**New Folder Name**

40m Support Structure Model

"40m Support Structure Model," by Michael Burka

---

Distributed by S. Whitcomb 4/30/91

A. Abramovici
R. Drever
Y. Gürsel
S. Kawamura
F. Raab
L. Sievers
R. Spero
M. Zucker

cc: C. Akutagawa (Science Library)
R. Vogt
S. Whitcomb
File

# 40m Support Structure Model

**Michael Burka**

**29 April 91**

LIGO T910024-00-R

# 40m Support Structure Model

Michael Burka

29 April 91

## 1 Introduction

An Abaqus finite element model of the 40m support structure is described. Eigenmodes and eigenfrequencies of the structure were predicted, and similar predictions for a shortened support structure proposed for the 40m rebuild were made. The model is fairly simple, and is easily modified to predict the eigenmodes of variant geometries.

## 2 Description of the models

### 2.1 Current support structure

The current stack support structure was modeled as four posts of 2.54 cm diameter and 105 cm length fixed rigidly to the ground at one end and connected by 2.54 cm square bars at the top. The posts define a 24 cm by 32 cm rectangle. Each post was comprised of ten two-node elements, and each cross piece was represented by four two-node elements. Post and cross pieces are made of aluminum. At a height of 54 cm from the ground, springs connect the posts to an infinitely rigid plate which measures $24 \times 32 \times 2.5$ cm. The density of the plate is chosen to make it equal in mass to the 40m prototype stack. The springs each have spring constants of $7.1 \times 10^6$ N/cm in each horizontal direction and $2.8 \times 10^7$ N/cm vertically. These spring constants were chosen to mimic the connection of the actual support structure to the stack.

### 2.2 Modified model

A shortened version of the support structure was also modeled. This model represents a possible geometry for the 40m rebuild. The short stack model was 54 cm in height, with both the cross pieces and the stack springs at the 54 cm level. Element types and cross sections and numbers of elements are the same as in the tall model.

## 3 Results

Table 1 shows the eigenfrequencies found by Abaqus for the two models. The first five eigenmodes represent motion of the stack relative to the support structure. Recall that the stack is represented here by a simple plate, and is attached to the support structure via springs. Since the mass of the stack/plate and the spring constants are the same in the two models, the difference in frequencies must be due to differences in the recoil of the support structure against the stack/plate.

The remaining eigenmodes represent deformations of the support structure itself. For a given mode shape, the eigenfrequency of the shorter stack is always higher. Figures 1-3 show the mode shapes of three of the tall stack modes. The dotted line is the undeformed structure, and the solid line represents the mode shape. The nodes are marked by x's.

The absolute values of the eigenfrequencies are probably not accurate. The model is a simple one, and for the higher frequency eigenmodes the number of elements may not be sufficient to model the mode shapes accurately. What can be learned from this exercise is the mode shapes and their approximate frequencies, and the effect on the eigenfrequency of changing the height of the stack. The effects of other changes, such as increasing the post diameter can be modeled as well.

| No. | 105 cm Stack | 54 cm Stack |
|---|---|---|
| 1 | 4.6 | 4.7 |
| 2 | 5.5 | 5.8 |
| 3 | 5.5 | 5.8 |
| 4 | 9.4 | 9.4 |
| 5 | 11.9 | 11.9 |
| 6 | 19.7 | 57.9 |
| 7 | 19.8 | 58.5 |
| 8 | 34.3 | 79.9 |
| 9 | 92.9 | 289.2 |
| 10 | 95.1 | 314.9 |
| 11 | 97.6 | 330.2 |
| 12 | 97.8 | 339.9 |
| 13 | 98.7 | 344.3 |
| 14 | 114.4 | 389.4 |
| 15 | 115.2 | 394.1 |
| 16 | 120.9 | 412.1 |
| 17 | 250.8 | 468.5 |
| 18 | 254.4 | 700.0 |
| 19 | 260.2 | 705.8 |
| 20 | 265.0 | 766.1 |
| 21 | 266.3 | 880.0 |
| 22 | 288.7 | 920.7 |
| 23 | 290.5 | 976.8 |
| 24 | 299.0 | 1004.6 |
| 25 | 380.1 | 1043.4 |
| 26 | 488.1 | 1055.2 |
| 27 | 510.2 | 1063.5 |
| 28 | 512.1 | 1068.6 |
| 29 | 521.0 | 1206.7 |
| 30 | 548.3 | 1293.4 |
| 31 | 549.8 | 1316.0 |
| 32 | 563.3 | 1579.4 |
| 33 | 576.3 | 1730.4 |
| 34 | 686.9 | 1769.0 |
| 35 | 695.7 | 1834.9 |
| 36 | 733.1 | 1876.0 |
| 37 | 823.0 | 1888.0 |
| 38 | 841.4 | 1920.0 |
| 39 | 878.6 | 1946.8 |
| 40 | 886.9 | 1978.4 |

Table 1. Eigenfrequencies (in Hz)

## 4 Appendix - Graphics routines

The output of the program can be processed into sm graphics with a routine gravity/abaqus/hardcop.f. This is the program which made figures 1-3. The input files, in the same directory, are supp8.fil and supp9.fil. A version called stereo.f makes 3-d pictures of the mode shapes. Figure 4 is an example of such a plot. To see the mode shape in three dimensions, hold the figure about thirty centimeters in front of your face and cross your eyes so that the two images overlap.

## 5 Appendix - Input deck

Listed below is the input deck for the tall stack model. This deck can be found in the MIT computer as gravity/abaqus/supp8.inp. It is provided here so that those who wish to learn abaqus can use and modify this problem as a first exercise.

Abaqus input decks consist of control cards (which always start with a '*'), and data cards which follow their control cards. Comments start with '**'. The first set of instructions defines the node mesh for the model. Then, the elements are defined relative to their respective nodes. Material properties and, for beam elements, beam cross sections follow, then boundary conditions. Once the model has been defined, dynamics steps allow the extraction of dynamical properties.

```
*HEADING
 supp8 - 40m Support Structure
**
** The NODE command defines nodes.  On each data card,
** an (integer) node number is followed by the cartesian
** coordinates of that node.
**
*NODE
   1,-12.,-16.,  0.
  11,-12.,-16., 54.
  21,-12.,-16.,105.
  31, 12.,-16.,  0.
  41, 12.,-16., 54.
  51, 12.,-16.,105.
  61,-12., 16.,  0.
  71,-12., 16., 54.
  81,-12., 16.,105.
  91, 12., 16.,  0.
 101, 12., 16., 54.
 111, 12., 16.,105.
 121, -6.,-16.,105.
 122,  0.,-16.,105.
 123,  6.,-16.,105.
 151, 12., -8.,105.
 152, 12.,  0.,105.
 153, 12.,  8.,105.
 181,-12.,  8.,105.
 182,-12.,  0.,105.
 183,-12., -8.,105.
 211,  8., 16.,105.
 212,  0., 16.,105.
 213, -8., 16.,105.
**
** The NGEN card allows automatic generation of nodes
** to fill in a mesh.  For each data card, the two nodes
** listed must already be defined.  Then, NGEN generates
** all of the nodes in between, and spaces them evenly
** between to two endpoints.
**
*NGEN
   1, 11
  11, 21
  31, 41
  41, 51
  61, 71
  71, 81
  91,101
 101,111
**
** The NSET command takes the listed nodes and labels them
** as belonging to a set of nodes called GROUND.  This is used
** later when boundary conditions are established.
**
*NSET,NSET=GROUND
 1,31,61,91
*NODE
  201,-12.,0.,54.
  202, 12.,0.,54.
**
** The ELEMENT card.  Here we are using B31 elements, which
** are beam elements with one node at each end.  The length of
** the beam is determined by the nodal coordinates, and the height
** and width are defined later.  We are defining these elements
** to belong to an element set called POST.  The data cards
** show an element number followed by the two node numbers which
** the element connects.
```

```
**
*ELEMENT,TYPE=B31,ELSET=POST
 1,1,2
31,31,32
61,61,62
91,91,92
**
** The ELGEN command allows automatic generation of elements.
** The first number is an element number for a previously defined
** element, and the second number is the number of elements to
** be generated.  The element and node numbers are automatically
** incremented for each element.
**
*ELGEN,ELSET=POST
 1,20
31,20
61,20
91,20
**
** The element definitions for the cross-pieces.  These elements
** are in an element set labeled BAR.
**
*ELEMENT,TYPE=B31,ELSET=BAR
 121, 21,121
 122,121,122
 123,122,123
 124,123, 51
 151, 51,151
 152,151,152
 153,152,153
 154,153,111
 181, 81,181
 182,181,182
 183,182,183
 184,183, 21
 211,111,211
 212,211,212
 213,212,213
 214,213, 81
**
** Element definitions for the springs connecting the stack to
** the posts.  A SPRING2 element is a spring connecting two
** nodes and acting in a single coordinate direction which is
** not necessarily along a line connecting the nodes.  The
** data cards contain the spring element number followed by
** the two nodes connected by the spring.
**
*ELEMENT,TYPE=SPRING2,ELSET=SPRINGX
 301, 11,201
 302, 71,201
 303, 41,202
 304,101,202
*ELEMENT,TYPE=SPRING2,ELSET=SPRINGY
 305, 11,201
 306, 71,201
 307, 41,202
 308,101,202
*ELEMENT,TYPE=SPRING2,ELSET=SPRINGZ
 309, 11,201
 310, 71,201
 311, 41,202
 312,101,202
**
** In this model, the stack is modeled as a simple plate, with
** a mass chosen to equal the stack mass and a stiffness chosen
** to be large.
```

```
**
*ELEMENT,TYPE=B31,ELSET=PLATE
 215,201,202
**
** Material definitions.  The ELASTIC and DENSITY control cards
** are suboordinate to the MATERIAL card.  The data following
** the ELASTIC card is Young's modulus and Poisson ratio.
** The number after the density card is the material density.
** Abaqus allows one to use any system of units, so long as
** that system is used consistently throughout the model.
** Here, we use CGS.
**
*MATERIAL,NAME=ALUMINUM
*ELASTIC
6.9E11,0.33
*DENSITY
2.71
*MATERIAL,NAME=STIFSTUF
*ELASTIC
9.E15,0.3
*DENSITY
 10.4
**
** BEAM SECTION cards are used to define the cross sections
** and orientations of beam elements.  For a circular cross
** section, the data card is the radius.  For a rectangular
** cross section the data card contains the two dimensions.
** For elements, that are horizontal, or nearly horizontal,
** the orientation data card may be ommitted.  For other
** beam elements, a card containing the direction cosines
** of the normal to the beam must be included.
**
*BEAM SECTION,MATERIAL=ALUMINUM,SECTION=CIRC,ELSET=POST
 1.27
0.,1.,0.
*BEAM SECTION,MATERIAL=ALUMINUM,SECTION=RECT,ELSET=BAR
2.54,2.54
*BEAM SECTION,MATERIAL=STIFSTUF,SECTION=RECT,ELSET=PLATE
32.,2.5
**
** SPRING properties.  The first data card contains the
** active coordinates at the two spring nodes.  The second
** card is the spring constant.
**
*SPRING,ELSET=SPRINGX
1,1
7.1E6
*SPRING,ELSET=SPRINGY
2,2
7.1E6
*SPRING,ELSET=SPRINGZ
3,3
2.8E7
**
** Boundary conditions.  The bottoms of the posts are fixed in
** all six degrees of freedom.  These nodes are defined to belong
** to node set GROUND, and ENCASTRE means fixed.  The plate which
** represents the stack is supported in such a way that it is unstable
** to rotations about its axis, so it is constrained at its nodes
** (201 & 202) to be unable to rotate about this axis (DOF 4).
**
*BOUNDARY
GROUND,ENCASTRE
201,4
202,4
**
```

```
** Dynamics steps always start with a STEP card and end with an
** END STEP card.
**
*STEP
**
** The FREQUENCY card indicates that eigenfrequencies are to be found.
** The data card gives the number of eigenfrequencies.
*FREQUENCY
40
**
** EL PRINT, FREQUENCY=0 means never print out element information.
**
*EL PRINT,FREQUENCY=0
**
** The NODE FILE card tells the program to store the displaced nodal
** coordinate information in a file for every eigenfrequency.  The
** data value U indicates that coordinate positions are to be stored
** (as opposed to velocities or accelerations).
**
*NODE FILE,FREQUENCY=1
 U
*END STEP
```

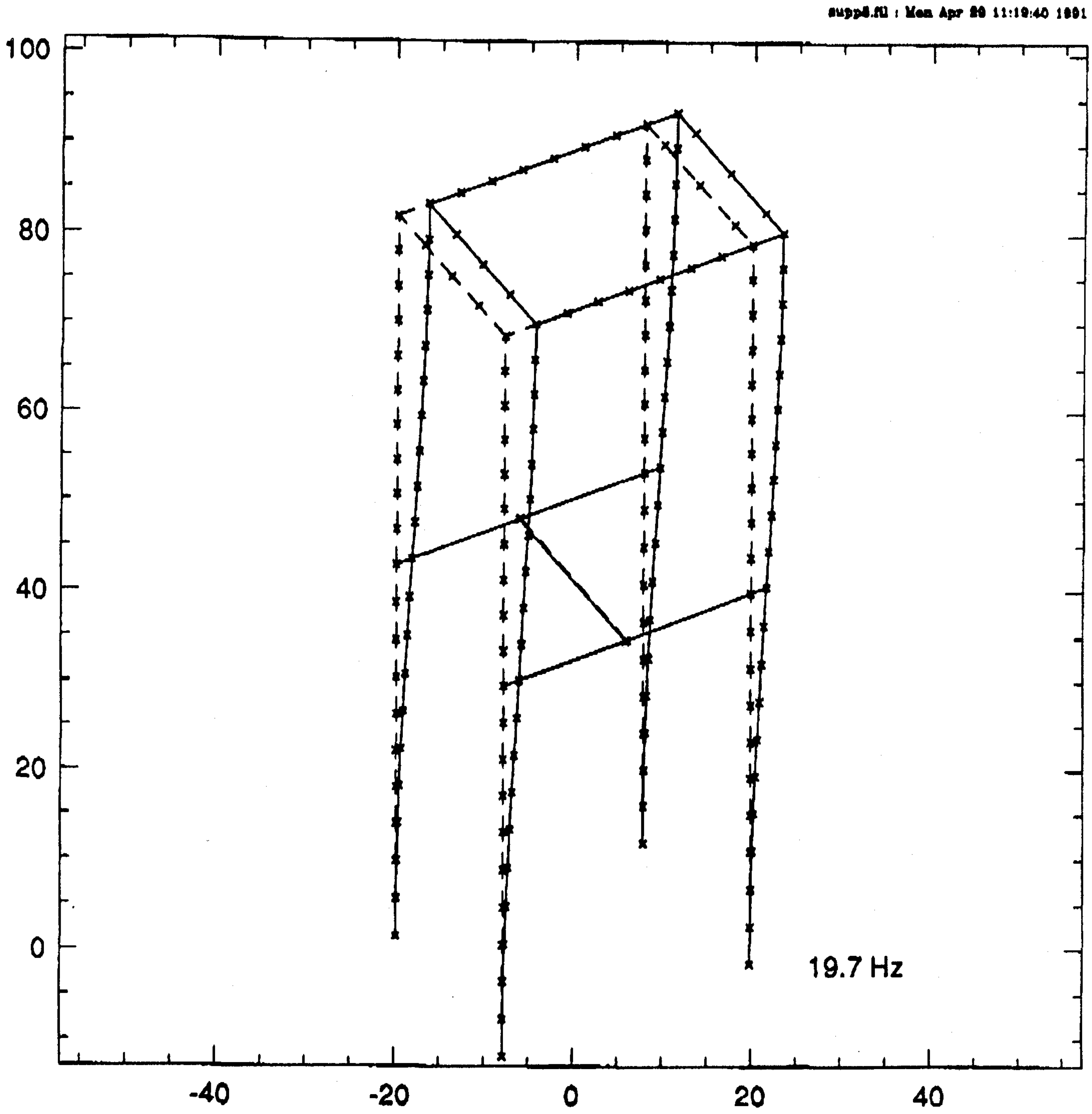
supp8.fil : Mon Apr 29 11:19:40 1991
100
80
60
40
20
0
-40
-20
0
20
40
19.7 Hz

Figure 1.

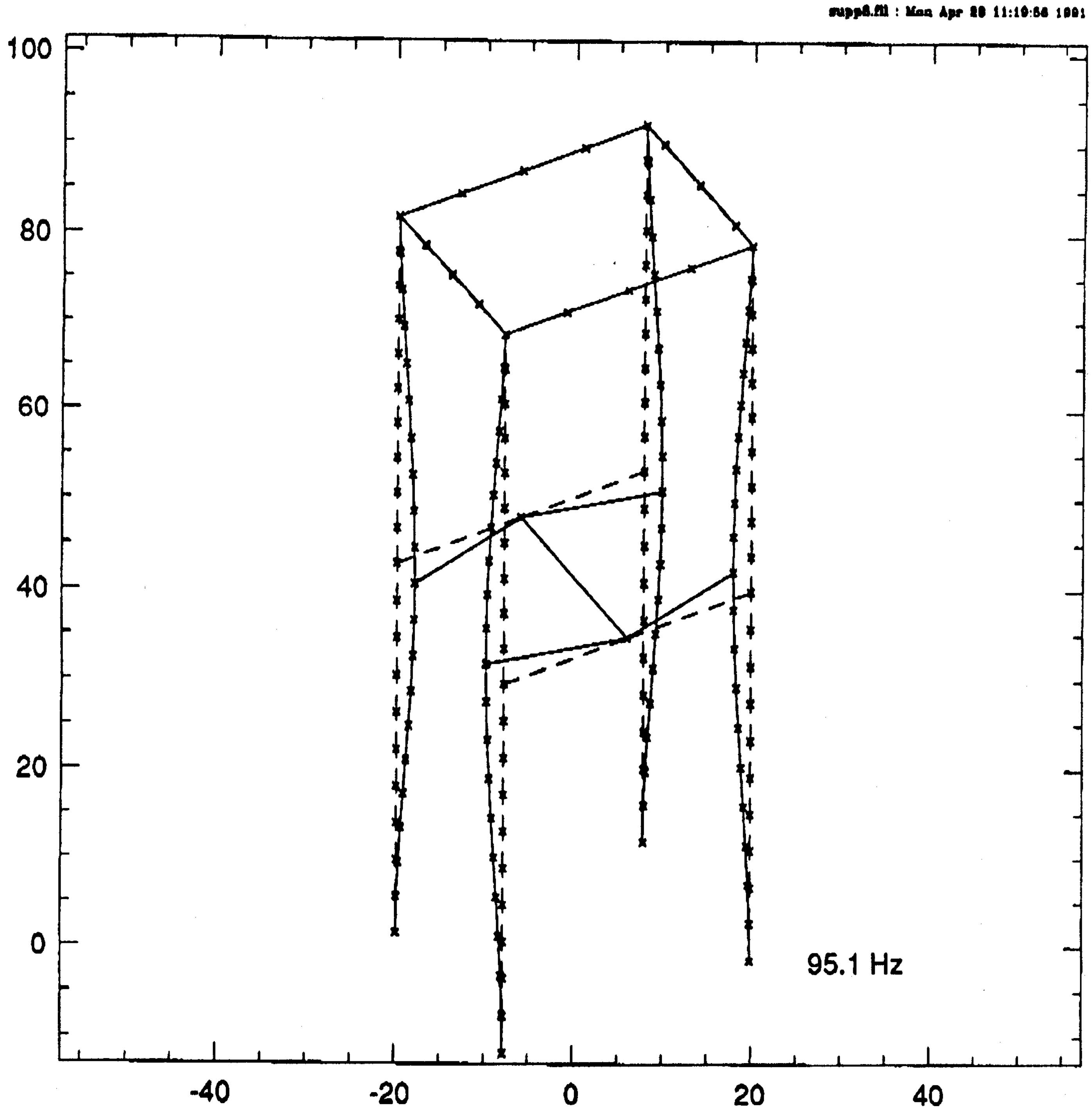
supp8.fil : Mon Apr 29 11:10:58 1991
100
80
60
40
20
0
-40
-20
0
20
40
95.1 Hz

Figure 2.

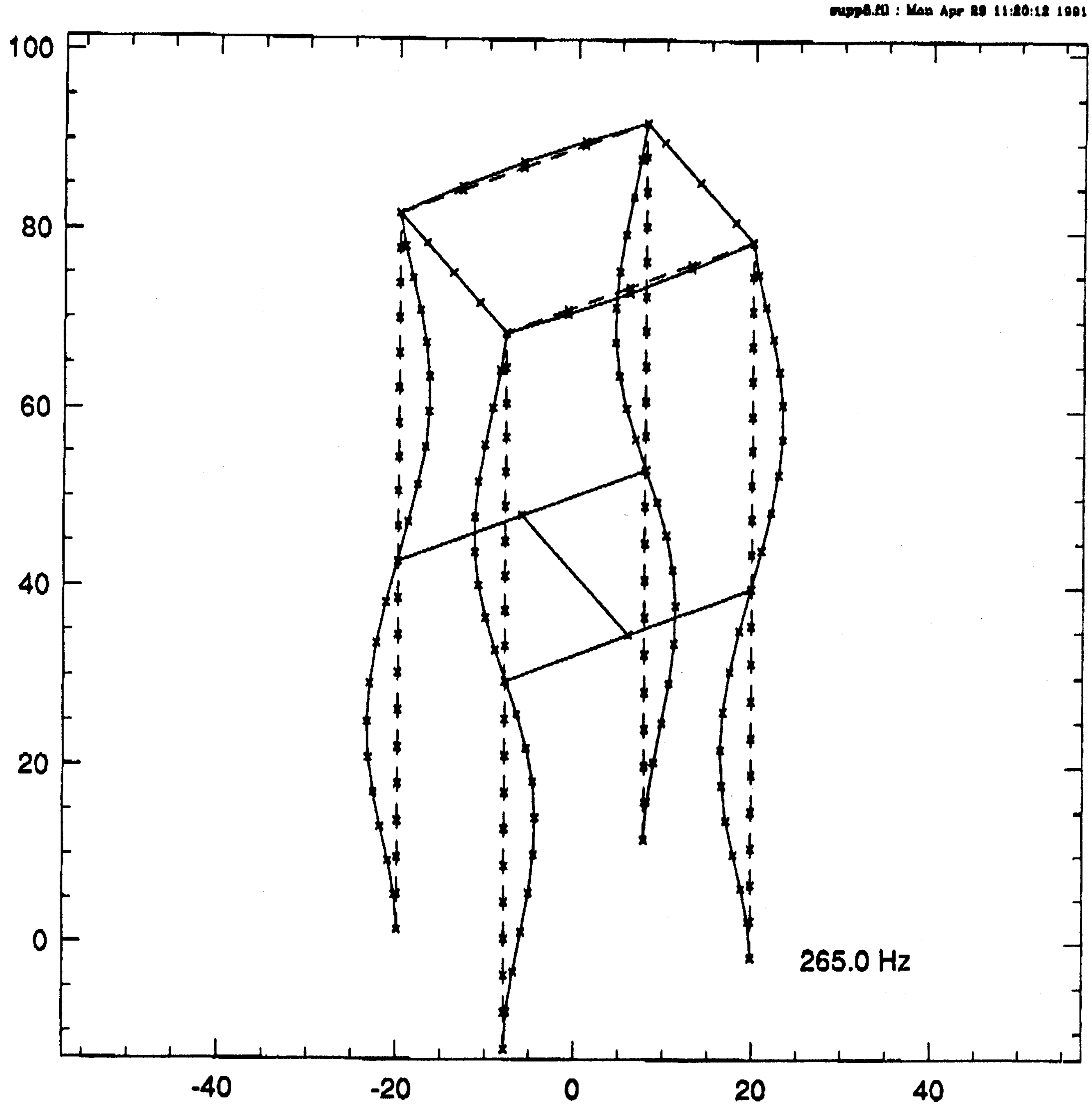
supp8.fil : Mon Apr 29 11:20:12 1991
100
80
60
40
20
0
-40
-20
0
20
40
265.0 Hz

Figure 3.

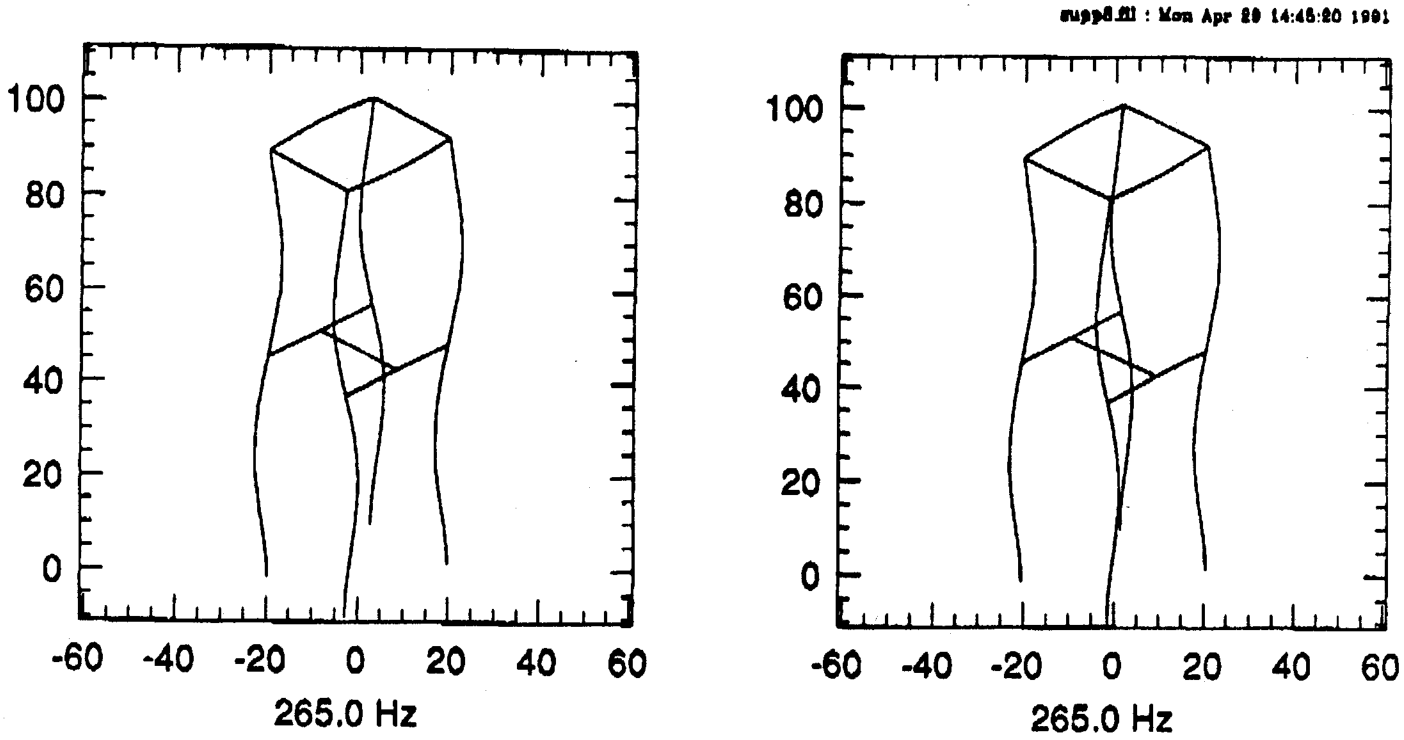
supp8.fil : Mon Apr 29 14:46:20 1991
100
80
60
40
20
0
-60
-40
-20
0
20
40
60
265.0 Hz
100
80
60
40
20
0
-60
-40
-20
0
20
40
60
265.0 Hz

Figure 4.