS OF THE WORTH FUNCTION

6.0						
-4.0	-4.0	-4.0	0.0			
4.0	3.0	1.0	3.0	2.0	1.0	
-3.0	-4.0	-2.0	-3.0			
5.0						

## STATE ACTION MATRIX 11

1111111	1111122	1112222	1122223	1122235	1222334	1223344
1111111	1111122	1112222	1112222	1122233	1122333	1222334
1111112	1111222	1112222	1112222	1122223	1122233	1122233
1111122	1111222	1112222	1112222	1112222	1112222	1122223
1112222	1112222	1112222	1112222	1112222	1112222	1112222
1122233	1223334	2233344	2333445	2334455	3344556	3445566
1122233	1222333	2223344	2233444	2334455	3344556	3344566
1122223	1222333	2223344	2233444	2334455	2344556	3344566
1112222	1222233	1223334	2233344	2233455	2334556	2344566
1112222	1222233	1222333	2223344	2233445	2334456	2334566

TABLE 7.2 Example of an input with the IBC method

## STATE ACTION MATRIX 9

1111222	1111222	1112222	1111222	1135555	1155555	1155555
1111222	1111222	1112222	1111222	1111555	1135555	1135555
1111222	1111222	1112222	1111222	1111555	1125555	1115555
1111222	1111222	1112222	1111222	1111555	1111555	1111555
1111222	1111222	1112222	1111111	1111333	1111333	1111333
1112666	1111666	1111666	1116666	1556666	4556666	4456666
1111136	1111666	1111666	1116666	1556666	2556666	2556666
1111136	1111666	1111666	1116666	2356666	2356666	2356666
1111136	1111666	1111666	1116666	2356666	2356666	2356666
1111136	1111666	1111666	1116666	2356666	2356666	2356666

TABLE 7.5 Example of a state action matrix

## 7.2 - TESTS WITH FUZZY INPUT RULES

The two different fuzzy sets given in tables (4.2) and (4.3) have been used in the experiments. The first set of rules was given with the fuzzy state action matrix, with the size of (5,2,5,3), which is slightly smaller than the deterministic state action matrix. The corresponding membership functions are shown in table (4.2).

The second set of inputs was produced with a much smaller state action matrix (table (4.3)). In order to make the inputs as simple as possible, the smallest reasonable state action matrix has been used. It seemed to be necessary that the attributes which have to be corrected or which run from a positive to a negative value should have at least three divisions, namely negative, zero and positive.

The next sections describe the control rules which have been entered using the large fuzzy sets. Table (7.7) shows fuzzy control rules which are directly the implication of the guidelines stated above. It was interesting to see what kind of performance direct adherence to the simple guidelines would give. Table (7.15) shows the resulting state action matrix. Figures (7.2) and (7.15) show the result of the simulation. It can be seen that regions 1 and 2 are relatively reasonable even though the separation is too large and the correction takes more than 100 seconds. Region 3 results in collision, region 4 in a gap of more than 60 ft, region 5 in collision. Region 6 is acceptable

and region 8 results in a very large gap. It is evident that this is not a very good result. However it is surprising that even four simple rules entered result in a system which behaves very reasonably if no extremes are encountered. There are no big overshoots, no oscillations and no drifts. The system is stable.

Table (7.8) is an extended set of rules. The resulting state action matrix is shown in table (7.16). Figures (7.3) and (7.16) show the result of the simulation. There is some overshoot during the correction of the large gap in region 1. This overshoot occurs because of the large velocity error difference shown in figure (7.16b). From table (7.16) it can be seen that the state action matrix has a control index of 5 in corner 11. This leads to one of the major problems in using quantized variables. Since region 7 in the separation points to separations of 28 or more, there is no difference for the computer whether it is 30 or 100. The system is "shortsighted". The state action matrix tells that if the separation is 28 or more, even if the velocity error is larger (negative) than -6, give power. This, in turn results in an unstable system and in very large negative velocity errors which might not be corrected in time. If the corner is followed backwards in the separation error blocks of the corresponding state action matrix, it can be seen that until block 4 is reached, there is always a control index of 5. This means that the separation has to

come down to 23 before the system reacts to the big velocity difference. The way to correct this problem is to use the rule that regardless of separation, if the velocity error is very negative, do not apply more power. In region 5 a collusion occurs and the subsequent separation is around 35. Region 8 results in a large separation of 45.

In order to correct these deficiencies, the control rules in table 8 have been modified to result in the rules given in table (7.9). The resulting state action matrix is shown in table (7.17) and the simulation results are Figs (7.4) and 17. The difference between the rules can be best seen by studying the state action matrices on tables (7.16) and (7.17). It can be seen that corners 3, 7, 8, 9, 10, 11, and 12 are different now. So are also the transitions between the corners. The resulting simulation behaves fine until region 5 where a huge gap occurs, and it does it also in region 8. The system is "overbraking". This is mainly attributable to the state action matrix values between corners 7 and 8. They come from rule 2 in table 9 which says:

"IF DISTANCE ERROR IS NORMAL OR FAR OR VERY FAR AND IF BRAKE LIGHTS ARE ON AND IF OWN SPEED IS FAST THEN FULL BRAKE"

This rule has been smoothed out in the control rules given in table (7.10) by adding a rule which asks for slight brake

in corner 8. Since the previous rule is also present, corner 8 results in an average value of 2. Figs (7.5) and (7.18) show the result. It can be seen that even though some oscillations occur, the system behaves much better in region 5. Region 8 has still to be corrected.

Table (7.11) shows the rules which resulted with the state action matrix in table (7.19) and the response in figs (7.6) and (7.19). These results have been included to give an example that the modification of rules does not always result in increased system performance.

Table (7.12) is considered to be the set of rules which give the best response. The corresponding state action matrix is shown in table (7.20), and figure (7.20) shows the result of the dynamic simulation. It can be seen that the initial separation is corrected within 100 seconds. Regions 2, 3, and 4 do not produce any significant disturbance. In region 5 the separation does not move higher than 30 or less than 15. Finally, in region 8 the final separation is normal and the overshoot is reasonable. As it will be seen later, the assignment of the state action matrix was almost perfect. That means that the operator was able to produce the exact state action matrix he wanted. A comparison with the simulation results done with a manual state action matrix proved this fact.

This might not always be the case. If the fuzzy sets are coarse and the membership functions are widespread as

stated earlier, it might be more difficult to obtain good results. The small fuzzy sets with the membership functions shown in table (4.3) are used to investigate this matter. The rules have been entered from scratch without looking at the results of the large fuzzy set rules. The reason for this is that now the term 'very' cannot be used and only one term is available in each direction. (i.e close (-), normal (0) and far(+) etc.) .

Table (7.13) is the first set of rules used with the new membership functions. Table (7.21) shows the resulting state action matrix and figures (7.8) and (7.21) show the simulation results. It can be seen that it takes a while before the separation can be closed, and there are overshoots. Region 5 and region 8 result in very large gaps. After several trials, the rules given in table (7.14) proved to be the best ones possible with these membership functions. The state action matrix is given in table (7.22) and the simulation results in figures (7.9) and (7.22). Compared with figure (7.20) this might look like a bad performance but it still is acceptable, and the difference in the performances is not as great as it looks on the graphs. In the real-time simulation an inexperienced observer might not be able to tell which of the set of rules produce better results.

### 7.3 - TESTS WITH INTERPOLATION BETWEEN CORNERS

Table (7.25) shows the first set of rules entered in this part of the tests. The relative scalings of the values were held even for this first set and the values of the corners were entered in such a way that a state action matrix similar to one derived through fuzzy sets was obtained. The resulting state action matrix is shown on the same table. It looks similar to the ones obtained earlier.

Figure (7.7) can be seen to be an unacceptable set of rules. It is evident that the system had troubles in correcting the separation errors. Therefore adjustments have been made in corners 11 and 15. Corner 11 was changed from 'decelerate' (3) to 'no power' (4) and corner 15 was changed from 'brake' (2) to 'no power' (4). The result is shown in table (7.26). As it can be seen from the simulation results shown in figs (7.10) and (7.24) the response was faster but very oscillatory. Collusion occurred in region 5. If the corresponding state action matrix in table (7.26) is investigated, it can be seen that there are large zones where the control value does not change. It is neutral or slight decelerate or slight accelerate. State action matrix values like 3344455 or 3333444 predominate. This can be expressed as the control being very flat or insensitive to changes in the velocity error which results in the underdamped response. It is known from linear control that the feedback on the rate of

change of the error results in damping.

In order to improve the response with respect to the velocity error, the relative scalings have been shifted towards the edges. This can be seen in table (7.27). Figures (7.11) and (7.25) show that the oscillations are not present anymore and no collusion occurs. The initial separation slowly corrected but no overshoot occurred (overdamped). It can be seen that region 4 results in a gap of around 38 feet. The separation in region 5 comes dangerously close and the steady state separation is large. Table (7.28) shows an attempt to decrease the gap occurring in region 4 by increasing the value of corner 10 from 'decelerate' to 'full power'. As it can be seen from figure 26, this decreased the gap but resulted in a very short collusion during this period. In table (7.29) the values of corners 10 and 14 have been changed to 'power' and 'decelerate'. Figure (7.27) shows that the separation comes dangerously close in regions 3 and 5 and the gap in region 4 has also not improved.

Another attempt to increase the value at corner 10 to fullpower, this being in contrast to table (7.28) where corner 14 has the value of 'decelerate', does not result in any improvements. (table (7.30)).

It can be set that under the conditions given, figure (7.27) gave the most successful results of the simulation. The corresponding relative scalings and corner values are

given in table (7.29).

#### 7.4 - OTHER MODES OF INPUT

It can be seen that the most important factor is determining the state action matrix. The following question arises in this respect: What would happen if the state action matrix could be exactly determined by directly entering the values?

Tables (7.23) and (7.24) show two state action matrices which have been directly assigned values by the operator by observing the simulation behaviour and adjusting values until satisfied. Figures (7.29) and (7.30) show the simulation results. Figure (7.30) seems to respond slightly better in region 3. There is a very slight difference between those two state action matrices and the preference might be up to the individual person. If a comparison is made, it can be seen that those manual inputs resulted in much better performance than any other method. The time required and the number of trials was also much less than in any of the other methods, but it should be considered that this was a relatively simple system where the state action matrix was very easily displayable and modified by means of sophisticated computer text editors.

Figure (7.31) shows the reaction of an experienced human operator to the same Leadcar input pattern. It can be seen that there is a very significant amount of noise present in the system. Some of the noise is of low amplitude and some of it is high amplitude. The high

amplitude noise can be attributed to the observation noise of the separation and the low amplitude results from observation noise in the velocity error. There is also some system noise which results from positioning of the joystick used in the control. If the noise is filtered, a similarity can be found in the general form of response between the automatic controllers and the human operator. Figures

The state action matrix shown in table (7.24) was considered to be a good matrix. Figures (7.34) and (7.35) show that system reacts to an input where noise is added to the observation of the separation. It can be seen that the type of response is very much more like the human response shown in figure (7.31). This supports the assumption that the oscillations in the human operator responses are due to observation noise.

It can be seen that region 5 results in a gap occurring from a panic reaction. If the system had been controlled by a mathematical controller which does not accept human input for different combinations of the attributes, such a gap would most probably not occur. But the panic feeling of the human operator for brake lights under full power gets transformed into the system by means of his linguistic inputs and can clearly be observed in the simulations.

## chapter 8

## IMPLICATION OF THE METHODS FOR DATABASE SEARCHES

## 8.1 - MULTI-ATTRIBUTE DATABASE SEARCH

During the course of this thesis, a set of programs were developed for searching through a multi-attribute database by means of the methods described above. The in-trail following simulation was used primarily as the basis to compare the different algorithms. No systematic tests have been done with the simple and static data-base search. However, in order to show a different application of the techniques developed above, the brief description of this work is included in this report.

The computer has access to a database which consists of cars. The operator defines by different means the kind of car he likes, and the computer searches the database and displays the cars which are assumed to match the operators desires.

The term "input variable" will refer to the input of the operator in terms of how much he likes a particular car. It is a worth variable which has a numerical value between 0 and 1, where 1 is the greatest worth.

The following sections describe the use of fuzzy set and interpolation between corners techniques for doing this search / selection task.

## 8.2 - FUZZY SET ALGORITHM

Table (8.1) shows the four attributes in terms of fuzzy sets, each having three levels. It should be noted that this is relatively coarse. However it was observed that in this application a finer division of the attributes was not necessary.

TABLE (8.1):

PRICE	FUEL CONS.	TYPE	SIZE
cheap	low	sports car	small
moderate	medium	family car	medium
expensive	high	utility	large

Table (8.2) shows the input variable. The operator's judgement is "liking" - how much a test subject likes a particular combination. The values are quantized into seven regions and range from 0 to 10. Linguistic values are associated with the values to give the user an approximate feeling on what that value means.

TABLE (8.2):

LINGUISTIC	MEMBERSHIP
excellent	10
very good	8
fine	6
acceptable	5
indifferent	4
not liked	2
unwanted	0

When the operator enters values, which may sound like: "if the fuel consumption is low and price is cheap, it is excellent" the quantized attribute space (price fuel consumption, type, size) consists of a (7,7,7,7) array. Through the above input statement, the elements of that array get a membership value of being liked according to the memberships they have of being low in fuel consumption and being cheap. Each of the cars in the database has attributes which are a combination of the elements of this array. The state action matrix in the control task is in this case the same as the database, with one car being in each of the elements of that matrix. For example car 25 has the attributes  $i=3$   $j=5$   $k=7$   $l=7$  where  $ijkl$  refer to price, fuel consumption, type and size to which that car belongs. The input variable is a single variable which is quantized

into 7 values as shown in table (8.3). The computer checks the membership function of being liked for (3577). If it is above a certain value, the car is displayed on the color graphics monitor.

It can be seen that the output variable is a single attribute where the fuzzy membership value determines the degree to which it is liked. Since the subject does not need to classify the cars into different groups, this kind of an algorithm is sufficient. If the task were classification, different groups (eg. dislike, indifferent, like) would be used and each of these different groups would be assigned a membership in the database.

## 8.3 - INTERPOLATION BETWEEN CORNERS

In the application of this method to the multi-attribute database search there are again four attributes assumed. The procedure is similar to the procedure given in example (2). The scaled variables which range between 0 and 1 are called X, Y, Z and T. The operator is first asked to scale them relative to each other in order to define their conceptual difference. The attributes and the points on which they are scaled are given in table (8.3).

TABLE(8.3):

ATTR. 1 PRICE
\$ LOW - 6000
\$6000 - 7000
\$7000 - 8000
\$8000 - 9000
\$9000 - 10000
\$10000 - 11000
\$11000 - UP

ATTR. 2 FUEL CONS.
40 MPG
35 MPG
30 MPG
25 MPG
20 MPG
15 MPG
10 MPG

ATTR. 3 STYLING
sporty car
2 door sedan
4 door sedan
2 door hatchback
4 door hatchback
station wagon
pickup truck

ATTR. 4 SIZE
subcompact
compact
medium
large
very large

It is obvious that neither a monotonic ordering nor the corner points are fixed at the outset. The ordering of the attributes is completely dependent on the test subject. In terms of this example, one operator might have a preference for large cars before small cars, whereas another might prefer small cars. First, therefore the preference ordering and relative worth for the four attributes treated independently, must be entered. The computer can then find what values of the attributes represent the corners.

Then, the operator enters his relative worth judgements with respect to those corner points as was done in example (2). There are sixteen corner points which have to be entered. The operator's judgement variable is a continuous worth value which ranges from 0 to 1.

After this procedure the operator's worth function is calculated. It is a function with sixteen variables and has

the following form:

Eq. (8.1)

$$W=A+Bx+Cy+Dz+Et+Fxy+Gxz+Hxt+Kyz+Lyt \\ +Mzt+Nxyz+Pxyt+Ryzt+Sxzt+Txyzt$$

The computer can then, according to this worth function, make a judgement on what the worth of each individual car in the database is going to be. This is done by transforming the physical variables of consumption, price, size and styling into the scaled variables  $xyzt$  by means of the relative scaling input of the human operator and then evaluating the above worth function.

The computer program which is used to do the input and search is similar to the interpolation between corners input program for the control task except for the following: The 7 values of each of the 4 attributes shown in table (8.3) are displayed one by one and scaled between 0 and 1 relative to each other. On the scaling of the individual points, the computer does not display the extremes since they are not clear on the attributes chosen and vary from user to user. Therefore, the computer evaluates the corner points after the individual scaling. After entering the worth of the sixteen corner points, a worth region can be entered and the computer displays all the cars which are within this worth region. By doing this, the computer

fetches the properties of a car in a data file, and using the properties of that car, finds the attribute space to which the car belongs. Then those values are mapped to the scaled down variables  $x, y, z$  and  $t$ . After this, the worth function is evaluated to calculate the worth of that car. If it is within the region specified by the user, the car is displayed on the screen. If not, the computer fetches the next car on the data file. The user can change his worth region and see the new set of cars which satisfy his requirement. This way cars which are above a certain worth, below a certain worth, or those which are inbetween a worth region can be seen. If any of the car displays is touched, a printout of that car describing its model and technical properties is obtained.

## Chapter 9

## COMPARISON OF FUZZY SETS WITH

## THE INTERPOLATION BETWEEN CORNERS METHOD

## 9.1 - SOFTWARE CONSIDERATIONS: FUZZY SETS

The concept of fuzzy set is relatively difficult to understand for the beginner. There are many mathematical maximum and minimum calculations and matrix manipulations involved. A concept of handling multi-dimensional matrices efficiently without losing track of the indices is necessary. It is difficult to apply fuzzy set theory with small computers which do not have large memories available. Very large dynamic memory is required because of the size of the matrices.

The storage required is proportional to the number of elements of the discrete state action matrix multiplied with the number of input levels plus 1. ( $490*(6+1)$  in this case) plus some little matrices. Because of the extensive matrix manipulations required, the computer has to do many little operations which require time. The calculation time is proportional to the square of the number of elements of the discrete state action matrix. ( $490*490$  in this case). The computation time takes around 10 minutes

## 9.2 - SOFTWARE CONSIDERATIONS: INTERPOLATION BETWEEN CORNERS

The basic calculation in the interpolation between corners method is the inversion of an (n,n) matrix where n is the square of the number of attributes, which is  $4*4=16$  in this case. The size of the program is directly proportional to the size of this matrix. Since this method is designed to work on a continuous system, the number of the elements of the state action matrix does not play any role in the computation time. Therefore the execution time is very short. Including the quantization of the worth function by evaluating at the centers of the quantization points the program runs for approximately 10 seconds. Tables (9.1) and (9.2) are part of the list file generated by the computer and show the relative sizes of the programs.

TABLE (9.1) FUZZY INPUT PROGRAM MEMORY LISTING  
PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	\$CODE1	007272 1885	RW, I, CON, LCL
2	\$PDATA	000204 66	RW, D, CON, LCL
3	\$IDATA	000500 160	RW, D, CON, LCL
4	\$VARS	000756 247	RW, D, CON, LCL
5	\$TEMPS	000016 7	RW, D, CON, LCL

6	RCOM	005602	1473		RW,D,OVR,GBL
7	INPCOM	001000	256		RW,D,OVR,GBL
8	OUTCOM	000024	10		RW,D,OVR,GBL

## ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS	
BLTEXT	L*1	4-000576	000002	1	(2)
ICIR	I*2	4-000556	000010	4	(4)
ICONTI	I*2	8-000010	000014	6	(6)
IIA	I*2	4-000050	000050	20	(20)
IOPA	I*2	4-000000	000050	20	(20)
IOUT	I*2	7-000610	000170	60	(3,20)
IVAL	I*2	8-000000	000010	4	(4)
IVARSI	I*2	7-000000	000610	196	(7,7,4)
IXL1	I*2	4-000170	000050	20	(20)
IXL2	I*2	4-000310	000050	20	(20)
IXME	I*2	4-000430	000070	28	(4,7)
IYL1	I*2	4-000240	000050	20	(20)
IYL2	I*2	4-000360	000050	20	(20)
IYME	I*2	4-000520	000010	4	(4)
JJA	I*2	4-000120	000050	20	(20)
JX	I*2	9-016502	000210	68	(4,17)
NCOUNT	I*2	4-000530	000026	11	(0:10)

S	L*1	6-000000	005574	1470	(6,7,2,7,5)
SAM	L*1	9-015530	000752	245	(7,2,7,5)
TEXT	L*1	4-000566	000010	4	(8)
XTEXTL	*1	4-000600	000002	1	(2)

TABLE (9.2) INTERPOLATION BETWEEN CORNERS PROGRAM MEMORY  
LISTING

## PROGRAM SECTIONS

NUMBER	NAME	SIZE	SIZE	ATTRIBUTES
1	\$CODE1	004110	1060	RW, I, CON, LCL
2	\$PDATA	000070	28	RW, D, CON LCL
3	\$IDATA	000240	80	RW, D, CON, LCL
4	\$VARS	004764	1274	RW, D, CON, LCL
5	\$TEMPS	000026	11	RW, D, CON, LCL
6	WORTHC	000200	64	RW, D, OVR, GBL

## ARRAYS

NAME	TYPE	ADDRESS	SIZE	SIZE	DIMENSIONS
A	R*8	4-000000	004000	1024	(16, 16)
C	R*8	4-004000	000200	64	(16)
FARR	R*4	4-004416	000160	56	(4, 7)
ICONTI	*2	4-004576	000014	6	(5)
IEE	I*2	4-004372	000010	4	(4)
IFV1	I*2	4-004402	000014	6	(6)
TEXT	L*1	4-004360	000006	3	(6)

WA	R*4	4-004200	000160	56	(4,7)
X	R*8	5-000000	000200	64	(16)

It should be noted that these program sizes and execution times may vary considerably from programmer to programmer and might be reduced significantly if sophisticated algorithms are developed.

### 9.3 - APPLICATION CONSIDERATIONS

It can be seen from the simulations that the fuzzy input method resulted in significantly better results than the IBC method. One reason for this was that in the IBC method the rules were applied to the corners and the intermittent values are directly determined by those values. It is not possible to modify the corners without changing the values of the centers. This problem can be corrected partially by using the relative scaling of the individual attributes. But this causes the relative weights to change over all attributes. In systems where the characteristics change with the values of the attributes, a change in the relative characteristics might disturb the balance somewhere else in the system. In most multi-attribute tasks, the relative scalings of the individual attributes depend very much on where in attribute space the scaling is done. For example, in the in-trail following task, it is preferable to scale the negative velocity values close to the negative extreme during braking, but far from the negative extreme during acceleration. In the experiments, the reason why the separation error occurred during acceleration is that: The SAM has the value of 1112333 between corners 9 and 10 in table (7.27), where the 1112 is necessary in order to enable a sufficient braking if the separation is very close. But the values of 333 have been put there in order to have the value of 2 in the middle. If corner 10 were increased, the

value increases too. If the relative weight is changed toward the left, the values between corner 11 and 12 change too so that the value in the middle becomes a 4. The braking values also get skewed towards negative velocity error, which increases the gap during braking. That means that if we tell the computer to put more attention on negative velocity errors, it happens over the entire attribute space.

There are several methods to solve this problem. Generally, if the coarseness of the input values is reduced, it might be possible to increase the performance of the system. Methods are suggested with the assumption that the discretization is going to stay the same.

If the linear interpolation between corners method were increased to polynomial interpolation between corners by including square terms, a better approximation to the required state action matrix would be obtained. This would result in many more inputs but it could be managed by using second order terms only for those attributes which are necessary, or by applying default values for the intermittent points which are not entered. The computer can check whether a value is entered to satisfy the higher order term, and if not, it can assume that it is varying linearly.

The second method is a similar approach but instead of using second order terms, several pointwise continuous worth functions can be used. The intermediate point can be another

corner and totally independent values can be entered for that point. This is similar to assigning relative weights to the variables according to the point in the attribute space. Again the computer can assume that points which are not entered are in between the extreme corners.

Those improvements would require extensive complication in the program. The worth function would be useless without a computer because of its complexity. But the improvement would be very significant. The application of the relative scalings need to be changed accordingly or eventually totally discarded.

It is important however to mention that the interpolation between corners method is designed to work in a continuous attribute space. If the original attributes are continuous and if quantization results in big deficiencies, it might be preferable not to use fuzzy variables.

At the expense of much longer computation time, fuzzy sets resulted in much better assignment of the state action matrix and consequently a better response in the simulation. The size of the fuzzy state action matrix played an important role in the performance. When set 2 was used, the simulation had worse response. Since the membership functions were much more spread out in set 2, the rules had a wider effect and certain local points could not be adjusted. These are similar problems to those happening

with the interpolation between corners method. It can be seen by a comparison of figures (7.20) and (7.29) that set 1 resulted in almost ideal results if the manual assignment of the state action matrix is assumed to be perfect.

The author had the impression that in applications such as the one done in this report, the exact shape of the membership function does not play an important role if the user has direct feedback on his commands. The commands can be adjusted accordingly, assuming that the memberships are reasonably shaped in a unimodal manner.

#### 9.4 - HUMAN INTERFACE CONSIDERATIONS

The experiments did not formally determine the suitability of the different methods to human input from a purely psychological point of view. The human factors evaluation was anecdotal. The fuzzy set algorithm, with predetermined membership functions, seemed to be more easily understandable by the test subjects if compared to the interpolation between corners method. The main reason for this is the freedom of using intermittent values of the variables. Since fuzzy set theory actually (in this application) groups the values of the attributes into discrete groups and allows logical operations on them, restrictions given in the interpolation between corners method do not exist. If an operator knows what the linguistic descriptions mean, and if he knows what the meanings of "AND" and "OR" are, the input procedure becomes a very simple task. If desired, or if the operator feels himself more comfortable with the actual variables than the fuzzy descriptions, a graph showing the membership functions can be very helpful.

As to the interpolation between corners method, judgement by the subject of the relative scaling of the variables is necessary. This is an issue which is difficult for some operators to understand. The handicaps of this method which were explained in the previous sections are apparent in the relative scalings. Operators prefer not to

enter a relative scaling until they see the state action matrix and try to adjust it in order to get the desired values. If tables (25) to (31) are investigated, it can be seen that most of the relative scalings have not been changed at all. The fact that in order to change the effective mapping from a combination of attributes to the output variable one is not free to change only an intermediate input, but must change the corner value, is another handicap of this method. If the human operator is not familiar with the system, it requires some thinking in order to see how the change of a relative scaling will affect a state action matrix.

## Chapter 10

## RECOMMENDATIONS

## 10.1 - IMPROVED INPUT ALGORITHMS

The color graphics terminal was used as a man-machine interface during the experimentation. The individual attributes were displayed and the user entered his commands accordingly. For some time, because of some hardware problems, it was necessary to use data files instead of the color graphics terminal in order to enter the commands. It was observed that once the human operator knows what numbers to enter in specifying the combinations of the attributes, it was easier to use the data files than the graphics terminal. The main reason was the flexibility and the ease of correcting mistakes.

This leads to the thought that using the color graphics terminal may be wrong. In this work as well as in others, the color graphics were used mainly to display variables and to enter the combination of attributes. There are better utilizations of computers in conjunction with colored displays. In our application the graphics terminal should enable the operator to get into more direct contact with the database than he usually does in his mind. Since he already has the concept of the individual attributes in his mind, it is not an improvement to display those again. The database with its attributes has to be displayed in such a manner that the user can address a combination of the attributes

with one single entry. In order to explain this concept better, we refer to our control task as an example.

It is assumed that the user wants to enter a point where the separation is close, brake lights are on, the velocity error is slightly closing and speed is fast. The reader has to grasp those individual attributes one by one in order to understand which region is meant. The experienced operator however, since he is planning to enter a value for that combination, has it as a point in his mind; not a combination. That is the way he thought about that point in the beginning when he decided to enter a rule. If he can directly access that point, which he can in this task by the use of the state action matrix display, his entry will be much easier. He would only have to touch the appropriate region on the display and enter the value. If he can have direct feedback by the change of the state action matrix, he knows what effect his entry has on the attribute space. This method could be considered to use conceptual variables without translating them to linguistic variables.

The following is proposed as an efficient input algorithm: The discrete state action matrix is displayed on the screen with the axes labeled. If the entire matrix cannot be displays so that a travel through the database and rotations are possible.

In this proposal, fuzzy sets do not have fixed

membership functions; instead, the slope or decay of the membership function is displayed. The operator can enter any input to any point in the space. This input will get distributed through the database by means of the membership functions. The color which represents the input value might appear on the display to show the user what the effect of the input is. The intensity of the color could define the amount of membership of the input value. A display on the screen which shows a table mapping the color intensities to the membership function would be very helpful since the sensitivity of the human eye is different for different colors.

The above can be visualized like shaping the surface of a flat piece of clay by pulling up hills and painting them. The clay surface represents a two-dimensional attribute space, the color the value of the input and the height the membership of the input value to that attribute region on the surface. When one point is pulled up, the adjacent area gets lifted up. The slope of the volcano-shaped hill is the slope of the membership function. Finally, the flat piece of clay will consist of hills with different colors. The procedure can be carried on until the minimum elevation on the hill has a predefined value.

By using conceptual variables instead of linguistic variables through directly addressing the database, the problem of assigning membership function to linguistic

variables has been eliminated. Instead, there is a shape of membership function which may vary according to attribute or database region or both. In critical zones, the membership function might be steeper, and in less critical zones it might be flatter.

## 10.2 - ON-LINE MONITORING OF THE OPERATORS DECISIONS DURING CONTROL

The basic idea is to teach the computer the human decision characteristics by letting it monitor the human operator during the manual execution of the task for extended periods of time. Amos Freedy and others have done research in this area. ("Adaptive Aiding in Dynamic Decision Processes" (21)) The computer would set an empty fuzzy state matrix. The terms "fuzzy" and "membership" are used because of their similarity to methods used with fuzzy sets. They do not necessarily satisfy all requirements of fuzzy set theory. The matrix would be filled up with the number of occurrences of the different possible control actions. An example of this procedure can be given as follows:

Assume that one of the sampled points is the same as the quantized attribute space of [1211] in the fuzzy sets. There are six different possible control actions which can be taken at this attribute space. The control values of that quantized attribute space can be seen in table (10.1). Let us assume that the user passed that point 200 times during the control task. Naturally, dependent on the state from which the operator reached this point and some other factors, his control will vary. Assuming that he acted in the following way:

TABLE (10.1)

control		
frequency	value	percentage
130	-10	65%
50	-5	25%
20	0	5%

The frequencies can be interpreted in the same way as the membership functions. Using the above example, the control of that attribute space would have a membership of  $130/200 = 0.65$  to be -10,  $0.25$  to be -5 and  $0.05$  to be 0. It is evident that some extensive requirements have to be satisfied in order for this assumption to be true. But if the procedure works, the fuzzy state action (or relation) matrix can be directly set up by monitoring the human operator behaviour. The task should be executed long enough so that almost all the possible combinations of the attribute space have been passed. This procedure would be very sensitive to the choice of the attributes. If they have been chosen wrong, the procedure has no chance to work. If the attributes have been chosen correctly, there is a chance that the procedure will work. There is a considerable amount of processing of the sampled data necessary in order to obtain a working system. The attributes might have to be contingent upon where that point

has been reached and other factors. Research has to be done using one or two attributes.

### 10.3 - MONITORING FOR INTELLIGENT SEARCHES

A somewhat analogous method to the computer monitoring task can be implemented in the multi-attribute database search task. The idea is that the computer learns by experience what kind of objects are liked by the operator and what kind of ones are disliked. The output variable is in terms of fuzzy sets. The screen consists of a color graphics monitor with a suitable input device like the touchpanel attached to it. The computer can display random objects and accept the input of the operator by means of the input device. The input is in fuzzy terms like "very good", "good", "very bad" or similar descriptions which will be referred to as the worth table. The computer can, using fuzzy set methodologies, analyse the input of the operator and get a trueness about how much the operator is going to like a particular object. By considering the attributes and the human reaction to those attributes, the computer can build up a relation matrix. After a specified degree of certainty has been reached, the computer can display the object or objects which satisfy the given criteria.

An advanced way of doing this is having the computer act as a smart displayer. After having gathered some initial data of a few randomly chosen objects, the computer can use fuzzy set techniques to discriminate the next objects to be displayed by analysing the build-up of the relation matrix. Each new input will give new information on the

shape of the relation matrix and will aid in evaluating the direction of progress in the values of the individual attributes.

Fuzzy sets also provide the possibility to display more than one object. A group of objects which have similar properties displayed might give interesting results in how the operator likes this particular group and might speed up the search. Research cited in references (1) and (2) were using the computer as an advanced mechanical (but not thinking) mechanism to display the multi-attribute database.

The intelligent searcher can be considered as having the operator and computer switch places in certain computer aided design systems. In Vermeulen's system (3) the computer knows the properties of a gas turbine, and displays the technical data of the turbine and variables which show the differences between it and the designer's wish. The designer attempts to drive the discrepancy to zero while conforming to known physical and cost constraints. In the proposed method, the technical data would be equivalent to what the operator wants as an object, and the computer wants to drive the differences between the display and the operator's criteria to zero.

The computer uses the following algorithm to make the decisions: Each object has variables associated with it. These variables are mapped into fuzzy sets with the aid of membership functions. When an object is displayed, actually

a certain part of the database space is displayed which is not a single point, but a fuzzy region. When more than a single object is displayed, the only mathematical difference is that this fuzzy region increases. This way a relationship between these regions and the response of the operator, which is in fact the state action matrix, can be formed.

This algorithm and further developments of it are suitable to design learning machines and contain the most powerful application of fuzzy sets. A extension of this algorithm would be the learning of robots where the operator monitors the robot doing a task and in some way praises or punishes the robot according to the moves it makes.

A P P E N D I X

The Appendix contains the following information:

- 1- Computer output describing the linguistic set of rules used with the fuzzy input procedure.
- 2- Computer output showing the various state action matrices generated through the fuzzy input methods and the special methods (i.e. manual input)
- 3- Computer output showing the relative scalings and the entered corner values in the interpolation between corners method.
- 4- Plots of the time history of the simulations conducted. Most of these plots are shown with two different scales.

TABLE (7.7)

```

IF :
    SEPARATION          IS VERY CLOSE
  OR SEPARATION        IS CLOSE
  OR SEPARATION        IS NORMAL
AND
    BRAKE LIGHT        IS ON
THEN FULL BRAKE
ELSE,

IF :
    SEPARATION          IS FAR
  OR SEPARATION        IS VERY FAR
AND
    BRAKE LIGHT        IS ON
THEN BRAKE
ELSE,

IF :
    SEPARATION          IS VERY CLOSE
  OR SEPARATION        IS CLOSE
  OR SEPARATION        IS NORMAL
  OR SEPARATION        IS FAR
AND
    BRAKE LIGHT        IS OFF
AND
    VELOCITY ERROR     IS CLOSING FAST
  OR VELOCITY ERROR     IS CLOSING
THEN DECELERATE
ELSE,

IF :
    SEPARATION          IS FAR
  OR SEPARATION        IS VERY FAR
AND
    BRAKE LIGHT        IS OFF
AND
    VELOCITY ERROR     IS OPENING
  OR VELOCITY ERROR     IS OPENING FAST
THEN FULL POWER
ELSE,

DO NOTHING.

```

TABLE (7.8)

```

IF :
    SEPARATION          IS VERY CLOSE
  OR SEPARATION        IS CLOSE
  OR SEPARATION        IS NORMAL
  AND
    BRAKE LIGHT        IS ON
  AND
    VELOCITY ERROR     IS CLOSING FAST
  OR VELOCITY ERROR    IS CLOSING
  OR VELOCITY ERROR    IS SAME SPEED
THEN FULL BRAKE
ELSE,

IF :
    SEPARATION          IS VERY CLOSE
  OR SEPARATION        IS CLOSE
  AND
    BRAKE LIGHT        IS OFF
  AND
    VELOCITY ERROR     IS CLOSING FAST
  OR VELOCITY ERROR    IS CLOSING
  OR VELOCITY ERROR    IS SAME SPEED
THEN BRAKE
ELSE,

IF :
    SEPARATION          IS VERY CLOSE
  OR SEPARATION        IS CLOSE
  AND
    BRAKE LIGHT        IS ON
  OR VELOCITY ERROR    IS OPENING
  OR VELOCITY ERROR    IS OPENING FAST
THEN BRAKE
ELSE,

IF :
    SEPARATION          IS NORMAL
  OR SEPARATION        IS FAR
  OR SEPARATION        IS VERY FAR
  AND
    BRAKE LIGHT        IS ON
  AND
    SPEED              IS NORMAL
  OR SPEED             IS FAST
THEN BRAKE
ELSE,

IF :
    SEPARATION          IS CLOSE
  AND
    VELOCITY ERROR     IS SAME SPEED

```

```

                OR VELOCITY ERROR IS OPENING
                OR VELOCITY ERROR IS OPENING FAST
    THEN DECELARATE
ELSE,
IF :
                SEPARATION IS NORMAL
                OR SEPARATION IS FAR
                OR SEPARATION IS VERY FAR
    AND
                BRAKE LIGHT IS ON
    AND
                SPEED IS SLOW
    THEN NO POWER
ELSE,
IF :
                SEPARATION IS CLOSE
    AND
                BRAKE LIGHT IS OFF
    AND
                VELOCITY ERROR IS SAME SPEED
                OR VELOCITY ERROR IS OPENING
                OR VELOCITY ERROR IS OPENING FAST
    THEN POWER
ELSE,
IF :
                SEPARATION IS FAR
                OR SEPARATION IS VERY FAR
    AND
                BRAKE LIGHT IS OFF
    AND
                VELOCITY ERROR IS CLOSING FAST
                OR VELOCITY ERROR IS CLOSING
    THEN POWER
ELSE,
IF :
                SEPARATION IS NORMAL
                OR SEPARATION IS FAR
                OR SEPARATION IS VERY FAR
    AND
                BRAKE LIGHT IS OFF
    AND
                VELOCITY ERROR IS SAME SPEED
                OR VELOCITY ERROR IS OPENING
                OR VELOCITY ERROR IS OPENING FAST
    THEN FULL POWER
ELSE,

    DO NOTHING.

```

TABLE (7.9)

```

IF :
    SEPARATION          IS VERY CLOSE
    OR SEPARATION      IS CLOSE
    OR SEPARATION      IS NORMAL
    AND
    BRAKE LIGHT        IS ON
    AND
    VELOCITY ERROR     IS CLOSING FAST
    OR VELOCITY ERROR  IS CLOSING
    OR VELOCITY ERROR  IS SAME SPEED
    THEN FULL BRAKE
ELSE,
IF :
    SEPARATION          IS NORMAL
    OR SEPARATION      IS FAR
    OR SEPARATION      IS VERY FAR
    AND
    BRAKE LIGHT        IS ON
    AND
    SPEED              IS FAST
    THEN FULL BRAKE
ELSE,
IF :
    SEPARATION          IS NORMAL
    OR SEPARATION      IS FAR
    OR SEPARATION      IS VERY FAR
    AND
    BRAKE LIGHT        IS ON
    AND
    VELOCITY ERROR     IS CLOSING FAST
    OR VELOCITY ERROR  IS CLOSING
    AND
    SPEED              IS SLOW
    THEN FULL BRAKE
ELSE,
IF :
    SEPARATION          IS VERY CLOSE
    OR SEPARATION      IS CLOSE
    AND
    BRAKE LIGHT        IS OFF
    AND
    VELOCITY ERROR     IS CLOSING FAST
    OR VELOCITY ERROR  IS CLOSING
    THEN FULL BRAKE
ELSE,
IF :
    SEPARATION          IS VERY CLOSE

```

```

                OR SEPARATION      IS CLOSE
            AND
                BRAKE LIGHT        IS ON
                OR VELOCITY ERROR   IS OPENING
                OR VELOCITY ERROR   IS OPENING FAST
        THEN BRAKE
    ELSE,

```

```

    IF :
                SEPARATION      IS CLOSE
            AND
                VELOCITY ERROR   IS SAME SPEED
                OR VELOCITY ERROR IS OPENING
                OR VELOCITY ERROR IS OPENING FAST
        THEN DECELERATE
    ELSE,

```

```

    IF :
                SEPARATION      IS NORMAL
                OR SEPARATION    IS FAR
                OR SEPARATION    IS VERY FAR
            AND
                BRAKE LIGHT      IS ON
            AND
                SPEED            IS SLOW
        THEN NO POWER
    ELSE,

```

```

    IF :
                SEPARATION      IS CLOSE
            AND
                BRAKE LIGHT      IS OFF
            AND
                VELOCITY ERROR   IS SAME SPEED
                OR VELOCITY ERROR IS OPENING
                OR VELOCITY ERROR IS OPENING FAST
        THEN POWER
    ELSE,

```

```

    IF :
                SEPARATION      IS FAR
                OR SEPARATION    IS VERY FAR
            AND
                BRAKE LIGHT      IS OFF
            AND
                VELOCITY ERROR   IS CLOSING FAST
        THEN BRAKE
    ELSE,

```

```

    IF :
                SEPARATION      IS FAR
                OR SEPARATION    IS VERY FAR
            AND

```

```

                BRAKE LIGHT      IS OFF
    AND
                VELOCITY ERROR   IS CLOSING
    THEN POWER
ELSE,
IF :
                SEPARATION       IS NORMAL
    OR SEPARATION   IS FAR
    OR SEPARATION   IS VERY FAR
    AND
                BRAKE LIGHT      IS OFF
    AND
                VELOCITY ERROR   IS SAME SPEED
    OR VELOCITY ERROR IS OPENING
    OR VELOCITY ERROR IS OPENING FAST
    THEN FULL POWER
ELSE,
    DO NOTHING.
```

TABLE (7.10)

```

IF :
    SEPARATION          IS VERY CLOSE
    OR SEPARATION      IS CLOSE
    OR SEPARATION      IS NORMAL
AND
    BRAKE LIGHT        IS ON
AND
    VELOCITY ERROR     IS CLOSING FAST
    OR VELOCITY ERROR  IS CLOSING
    OR VELOCITY ERROR  IS SAME SPEED
THEN FULL BRAKE
ELSE,

IF :
    SEPARATION          IS NORMAL
    OR SEPARATION      IS FAR
    OR SEPARATION      IS VERY FAR
AND
    BRAKE LIGHT        IS ON
AND
    SPEED              IS FAST
THEN FULL BRAKE
ELSE,

IF :
    SEPARATION          IS NORMAL
    OR SEPARATION      IS FAR
    OR SEPARATION      IS VERY FAR
AND
    BRAKE LIGHT        IS ON
AND
    VELOCITY ERROR     IS CLOSING FAST
    OR VELOCITY ERROR  IS CLOSING
AND
    SPEED              IS SLOW
THEN FULL BRAKE
ELSE,

IF :
    SEPARATION          IS VERY CLOSE
    OR SEPARATION      IS CLOSE
AND
    BRAKE LIGHT        IS OFF
AND
    VELOCITY ERROR     IS CLOSING FAST
    OR VELOCITY ERROR  IS CLOSING
THEN FULL BRAKE
ELSE,

IF :
    SEPARATION          IS VERY CLOSE

```

```

                OR SEPARATION      IS CLOSE
            AND
                BRAKE LIGHT      IS ON
                OR VELOCITY ERROR IS OPENING
                OR VELOCITY ERROR IS OPENING FAST
        THEN BRAKE
    ELSE,

```

```

    IF :
                SEPARATION      IS FAR
                OR SEPARATION    IS VERY FAR
            AND
                BRAKE LIGHT      IS ON
                OR VELOCITY ERROR IS OPENING FAST
        THEN DECELERATE
    ELSE,

```

```

    IF :
                SEPARATION      IS CLOSE
            AND
                VELOCITY ERROR   IS SAME SPEED
                OR VELOCITY ERROR IS OPENING
                OR VELOCITY ERROR IS OPENING FAST
        THEN DECELERATE
    ELSE,

```

```

    IF :
                SEPARATION      IS NORMAL
                OR SEPARATION    IS FAR
                OR SEPARATION    IS VERY FAR
            AND
                BRAKE LIGHT      IS ON
            AND
                SPEED            IS SLOW
        THEN NO POWER
    ELSE,

```

```

    IF :
                SEPARATION      IS CLOSE
            AND
                BRAKE LIGHT      IS OFF
            AND
                VELOCITY ERROR   IS SAME SPEED
                OR VELOCITY ERROR IS OPENING
                OR VELOCITY ERROR IS OPENING FAST
        THEN POWER
    ELSE,

```

```

    IF :
                SEPARATION      IS FAR
                OR SEPARATION    IS VERY FAR
            AND
                BRAKE LIGHT      IS OFF

```

```

      AND      VELOCITY ERROR  IS CLOSING FAST
    THEN BRAKE
ELSE,
IF :
      SEPARATION      IS FAR
    OR SEPARATION     IS VERY FAR
  AND
      BRAKE LIGHT     IS OFF
  AND
      VELOCITY ERROR  IS CLOSING
    THEN POWER
ELSE,
IF :
      SEPARATION      IS NORMAL
    OR SEPARATION     IS FAR
    OR SEPARATION     IS VERY FAR
  AND
      BRAKE LIGHT     IS OFF
  AND
      VELOCITY ERROR  IS SAME SPEED
    OR VELOCITY ERROR IS OPENING
    OR VELOCITY ERROR IS OPENING FAST
    THEN FULL POWER
ELSE,
  DO NOTHING.

```

TABLE (7.11)

```

IF :
    SEPARATION          IS VERY CLOSE
  OR SEPARATION        IS CLOSE
  OR SEPARATION        IS NORMAL
AND
    BRAKE LIGHT        IS ON
AND
    VELOCITY ERROR     IS CLOSING FAST
  OR VELOCITY ERROR   IS CLOSING
  OR VELOCITY ERROR   IS SAME SPEED
THEN FULL BRAKE
ELSE,

IF :
    SEPARATION          IS NORMAL
  OR SEPARATION        IS FAR
  OR SEPARATION        IS VERY FAR
AND
    BRAKE LIGHT        IS ON
AND
    SPEED              IS FAST
THEN FULL BRAKE
ELSE,

IF :
    SEPARATION          IS NORMAL
  OR SEPARATION        IS FAR
  OR SEPARATION        IS VERY FAR
AND
    BRAKE LIGHT        IS ON
AND
    VELOCITY ERROR     IS CLOSING FAST
  OR VELOCITY ERROR   IS CLOSING
AND
    SPEED              IS SLOW
THEN FULL BRAKE
ELSE,

IF :
    SEPARATION          IS VERY CLOSE
  OR SEPARATION        IS CLOSE
AND
    BRAKE LIGHT        IS OFF
AND
    VELOCITY ERROR     IS CLOSING FAST
  OR VELOCITY ERROR   IS CLOSING
THEN FULL BRAKE
ELSE,

IF :
    SEPARATION          IS VERY CLOSE

```

```

        OR SEPARATION: IS CLOSE
    AND
        BRAKE LIGHT IS ON
        OR VELOCITY ERROR IS OPENING
        OR VELOCITY ERROR IS OPENING FAST
    THEN BRAKE
ELSE,

```

```

IF :
        SEPARATION IS FAR
    OR SEPARATION IS VERY FAR
    AND
        BRAKE LIGHT IS ON
        OR VELOCITY ERROR IS OPENING FAST
    THEN DECELERATE
ELSE,

```

```

IF :
        SEPARATION IS CLOSE
    AND
        BRAKE LIGHT IS ON
    AND
        VELOCITY ERROR IS SAME SPEED
        OR VELOCITY ERROR IS OPENING
        OR VELOCITY ERROR IS OPENING FAST
    THEN DECELERATE
ELSE,

```

```

IF :
        SEPARATION IS NORMAL
        OR SEPARATION IS FAR
        OR SEPARATION IS VERY FAR
    AND
        BRAKE LIGHT IS ON
    AND
        SPEED IS SLOW
    THEN NO POWER
ELSE,

```

```

IF :
        SEPARATION IS CLOSE
    AND
        BRAKE LIGHT IS OFF
    AND
        VELOCITY ERROR IS SAME SPEED
        OR VELOCITY ERROR IS OPENING
        OR VELOCITY ERROR IS OPENING FAST
    THEN POWER
ELSE,

```

```

IF :
        SEPARATION IS FAR
    OR SEPARATION IS VERY FAR

```

```

AND
    BRAKE LIGHT      IS OFF
AND
    VELOCITY ERROR  IS CLOSING FAST
AND
    SPEED            IS NORMAL
    OR SPEED        IS FAST
THEN BRAKE
ELSE,

```

```

IF :
    SEPARATION      IS VERY FAR
AND
    BRAKE LIGHT     IS OFF
AND
    VELOCITY ERROR  IS CLOSING FAST
AND
    SPEED           IS SLOW
THEN NO POWER
ELSE,

```

```

IF :
    SEPARATION      IS FAR
    OR SEPARATION   IS VERY FAR
AND
    BRAKE LIGHT     IS OFF
AND
    VELOCITY ERROR  IS CLOSING
THEN POWER
ELSE,

```

```

IF :
    SEPARATION      IS NORMAL
    OR SEPARATION   IS FAR
    OR SEPARATION   IS VERY FAR
AND
    BRAKE LIGHT     IS OFF
AND
    VELOCITY ERROR  IS SAME SPEED
    OR VELOCITY ERROR IS OPENING
    OR VELOCITY ERROR IS OPENING FAST
THEN FULL POWER
ELSE,

```

```

IF :
    SEPARATION      IS VERY CLOSE
AND
    BRAKE LIGHT     IS OFF
AND
    VELOCITY ERROR  IS OPENING FAST
AND
    SPEED           IS SLOW
THEN FULL POWER

```

ELSE,

DO NOTHING.

TABLE (7.12)

```

IF :
    SEPARATION          IS VERY CLOSE
    OR SEPARATION       IS CLOSE
    OR SEPARATION       IS NORMAL
    AND
    BRAKE LIGHT        IS ON
    AND
    VELOCITY ERROR     IS CLOSING FAST
    OR VELOCITY ERROR  IS CLOSING
    OR VELOCITY ERROR  IS SAME SPEED
    THEN FULL BRAKE
ELSE,

IF :
    SEPARATION          IS NORMAL
    OR SEPARATION       IS FAR
    OR SEPARATION       IS VERY FAR
    AND
    BRAKE LIGHT        IS ON
    AND
    SPEED              IS FAST
    THEN FULL BRAKE
ELSE,

IF :
    SEPARATION          IS NORMAL
    OR SEPARATION       IS FAR
    OR SEPARATION       IS VERY FAR
    AND
    BRAKE LIGHT        IS ON
    AND
    VELOCITY ERROR     IS CLOSING FAST
    AND
    SPEED              IS SLOW
    OR SPEED           IS NORMAL
    THEN FULL BRAKE
ELSE,

IF :
    SEPARATION          IS VERY CLOSE
    OR SEPARATION       IS CLOSE
    AND
    BRAKE LIGHT        IS ON
    OR VELOCITY ERROR  IS OPENING
    OR VELOCITY ERROR  IS OPENING FAST
    THEN BRAKE
ELSE,

IF :
    SEPARATION          IS FAR
    OR SEPARATION       IS VERY FAR

```

```

AND
    BRAKE LIGHT IS ON
AND
    VELOCITY ERROR IS SAME SPEED
    OR VELOCITY ERROR IS OPENING
    OR VELOCITY ERROR IS OPENING FAST
AND
    SPEED IS SLOW
THEN POWER
ELSE,

```

```

IF :
    SEPARATION IS FAR
    OR SEPARATION IS VERY FAR
AND
    BRAKE LIGHT IS ON
AND
    VELOCITY ERROR IS OPENING
    OR VELOCITY ERROR IS OPENING FAST
THEN POWER
ELSE,

```

```

IF :
    SEPARATION IS CLOSE
AND
    BRAKE LIGHT IS ON
AND
    VELOCITY ERROR IS SAME SPEED
    OR VELOCITY ERROR IS OPENING
    OR VELOCITY ERROR IS OPENING FAST
THEN DECELERATE
ELSE,

```

```

IF :
    SEPARATION IS VERY CLOSE
    OR SEPARATION IS CLOSE
AND
    BRAKE LIGHT IS OFF
AND
    VELOCITY ERROR IS CLOSING FAST
    OR VELOCITY ERROR IS CLOSING
THEN FULL BRAKE
ELSE,

```

```

IF :
    SEPARATION IS VERY CLOSE
AND
    BRAKE LIGHT IS OFF
AND
    VELOCITY ERROR IS SAME SPEED
AND
    SPEED IS SLOW
THEN BRAKE

```

```

ELSE,
IF :
    SEPARATION      IS CLOSE
    AND
    BRAKE LIGHT     IS OFF
    AND
    VELOCITY ERROR  IS OPENING
    OR VELOCITY ERROR IS OPENING FAST
    THEN FULL POWER
ELSE,
IF :
    SEPARATION      IS FAR
    OR SEPARATION   IS VERY FAR
    AND
    BRAKE LIGHT     IS OFF
    AND
    VELOCITY ERROR  IS CLOSING FAST
    AND
    SPEED           IS NORMAL
    OR SPEED        IS FAST
    THEN BRAKE
ELSE,
IF :
    SEPARATION      IS VERY FAR
    AND
    BRAKE LIGHT     IS OFF
    AND
    VELOCITY ERROR  IS CLOSING FAST
    AND
    SPEED           IS SLOW
    THEN NO POWER
ELSE,
IF :
    SEPARATION      IS FAR
    OR SEPARATION   IS VERY FAR
    AND
    BRAKE LIGHT     IS OFF
    AND
    VELOCITY ERROR  IS CLOSING
    THEN POWER
ELSE,
IF :
    SEPARATION      IS NORMAL
    OR SEPARATION   IS FAR
    OR SEPARATION   IS VERY FAR
    AND
    BRAKE LIGHT     IS OFF
    AND

```

```
                VELOCITY ERROR IS SAME SPEED
                OR VELOCITY ERROR IS OPENING
                OR VELOCITY ERROR IS OPENING FAST
    THEN FULL POWER
ELSE,

IF :
    AND SEPARATION IS VERY CLOSE
    AND BRAKE LIGHT IS OFF
    AND VELOCITY ERROR IS OPENING FAST
    OR VELOCITY ERROR IS OPENING
    AND SPEED IS SLOW
    THEN FULL POWER
ELSE,

    DO NOTHING.
```

TABLE (7.14)

```

IF :
    SEPARATION          IS CLOSE
  OR SEPARATION        IS NORMAL
  AND
    BRAKE LIGHT        IS ON
  AND
    VELOCITY ERROR     IS CLOSING
  OR VELOCITY ERROR   IS SAME SPEED
  THEN FULL BRAKE
ELSE,

IF :
    SEPARATION          IS CLOSE
  OR SEPARATION        IS NORMAL
  AND
    BRAKE LIGHT        IS ON
  AND
    VELOCITY ERROR     IS OPENING
  AND
    SPEED               IS SLOW
  THEN NO POWER
ELSE,

IF :
    SEPARATION          IS CLOSE
  OR SEPARATION        IS NORMAL
  AND
    BRAKE LIGHT        IS ON
  AND
    VELOCITY ERROR     IS OPENING
  AND
    SPEED               IS FAST
  THEN BRAKE
ELSE,

IF :
    SEPARATION          IS FAR
  AND
    BRAKE LIGHT        IS ON
  AND
    VELOCITY ERROR     IS CLOSING
  OR VELOCITY ERROR   IS SAME SPEED
  AND
    SPEED               IS SLOW
  THEN BRAKE
ELSE,

IF :
    SEPARATION          IS FAR
  AND
    BRAKE LIGHT        IS ON

```

AND  
     VELOCITY ERROR IS OPENING  
 THEN POWER  
 ELSE,

IF :  
     SEPARATION IS FAR  
 AND  
     BRAKE LIGHT IS ON  
 AND  
     VELOCITY ERROR IS CLOSING  
     OR VELOCITY ERROR IS SAME SPEED  
 AND  
     SPEED IS FAST  
 THEN FULL BRAKE  
 ELSE,

IF :  
     SEPARATION IS CLOSE  
     OR SEPARATION IS NORMAL  
 AND  
     BRAKE LIGHT IS OFF  
 AND  
     VELOCITY ERROR IS CLOSING  
 AND  
     SPEED IS FAST  
 THEN FULL BRAKE  
 ELSE,

IF :  
     SEPARATION IS CLOSE  
 AND  
     BRAKE LIGHT IS OFF  
 AND  
     VELOCITY ERROR IS CLOSING  
 AND  
     VELOCITY ERROR IS SAME SPEED  
 AND  
     SPEED IS SLOW  
 THEN FULL BRAKE  
 ELSE,

IF :  
     SEPARATION IS CLOSE  
 AND  
     BRAKE LIGHT IS OFF  
 AND  
     VELOCITY ERROR IS OPENING  
 AND  
     SPEED IS SLOW  
 THEN FULL POWER  
 ELSE,

```

IF :
    SEPARATION      IS CLOSE
    AND
    BRAKE LIGHT     IS OFF
    AND
    VELOCITY ERROR  IS OPENING
    AND
    SPEED           IS FAST
    THEN POWER
ELSE,

IF :
    SEPARATION      IS FAR
    AND
    BRAKE LIGHT     IS OFF
    AND
    VELOCITY ERROR  IS CLOSING
    AND
    SPEED           IS FAST
    THEN DECELERATE
ELSE,

IF :
    SEPARATION      IS NORMAL
    OR SEPARATION   IS FAR
    AND
    BRAKE LIGHT     IS OFF
    AND
    VELOCITY ERROR  IS SAME SPEED
    OR VELOCITY ERROR IS OPENING
    THEN FULL POWER
ELSE,

IF :
    SEPARATION      IS FAR
    AND
    BRAKE LIGHT     IS OFF
    AND
    VELOCITY ERROR  IS CLOSING
    AND
    SPEED           IS SLOW
    THEN NO POWER
ELSE,

IF :
    SEPARATION      IS NORMAL
    AND
    BRAKE LIGHT     IS OFF
    AND
    VELOCITY ERROR  IS SAME SPEED
    AND
    SPEED           IS FAST
    THEN NO POWER

```

ELSE,

IF :

SEPARATION IS CLOSE

AND

BRAKE LIGHT IS OFF

AND

VELOCITY ERROR IS SAME SPEED

THEN DECELERATE

ELSE,

DO NOTHING.

TABLE (7.15)

## STATE ACTION MATRIX 1

111111	111111	111111	111111	222222	222222	222222
111111	111111	111111	111111	222222	222222	222222
111111	111111	111111	111111	222222	222222	222222
111111	111111	111111	111111	222222	222222	222222
111111	111111	111111	111111	222222	222222	222222
333333	333333	333334	333346	333366	333466	334666
333333	333333	333334	333346	333366	333466	334666
333333	333333	333334	333346	333366	333466	334666
333333	333333	333334	333346	333366	333466	334666
333333	333333	333334	333346	333366	333466	334666

TABLE (7.16)

## STATE ACTION MATRIX 2

111222	111222	111222	222444	444444	444444	444444
111222	111222	111222	111222	111122	222222	222222
111222	111222	111222	111222	222222	222222	222222
111222	111222	111222	111222	222222	222222	222222
111222	111222	111222	111222	222222	222222	222222
222234	222444	222344	222666	555666	555666	555666
222234	222444	222344	222666	555666	555666	555666
222234	222444	222344	222666	555666	555666	555666
222234	222444	222344	222666	555666	555666	555666
222234	222444	222344	222666	555666	555666	555666

TABLE (7.17)

## STATE ACTION MATRIX 3

1111222	1111222	1112222	2222444	2224444	2224444	2224444
1111222	1111222	1112222	1111222	1111122	2222223	2222224
1111222	1111222	1112222	1111222	1111122	1222222	1111111
1111222	1111222	1112222	1111222	1111111	1111111	1111111
1111222	1111222	1112222	1111111	1111111	1111111	1111111
1111134	1113444	1114444	1116666	2356666	2356666	2356666
1111134	1113444	1114444	1116666	2356666	2356666	2356666
1111134	1113444	1114444	1116666	2356666	2356666	2356666
1111134	1113444	1114444	1116666	2356666	2356666	2356666
1111134	1113444	1114444	1116666	2356666	2356666	2356666

TABLE (7.18)

## STATE ACTION MATRIX 4

1111222	1111222	1112222	2222444	2224443	2224443	2224443
1111222	1111222	1112222	1111222	1111123	2222233	2222233
1111222	1111222	1112222	1111222	1111123	1222233	1111233
1111222	1111222	1112222	1111222	1111113	1111113	1111113
1111222	1111222	1112222	1111111	1111112	1111112	1111112
1111134	1113444	1114444	1116666	2356666	2356666	2356666
1111134	1113444	1114444	1116666	2356666	2356666	2356666
1111134	1113444	1114444	1116666	2356666	2356666	2356666
1111134	1113444	1114444	1116666	2356666	2356666	2356666
1111134	1113444	1114444	1116666	2356666	2356666	2356666

TABLE (7.19)

## STATE ACTION MATRIX 5

1111222	1111222	1112222	2222444	2224443	2224443	2224443
1111222	1111222	1112222	1111222	1111123	2222233	2222233
1111222	1111222	1112222	1111222	1111123	1222233	1111233
1111222	1111222	1112222	1111222	1111113	1111113	1111113
1111222	1111222	1112222	1111111	1111112	1111112	1111112
1111166	1113555	1115555	1116666	1556666	4556666	4456666
1111135	1113555	1115555	1116666	1556666	2556666	2556666
1111135	1113555	1115555	1116666	2356666	2356666	2356666
1111135	1113555	1115555	1116666	2356666	2356666	2356666
1111135	1113555	1115555	1116666	2356666	2356666	2356666

TABLE (7.20)

## STATE ACTION MATRIX 6

1111222	1111222	1112222	1111222	1135555	1155555	1155555
1111222	1111222	1112222	1111222	1111555	1135555	1135555
1111222	1111222	1112222	1111222	1111555	1125555	1115555
1111222	1111222	1112222	1111222	1111555	1111555	1111555
1111222	1111222	1112222	1111111	1111333	1111333	1111333
1112666	1111666	1111666	1116666	1556666	4556666	4456666
1111136	1111666	1111666	1116666	1556666	2556666	2556666
1111136	1111666	1111666	1116666	2356666	2356666	2356666
1111136	1111666	1111666	1116666	2356666	2356666	2356666
1111136	1111666	1111666	1116666	2356666	2356666	2356666

TABLE (7.21)

## STATE ACTION MATRIX 7

1111222	1111222	1111222	1111222	1111333	2222444	2222444
1111222	1111222	1111222	1111222	1111333	2222444	2222444
1111222	1111222	1111222	1111222	1111222	1111122	1111122
1111222	1111222	1111222	1111222	1111111	1111111	1111111
1111222	1111222	1111222	1111222	1111111	1111111	1111111
1113666	1113666	1123666	2223466	4426666	4446666	4446666
1113666	1113666	1123666	1223466	4426666	4446666	4446666
1113555	1113555	1123555	1124555	3325555	3336666	3336666
1113555	1113555	1113555	1114555	2215555	2226666	2226666
1113555	1113555	1113555	1114555	2215555	2226666	2226666

TABLE (7.22)

## STATE ACTION MATRIX 8

1111444	1111444	1111444	1111444	1111444	2222555	2222555
1111444	1111444	1111444	1111444	1111444	2222555	2222555
1111333	1111333	1111333	1111333	1111333	1111555	1111555
1111222	1111222	1111222	1111222	1111333	1111555	1111555
1111222	1111222	1111222	1111222	1111333	1111555	1111555
1112666	1112666	1113666	2236666	4446666	4446666	4446666
1113666	1112666	1113666	2236666	4446666	4446666	4446666
1113555	1113555	1113555	1116666	2226666	3336666	3336666
1113555	1113555	1114555	1116666	2226666	3336666	3336666
1113555	1113555	1114555	1116666	2226666	3336666	3336666

TABLE (7.23)

## STATE ACTION MATRIX 9

1111222	1111222	1112222	1111222	1135555	1155555	1155555
1111222	1111222	1112222	1111222	1111555	1135555	1135555
1111222	1111222	1112222	1111222	1111555	1125555	1115555
1111222	1111222	1112222	1111222	1111555	1111555	1111555
1111222	1111222	1112222	1111111	1111333	1111333	1111333
1112666	1111666	1111666	1116666	1556666	4556666	4456666
1111136	1111666	1111666	1116666	1556666	2556666	2556666
1111136	1111666	1111666	1116666	2356666	2356666	2356666
1111136	1111666	1111666	1116666	2356666	2356666	2356666
1111136	1111666	1111666	1116666	2356666	2356666	2356666

TABLE (7.24)

## STATE ACTION MATRIX 10

1111222	1111222	1112222	1111222	1135555	1155555	1155555
1111222	1111222	1112222	1111222	1111555	1135555	1134555
1111222	1111222	1112222	1111223	1111444	1124445	1114555
1111222	1111222	1112223	1111224	1111444	1111445	1112555
1111222	1111222	1111224	1111225	1111335	1111335	1111335
1112666	1111666	1111666	1114666	1456666	3566666	6666666
1111556	1111666	1111666	1116666	1556666	3566666	4666666
1111556	1111666	1111666	1116666	2356666	2266666	3666666
1111556	1111666	1111666	1116666	2356666	2266666	3666666
1111556	1111666	1111666	1116666	2356666	2266666	2666666

TABLE (7.25)

DATA FILE NAME= III 4

SEPERATION RELATIVE SCALINGS

1.00 0.80 0.65 0.50 0.30 0.10 0.00

VELOCITY ERROR RELATIVE SCALINGS

1.00 0.80 0.65 0.50 0.30 0.10 0.00

SPEED RELATIVE SCALING

1.00 0.75 0.50 0.25 0.00

#	SEPERATION	BRAKE LIGHTS	VELOCITY ERROR	SPEED	INPUT	
1	16	ON	-6	20	FULL BRAKE	1
2	16	ON	6	20	FULL BRAKE	1
3	30	ON	-6	20	FULL BRAKE	1
4	30	ON	6	20	NO POWER	4
5	16	ON	-6	90	FULL BRAKE	1
6	16	ON	6	90	BRAKE	2
7	30	ON	-6	90	FULL BRAKE	1
8	30	ON	6	90	BRAKE	2
9	16	OFF	-6	20	FULL BRAKE	1
10	16	OFF	6	20	DECELERATE	3
11	30	OFF	-6	20	DECELERATE	3
12	30	OFF	6	20	FULL POWER	6
13	16	OFF	-6	90	FULL BRAKE	1
14	16	OFF	6	90	BRAKE	2
15	30	OFF	-6	90	BRAKE	2
16	30	OFF	6	90	FULL POWER	6

## COEFFICIENTS OF THE WORTH FUNCTION

6.0						
-4.0	-4.0	-4.0	0.0			
4.0	3.0	1.0	3.0	2.0	1.0	
-3.0	-4.0	-2.0	-3.0			
5.0						

## STATE ACTION MATRIX 11

1111111	1111122	1112222	1122223	1122233	1222334	1223344
1111111	1111122	1112222	1112222	1122233	1122333	1222334
1111112	1112222	1112222	1112222	1122223	1122233	1122233
1111222	1111222	1112222	1112222	1112222	1112222	1122223
1112222	1112222	1112222	1112222	1112222	1112222	1112222
1122233	1223334	2233344	2333445	2334455	3344556	3445566
1122233	1222333	2223344	2233444	2334455	3344556	3344566
1122223	1222333	2223344	2233444	2334455	2344556	3344566
1112222	1222233	1223334	2233344	2233455	2334556	2344566
1112222	1222233	1222333	2223344	2233445	2334456	2334566

TABLE (7.26)

DATA FILE NAME= DI 5

SEPERATION RELATIVE SCALINGS

1.00 0.80 0.65 0.50 0.30 0.10 0.00

VELOCITY ERROR RELATIVE SCALINGS

1.00 0.80 0.65 0.50 0.30 0.10 0.00

SPEED RELATIVE SCALING

1.00 0.75 0.50 0.25 0.00

#	SEPERATION	BRAKE LIGHTS	VELOCITY ERROR	SPEED	INPUT
1	16	ON	-6	20	FULLBRAKE 1
2	16	ON	6	20	FULLBRAKE 1
3	30	ON	-6	20	FULLBRAKE 1
4	30	ON	6	20	NO POWER 4
5	16	ON	-6	90	FULLBRAKE 1
6	16	ON	6	90	BRAKE 2
7	30	ON	-6	90	FULLBRAKE 1
8	30	ON	6	90	BRAKE 2
9	16	OFF	-6	20	FULLBRAKE 1
10	16	OFF	6	20	DECELERATE 3
11	30	OFF	-6	20	NO POWER 4
12	30	OFF	6	20	FULLPOWER 6
13	16	OFF	-6	90	FULLBRAKE 1
14	16	OFF	6	90	BRAKE 2
15	30	OFF	-6	90	NO POWER 4
16	30	OFF	6	90	FULLPOWER 6

## COEFFICIENTS OF THE WORTH FUNCTION

6.0						
-4.0	-4.0	-2.0	0.0			
4.0	1.0	1.0	1.0	2.0	0.0	
-1.0	-4.0	-1.0	-2.0			
4.0						

## STATE ACTION MATRIX 12

1111111	1111122	1112222	1122223	1122233	1222334	1223344
1111111	1111122	1112222	1112222	1122233	1122333	1222334
1111112	1111222	1112222	1112222	1122223	1122233	1122233
1111222	1111222	1112222	1112222	1112222	1112222	1122223
1112222	1112222	1112222	1112222	1112222	1112222	1112222

1122233	2223334	2233344	3334445	3444555	4445566	4455566
1122233	2223333	2233344	3333444	3344455	4445556	4455566
1122223	2222333	2233344	3333444	3344455	4445556	4455566
1112222	2222333	2233334	3333444	3344455	4445556	4455566
1112222	2222233	2233333	3333444	3344455	4445556	4455566

TABLE (7.27)

DATA FILE NAME= DI 6

SEPERATION RELATIVE SCALINGS

1.00 0.80 0.65 0.50 0.30 0.10 0.00

VELOCITY ERROR RELATIVE SCALINGS

1.00 0.80 0.79 0.50 0.11 0.10 0.00

SPEED RELATIVE SCALING

1.00 0.75 0.50 0.25 0.00

#	SEPERATION	BRAKE LIGHTS	VELOCITY ERROR	SPEED	INPUT	
1	16	ON	-6	20	FULLBRAKE	1
2	16	ON	6	20	FULLBRAKE	1
3	30	ON	-6	20	FULLBRAKE	1
4	30	ON	6	20	NO POWER	4
5	16	ON	-6	90	FULLBRAKE	1
6	16	ON	6	90	BRAKE	2
7	30	ON	-6	90	FULLBRAKE	1
8	30	ON	6	90	BRAKE	2
9	16	OFF	-6	20	FULLBRAKE	1
10	16	OFF	6	20	DECELERATE	3
11	30	OFF	-6	20	NO POWER	4
12	30	OFF	6	20	FULLPOWER	6
13	16	OFF	-6	90	FULLBRAKE	1
14	16	OFF	6	90	BRAKE	2
15	30	OFF	-6	90	NO POWER	4
16	30	OFF	6	90	FULLPOWER	6

COEFFICIENTS OF THE WORTH FUNCTION

6.0						
-4.0	-4.0	-2.0	0.0			
4.0	1.0	1.0	1.0	2.0	0.0	
-1.0	-4.0	-1.0	-2.0			
4.0						

STATE ACTION MATRIX 13

111111	1111222	1112222	1112223	1112333	1222334	1223444
1111111	1111222	1112222	1112222	1112333	1112333	1222334
1111112	1111222	1112222	1112222	1112223	1112333	1112333
1111222	1111222	1112222	1112222	1112222	1112222	1112223
1112222	1112222	1112222	1112222	1112222	1112222	1112222
1112333	2223334	2223444	3334445	3444555	4445556	4445666
1112333	2223333	2223444	3333444	3344555	4445556	4445666
1112223	2222333	2223444	3333444	3334555	4445556	4445666
1112222	2222333	2223334	3333444	3334555	4445556	4445666
1112222	2222333	2223333	3333444	3334555	4445556	4445666

TABLE (7.28)

DATA FILE NAME= DI 7

SEPERATION RELATIVE SCALINGS

1.00 0.80 0.65 0.50 0.30 0.10 0.00

VELOCITY ERROR RELATIVE SCALINGS

1.00 0.80 0.79 0.50 0.11 0.10 0.00

SPEED RELATIVE SCALING

1.00 0.75 0.50 0.25 0.00

#	SEPERATION	BRAKE LIGHTS	VELOCITY ERROR	SPEED	INPUT
1	16	ON	-6	20	FULLBRAKE 1
2	16	ON	-6	20	FULLBRAKE 1
3	30	ON	-6	20	FULLBRAKE 1
4	30	ON	6	20	NO POWER 4
5	16	ON	-6	90	FULLBRAKE 1
6	16	ON	6	90	BRAKE 2
7	30	ON	-6	90	FULLBRAKE 1
8	30	ON	6	90	BRAKE 2
9	16	OFF	-6	20	FULLBRAKE 1
10	16	OFF	6	20	FULLPOWER 6
11	30	OFF	-6	20	NO POWER 4
12	30	OFF	6	20	FULLPOWER 6
13	16	OFF	-6	90	FULLBRAKE 1
14	16	OFF	6	90	BRAKE 2
15	30	OFF	-6	90	NO POWER 4
16	30	OFF	6	90	FULLPOWER 6

COEFFICIENTS OF THE WORTH FUNCTION

6.0						
-4.0	-4.0	-2.0	0.0			
4.0	1.0	4.0	1.0	2.0	0.0	
-1.0	-7.0	-4.0	-2.0			
7.0						

STATE ACTION MATRIX 14

1111111	1111222	1112222	1112223	1112333	1222334	1223444
1111111	1111222	1112222	1112222	1112333	1112333	1222334
1111112	1111222	1112222	1112222	1112223	1112333	1112333
1111222	1111222	1112222	1112222	1112222	1112222	1112223
1112222	1112222	1112222	1112222	1112222	1112222	1112222

1224566	2234666	2334666	3334666	3445666	4445666	4445666
1223555	2223555	2334555	3334555	3445556	4445666	4445666
1223444	2223444	2333445	3334555	3444555	4445666	4445666
1112333	2223334	2223444	3334445	3444555	4445566	4445666
1112222	2222333	2223333	3333444	3334555	4445556	4445666

TABLE (7.29)

DATA FILE NAME= DT 8

SEPERATION RELATIVE SCALINGS

1.00 0.80 0.65 0.50 0.30 0.10 0.00

VELOCITY ERROR RELATIVE SCALINGS

1.00 0.80 0.79 0.50 0.11 0.10 0.00

SPEED RELATIVE SCALING

1.00 0.75 0.50 0.25 0.00

#	SEPERATION	BRAKE LIGHTS	VELOCITY ERROR	SPEED	INPUT	
1	16	ON	-6	20	FULLBRAKE	1
2	16	ON	6	20	FULLBRAKE	1
3	30	ON	-6	20	FULLBRAKE	1
4	30	ON	6	20	NO POWER	4
5	16	ON	-6	90	FULLBRAKE	1
6	16	ON	6	90	BRAKE	2
7	30	ON	-6	90	FULLBRAKE	1
8	30	ON	6	90	BRAKE	2
9	16	OFF	-6	20	FULLBRAKE	1
10	16	OFF	6	20	POWER	5
11	30	OFF	-6	20	NO POWER	4
12	30	OFF	6	20	FULLPOWER	6
13	16	OFF	-6	90	FULLBRAKE	1
14	16	OFF	6	90	DECELERATE	3
15	30	OFF	-6	90	NO POWER	4
16	30	OFF	6	90	FULLPOWER	6

COEFFICIENTS OF THE WORTH FUNCTION

6.0  
 -3.0      -4.0      -2.0      0.0  
 3.0      0.0      2.0      1.0      2.0      0.0  
 0.0      -5.0      -2.0      -2.0  
 5.0

STATE ACTION MATRIX 15

1111111	1111222	1112222	1112223	1112333	1222334	1223444
1111111	1111222	1112222	1112222	1112333	1112333	1222334
1111112	1111222	1112222	1112222	1112223	1112333	1112333
1111222	1111222	1112222	1112222	1112222	1112222	1112223
1112222	1112222	1112222	1112222	1112222	1112222	1112222
1223555	2223555	2334555	3334555	3444555	4445666	4445666
1223444	2223444	2334555	3334555	3444555	4445666	4445666
1223444	2223444	2334444	3334555	3444555	4445666	4445666
1222334	2223444	2333444	3334555	3444555	4445666	4445666
1112333	2223334	2223444	3334444	3444555	4445556	4445556

TABLE (7.30)

DATA FILE NAME= DI 9  
 SEPERATION RELATIVE SCALINGS  
 1.00 0.80 0.65 0.50 0.30 0.10 0.00  
 VELOCITY ERROR RELATIVE SCALINGS  
 1.00 0.95 0.79 0.50 0.11 0.10 0.00  
 SPEED RELATIVE SCALING  
 1.00 0.75 0.50 0.25 0.00

#	SEPERATION	BRAKE LIGHTS	VELOCITY ERROR	SPEED	INPUT
1	16	ON	-6	20	FULLBRAKE 1
2	16	ON	6	20	FULLBRAKE 1
3	30	ON	-6	20	FULLBRAKE 1
4	30	ON	6	20	NO POWER 4
5	16	ON	-6	90	FULLBRAKE 1
6	16	ON	6	90	BRAKE 2
7	30	ON	-6	90	FULLBRAKE 1
8	30	ON	6	90	BRAKE 2
9	16	OFF	-6	20	FULLBRAKE 1
10	16	OFF	6	20	FULLPOWER 6
11	30	OFF	-6	20	NO POWER 4
12	30	OFF	6	20	FULLPOWER 6
13	16	OFF	-6	90	FULLBRAKE 1
14	16	OFF	6	90	DECELERATE 3
15	30	OFF	-6	90	NO POWER 4
16	30	OFF	6	90	FULLPOWER 6

## COEFFICIENTS OF THE WORTH FUNCTION

6.0						
-3.0	-4.0	-2.0	0.0			
3.0	0.0	3.0	1.0	2.0	0.0	
0.0	-6.0	-3.0	-2.0			
6.0						

## STATE ACTION MATRIX 16

1111111	1111222	1112222	1112223	1112333	1122334	1123444
1111111	1111222	1112222	1112222	1112333	1112333	1122334
1111112	1111222	1112222	1112222	1112223	1112333	1112333
1111222	1111222	1112222	1112222	1112222	1112222	1112223
1112222	1112222	1112222	1112222	1112222	1112222	1112222

1124566	2234666	2234666	3334666	3345666	4445666	4445666
1123555	2224555	2234556	3334556	3344566	4445666	4445666
1123445	2223445	2234555	3334555	3344556	4445666	4445666
1122334	2223444	2233445	3334555	3344555	4445666	4445666
1112333	2223334	2223444	3334445	3344555	4445566	4445666

TABLE (7.31)

DATA FILE NAME= DI10

SEPERATION RELATIVE SCALINGS

1.00 0.80 0.65 0.50 0.30 0.10 0.00

VELOCITY ERROR RELATIVE SCALINGS

1.00 0.80 0.79 0.50 0.11 0.10 0.00

SPEED RELATIVE SCALING

1.00 0.75 0.50 0.25 0.00

#	SEPERATION	BRAKE LIGHTS	VELOCITY ERROR	SPEED	INPUT	
1	16	ON	-6	20	FULLBRAKE	1
2	16	ON	6	20	FULLBRAKE	1
3	30	ON	-6	20	FULLBRAKE	1
4	30	ON	6	20	NO POWER	4
5	16	ON	-6	90	FULLBRAKE	1
6	16	ON	6	90	BRAKE	2
7	30	ON	-6	90	FULLBRAKE	1
8	30	ON	6	90	BRAKE	2
9	16	OFF	-6	20	FULLBRAKE	1
10	16	OFF	6	20	POWER	5
11	30	OFF	-6	20	NO POWER	4
12	30	OFF	6	20	FULLPOWER	6
13	16	OFF	-6	90	FULLBRAKE	1
14	16	OFF	6	90	DECELERATE	3
15	30	OFF	-6	90	NO POWER	4
16	30	OFF	6	90	FULLPOWER	6

COEFFICIENTS OF THE WORTH FUNCTION

6.0						
-3.0	-4.0	-2.0	0.0			
3.0	0.0	2.0	1.0	2.0	0.0	
0.0	-5.0	-2.0	-2.0			
5.0						

STATE ACTION MATRIX 17

1111111	1111222	1112222	1112223	1112333	1222334	1223444
1111111	1111222	1112222	1112222	1112333	1112333	1222334
1111112	1111222	1112222	1112222	1112223	1112333	1112333
1111222	1111222	1112222	1112222	1112222	1112222	1112223
1112222	1112222	1112222	1112222	1112222	1112222	1112222
1223555	2223555	2334555	3334555	3444555	4445666	4445666
1223445	2223445	2334555	3334555	3444555	4445666	4445666
1223444	2223444	2333445	3334555	3444555	4445665	4445666
1222334	2223444	2333444	3334555	3444555	4445666	4445666
1112333	2223334	2223444	3334445	3444555	4445556	4445666

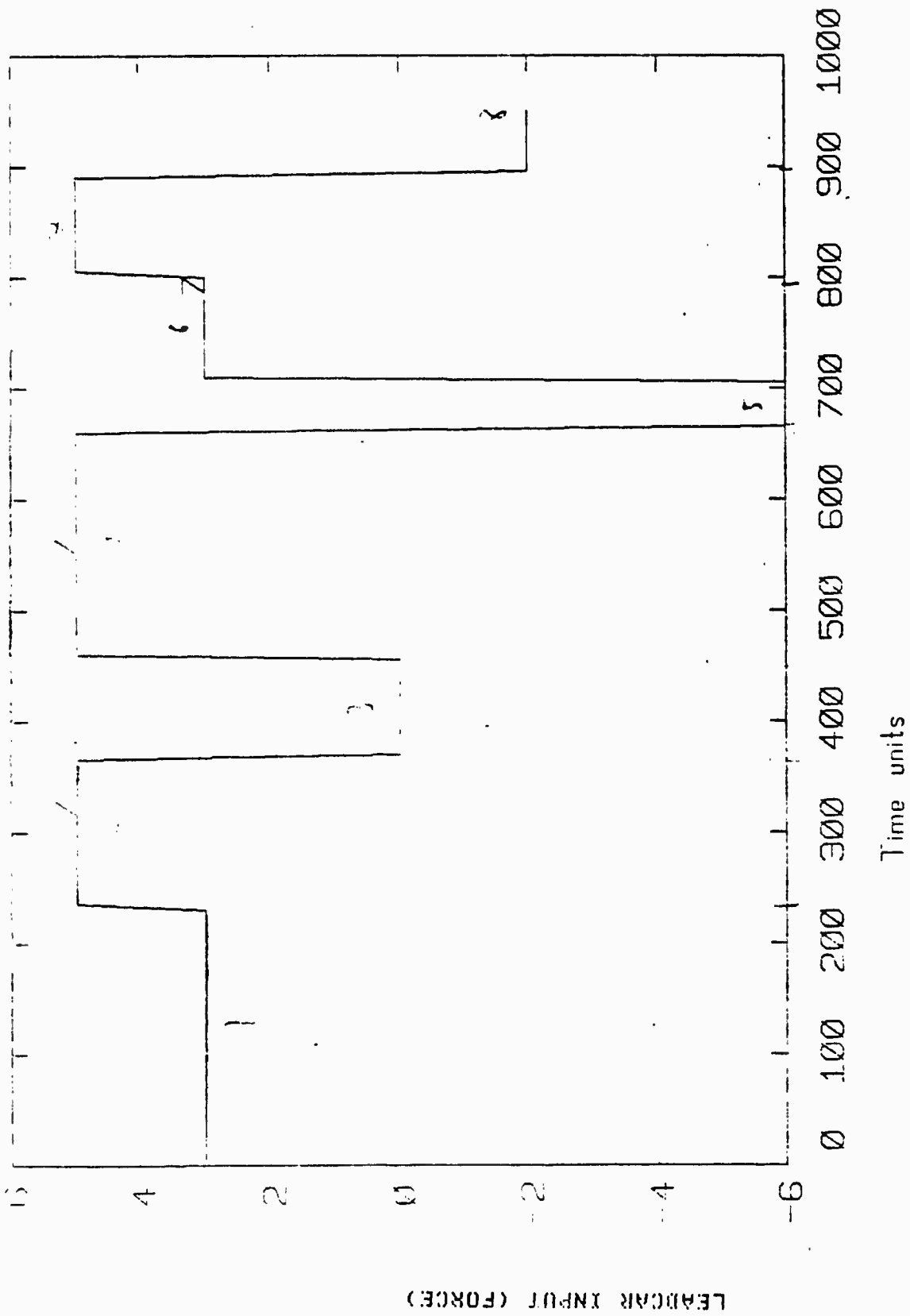


FIGURE (7.1) DISTURBANCE INPUT TO LEADCAR

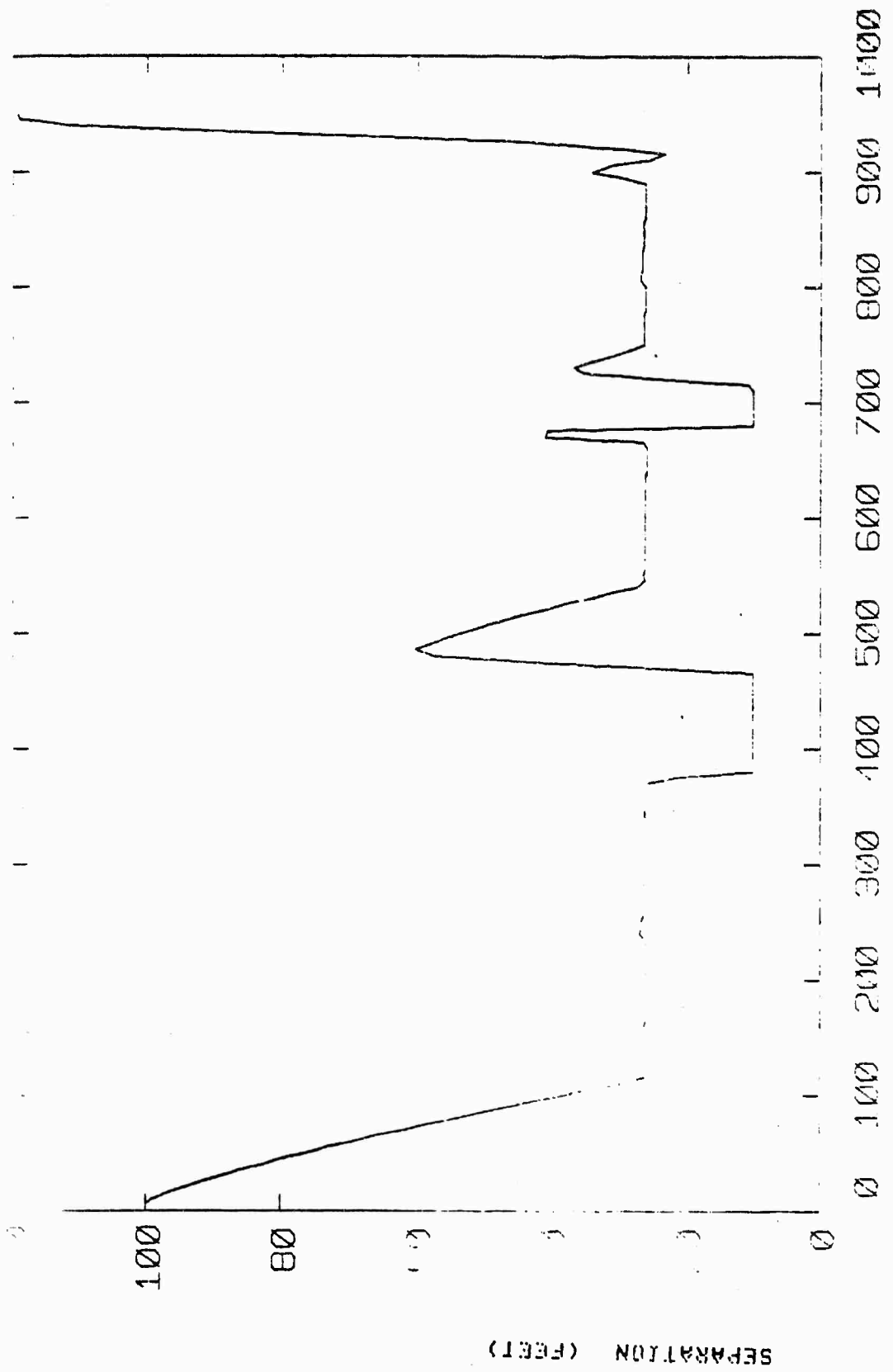


FIGURE (7.2) RESPONSE OF STATE ACTION MATRIX 1

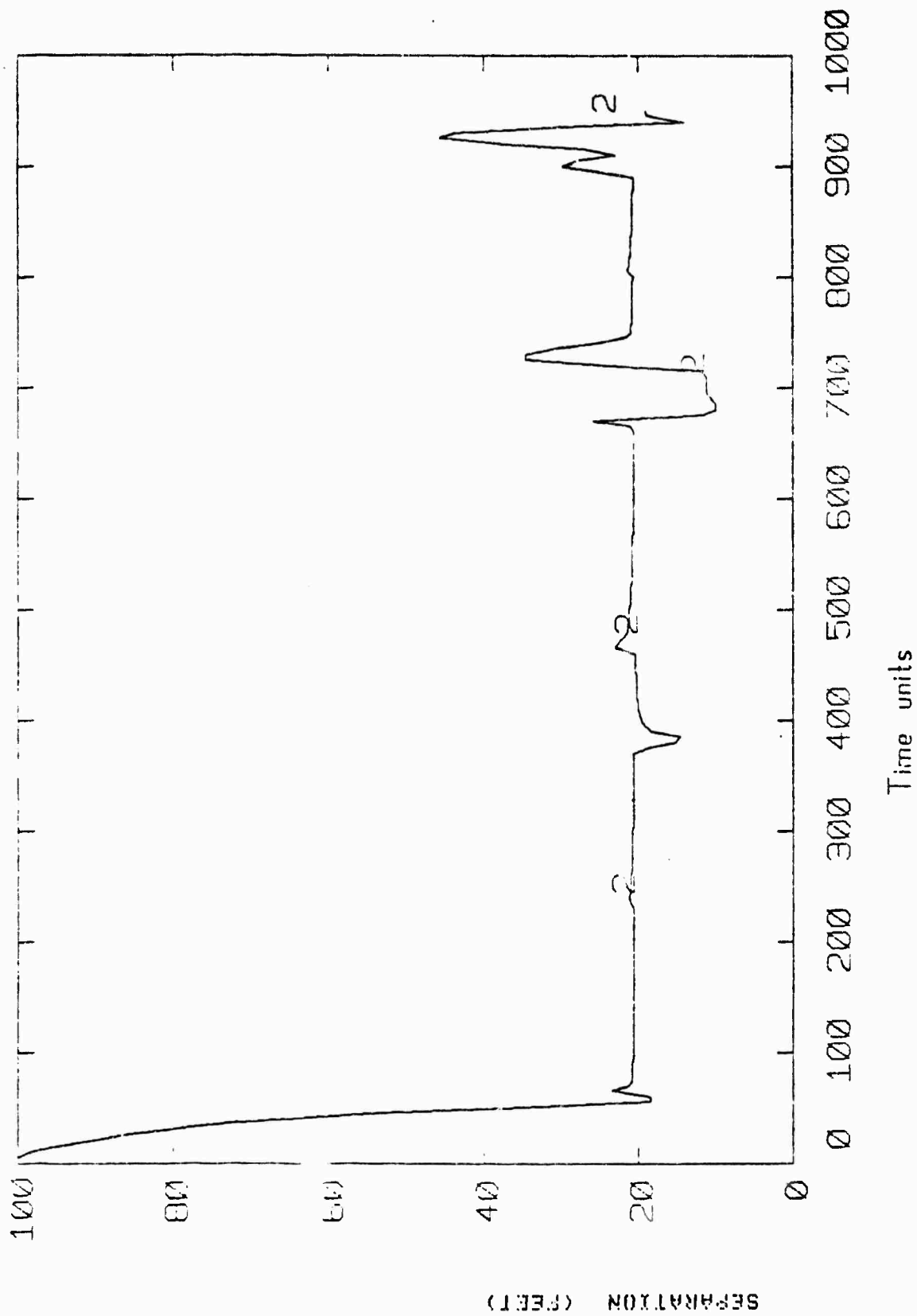


FIGURE (7.3) RESPONSE OF STATE ACTION MATRIX 2

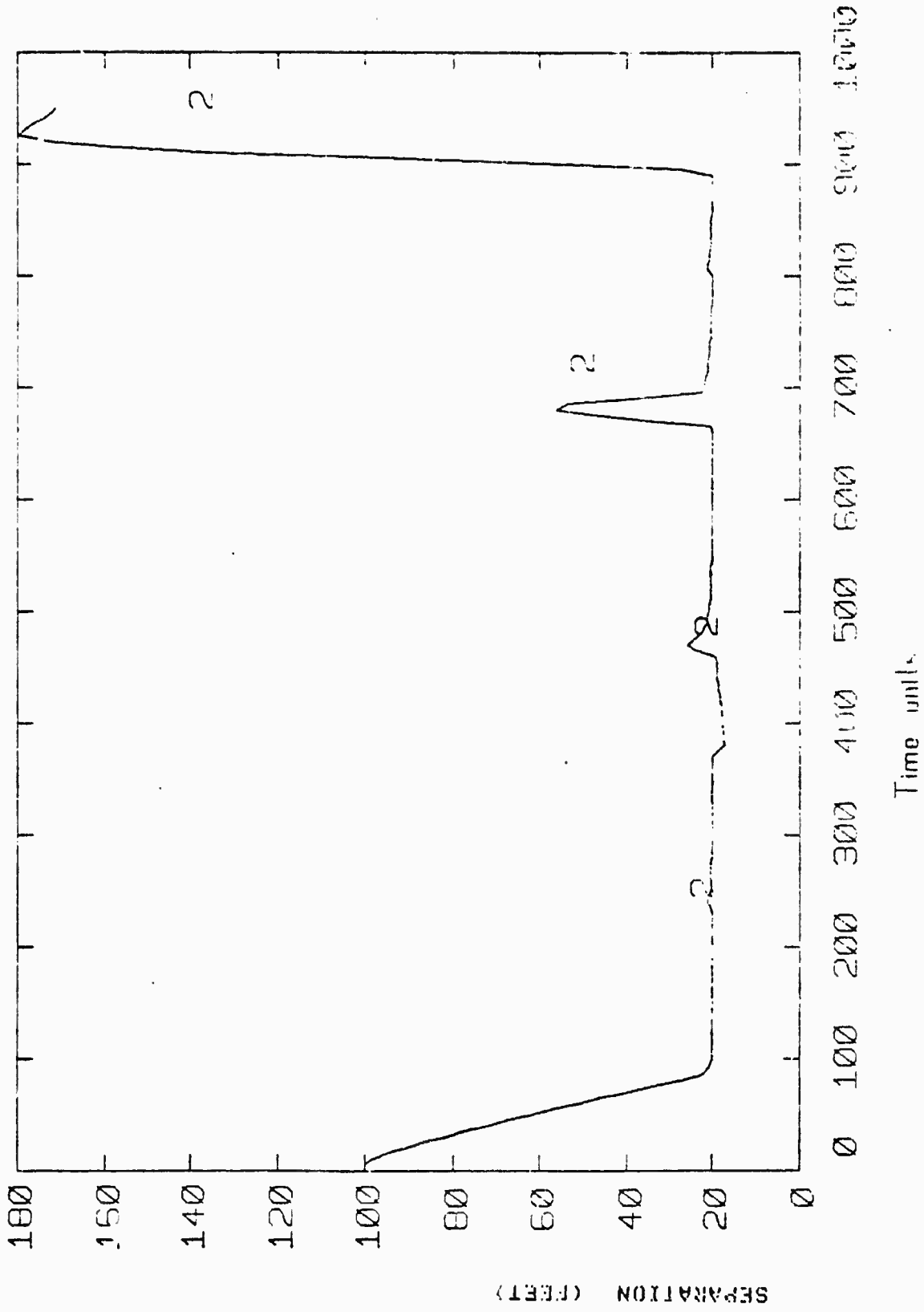


FIGURE (7.4) RESPONSE OF STATE ACTION MATRIX 3

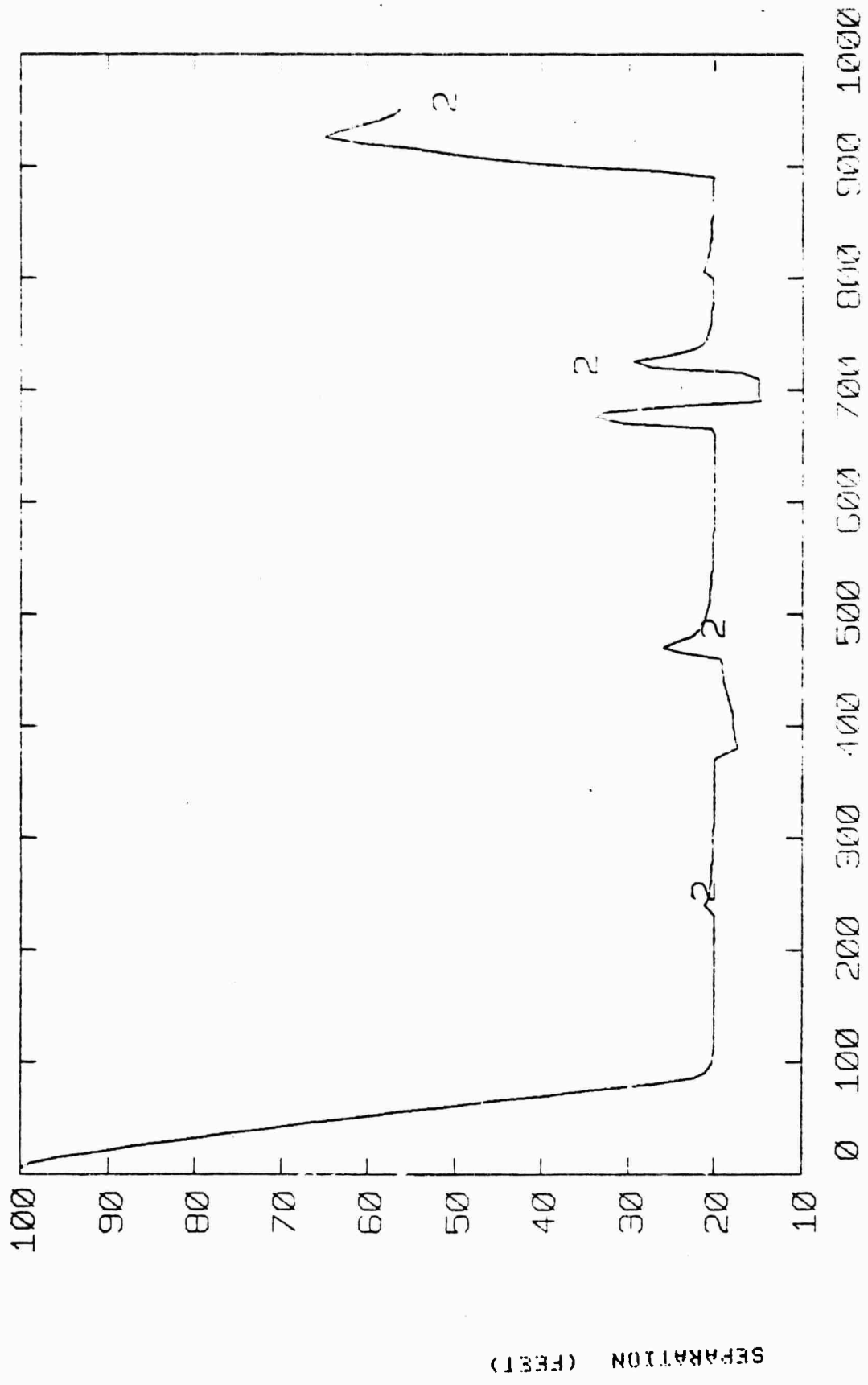


FIGURE (7.5) RESPONSE OF STATE ACTION MATRIX 4

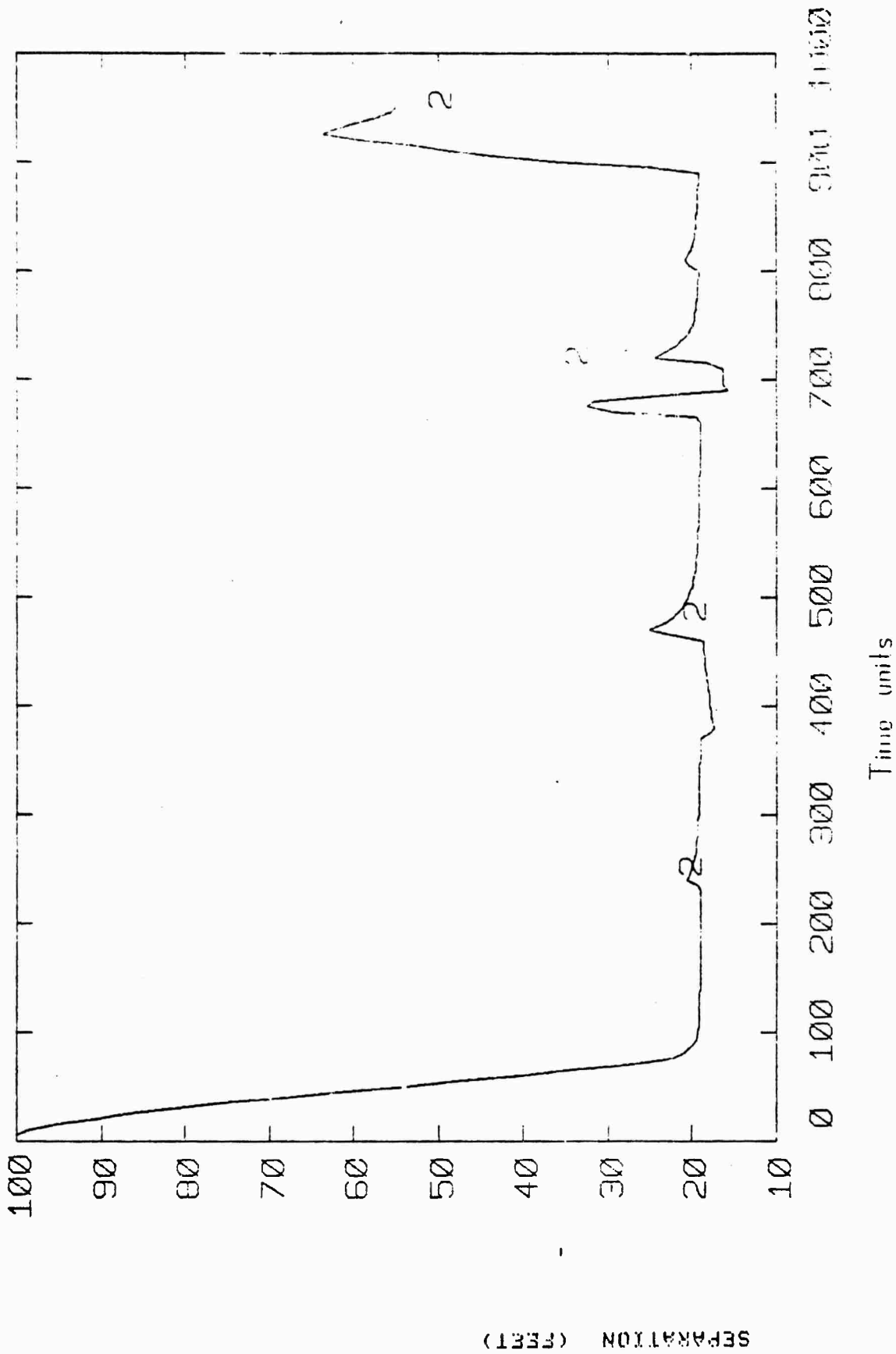


FIGURE (7.6) RESPONSE OF STATE ACTION MATRIX 5

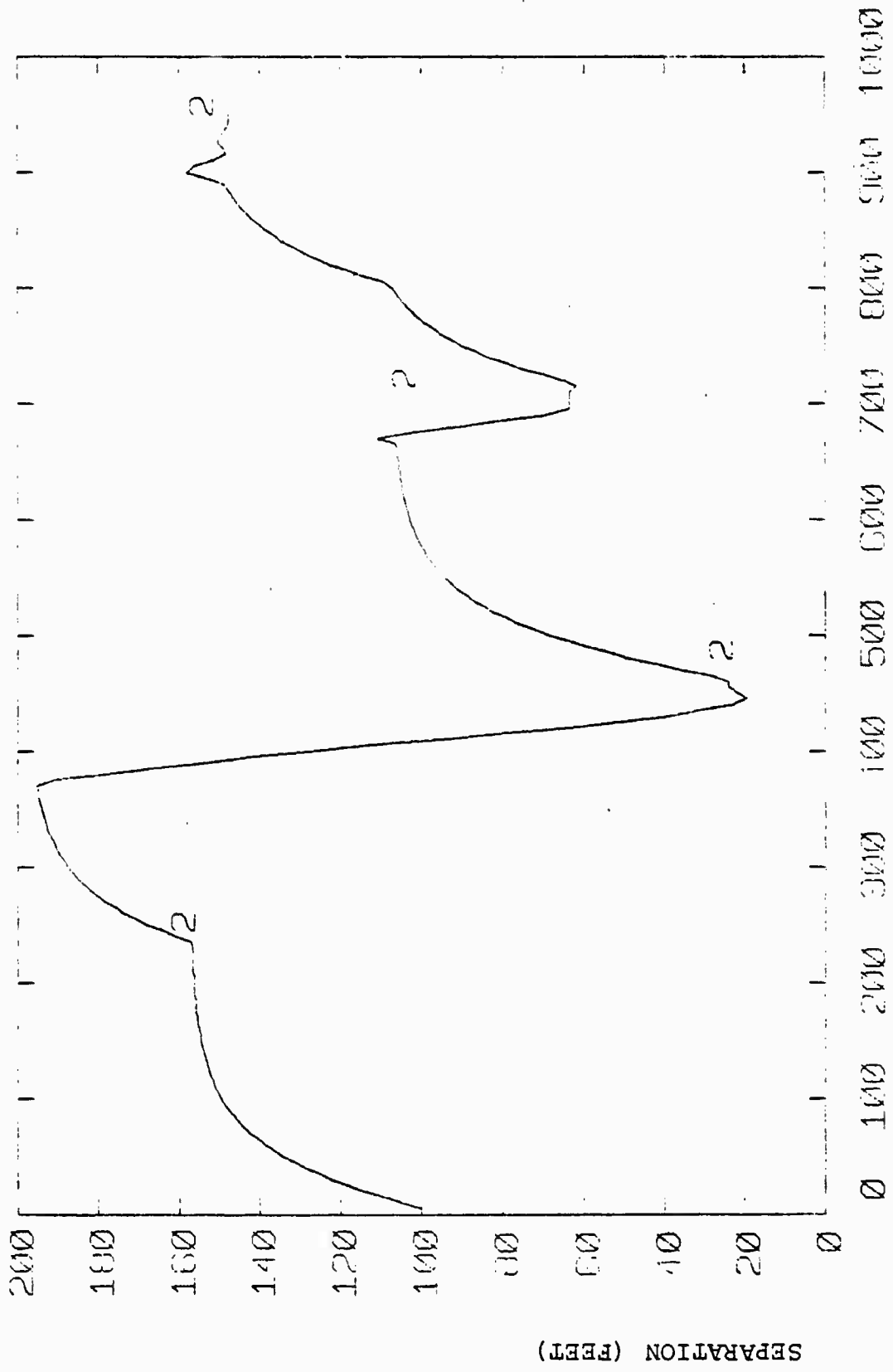


FIGURE 7.7 RESPONSE OF STATE ACTION MATRIX II

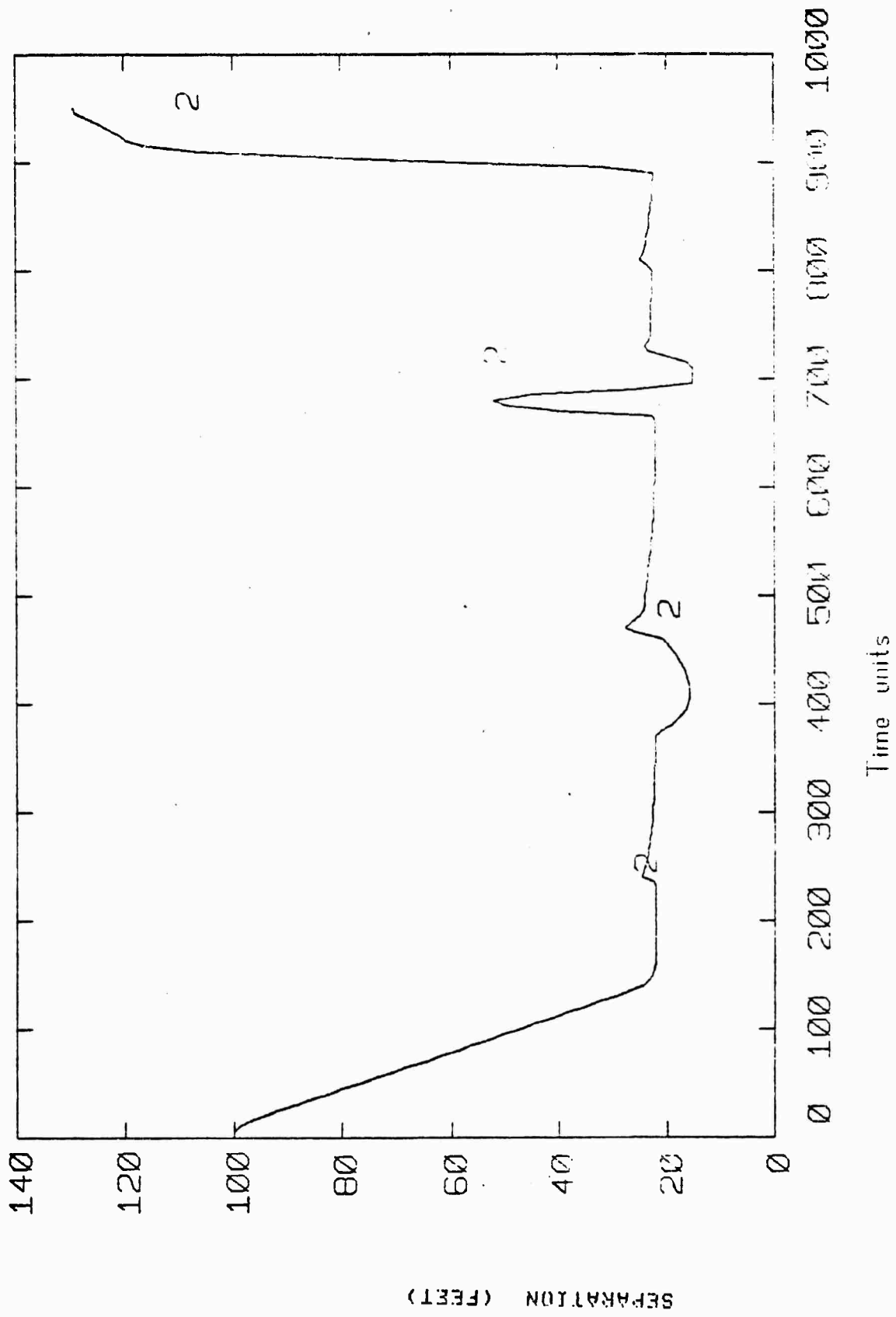


FIGURE (7.8) RESPONSE OF STATE ACTION MATRIX 7

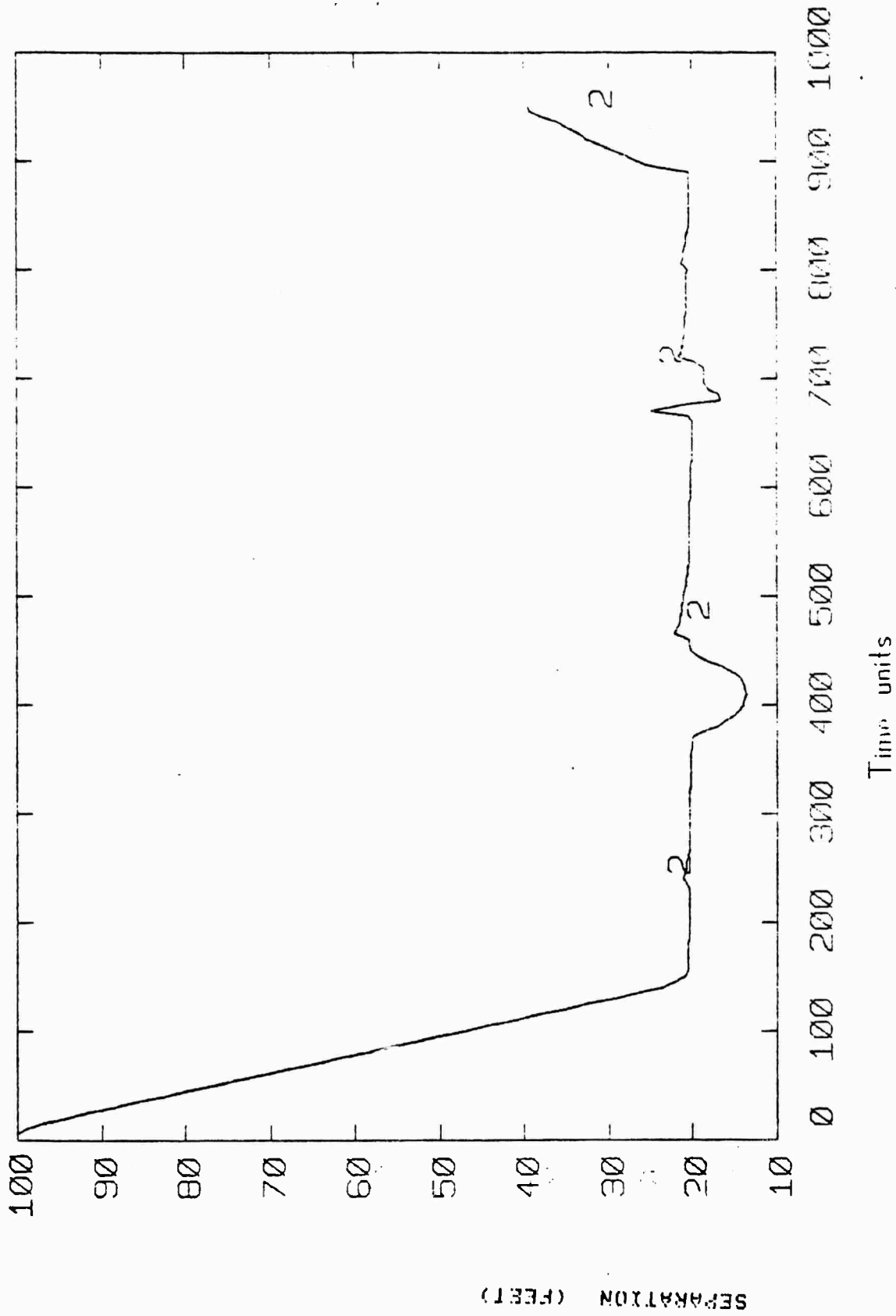


FIGURE (7.9) RESPONSE OF STATE ACTION MATRIX B

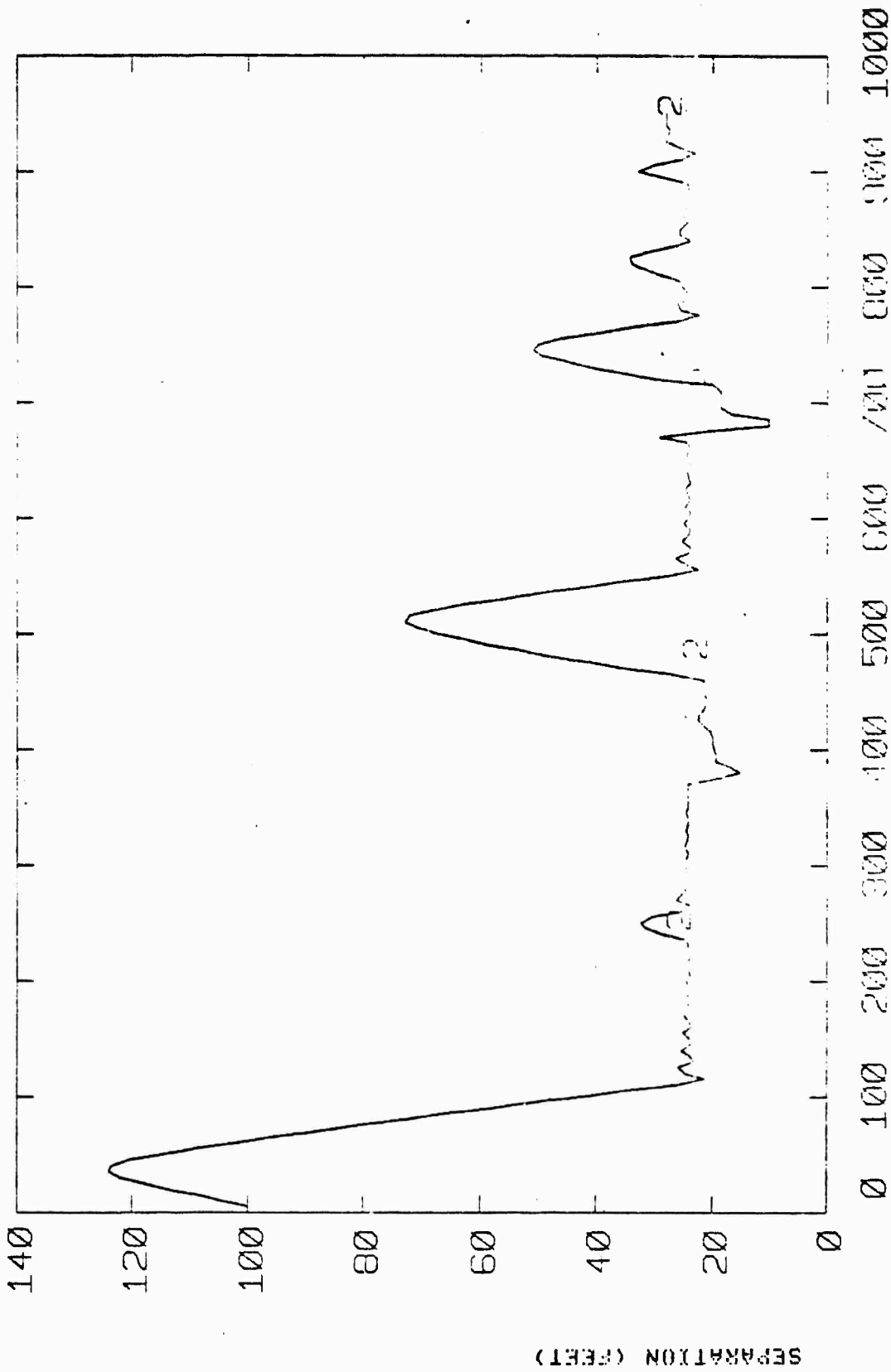


FIGURE (7.10) RESPONSE OF STATE ACTION MATRIX 12

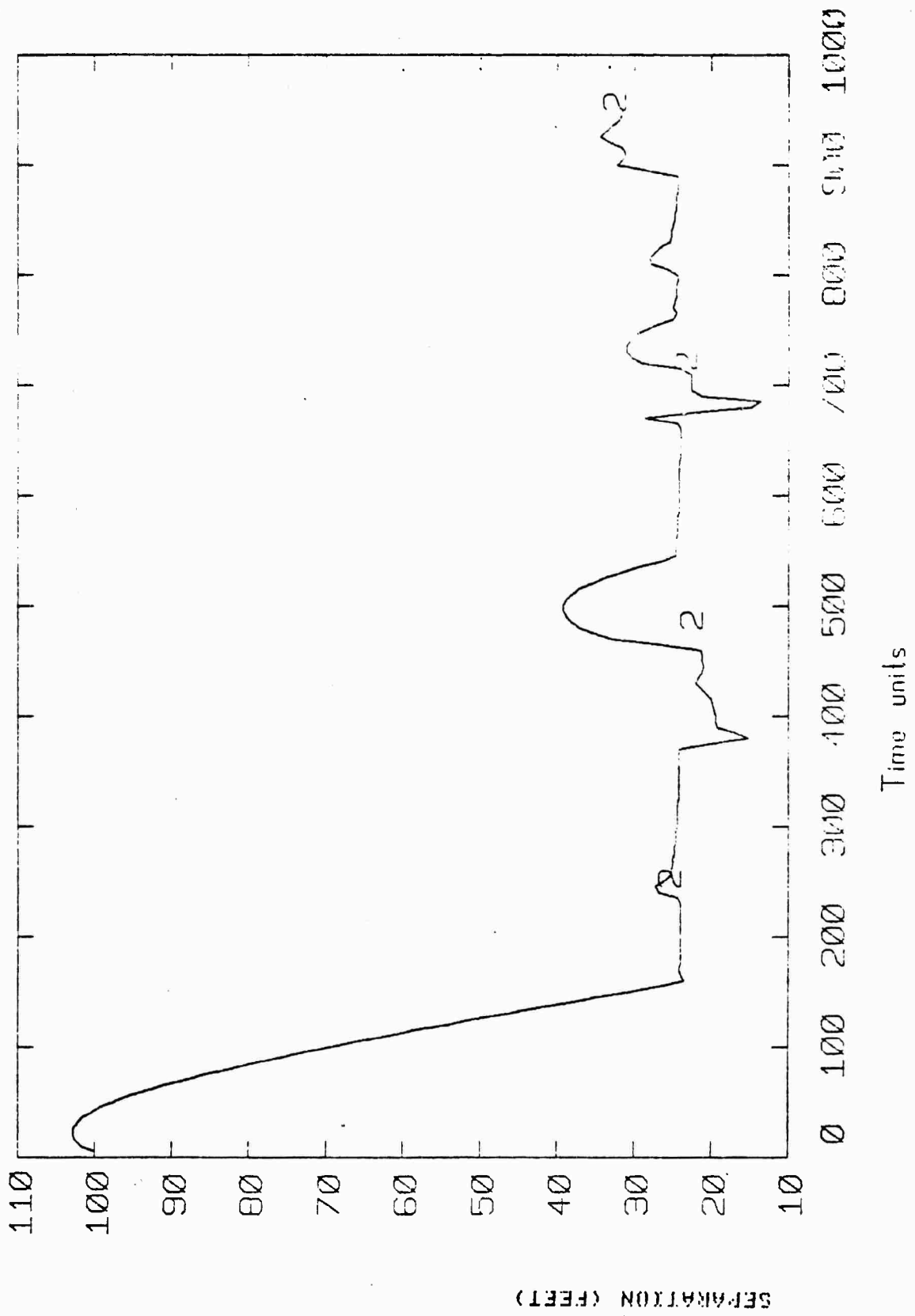


FIGURE (7.11) RESPONSE OF STATE ACTION MATRIX 13

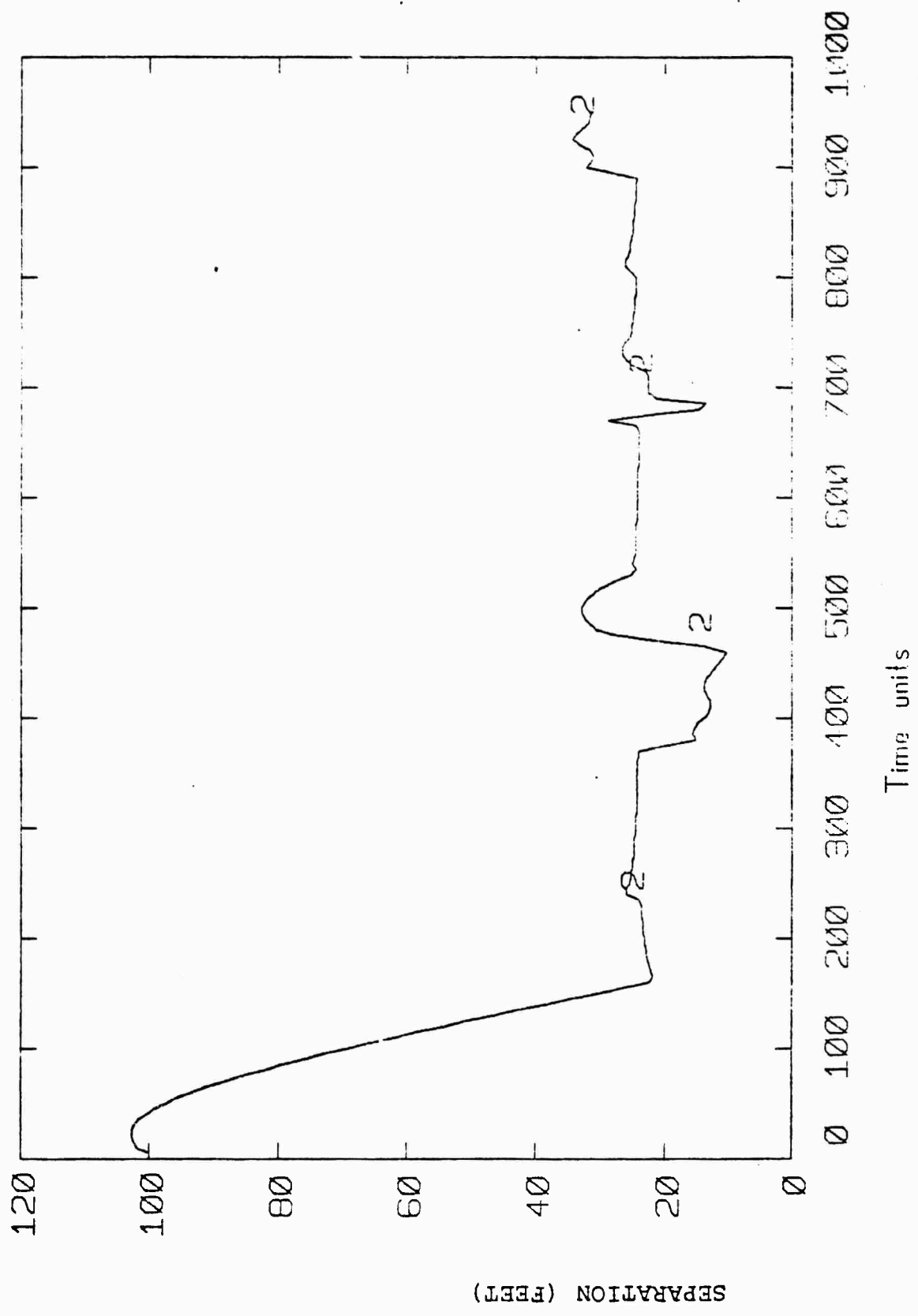


FIGURE 7.12 RESPONSE OF STATE ACTION MATRIX 14

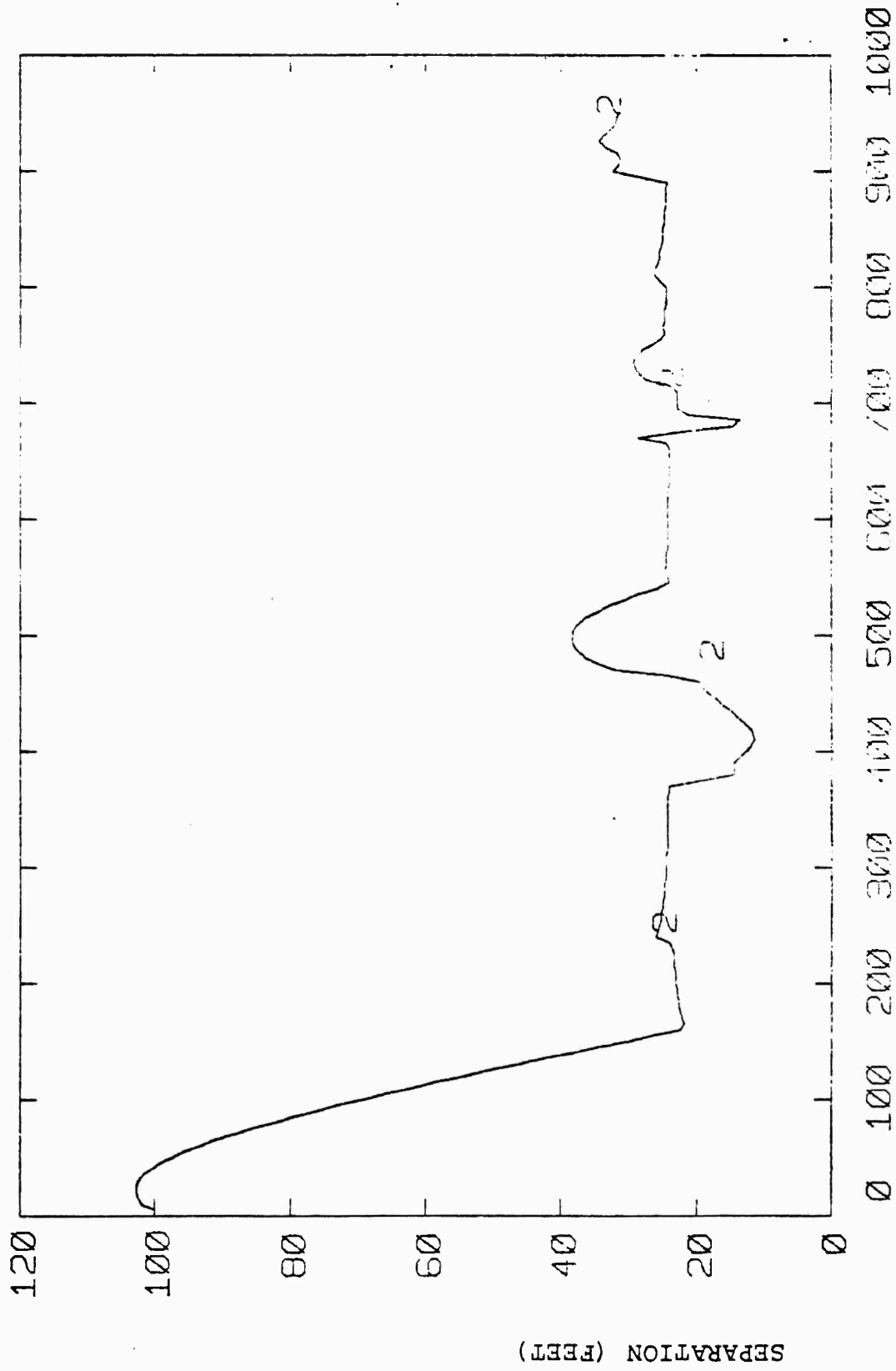


FIGURE 7.13 RESPONSE OF STATE ACTION MATRIX 15

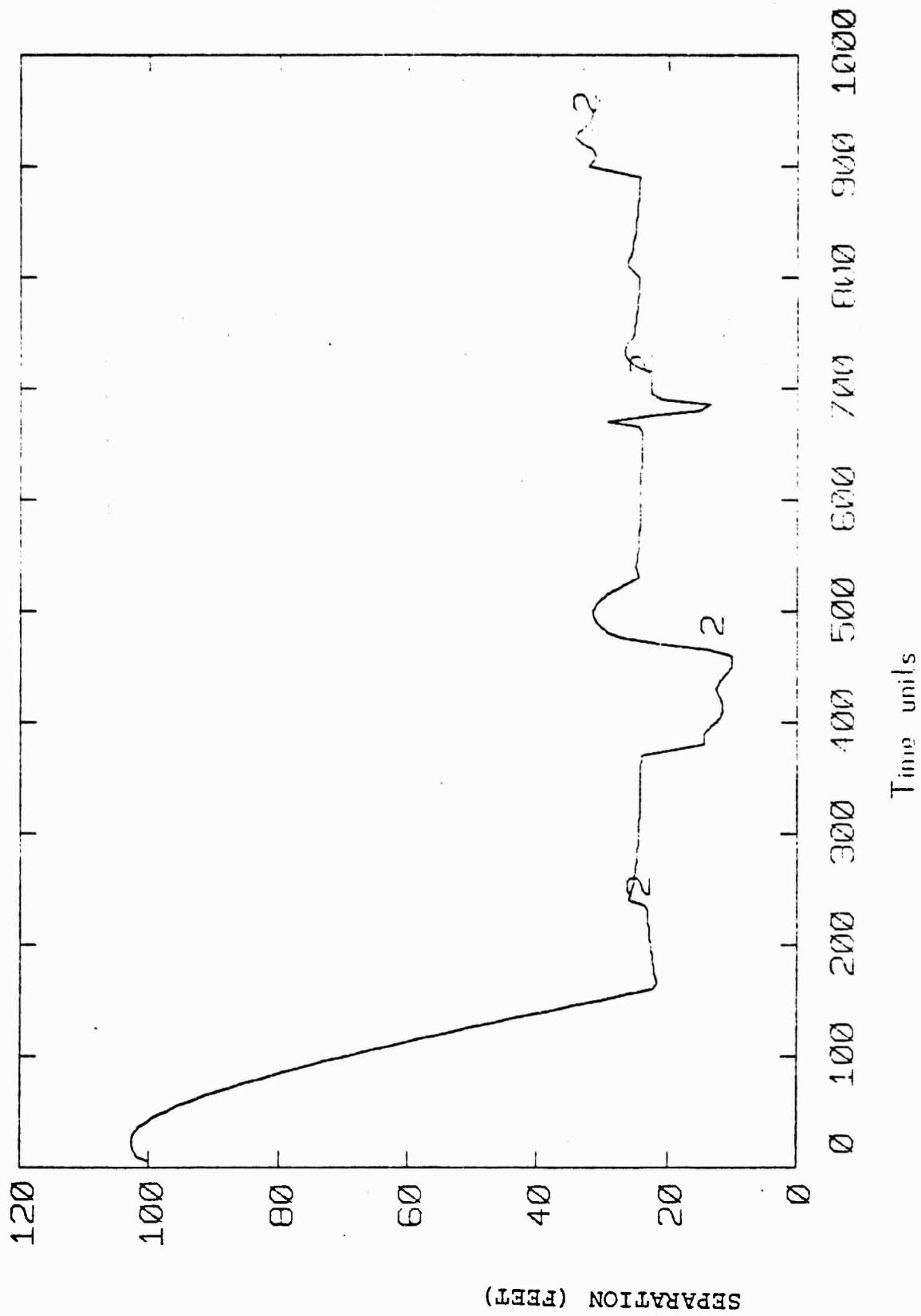


FIGURE 7.14 RESPONSE OF STATE ACTION MATRIX 16

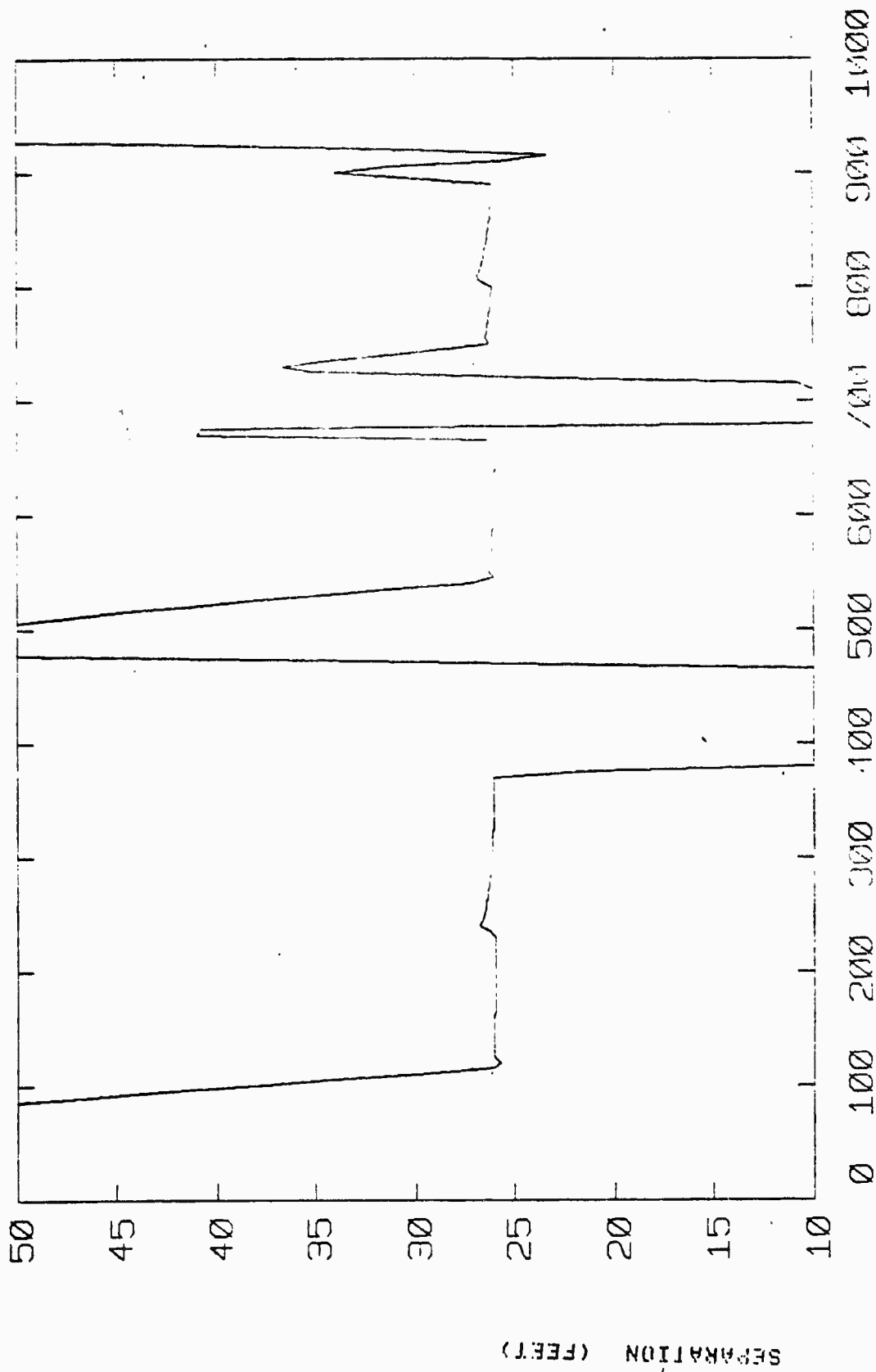


FIGURE (7.15) RESPONSE OF STATE ACTION MATRIX J

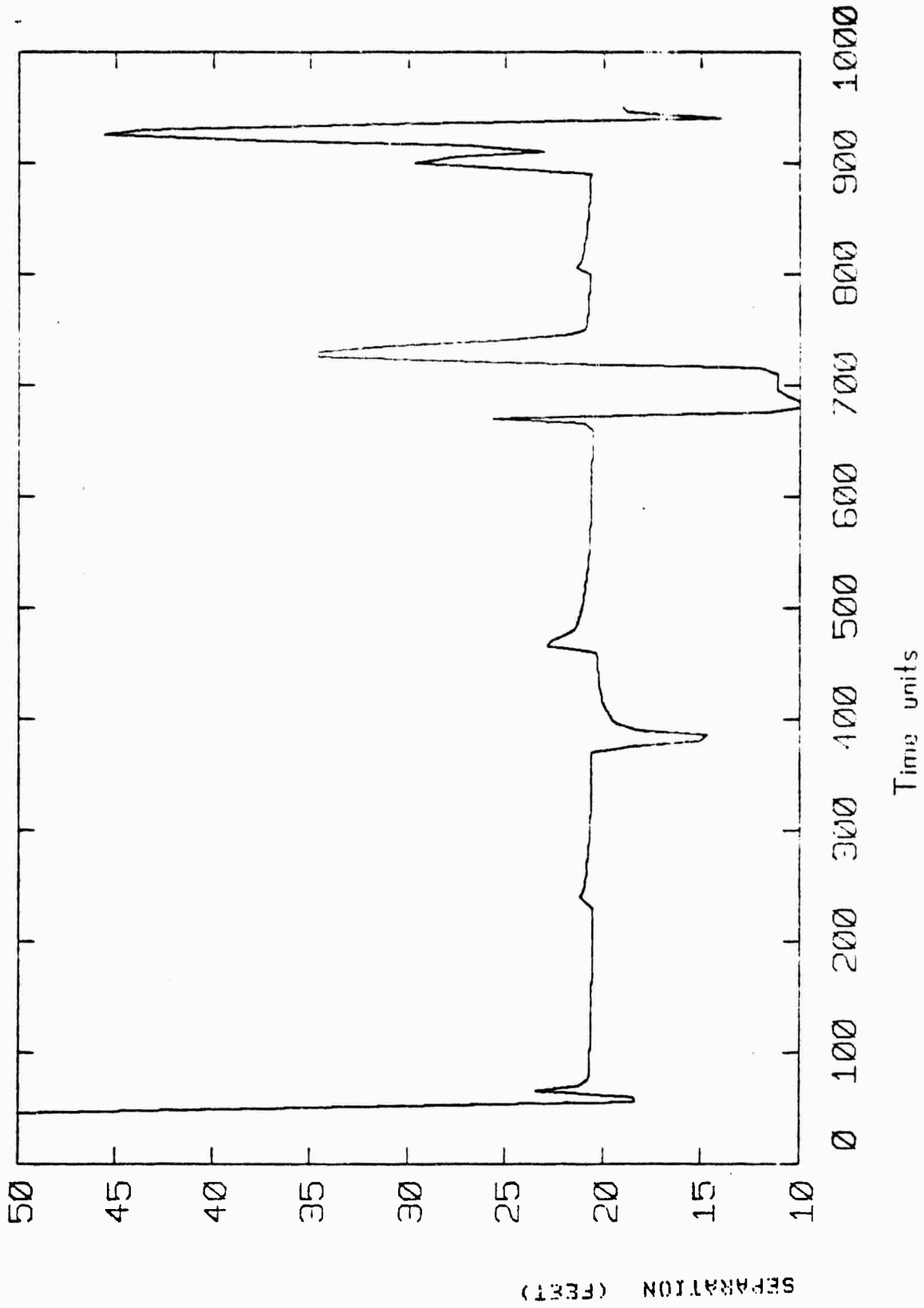


FIGURE (7.16) RESPONSE OF STATE ACTION MATRIX 2

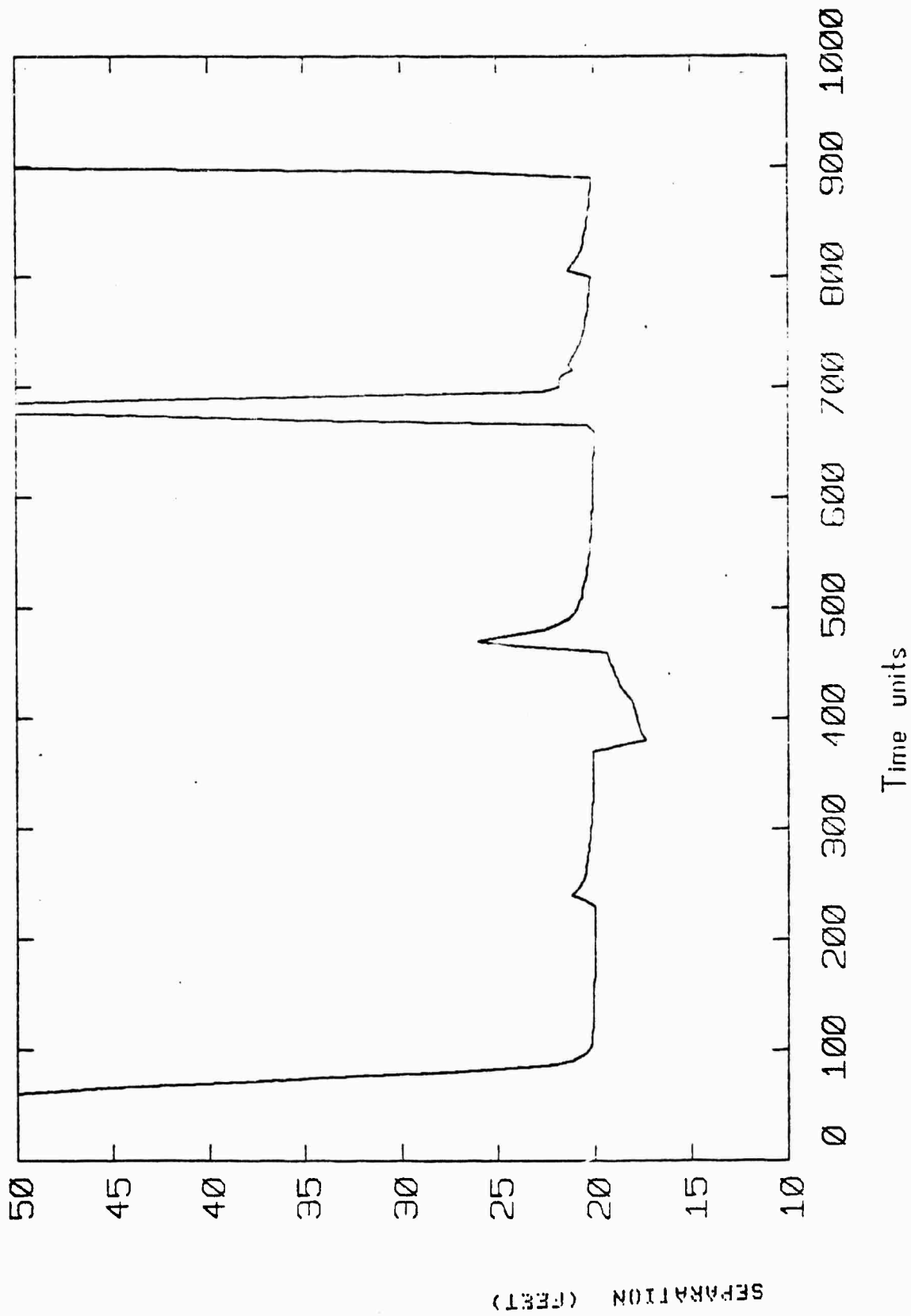


FIGURE (7.17) RESPONSE OF STATE ACTION MATRIX 3

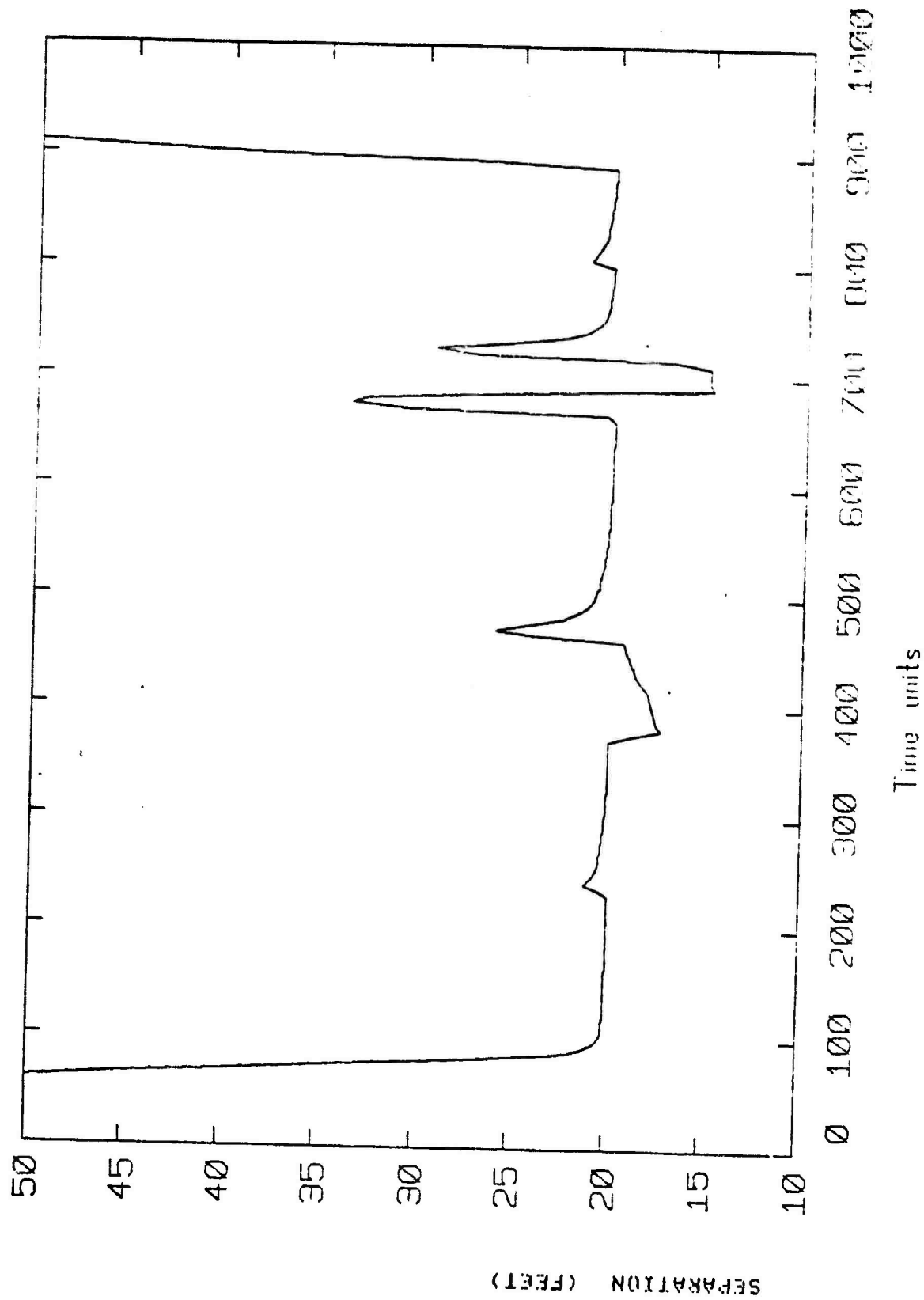


FIGURE (7.10) RESPONSE OF SDOF SYSTEM FOR CASE 9

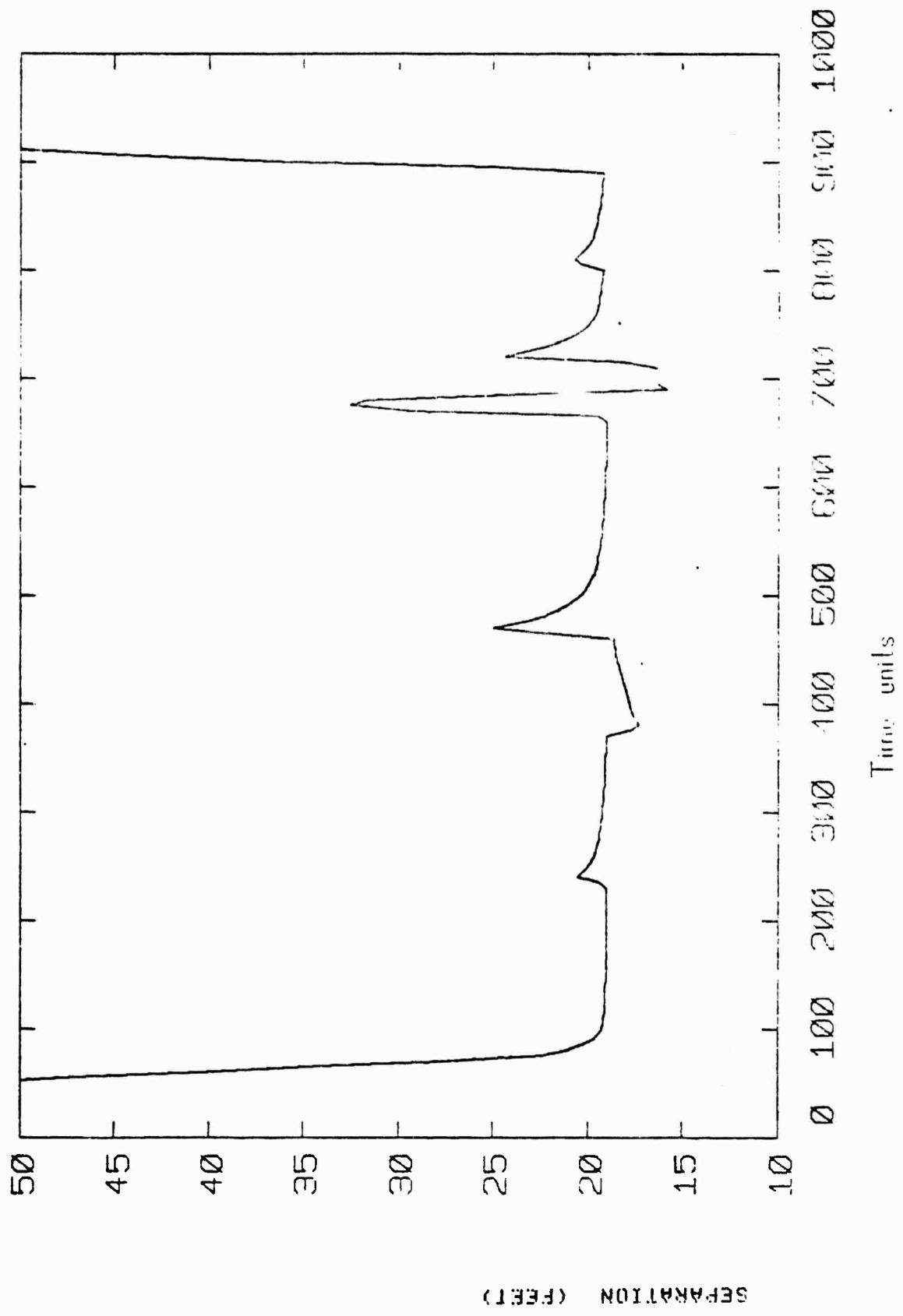


FIGURE (7.19) RESPONSE OF STATE ACTION MATRIX 5

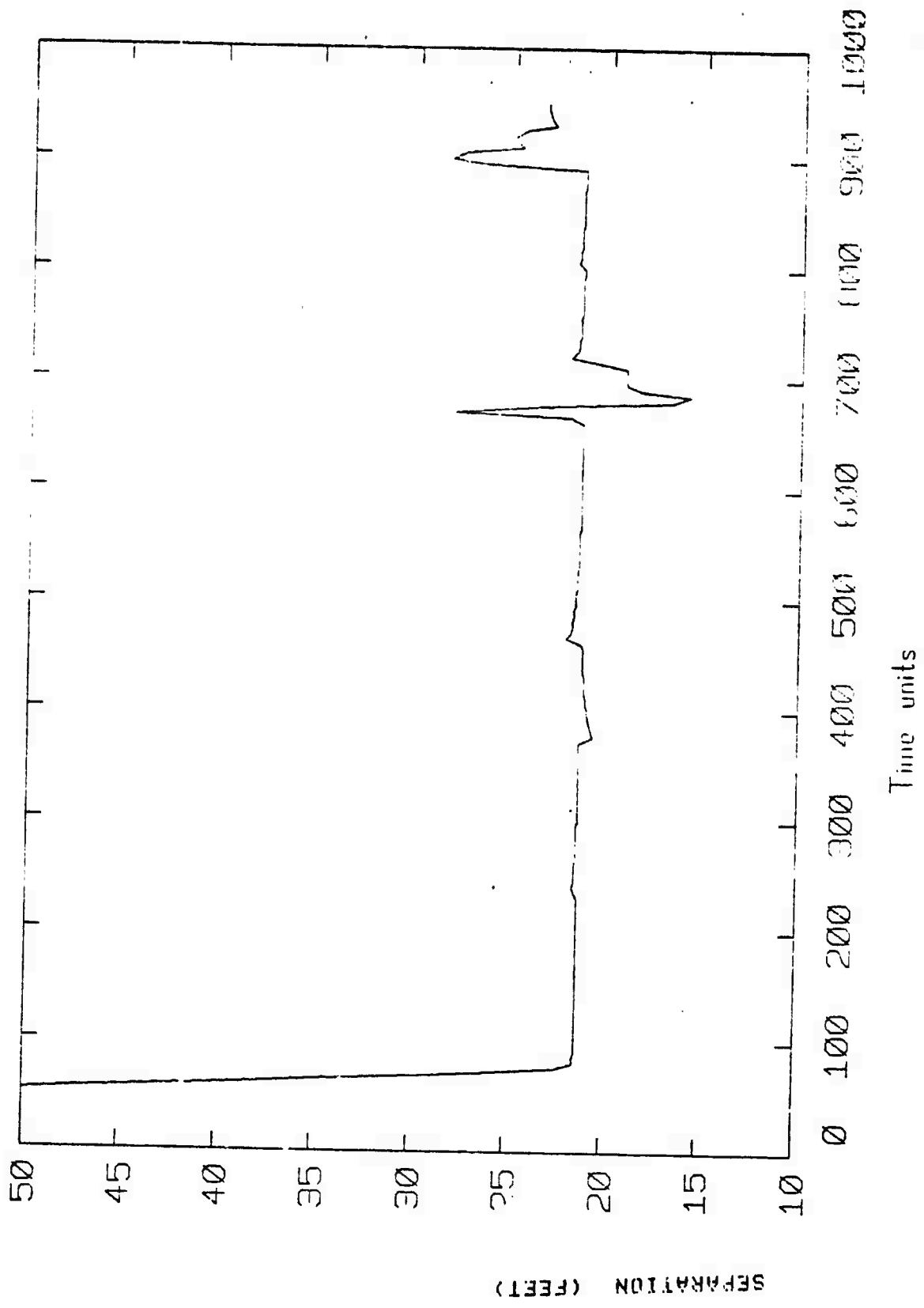


FIGURE (7.20) RESPONSE OF STATE ACTION MATRIX 6

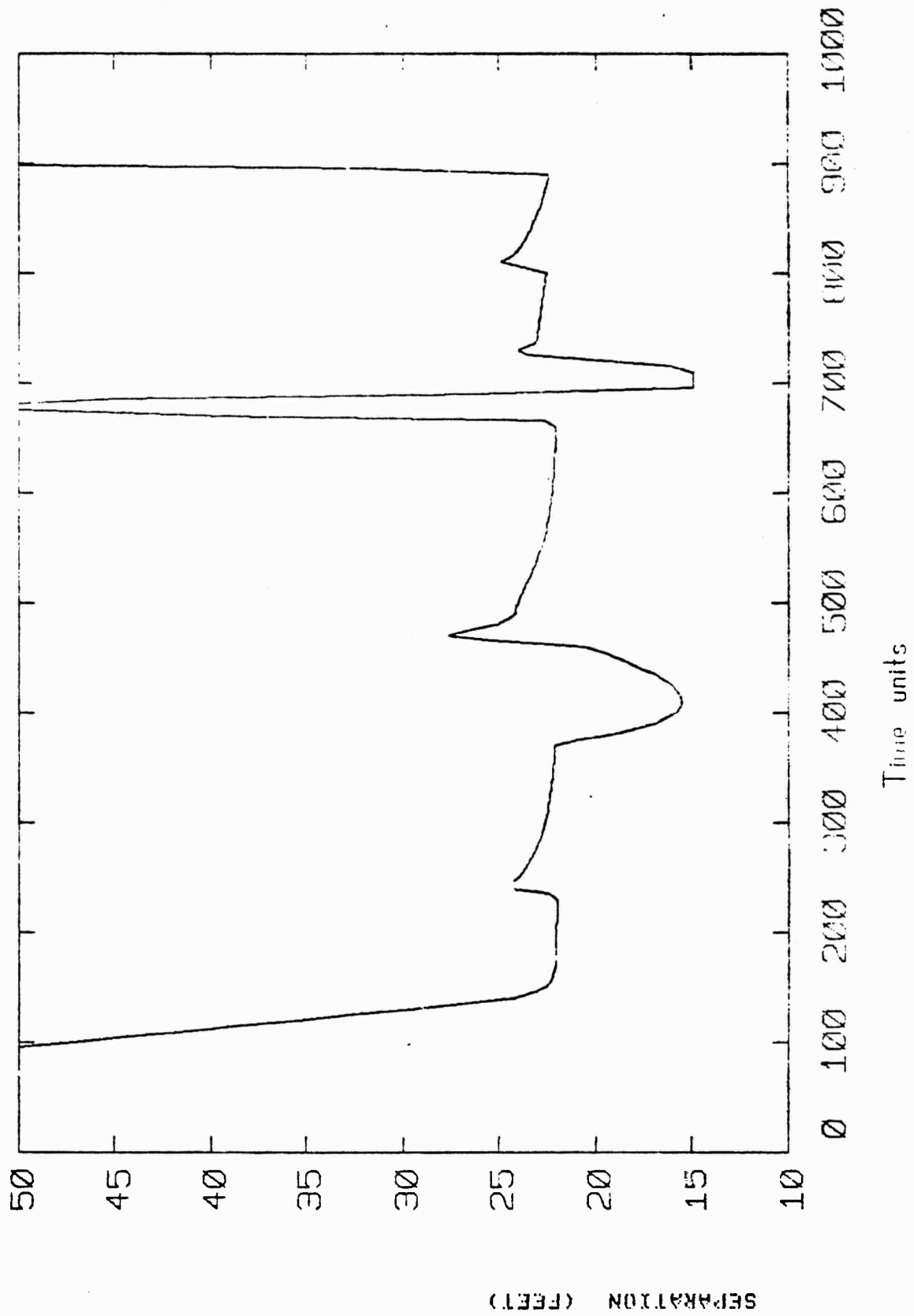


FIGURE (7.21) RESPONSE OF STATE ACTION MATRIX 7

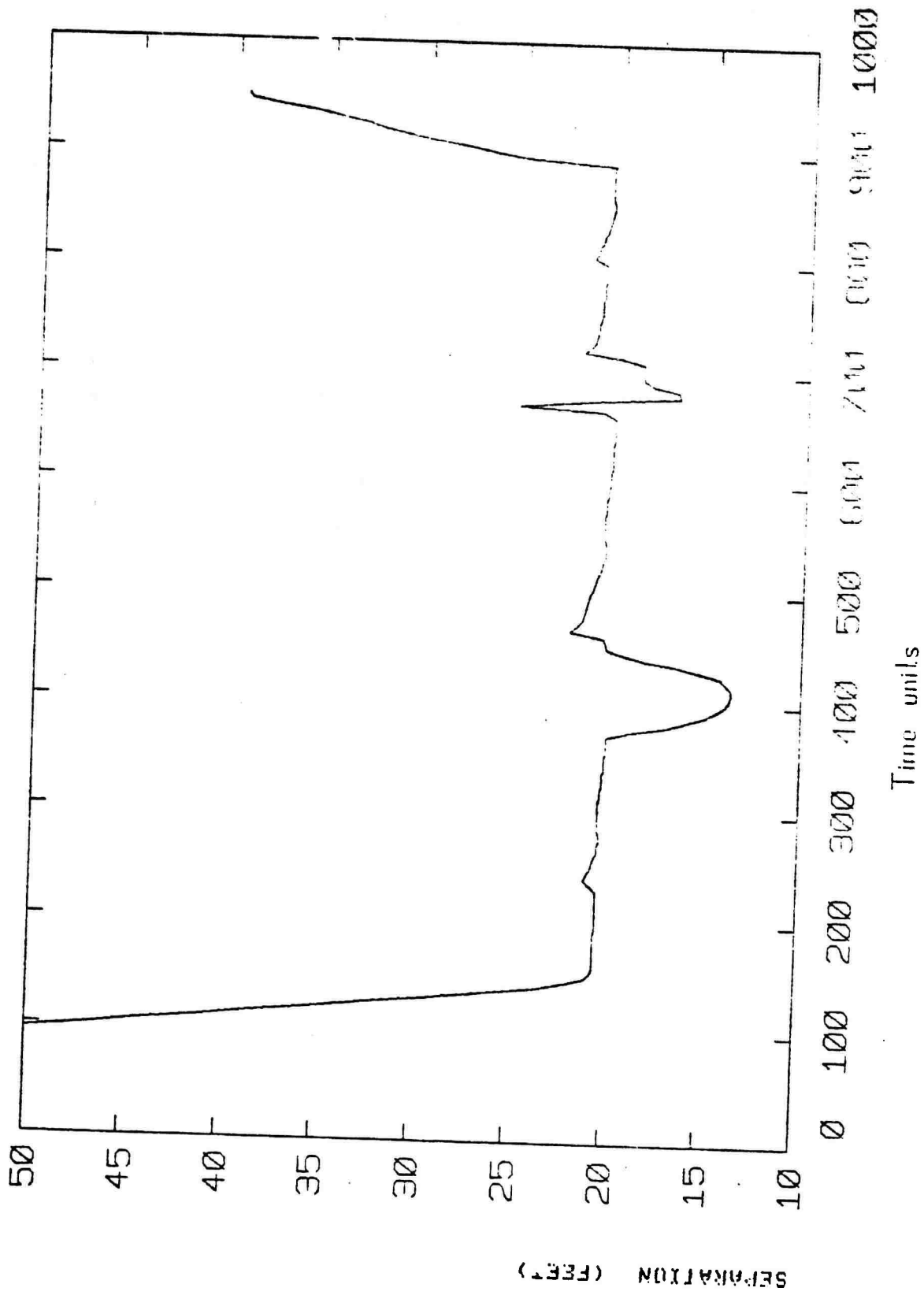


FIGURE (7.22) RESPONSE OF STATE ACTION MATRIX B

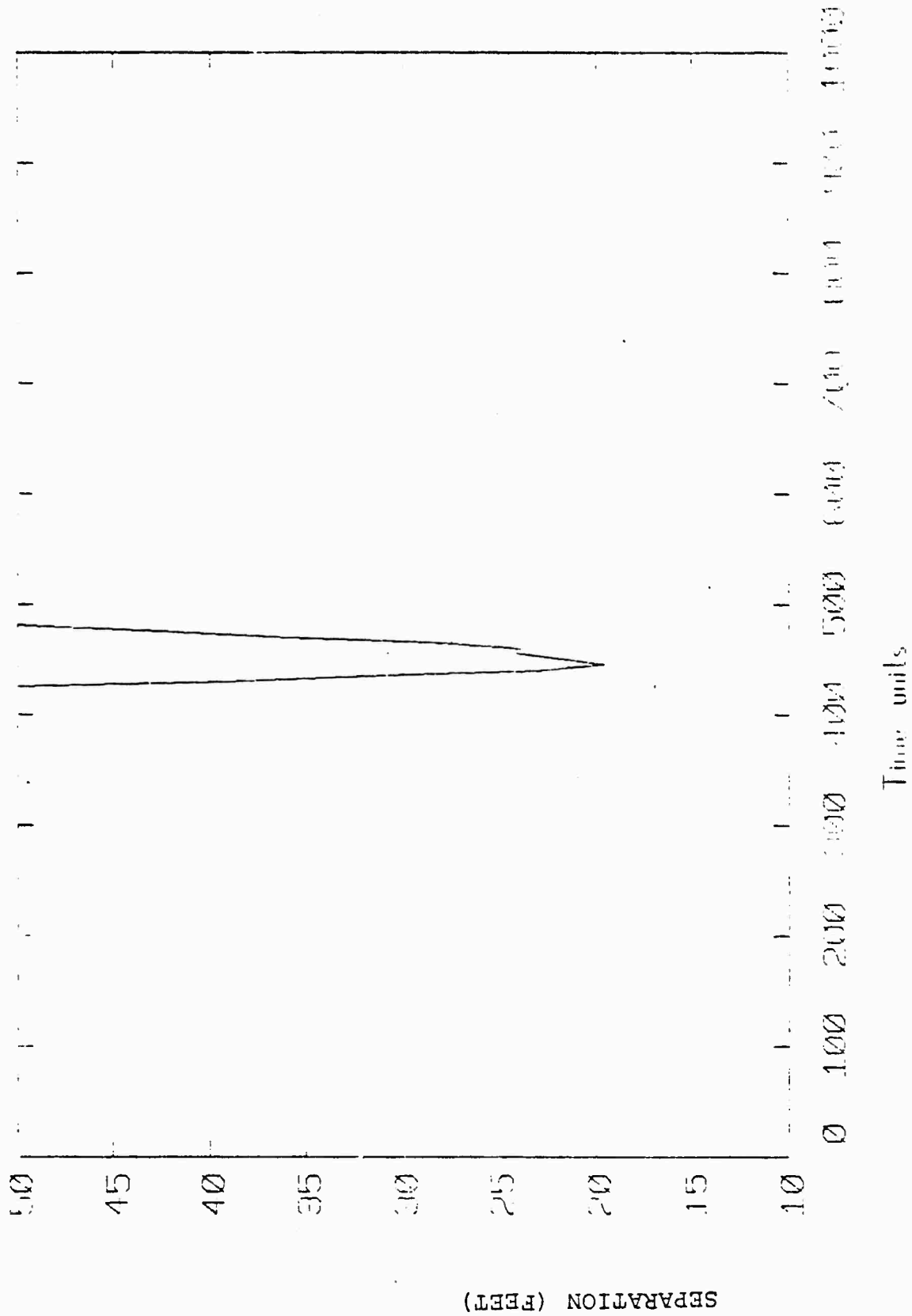


FIGURE 7.23 RESPONSE OF STATE ACTION MATRIX II

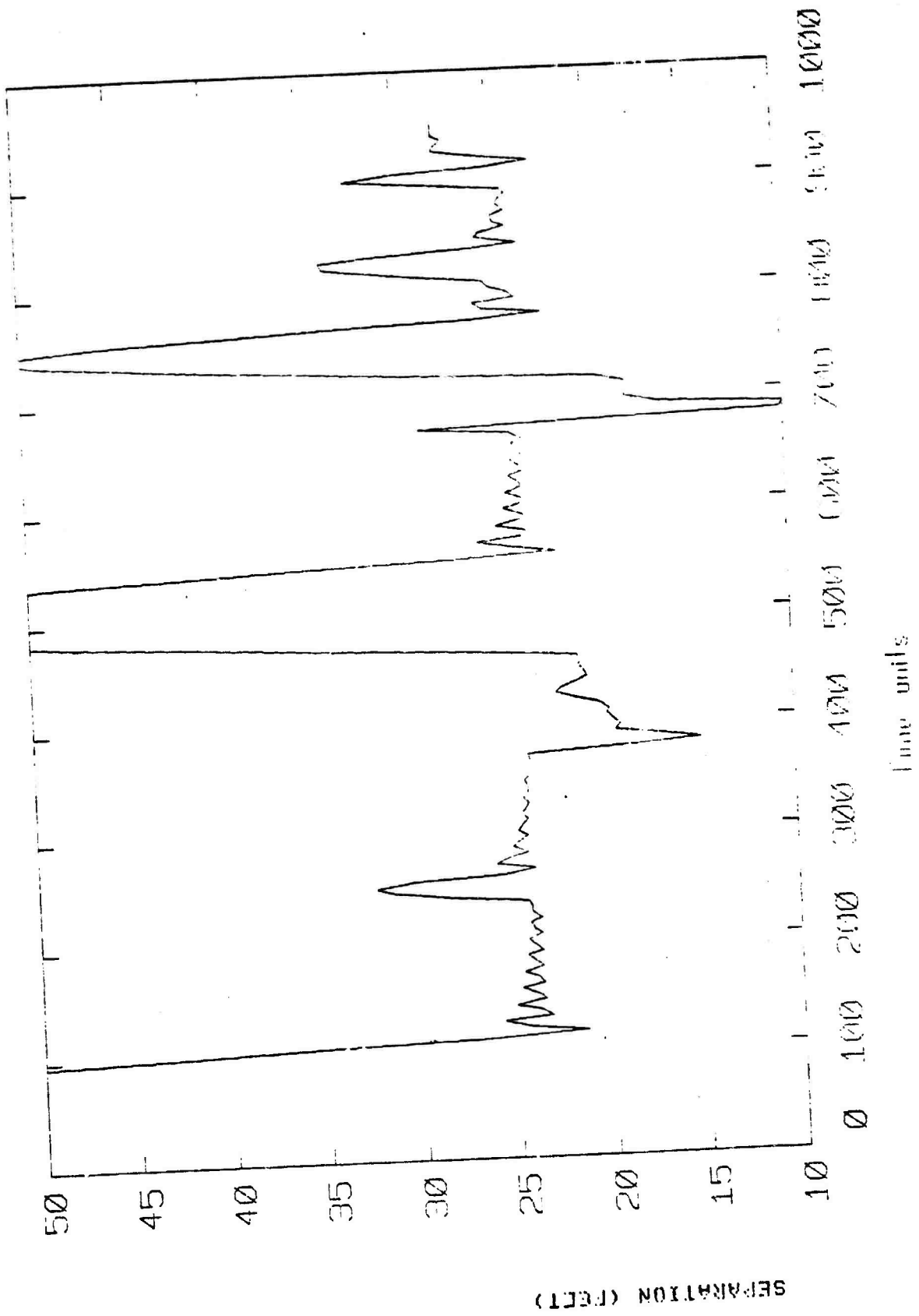


FIGURE (7.24) RESPONSE OF STATE ACTION MATRIX 12

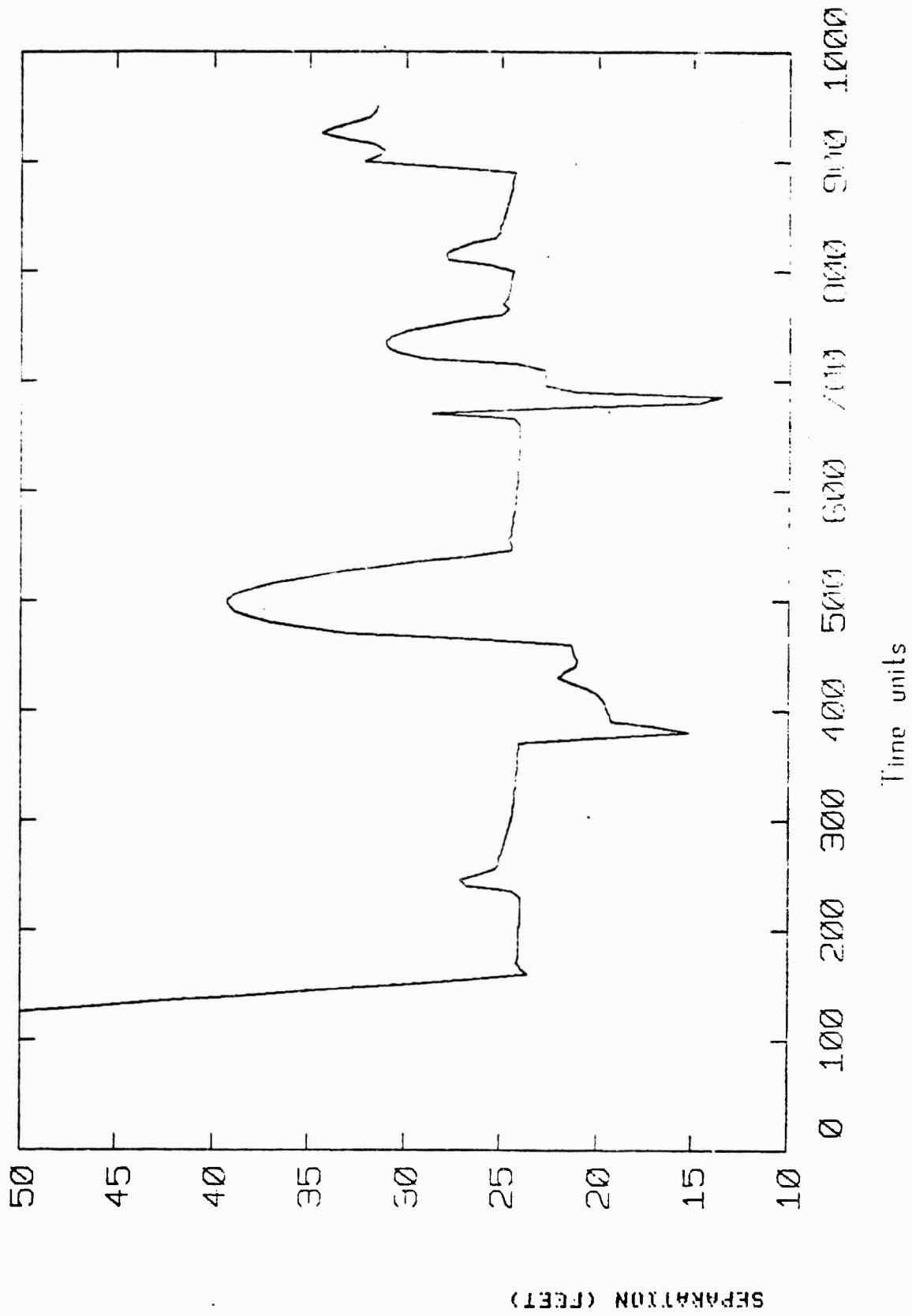


FIGURE (7.25) RESPONSE OF STATE ACTION MATRIX 13

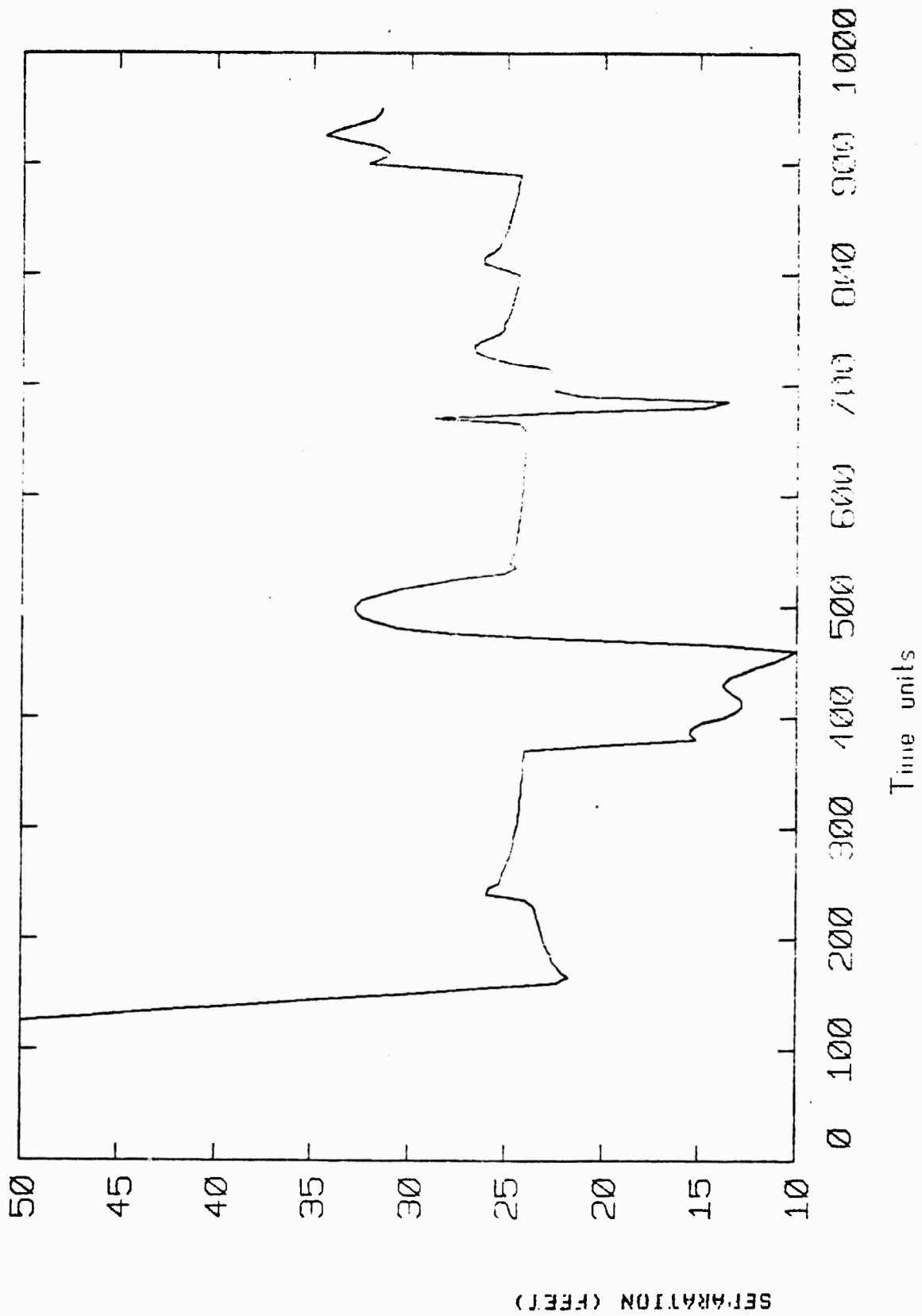


FIGURE (7.26) RESPONSE OF STATE ACTION MATRIX 14

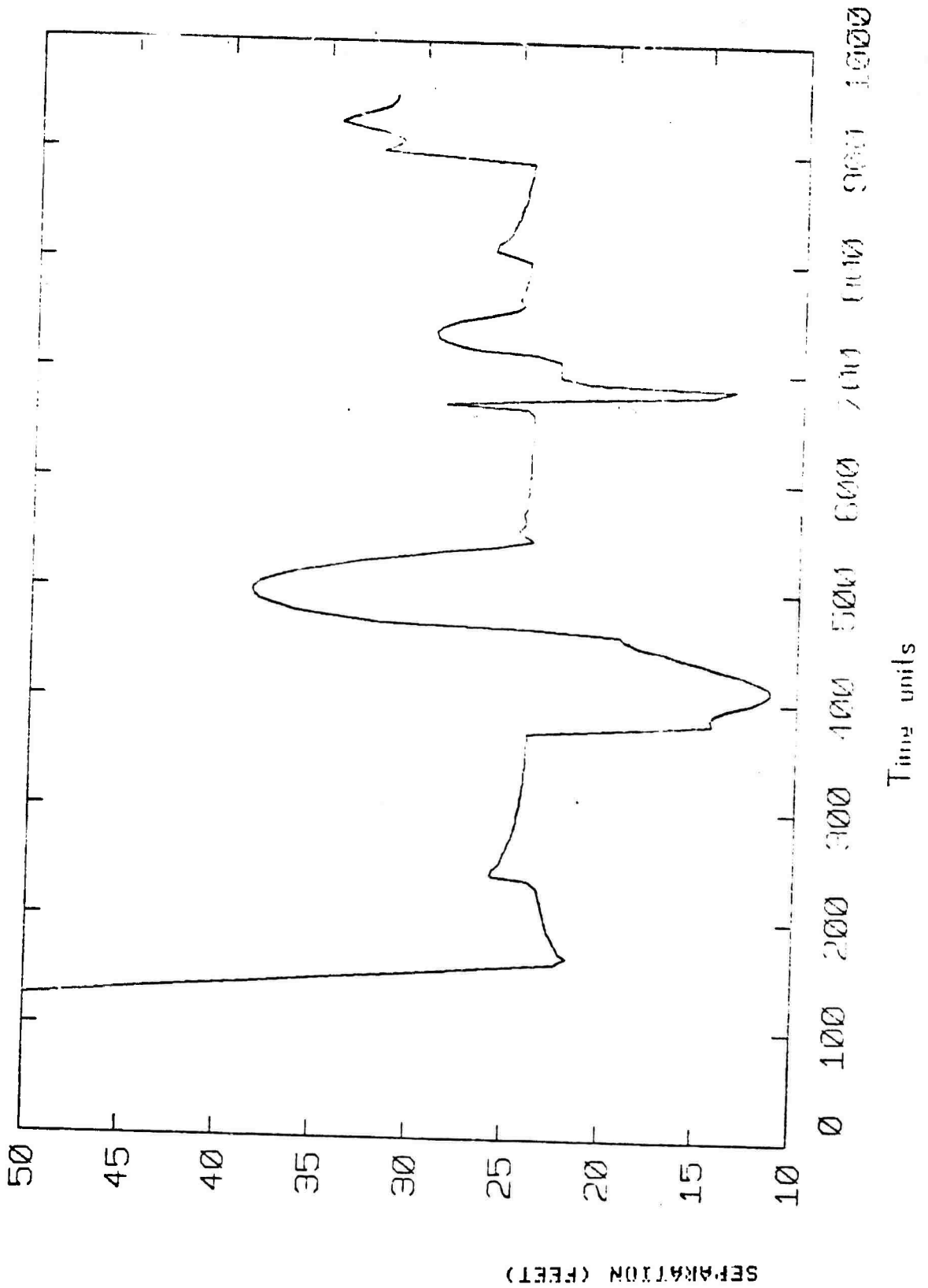
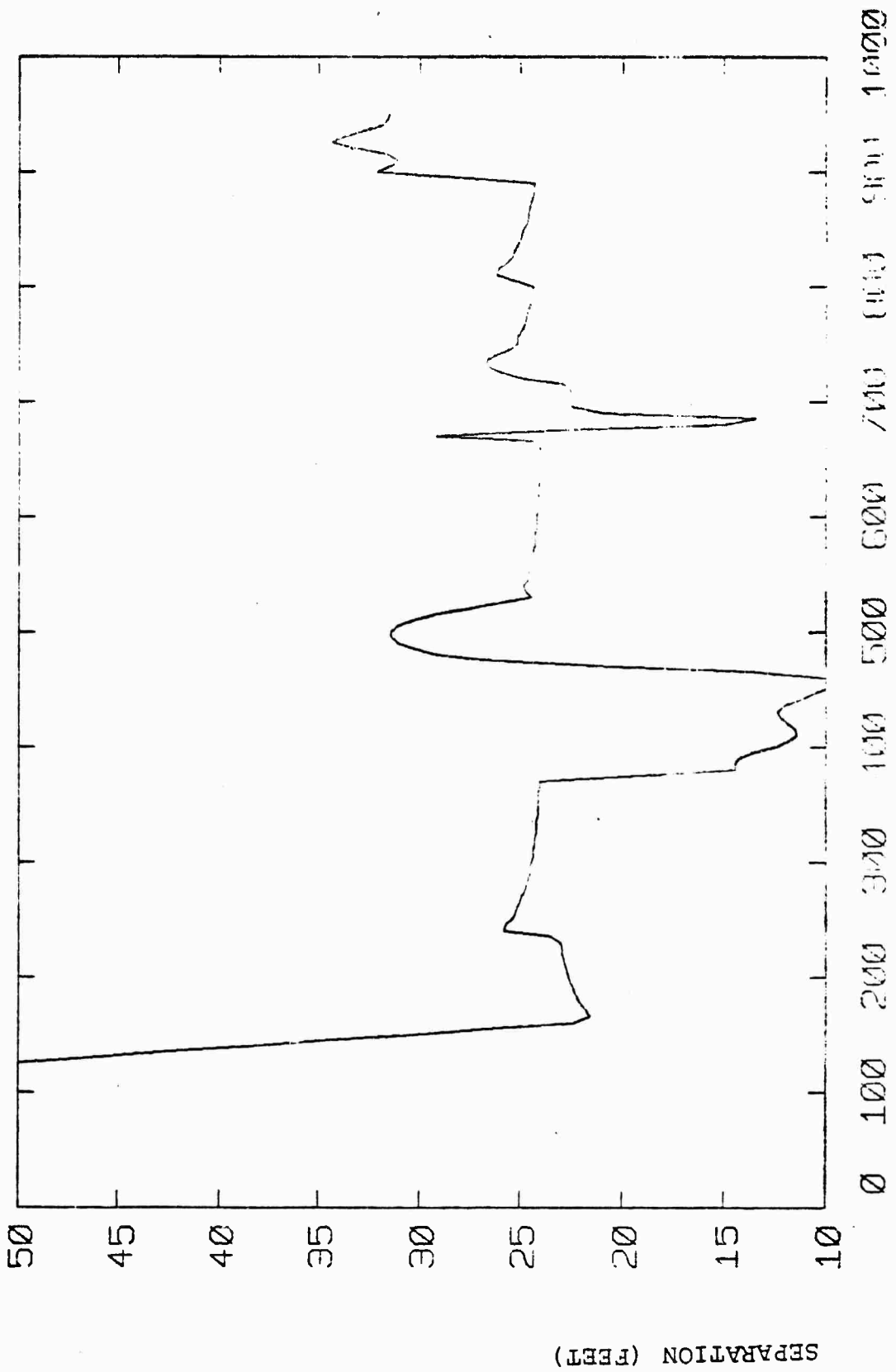


FIGURE (7.27) RESPONSE OF STATE ACTION MATRIX 15



Time units

FIGURE 7.28 RESPONSE OF STATE ACTION MATRIX 16

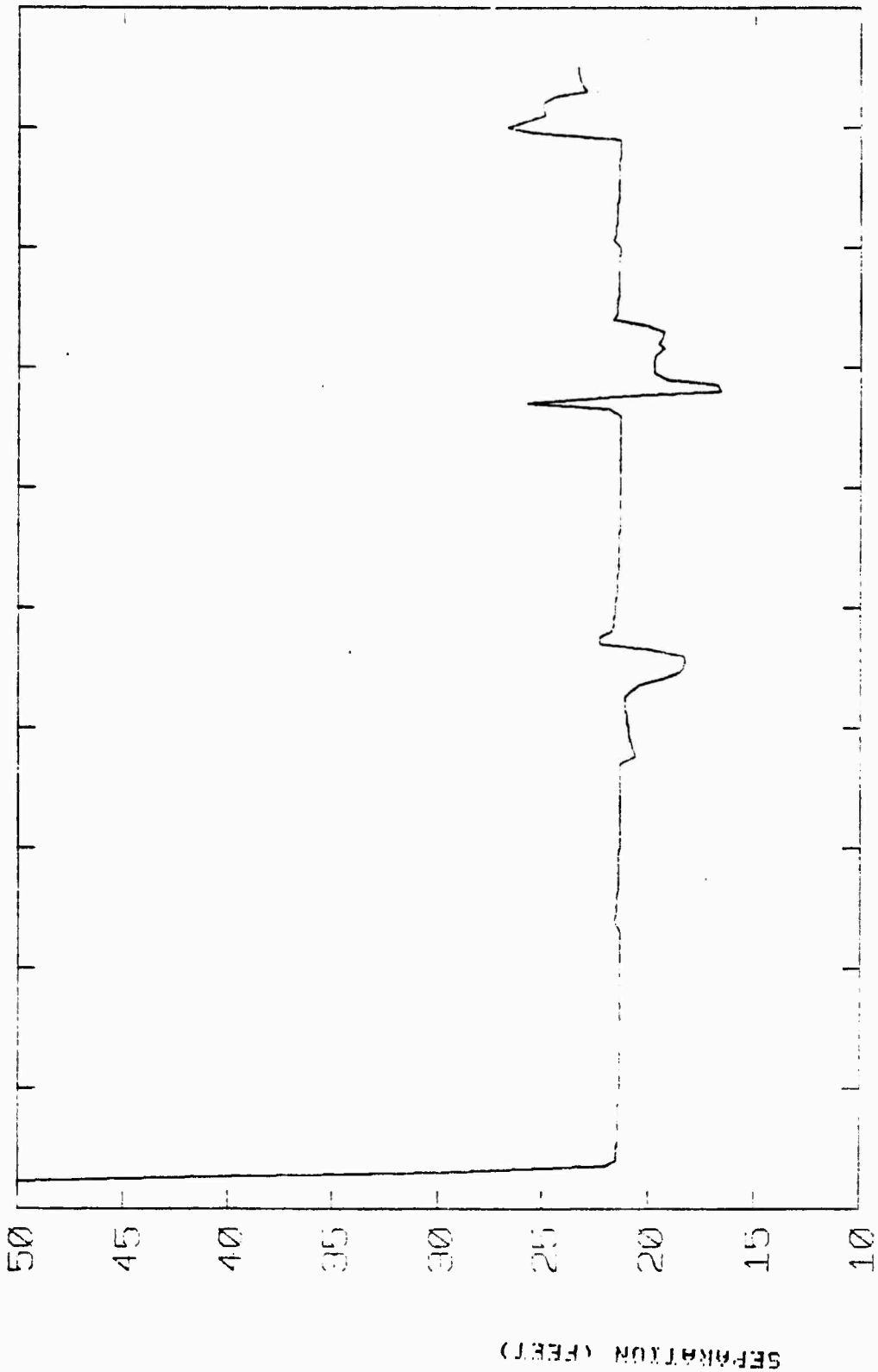


FIGURE (7.29) RESPONSE OF STATE ACTION MATRIX 9

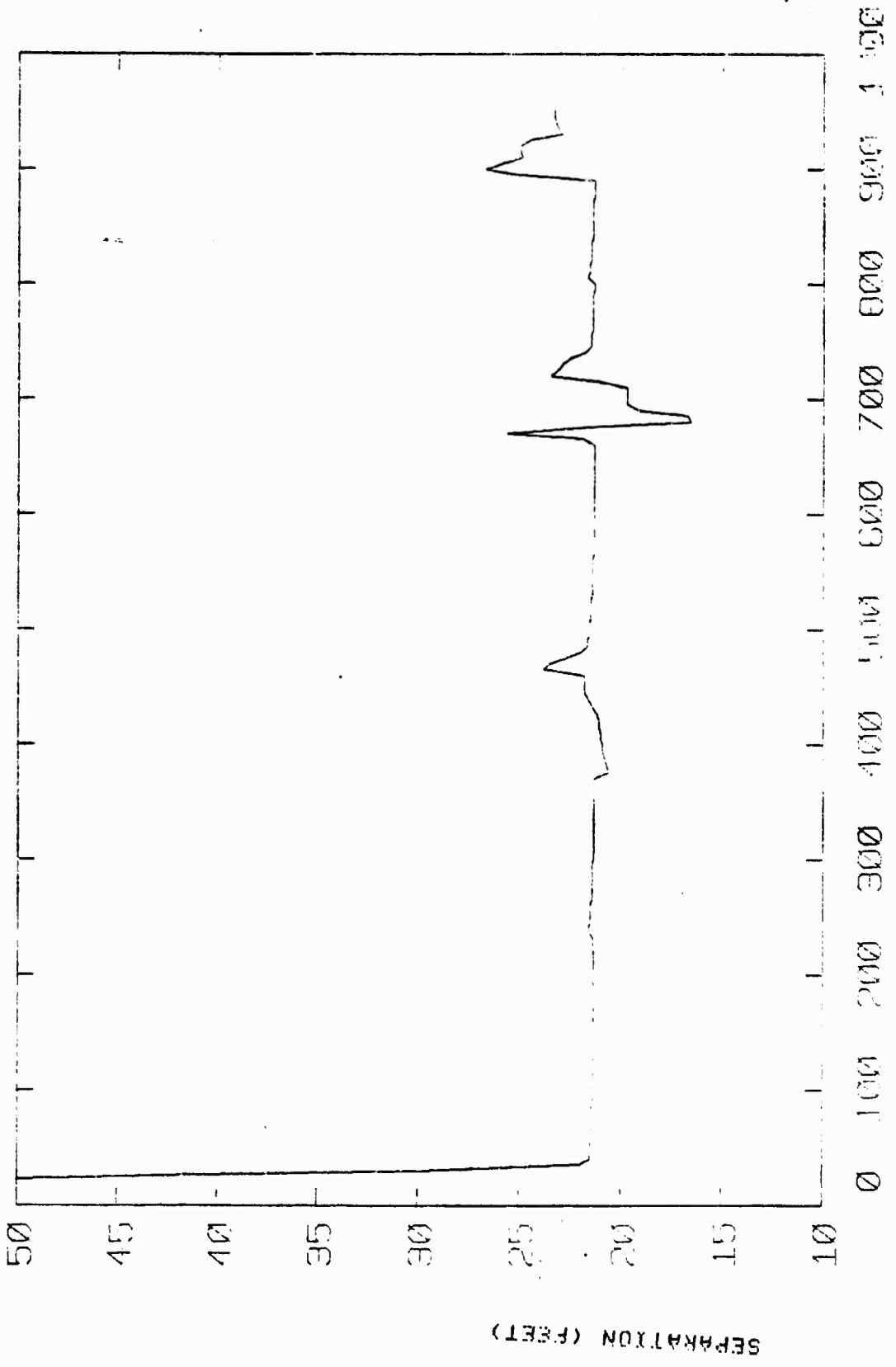
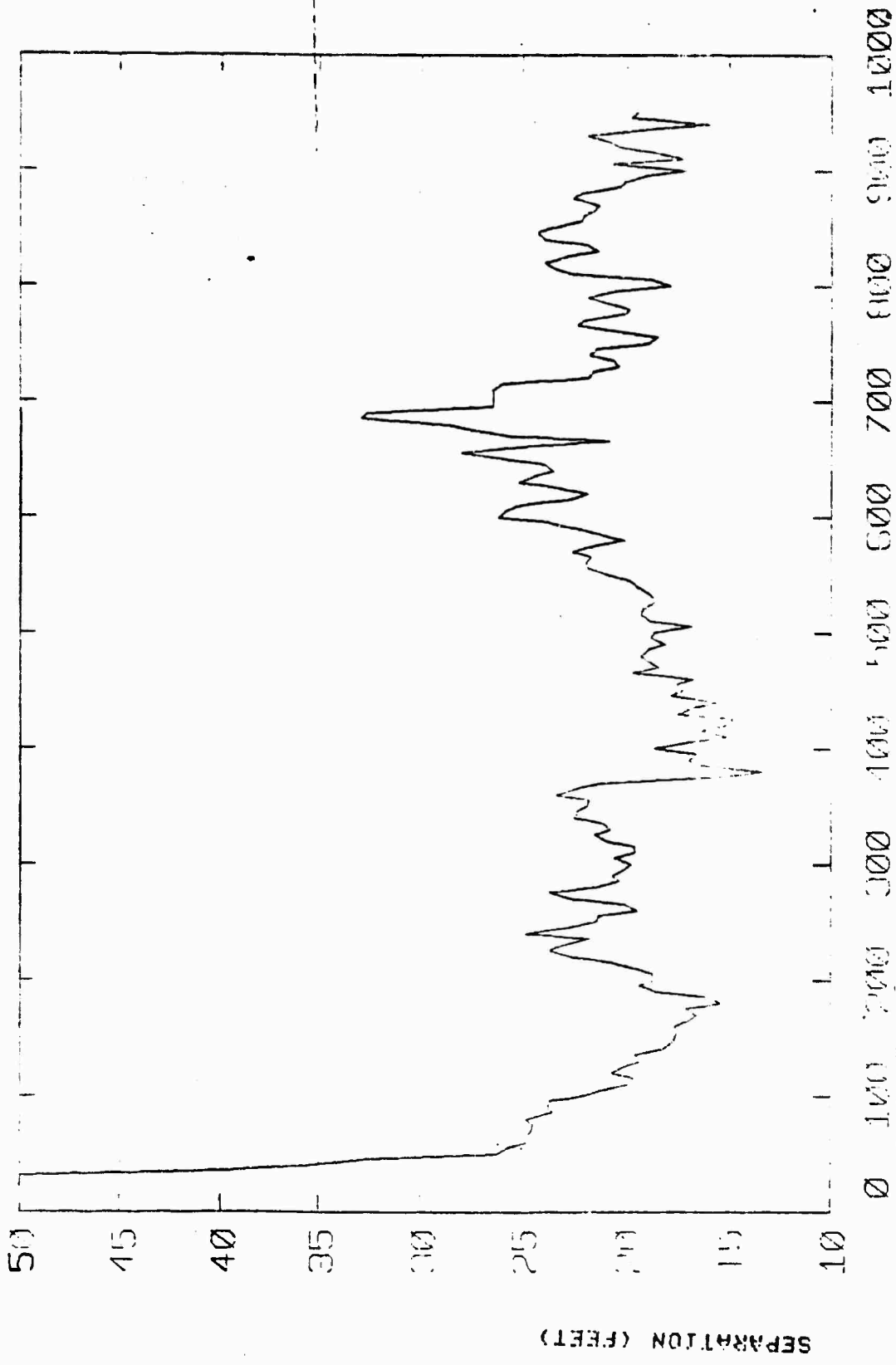


FIGURE (7.30) RESPONSE OF STATE ACTION MATRIX 10



Time units  
 SEPARATION (FEET)  
 FIGURE (7.31) RESPONSE OF HUMAN OPERATOR

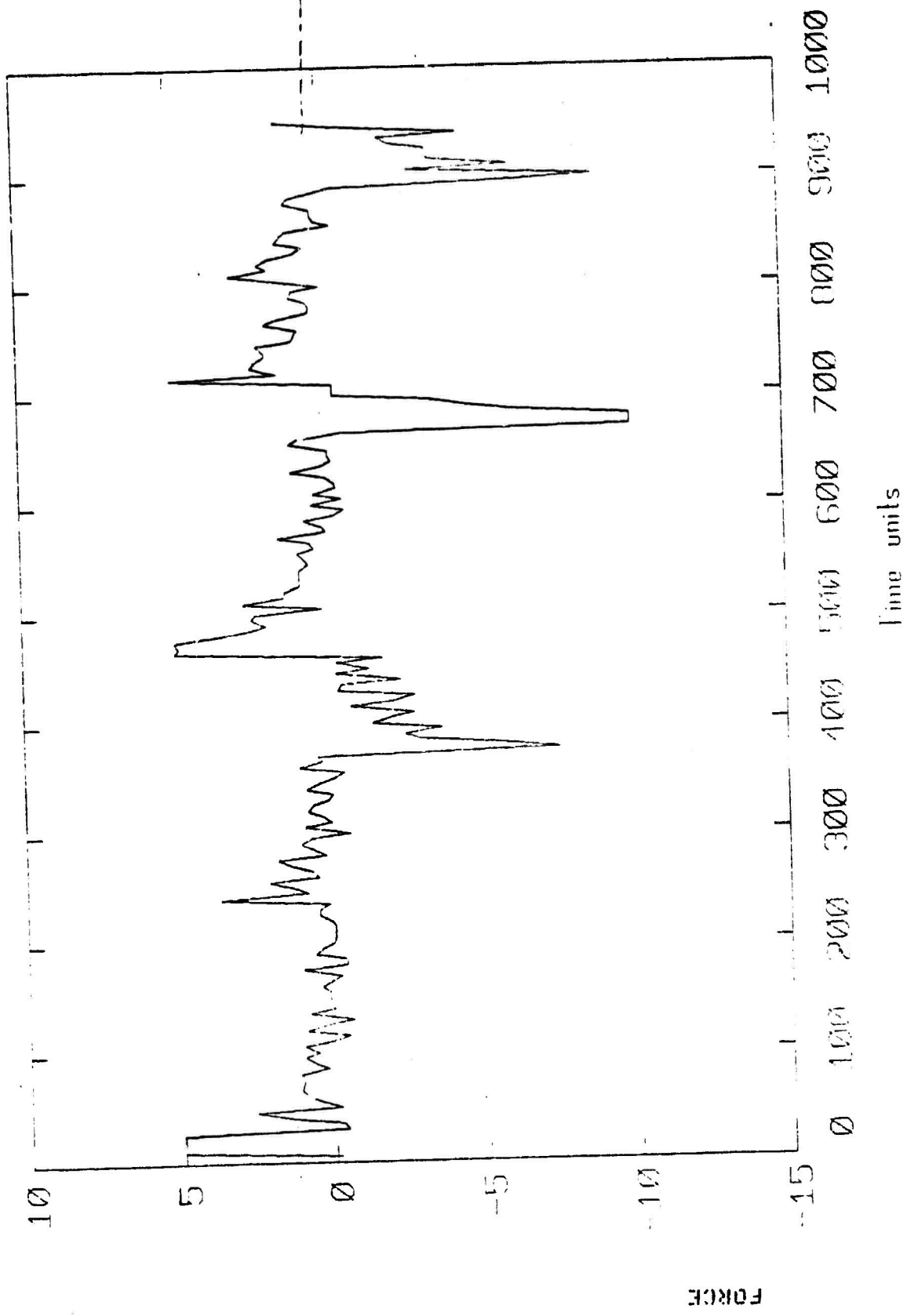


FIGURE (7.32) JOYSTICK INPUT OF HUMAN OPERATOR

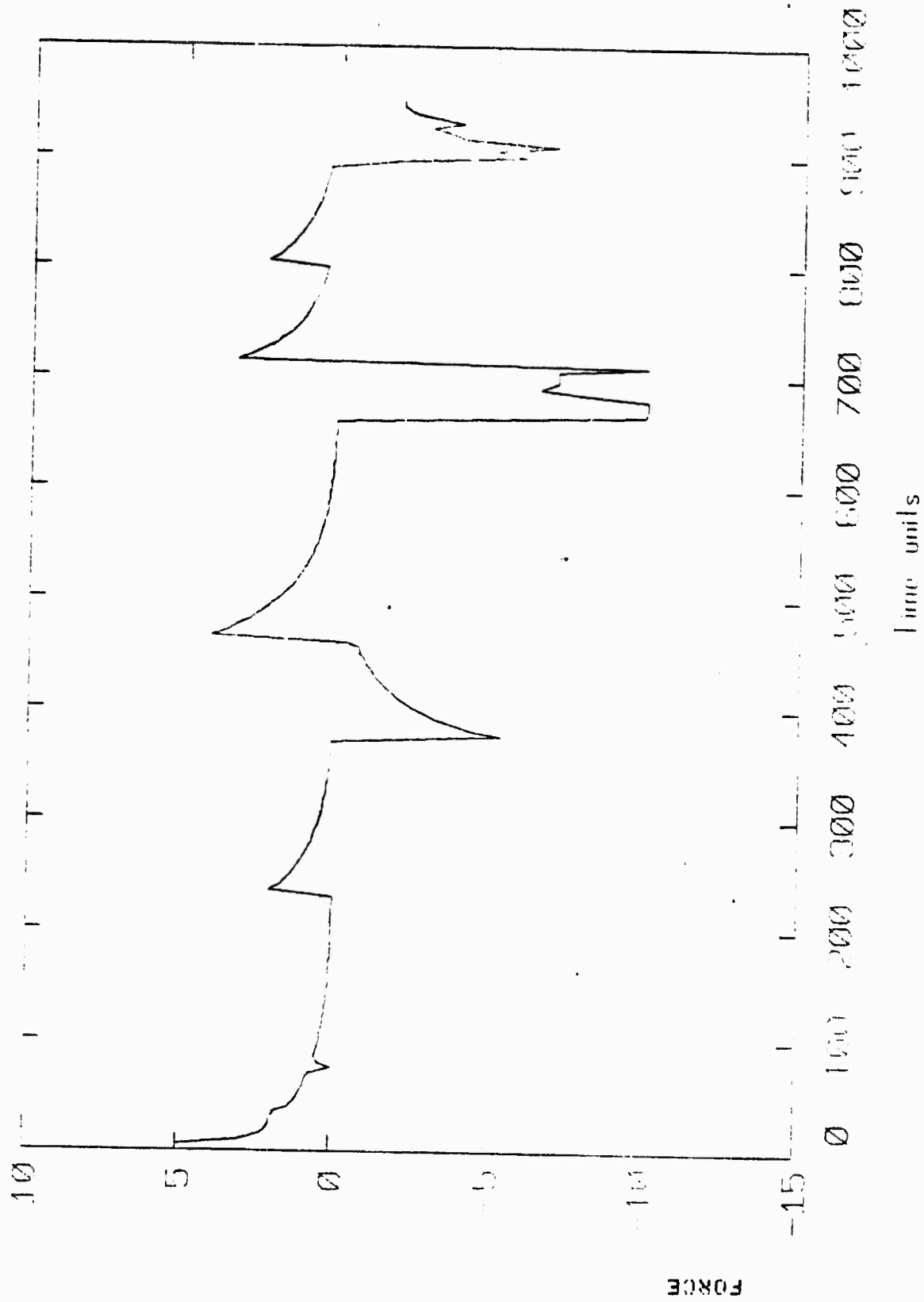


FIGURE (7.33) INPUT OF STATE ACTION MATRIX 10

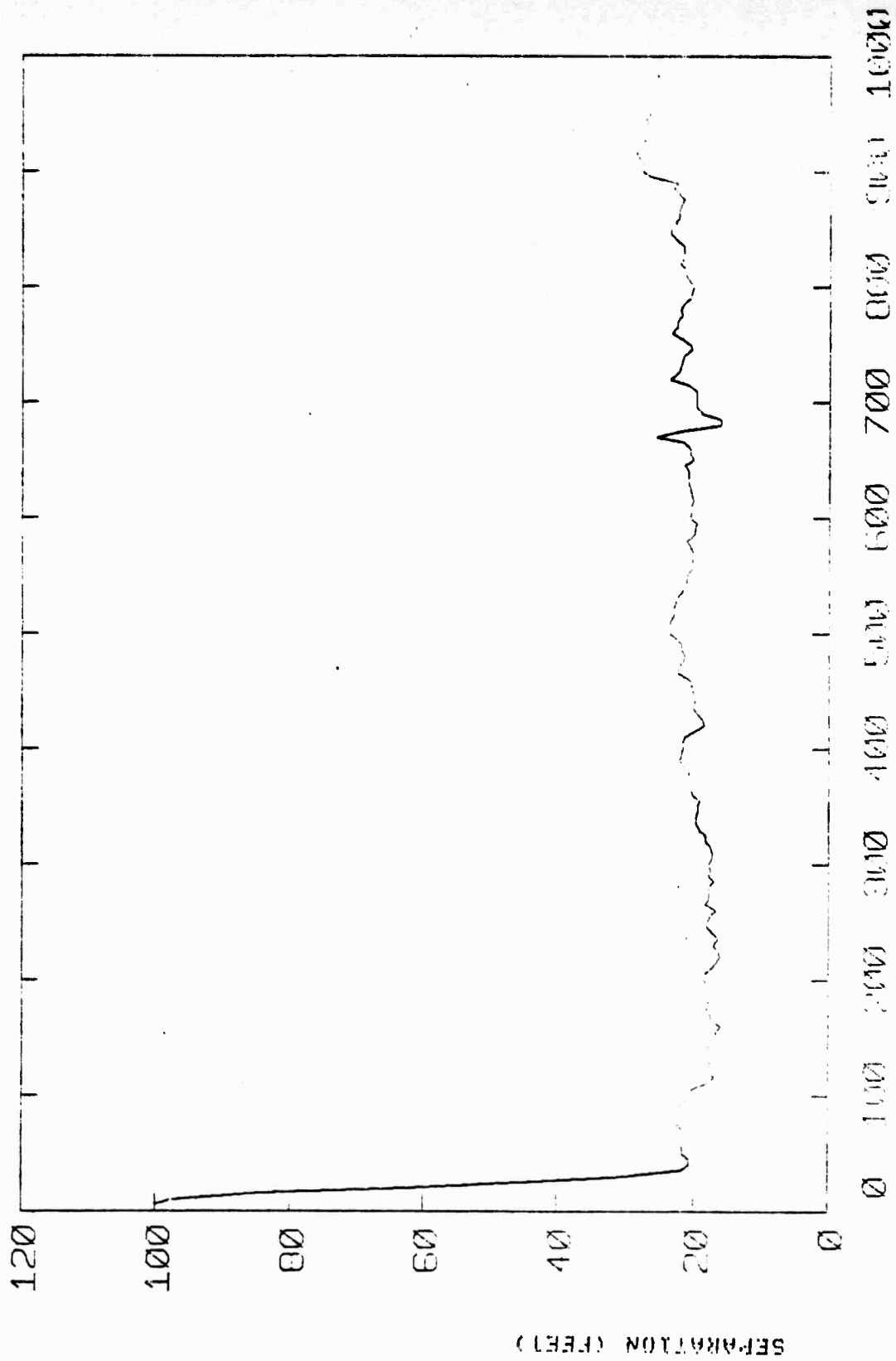


FIGURE (7.34) RESPONSE OF STATE ACTION MATRIX 10 WITH OBSERVATION NOISE

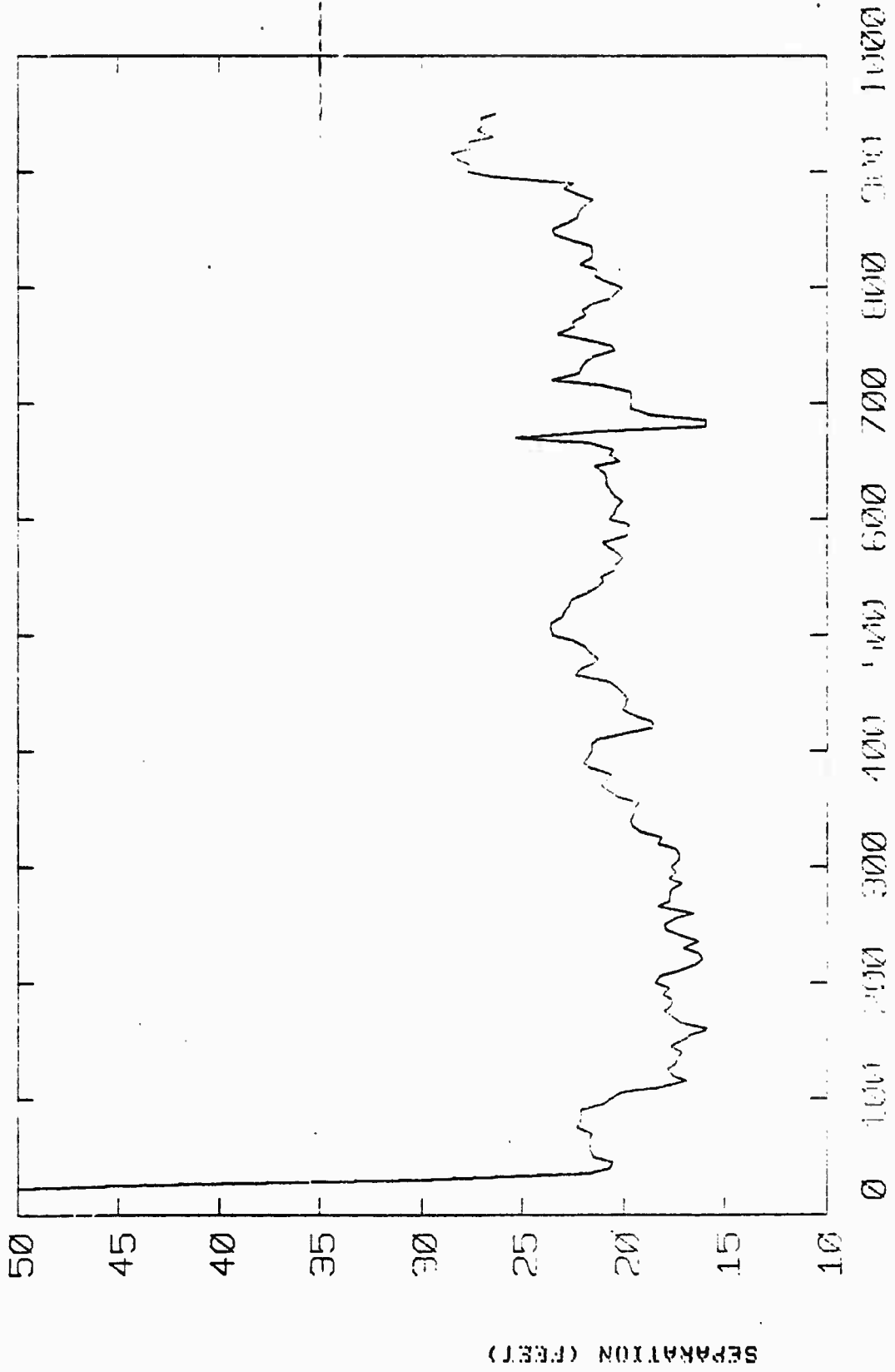


FIGURE (7.30) RESPONSE OF STATE ACTION MATRIX 10 WITH OBSERVATION NOISE

## REFERENCES

- 1- Barrett, D.S. A COMPUTER GRAPHIC METHOD FOR VISUAL SEARCH THROUGH MULTIATTRIBUTE DATA  
MIT thesis, Mechanical Engineering, 1981
- 2- Knepp, L.A. COMPUTER GRAPHICS IN MULTI-ATTRIBUTE DATA BASE SEARCHES: IMPLEMENTATION AND EVALUATION  
MIT thesis, Mechanical Engineering, 1981
- 3- Vermeulen, B. COMPUTER-AIDED VISUALIZATION FOR DESIGN DECISIONS  
MIT thesis, Mechanical Engineering, 1981
- 4- Zadeh, L.A. FUZZY SETS  
Information and Control, 8 338-353  
1965
- 5- Gaines, B.R. FOUNDATIONS OF FUZZY REASONING  
IEEE, IJMMMS Conference (1976)8
- 6- Freeling, A.N.S. FUZZY SETS AND DECISION ANALYSIS  
IEEE Trans SMC-10 NO. 7 July 1980
- 7- Baptisella, L.F.B. FUZZY METHODOLOGIES FOR INTERACTIVE MULTICRITERIA OPTIMIZATION  
Ollero Anibal  
IEEE Trans SMC-10 No.7 July 1980
- 8- Saridis, G.N. A HIERARCHICAL APPROACH TO THE CONTROL OF A PROSTHETIC ARM  
Stephanou, H.E.  
IEEE Trans SMC-7 No 6 June 1977
- 9- Cheng, Y.M. AN ALGORITHM FOR MULTIPLE ATTRIBUTE MULTIPLE ALTERNATIVE DECISION PROBLEMS BASED ON FUZZY SETS WITH APPLICATION TO MEDICAL DIAGNOSIS  
McInnis, b.  
IEEE Trans SMC-10 No 10 Oct. 1980
- 10- Yager, R.R. A LOGICAL ON-LINE BIBLIOGRAPHIC SEARCHER: AN APPLICATION OF FUZZY SETS  
IEEE Trans SMC-10, Jan 1980
- 11- Tribus, M. COMMENTS ON FUZZY SETS, FUZZY ALGEBRA AND FUZZY STATISTICS  
Proc. of IEEE vol 67. pp.1168-1169  
Aug 1979
- 12- Negoita, C.V. ON FUZZINESS IN INFORMATION

- RETRIEVAL  
IEEE IJMMS conf (1976)8, 711-716
- 13- King, P.J.  
Mamdani, E.H. APPLICATION OF FUZZY CONTROL  
SYSTEMS TO INDUSTRIAL PROCESSES  
6th Triennial IFAC World Congress  
Boston, Mass August 1975
- 14- Mamdani, E.H. ADVANCES IN THE LINGUISTIC  
SYNTHESIS OF FUZZY CONTROLLERS  
IEEE IJMMS Conf (1976)8
- 15- Kickert, W.J.M.  
Lemke, H.R.V.N. APPLICATION OF A FUZZY CONTROLLER  
IN A WARM WATER PLANT  
Automatica vol 12 pp.301-308 (1976)
- 16- Tong, R.M. ANALYSIS OF FUZZY CONTROL ALGORITHMS  
USING THE RELATION MATRIX  
IEEE IJMMS Conf (1976)8 pp.679-686
- 17- MacVicar-Whelan FUZZY SETS FOR MAN-MACHINE INTER-  
ACTION  
IEEE IJMMS Conf (1976)8 PP.687-697
- 18- Misc. Authors INTERNATIONAL JOURNAL OF MAN-  
MACHINE STUDIES  
(1976)8
- 19- Misc. Authors IEEE TRANSACTIONS ON SYSTEMS,  
MAN AND CYBERNETICS.  
VOL. SMC-10, No 7 July 1980
- 20- Sheridan, T.B.  
Ferrel, W.R. MAN-MACHINE SYSTEMS  
MIT Press Cambridge Mass (1974)
- 21- Weisbrod, R.L.  
Davis, K.B.  
Freedy, A.  
Weltman, G. ADAPTIVE COMPUTER AIDING IN DYNAMIC  
DECISION PROCESSES: AN INITIAL STUDY  
IN DYNAMIC UTILITY CONVERGENCE AND  
DECISION AIDING  
Perceptrons Tech. Rep.  
No. PTR-1016-74-11 (1974)