NASA Technical Memorandum **104126**

LARGE ANGLE TRANSIENT DYNAMICS (LATDYN) DOCUMENTATION

POST-PROCESSOR MANUAL

Maria V. Mitchum and A. Louis Abrahamson

October 1991

National Aeronautics and Space Administration

Langley Research Center
Hampton, Virginia 23665

TABLE OF CONTENTS

[redacted] INTENTIONALLY [redacted]

LATDYN POST-PROCESSOR MANUAL

INTRODUCTION

The purpose of the post-processor program is to view LATDYN output data in predefined, predetermined formats. The post-processor may be used for plotting data, creating and maintaining a data base of plotting requests, comparing and manipulating data sets in the data base, and the preparation of plots for documentation. There are several ways to view data available to the user such as listing (x,y) table of values for data sets, various manipulations of the data arrays for comparison and display viewing in one of several different plot options. Customized plots may be produced with user defined labels, legends, and title.

RESTRICTIONS

The post-processor executes as an interactive program, with an extensive online HELP feature, reducing the need for printed documentation to a minimum. All manipulations of files and data are done via instructions called COMMANDS. A synopsis of all available commands appears in the initial HELP screen with the option of detail help upon request for a specific COMMAND. The user may obtain hard copies of requested plots by copying the screen or by issuing a command to a postscript printer.

The post-processor has been written in Fortran V4.0 for the VAX/VMS minicomputer in conjunction with a plotting software package: DIGLIB Library. Device drivers are displayed on the screen at initiation of execution for user selection.

```
Response:
   Device  1 is Tek. 4010
   Device  2 is Tek. 4014
   Device  3 is Tek. 4025
   Device  4 is Tek. 4107
   Device  5 is Tek. 4115B
   Device  6 is HP 2647/2648
   Device  7 is HP generic 7475A plotter
   Device  8 is Postscript/QMS-800 tall
   Device  9 is DEC VT240
   Device 10 is Postscript/QMS-800 wide(HC-Hard Copy)

   Enter Device #:(cr)

   (user enters a number 1,... up to 10)
```

DESCRIPTION

The post-processor is designed to handle up to five (5) LATDYN output files at one time for parameter analysis. The program is flexible to the extent that these LATDYN files may be opened and closed at any time during execution. Only one (1) file may be active or in 'USE' at a time. External or Stranger files of data may also be read into the program for further comparative studies.(See READ Command)

The post-processor creates and maintains, at the user's discretion, an internal file called the STORFILE. This file stores the data base information as designated by the user. Once a LATDYN plot has been created the data, through certain commands, may be stored in the data base for reference later during execution or the data can be ignored. The user may at anytime delete data from the data base, write data from the STORFILE onto a permanent user designated external file, list the arrays on the STORFILE, copy data from a previously saved post-processor STORFILE file and add some or all to the present STORFILE, read an external user-defined file with one set of $(x,y_1..y_i)$ data and add that to the STORFILE data base or save the entire STORFILE for a later execution.

Plot requests are divided into two (2) separate and distinct sets of instructions or COMMANDS. One set deals with data as it exists on the LATDYN output files. These plots cannot be customized or changed or personalized in any manner. The only action that can be taken is to store the data in the data base. The second set of plot instructions deals exclusively with data sets that reside in the data base. Once the data exists on the data base then the user can proceed with all the options available to the STORFILE data, such as plotting more than one type of data set on the same graph or manipulating data sets and then displaying the results.

Four (4) types of plotting options are available to the user for displaying data sets from the data base through the 'PLOTSTORi(or PSi)' command. The user may also customize an array in the data base by adding new labels and legend to the data, which are optional when requesting a plot.

A 'TITLE' may be added at anytime during execution to a previously requested (good) plot to remain in effect for future plot requests until deleted or changed.

The plotting window may also be changed at the user's discretion or reset to auto, the default condition at start-up.

The default file extension name expected by the program for a LATDYN output file is '.plt' and for a STORFILE file is '.str'. Any other file name must be declared in its entirety.

COMMAND SUMMARY

At start-up of the program execution, the user will pick the appropriate device driver, then the program will respond with the title/version number and the new features included in that version of the code. All responses and COMMANDS issued by the user must be accompanied by a carriage return (cr). The following 6 characters are legitimate field delimiters available for use in issuing a command:

' ' = a blank	':' = a colon
',' = a comma	'/' = a forward slash
';' = a semicolon	'\' = a backward slash

Information and program control will appear as:

Enter next command: (HELP/INFO/EXIT)

At this time the program is waiting for a COMMAND.

Type: HELP
Response: All available commands, for further help type specific command name, or (cr) to return to main program.
Example:

```
     Summary of COMMANDS :
Information and  Program Control:  HELP, INFO, EXIT
Latdyn output files:  OPEN, USE, CLOSE, LISTFILES
External data files:  READ,   SINDA
Current Storfile:  INDEX, DELETE, SAVE, LIST, WRITE,
                   LABELS, PLOTSTORi(PSi), STORON, STOROFF, STORE
Old Storfile:  COPY
Plot Window:  AUTO, WINDOW
Device control:  DEVICE
Hard Copy:  HC
Special P-,PSi cosmetic feature:  TITLE
Special Storfile data operations: OVO, OPO, OMO, OXO, CXO
                   LOGO, LNO, SINO, COSO, ABSO, POWERO
PLOT(P-), STOR(S-) requests:
   X,    Y,    Z;      LS,  LD,  LA;     FRC,  DEF;            H
  WX,   WY,   WZ;      RS,  RD,  RA;     QVX,  QVY,   QVZ;    C
XLOC, YLOC, ZLOC;                        MAG, WMAG, QVMAG;    Q

ENTER COMMAND NAME FOR FURTHER HELP...(CR)
 ENTER (CR)....TO RETURN TO MAIN PROGRAM
```

Type: INFO
Response: The 'USE' file name followed by 'node' and 'member' information.
Example:(See APPENDIX A)

```
INFO

 ACTIVE LATDYN FILE : A6DAT.PLT

 # OF USER DEF. GRIDPOINTS =      5, # OF HINGESPOINTS =      2
 # OF  FLEXIBLE  ELEMENTS  =      2
 # OF  LINEAR:      SPRINGS =     0, DAMPERS =      0, ACTUATORS =      0
 # OF  ROTATIONAL: SPRINGS =      0, DAMPERS =      0, ACTUATORS =      0
 # OF       Q-VARIABLES    =      0, # OF CONDITIONS =      0
 # OF LINEAR SPRINGS,DAMPERS,ACTUATORS =              0
 # OF ROTATIONAL SPRINGS,DAMPERS,ACT.  =              0
 # OF  MEMBER LOCATIONS =            2
 # OF   RIGID   MEMBERS =            0
 # OF FLEXIBLE  MEMBERS =            2
 # OF USER DEF.ELEMENTS =            2
 # OF        Q-VECTORS  =            0
 # OF COMPONENTS DEFINED=            0
 HIGHEST GRIDPOINT DEF. =            5
 HIGHEST HINGEPOINT DEF.=            1

 Enter next command (HELP/INFO/EXIT)
```

Type: EXIT
Response: THE END (EXIT program)

COMMAND SYNOPSIS

LATDYN output files(.plt)
  OPEN -- Open a LATDYN binary output file
  USE -- Activate opened file for program use
  CLOSE -- Close a LATDYN file
  LISTFILES -- List all LATDYN files currently open

External data files
  READ -- Read a user created file and add to STORFILE
  SINDA -- Read a user created SINDA file, add to STORFILE

Current STORFILE
  INDEX -- List plot identification existing on STORFILE
  DELETE -- Delete (x,y) data set on STORFILE
  SAVE -- Save all (x,y) data sets on STORFILE
  LIST -- List an (x,y) data set on STORFILE
  WRITE -- Write an (x,y) data set onto an external file
  LABELS -- Create labels, legend for a data set on STORFILE
  PLOTSTORi(PSi) -- Plot (option,i=1,2,3,or 4) data from
                    STORFILE
  STORON -- Store (x,y) LATDYN data from plot requests to
            STORFILE
  STOROFF -- Turn off STORON
  STORE -- Store previous LATDYN plot data onto STORFILE

Old STORFILE(.str)
  COPY -- Copy data from previous post-processor STORFILE

Plot Window
  AUTO -- Automatic plot window set by data
  WINDOW -- User defined plot window

Device Control
  DEVICE -- Change device driver

Hard Copy
  HC -- Hard copy request on Postscript Printer

Special P-, PSi cosmetic feature
  TITLE -- Add title to plots

Special STORFILE data operations(ord=ordinate,absc=abscissa)

```
  OVO -- ord1 vs. ord2
  OPO -- (ord1 + ord2) vs. absc
  OMO -- (ord1 - ord2) vs. absc
  OXO -- (ord1 * ord2) vs. absc
  CXO -- (constant * ord) vs. absc
  LOGO -- log(ord) vs. absc
  LNO -- ln(ord) vs. absc
  SINO -- sin(ord) vs. absc
  COSO -- cos(ord) vs. absc
  ASBO -- abs(ord) vs. absc
  POWERO -- (ord**constant) vs. absc
```

Plot(P-),Stor(s-) requests:(LATDYN data)

```
                Example :PLOTX, or P-X; plot data vs. time
                         :STORX, or S-X; store data on STORFILE
  X;Y;Z -- Cartesian components
  WX;WY;WZ -- Rotational motion vectors
  H -- Magnitude of Hingepoint
  XLOC;YLOC;ZLOC -- gridpoint location
  FRC;DEF -- Internal force; Deformation
  LS;LD;LA -- Linear: spring; damper; actuator
  RS;RD;RA -- Rotational: spring; damper; actuator
  Q -- Q-variable
  QVX;QVY;QVZ -- X;Y;Z Components of a Q-vector
  C -- Logical state of Conditions
  MAG;WMAG;QVMAG -- Magnitude of motion vector; Rotational
                    motion vector;  Q-vector mag
```

PROGRAM LIMITATIONS

The maximum number of curves that can be requested on one graph is 4.

The maximum number of data sets stored in the data base STORFILE is 100.

The maximum number of LATDYN files opened at one time is 5.

The maximum number of data values for a single array that can be transferred from the LATDYN Program to the POST-PROCESSOR is 2000.

The maximum number of special labels/legends table entries is 100.

The maximum number of external arrays, entered through the READ command, is 25.

HOW TO USE THE POST-PROCESSOR FOR PLOTTING LATDYN DATA

Before a LATDYN data plot can be created, a LATDYN binary output file must be opened. The default file extension expected is '.plt'. If the file has another extension then it must be included in the open command. Up to five (5) files may be opened at one time and as the user proceeds from one to another he must issue the use command. The close command closes a single file or 'all' files. The listfiles command can be used to review which files have been opened and which file is the current use file. When a user plots a set of data from the LATDYN data file, the filename is appended to the legend of the plot for documentation.

LATDYN FILE COMMANDS:

COMMAND: OPEN,latdynfile

DESCRIPTION: OPEN A LATDYN BINARY OUTPUT FILE

latdynfile - MUST BE AN EXISTING FILE NAME

NOTE: .PLT IS THE DEFAULT FILE EXTENSION
ONLY 5 FILES CAN BE OPENED AT ONE TIME
FILE IS THE DEFAULT 'USE' FILE

COMMAND: USE,latdynfile

DESCRIPTION: UP TO 5 LATDYN BINARY OUTPUT FILES MAY BE OPENED AT ONE TIME, BUT ONLY ONE FILE MAY BE ACTIVE OR IN USE. THE LATEST FILE THAT HAS BEEN OPENED IS THE DEFAULT USE FILE UNLESS THE USE COMMAND IS ISSUED FOR ANOTHER OPENED FILE.

latdynfile - AN EXISTING OPENED LATDYN BINARY OUTPUT FILE

NOTE: TO PERUSE THE FILE SITUATION ISSUE
COMMAND: LISTFILES

COMMAND: CLOSE,filename

DESCRIPTION: CLOSE LATDYN FILE WHICH HAS BEEN PREVIOUSLY OPENED

filename - USER LATDYN FILE NAME
ALL - CLOSE ALL LAYDYN FILES

NOTE: IF THE CLOSED FILE IS THE 'USE' FILE,
THEN ANOTHER FILE MUST BE DECLARED IN 'USE'

COMMAND: LISTFILES

DESCRIPTION: LIST ALL THE LATDYN FILES CURRENTLY OPEN AND THE ACTIVE OR USE FILE

LATDYN PLOT COMMANDS:

LATDYN data plot request commands are listed alphabetically. All commands start with PLOT or (P-), such as PLOTX or P-X. The following commands are as they appear in the HELP feature of the program; arguments are explained therein, using the nomenclature established in the LATDYN USERS MANUAL for component/gridpoint/hinge/member notation.

```
   COMMAND:  PLOTC,list of condition numbers
 ALTERNATE:    P-C,list of condition numbers

DESCRIPTION: PLOT THE LOGICAL STATE OF USER
             DEFINED CONDITIONS VERSES TIME, WHERE:

             list of condition numbers - AS DEFINED BY USER

      NOTE:  PLOTS ARE SYMBOLIZED TRUE(T) - FALSE(F)
```

```
   COMMAND:  PLOTDEF,option,element#
 ALTERNATE:    P-DEF,option,element#

DESCRIPTION: PLOT THE DEFORMATION
             IN A BEAM ELEMENT VERSES TIME, WHERE:

             option - AX, FOR AXIAL DEFORMATION
                      TW, FOR RELATIVE TWISTING ANGLE BETWEEN 2 ENDS
                      YA, FOR BENDING ROTATION AT END A(Y DIR)
                      YB, FOR BENDING ROTATION AT END B(Y DIR)
                      ZA, FOR BENDING ROTATION AT END A(Z DIR)
                      ZB, FOR BENDING ROTATION AT END B(Z DIR)
             element# - ELEMENT IDENTIFIERS( MAX 4)
                        ex: #Mj, or #CiMj, or #MjEk ECT.
```

```
   COMMAND:  PLOTFRC,option,element#
 ALTERNATE:    P-FRC,option,element#

DESCRIPTION: PLOT THE INTERNAL FORCES
             IN A BEAM ELEMENT VERSES TIME, WHERE:

             option - AX, FOR AXIAL FORCE
                      TW, FOR TWISTING FORCE
                      YA, FOR BENDING MOMENT AT END A(Y DIR)
                      YB, FOR BENDING MOMENT AT END B(Y DIR)
                      ZA, FOR BENDING MOMENT AT END A(Z DIR)
                      ZB, FOR BENDING MOMENT AT END B(Z DIR)
             element# - ELEMENT IDENTIFIERS( MAX 4)
                        ex: #Mj, or #CiMj, or #MjEk ECT.
```

```
   COMMAND:  PLOTH,p,hingepoint# list
 ALTERNATE:    P-H,p,hingepoint# list

DESCRIPTION: PLOT THE MAGNITUDE OF THE HINGEPOINT
             MOTION VECTOR RELATIVE TO THE GRIDPOINT TO WHICH
             IT IS ATTACHED VERSES TIME, WHERE:

             p = D, V, OR A (D - DISPLACEMENT, V - VELOCITY,
                             A - ACCELERATION)
             hingepoint# list - HINGEPOINT IDENTIFIERS( MAX 4)
                          ex: #GiHk, or #CiGjHk, ECT.
```

```
   COMMAND:  PLOTLA,list of lin.actuator names
 ALTERNATE:    P-LA,list of lin.actuator names

DESCRIPTION: PLOT LINEAR ACTUATOR DATA VERSES TIME, WHERE:

             list of lin.actuator names - NAMES AS DEFINED IN
                                          COMMAND: LINACTUATOR
```

```
   COMMAND:  PLOTLD,option,list of lin.damper names
 ALTERNATE:    P-LD,option,list of lin.damper names

DESCRIPTION: PLOT LINEAR DAMPER DATA VERSES TIME, WHERE:

             option - RV, FOR RELATIVE VELOCITY
                      DF, FOR DAMPING FORCE
             list of lin.damper names - NAMES AS DEFINED IN
                                        COMMAND: LINDAMPER
```

```
   COMMAND:  PLOTLS,option,list of lin.spring names
 ALTERNATE:    P-LS,option,list of lin.spring names

DESCRIPTION: PLOT LINEAR SPRING DATA VERSES TIME, WHERE:

             option - SD, FOR SPRING DISPLACEMENT
                      SF, FOR SPRING FORCE
             list of lin.spring names - NAMES AS DEFINED IN
                                        COMMAND: LINSPRING
```

```
   COMMAND:  PLOTMAG,p,grid# list
 ALTERNATE:    P-MAG,p,grid# list

DESCRIPTION: PLOT THE MAGNITUDE OF THE
             MOTION VECTOR VERSES TIME, WHERE:

             p = D, V, OR A (D - DISPLACEMENT, V - VELOCITY,
                             A - ACCELERATION)
             grid# list - GRIDPOINT IDENTIFIERS( MAX 4)
                          ex: #Gi, or #CiGj, ECT.
```

COMMAND: PLOTQ,list of Q-variable numbers
ALTERNATE: P-Q,list of Q-variable numbers

DESCRIPTION: PLOT Q-VARIABLE VALUES VERSES TIME AT:

list of Q-variable numbers - AS DEFINED BY USER

COMMAND: PLOTQVMAG,list of Q-vector numbers
ALTERNATE: P-QVMAG,list of Q-vector numbers

DESCRIPTION: PLOT THE MAGNITUDE OF A
Q-VECTOR VERSES TIME, WHERE:

list of Q-vector numbers - AS DEFINED BY USER

COMMAND: PLOTQVX,list of Q-vector numbers
ALTERNATE: P-QVX,list of Q-vector numbers

DESCRIPTION: PLOT X COMPONENT OF A Q-VECTOR VERSES TIME AT:

list of Q-vector numbers - AS DEFINED BY USER

COMMAND: PLOTQVY,list of Q-vector numbers
ALTERNATE: P-QVY,list of Q-vector numbers

DESCRIPTION: PLOT Y COMPONENT OF A Q-VECTOR VERSES TIME AT:

list of Q-vector numbers - AS DEFINED BY USER

COMMAND: PLOTQVZ,list of Q-vector numbers
ALTERNATE: P-QVZ,list of Q-vector numbers

DESCRIPTION: PLOT Z COMPONENT OF A Q-VECTOR VERSES TIME AT:

list of Q-vector numbers - AS DEFINED BY USER

COMMAND: PLOTRA,list of rot.actuator names
ALTERNATE: P-RA,list of rot.actuator names

DESCRIPTION: PLOT ROTATIONAL ACTUATOR DATA VERSES TIME, WHERE:

list of rot.actuator names - NAMES AS DEFINED IN COMMAND: ROTACTUATOR

COMMAND: PLOTRD,option,list of rot.damper names
ALTERNATE: P-RD,option,list of rot.damper names

DESCRIPTION: PLOT ROTATIONAL DAMPER DATA VERSES TIME, WHERE:

option - RW, FOR RELAVTIVE ANGULAR VELOCITY
DT, FOR DAMPING TORQUE
list of rot.damper names - NAMES AS DEFINED IN COMMAND: ROTDAMPER

COMMAND: PLOTRS,option,list of rot.spring names
ALTERNATE: P-RS,option,list of rot.spring names

DESCRIPTION: PLOT ROTATIONAL SPRING DATA VERSES TIME, WHERE:

option - AD, FOR ANGULAR DEFORMATION
ST, FOR SPRING TORQUE
list of rot.spring names - NAMES AS DEFINED IN COMMAND: ROTSPRING

COMMAND: PLOTWMAG,p,grid# list
ALTERNATE: P-WMAG,p,grid# list

DESCRIPTION: PLOT THE MAGNITUDE OF THE ROTATIONAL MOTION VECTOR VERSES TIME, WHERE:

p = V, OR A (V - VELOCITY, A - ACCELERATION)
grid# list - GRIDPOINT IDENTIFIERS(MAX 4)
ex: #Gi, or #CiGj, ECT.

COMMAND: PLOTWX,p,grid# list
ALTERNATE: P-WX,p,grid# list

DESCRIPTION: PLOT THE X CARTESIAN COMPONENT OF GRIDPOINT ROTATIONAL MOTION VECTOR VERSES TIME, WHERE:

p = V, OR A (V - VELOCITY, A - ACCELERATION)
grid# list - GRIDPOINT IDENTIFIERS(MAX 4)
ex: #Gi, or #CiGj, ECT.

COMMAND: PLOTWY,p,grid# list
ALTERNATE: P-WY,p,grid# list

DESCRIPTION: PLOT THE Y CARTESIAN COMPONENT OF GRIDPOINT ROTATIONAL MOTION VECTOR VERSES TIME, WHERE:

p = V, OR A (V - VELOCITY, A - ACCELERATION)
grid# list - GRIDPOINT IDENTIFIERS(MAX 4)
ex: #Gi, or #CiGj, ECT.

COMMAND: PLOTWZ,p,grid# list
ALTERNATE: P-WZ,p,grid# list

DESCRIPTION: PLOT THE Z CARTESIAN COMPONENT OF GRIDPOINT ROTATIONAL MOTION VECTOR VERSES TIME, WHERE:

p = V, OR A (V - VELOCITY, A - ACCELERATION)
grid# list - GRIDPOINT IDENTIFIERS(MAX 4)
ex: #Gi, or #CiGj, ECT.

COMMAND: PLOTX,p,grid# list
ALTERNATE: P-X,p,grid# list

DESCRIPTION: PLOT THE X CARTESIAN COMPONENT OF GRIDPOINT TRANSLATIONAL MOTION VECTOR VERSES TIME, WHERE:

p = D, V, OR A (D - DISPLACEMENT, V - VELOCITY,
A - ACCELERATION)
grid# list - GRIDPOINT IDENTIFIERS(MAX 4)
ex: #Gi, or #CiGj, ECT.

COMMAND: PLOTXLOC,grid# list
ALTERNATE: P-XLOC,grid# list

DESCRIPTION: PLOT THE X COMPONENT OF GRIDPOINT
LOCATION VERSES TIME, WHERE:

grid# list - GRIDPOINT IDENTIFIERS(MAX 4)
ex: #Gi, or #CiGj, ECT.

COMMAND: PLOTY,p,grid# list
ALTERNATE: P-Y,p,grid# list

DESCRIPTION: PLOT THE Y CARTISIAN COMPONENT OF GRIDPOINT
TRANSLATIONAL MOTION VECTOR VERSES TIME, WHERE:

p = D, V, OR A (D - DISPLACEMENT, V - VELOCITY,
A - ACCELERATION)
grid# list - GRIDPOINT IDENTIFIERS(MAX 4)
ex: #Gi, or #CiGj, ECT.

COMMAND: PLOTYLOC,grid# list
ALTERNATE: P-YLOC,grid# list

DESCRIPTION: PLOT THE Y COMPONENT OF GRIDPOINT
LOCATION VERSES TIME, WHERE:

grid# list - GRIDPOINT IDENTIFIERS(MAX 4)
ex: #Gi, or #CiGj, ECT.

COMMAND: PLOTZ,p,grid# list
ALTERNATE: P-Z,p,grid# list

DESCRIPTION: PLOT THE Z CARTISIAN COMPONENT OF GRIDPOINT
TRANSLATIONAL MOTION VECTOR VERSES TIME, WHERE:

p = D, V, OR A (D - DISPLACEMENT, V - VELOCITY,
A - ACCELERATION)
grid# list - GRIDPOINT IDENTIFIERS(MAX 4)
ex: #Gi, or #CiGj, ECT.

COMMAND: PLOTZLOC,grid# list
ALTERNATE: P-ZLOC,grid# list

DESCRIPTION: PLOT THE Z COMPONENT OF GRIDPOINT
LOCATION VERSES TIME, WHERE:

grid# list - GRIDPOINT IDENTIFIERS(MAX 4)
ex: #Gi, or #CiGj, ECT.

HOW TO USE THE POST-PROCESSOR FOR CREATING AND MAINTAINING A DATA BASE

The (x,y) data values of a plot is called a data set. When the user creates a plot from the LATDYN binary output file he has several options available to save the data set for later reference. All the options share the same prefix name STOR for store. Upon command the data is stored in the data base on a file called the STORFILE. This section is divided in 3 parts. The first part pertains to storing the data in the data base, the second part deals with external data, the third part addresses the options available for maintaining the data sets in the data base.

PART 1. STORE DATA SETS IN DATA BASE (STORFILE)

SECTION A. General COMMANDS for storing data into the data base, invoked when and only when a good LATDYN plot has been generated.

COMMAND: STORON

DESCRIPTION: AUTOMATIC STORING OF SUBSEQUENT PLOT REQUEST DATA ON STORFILE (TO BE USED FOR PSi PLOTTING REQUESTS)

COMMAND: STOROFF

DESCRIPTION: CANCEL COMMAND: STORON

COMMAND: STORE

DESCRIPTION: STORE PREVIOUS PLOT REQUEST DATA ON STORFILE

SECTION B. Specific COMMANDS for storing certain data sets into the data base. These commands are independent of plotting, although their format is the same as the plot commands in the previous section,(alphabetic order).

COMMAND: STORC,list of condition numbers
ALTERNATE: S-C,list of condition numbers

DESCRIPTION: SAVE THE STORFILE DATA FOR THE PLOT OF THE LOGICAL STATE OF USER DEFINED CONDITIONS VERSES TIME, WHERE:

list of condition numbers - AS DEFINED BY USER

NOTE: PLOTS ARE SYMBOLIZED TRUE(T) - FALSE(F)

```
    COMMAND:  STORDEF,option,element#
  ALTERNATE:    S-DEF,option,element#

DESCRIPTION:  SAVE ON STORFILE THE DATA FOR THE
              PLOT OF THE INTERNAL FORCES
              IN A BEAM ELEMENT VERSES TIME, WHERE:

              option - AX, FOR AXIAL DEFORMATION
                       TW, FOR RELATIVE TWISTING ANGLE BETWEEN 2 ENDS
                       YA, FOR BENDING ROTATION AT END A(Y DIR)
                       YB, FOR BENDING ROTATION AT END B(Y DIR)
                       ZA, FOR BENDING ROTATION AT END A(Z DIR)
                       ZB, FOR BENDING ROTATION AT END B(Z DIR)
              element# - ELEMENT IDENTIFIERS( MAX 4)
                           ex: #Mj, or #CiMj, or #MjEk ECT.
```

```
    COMMAND:  STORFRC,option,element#
  ALTERNATE:    S-FRC,option,element#

DESCRIPTION:  SAVE ON STORFILE THE DATA FOR THE
              PLOT OF THE INTERNAL FORCES
              IN A BEAM ELEMENT VERSES TIME, WHERE:

              option - AX, FOR AXIAL FORCE
                       TW, FOR TWISTING FORCE
                       YA, FOR BENDING MOMENT AT END A(Y DIR)
                       YB, FOR BENDING MOMENT AT END B(Y DIR)
                       ZA, FOR BENDING MOMENT AT END A(Z DIR)
                       ZB, FOR BENDING MOMENT AT END B(Z DIR)
              element# - ELEMENT IDENTIFIERS( MAX 4)
                           ex: #Mj, or #CiMj, or #MjEk ECT.
```

```
    COMMAND:  STORH,p,hingepoint# list
  ALTERNATE:    S-H,p,hingepoint# list

DESCRIPTION:  SAVE ON STORFILE THE DATA FOR THE
              PLOT OF THE MAGNITUDE OF THE HINGEPOINT
              MOTION VECTOR RELATIVE TO THE GRIDPOINT TO WHICH
              IT IS ATTACHED VERSES TIME, WHERE:

              p = D, V, OR A (D - DISPLACEMENT, V - VELOCITY,
                              A - ACCELERATION)
              hingepoint# list - HINGEPOINT IDENTIFIERS( MAX 4)
                           ex: #GiHk, or #CiGjHk, ECT.
```

```
   COMMAND:  STORLA,list of lin.actuator names
 ALTERNATE:    S-LA,list of lin.actuator names

DESCRIPTION: SAVE ON STORFILE THE DATA FOR
             PLOT OF THE LINEAR ACTUATOR DATA VERSES TIME, WHERE:

             list of lin.actuator names - NAMES AS DEFINED IN
                                          COMMAND: LINACTUATOR
```

```
   COMMAND:  STORLD,option,list of lin.damper names
 ALTERNATE:    S-LD,option,list of lin.damper names

DESCRIPTION: SAVE ON STORFILE THE DATA FOR
             PLOT OF THE LINEAR DAMPER DATA VERSES TIME, WHERE:

             option - RV, FOR RELATIVE VELOCITY
                      DF, FOR DAMPING FORCE
             list of lin.damper names - NAMES AS DEFINED IN
                                        COMMAND: LINDAMPER
```

```
   COMMAND:  STORLS,option,list of lin.spring names
 ALTERNATE:    S-LS,option,list of lin.spring names

DESCRIPTION: SAVE ON STORFILE THE DATA FOR
             PLOT OF THE LINEAR SPRING DATA VERSES TIME, WHERE:

             option - SD, FOR SPRING DISPLACEMENT
                      SF, FOR SPRING FORCE
             list of lin.spring names - NAMES AS DEFINED IN
                                        COMMAND: LINSPRING
```

```
   COMMAND:  STORMAG,p,grid# list
 ALTERNATE:    S-MAG,p,grid# list

DESCRIPTION: SAVE ON STORFILE THE DATA FOR THE
             PLOT OF THE MAGNITUDE OF THE
             MOTION VECTOR VERSES TIME, WHERE:

             p = D, V, OR A (D - DISPLACEMENT, V - VELOCITY,
                             A - ACCELERATION)
             grid# list - GRIDPOINT IDENTIFIERS( MAX 4)
                          ex: #Gi, or #CiGj, ECT.
```

COMMAND: STORQ,list of Q-variable numbers
ALTERNATE: S-Q,list of Q-variable numbers

DESCRIPTION: SAVE ON STORFILE THE DATA FOR THE
PLOT OF THE Q-VARIABLE VALUES VERSES TIME AT:

list of Q-variable numbers - AS DEFINED BY USER

COMMAND: STORQVMAG,list of Q-vector numbers
ALTERNATE: S-QVMAG,list of Q-vector numbers

DESCRIPTION: SAVE ON STORFILE THE DATA FOR THE
PLOT OF THE MAGNITUDE OF A
Q-VECTOR VERSES TIME, WHERE:

list of Q-vector numbers - AS DEFINED BY USER

COMMAND: STORQVX,list of Q-vector numbers
ALTERNATE: S-QVX,list of Q-vector numbers

DESCRIPTION: SAVE ON STORFILE THE DATA FOR THE
PLOT OF THE X COMPONENT OF A Q-VECTOR VERSES TIME AT:

list of Q-vector numbers - AS DEFINED BY USER

COMMAND: STORQVY,list of Q-vector numbers
ALTERNATE: S-QVY,list of Q-vector numbers

DESCRIPTION: SAVE ON STORFILE THE DATA FOR THE
PLOT OF THE Y COMPONENT OF A Q-VECTOR VERSES TIME AT:

list of Q-vector numbers - AS DEFINED BY USER

COMMAND: STORQVZ,list of Q-vector numbers
ALTERNATE: S-QVZ,list of Q-vector numbers

DESCRIPTION: SAVE ON STORFILE THE DATA FOR THE
PLOT OF THE Z COMPONENT OF A Q-VECTOR VERSES TIME AT:

list of Q-vector numbers - AS DEFINED BY USER

COMMAND: STORRA,list of rot.actuator names
ALTERNATE: S-RA,list of rot.actuator names

DESCRIPTION: SAVE ON STORFILE THE DATA FOR
PLOT OF THE ROTATIONAL ACTUATOR DATA VERSES TIME, WHERE:

list of rot.actuator names - NAMES AS DEFINED IN
COMMAND: ROTACTUATOR

COMMAND: STORRD,option,list of rot.damper names
ALTERNATE: S-RD,option,list of rot.damper names

DESCRIPTION: SAVE ON STORFILE THE DATA FOR
PLOT OF THE ROTATIONAL DAMPER DATA VERSES TIME, WHERE:

option - RW, FOR RELAVTIVE ANGULAR VELOCITY
DT, FOR DAMPING TORQUE
list of rot.damper names - NAMES AS DEFINED IN
COMMAND: ROTDAMPER

COMMAND: STORRS,option,list of rot.spring names
ALTERNATE: S-RS,option,list of rot.spring names

DESCRIPTION: SAVE ON STORFILE THE DATA FOR
PLOT OF THE ROTATIONAL SPRING DATA VERSES TIME, WHERE:

option - AD, FOR ANGULAR DEFORMATION
ST, FOR SPRING TORQUE
list of rot.spring names - NAMES AS DEFINED IN
COMMAND: ROTSPRING

COMMAND: STORWMAG,p,grid# list
ALTERNATE: S-WMAG,p,grid# list

DESCRIPTION: SAVE ON STORFILE THE DATA FOR THE
PLOT OF THE MAGNITUDE OF THE
ROTATIONAL MOTION VECTOR VERSES TIME, WHERE:

p = V, OR A (V - VELOCITY, A - ACCELERATION)
grid# list - GRIDPOINT IDENTIFIERS (MAX 4)
ex: #Gi, or #CiGj, ECT.

COMMAND: STORWX,p,grid# list
ALTERNATE: S-WX,p,grid# list

DESCRIPTION: SAVE ON STORFILE THE DATA FOR THE PLOT OF THE X CARTISIAN COMPONENT OF GRIDPOINT ROTATIONAL MOTION VECTOR VERSES TIME, WHERE:

p = V, OR A (V - VELOCITY, A - ACCELERATION)
grid# list - GRIDPOINT IDENTIFIERS(MAX 4)
ex: #Gi, or #CiGj, ECT.

COMMAND: STORWY,p,grid# list
ALTERNATE: S-WY,p,grid# list

DESCRIPTION: SAVE ON STORFILE THE DATA FOR THE PLOT OF THE Y CARTISIAN COMPONENT OF GRIDPOINT ROTATIONAL MOTION VECTOR VERSES TIME, WHERE:

p = V, OR A (V - VELOCITY, A - ACCELERATION)
grid# list - GRIDPOINT IDENTIFIERS(MAX 4)
ex: #Gi, or #CiGj, ECT.

COMMAND: STORWZ,p,grid# list
ALTERNATE: S-WZ,p,grid# list

DESCRIPTION: SAVE ON STORFILE THE DATA FOR THE PLOT OF THE Z CARTISIAN COMPONENT OF GRIDPOINT ROTATIONAL MOTION VECTOR VERSES TIME, WHERE:

p = V, OR A (V - VELOCITY, A - ACCELERATION)
grid# list - GRIDPOINT IDENTIFIERS(MAX 4)
ex: #Gi, or #CiGj, ECT.

COMMAND: STORX,p,grid# list
ALTERNATE: S-X,p,grid# list

DESCRIPTION: SAVE ON STORFILE THE DATA FOR THE PLOT OF
THE X CARTESIAN COMPONENT OF GRIDPOINT
TRANSLATIONAL MOTION VECTOR VERSES TIME, WHERE:

p = D, V, OR A (D - DISPLACEMENT, V - VELOCITY,
A - ACCELERATION)
grid# list - GRIDPOINT IDENTIFIERS(MAX 4)
ex: #Gi, or #CiGj, ECT.

COMMAND: STORXLOC,grid# list
ALTERNATE: S-XLOC,grid# list

DESCRIPTION: SAVE ON STORFILE THE DATA FOR THE PLOT OF
THE X COMPONENT OF GRIDPOINT
LOCATION VERSES TIME, WHERE:

grid# list - GRIDPOINT IDENTIFIERS(MAX 4)
ex: #Gi, or #CiGj, ECT.

COMMAND: STORY,p,grid# list
ALTERNATE: S-Y,p,grid# list

DESCRIPTION: SAVE ON STORFILE THE DATA FOR THE PLOT OF
THE Y CARTESIAN COMPONENT OF GRIDPOINT
TRANSLATIONAL MOTION VECTOR VERSES TIME, WHERE:

p = D, V, OR A (D - DISPLACEMENT, V - VELOCITY,
A - ACCELERATION)
grid# list - GRIDPOINT IDENTIFIERS(MAX 4)
ex: #Gi, or #CiGj, ECT.

COMMAND: STORYLOC,grid# list
ALTERNATE: S-YLOC,grid# list

DESCRIPTION: SAVE ON STORFILE THE DATA FOR THE PLOT OF
THE Y COMPONENT OF GRIDPOINT
LOCATION VERSES TIME, WHERE:

grid# list - GRIDPOINT IDENTIFIERS(MAX 4)
ex: #Gi, or #CiGj, ECT.

COMMAND: STORZ,p,grid# list
ALTERNATE: S-Z,p,grid# list

DESCRIPTION: SAVE ON STORFILE THE DATA FOR THE PLOT OF
THE Z CARTESIAN COMPONENT OF GRIDPOINT
TRANSLATIONAL MOTION VECTOR VERSES TIME, WHERE:

p = D, V, OR A (D - DISPLACEMENT, V - VELOCITY,
A - ACCELERATION)
grid# list - GRIDPOINT IDENTIFIERS(MAX 4)
ex: #Gi, or #CiGj, ECT.

COMMAND: STORZLOC,grid# list
ALTERNATE: S-ZLOC,grid# list

DESCRIPTION: SAVE ON STORFILE THE DATA FOR THE PLOT OF
THE Z COMPONENT OF GRIDPOINT
LOCATION VERSES TIME, WHERE:

grid# list - GRIDPOINT IDENTIFIERS(MAX 4)
ex: #Gi, or #CiGj, ECT.

PART 2. EXTERNAL DATA - Data created by the user or another program.

This feature has been very useful for reading and plotting external data created from another source. The data is stored on the STORFILE, ready to be processed.

```
    COMMAND:  READ,filename,'label',(fortran format)

DESCRIPTION:  READ A USER CREATED FILE AND ADD TO STORFILE

              filename - USERS FILE, DATA MUST BE X-Y PAIRS/RECORD
                         OR X-Y1,Y2,Y3,ETC./RECORD (MAX Y25)
              'label'  - ANY USER DEFINED LABEL ... MUST BE IN' QUOTES
              (fortran format) - OPTIONAL INPUT, 3 OPTIONS EXIST
                 a. A BLANK - BINARY READ
                 b.       * - FREE FIELD READ
                 c. (fortran format)- ANY FORTRAN FORMAT IN ()
                                      IE.(F10.3,3X,F12.4)

        NOTE: A LABEL WHICH IS BLANK (SPECIFIED BY: '' ), IMPLIES THAT
              THE LABEL IS TO BE FOUND ON filename...
              THIS MUST BE IN QUOTES IN THE FILE

              IF MORE THAN ONE Y EXISTS, NEED TO INPUT #, ON REQUEST
              ALL Ys WILL BE STORED ON STORFILE
```

```
    COMMAND:  SINDA,filename

DESCRIPTION:  Read a user written SINDA output file
              and add (time,temperature) profiles for certain
              nodes to the STORFILE

              filename - USERS FILE; OUTPUT FROM THE PROGRAM SINDA.
                         THE FILE MUST BE FORMATTED AS FOLLOWS:

                         RECORD 1:(NN) # OF NODES,(NT) # OF TIME STEPS(FREE FIELD)
                         RECORD 2: LIST OF NODE NUMBERS(1,TO NN;FREE FIELD)
                         RECORD 3 - # OF TIME STEPS: (TIME, TEMPERATURE ARRAY),
                         WHERE: TEMP ARRAY FOR ALL NODES (1, TO NN: 13F6.1)
```

PART 3. DATA BASE MAINTAINENCE COMMANDS

The data base file STORFILE is a local file that exists only during the execution of the program. It is the responsibility of the user through the available commands to maintain and preserve the data, if desired, for a later execution of the POST-PROCESSOR.

```
    COMMAND:  INDEX,s

DESCRIPTION:  LIST PLOT IDENTIFICATION EXISTING ON STORFILE

              s - S, OPTIONAL PARAMETER TO PRINT OUT THE SPECIAL
                  LEGENDS/AXES LABELS, IF THEY EXIST
```

COMMAND: COPY,filename

DESCRIPTION: COPY PLOT DATA FROM A PREVIOUS RUN OF THE POSTPROCESSOR AND ADD TO THE STORFILE

filename - IS AN EXISTING FILE HAVING BEEN CREATED USING COMMAND: SAVE

NOTE: .STR IS THE DEFAULT FILE EXTENSION

COMMAND: DELETE,iplot1,iplot2,...OR..ALL

DESCRIPTION: DELETE X-Y DATA ON THE STORFILE

iplot1,iplot2,etc. - ARRAY INDEX#
ALL - WILL DELETE ALL THE PLOT DATA

NOTE: USE COMMAND: INDEX, FOR ARRAY IDENTIFICATION AND DESCRIPTION

COMMAND: SAVE,filename

DESCRIPTION: SAVE ALL PLOT X-Y DATA EXISTING ON STORFILE

filename - USER DEFINED FILE NAME FOR DATA TO BE WRITTEN

NOTE: DATA CAN ONLY BE READ USING COMMAND: COPY

NOTE: .STR IS THE DEFAULT FILE EXTENSION

COMMAND: WRITE,filename,iplot

DESCRIPTION: WRITE PLOT X-Y DATA EXISTING ON STORFILE ONTO AN EXTERNAL FILE FOR ANOTHER USE

filename - USER DEFINED FILE NAME
iplot - INDEX# OF PLOT DATA ON STOREFILE

NOTE: FORMAT OF DATA IS:
RECORD 1. X-Y PAIR 1, LABEL(1:30) (E16.8,1X,E16.8,1X,(A))
RECORD 2. X-Y PAIR 2, LABEL(31:60)
RECORDS 3-EOF. X-Y PAIRS (E16.8,1X,E16.8)

A WRITE FILE CAN BE READ BY USING COMMAND:
READ,filename,'',*

HOW TO USE THE DATA BASE FOR MANIPULATING AND DISPLAYING THE DATA SETS

Any reference to a data set in the data base is done through the index number which may be accessed through the INDEX command. For example, if 4 plots are requested of the LATDYN data with the STORON option, each of the 4 data sets will have a unique index number. One hundred (100) is the maximum number of data sets that can exist in the data base.

MANIPULATING DATA SETS IN THE DATA BASE

Several operations are available to the user for manipulating data sets. All of these commands result in another data set being created and added to the data base; no data display is invoked. This new data set will have its own unique index number, x and y axis labels, if applicable, and a new legend or curve description defining the particular operation. There are 2 types of operations. One type involves 2 data sets reacting together, the second type involves a mathematical operation on one data set.

TYPE 1.

COMMAND: OVO,iplot1,iplot2

DESCRIPTION: STORE ON STOREFILE THE DATA FROM:

iplot1 - INDEX ON STOREFILE INDICATING THE ARRAY WHOSE ORDINATE BECOMES THE ABSCISSA OF THE NEW STORED CURVE

iplot2 - INDEX ON STOREFILE INDICATING THE ARRAY WHOSE ORDINATE BECOMES THE ORDINATE OF THE NEW STORED CURVE

NOTE: THIS COMMAND ONLY STORES DATA, MUST PSi TO PLOT CURVE

COMMAND: OPO,iplot1,iplot2

DESCRIPTION: STORE ON STOREFILE THE DATA RESULTING FROM ORDINATE1 + ORDINATE2 VS. THE SAME ABSCISSA, WHERE:

iplot1 - INDEX ON STOREFILE INDICATING THE ARRAY OF ORDINATE1

iplot2 - INDEX ON STOREFILE INDICATING THE ARRAY OF ORDINATE2

NOTE: THIS COMMAND ONLY STORES DATA, MUST PSi TO PLOT CURVE

COMMAND: OMO,iplot1,iplot2

DESCRIPTION: STORE ON STOREFILE THE DATA RESULTING FROM ORDINATE1 - ORDINATE2 VS. THE SAME ABSCISSA, WHERE:

iplot1 - INDEX ON STOREFILE INDICATING THE ARRAY OF ORDINATE1

iplot2 - INDEX ON STOREFILE INDICATING THE ARRAY OF ORDINATE2

NOTE: THIS COMMAND ONLY STORES DATA, MUST PSi TO PLOT CURVE

COMMAND: OXO,iplot1,iplot2

DESCRIPTION: STORE ON STOREFILE THE DATA RESULTING FROM ORDINATE1 X ORDINATE2 VS. THE SAME ABSCISSA, WHERE:

iplot1 - INDEX ON STOREFILE INDICATING THE ARRAY OF ORDINATE1

iplot2 - INDEX ON STOREFILE INDICATING THE ARRAY OF ORDINATE2

NOTE: THIS COMMAND ONLY STORES DATA, MUST PSi TO PLOT CURVE

TYPE 2.

COMMAND: CXO,iplot,constant

DESCRIPTION: STORE ON STOREFILE THE DATA RESULTING FROM CONSTANT * ORDINATE VS. THE SAME ABSCISSA, WHERE:

iplot - INDEX ON STOREFILE INDICATING THE ARRAY OF ORDINATE

constant - CONSTANT REAL VALUE

NOTE: THIS COMMAND ONLY STORES DATA, MUST PSi TO PLOT CURVE

COMMAND: LNO,iplot

DESCRIPTION: STORE ON STOREFILE THE DATA RESULTING FROM LN(ORDINATE) VS. THE SAME ABSCISSA, WHERE:

iplot - INDEX ON STOREFILE INDICATING THE ARRAY OF ORDINATE

NOTE: THIS COMMAND ONLY STORES DATA, MUST PSi TO PLOT CURVE

COMMAND: SINO,iplot

DESCRIPTION: STORE ON STOREFILE THE DATA RESULTING FROM SIN(ORDINATE) VS. THE SAME ABSCISSA, WHERE:

iplot - INDEX ON STOREFILE INDICATING THE ARRAY OF ORDINATE

NOTE: THIS COMMAND ONLY STORES DATA, MUST PSi TO PLOT CURVE

IF ORDINATE IS IN DEGREES - USE: CXO TO CONVERT TO RADIANS(PI/180=.01745329)

COMMAND: COSO,iplot

DESCRIPTION: STORE ON STOREFILE THE DATA RESULTING FROM COS(ORDINATE) VS. THE SAME ABSCISSA, WHERE:

iplot - INDEX ON STOREFILE INDICATING THE ARRAY OF ORDINATE

NOTE: THIS COMMAND ONLY STORES DATA, MUST PSi TO PLOT CURVE

IF ORDINATE IS IN DEGREES - USE: CXO TO CONVERT TO RADIANS(PI/180=.01745329)

COMMAND: LOGO,iplot

DESCRIPTION: STORE ON STOREFILE THE DATA RESULTING FROM
LOG(ORDINATE) VS. THE SAME ABSCISSA, WHERE:

iplot - INDEX ON STOREFILE INDICATING THE
ARRAY OF ORDINATE

NOTE: THIS COMMAND ONLY STORES DATA, MUST PSi TO PLOT CURVE

COMMAND: ABSO,iplot

DESCRIPTION: STORE ON STOREFILE THE DATA RESULTING FROM
ABS(ORDINATE) VS. THE SAME ABSCISSA, WHERE:

iplot - INDEX ON STOREFILE INDICATING THE
ARRAY OF ORDINATE

NOTE: THIS COMMAND ONLY STORES DATA, MUST PSi TO PLOT CURVE

COMMAND: POWERO,iplot,constant

DESCRIPTION: STORE ON STOREFILE THE DATA RESULTING FROM
(ORDINATE)**constant VS. THE SAME ABSCISSA, WHERE:

iplot - INDEX ON STOREFILE INDICATING THE
ARRAY OF ORDINATE

constant - CONSTANT REAL VALUE

NOTE: THIS COMMAND ONLY STORES DATA, MUST PSi TO PLOT CURVE

DISPLAY OF DATA SETS IN THE DATA BASE

Once the data exists in the data base the user may display the data sets by listing the (x,y) values on the screen or by plotting. Only one, very powerful, command exists to plot the data sets. The plotting feature was designed for the user to support comparative and parametric studies. Different data types can be mixed and matched on any of the four available graph options. Additional options exists, such as special labels/legends and/or log axis, to be discussed in detail in the following section on PREPARATION OF PRESENTATIONS. These options may be turned on at the user's discretion.

```
   COMMAND:  LIST,iplot,i,j

DESCRIPTION:  LIST X-Y PLOT DATA EXISTING ON STORFILE

              iplot - ARRAY INDEX#
               i, j - OPTIONAL PARAMETERS
            IF: i = blank or *,(j=blank); LIST ALL DATA
                i = #1, j= #2; LIST FROM #1 TO #2,DATA INDEX NUMBERS
                i =  1, j = *; LIST ALL DATA
                i = #1, j = *, LIST FROM #1 TO END OF DATA

      NOTE:  USE COMMAND:  INDEX, FOR ARRAY IDENTIFICATION
                           AND DESCRIPTION
```

```
   COMMAND:  PLOTSTORi,iplot1,iplot2,etc.,s
ALTERNATE:   PSi,iplot1,iplot2,etc.,s

DESCRIPTION:  PLOT X-Y DATA ON THE STORFILE

              i - PLOT OPTION,1,2,3, OR 4 WHERE:
                 i = 1, SINGLE AXIS PLOT(MAX 4 PLOTS)
                 i = 2, SINGLE ABSCISSA, UP TO 3 INDIVIDUALLY
                        SCALED ORDINATES(MAX 4 PLOTS)
                 i = 3, UP TO 3 VERTICALLY STACKED PLOTS
                        (SINGLE ABSCISSA)
                 i = 4, UP TO 4 INDEPENDENTLY SCALED AND LABELED PLOTS
                        IN A 2X2 MATRIX

              iplot1,iplot2,..- ARRAY INDEX# ON STORFILE

              OPTIONAL PARAMETERS:

              s - SLAB, USE THE SPECIAL LEGENDS/LABELS,
                        IF THEY EXIST
                  XLOG, USE LOG SCALE FOR X AXIS ON ALL PLOTS
                  YLOG, USE LOG SCALE FOR Y AXIS ON ALL PLOTS

      NOTE:  USE COMMAND:  INDEX, FOR ARRAY IDENTIFICATION
                           AND DESCRIPTION
```

HOW TO USE THE POST-PROCESSOR FOR PREPARATION OF PRESENTATIONS

Two commands are available to the user for customizing plots. The TITLE command is used to place a title on either a LATDYN plot or a PSi plot from the STORFILE. The title will stay in effect for all plot requests until the user cancels the TITLE. The second command, LABELS, effects PSi plots from the STORFILE, exclusively. This command is self-prompting but requires the user to be aware of the index numbers of the data sets in the data base. Use the INDEX command with the s option to review all the special labels/legends assigned to the data sets. The LABELS command does not invoke plotting; this is done through the PSi command using the SLAB option. As special labels and legends are created by the user, they are saved in a table. The maximum number of entries in this table is 100.

COMMAND: TITLE

DESCRIPTION: CREATE TITLE FOR PREVIOUS (AND SUBSEQUENT) LATDYN FILE or PSi PLOT(S)

NOTE: THIS COMMAND IS SELF-PROMPTING.
TO CANCEL: RE-ISSUE "TITLE" WITH A blank (or, cr)

COMMAND: LABELS

DESCRIPTION: SPECIAL USER DEFINED LEGENDS/LABELS FOR PLOTS STORED ON THE STORFILE

OPTIONS INCLUDE:
- N - NEW ENTRY INTO TABLE OF LEGENDS/LABELS
- D - DELETE AN ENTRY
- C - CHANGE AN ENTRY
- A - APPLICATION OF AN ENTRY TO A STORFILE PLOT
- L - LIST ENTRIES IN TABLE

NOTE: SEE PSi COMMAND TO INVOKE THE SPECIAL LEGENDS/LABELS

SPECIAL FEATURES OF THE POST-PROCESSOR

At the beginning of execution, the POST-PROCESSOR uses an automatic (AUTO) scanning feature for establishing plot data array maximum/minimum values in defining the plotting window. This window may be increased or decreased by the WINDOW command. This command does not effect the data in the data base, only the plot window, and can be thought of as a zoom feature. At any time during execution the user may switch to AUTO. Thus WINDOW and AUTO may be interchanged at the user's discretion.

COMMAND: WINDOW

DESCRIPTION: INITIATE CHANGE OF XMIN,XMAX,YMIN,YMAX PLOT WINDOW (USE AUTO TO RESET). USER WILL BE ADVISED OF CURRENT WINDOW AND STATUS

COMMAND: AUTO

DESCRIPTION: RESET PLOT REQUEST TO ACTUAL DATA VALUES OF XMIN,XMAX

The DEVICE command may be used to change device drivers during execution. Care must be taken in avoiding an unsupported graphics device. Device,10, the Postscript Printer, may, for example, be activated during program execution, directing all plotting commands to the printer and thereby bypassing screen viewing. Command HC, for hard copy on the Postscript Printer, may be invoked at any time after a "good", successful plot on the screen has been completed. The HC command works on one hard copy per one plot, and must be reissued if more are desired.

COMMAND: DEVICE,idev

DESCRIPTION: CHANGE CURRENT DEVICE

idev - INTEGER FROM 1, TO 10, WHERE
- 1 = TEK. 4010
- 2 = TEK. 4014
- 3 = TEK. 4025
- 4 = TEK. 4107
- 5 = TEK. 4115B
- 6 = HP 2647/2648
- 7 = HP GENERIC 7475A PLOTTER
- 8 = POSTSCRIPT/QMS-800 TALL
- 9 = DEC VT240
- 10 = POSTSCRIPT/QMS-800 WIDE

COMMAND: HC

DESCRIPTION: PRODUCE A HARD COPY ON THE POSTSCRIPT/QMS-800 PRINTER OF THE LAST PLOT

NOTE: THIS WILL ONLY BE DONE IF THE LAST PLOT REQUEST WAS ENTERED WITH NO ERRORS AND THE PLOT WAS SUCESSFULLY GENERATED

APPENDIX A.

EXAMPLE PROBLEM

EXAMPLE PROBLEM

```
   COMMANDS:

   OPEN A6DAT    (A6DAT.PLT exists with LATDYN data)
   OPEN C6A3     (C6A3.PLT  exists with LATDYN data)
   STORON
   STORQ,1       (Stored Q data into data base with no plot)
   PLOTC,1,2,3   (See FIGURE 1)
   USE,A6DAT
   PLOTX,D,#G2,#G3,#G4,#G5    (See FIGURE 2)
   STORX,V,#G3
   STORX,A,#G3
   OVO,5,6
   CXO,1,2.5
   LABELS      (Created special y label for index=6)
   LISTFILES
Responce:

   Opened LATDYN files:  A6DAT.PLT
                         C6A3.PLT

   File in use(active) = A6DAT.PLT

   INDEX,S
Responce:

   Curves currently defined ON STORPLOT  =   12
     # #Points   Curve Description Y.X(FileY,FileX)
   ---  ------     ------------------------------
     1     65   Q(Q1).TIME(C6A3)
     2     65   C(C1).TIME(C6A3)
     3     65   C(C2).TIME(C6A3)
     4     65   C(C3).TIME(C6A3)
     5    151   X(D,#G2).TIME(A6DAT)
     6    151   X(D,#G3).TIME(A6DAT)
   SPECIAL LEGEND:(NONE)
   SPECIAL LABELS:(NONE),LABELS DON'T JIVE (D,V,A)

     7    151   X(D,#G4).TIME(A6DAT)
     8    151   X(D,#G5).TIME(A6DAT)
     9    151   X(V,#G3).TIME(A6DAT)
    10    151   X(A,#G3).TIME(A6DAT)
    11    151   X(D,#G3).X(D,#G2)(A6DAT)
    12     65   2.5*[Q(Q1)].TIME(C6A3)

   FIGURES 3-10 are self explanatory.

   EXIT
```

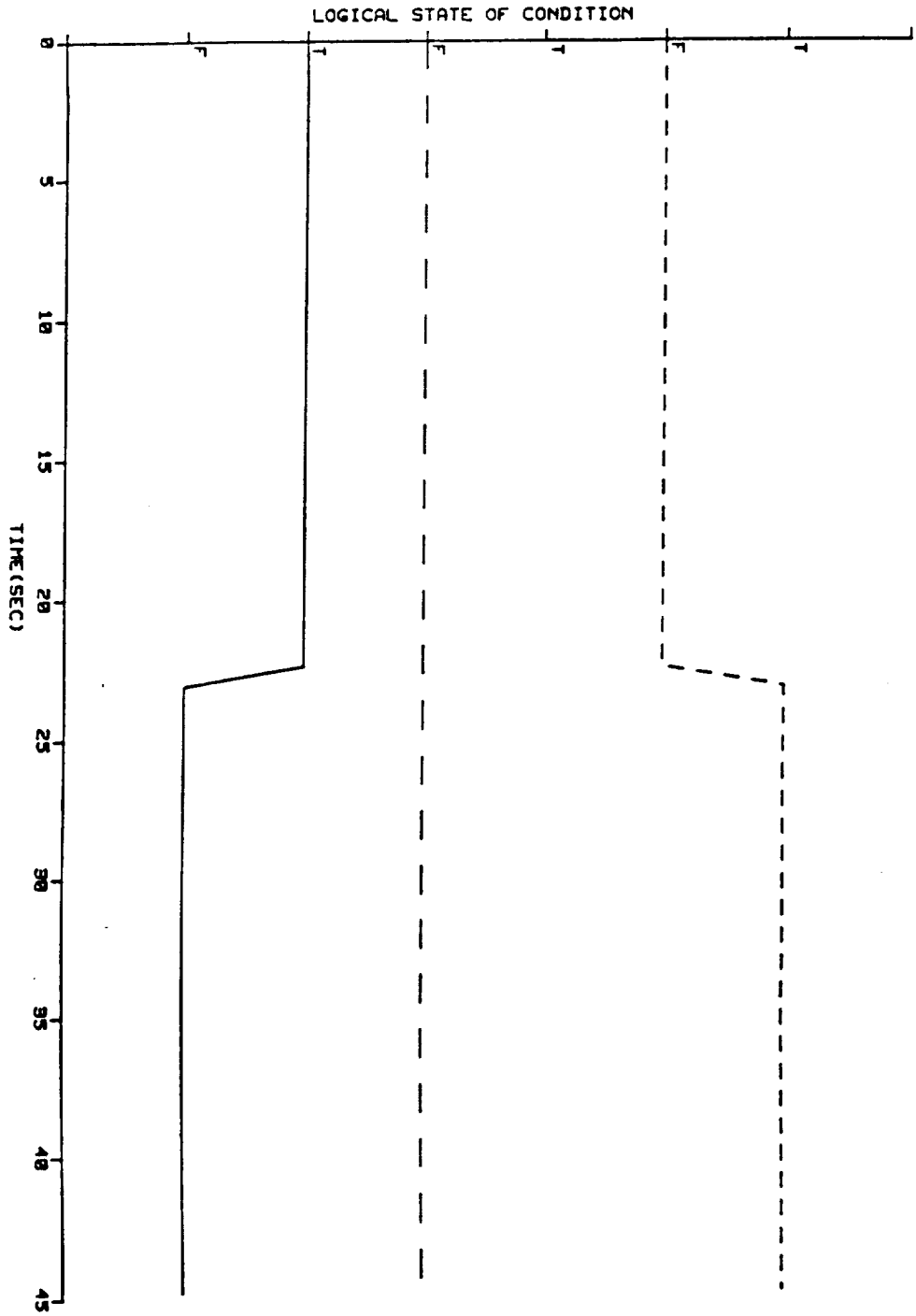
THIS IS AN EXAMPLE OF A CONDITION PLOT
C(C1).TIME(GA3)
C(C2).TIME(GA3)
C(C3).TIME(GA3)
LOGICAL STATE OF CONDITION
TIME(SEC)
0
5
10
15
20
25
30
35
40
45
T
F

FIGURE 1

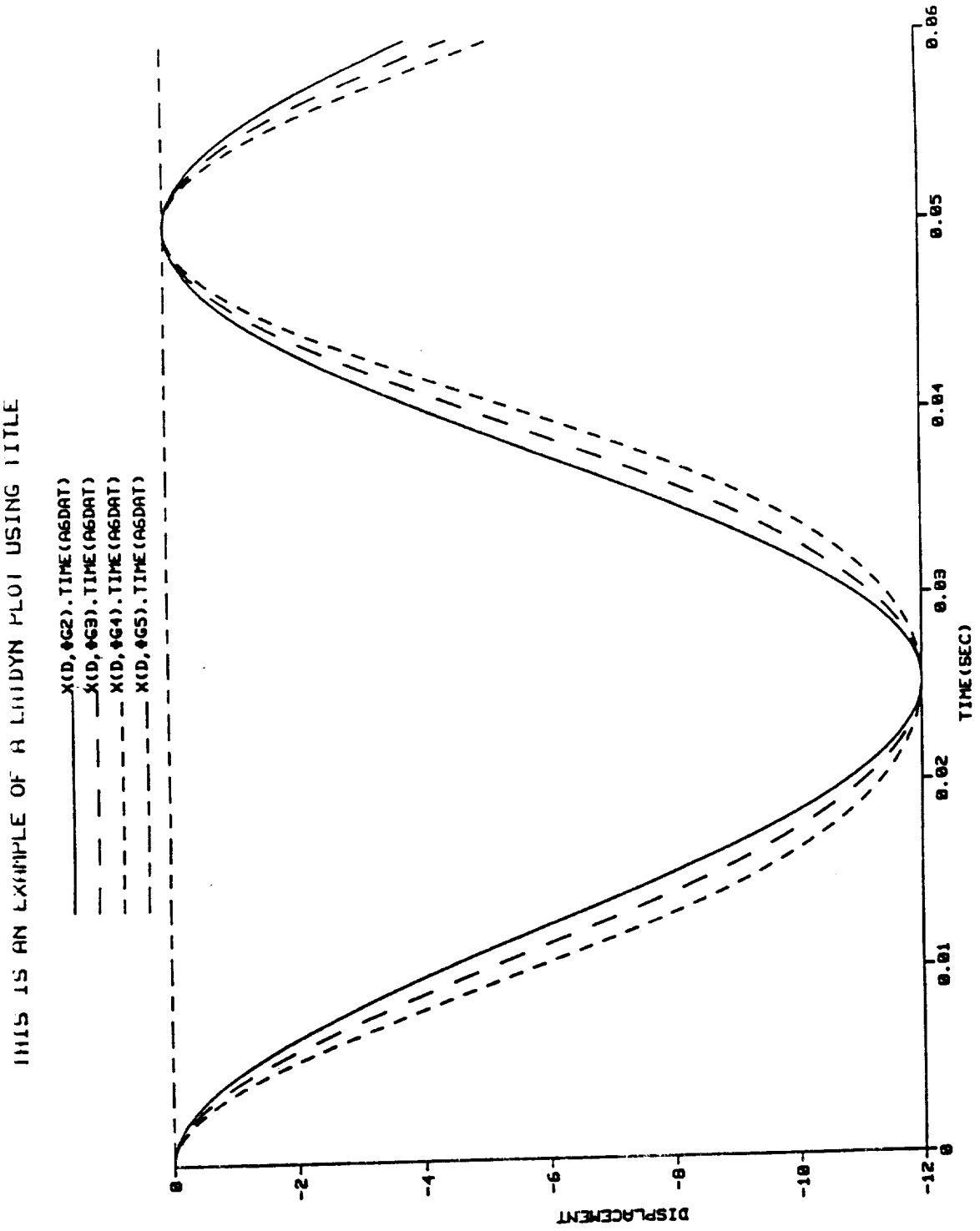
THIS IS AN EXAMPLE OF A LMDYN PLOT USING TITLE
X(D,#G2).TIME(A6DAT)
X(D,#G3).TIME(A6DAT)
X(D,#G4).TIME(A6DAT)
X(D,#G5).TIME(A6DAT)
DISPLACEMENT
TIME(SEC)
0
-2
-4
-6
-8
-10
-12
0.01
0.02
0.03
0.04
0.05
0.06

FIGURE 2

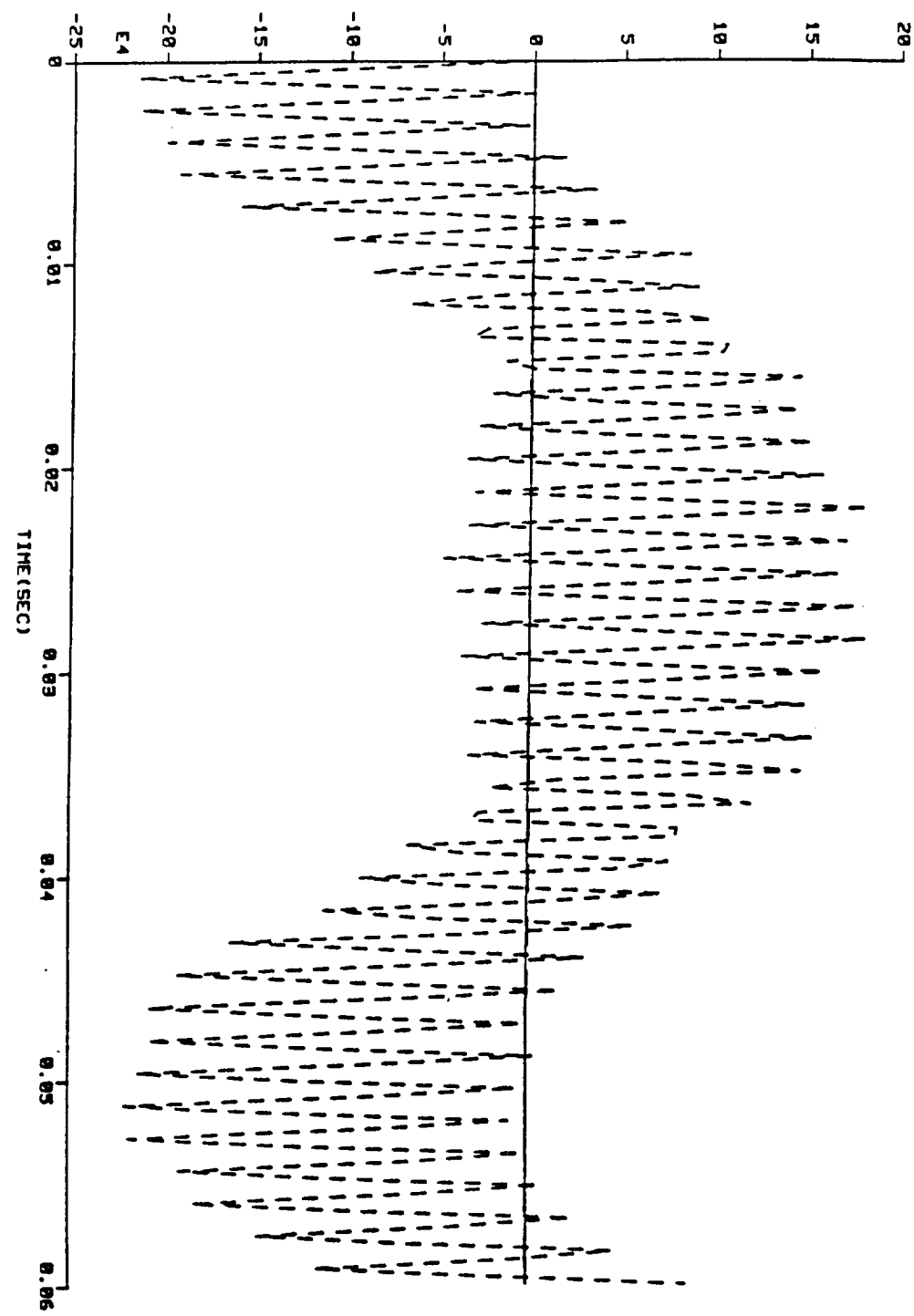
THIS IS AN EXAMPLE OF PSI,6,9,10 COMMAND
X(D,#G3).TIME(#GDAT)
X(V,#G3).TIME(#GDAT)
X(A,#G3).TIME(#GDAT)
E4
TIME(SEC)
-25
-20
-15
-10
-5
0
5
10
15
20
0
0.01
0.02
0.03
0.04
0.05
0.06

FIGURE 3

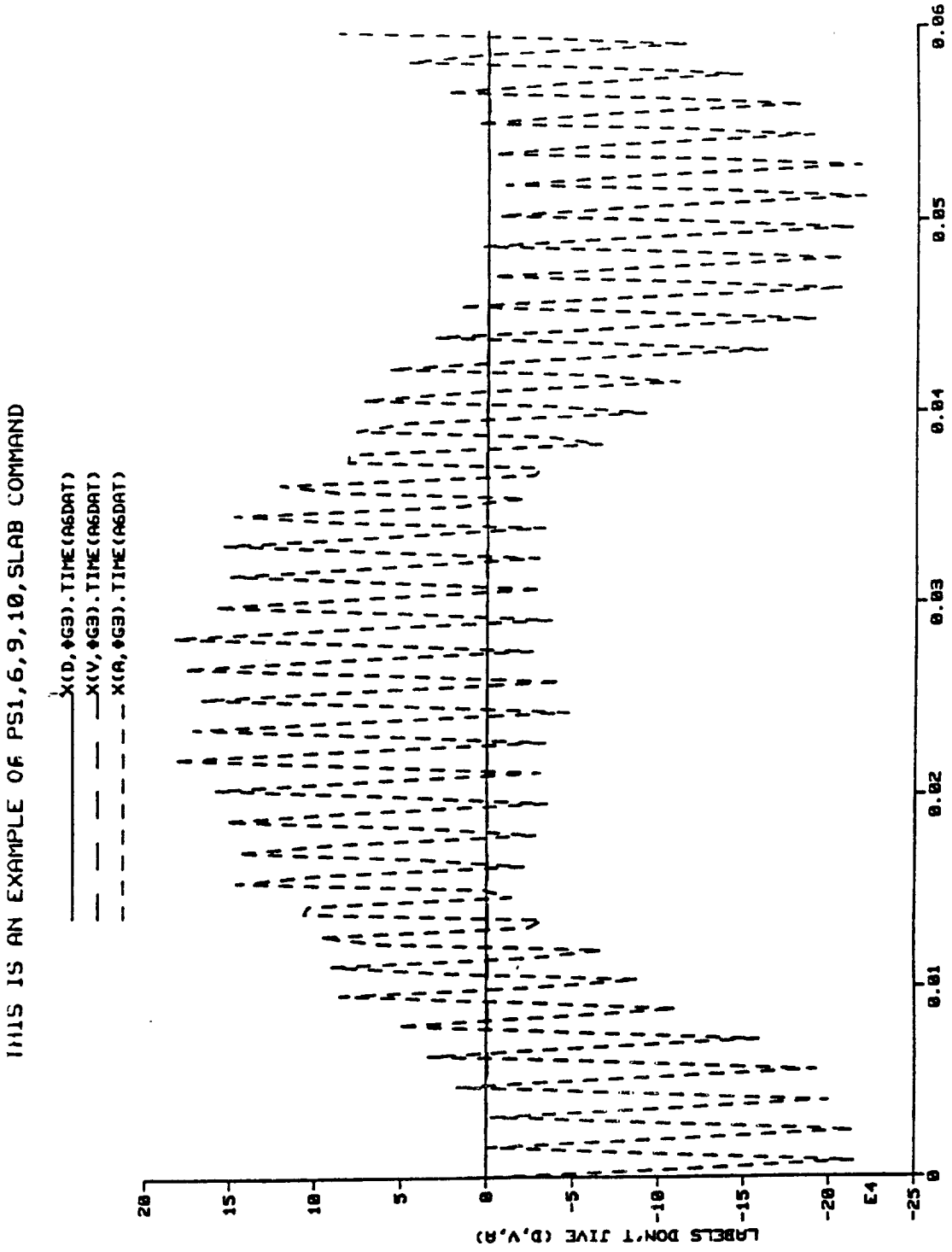
THIS IS AN EXAMPLE OF PS1,6,9,10,SLAB COMMAND
X(D,#G3).TIME(A6DAT)
X(V,#G3).TIME(A6DAT)
X(A,#G3).TIME(A6DAT)
LABELS DON'T JIVE (D,V,A)
TIME(SEC)
E4
20
15
10
5
0
-5
-10
-15
-20
-25
0
0.01
0.02
0.03
0.04
0.05
0.06

FIGURE 4

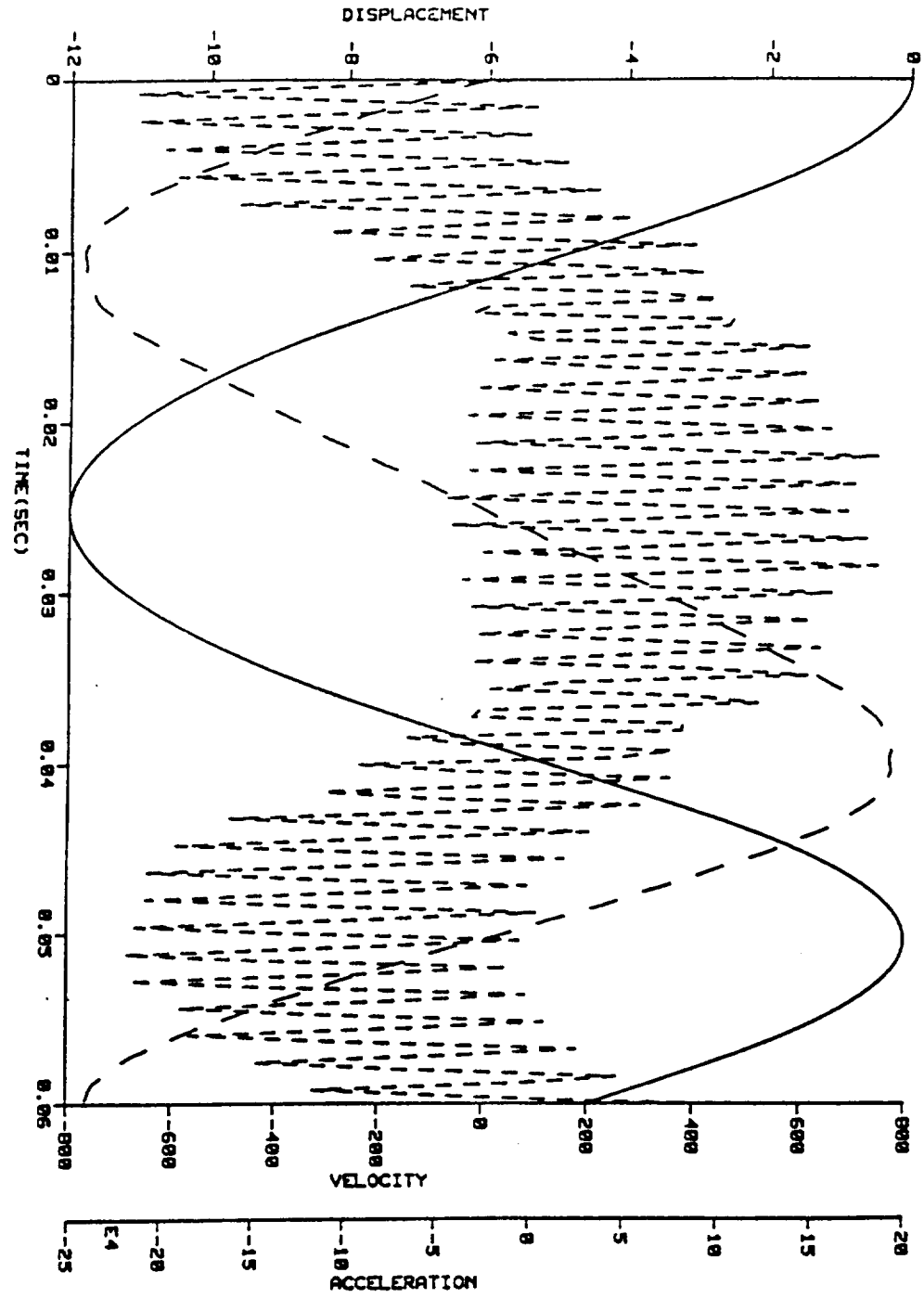
THIS IS AN EXAMPLE OF PS2,6,9,10 COMMAND
X(D,$G3).TIME(A6DAT)
X(V,$G3).TIME(A6DAT)
X(A,$G3).TIME(A6DAT)
DISPLACEMENT
VELOCITY
ACCELERATION
TIME(SEC)
E4

FIGURE 5

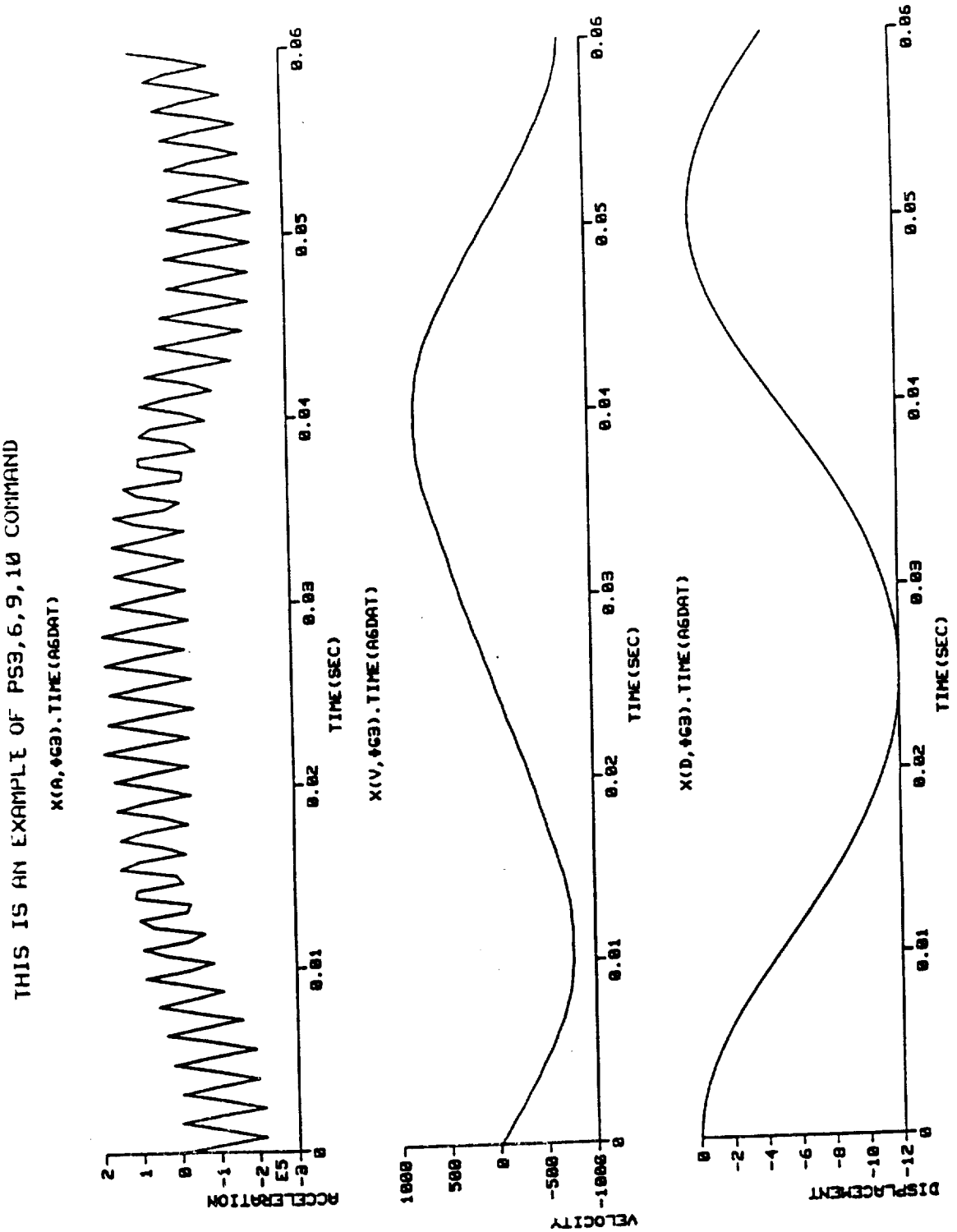
THIS IS AN EXAMPLE OF PS3,6,9,10 COMMAND
X(A,463).TIME(A6DAT)
ACCELERATION
E5
TIME(SEC)
X(V,463).TIME(A6DAT)
VELOCITY
TIME(SEC)
X(D,463).TIME(A6DAT)
DISPLACEMENT
TIME(SEC)

FIGURE 6

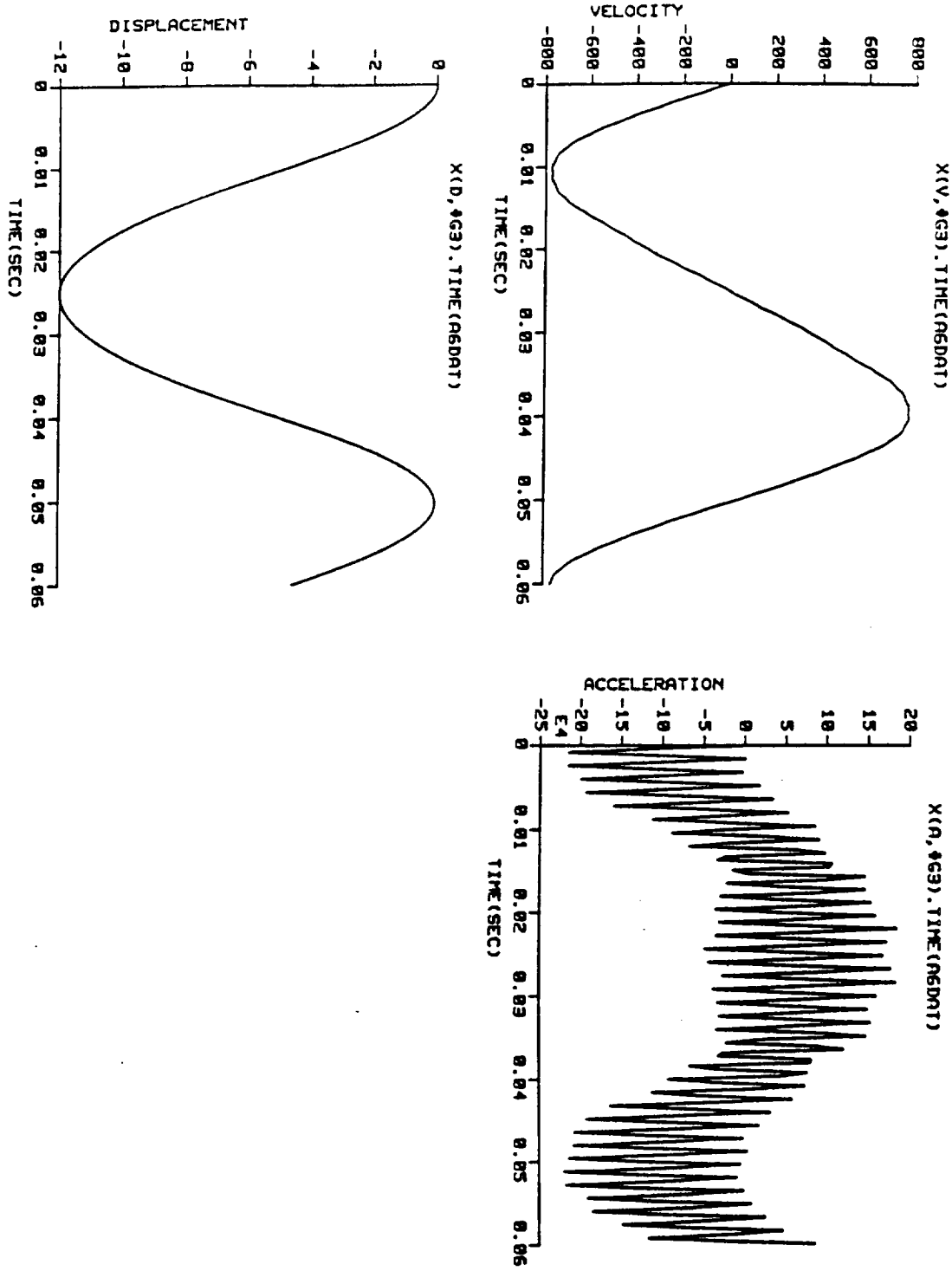
THIS IS AN EXAMPLE OF PS4,6,9,10 COMMAND
DISPLACEMENT
X(D,4G3).TIME(A6DAT)
TIME(SEC)
VELOCITY
X(V,4G3).TIME(A6DAT)
TIME(SEC)
ACCELERATION
E4
X(A,4G3).TIME(A6DAT)
TIME(SEC)

FIGURE 7

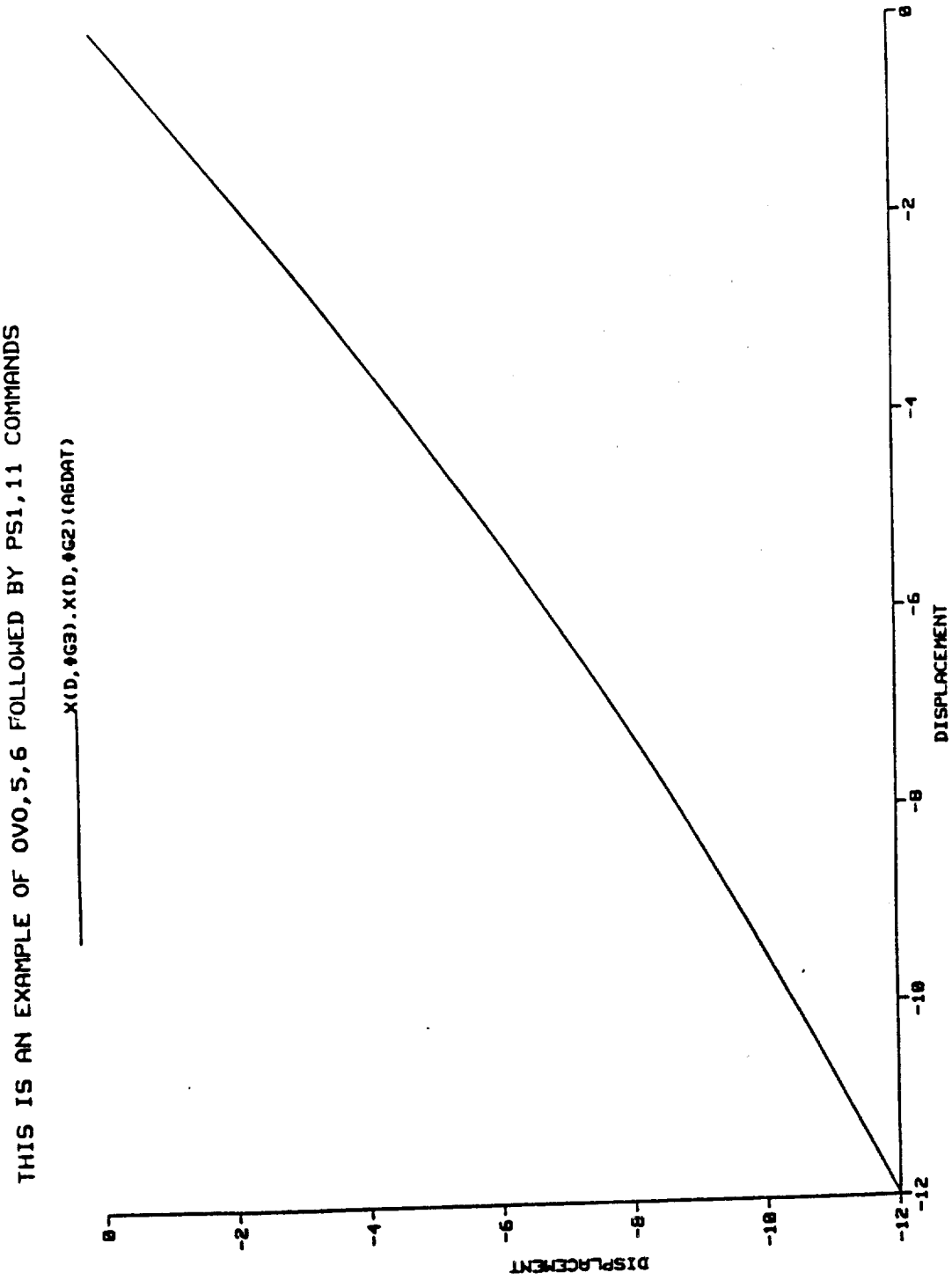
THIS IS AN EXAMPLE OF OVO,5,6 FOLLOWED BY PS1,11 COMMANDS
X(D,@G3).X(D,@G2)(A6DAT)
DISPLACEMENT
0
-2
-4
-6
-8
-10
-12
DISPLACEMENT
0
-2
-4
-6
-8
-10
-12

FIGURE 8

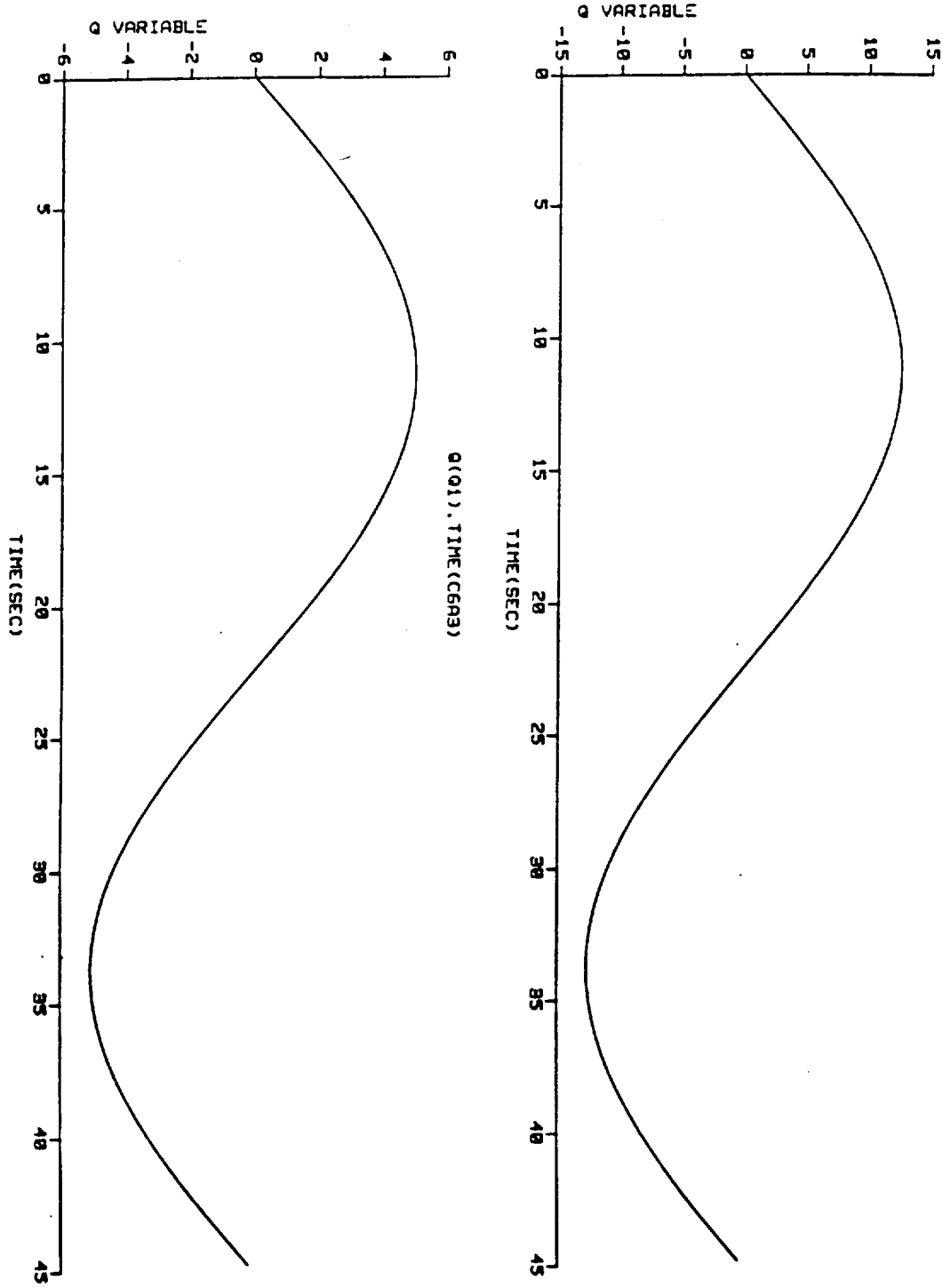
THIS IS AN EXAMPLE OF Q AND 2.5*Q PLOT(USING PS3)
Q VARIABLE
TIME(SEC)
Q(Q1),TIME(CGA3)
2.5*Q(Q1),TIME(CGA3)

FIGURE 9

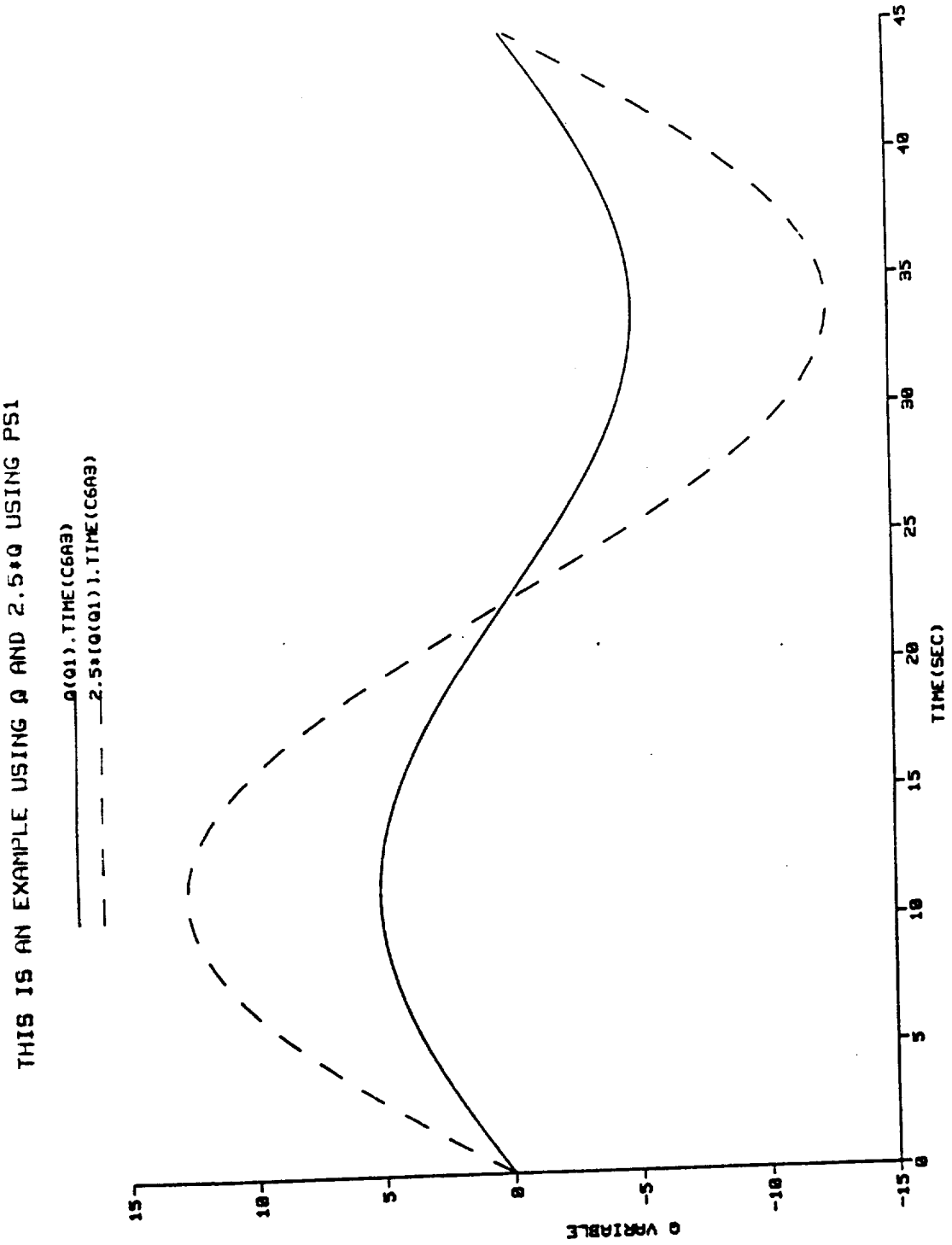
THIS IS AN EXAMPLE USING Q AND 2.5*Q USING PS1
Q(Q1),TIME(C6A3)
2.5*(Q(Q1)),TIME(C6A3)
Q VARIABLE
TIME(SEC)
15
10
5
0
-5
-10
-15
0
5
10
15
20
25
30
35
40
45

FIGURE 10

NASA

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-104126		

4. Title and Subtitle
Large Angle Transient Dynamics (LATDYN) Documentation
Post-Processor Manual

5. Report Date
October 1991

6. Performing Organization Code

7. Author(s)
Maria V. Mitchum
A. Louis Abrahamson

8. Performing Organization Report No.

10. Work Unit No.
505-63-53-10

9. Performing Organization Name and Address
NASA Langley Research Center
Hampton, VA 23665-5225

11. Contract or Grant No.

13. Type of Report and Period Covered
Technical Memorandum

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, DC 20546-0001

14. Sponsoring Agency Code

15. Supplementary Notes
Maria V. Mitchum: Langley Research Center, Hampton, Virginia.
A. Louis Abrahamson: COMTEK, 702 East Woodland Road, Grafton, Virginia.

16. Abstract

The purpose of the post-processor program is to view LATDYN output data in predefined, predetermined formats. The post-processor may be used for plotting data, creating and maintaining a data base of plotting requests, comparing and manipulating data sets in the data base, and the preparation of plots for documentation.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Large Angle Transient Dynamics Control - Structure Interaction Flexible Multi-Body Dynamics	Unclassified - Unlimited Subject Category 39

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	44	A03