Report No. UMTRI-85-45

THREE-DIMENSIONAL OCCUPANT DYNAMICS SOFTWARE:
BELT MODEL USE

Prepared by:

D. H. Robbins and R. O. Bennett
Biosciences Division
Transportation Research Institute
The University of Michigan
Ann Arbor, Michigan 48109

Prepared for:

Engineering Mechanics Department
General Motors Research Laboratories
Twelve Mile and Mound Roads
Warren, Michigan 48090

Date:

August 31, 1986

Technical Report Documentation Page

1. Report No.
UMTRI-85-45

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
Three-Dimensional Occupant Dynamics Software:
Belt Model Use

5. Report Date
August 31, 1986

6. Performing Organization Code

7. Author(s)
D. H. Robbins and R. O. Bennett

8. Performing Organization Report No.
UMTRI-85-45

9. Performing Organization Name and Address
Transportation Research Institute
The University of Michigan
Ann Arbor, Michigan 48109

10. Work Unit No. (TRAIS)
388653

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
Engineering Mechanics Department
General Motors Research Laboratories
Twelve Mile and Mound Roads
Warren, Michigan 48090

13. Type of Report and Period Covered
Final Report
(Oct. 1, 1983-Aug.31, 1985)

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

The University of Michigan Transportation Research Institute (UMTRI) conducted a study of seat belt modeling for use with the CAL 3-D crash victim simulation code. The objective was to examine the capabilities of existing software and to improve the codes insofar as possible. The work included:

- Implementation of the CAL 3-D (Version 20) at UMTRI
- Study of the experimental basis for seat belt analytical modeling
- Development of belt slip and submarining models using the existing code
- Development and testing of an analytical basis and software for the transfer of belt material across body surfaces
- Review of the status of the HARNESS sub-program

17. Key Words
Crash victim simulation
Seat belts
Occupant dynamics
Restraint System Modeling

18. Distribution Statement
Unlimited

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of Pages
153

22. Price

CONTENTS

LIST OF TABLES

LIST OF FIGURES

1.0 INTRODUCTION

The University of Michigan Transportation Research Institute has conducted a study of seat belt modeling for use with three-dimensional crash victim simulation codes. The objective was to examine the capabilities of existing software and to improve the codes insofar as possible.

The restraint belt algorithm of the Calspan CVS, Version 20 (Fleck et al. 1981), defines a restraint belt in terms of two anchor points and a fixed point on the surface of a contact ellipsoid. The belt is presumed to lie entirely within the belt plane defined by these three points in space. The two anchor points are restricted to be attached to any one segment (usually the vehicle) while the contact ellipsoid, and along with it the fixed point, is restricted to be attached to any other segment (usually one of the body segments). Up to eight belts are allowed but each is totally independent of the other.

British Leyland developed a model for slipping between two belts for the case in which one anchor of one of the belts was coincident with one anchor of the other belt (Newman et al. 1981). Butler et al. (1980, 1983) developed a major new sub-program (HARNESS) which allowed both slipping of the belt over the surface of the occupant and penetration of the belt into the body.

General Motors Research Laboratories made modifications to the Version 20 code particularly in the area of problem size. The University of Michigan Transportation Research Institute reorganized the output sections to reduce the amount of storage required and simultaneously installed some of the output features from the updated version of Version 18A (Bennett et al. 1982).

The remainder of the report describes the work conducted during the project:

- Development of experimental basis for seat belt modeling
- Development of belt slip and submarining models using the existing codes

- Development and testing of an analytical basis for the transfer of belt material across body surfaces
- Review of the status of the HARNESS sub-program
- The UMTRI implementation of the CVS

2.0 BACKGROUND AND BASICS FOR SEAT BELT MODELING

2.1 Review of Impact Test Data

In order to understand the physical phenomena which should be incorporated in a model of seat belt interaction with a motor vehicle occupant during a crash, a review was conducted of impact test data. The two questions posed were:

1. Did the belt move or slip over the surface of the test subject; and,

2. Did the belt compress the chest structures during load application?

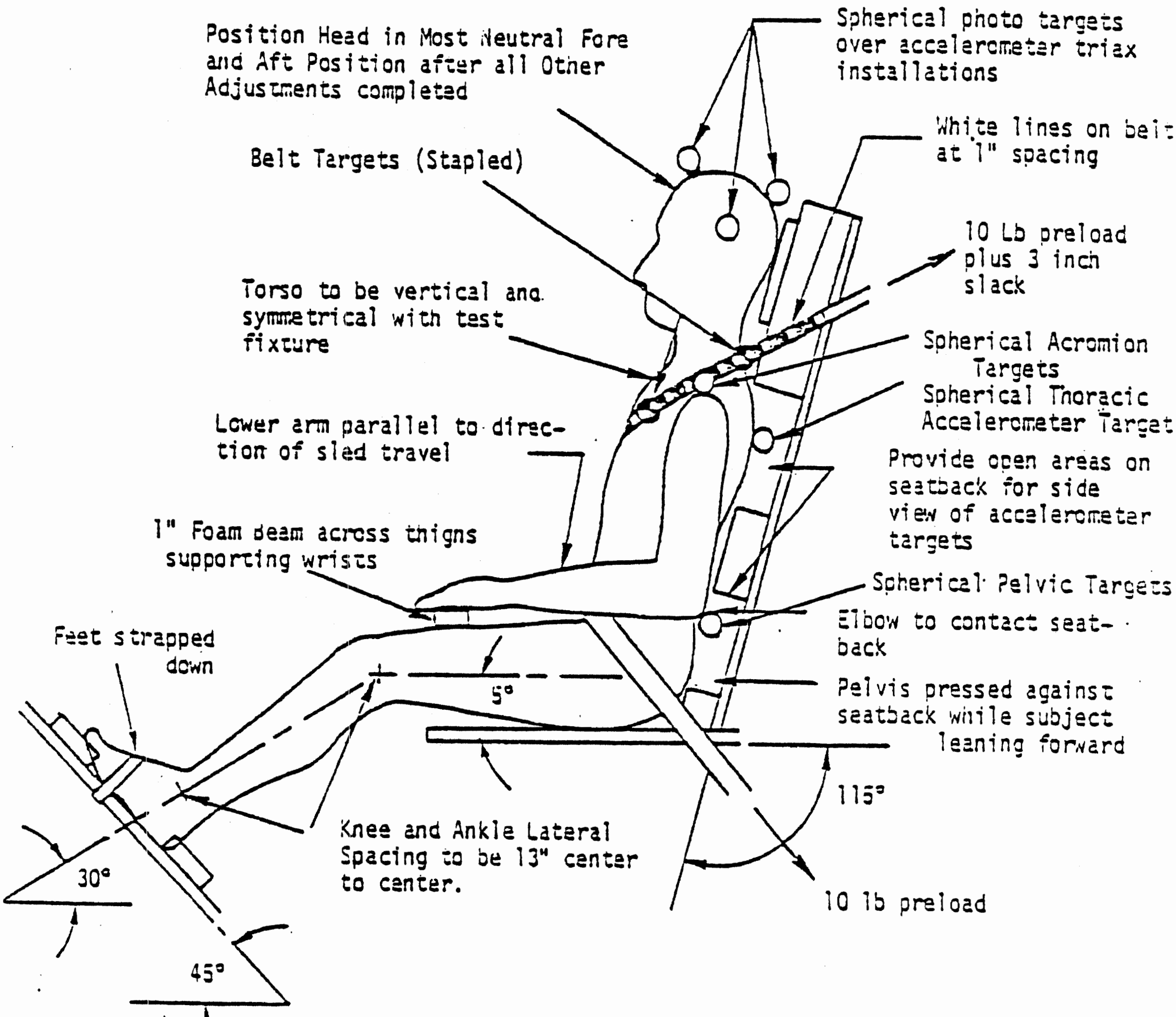
Two papers by Biomedical Science Department staff at the General Motors Research Laboratories were reviewed in this context. A Part 572 dummy was used in a study of test dummy interactions with a shoulder or lap belt by Viano and Culver (1981). These frontal impact tests were conducted with the thorax skin removed in order to better visualize the geometry of the interactions. The major phases of the dummy's interaction with the restraint system involved: "(1) forward movement of the dummy and take up of belt slack; (2) initial belt slip and adjustment to the thoracic structure; (3) inertial acceleration of the thorax with primarily planar thoracic compression; and, (4) substantial non-symmetric deformation of the thoracic structure as the chest rotates about the belt. . . . " An independent review of these movies revealed some slipping of the belt across the surface of the thorax during the loading phase. This was particularly marked where the lower portion of the shoulder belt interacted with the lower right side of the dummy's rib cage. Also, the stalk where the three-point attachment is routed to vehicle structures is observed to rotate.

Horsch (1980) reported a study of occupant dynamics as a function of impact angle and belt restraint. Tests were conducted at impact angles of 0, ±30, ±45, ±60, and ±90 degrees. The test velocities were 35 km/hr. In "opposite side" impacts, the body escaped from under the torso belt at 60 and 90 degrees but importantly it was noted that much of the impact energy had already been absorbed.

The observation of belt sliding across the surface of the Part 572 thorax in the frontal impacts caused consternation among the present researchers. The reason was that a foundation of the original Calspan 3-D CVS model was attachment of the belt at a fixed point on the thorax. Because of this a further review of test movies using Hybrid III dummies and cadavers was conducted. The data were obtained from the whole body response study, a major effort at UMTRI funded by General Motors and reported by Alem et al. (1977). The test buck was provided by General Motors and test velocities at 16, 22, and 33 mph were used. A schematic of the test setup is shown in Figure 2.1.

A quick look at the cadaver movies revealed what appears to be considerable slipping of the belt over the torso, perhaps 10 inches or more. In the eight side views of a low-level test (16 mph), there seemed to be some sliding of the torso belt as it penetrated down and into the lower right-hand side of the subject. From the left side, the belt slid over the shoulder. This apparent extensive amount of slipping occurred at the same time the belt appeared to be compressing the center of the torso. This was confirmed in a front view of the test. In this view, the belt appeared to slide a little bit initially as if the belt were using up slack while adjusting to the surface of the body before applying any significant loads. As the load was applied, there was a tremendous compression of the chest, during which time the belt was pocketed by the surrounding tissue. No slipping along the belt line was observed. Side slipping was virtually impossible because of the pocketing. It should be noted that the shoulder complex rotated around the thorax and belt to accentuate this effect. The same pattern of behavior was observed in the higher velocity impacts (22 and 33 mph).

The tests at UMTRI using the Hybrid III were conducted using General Motors equipment (test buck and dummies). Tests were reviewed at three velocities (16 mph, 22 mph, 33 mph). With respect to belt slipping, the results were similar to those observed in the cadaver tests. The major difference was the lack of shoulder complex forward



Initial Belt Lengths: R. Lap = 12"
(Center to Center of Ends) L. Lap = 32.5"
Shoulder = 42"

Femur Target Spacing:

Right Side = in.

Left Side = in.

Belt Sequence: L. Lap, R. Lap, Shoulder
(Out from Subject)

Belt End Orientation: Away, Away, Toward
(Ref. To Subject)

Figure 2.1. Schematic of Test Setup

rotation and the general apparent excessive stiffness of the thoracic structure and spine. Some chest compression was observed.

In conclusion, the following observations can be made in answer to the questions posed at the beginning of this section.

1. Most of the slippage of the belt over the surface of the subject was observed to occur in the initial phase of the test in order to take up slack which may exist in any of the belt segments.

2. Apparent slippage is observed in the upper and lower torso belt segments as body loads are applied. This is primarily the result of thorax compression and, to a lesser extent, belt stretch.

3. During application of a frontal impact load to a subject having a well-positioned belt system, it appears that there is no slippage for one point on the thorax.

4. While the belt interaction with Hybrid III and cadaver test subjects are similar, the Hybrid II response appears to be rather different.

2.2 Limitations of Existing BELT Code

In the BELT subprogram, each of the restraint belts is assumed to lie in a plane defined by two anchor points attached to a segment (usually the vehicle) and by a fixed point on a contact ellipsoid rigidly attached to some other segment (see Figure 2.2). The calculation of the belt length from the fixed point to the two anchor points is done separately. The friction of the contact between the belt and the segment ellipsoid may be assumed to be either zero or infinite. In the zero friction option the total belt length is used to compute the strain and a single force-strain history is used to determine the force which is applied equally at each of the tangent points. In the infinite friction option each of the partial belt lengths (one from the fixed point to anchor point A and the other from the fixed point to anchor point B) are treated independently. Separate force-strain histories are carried for each part resulting in different forces. It is assumed that the force-strain functions are defined in such a manner as to account for deformation of the contact ellipsoid.

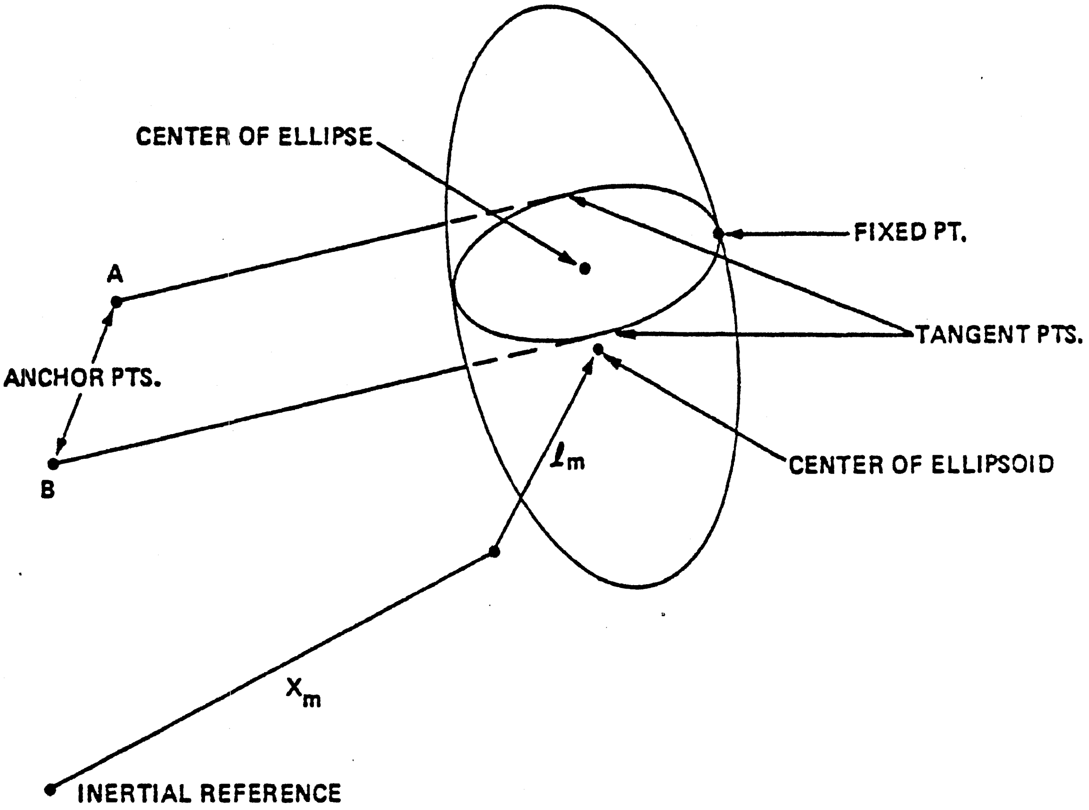


Figure 2.2. Restraint Belt Geometry

Based on the observations put forth in Section 2.1, the BELT code is deficient in three ways:

1. There is no capability to transfer material from one side of the belt segment to the other. The only mechanism is force equalization which, rather than feeding slack material from one side to the other, computes the average of the non-zero force on one side with the zero force on the slack side. Based on the observation of adjustment of the belt to fit the shape of the occupant just discussed in Section 2.1, this assumption is inadequate.

2. It is not possible to model a three-point belt. In other words, it is not possible to connect two belts, such as those defined in Figure 2.2, together in order to represent the torso and lap sections of a seat belt assembly. Even if this were possible, the rotation of the stalk attached to the D-ring could not be represented.

3. It is not possible to allow migration of the fixed point on the segment ellipsoid. Although this assumption appears to be good for frontal impact (in the case where there is little transfer of slack belt material), it is not valid for opposite side lateral or oblique impacts where the occupant can slip completely out of the belt.

3.0 BELT SLIP AND SUBMARINING MODELS USING EXISTING CODES

This section of the report discusses the use of the original BELT software to simulate sliding of the belt over body surfaces and penetration of the belt into body surfaces during dynamic loading. Section 3.1 shows the modeling differences between applications to various conceptual problems. Section 3.2 summarizes various simple restraint models and their effectiveness. Section 3.3 presents a full-scale example of a submarining model which appears to work well and which should be laboratory tested for general use.

3.1 Sequence of Seat Belt Model Development

Figure 3.1 is a schematic which shows a variety of modeling concepts including:

- Current BELT code
- Modifications to the code generated by British Leyland
- Further new modifications by UMTRI, made during the present project, suggested for complete implementation

The two drawings in the upper-left section of the figure represent the current state. The two ends of any belt segment are fixed to the vehicle. The belt itself is constrained to conform to an ellipsoid attached to the occupant. This belt is rigidly attached to a point on the surface of the ellipsoid. The forces in the two separate elements of the belt can either be equalized or totally independent. To model a three-point belt system, two independent sets of belts must be used. Each of the four belt ends is fixed independently to the vehicle and each pair has a fixed attachment to a body ellipsoid. This concept is most relevant to the older four-point belt system.

The modifications made to the BELT code by British Leyland (Newman et al. 1981) have the effect illustrated in the upper right portion of Figure 3.1. This allows the transfer of material between pairs of belts and is intended to represent this effect at the D-ring. However, no slippage across the occupant is allowed as the fixed points on the ellipsoid surfaces are still required.

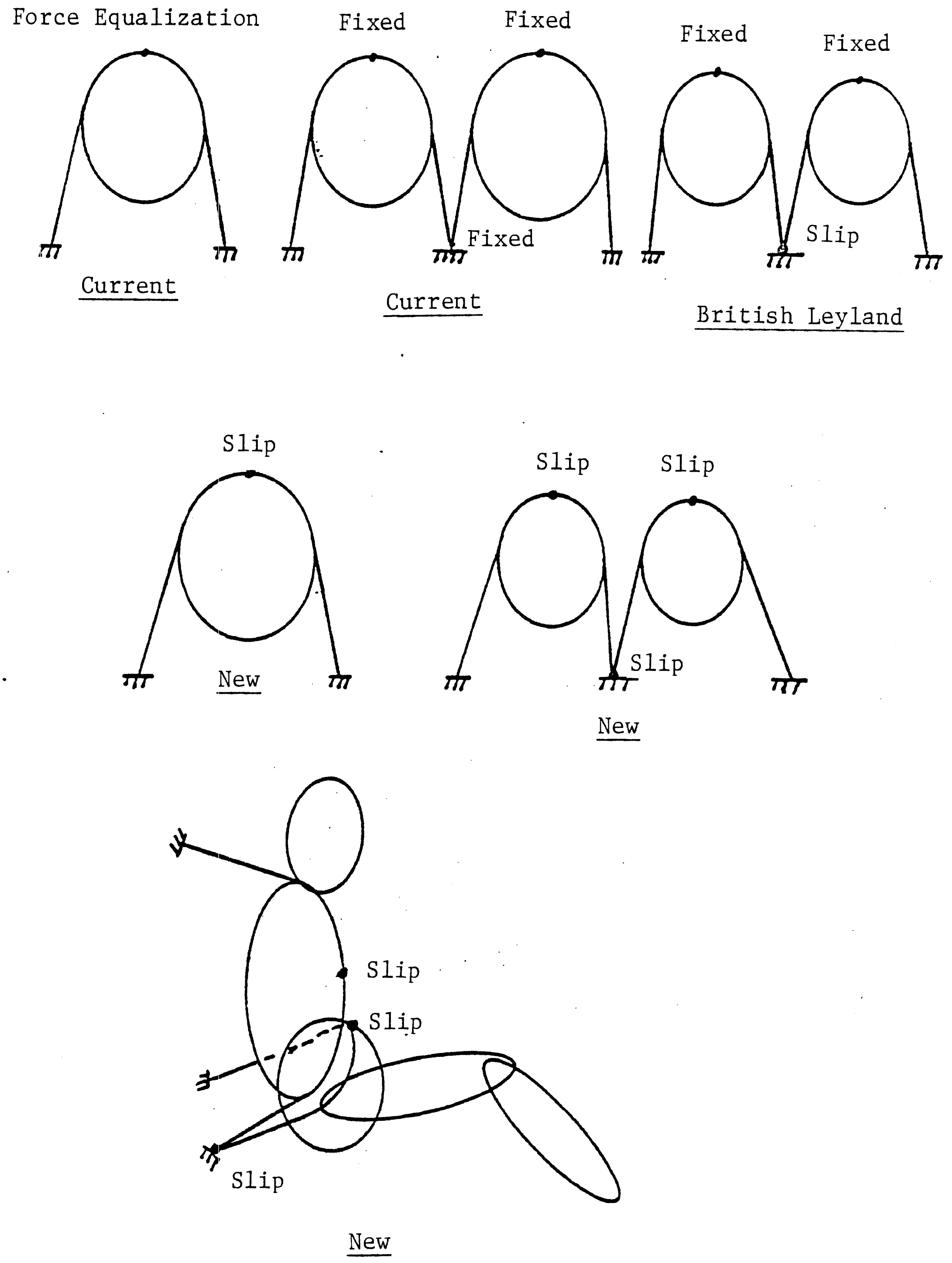


Figure 3.1. Sequence of Belt Models

The new UMTRI concepts, implemented and tested at the subroutine level, but not yet integrated into the CAL 3-D code, are illustrated in the bottom three drawings in Figure 3.1. In these cases, slipping of material is allowed across slip points which are rigidly attached to the surface of body ellipsoids, as before. Further, slipping can occur at the point where pairs of belts are connected, as is the case with the British Leyland improvements. Because of this general capability to slip, or transfer material, between all the belt segments, the take-up of initial slack discussed in Section 2 of this report is accommodated. The potential application to the three-point belt system is shown in the bottom section of Figure 3.1. It should be noted, however, that mobility of the stalk into which the D-ring is inserted is not accommodated.

More advanced belt modeling concepts are discussed in Section 5 where the HARNESS sub-program is described.

3.2 Test Cases Using Various Simple Concepts

Simple modeling concepts have been used to study various potential methods of simulating belt slip and submarining. The first of these, shown in Figure 3.2, is intended to demonstrate slipping of a belt either over the thorax or lower torso pelvic region. The mass and geometry of the spherical shaped segment is intended to represent the thorax restrained by an upper torso belt, as shown. A contact surface below the sphere represents the interaction of the lower portion of the body with seat structures. The sphere can slide on, and penetrate, the surface.

To represent slipping of the belt over this surface, a second mass (10% of the first mass) was superimposed and pin-jointed at their mutual centers-of-mass. The belt was attached to the second, and smaller, mass. In order to simulate resistance to the slipping, a moment resistance is added at the pin joint as a function of the relative angle between the two masses.

This modeling concept appeared to function properly. It worked especially well when some resistance to the motion between the two segments was added. Because of the ability to control the slipping over

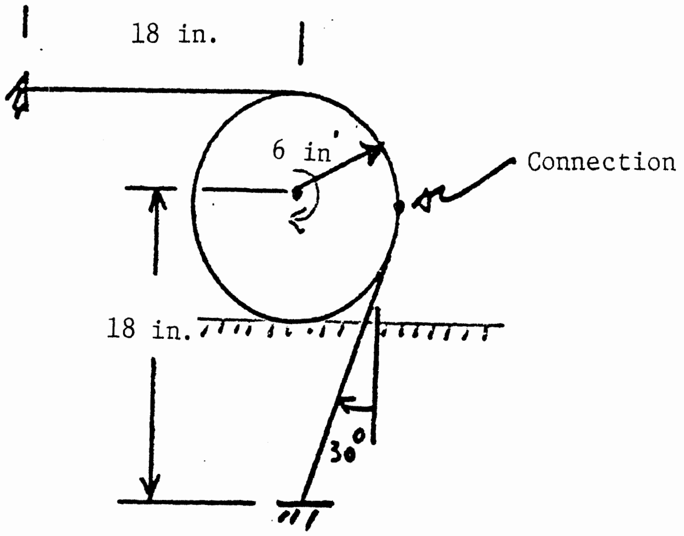


Figure 3.2. Superimposed Segments for Representation of Slip

a surface in this manner, it was felt that this model concept could be adapted to use as a submarining indicator.

A second series of simple modeling concepts was explored based on the geometry of Figure 3.3. As before, a spherical mass was used to simulate the thorax or lower torso region. The general placement of belt attachments to the vehicle was also as before. In these cases, however, a second mass (small) was located at the point where the belt would pass over the primary body segment. The belt was then attached to a separate ellipsoid on this new small mass.

Three options available with the CAL 3-D were tested as means of attachment of the small mass to the larger primary mass to simulate slippage between the two. These were the fixed distance constraint, the fixed point constraint, and the massless link. The fixed distance constraint is designed to keep a point on one mass a fixed distance from a point on the other. The fixed point constraint is intended to keep a point on one mass at the same point in inertial space as a point on the other mass. These constraints are approximate and use force to maintain the geometry rather than changing the number of degrees of freedom.

The two constraint concepts exhibited problems with instability due to a trading of energy and oscillation from one side of the belt to the other. The only way to damp this behavior was to alter the belt properties. This was considered a poor method because alterations to damp this spurious behavior would also affect the primary properties of the belt which are to stop the forward motion of the primary mass.

The massless link idea does alter the equations of motion. This concept appears to be sound based on the work of Fleck et al. (1975) and Wittenberg (1977). However, in practice, the software did not work.

A final attempt to use the geometric ideas of Figure 3.3 was to employ the traditionally jointed superimposed masses of Figure 3.2. As was the case with the original Figure 3.2 models already described, this concept worked well. It was possible to control the motion of the forward mass. These controls could be based on test data defining the direction and magnitude of forces associated with movement of the belt over the pelvis. These data could be represented as a torque-angle

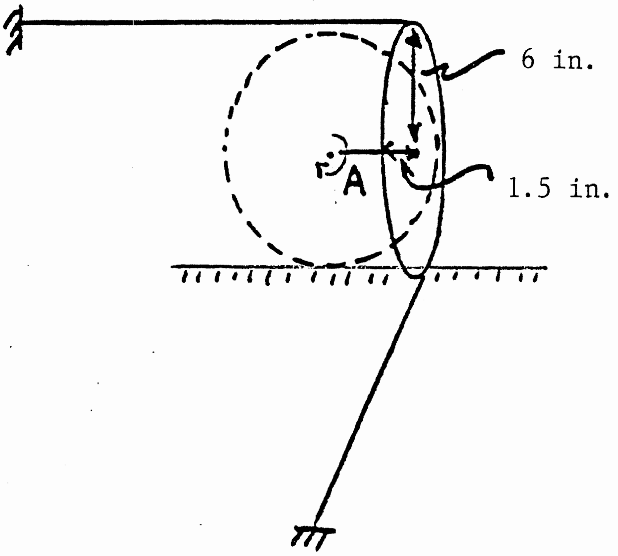


Figure 3.3. Attachment of Belt to Mass which Moves Relative to Primary Segment

curve. Because of the promise of this technique, the full-scale exercise discussed in Section 3.3 was conducted.

3.3 Full-Scale Testing of Submarining Model

A basic data set provided by General Motors was modified to include a prototype lap belt submarining indicator. Figure 3.4 is a schematic of this data set at the initial time. An overlay of the general shape of the pelvis is included to show the orientation of the lap belts. They are well positioned initially to take crash loading. The pelvic overlay is based on the positions specified in the data set of the joints connecting the pelvis to the spine and the femurs. The point labelled with an "X" on the front of the ellipsoid representing the lower torso is the attachment point for the two elements in the lower torso belt.

The data structure for the submarining indicator is based on the concept described in Section 3.3 where two masses are superimposed at a pinned joint. The small mass, representing a portion of the pelvic mass, is attached to the belt while the primary pelvic mass is attached normally to the remainder of the body linkage. The two masses are connected at a pinned joint which resists relative rotation by the mechanism of a torque-angle resistance.

Table 3.1 is the input data set for the full-scale submarining model using the UMTRI version of the CAL 3-D. Each line in the data file has a number. Those which define changes for the implementation of the submarining indicator are as follows:

Line 24.5. B2 Card. This defines the mass, inertial, and ellipsoid properties of the overlaid mass, named BPEL.

Line 56.1, 56.2. B3 Cards. The attachment of BPEL to the lower torso mass, LT, is defined.

Line 80.1. B4 Card. The torque-angle resistance of the mass to which the belt is attached (BPEL) with respect to the primary mass (LT) is defined. Values have been selected without the aid of experimental information. They are consistent with other properties of the dummy.

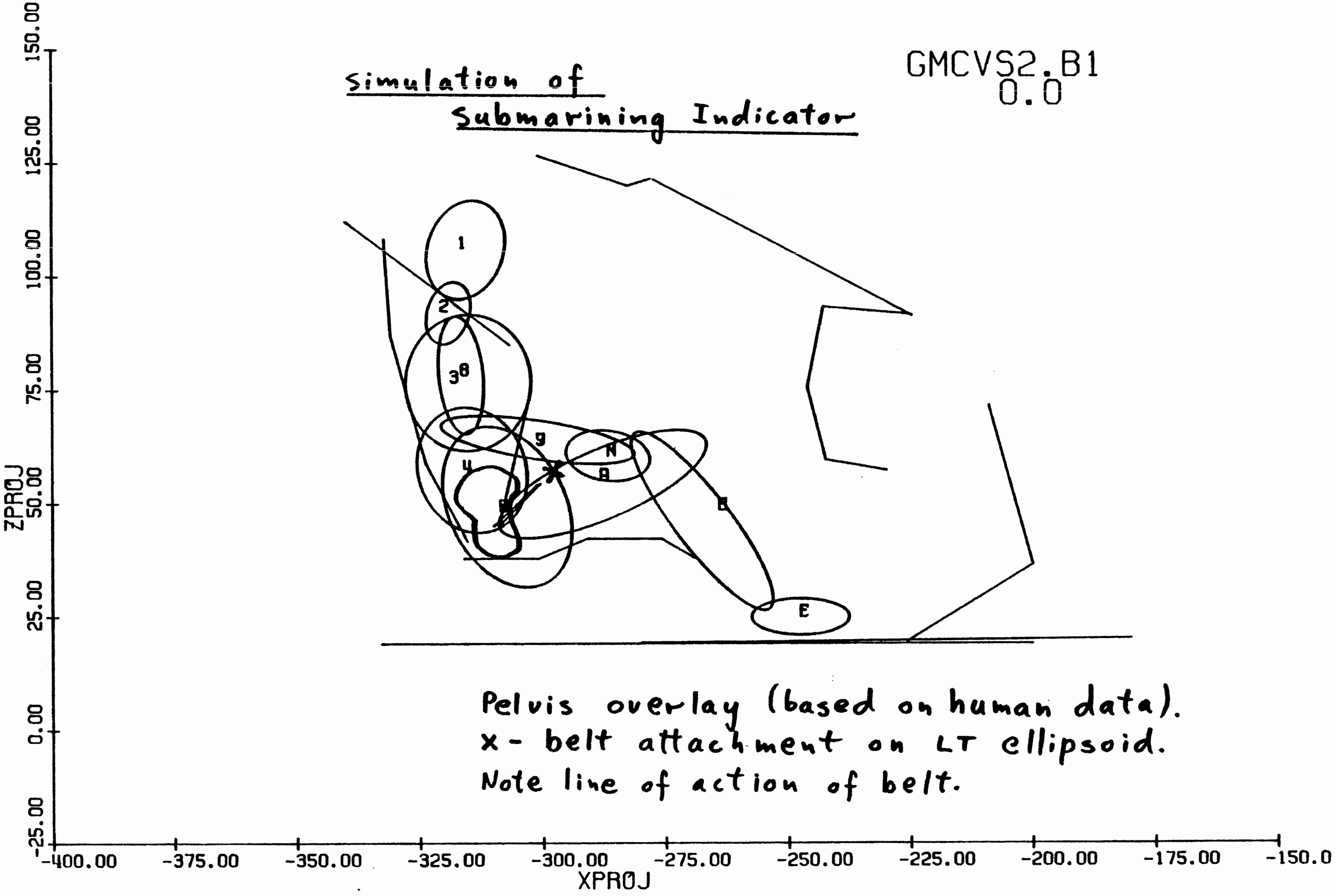


Figure 3.4. Simulation of Submarining Indicator (t = 0 ms)

```
Listing of GMCVS2.B1 at 03:24:15 on JAN 31, 1986 for CCid=SU33                                          Page    1
 1       04 NOV 1983
 2        TRIAL DATASET FOR SEAT BELT GRAPHICS ALGORITHM
 3        RESTR PASSR,FRONTAL BARRIER IMPACT, NO PITCH
 4        CM. KG.SEC.    0.0         0.0      980.66
 5         6 150 0.00100 0.00025  0.001 0.000025
 6      00 005 0 0 0 0 000 0 0 0 00000 0    1
 7          18    17        GM HYB3 50 R10.27.82
 8        LT 517.331 2.479 2.479 1.468 12.00 12.50 18.50  1.70 0.000 -.350
 9        CT 4 5.760 .3650 .3650 .2790 11.00 11.00 14.00 1.660 0.000 0.000
10        UT 317.227 2.504 2.504 2.170 12.75 12.75 15.00 3.610 0.000 0.000
11         N 2 1.545 0.060 0.060 0.006  4.30  4.30  7.00 1.670 .0000 0.000
12         H 1 4.545 .2190 .2060 .2060  7.75  7.75 11.00 1.500 .0000 0.000
13       LUL A 6.091 .6860 .6860 .1310  7.50  7.50 23.10 -0.94 .0000 .0000
14       LLL B 3.300 .7650 .7650 .0970  5.75  5.75 23.70 -4.22 .0000 .0000
15        LF C 1.227 .0550 .0550 .0150  4.00  4.00 10.00 .0000 .0000 .0000
16       RUL D 6.091 .6860 .6860 .1310  7.50  7.50 23.10 -.940 .0000 .0000
17       RLL E 3.300 .7650 .7650 .0970  5.75  5.75 23.70 -4.22 .0000 .0000
18        RF F 1.227 .0550 .0550 .0150  4.00  4.00 10.00 .0000 .0000 .0000
19       LUA 6 2.000 .1160 .1160 .0250  4.50  4.50 13.00  0000  .000 .0000
20       LLA 7 2.273 .3600 .3600 .0320  4.25  4.25 20.80  1.15 .0000 .0000
21       RUA 8 2.000 .1160 .1160 .0250  4.50  4.50 13.00 .0000  0.00 .0000
22       RLA 9 2.273 .3600 .3600 .0320  4.25  4.25 20.80  1.15 .0000 .0000
23        LH N 0.800 .0115 .0115 .0012  5.50  3.05  9.00  .000 .000  .0000
24        RH M 0.80  .0115 .0115 .0012  5.50  3.05  9.00  .000 .0000 .0000
24.5    BPEL P4.    2.479 2.479 1.468 12.   12.5  18.5  1.7   0.    -.35
25         P O   1  -2 -4.25       -4.32       .0000  6.53
26                         0.0   0.0   0.0   0.0   0.0   0.0
27         W P   2  -2 -1.60       -6.48       .0000 13.93
28                         0.0   0.0   0.0   0.0   0.0   0.0
29        NP Q   3  -2 -2.54      -15.59       .0000 .0000
30
31        HP R   4  -2 0.000       -9.59       .0000 4.780
32
33        LH W   1  -4 -2.45 -8.85  2.57   0.0 .0000-23.99
34                     -90.  90.                4.76     27.0  -70.0
35        LK X   6   1  0.00 .0000 16.18   0.0 .0000-16.36
36                                               43.0
37        LA G   7  -4-0.250 .0000 22.04  5.36 .0000 -4.09
38                     90.                90.                    79.0
39        RH Y   1  -4 -2.45  8.85  2.57   0.00.0000-23.99
40                     90.    90.               4.76     -27.   70.
41        RK Z   9   1   0.0 .0000 16.18   0.0 .0000-16.36
42                                               43.0
43        RA H  10  -4-0.250 .0000 22.04  5.36 .0000 -4.09
44                     90.                90.                    79.0
45        LS S   3  -4  0.45-18.00-14.59   0.0 .0000-13.06
46                                 90.                    -55.   -65.
47        LE T  12  -4   0.0 .0000 13.07   0.0 .0000-15.11
48                       90.0             90.          8.0   70.
49        RS U   3  -4  0.45 18.00-14.59   0.0 .0000-13.06
50                               -90.                      55.0  65.
51        RE V  14  -4   0.0 .0000 13.07   0.0 .0000-15.11
52                      90.                90.         -8.   70.
53        LW J  13   1   0.0 .0000 10.20   0.0 .0000 -5.11
54                       0.0 .0000              -10.  00.0
55        RW K  15   1   0.0 .0000 10.20   0.0 .0000 -5.11
56                       0.0 .0000              -10.  00.0
56.1    PELJ P   1   10.    0.    0.    0.    0.    0.
```

Table 3.1. Input Data Set for Full-scale Submarining Model (Page 1 of 10)

```
 56.2
 57      115.8 0.000 .2000 0.500 90.000      115.8 .0000 .2000 0.500 70.000
 58      115.8 0.000 0.200 0.500 90.000      115.8  .000 0.200 0.5   70.0
 59      30.0  .108  .0000 0.500 60.000      150.  .108  .0000 0.500 60.000
 60      30.0  .108  .0000 0.500 60.000      150.  .108  .0000 0.500 60.000
 61      6.000 .2500 .0000 0.500 10.000      2.972 .1088 0.000 0.500 100.00
 62      16.0  160.   160. 0.5   60.0         0.     -90.0  0.0
 63       0.00 10.00  10.0 0.950 45.000      0.000  .000 0.000 0.950
 64      1.00  10.00 10.00 0.950 50.000      1.0    10.0  10.  0.950  42.
 65      0.     0.
 66      6.000 .2500 .0000 0.500 10.000      2.972 .1088 0.000 .500   100.
 67      16.0   160.  160.0 .5   60.0         0.0   0.    0.
 68            10.00 10.00 0.950 45.000                        0.950  0.000
 69      1.00  10.00 10.00 0.950 50.000      1.0    10.0  10.  0.950  42.
 70      0.     0.
 71      14.    00.0  00.0 0.950 125.00                        0.950  60.0
 72      0.     0.
 73             20.0  20.0 0.950 45.000                        0.950 70.000
 74      0.     0.
 75      14.    00.0  00.0 0.950 125.00                        0.950 60.0
 76      0.     0.    00.0
 77             20.0  20.0 0.950 45.000                        0.950 70.00
 78      0.     0.
 79      14.  20.   0.    .5    60.         14.   20.   0.     .5    60.
 80      14.  20.   0.    .5    60.         14.   20.   0.     .5    60.
 80.1   50.  10.   10.   .95   30.
 81        0.0 1000. 200.0 174.5    .0                     0.0   0.0
 82        0.0 1000. 200.0 105.0    .0                     0.0   0.0
 83        0.0  20.0 200.0 174.5    .0                     0.0   0.0
 84        0.0  20.0 200.0 180.9    .0                     0.0   0.0
 85        0.0 500.0 200.0 180.9    .0                     0.0   0.0
 86        0.0 500.0 200.0 180.9    .0                     0.0   0.0
 87        0.0 500.0 200.0 180.9    .0
 88        0.0 125.0  10.0  15.4    .0                     0.0   0.0
 89        0.0  20.0  10.0   30.    .0                     0.0   0.0
 90        0.0  20.0  10.0   30.    .0
 91        0.0   0.0  10.0
 92        0.0 500.0 200.0 180.9   0.0                     0.0   0.0
 93        0.0 500.0 200.0 180.9   0.0                     0.0   0.0
 94        0.0 500.0 200.0 180.9   0.0
 95        0.0 125.0  10.0  15.4    .0                     0.0   0.0
 96        0.0  20.0  10.0   30.    .0
 97        0.0  20.0  10.0   30.    .0
 98        0.0   0.0  10.0  0.0
 99        0.0 125.0 400.0  65.0   0.0                     0.0   0.0
100        0.0  50.0 400.0  65.0
101        0.0  0.0  400.0   0.0
102        0.0 175.0 100.0  16.3   0.0                     0.0   0.0
103        0.0 175.0 100.0  16.3
104        0.0   0.0 100.0
105        0.0 120.0 400.0  65.0   0.0                     0.0   0.0
106        0.0  50.0 400.0  65.0
107        0.0  0.0  400.0   0.0
108        0.0 175.0 100.0  16.3   0.0                     0.0   0.0
109        0.0 175.0 100.0  16.3
110        0.0   0.0 100.0
111     .1    50.   100.  25.   0.
112     .1    50.   100.  25.   0.
```

Table 3.1. Input Data Set for Full-scale Submarining Model (Page 2 of 10)

```
112.1   0.     50.    50.    50.    0.
113                                                      00.1 00.10 00.10 00.01 00.01 00.01
114                                                      00.1 00.10 00.1
115                                                      00.1 00.10 00.10
116                                                      00.1 00.10 00.10
117                                                      00.1 00.10 00.10
118                                                      00.1 00.10 00.10
119                                                      00.1 00.10 00.10
120                                                      00.1 00.10 00.10
121                                                      00.1 00.10 00.10
122                                                      00.1 00.10 00.10
123                                                      00.1 00.10 00.10
124                                                      00.1 00.10 00.10
125                                                      00.1 00.1  00.1
126                                                      00.1 00.1  00.1
127                                                      00.1 00.1  00.1
128                                                      00.1 00.1  00.1
129                                                      00.1 00.1  00.1
129.1                                                   .1    .1    .1
130       FRONT BARRIER PULSE,  PRIMARY VEHICLE MOTION
131        0.0    0.0    0.0   1369. 0.000  0.0    0.0    0.0      59   0.0 0.002      00
132       0.0    0.0    1.0  7.0    17.0   19.0   18.0   12.5   8.5    5.0    2.0    22.5
133       27.5   15.0   18.0 20.5   37.0   33.0   22.5   14.5   15.5   20.0   22.0   25.0
134       23.0   25.5   26.0 25.5   22.5   21.5   21.0   25.0   24.0   23.0   22.0   20.0
135       17.0   15.0   12.5 10.5   10.0   10.0    7.5    5.0    1.0    0.0   -3.0   -3.0
136       -2.5   -2.0   -2.0 -1.5   -1.0   -0.5    0.0    0.0    0.0    0.0    0.0    0.0
137          23      2      0      0
138         1           WINDSHIELD
139        -224.8          50.0       -91.2
140        -277.5          50.0       -121.2
141        -224.8         -50.0       -91.2
142         2           DASHBOARD 1
143        -200.1          50.0       -36.4
144        -209.2         50.0        -71.5
145        -200.1         -50.0       -36.4
146         3         BOTTOM I.P.
147        -230.0           50.       -57.2
148        -242.4          50.        -59.5
149        -230.0         -50.        -57.2
150         4        LOWER I.P.
151        -242.4           50.       -59.5
152        -246.2          50.        -75.5
153        -242.4         -50.        -59.5
154         5        MID I.P.
155        -246.2           50.       -75.5
156        -242.9          50.        -93.1
157        -246.2         -50.        -75.5
158         6        FR SEAT BACK1
159        -323.8          20.0      -58.82
160        -315.3          20.0      -41.42
161        -323.8         -20.0      -58.82
162         7        FR SEAT BACK2
163        -330.9          20.0      -86.72
164        -323.8          20.0      -58.82
165        -330.9         -20.0      -86.72
166         8       FR SEAT CUSHION1
167        -316.0          25.0       -37.72
168        -301.0          25.0       -37.72
```

Table 3.1. Input Data Set for Full-Scale Model (Page 3 of 10)

```
169      -316.0      -25.0      -37.72
170       9       FLOORBOARD
171      -280.        50.0      -19.2
172      -180.0       50.0      -20.1
173      -280.       -50.0      -19.2
174       10          TOEBOARD
175      -225.8       50.0     -19.5
176      -200.1       50.0     -36.4
177      -225.8      -50.0     -19.50
178       11      UPPER I.P.
179        -242.9    50.0      -93.1
180        -224.8    50.0      -91.4
181        -242.9   -50.       -93.1
182       12      FRONT SEAT BACK3
183      -332.2      20.0      -108.12
184      -330.9     20.0       -86.72
185      -332.2     -20.0      -108.12
186       13     LOWER I.P.
187      -242.4        50.      -59.5
188      -246.2       50.       -75.5
189      -242.4      -50.       -59.5
190       14     MID I.P.
191      -246.2        50.      -75.5
192      -242.9       50.       -93.1
193      -246.2      -50.       -75.5
194       15    FR SEAT CUSHION2
195      -301.0       25.0      -37.72
196      -291.0       25.0      -42.12
197      -301.0      -25.0      -37.72
198       16     FR SEAT CUSHION3
199      -291.0       25.0      -42.12
200      -276.0       25.0      -42.12
201      -291.0      -25.0      -42.12
202       17    FR SEAT CUSHION4
203      -276.0       25.0      -42.12
204      -268.9       25.0      -37.72
205      -276.0      -25.0      -42.12
206       18          SILL
207      -332.8       50.0     -19.0
208      -200.1       50.0     -19.0
209      -332.8      -50.0     -19.00
210       19    ROOF PANEL 1
211      -282.6       50.0      -119.8
212      -277.9       50.0      -121.3
213      -282.6      -50.0      -119.8
214       20    ROOF PANEL 2
215      -282.6        50.      -119.8
216      -301.0       50.       -126.5
217      -282.6      -50.       -119.8
218       21      BOTTOM I.P.
219      -230.0        50.      -57.2
220      -242.4       50.       -59.5
221      -230.0      -50.       -57.2
222       22     LOWER I.P.
223      -242.4        50.      -59.5
224      -246.2       50.       -75.5
225      -242.4      -50.       -59.5
226       23     MID I.P.
```

Table 3.1. Input Data Set for Full-scale Submarining Model (Page 4 of 10)

```
-246.2         50.      -75.5
-242.9        50.       -93.1
-246.2        -50.      -75.5
   LAP BELT
-310.         -20.00     -45.00       -309.          21.00       -45.00
 12.0                    -4.000                      0.010
  SHOULDER BELT
-340.0        -24.      -112.         -309.         21.00       -45.
  11.9          0.000     1.0000                    1.000

  1    HEADREST FORCE(KG)
   0.0         -15.         0.0                     0.0
   4
   0.0       0.0          5.0         100.          10.0        200.
   15.0      300.
  2    SEAT CUSHION FORCE(KG)
   0.0         -14.         0.0         1.0
   8
   0.0      0.0           2.0         20.          4.0          45.
   6.0       80.          8.0          130.         10.0         200.
   12.0      300.         14.0         450.
  3       SEATBACK FORCE(KG)
   0.0         -16.         0.0         1.0
   8
   0.0      0.0           1.0         7.5          2.            20.
   3.        50.          4.          90.          6.            200.
  10.        500.         16.         1000.
  4       TOEBRD FORCE(KG)
   0.0         -10.                     1.0
  18
   0.0       0.0          0.1         14.0         0.2          22.
   0.3       40.          0.4          56.         0.5          74.
   0.6       95.          0.7          115.        0.8          145.
   1.0        150.        2.0          240.        3.0          300.
   4.0       350.         5.0         500.         6.0          900.
   7.0        1300.       9.0         1650.       10.0          1800.
  5       WNDSHLD R FUNCTION
   0.0       -20.           0.0
   4
   0.0         1.0          2.0          0.5          9.0          0.25
  20.        0.0
  6       WNDSHLD G FUNCTION
   0.0       -20.           0.0
   4
   0.0         0.0          2.0          .75          9.0          0.95
  20.        0.95
  8    MID I.P. LOADING(KG)
   0.0         -20.         0.                      0.0
   9
   0.0         0.0          1.0         1200.        1.5          1620.
   2.        1800.          2.5         1860.        3.0          1910.
   4.        2010.          6.0         2055.        20.          2160.
  9       CONSTANT F=0.40
   0.0         0.0          0.40
 10       CONSTANT F=0.1
   0.0         0.0          0.1
 11    UPP I.P. LOAD(KG)
   0.0         -24.         0.0                     1.0
```

Table 3.1. Input Data Set for Full-scale Submarining Model (Page 5 of 10)

```
       7
       0.0        0.0         3.0        80.           6.        120.
       9.       240.          12.       560.          18.       1150.
      24.       1800.
    12        CONSTANT F=0.9
       0.0        0.0         0.9
    13      TOEBRD F
       0.0        0.          0.95
    14      CONSTANT  F=0.67
       0.0        0.0         0.67
    15    UNLOADING G FUNCTION
       0.0     -1000.         0.0
       4
       0.0        0.0         0.5         0.0         `0.6          0.95
    1000.         0.95
    16    UNLOADING R FUNCTION
       0.0     -1000.         0.0
       4
       0.0        1.0         0.5         1.0          0.6          0.05
    1000.         0.05
    17        WNDSHLD FORCE(KG)
       0.0       -20.         0.0                     1.0
      16
       0.0        0.0         1.         70.           2.        110.
       3.       145.          4.        170.           5.        182.
       6.       188.          7.        190.           8.        189.
       9.       176.         10.        166.          11.        153.
      12.       130.         13.         90.          15.          0.
      20.       0.
    18      WSHLD  SPIKE(KG)
       0.0       -2.5        0.0         1.5          1.0
      13
       0.0        0.0        0.4         60.          0.5        120.
       0.7      160.         0.9        190.          1.0        200.
       1.2      190.         1.4        160.          1.5        130.
       1.7       80.         2.0         40.          2.2         20.
       2.5       0.
    19        LWR IP LOAD(KG)
       0.0       -40.         0.0                     0.0
      17
       0.0        0.0         1.         20.           2.         50.
       3.       100.          4.        200.           5.        300.
       6.       400.          7.        485.           8.        495.
      11.       525.         13.        550.          15.        600.
      17.       700.         19.        900.          20.       1070.
      21.      1300.         40.5     160000.
    20        HEADER LOAD(KG)
       0.0       -40.         0.0                     0.0
       6
       0.0        0.0         5.08       445.         7.62       980.
      12.7      980.         20.3       2225.         40.5      16000.
    21        TOEBRD G
       0.0       -10.         0.0                     0.0
       8
       0.0        0.0        0.2         0.05         0.4         0.13
       0.65       0.22       1.50        0.55         2.20        0.80
       2.8        0.92      10.0         0.99
    22        TOEBRD R
```

Table 3.1. Input Data Set for Full-scale Submarining Model (Page 6 of 10)

```
 343        0.0        -10.        0.0                  0.0
 344        8
 345        0.0         0.0        0.2        0.47      0.4        0.37
 346       0.65      0.31          1.50       0.26      2.20       0.24
 347       2.8       0.22          10.0       0.01
 348      23     WNDSHLD F
 349        0.0        10.0
 350        0.76       0.156         0.0
 351      24     ROOFPAN F
 352        0.0         0.0        0.80
 353      25     SEAT F
 354        0.0        10.0
 355        0.30       0.1           0.0
 356      26     BODY FORCES(KG)
 357        0.0        -4.5       0.0                   1.0
 358        6
 359       '0.0         0.0       0.4         1.         1.0        2.
 360        1.5         3.        2.0         4.         4.5        6.
 361      27     LAP BELT FDF
 362         0.0        -1.50
 363        5
 364       0.00        0.0         .125      70.0        .254      400.
 365       1.25      800.0         1.50     800.0
 366      28    SHOULDER BELT FDF
 367         0.0        -1.25
 368        5
 369       0.00        0.0         .0254     44.0        0.25     800.0
 370       0.950     900.0         1.25     900.0
 371      29    BELT R
 372         0.0         0.0        0.750
 373      30    BELT G
 374         0.0         0.0        0.150
 375      31    BELT CF
 376                                1.0
 377      32    BELT SEGMENT CF
 378                                1.0        0.0        0.5
 379      33    TIE POINT CF
 380                                0.5        0.         0.5
 381     999
 382      1   1   2   5   5   3   3   1   2   2   2   4   4   2   3   3   1
 383      1   1   2   2   2
 384      1  19   5   5  17  18   5   6  23
 385      2  19   5   5  11      16  15  24
 386      3  19   7   7  19   0  16  15  12
 387      3  19  10  10  19   0  16  15  12
 388      4  19   6   6  19   0  16  15  12
 389      4  19   7   7  19   0  16  15   9
 390      4  19   9   9  19   0  16  15   9
 391      4  19  10  10  19      16  15   9
 392      4  19   1   1   8      16  15  12
 393      5  19   1   1   8   0  16  15  12
 394      5  19   2   2   8   0  16  15  12
 395      5  19   3   3   8   0  16  15  12
 396      5  19  13  13  19      16  15  12
 397      5  19  15  15  19      16  15  12
 398      6  19   1   1   3   0  16  15  10
 399      6  19   2   2   3   0  16  15  10
 400      6  19   3   3   3      16  15  10
```

Table 3.1. Input Data Set for Full-scale Submarining Model (Page 7 of 10)

```
  401         7   19    1    1    3        16   15   10
  402         7   19    2    2    3    0   16   15   10
  403         7   19    3    3    3    0   16   15   14
  404         8   19    1    1    2        16   15   25
  405         9   19    8    8    4        22   21   13
  406         9   19   11   11    4    0   22   21   13
  407        10   19    8    8    4        22   21   13
  408        10   19   11   11    4        22   21   13
  409        11   19    5    5   11        16   15   10
  410        11   19    3    3   11        16   15   10
  411        12   19    5    5    1        16   15   10
  412        12   19    1    1    2        16   15   25
  413        12   19    2    2    2        16   15   25
  414        12   19    3    3    2        16   15   25
  415        13   19   12   12   19        16   15   12
  416        13   19   14   14   19        16   15   12
  417        13   19   13   13   19        16   15   12
  418        13   19   15   15   19        16   15   12
  419        14   19   12   12   19        16   15   12
  420        14   19   14   14   19        16   15   12
  421        15   19    1    1    2        16   15   25
  422        15   19    6    6    2        16   15   25
  423        15   19    9    9    2        16   15   25
  424        16   19    1    1    2        16   15   25
  425        16   19    6    6    2        16   15   10
  426        16   19    9    9    2        16   15   10
  427        17   19    1    1    2        16   15   10
  428        19   19    5    5   10        16   15   24
  429        20   19    5    5   20        16   15   24
  430        21   19    6    6   19        16   15   24
  431        21   19    9    9   19        16   15   24
  432        22   19   17   17   19        16   15   10
  433        22   19   16   16   19        16   15   10
  434        23   19   16   16   19        16   15   10
  435        23   19   17   17   19        16   15   10
  436         1    1
  437         1   19   18   18   27        29   30   31
  438         2   19    3    3   28        29   30   31
  439         2    2    4    4    0    1    0    0    1    0    0    5    3    3    1    0    0    0
  440         1    1   13   13    4        16   15   10
  441         1    1   15   15    4        16   15   10
  442         2    2   16   16   26        16   15   10
  443         2    2   17   17   26        16   15   10
  444         3    3   13   13    4        16   15   10
  445         3    3   15   15    4        16   15   10
  446         3    3   16   16   26        16   15   10
  447         3    3   17   17   26        16   15   10
  448         4    4   12   12   26        16   15   12
  449         4    4   13   13   26        16   15   12
  450         4    4   14   14   26        16   15   12
  451         4    4   15   15   26        16   15   12
  452         6    6   16   16   26        16   15   12
  453         9    9   17   17   26        16   15   12
  454        12   12   14   14   26        16   15   12
  455        12   12   15   15   26        16   15   12
  456        12   12    1    1   26        16   15   12
  457        12   12    2    2   26        16   15   12
  458        12   12    3    3   26        16   15   12
```

Table 3.1. Input Data Set for Full-scale Submarining Model (Page 8 of 10)

```
  459        13   13   14   14   26         16   15   12
  460        13   13   15   15   26         16   15   12
  461        13   13    2    2   26         16   15   12
  462        14   14    1    1   26         16   15   12
  463        14   14    2    2   26         16   15   12
  464        14   14    3    3   26         16   15   12
  465        15   15    2    2   26         16   15   12
  466
  467         10.0       30.0       1.0       18
  468        -308.71     0.0      -48.19
  469          0.0      22.50       0.0
  470          0.0      15.00       0.0
  471          0.0      -2.00       0.0
  472          0.0     -15.00       0.0
  473          0.0     -15.0        0.0
  474          0.0     115.00       0.0
  475          0.0     35.000       0.0
  476          0.0      90.0        0.0
  477          0.0     115.00       0.0
  478          0.0     35.000       0.0
  479          0.0      90.0        0.0
  480          0.0       5.0        0.
  481         17.5      81.0        0.        0.0      0.0          0.0        3   2   1
  482          0.0       5.0       00.
  483        -17.5      81.0        0.                                         3   2   1
  484         17.5      81.0        0.                                         3   2   1
  485        -17.5      81.0        0.                                         3   2   1
  486      0.         22.5        0.
 1000
 1001      0.       150.    1.         0
 1002         1    2    3    4    5    6    7    8    9   10
 1003
 1004         18   0   10.           0.           0.
 1005              0   20.           0.           0.
 1006              0   30.           0.           0.
 1007              0   40.           0.           0.
 1008              0   50.           0.           0.
 1009              0   60.           0.           0.
 1010              0   70.           0.           0.
 1011              0   80.           0.           0.
 1012              0   90.           0.           0.
 1013              0  100.           0.           0.
 1014              0  110.           0.           0.
 1015              0  120.           0.           0.
 1016              0  130.           0.           0.
 1017              0  140.           0.           0.
 1018              0  150.           0.           0.
 1019              0  160.           0.           0.
 1020              0  170.           0.           0.
 1020.5            0  180.           0.           0.
 1021         18   0   10.           0.           0.
 1022              0   20.           0.           0.
 1023              0   30.           0.           0.
 1024              0   40.           0.           0.
 1025              0   50.           0.           0.
 1026              0   60.           0.           0.
 1027              0   70.           0.           0.
 1028              0   80.           0.           0.
```

Table 3.1. Input Data Set for Full-scale Submarining Model (Page 9 of 10)

```
 1029              0  90.        0.          0.
 1030              0 100.        0.          0.
 1031              0 110.        0.          0.
 1032              0 120.        0.          0.
 1033              0 130.        0.          0.
 1034              0 140.        0.          0.
 1035              0 150.        0.          0.
 1036              0 160.        0.          0.
 1037              0 170.        0.          0.
 1037.5            0 180.        0.          0.
 1038          18  0  10.        0.          0.
 1039              0  20.        0.          0.
 1040              0  30.        0.          0.
 1041              0  40.        0.          0.
 1042              0  50.        0.          0.
 1043              0  60.        0.          0.
 1044              0  70.        0.          0.
 1045              0  80.        0.          0.
 1046              0  90.        0.          0.
 1047              0 100.        0.          0.
 1048              0 110.        0.          0.
 1049              0 120.        0.          0.
 1050              0 130.        0.          0.
 1051              0 140.        0.          0.
 1052              0 150.        0.          0.
 1053              0 160.        0.          0.
 1054              0 170.        0.          0.
 1054.5            0 180.        0.          0.
 1055          18       1     2     3     4     5     6     7     8     9    10    11
 1056                  12    13    14    15    16    17    18
 1057          18       1     2     3     4     5     6     7     8     9    10    11
 1058                  12    13    14    15    16    17    18
 1059          18       1     2     3     4     5     6     7     8     9    10    11
 1060                  12    13    14    15    16    17    18
 1061          17       1     2     3     4     5     6     7     8     9    10    11
 1062                  12    13    14    15    16    17
 1063        5   3
 2001      0 1 1 1 0 1 1 0
 2002     0.       .15      .01
 2003     0        0.       0.
 2004     1.       0.       0.
 2005     0.       1.       0.
 2006     0.       0.       1.
 2007     1.       0.       0.
 2008     0.       1.       0.
 2009     0.       0.       1.
 2010     .5       .5       11.      8.5
 2011     .5       .5       10.      7.
 2012     -400.    -150.    250.     175.     .3
 2013     GMCVS2.D  XZ


$EMP GUC7
File "GUC7" has been emptied.
```

Table 3.1. Input Data Set for Full-scale Submarining Model (Page 10 of 10)

However, when specific data are obtained which can directly relate pelvic orientation to belt placement and submarining, they should be used.

Line 112.1. B5 Card. The second card describing the torque-angle data.

Line 129.1. B6 Card. The three quantities included are magnitude, absolute error, and relative error tests for angular acceleration of BPEL. The values chosen are the same as for the other variables.

Line 437. F2 Card. This is a variation of the card used in the original data set. The belt attachment for the occupant is shifted from the lower torso (segment 1) to the overlay mass (segment 18).

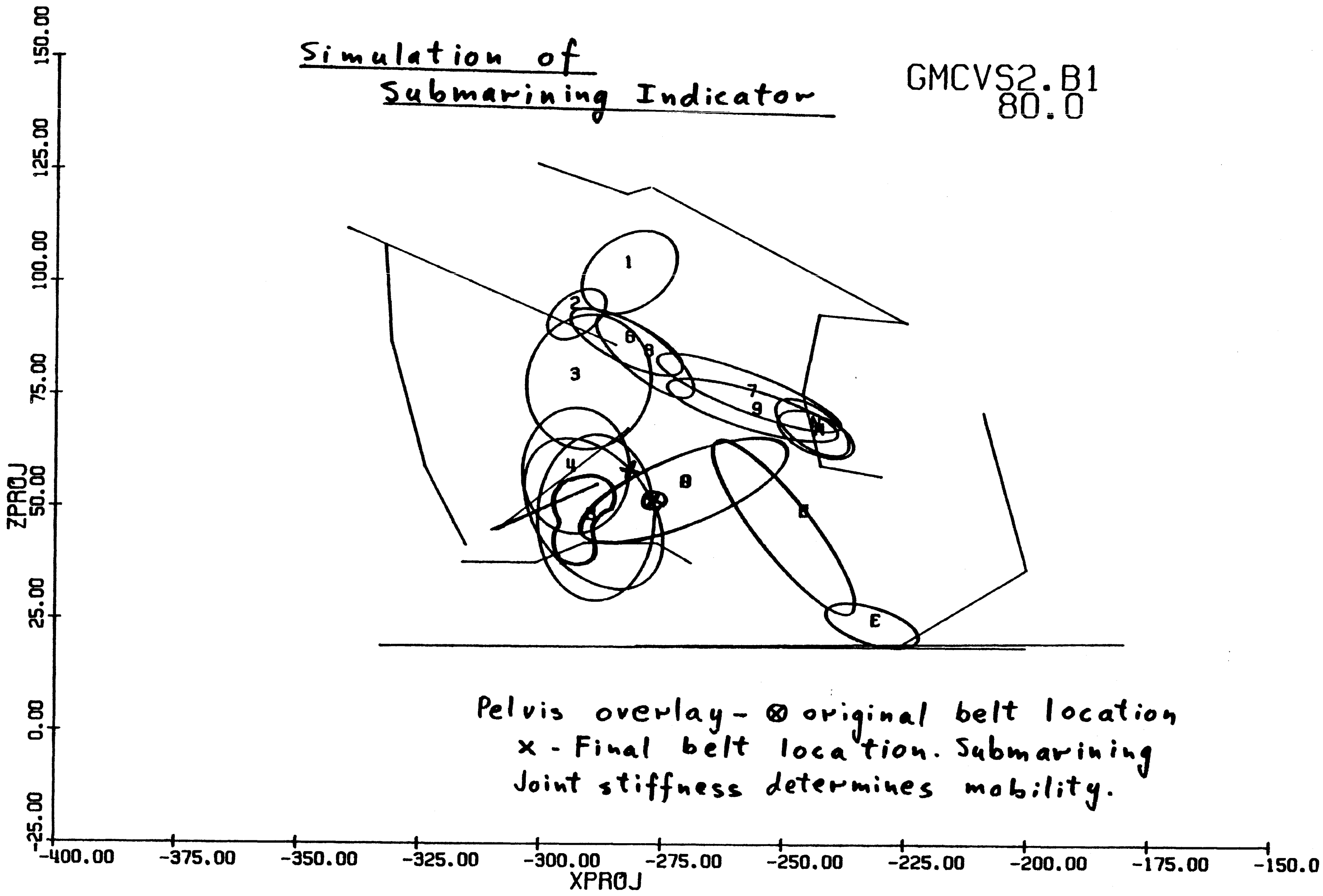
NOTE: Many lines in the data set use the number of segments included in the model. The segment number representing the vehicle has been changed to 19 with BPEL, the new mass in the occupant linkage, now being assigned to segment 18.

Figure 3.5 is a schematic showing the occupant at 80 milliseconds into simulation. It should be noted that the overlay mass has rotated backward toward the seat back. The original position of the belt attachment to the lower torso is indicated by an "X" with a circle around it. This point has migrated upward, in the direction of submarining, to the point indicated with an "X." Some penetration into the original lower torso mass has also occurred.

A summary of the general results from this simulation are as follows:

-- Velocity: 1369 cm/sec
-- Deceleration: Peak of 37 g and duration of 90 ms
-- Peak Head Deceleration: 61 g's at 74 ms
-- Peak Thorax Deceleration: 65 g's at 71 ms
-- Peak Pelvis Deceleration: 64 g's at 73 ms
-- Upper Torso Belt: 487 kg at 84 ms
-- Lower Torso Belt: 471 kg at 84 ms
-- Lap Belt A: 809 kg at 75 ms
-- Lap Belt B: 808 kg at 78 ms

This simulation has demonstrated the capabilities of the existing BELT subprogram to model submarining without changes to the code. In



Simulation of Submarining Indicator

GMCVS2.B1
80.0

Pelvis overlay – ⊗ original belt location
x - Final belt location. Submarining
Joint stiffness determines mobility.

Figure 3.5. Simulation of Submarining Indicator (t = 80 ms)

order to use it in realistic design studies, impact studies should be conducted to refine the following input quantities:

- -- Location of the joint on the overlay mass.
- -- Properties of the torque-angle resistance.
- -- Shape and location of the overlap ellipsoid.

This will require new kinds of tests and analyses of results which graphically document the position of the belt on the pelvic region as a function of the pelvic structure of the test dummy.

4.0 ANALYSIS OF BELT MATERIAL TRANSFER ACROSS SURFACES

4.1 The Belt Slip Model Developed by British Leyland

As was mentioned in Section 3.1, each belt has a plane called the belt plane which is defined by the positions of two belt anchor points and the fixed point in inertial space. The intersection of the belt plane and the belt contact ellipsoid is the belt ellipse within the belt plane. This situation is pictured in Figure 4.1. From each slip point the belt proceeds as a straight line in the belt plane until it intersects the belt ellipse at a tangent point and then conforms to the belt ellipse to the fixed point. In the case that the belt ellipse is oriented so that there exists line-of-sight between one or both anchors and the fixed point, the belt is assumed to proceed along the line-of-sight and of course no tangent point exists on that side of the fixed point. Figure 4.2 illustrates several such cases. In the original Calspan report, this figure illustrated the algorithm for choosing the belt path from the two tangent lines to an ellipse through any point outside of the ellipse. In the cases where no actual tangent point exists, the term "tangent point" will be defined to mean "fixed point."

A point at which anchors from two belts coincide lies in a plane defined by the common anchor and the near tangent point or fixed point from each of the two belts. We will call this plane the anchor slip plane. In general the anchor slip plane will not coincide with the belt planes of either belt and may or may not contain the fixed points of the two belts. This definition will be used in cases where webbing material passes through a ring. However, in all cases the points at which the belts proceeding from the common anchor first intersect the two belt ellipses will be within the belt slip plane by definition. If the belt is imagined to fasten to the anchor point, then the force vector representing the belt tension's action upon the belt anchor will lie along the belt from the anchor to the nearer tangent point with a magnitude equal to the belt tension. The resultant of both belts upon the common anchor will be the vector sum of the two belt vectors. The

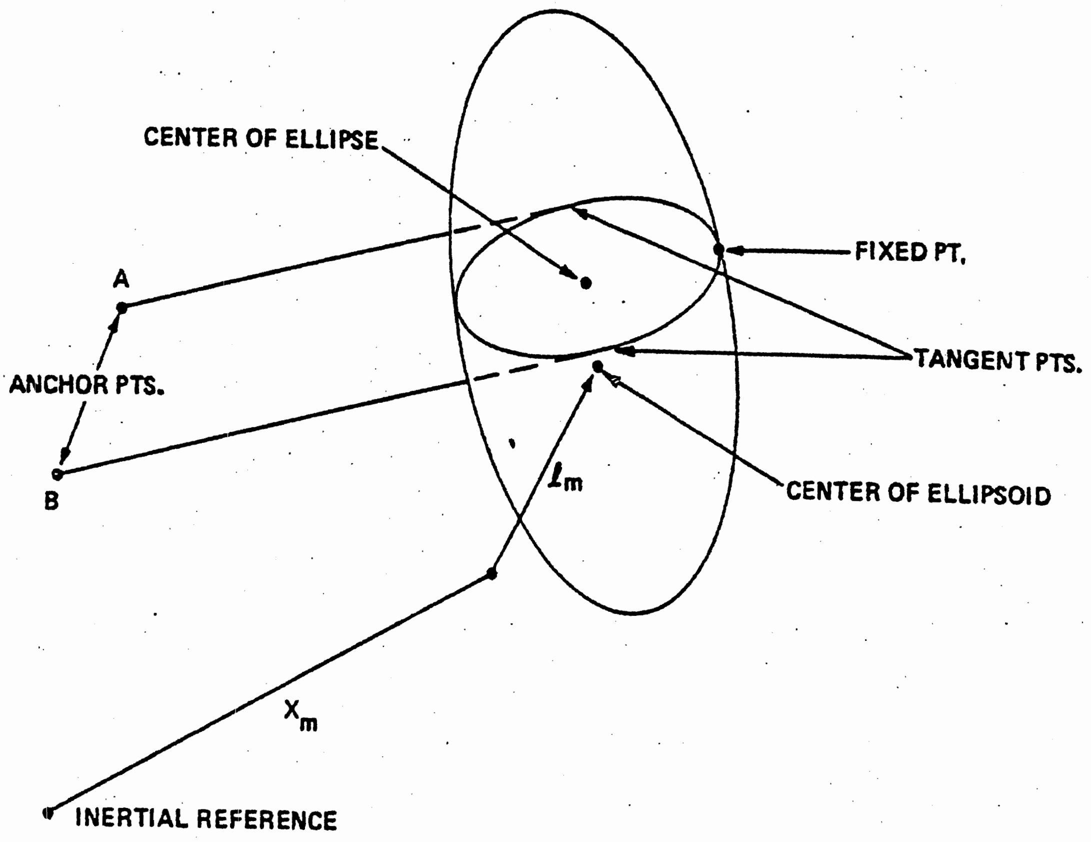
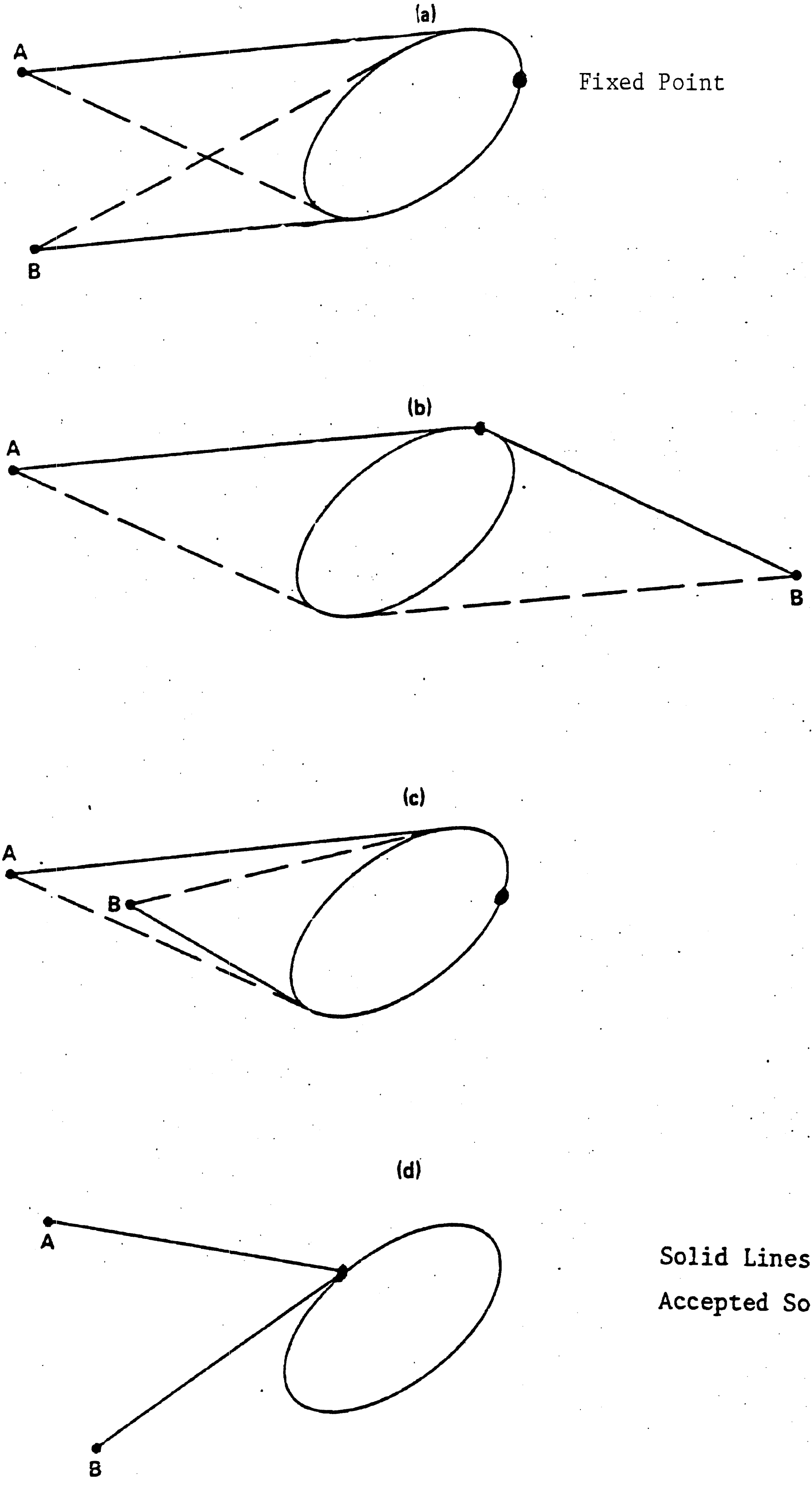


Figure 4.1. The Belt Plane in Inertial Space.



Fixed Point

Solid Lines Indicate Accepted Solutions

Figure 4.2. Belt Layout Examples.

resultant belt force is broken into "normal" and "tangential" components. The "normal" direction is defined by the unit vector at the common anchor point for which the scalar products with the unit vectors along the two belt force vectors at the same point are equal. The "tangential" direction is defined as an unit vector at the common anchor which is perpendicular. There are two unit vectors which satisfy both the "normal" and "tangential" definitions. In each case the unit vector is chosen for which the component of the resultant force is positive. Figure 4.3 shows the various vectors in the belt slip plane within the belt segment on one side labeled "A" and on the other side labeled "B."

The component of the resultant force in the tangential direction is called the "slip force." The slip force is assumed to be opposed by a "slip friction force" and a "threshold force." The slip friction force is modelled as kinetic friction while the larger effect of static friction is modelled as a threshold force. If slip force exceeds the sum of slip friction force and threshold force then the unstrained belt lengths are modified until slip force is reduced below this sum but not below slip friction force. The satisfaction of this criterion will be referred to as reaching "equilibrium." Note that equilibrium is used relative to the iteration which takes place at each point in time in order to calculate what would physically be seen at that time. Equilibrium is not used in the usual physical sense.

4.2 UMTRI Extensions to the British Leyland Slip Algorithm

Friction-opposed slipping was desired at both fixed points on segments and at vehicle anchors for as many belt segments as were considered to be defined by a single piece of webbing. This was done in order to consider a complete three-point belt system which consists of two lap and shoulder sections with the potential for slipping across the lap, through the ring connecting the lap and shoulder sections, and across the chest. The first extension required was to find corresponding definitions for normal direction, tangential direction, and slip force for the case of the fixed point on a belt ellipse. It is clear that the direction of belt tension at the fixed point itself is

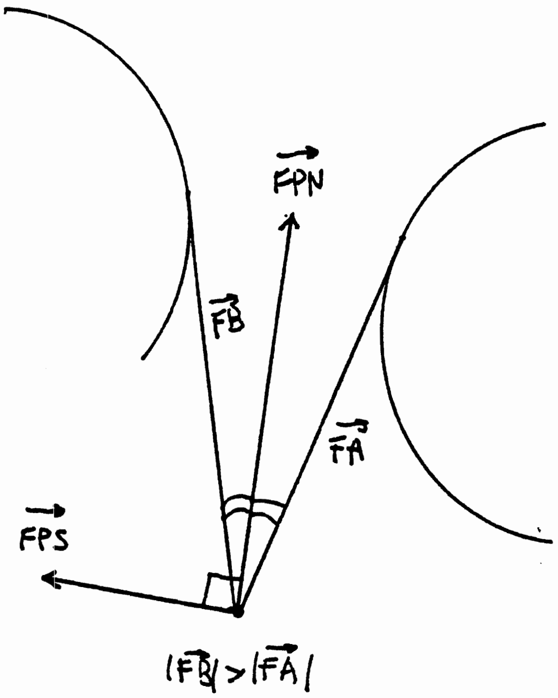


Figure 4.3. British Leyland Slippage Between Belts

irrelevant to the effect of the belt on the contact ellipse and on the tendency of the belt to slip over the contact ellipse. The directions of the force vectors at the fixed point for the purposes of determining slip force were defined as parallel to the directions of the corresponding anchor points as seen from the two tangent points (with the line-of-sight exception raised in the last section). All the other definitions are carried over from the previous section in terms of these two new force vectors and their resultant. Figure 4.4 illustrates the situation at the fixed point and how it corresponds to the situation pictured in Figure 4.3 at the common anchor point.

UMTRI also adapted the "F.9" data cards which were added by British Leyland to input slip algorithm quantities, as well as made other necessary input changes. Appendix A presents the updated Input Description pages covering these changes.

4.2.1 Definition of Terms for Belt Slip Algorithm. It is useful to define four new terms with which to describe the combining of multiple CVS belts.

1. **Belt Node** is either any belt anchor point or any belt fixed point for which infinite friction is prescribed. There is a maximum of 24 belt nodes (two or three per belt). Any belt node is characterized by the number of the belt on which the point is situated together with an indicator designating which point (A = Anchor A, F = Fixed Point, or B = Anchor B).

2. **Belt Segment** is the portion of any belt which connects any two consecutive belt nodes. There is a maximum of 16 belt segments (one or two per belt). Any belt segment can be characterized by the designation of the belt nodes which it connects; however, usually a belt segment will be characterized by its number in the ordered list of belt segments which comprise a belt loop (see remaining definitions).

3. **Slip Point** is either a terminating belt node (Anchor B) of one belt segment which is coincident in both location and attachment system with the initial belt node (Anchor A) of another belt segment or a fixed point on any belt segment and in either case for which slipping is prescribed. There is a maximum of 15 slip points for all belts. Slip Points can be characterized by the belt nodes which form them or, as in the case of belt segments, by its number in the ordered list of slip points which connect the ordered list of belt segments, respectively.

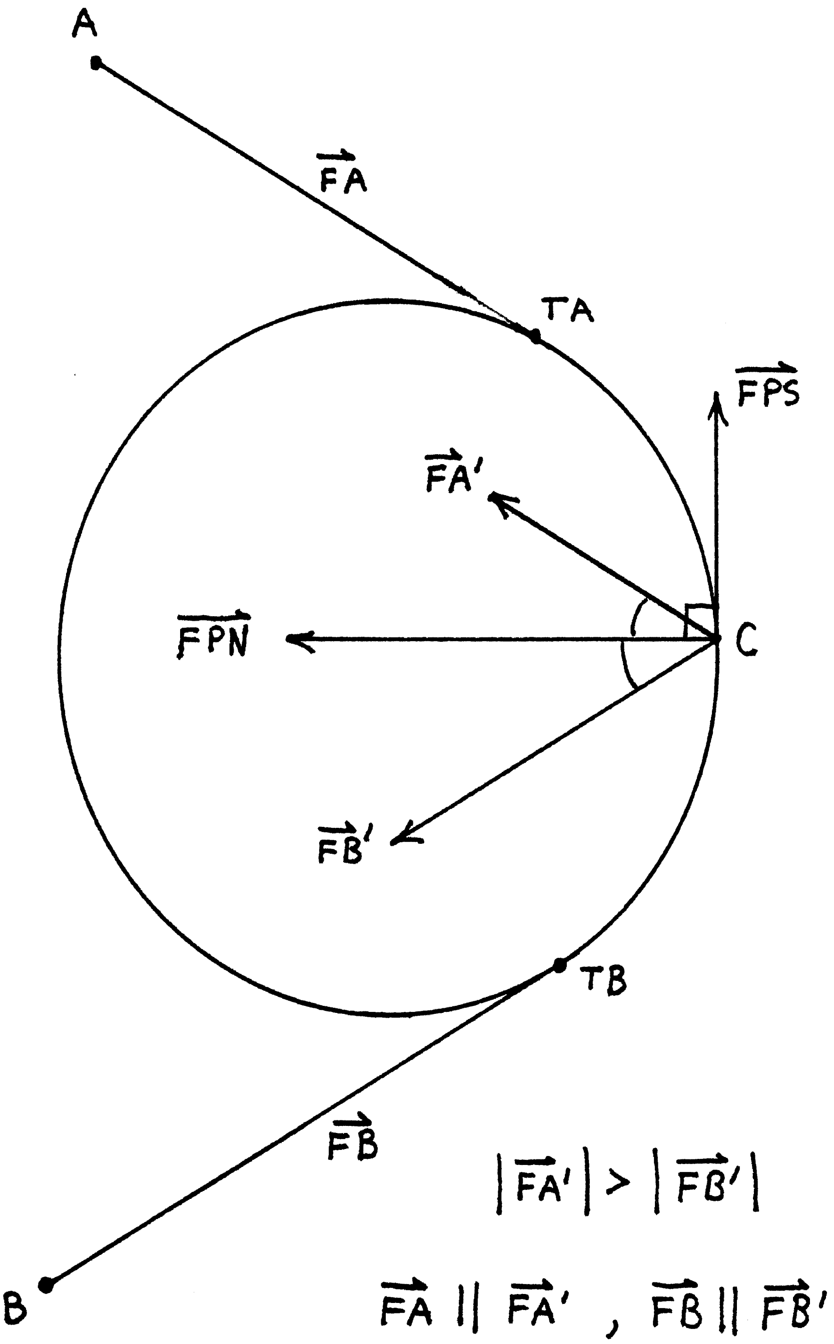


Figure 4.4 Modified Slippage at Fixed Point

4. **Belt Loop** is any group of belt segments for which all but exactly two of their constituent belt nodes are slip points connecting belt segments within the group. The two belt nodes which are not slip points within the loop can not be slip points at all and are called loop anchors. There is a maximum of eight belt loops. Belt loops are characterized by the two loop anchors, by the ordered list of belt segments which comprise it, or by the ordered list of slip points which hold it together. Every belt loop is necessarily independent. The typical belt loop in what follows will consist of N belt segments and N-1 slip points.

4.2.2 Belt Slip in the Presence of Slack. Observations of sled tests appear to reveal the characteristic that belts freely slip until all slack is taken up. At this time the slipping mechanism changes and becomes very restricted. This type of slipping behavior has been added by UMTRI by looking at each belt segment individually and prorating unstrained total belt length for the entire belt loop to the belt segment according to the ratio of the belt segment's strained length to the total strained belt length for the entire belt loop. Expressed in equation form, let l_k be the strained belt length for the "k-th" belt segment in the loop. Let L_k be the unstrained belt length for the "k-th" belt segment in the loop.

Let l_T be the total strained belt length over the entire belt loop.

Let L_T be the total unstrained belt length over the entire belt loop.

Then if $L_T \geq l_T$, the unstrained segment lengths are determined by

$$L_k = \lambda l_k \qquad \text{for } k = 1, \ldots, N$$

where

$$\lambda = \frac{L_T}{l_T}$$

and all belt forces are defined to be zero. If this free-slipping condition is not met, then the methods of the succeeding subsections are used.

4.2.3 Belt Slip Under Load at a Typical Slip Point. Belt segment k-1 will connect to belt segment k at slip point k-1 where k can be any of 2, ... , N. Assume that the belt segments are both producing belt tension. The following equation represents the relationship between the belt segment tension forces at the slip point which is necessary to achieve slip equilibrium. (See Appendix B for the derivation of this expression.) Figure 4.5 shows the typical slip point and pertinent quantities.

$$\lambda_{A,k-1} \tilde{F}_{k-1} + \lambda_{B,k-1} \tilde{F}_{k} + \lambda_{C,k-1} = 0$$

where the tilde written over any variables signifies "at equilibrium"

$\lambda_{A,k-1} = \sin(\Omega_{k-1}) + \eta_{k-1} h_{1,k-1} \cos(\Omega_{k-1})$

$\lambda_{B,k-1} = -\sin(\Omega_{k-1}) + \eta_{k-1} h_{1,k-1} \cos(\Omega_{k-1})$

$\lambda_{C,k-1} = \eta_{k-1} h_{2,k-1}$

Ω_{k-1} = one-half of included angle between belt segments

$h_{1,k-1} = \mu_{k-1} + \frac{1}{2} C_{1,k-1}$

$h_{2,k-1} = \frac{1}{2} C_{2,k-1}$

η_{k-1} = relative direction indicator for slip resistance

μ_{k-1} = coefficient of friction

$C_{1,k-1}$ = coefficient of threshold force

$C_{2,k-1}$ = constant threshold force

This is the characteristic equation of belt slipping equilibrium at each slip point. The computational problem is to adjust the unstrained belt segment lengths throughout the belt loop so that this characteristic equilibrium equation is simultaneously satisfied at all slip points in the belt loop. In what follows, consideration is limited to a maximum of three slip points and four belt segments for each belt loop. Figure 4.6 illustrates one such configuration for a belt loop involving two belt contact ellipsoids.

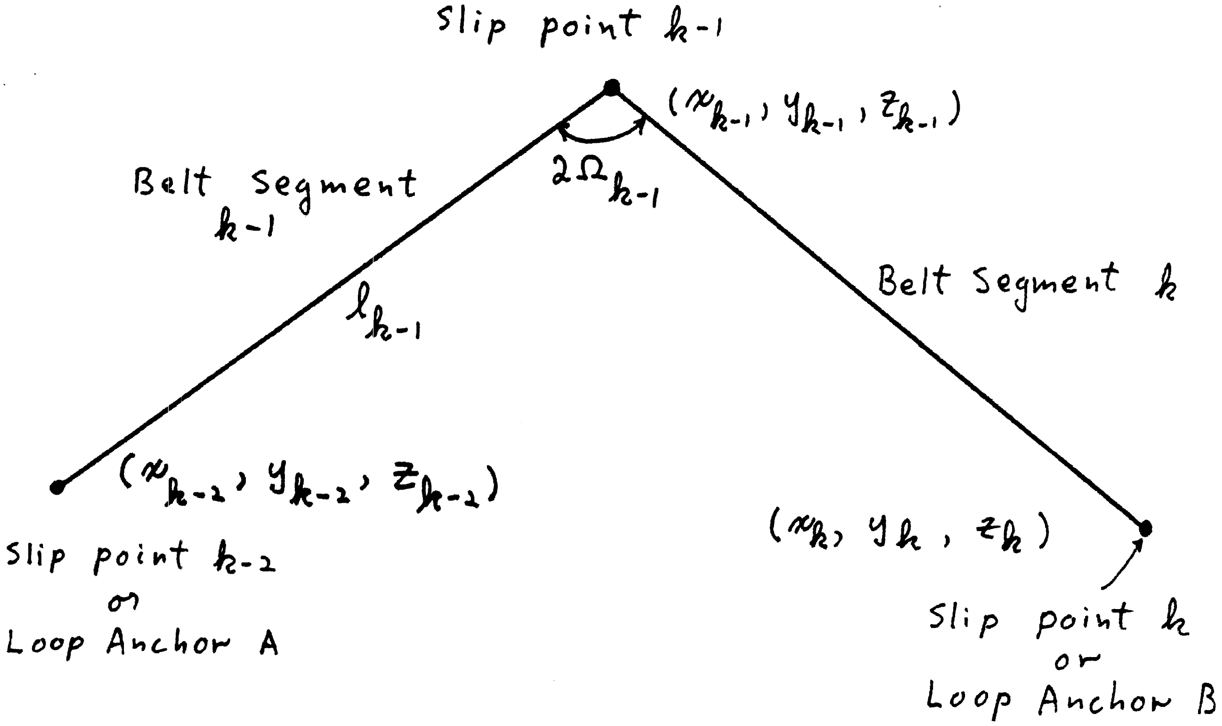


Figure 4.5. The Typical Slip Point k-1

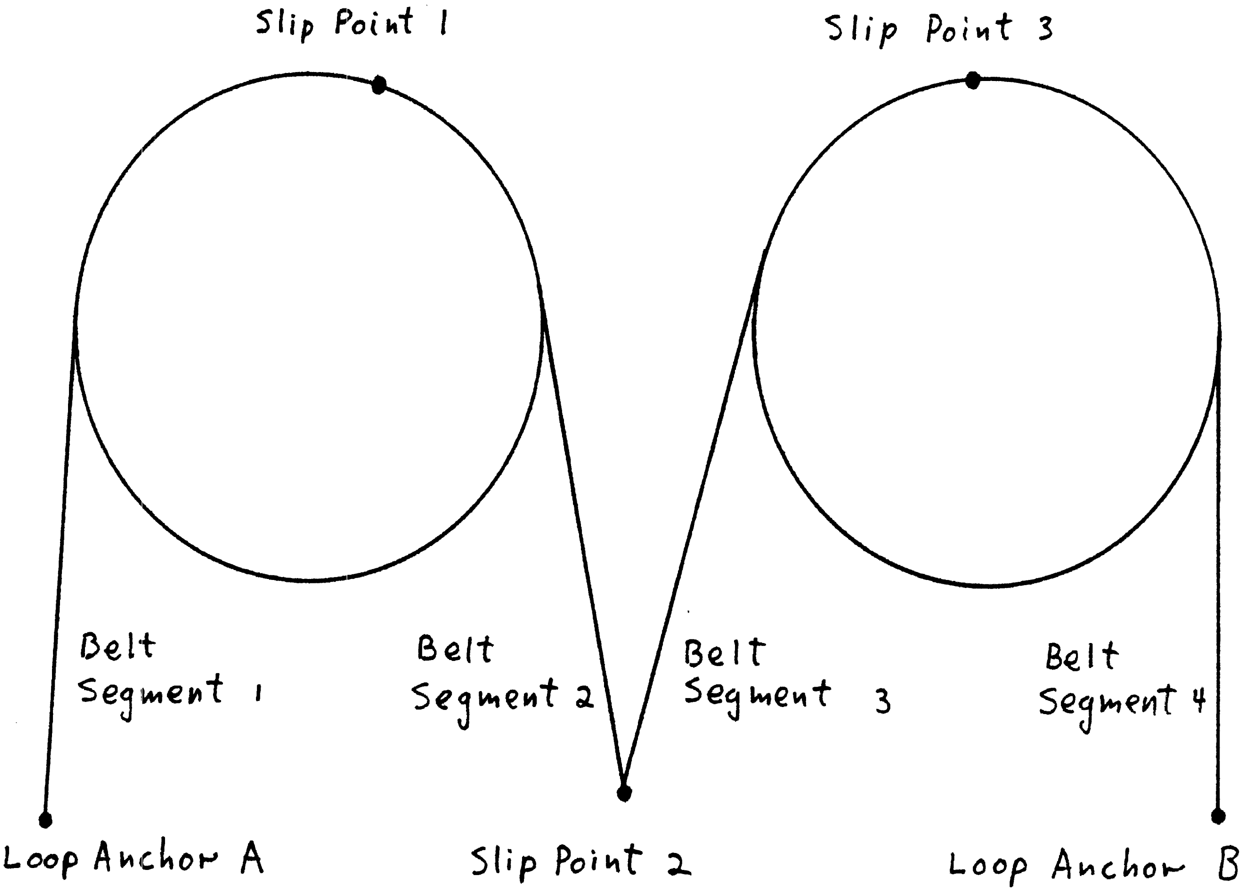


Figure 4.6. A Maximum Belt Loop Configuration

4.2.4 The Progressive Solution Approach. Belt slipping over a whole belt loop can be conceived as a sequence of slips at individual slip points. If this approach were valid, then the computational problem would be reduced to several simpler tasks (searching for the zero of a non-linear equation in one variable).

The non-linear equation is simply the equilibrium condition equation for the slip point k-1. The force for belt segment k-1 is a function of the strained and unstrained belt lengths for belt segment k-1, of which only the unstrained belt length is subject, during the iteration process. Likewise, the force for belt segment k is a function of the strained and unstrained belt lengths for belt segment k of which only the unstrained belt length is subject to change in the iteration. By looking at slipping only at slip point k-1, it follows that the change in the unstrained belt length for segment k-1 must be equal and opposite to the change in the unstrained belt length for segment k. The following can then be written:

$$\tilde{L}_{k-1} = L_{k-1} + \Delta L_{k-1}$$

$$\tilde{L}_{k} = L_{k} - \Delta L_{k-1}$$

where ΔL_{k-1} is the unstrained belt length added to belt segment k-1 by slipping and is the only unknown.

An effective algorithm for the solution of the non-linear equation was put together from similar work in earlier models and makes use of the methods of Newton, the secant, and halving the interval.

This approach worked very well for single slip point belt loops with all possible belt material properties, but it was very capricious in cases involving more than one slip point. Sometimes the results would converge very quickly and smoothly and other times the results would oscillate chaotically. The apparent flaw in this approach was the selection criterion for determining the order to do the slip points. Although this problem is completely analogous to the selection problem occurring during the relaxation method, no attempt was made to investigate selection from this point of view since the needed partial derivatives are both complex and difficult to obtain accurately when involving tabular information. It was that there must be some appropriate order of individual slip point adjustments that would converge for any given case since the actual slipping (at least at the quantum level) theoretically occurs this way. It was also felt that some other approaches which seemed to promise a quicker solution should be tried.

4.2.5 The Simultaneous Solution Based on Change in Unstrained Length for Belt Segment 1. A linear approximation to the belt material properties yields:

$$F_{k-1} = K_{k-1} \left(\frac{l_{k-1}-L_{k-1}}{L_{k-1}}\right)$$

and

$$F_k = K_k \left(\frac{l_k-L_k}{L_k}\right)$$

where K_{k-1}, K_k are the effective linear coefficients of force versus strain. Note that while there is only one set of material properties over both belt segments, it is likely that the two belt segments will be in different strain ranges and therefore will need separate linear approximation.

For each belt segment k, a variable Δ_k can be defined as

$$\Delta_k = \tilde{L}_k - L_k$$

Δ_k is the amount of additional unstrained belt segment length for each belt segment in order to reach equilibrium (may be positive or negative). Since the total unstrained belt length over each loop is invariant during this iteration, therefore it is true that

$$\sum_{k=1}^{N} \Delta_k = 0$$

We can write

$$\lambda_{A,k-1}\left(\frac{l_{k-1}-L_{k-1}-\Delta_{k-1}}{L_{k-1}+\Delta_{k-1}}\right) + \lambda_{B,k-1}\left(\frac{l_k-L_k-\Delta_k}{L_k+\Delta_k}\right) + \lambda_{C,k-1} = 0$$

and solving for Δ_k, we get

$$\Delta_k = \frac{a_{2,k-1}\Delta_{k-1}+a_{4,k-1}}{a_{1,k-1}\Delta_{k-1}+a_{3,k-1}}$$

where

$$a_{1,k-1} = \lambda_{A,k-1}K_{k-1} + \lambda_{B,k-1}K_k - \lambda_{C,k-1}$$

$$a_{2,k-1} = a_{A,k-1}L_k - \lambda_{B,k-1}K_k l_k$$

$$a_{3,k-1} = -a_{1,k-1}L_{k-1} + \lambda_{A,k-1}K_{k-1}l_{k-1}$$

$$a_{4,k-1} = -a_{1,k-1}L_{k-1}L_k + \lambda_{A,k-1}K_{k-1}L_k l_{k-1} + \lambda_{B,k-1}k_k L_{k-1}l_k$$

Since this is a forward recursion relationship, it can be used to solve for each Δ_k in terms of Δ_1 and then these can be substituted into $\sum_{k-1}^{N} \Delta_k = 0$ to get a polynomial in terms of Δ_1. For N=2, the expression is a quadratic; for N=3, it is a cubic; for N=4, a quartic. For the last case with reasonably typical values of physical quantities the coefficients of the quartic ranged from 10^{44} to 10^{23}. These were big enough to cause exponent overflow using a good real polynomial solver on the University of Michigan Amdahl. When the coefficients were scaled by the leading coefficient, good zeroes were always obtained even through an occasional false "lack of convergence" was indicated. It was possible to demonstrate that there was loss of significance of at least eight, and as much as twelve, places during this calculation; however, since sixteen places were carried, the results were good enough.

It was necessary to devise an algorithm to select the appropriate root. This turned out to be easily done by choosing a real root which yielded a positive unstrained belt length which was not greater than the total unstrained belt length.

For materials which were linear, the results were always correct. For non-linear materials, the results always overshot. This type of behavior was expected since it was anticipated that this linear approximation based on the first derivative would act like Newton's Method. This did not turn out to be the case since when repeated applications of this algorithm were tried, cyclical behavior was noted which after a point did not improve the results further.

4.2.6 The Simultaneous Solution Based on Deflection of Belt Segments. It was felt that perhaps the numerical properties could be improved if we were to solve for optimal deflection instead of the change in unstrained belt segment length unless the coefficients were driven still larger. While it is true that physically the length of webbing which slips is the variable to be determined, mathematically we could choose any variable in terms of which we could write the unstrained belt length. So force was taken to be

$$F_k = K_k\left(\frac{\delta_k}{l_k-\delta_k}\right)$$

and substitute as before to come up with another forward recursion relationship in δ_k

$$\delta_k = \frac{a_{1,k-1}\delta_{k-1}+a_{2,k-1}}{a_{3,k-1}\delta_{k-1}+a_{4,k-1}}$$

where

$$a_{1,k-1} = l_k[\lambda_{C,k-1}-\lambda_{A,k-1}K_{k-1}]$$

$$a_{2,k-1} = -\lambda_{C,k-1}l_{k-1}l_k$$

$$a_{3,k-1} = -\lambda_{A,k-1}K_{k-1} - \lambda_{B,k-1}K_{k-1} + \lambda_{C,k-1}$$

$$a_{4,k-1} = l_{k-1}[\lambda_{B,k-1}K_k-\lambda_{C,k-1}]$$

As before this expression was used to write all δ's in terms of δ_1 and then substituted into the equation resulting from the fact that the sum of all deflections over the loop is invariant during the iteration.

This approach caused a small improvement in coefficient size and in loss of significance but no improvement in non-linear behavior. It was realized that the huge coefficients were due to many applications of the recursion relationship and it was decided that the number of applications could be reduced by starting in the middle and working both ways instead of starting at one end and working across. So the forward recursion relationship was solved backwards to obtain:

$$\delta_{k-1} = \frac{a_{4,k-1}\delta_k - a_{2,k-1}}{-a_{3,k-1}\delta_k + a_{1,k-1}}$$

where the a's are the same as before.

This made a dramatic improvement in coefficient size (now ranging 10^{23} to 10^{12} and in loss of significance to only four or five places, but did not completely solve the non-linear convergence problem.

4.2.7 The Simultaneous Solution Based on Strain for Belt Segment 2. It was felt that the lack of convergence for linear approximations of non-linear material properties was probably due to the form of the definition of force since it was force which was being controlled. It was decided to solve directly in terms of strain and then compute unstrained belt length and deflection from the expressions:

$$L_{k-1} = \frac{l_{k-1}}{1+S_{k-1}}$$

and

$$\delta_{k-1} = \frac{S_{k-1} l_{k-1}}{1+S_{k-1}}$$

where S_{k-1} is strain for belt segment k.

The force equation was greatly simplified as

$$F_{k-1} = K_{k-1}S_{k-1}$$

and the recursion relationships became

$$S_k = a_{1,k-1} + a_{2,k-1}S_{k-1}$$

and

$$S_{k-1} = \frac{1}{a_{2,k-1}(S_k - a_{1,k-1})}$$

where

$$a_{1,k-1} = \frac{-\lambda_{C,k-1}}{K_k\lambda_{B,k-1}}$$

and

$$a_{2,k-1} = \frac{-\lambda_{A,k-1}K_{k-1}}{\lambda_{B,k-1}K_k}$$

This leads to easy solution of all strained in terms of S_2, but now the invariant takes a more complex form

$$\sum_{k=1}^{N} \left(\frac{S_k l_k}{1+S_k}\right) = \delta_T$$

This yields coefficient expressions which are as complex as those from the other approaches but promise to yield smaller sizes for coefficients. Unfortunately, it was not possible to finish and test this approach within the resources of the project and extension. The original equations were developed. A satisfactory hand case for the N=2 linear case was done. However, a dimensional check and rederivation of the equations were not accomplished.

4.2.8 Set Up of the Slip Resistance Direction Factors.

In the proceeding several sections, we have made use of the Y_{k-1} slip resistance direction factors saying only that they must be properly chosen for a valid solution. In this section, the algorithm for determining the proper choice of these factors is discussed.

For the N=2 case, it is clear that the proper choice is to oppose belt slip from the lower tension belt segment to the higher tension belt segment (since slipping would stop before it would overshoot).

Likewise it is clear that any monotonic arrangement of belt segment tensions in a multiple slip point case would remain in that configuration at equilibrium (with reduced tension differences). It is also clear that there are arrangements in which two relatively small belt tensions could be reversed by the presence of a much larger belt tension nearby on the way to equilibrium.

The approach used was to evaluate the given configuration, find the maximum tension belt segment and proceed in both directions from there searching for reversals in monotonicity (which for the N = 4 case can be a maximum of two). For each of these reversals, the problem is solved with both choices of y_k for the affected belt segments (a maximum of four solutions). Finally, the single solution is chosen which minimizes total belt slipping as the correct solution. This approach may be more laborious than necessary, but until an algorithm which would give good results was developed, it was not possible to investigate improvements in this procedure.

4.3 Summary of Progress

While a major portion of the effort in this project was spent investigating a suitable general method for the solution of the slip equilibrium equations at all slip points, several other tasks were necessary to integrate any such algorithm in the Calspan CVS, Version 20 code. A new Slip Input routine along with modifications to other input routines was also necessary. These have been coded and tested both in isolation and within the model.

It was also necessary to modify the belt routines so that they would work along with the new belt slip routines. The belt slip routines determine the situation, handle any slack, set up the problem for solution, do the solution for slipping, compute the quantities to be outputted and records these for later printing. These routines have also been coded and tested in isolation and also with a dummy solution routine within the CVS model. Finally, the output changes necessary to print the output quantities have been coded and tested.

It was necessary to write a separate test program for the various approaches to the equilibrium equation solution routine and determine the numerical characteristics of each approach. The plan was that when the slip solver was satisfactory in isolation, it would be inserted into the CVS code and final testing of the whole package would take place.

4.4 Recommendations

It is recommended that the strain approach discussed in Section 4.2.7 be implemented, tested, and the numerical characteristics determined.

If this fails, it is felt that an incremental approach should be tried where controlled steps are taken in the direction determined by a Newton-type or secant-type method. The numerical studies already done give some hope that this approach could be successful.

If both of these approaches fail, it is felt that everything done to this point should be reviewed and carefully reevaluated before anything else is considered.

The reasoning behind these recommendations is that the concept and the algorithms are inherently simpler than those used in the more sophisticated HARNESS routines which are not yet working entirely satisfactorily. It is estimated that these concepts are satisfactory for many standard three-point belt applications, and when completely installed and debugged, should be relatively inexpensive and easy to use.

5.0 THE HARNESS ALGORITHM

An advanced restraint system submodel (HARNESS) has been under development of a number of years by Fleck and associates (Butler et al. 1975, 1980, 1983). The model requires the user to specify a set of potential reference points on ellipsoids which are in contact with the belts comprising the harness (see Figure 5.1). The anchor points do not have to be fixed to the same segment. They can also be attached to another belt. Each belt may pass over more than one ellipsoid. Major aspects of this code are described as follows.

1. The position of the reference points is mobile. That is, they can move over the surface of the ellipsoid to represent belt sliding. This sliding can include frictional effects.

2. The reference points can penetrate the ellipsoid to represent the deformation of the non-rigid body based on its force-deformation characteristics.

3. If the force is removed from the belt at a reference point, the belt can lift off from contact.

This subprogram has potential, not present in the older BELT subprogram, to overcome problems observed (see Section 2) in laboratory tests of belt hardware. These include:

1. Slippage of belt over the surface in non-frontal impacts and release from contact if the occupant slips out of the belt.
2. Modeling of submarining.
3. Modeling of the complete three-point belt system in three dimensions.
4. Modeling of the belt/chest interaction.
5. Attachment of belts to non-rigid anchors.

During the early stages of the project, UMTRI was informed by the sponsor of development (USAF) that the software was essentially non-frictional. By the end of 1984, some modest successes were reported. By the end of 1985, UMTRI was provided with a short report (Obergefell and Kaleps 1984) and a list of code modifications to Version 20 of the CVS software. An example was included for a rather sophisticated Air Force harness (see Figure 5.2). It was indicated that

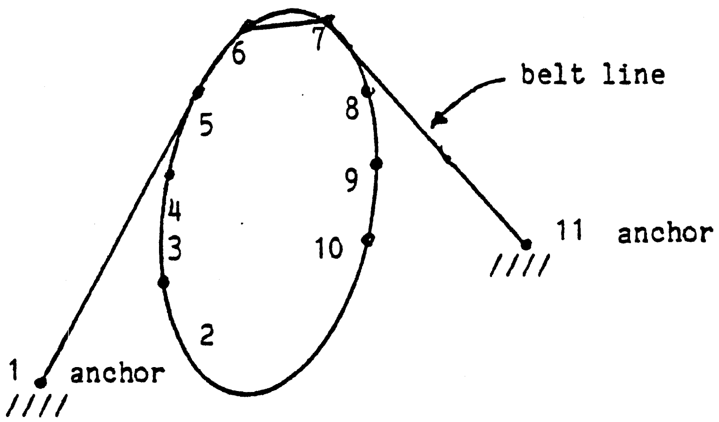


Figure 5.1. Points on Simple Harness.

most features worked but that great care had to be used in developing data sets. It is clear that a body of user knowledge must be developed before this code is ready for general use.

It is the recommendations of UMTRI that a considerable effort be expended to build a body of user knowledge on real problems, starting with a matrix of simple test cases. Most design problems involving belts can be handled with this code. Because the code was not recommended for use during the formal work on the project without a considerable expenditure of time and money, no effort was made to exercise it.

HARNESS MODEL - MIGRATION AND PENETRATION POSSIBLE (WITH GREAT CARE). CODE CAN BE OBTAINED

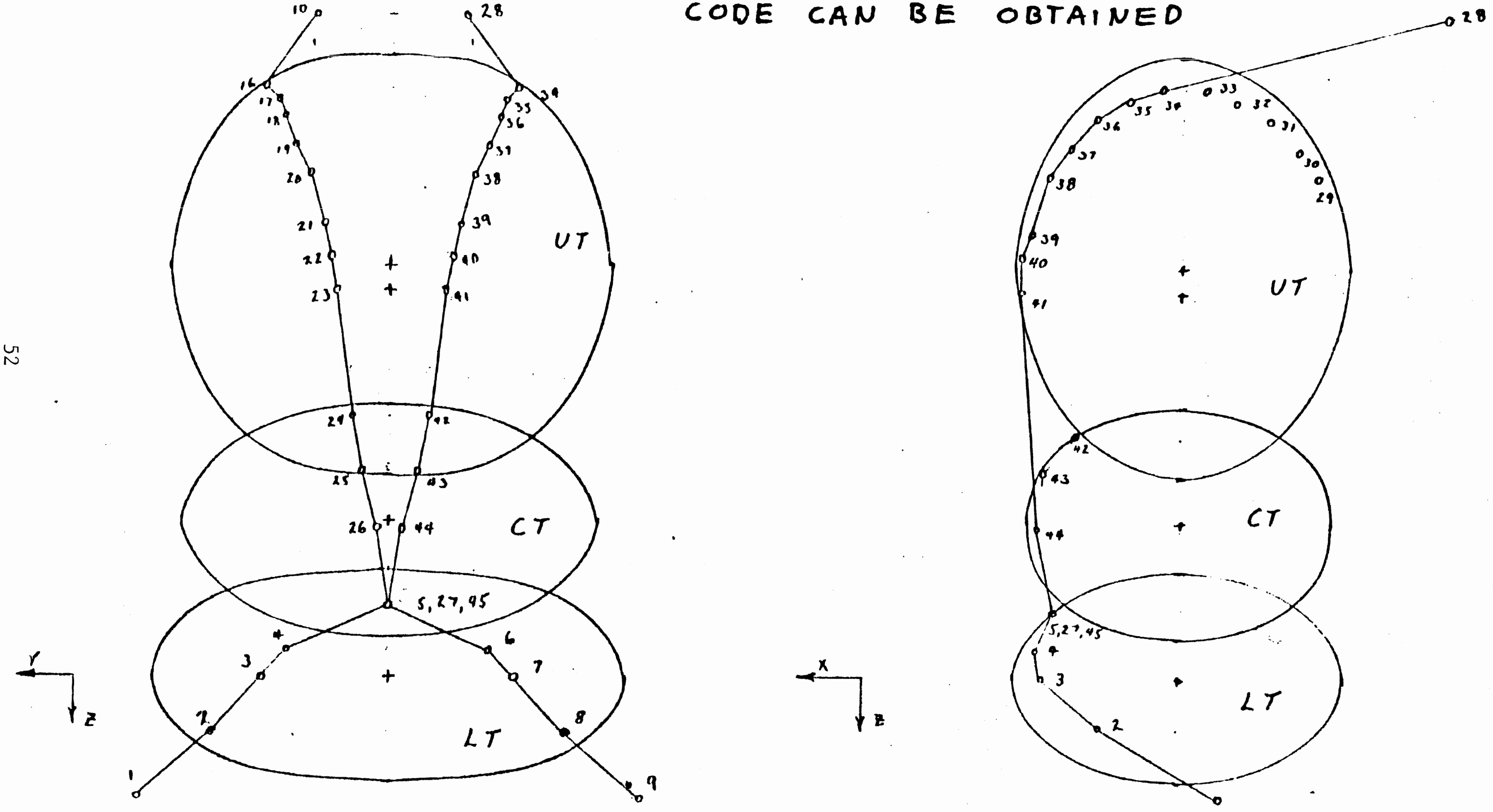


Figure 5.2. Air Force Harness Simulation Using HARNESS.

6.0 UMTRI VERSION OF GMCVS

The UMTRI version of GMCVS includes a variety of improvements as well as modifications which were required in order for the program to function on MTS. In particular, the program was split into two separate processors. The first one was used to produce output. Most of these improvements and changes had already been developed during a previous contract with NHTSA (Bennett and Robbins 1982).

The Calspan 3-D CVS, Version 20, as it came to UMTRI from GM, made use of well over three megabytes of virtual storage. This large use of storage is considered excessive on the MTS System and usually causes job cancellation. Also, previous versions of the Calspan 3-D CVS made use of a separate logical device number for each output page as a means of sorting information into output pages as the run proceeded. This practice was not permitted on the MTS System. To cope with these problems during previous contracts with NHTSA, UMTRI reorganized the output sections using direct access (or indexed) input/output to achieve the sorting of information into pages. This reorganization was easily achieved in the past.

In recent versions such as CVS18 and CVS19, new output options were added requiring access to output quantities from previous time points in the simulation and so storage arrays were set up to hold this information. The options of printing all pages on one logical device number at the end of the run and of recording the information in an exterior file (sequentially in simulated time) were also added while keeping the ability to output on multiple logical device numbers. The increased complexity of the output sections and the huge amount of information now stored required the alternate use of three direct access hold files (due to file size restrictions) and multiplied the amount of effort required for the reorganization at UMTRI.

While the necessary reorganization was being carried out, UMTRI installed a number of useful output options which had already been developed in our previous work with NHTSA. As well as the two output recording options supported in Version 20 (every integration step or every evaluation) it is now possible to record in output at equal increments of time. Regardless of which information recording concept is used, it is possible to print output in equal increments of integral multiples of DT or to print all information present.

The kinematics printout controlled by the "H" cards now allow the user several new options including:

- Specification of the coordinate system in which each output quantity is printed with the original conventions being used as a default;
- Optional printout of contact information both as to category of information and individual interaction; and
- Specifications of the order of printing by category of information.

Discrete use of these options enable the user to cheaply determine the important trends in a run and then set up more detailed printout and graphic displays of pertinent information. This is now possible because the output sections of the program are in a separate processor and can be rerun as often as useful as long as the hold files are kept intact.

Several parts of the general CVS program were not implemented in the UMTRI versions. These included HIC, SI, the variable graphics printer plots, the equilibrium positioning software, the RESTART option, and the VIEW graphics postprocessor. The first four were not implemented since alternatives such as the Validation Command Language Postprocessor (Bennett et al. 1979, Bennett 1983) were available from previous NHTSA work as part of the reorganization of output storage and processing.

The RESTART option was not implemented in GMCVS and so was left out of the UMTRI version. However, it is strongly recommended that this software be implemented as it gives the user the capability to interactively run the program and change the input data structure as simulation proceeds. This can be very useful, particularly in controlling integration time step, during long simulations such as rollovers. Interim graphic output should also be available at local workstations to determine whether to continue, change data and continue, or stop.

The connection to the VIEW postprocessor was also not implemented as the required DISSPLA software package was not installed on at The University of Michigan mainframe. The alternative was the Ellipsoidal Man Plot Postprocessor (Lehman et al. 1983) which was developed under NHTSA contract. It should be noted that the VIEW program cannot include more than one ellipsoid per segment. This limitation has been recently illustrated in attempts to compare graphic results of steering column simulations generated by GMCVS with those generated at UMTRI using the simpler, but more flexible, software.

One code modification was made in the UMTRI version in order to meet the need for a general vehicle motion input vector in simulations of general vehicle rollovers (Robbins 1986).

7.0 REFERENCES

Alem, N.M., Benson, J.B., and Melvin, J.W. (1977) Whole Body Response of the Hybrid III Anthropomorphic Test Device. Report no. UM-HSRI-77-10. University of Michigan Transportation Research Institute, Ann Arbor. 235p.

Bennett, R.O. (1983) Validation Command Language: Supplement to 1979 Version. Report no. UMTRI-83-28. University of Michigan Transportation Research Institute, Ann Arbor.

Bennett, R.O., and Robbins, D.H. (1982) HSRI Version of the Improved Three-Dimensional Computer Simulation of Vehicle Crash Victims. In three volumes, Volume I: Analysis, Volume II: User's Guide, Volume III: Programmer's Guide. Final Draft. University of Michigan Transportation Research Institute, Ann Arbor.

Bennett, R.O., Becker, J., and Robbins, D.H. (1979) Validation Command Language, 1979 Version. Report no. UM-HSRI-79-6. The University of Michigan, Highway Safety Research Institute, Ann Arbor.

Butler, F.E., and Fleck, J.T. (1975) Development of an Improved Computer Model of the Human Body and Extremity Dynamics. AMRL-TR-75-14. Wright-Patterson AFB, Ohio.

Butler, F.E., and Fleck, J.T. (1980) Advanced Restraint System Modeling. AMRL-TR-80-14. Wright-Patterson AFB, Ohio.

Butler, F.E., Fleck, J.T., and DiFranco, D.A. (1983) Modeling of Whole-Body Response to Windblast. AFAMRL-TR-83-073. Wright-Patterson AFB, Ohio.

Fleck, J.T., and Butler, F.E. (1981) Validation of the Crash Victim Simulator, Volume 1, Engineering Manual--Part I: Analytical Formulation. Report no. ZA-5881-V-1. Calspan Corporation, Buffalo, New York. 264p.

Fleck, J.T., Butler, F.E., and Vogel, S. L. (1975) An Improved Three-Dimensional Computer Simulation of Vehicle Crash Victims. In four volumes, Volume 1: Engineering Manual, Volume II: Model Validation, Volume III: User's Manual, Volume IV: Programmer's Manual. Report nos. DOT-HS-801507 through DOT-HS-801510. NTIS order numbers PB-241 692 through PB-241 695.

Horsch, J.D. (1980) "Occupant Dynamics as a Function of Impact Angle and Belt Restraint," Proceedings of the 24th Stapp Car Crash Conference, pp. 417-438. Society of Automotive Engineers, Warrendale, Pa.

Lehman, R.J., Bennett, R.O., and Robbins, D.H. (1983) Ellipsoidal Man Plotting Package for MVMA 2-D and CVS (HSRI Version) Occupation Motion Models. Report no. UMTRI-83-27. University of Michigan Transportation Research Institute, Ann Arbor.

Newman, K.F., Grew, N.D., and Dowzall, G. (1981) "Cal 3-D: Its Use in BL Cars Ltd." Proceedings: CAL 3-D Users Conference. Motor Vehicle Manufacturers Association and the National Highway Traffic Safety Administration.

Robbins, D.H. (1986) 3-D Simulation of Occupant Kinematics in Rollovers. Report no. UMTRI-86-44. University of Michigan Transportation Research Institute, Ann Arbor.

Viano, D.C., and Culver, C.C. (1981) "Test Dummy Interaction with a Shoulder or Lap Belt," Proceedings of the 25th Stapp Car Crash Conference, pp. 583-609. Society of Automotive Engineers, Warrendale, Pa.

Wittenburg, J. (1977) Dynamics of Systems of Rigid Bodies. B.G. Teubner, Stuttgart, 222p.

APPENDIX A

REVISION PAGES TO GMCVS INPUT DESCRIPTION

The following pages are replacement pages for the Calspan CVS model, Version 20, Input Description covering the changes made to the input sections of the model in order to provide a structure for the algorithms presented in Part 4 of this report.

IF NBLT IS NONZERO ON CARD D.1, NBLT SETS OF CARDS D.3 ARE REQUIRED.

CARD D.3.A	FORMAT (5A4)
BLTTTL(I,J),I=1,5	A 20 CHARACTER DESCRIPTION OF THE JTH BELT.
CARD D.3.B	FORMAT (6F12.0)
BELT(I,J),I=1,3	POSITION OF ANCHOR POINT A FOR THE JTH BELT W.R.T. SEGMENT NSEGA (SPECIFIED ON CARD F.2.B IN NS(1) FIELD). X, Y, AND Z COORDINATES (IN.)
BELT(I,J),I=4,6	POSITION OF ANCHOR POINT B FOR THE JTH BELT W.R.T. SEGMENT NSEGB (SPECIFIED ON CARD F.2.B IN NS(1) FIELD). X, Y, AND Z COORDINATES (IN.)

NOTE: THE PROGRAM MUST PASS A PLANE THROUGH THE THREE POINTS, ANCHOR POINT A, ANCHOR POINT B, AND A FIXED POINT ON THE CONTACTED BODY SEGMENT. IF ANCHOR POINTS A AND B COINCIDE, THEY MUST BE SEPARATED SLIGHTLY FOR INPUT SUCH THAT THE DESIRED BELT PLANE WILL BE DEFINED.

CARD D.3.C FORMAT (5F12.0)

BELT(I,J),I=7,9	POSITION OF BELT FIXED CONTACT POINT WITH RESPECT TO SYSTEM THAT HAS ORIGIN AT ELLIPSOID CENTER AND IS PARALLEL TO LOCAL BODY SEGMENT SYSTEM. (X, Y, AND Z COORDINATES (IN.))
NSLPBL(J)	NUMBER OF SLIP POINTS FOR THIS BELT. SLIP POINTS MAY OCCUR AT EITHER BELT ANCHOR IF THERE IS ANOTHER BELT WHICH HAS A COINCIDENT ANCHOR WHICH IS ALSO SPECIFIED AS A SLIP POINT. SLIP POINTS MAY ALSO OCCUR AT FIXED POINTS FOR BELTS. (MAXIMUM VALUE IS 3; ENTER AS REAL NUMBER.)
BELT(11,J)	BELT SLACK (IN). THE SLACK IS ADDED TO THE INITIAL GEOMETRIC LENGTH TO OBTAIN THE INITIAL BELT LENGTH. IF NEGATIVE NUMBER IS ENTERED, IT WILL BE INTERPRETED AS INITIAL BELT LENGTH FROM WHICH THE PROGRAM WILL COMPUTE THE SLACK.

NOTE: NSPT IS DEFINED AS THE SUM OF ALL FIXED POINTS PLUS ONE HALF OF ALL ANCHOR POINTS SPECIFIED IN THE NSLPBL FIELDS OF ALL D.3.C CARDS, IN OTHER WORDS, THE TOTAL NUMBER OF INDIVIDUAL SLIP POINTS FOR ALL BELTS.

IF NBLT (THE NUMBER OF BELTS) IS NONZERO ON CARD D.1, CARDS F.2 ARE REQUIRED.

CARD F.2.A FORMAT (8I4)

MNBLT(J),J=1,NBLT	FOR BELT J, THE NUMBER OF SEGMENTS FOR WHICH SEGMENT-BELT INTERACTION IS ALLOWED (0 OR 1 ONLY).

FOR EACH BELT J, MNBLT(J) CARDS OF THE FOLLOWING MUST BE SUPPLIED.

CARDS F.2.B - F.2.N FORMAT (9I4)

NJ	THE BELT NUMBER TO BE CONTACTED, MUST CORRESPOND TO J ABOVE. THERE MUST BE MNBLT(J) CARDS WITH THE SAME NJ. IF MNBLT(J) = 0, NO NJ = J SHOULD BE PRESENT.
NS(1)	SUPPLY NSEGA + 100 * NSEGB WHERE NSEGA IS THE SEGMENT NUMBER FOR ANCHOR A AND NSEGB IS THE SEGMENT NUMBER FOR ANCHOR B. IF NSEGA AND NSEGB ARE THE SAME, SUPPLY ONLY NSEGA. (NOTE 1 APPLIES TO BOTH NSEGA AND NSEGB)
NS(2)	SUPPLY NOBODY WHICH IS THE NUMBER OF THE SEGMENT TO WHICH THE CONTACT ELLIPSOID IS ATTACHED. (NOTE 1 APPLIES TO NOBODY).
NS(3)	THE NUMBER OF THE CONTACT ELLIPSOID ATTACHED TO THE SEGMENT NOBODY.
NF(1)	THE FUNCTION NUMBER FORM CARD E.1 TO DEFINE THE FORCE-DEFLECTION FUNCTION FOR THIS CONTACT. THE ABSCISSA FOR THIS FUNCTION SHOULD BE STRAIN (IN/IN).
NF(I),I=2,4	SAME DEFINITION AS ON CARD F.1.B ABOVE.
NF(5)	IF NON-ZERO, FULL BELT FRICTION IS ASSUMED, I.E., FORCES ARE COMPUTED FOR EACH HALF OF THE BELT SEPARATELY. IF ZERO, ZERO BELT FRICTION IS ASSUMED I.E., BELT TENSION IS THE SAME AT BOTH BELT ANCHOR POINTS. IF THE FIXED POINT FOR BELT NJ IS SPECIFIED AS A SLIP POINT ON A F.9 CARD, THIS FIELD IGNORED.

NOTES:

1. ONE OF THREE OPTIONS IS CHOSEN BY USER:
   (A) THE CARD NUMBER I UNDER CARD B.2.A FOR ANY BODY SEGMENT,
   (B) NVEH FOR THE PRINCIPAL VEHICLE, OR
   (C) NGRND FOR THE INERTIAL FRAME (GROUND).
2. THE USE OF RATE DEPENDENT FUNCTION AS DEFINED UNDER CARDS F.1.B. ARE NOT CURRENTLY OPERATIONAL FOR BELT-SEGMENT CONTACTS.

F.9 SUBROUTINE SLPINP - CARD INPUT FOR SEAT BELT FEED THROUGH DATA.

(NSPT CARDS REQUIRED, SEE NOTE ON CARD D.3.C.)

CARDS F.9	FORMAT (I4,3X,A1,I4,3X,A1,5F10.0)
NBSP(I,J),I=1,4	TWO PAIRS OF VALUES SPECIFYING THE BELT NUMBER AND POINT CODE FOR EACH BELT NODE COINCIDENT AT SLIP POINT J. POINT CODES: A = ANCHOR A B = ANCHOR B F = FIXED POINT (ONLY ONE BELT INVOLVED IF POINT CODE "F".)
SFRICT(1,J)	IF POSITIVE, THE FRICTION COEFFICIENT AT SLIP POINT J. (LBS/LBS). IF NEGATIVE, THE FUNCTION NUMBER FOR FRICTION FORCE (MUST BE NON-NEGATIVE) VS NORMAL FORCE.
SFRICT(2,J)	THE CONSTANT TERM OF THRESHOLD FORCE (MUST BE NON-NEGATIVE). (LBS.) THIS ACTS TOGETHER WITH NEXT QUANTITY.
SFRICT(3,J)	THE COEFFICIENT OF THRESHOLD FORCE (LBS/LBS). THE TOTAL THRESHOLD FORCE IS OBTAINED BY MULTIPLYING THIS BY SLIP NORMAL FORCE AND ADDING SFRICT(2,J).
SFRICT(4,J)	THE INITIAL SLIP INCREMENT TO BE USED. (IN.) (UNUSED AT PRESENT TIME)
SFRICT(5,J)	MAXIMUM RATE OF SLIP WHICH IS ALLOWABLE (IN./SEC.). (UNUSED AT PRESENT TIME)

NOTES: 91. BELT NODES SPECIFIED MUST BE EITHER ONE FIXED POINT OR TWO ANCHOR POINTS WHICH MUST BE COINCIDENT IN BOTH POSITION AND SPECIFICATION OF ATTACHMENT SEGMENT.

92. SFRICT(1,J) AND SFRICT(2,J) CAN NOT BOTH BE SPECIFIED ZERO.

APPENDIX B

DERIVATION OF THE SLIP FORCE EQUILIBRIUM EQUATION

This appendix refers to an amplifies Section 4.2.3 and Figure 4.5. Slip point k-1 is located at $(x_{k-1}, y_{k-1}, z_{k-1})$ in inertial space. Belt segment k-1 terminates at slip point k-1 and begins at slip point k-2 or loop anchor A which is located at $(x_{k-2}, y_{k-2}, z_{k-2})$ in inertial space. Belt segment k begins at slip point k-1 and terminates at slip point k or loop anchor B which is located at (x_k, y_k, z_k) in inertial space. As before let l_k be the strained belt length of belt segment k and let L_k be the unstrained belt length of belt segment k. The strained belt lengths are determined geometrically. Total unstrained belt length for any loop is determined from the initial conditions supplied by the user. Individual unstrained belt lengths are determined to meet the slip equilibrium requirements and maintain the user-specified total unstrained belt length of the loop.

So

$$l_{k-1} = \sqrt{(x_{k-1}-x_{k-2})^2+(y_{k-1}-y_{k-2})^2+(z_{k-1}-z_{k-2})^2}$$

and

$$l_k = \sqrt{(x_k-x_{k-1})^2+(y_k-y_{k-1})^2+(z_k-z_{k-1})^2}$$

Define

$$\bar{f}_{k-1} = \begin{pmatrix} \alpha_{k-1} \\ \beta_{k-1} \\ \gamma_{k-1} \end{pmatrix} \quad \text{and} \quad \bar{f}_k = \begin{pmatrix} \alpha_k \\ \beta_k \\ \gamma_k \end{pmatrix}$$

where

$$\alpha_{k-1} = \frac{1}{l_{k-1}}(x_{k-2}-x_{k-1}) \qquad \alpha_k = \frac{1}{l_k}(x_k-x_{k-1})$$

$$\beta_{k-1} = \frac{1}{l_{k-1}}(y_{k-2}-y_{k-1}) \qquad \beta_k = \frac{1}{l_k}(y_k-y_{k-1})$$

$$\gamma_{k-1} = \frac{1}{l_{k-1}}(z_{k-2}-z_{k-1}) \qquad \gamma_k = \frac{1}{l_k}(z_k-z_{k-1})$$

These are the unit vectors along the two belt segments k-1 and k, respectively based at the slip point and pointing toward the respective anchors.

Now

$$\cos(2\Omega_{k-1}) = \bar{f}_{k-1}\cdot \bar{f}_k = \alpha_{k-1}\alpha_k+\beta_{k-1}\beta_k+\gamma_{k-1}\gamma_k$$

Define

$$\bar{F}_{k-1} = F_{k-1}\bar{f}_{k-1} \qquad \text{and} \qquad \bar{F}_k = F_k\bar{f}_k$$

These are the belt tension force vectors at the slip point along the two belt segments, respectively. British Leyland defines the normal direction as a unit vector called $f_{n,k-1}$ where

$$\bar{f}_{n,k-1} = \frac{\bar{f}_{k-1}+\bar{f}_k}{|\bar{f}_{k-1}+\bar{f}_k|}$$

and the tangential direction as

$$\bar{f}_{s,k-1} = \frac{\bar{f}_{k-1}-\bar{f}_k}{|\bar{f}_{k-1}-\bar{f}_k|}$$

Note that

$$|\bar{f}_{k-1}+\bar{f}_k| = \pm 2 \cos(\Omega_{k-1})$$

and

$$|\bar{f}_{k-1}-\bar{f}_k| = \pm 2 \sin(\Omega_{k-1})$$

These expressions are determined by substitution of components into the definitions of vector magnitudes and application of trignometric identities. British Leyland defined the normal and tangential force magnitudes as the scalar product of the force resultant vector with the two direction vectors.

$$F_{n,k-1} = (\bar{F}_{k-1}+\bar{F}_k) \cdot \bar{f}_{n,k-1}$$

and

$$F_{s,k-1} = (\bar{F}_{k-1}+\bar{F}_k) \cdot \bar{f}_{s,k-1}$$

Substituting definitions and simplifying, the normal force magnitude becomes

$$F_{n,k-1} = (F_{k-1}\bar{f}_{k-1}+F_k\bar{f}_k) \cdot \frac{\bar{f}_{k-1}+\bar{f}_k}{|\bar{f}_{k-1}+\bar{f}_{k-1}|}$$

$$= \frac{(F_{k-1}+F_k)\ \left(1+\cos(2\Omega_{k-1})\right)}{\pm 2 \cos(\Omega_{k-1})}$$

$$= \pm\left(F_{k-1}+F_k \cos(\Omega_{k-1})\right)$$

Likewise for tangential or slip force magnitude,

$$F_{s,k-1} = \frac{(F_{k-1}-F_k)\ \left(1-\cos(2\Omega_{k-1})\right)}{\pm 2 \sin(\Omega_{k-1})}$$

$$= \pm(F_{k-1}-F_k) \sin(\Omega_{k-1})$$

The slip resistance force vector at the slip point ($\bar{F}_{R,k-1}$) is assumed to be due to friction force at the slip point ($\bar{F}_{F,k-1}$) and threshold force at the slip point ($F_{T,k-1}$). Threshold force includes the differences between static and kinetic friction while friction force is includes kinetic effects only.

So that

$$F_{R,k-1} = F_{F,k-1} + F_{T,k-1}$$

where

$$F_{F,k-1} = \mu_{k-1} F_{n,k-1}$$

and

$$F_{t,k-1} = C_{1,k-1}\ F_{n,k-1} + C_{2,k-1}$$

μ_{k-1} is the inputted coefficient of friction,

$C_{1,k-1}$ is the inputted coefficient of threshold force, and

$C_{2,k-1}$ is the inputted constant threshold force.

The direction of $\bar{F}_{R,k-1}$ is opposite to that of $\bar{F}_{s,k-1}$.

In the British Leyland algorithm, slipping will occur if

$$F_{s,k-1} > F_{F,k-1} + F_{T,k-1}$$

and the iteration to adjust the unstrained lengths in order to reduce $F_{s,k-1}$ until $F_{F,k-1}<F_{s,k-1}<F_{F,k-1}+F_{T,k-1}$. This is the interval of convergence.

We will define a target variable V_{k-1} which assumes the value of zero in the center of the interval of convergence as

$$V_{k-1} = F_{s,k-1} + \eta_{k-1}\ F_{R,k-1}$$

where η_{k-1} is assigned the value of +1 or -1 depending on the relative direction of slip resistance forces when equilibrium is reached. Note that slip resistance will always oppose slip force, but if the equation is to predict equilibrium, then it must reflect the direction of slip resistance at equilibrium which may or may not be the same as the direction of slip resistance where the iteration is started.

The equilibrium condition then is taken to be $V_{k-1}=0$.

When computed during the iteration, convergence is defined as

$$|V_{k-1}| \le \frac{1}{2} F_{T,k-1}$$

Note that this condition corresponds to the previous interval of convergence. Now substituting the previously developed expressions

$$V_{k-1} = 0 = (F_{k-1}-F_k) \sin(\Omega_{k-1})+\eta_{k-1} [h_{1,k-1}(F_{k-1}+F_k) \cos(\Omega_{k-1})+h_{2,k-1}]$$

and collecting on F_{k-1} and F_k,

$$\lambda_{A,k-1}F_{k-1} + \lambda_{B,k-1}F_k + \lambda_{C,k-1} = 0$$

where

$$\lambda_{A,k-1} = \sin(\Omega_{k-1}) + \eta_{k-1}h_{1,k-1} \cos(\Omega_{k-1})$$

$$\lambda_{B,k-1} = -\sin(\Omega_{k-1}) + \eta_{k-1}h_{1,k-1} \cos(\Omega_{k-1})$$

$$\lambda_{C,k-1} = \eta_{k-1}h_{2,k-1}$$

APPENDIX C

TAPE DESCRIPTION AND INSTALLATION NOTES

C.1 Introduction

The software produced as part of the CVS belt software project is submitted in the form of magnetic tape number BSC023. The remaining sections of this appendix present a description of the physical layout of tape BSC023, a description of each piece of software included on the tape, and notes concerning installation of this software at GMR including a summary of changes made in the UMTRI version (GUCVS) of the GMR version (GMCVS) of the Calspan CVS, Version 20.

C.2 Tape Description of Tape BSC023

Tape BSC023 is an unlabelled, EBCD, nine-track magnetic tape with eleven files and the following properties: RECFM = FB, BLKSIZE = 2400, and LRECL = 80 written at 6250 BPI. Table C.1 summarizes the contents by file.

TABLE C.1

DESCRIPTION OF TAPE BSC023

File No.	Code Name	Description	Seq ID (col 73-74)	No. of Records	No. of Blocks
1	ODIC1.FORT	GMCVS Part I	num 73-80	6,824	228
2	ODIC3.FORT	GMCVS Part II	num 73-80	6,443	215
3	CIGS	GUCVS Part I	IG	13,080	436
4	CUTS	GUCVS Part II	UT	1,363	46
5	ROBMULTESTS	Deflection 2 Code	ED	830	28
6	ROBEP1MULT	Unstrained Change	EP	792	27
7	ROBDELOMULT	Deflection 1 Code	DZ	811	28
8	RIGS	Slip GUCVS Part I	RI	3,395	114
9	RUTS	Slip GUCVS Part II	RU	353	12
10	ROBPLOTCS.S	Ellip Man Plotter	CS	1,330	45
11	ROBMULTEST.D	Test Data for 5-7	None	168	6

C.3 Software Descriptions

ODIC1.FORT and ODIC3.FORT are duplicates of the GMCVS files from which UMTRI started. They are included to serve as a reference point from which to understand the updates which UMTRI made in creating GUCVS. It may be useful to compare these two old GM files with the corresponding current GM files in order to pinpoint where the changes reported in the next section relative to these two old files fit relative to current GM files.

While ODIC1.FORT and ODIC3.FORT were just the GMCVS divided into two rather equal files for ease of handling, CIGS and CUTS represent a division of the model into two processors to be run usually in succession. This division was made for two reasons: first, it makes recovering printout on an aborted run easier (sometimes automatic if the model is run under procedure control), and secondly, it reduces the total virtual memory needed to run the model without either reducing model features or increasing the cost of a model run. Part 6 of this report gives a short discussion of model improvements incorporated in GUCVS.

ROBMULTESTS, ROBEP1MULT, and ROBDEL0MULT are three stand alone packages which consist of a test main, a trial version of the new CVS model subprogram, "BLTSOL" (which has the task of obtaining a solution to the belt slip equilibrium equations, see Section 4.2.3), and needed service routines either pulled from the model intact or dummied. Each of these packages use the same data formats to describe the desired test cases. ROBMULTEST.D is the data for the standard group of tests which was used to test these three approaches.

ROBEP1MULT is the approach discussed in Section 4.2.5 . This package contains one coding error which invalidates the results for the N=4 cases. Since it was clear from the N=2 and N=3 cases that the approach had failed, there was no purpose in finding and correcting this error.

ROBDEL0MULT and ROBMULTESTS are the two approaches discussed in Section 4.6 of Part I. There are no known coding errors in these two packages.

RIGS and RUTS contain replacement and new subprograms for GUCVS to implement the new belt slip algorithms. The subprograms in CIGS for which changes were required (and for which changed versions are included in RIGS) are BELTRT, CINPUT, CONTCT, OUTREC, OUTPUT, SINPUT, and UPDATE. Table C.2 contains a short description of the functions of each of the new subprograms included in RIGS. The subprograms in CUTS for which changes were required (and for which changed versions are included in RUTS) are BLOCK DATA, HEDING, and PRTLIN. All these routines are only partially tested and are subject to further changes.

ROBPLOTCS.S is the source for the Ellipsoid Man Plotting Program which UMTRI used instead of the VIEW Plotting Program. GUCVS will optionally produce the exterior binary hold files required by both/either/neither of these plotting programs. ROBPLOTCS.S makes use of the regular Calcomp Plotting Routines whereas VIEW makes use of the DISSPLA plotting routines.

TABLE C.2

NEW BELT SLIP ALGORITHM ROUTINES

Routine Name	Description of Function
BLDERV	Obtains approximations to belt material derivatives for one specified belt loop and one specified unstrained belt length configuration of four possible unstrained belt length configurations.
BLEVL	Evaluates belt tension, normal forces, and slip forces for all slip points in all belt loops for one specified unstrained belt length configuration.
BLTSLP	Is the main control routine for the belt slip algorithm. It calls the other belt slip routines as necessary to handle for each belt loop separately the applicable cases of non-slipping belts, free-slipping belts under slack, or slipping under friction. The last case is handled by setting up and solving up to four belt configurations in order to determine the minimum amount of belt slip which satisfies the belt slip equilibrium equations.
BLTSOL	Solves for unstrained belt lengths which satisfy the belt slip equilibrium equations for a specified belt loop and a specified belt configuration.
BLTUPD	Updates the unstrained belt lengths for which values at current and last previous simulated times are maintained.
BLTVAL	Evaluates belt tension forces for a specified regular belt or around a specified belt slip point.
POLRT	Computes real and complex roots of a real polynomial by the Newton-Raphson iterative technique.
SLPINP	Reads the F.9 input data cards and sets up the appropriate control and data internal storage arrays.

C.4 Installation Notes

This section documents the updates necessary to turn the GMCVS presented on tape BSC023 into the GUCVS also presented on this tape. Table C.3 is the index of the collected output of several runs of a comparison program on the MTS System of the files CVSA and CVSB (which are MTS line file versions of tape files ODIC1.FORT and ODIC3.FORT, respectively) and the files CIGS and CUTS (which are MTS line file versions of tape files CIGS and CUTS, respectively). The collected output is included as Table C.5. In all cases the sequence fields have been stripped off to enable the comparisons to be based solely on program content.

Line files are characterized by line numbers which identify each line of information separately from line contents. This allows sequencing to be maintained without disturbing comparisons of content. Line numbers in the file listings which accompany tape BSC023 correspond to the number of the record in the tape file in each case. These tape file listings show the line number for each tape record on the left followed by the 80 characters of contents which includes the sequencing information.

Table C.3 also shows the line number for each displayed line at the left. The output of the comparison program shows lines from the two files being compared side by side on as many printout lines as required to display the total contents of both lines. Every pair of lines which are the same character by character and which are displayed are designated by "=" on the extreme left and right of each printout line required to display the entire lines. The only equal lines which are displayed are those which begin the interval of comparison, end the interval of comparison, precede an unequal line, or follow an unequal line. The equal lines which are not displayed are counted in a summary of omitted lines which is printed in the place of the omitted equal lines. Every line in either file for which a difference is found or for which no corresponding line is found is displayed without the "=" on the extreme left and right of each printout line.

The output of the several comparison runs are ordered in Table C.3 on the sequence of the line intervals in CVSA and CVSB. Any line intervals which are missing in Table C.3 are either exactly the same, or bear no relationship to the other file. A list of the sections of the code which have been changed are included as Table C.4. The changed listings are included as Table C.6.

TABLE C.3

COMPARISON OF GUCVS WITH GMCVS

GUCVS Lines	GMCVS Lines
CVSA(1,249)	CIGS(1,246)
CVSA(250,856)	CIGS(247,856)
CVSA(861,1778)	CIGS(857,1781)
CVSA(1779,3506)	CIGS(1782,3508)
CVSA(3507,3781)	CIGS(3509,3782)
CVSA(5235,6258)	CIGS(5257,6304)
CVSA(6420,6824)	CUTS(835,1122)
CVSB(166,1501)	CIGS(6527,7862)
CVSB(1502,1830)	CIGS(7863,8445)
CVSB(2564,2877)	CIGS(9161,9474)
CVSB(2918,4698)	CIGS(9544,11350)

TABLE C.4

COMPLETELY NEW OR EXTENSIVELY CHANGED CODE

1.	Subroutine DRCQUA	(new code)
2.	Subroutine OUTREC	(new code)
3.	Subroutine PLOTR1,PLOTR2, PLOTRA	(new code)
4.	Subroutine QUAT	(new code)
5.	Subroutine STASH	(new code)
6.	Main program for CUTS replaces old subroutine POSTPR	(new code)
7.	Old subroutine POSTPR (CVSB) replaced by CUTS Main	
8.	Subroutine FETCH	(new code)
9.	New dummied subroutine HICCSI	
10.	Old subroutine HICCSI	
11.	Subroutine INREAD	(new code)
12.	Subroutines PICKUP and PRTLIN	(new code)
13.	New dummied subroutine SLPLOT	
14.	Old subroutine SLPLOT	
15.	Subroutines VECADD and VECSUB	(new code)

```
Unit 0: CVSA(1,249)                                                   Unit 1: SUSP:CIGS(1,246)
=    1          BLOCK DATA                                              1          BLOCK DATA                                   =
==================================================== 28 equal lines ====================================================
=   30          *              PRJNT(7,100),NPANEL(5),NPSF,NBS         30          *              PRJNT(7,100),NPANEL(5),NPSF,NBS =
=         F,NSSF,NBGSF                                                        F,NSSF,NBGSF                                     =
    31          COMMON/RSAVE/ XSG(3,20,3),DPMI(3,3,100),LPMI(1         31          COMMON/RSAVE/ DPMI(3,3,100),LPMI(100)
          00),NSG(7),
    32          *MSG(20,7)
=   33          COMMON/CDINT/  UU(4),GH(3,4),                          32          COMMON/CDINT/  UU(4),GH(3,4),                    =
=   34          *              E(3,800),FF(5,800),GG(5,800),Y(         33          *              E(3,800),FF(5,800),GG(5,800),Y( =
=         5,800),U(5,800),                                                    5,800),U(5,800),                                 =
    35          *              H,HPRINT,TSAVE,TPRINT,TSTART,IC         34          *              H,HPRINT,TSAVE,TPRINT,TSTART,IC
          NT,IDBL,IFLAG,                                                      NT,IDBL,IFLAG
    36          *              DMMY(200,25),DMMY2(5000,25)
=   37          COMMON/DAMPER/ APSDM(3,20),APSDN(3,20),ASD(5,2         35          COMMON/DAMPER/ APSDM(3,20),APSDN(3,20),ASD(5,2 =
=         0),MSDM(20),MSDN(20)                                                0),MSDM(20),MSDN(20)                             =
==================================================== 60 equal lines ====================================================
=   98          *              ISTEP,NSTEPS,NDINT,NEQ,IRSIN,IR         96          *              ISTEP,NSTEPS,NDINT,NEQ,IRSIN,IR =
=         SOUT                                                                SOUT                                             =
    99          ·LOGICAL NPRT1,NPRT2,NPRT3                             97          LOGICAL NPRT1,NPRT2,NPRT3 ,NPRT18
=  100          CALL ELTIME(1, 1)                                      98          CALL ELTIME(1, 1)                               =
==================================================== 8 equal lines =====================================================
=  109          1'****************************************')          107          1'****************************************')    =
   110        111 FORMAT(3(22X,'******',70X,'******',/),22X,'***      108        111 FORMAT(3(22X,'******',73X,'******',/),22X,'***
          ***',9X,                                                            ***',9X,
   111          2'GGGGGGG',6X,'MM    MM',6X,'CCCCCCC',6X,'VV    V     109          2'GGGGGGG',6X,'UU    UU',6X,'CCCCCCC',6X,'VV    V
          V',6X,'SSSSSSS',5X,                                                 V',6X,'SSSSSSS',5X,
=  112          2'******',/,22X,'******',9X,                          110          2'******',/,22X,'******',9X,                    =
   113          3'GG     G',6X,'MM    MM',6X,'CCCCCCC',6X,'VV    V    111          3'GG     G',6X,'UU    UU',6X,'CCCCCCC',6X,'VV    V
          V',6X,'SSSSSSS',5X,                                                 V',6X,'SSSSSSS',5X,
=  114          3'******',/,22X,'******',9X,                          112          3'******',/,22X,'******',9X,                    =
   115          4'GG      ',6X,'MMM MMM',6X,'CC      ',6X,' VV VV    113          4'GG      ',6X,'UU    UU',6X,'CC      ',6X,' VV VV
           ',6X,'SS     ',5X,                                                  ',6X,'SS     ',5X,
=  116          4'******',/,22X,'******',9X,                          114          4'******',/,22X,'******',9X,                    =
   117          9'GG  GGG',6X,'MMM MMM',6X,'CC      ',6X,' VV VV    115          9'GG  GGG',6X,'UU    UU',6X,'CC      ',6X,' VV VV
           ',6X,'SSSSSSS',5X,                                                  ',6X,'SSSSSSS',5X,
=  118          9'******',/,22X,'******',9X,                          116          9'******',/,22X,'******',9X,                    =
   119          5'GG    GG',6X,'MM M MM',6X,'CC      ',6X,'   VVV    117          5'GG    GG',6X,'UU    UU',6X,'CC      ',6X,'   VVV
           ',6X,'     SS',5X,                                                  ',6X,'     SS',5X,
=  120          5'******',/,22X,'******',9X,                          118          5'******',/,22X,'******',9X,                    =
   121          6'GG    GG',6X,'MM M MM',6X,'CCCCCCC',6X,'   VVV     119          6'GG    GG',6X,'UUUUUUU',6X,'CCCCCCC',6X,'   VVV
           ',6X,'SSSSSSS',5X,                                                  ',6X,'SSSSSSS',5X,
=  122          6'******',/,22X,'******',9X,                          120          6'******',/,22X,'******',9X,                    =
   123          7'GGGGG G',6X,'MM    MM',6X,'CCCCCCC',6X,'    V      121          7'GGGGG G',6X,'UUUUUUU',6X,'CCCCCCC',6X,'    V
           ',6X,'SSSSSSS',5X,                                                  ',6X,'SSSSSSS',5X,
=  124          7'******')                                            122          7'******')                                      =
=  125        112 FORMAT(3(22X,'******',73X,'******',/),22X,'***      123        112 FORMAT(3(22X,'******',73X,'******',/),22X,'*** =
=         ***',17X,                                                           ***',17X,                                        =
   126          8'          G. M. R.',8X,3A4,17X,'******',/,22      124          8' GMR/UMTRI MAIN PROCESSOR',2X,3A4,17X,'******
          X,                                                                  ',/,22X,
=  127          *'******',73X,'******',/                              125          *'******',73X,'******',/                        =
==================================================== 30 equal lines ====================================================
=  158        17 FORMAT('0 NPRT ARRAY'/3X,36I3/3X,36I3)               156        17 FORMAT('0 NPRT ARRAY'/3X,36I3/3X,36I3)       =
   159          NPRT4 = NPRT(4)
   160          IF  (NPRT(4).LT.0)  GO TO 50
=  161    C                                                           157    C                                                 =
==================================================== 21 equal lines ====================================================
=  183        18  WRITE (6,999)                                       179        18  WRITE (6,999)                               =
   184       999 FORMAT(1X,'****   RESTART OPTION NOT OPERETION       180       999 FORMAT(1X,'****   RESTART OPTION NOT OPERATION
```

Table C.5. Comparison of GUCVS with GMCVS (Page 1 of 41)

```
          AL ****')                                                  AL ****')
=   185        RETURN                                        181          RETURN                                     =
    186        CALL RSTART(1,IRSIN)                          182     C  **********************
    187        CALL RSTART(4,5)                              183     C    REGULAR RESTART CODE IS COMMENTED OUT BELOW.
    188        NPRT4 = NPRT(4)                               184     C
                                                             185     C        CALL RSTART(1,IRSIN)
                                                             186     C        CALL RSTART(4,5)
=   189      19 IF (IRSOUT.NE.O) CALL RSTART(2,IRSOUT)       187        19 IF (IRSOUT.NE.O) CALL RSTART(2,IRSOUT)        =
==================================================   11 equal lines ====================================================
=   201      21 CALL RSTART(4,5)                             199        21 CALL RSTART(4,5)                              =
    202         IF  (NPRT(4).LT.O)  GO TO 50
=   203      23 CALL DINT                                    200        23 CALL DINT                                     =
==================================================   21 equal lines ====================================================
=   225     C                                                222     C                                                   =
    226     C      5. SUBROUTINE ELTIME ON PRIMARY OUTPUT UNIT CO 223 C      5.  PROGRAM PLOTCS PLOT DATA ON UNIT 9 CONTROL
          NTROLLED BY NPRT(2).                                       LED BY NPRT(18).
                                                             224     C
                                                             225          NPRT18 = (NPRT(18) .EQ. 1)
                                                             226          IF (NPRT(18) .GT. 1) NPRT18 = (MOD(ISTEP,NPRT(
                                                                     18)) .EQ. O)
                                                             227          IF (NPRT(18) .NE. O .AND. ISTEP .EQ. O) CALL P
                                                                     LOTR1
                                                             228          IF (NPRT18) CALL PLOTR2
                                                             229     C
                                                             230     C      6. SUBROUTINE ELTIME ON PRIMARY OUTPUT UNIT CO
                                                                     NTROLLED BY NPRT(2).
=   227     C                                                231     C                                                   =
==================================================    7 equal lines ====================================================
=   235            IF (ISTEP.LE.NSTEPS) GO TO 20             239          IF (ISTEP.LE.NSTEPS) GO TO 20                  =
    236     C                                                240      50  IF  (NPRT2)  CALL ELTIME (2,1)
    237     C      6. SUBROUTINE POSTPR ON PRIMARY OUTPUT UNIT CO
          NTROLLED BY NPRT(4).
    238     C
    239       50 IF  (NPRT4.GT.O)  END FILE 8
    240          IF  (NPRT(4).EQ.O .OR. NPRT(4).EQ.4)  GO TO 60
    241          PRDT = 1000.O*DT
    242          CALL POSTPR (PRDT)
    243          IF  (NPRT2)  CALL ELTIME (2,1)
=   244     C                                                241     C                                                   =
==================================================    1 equal line  ====================================================
=   246     C                                                243     C                                                   =
    247       60 IF  (.NOT.NPRT2)  CALL ELTIME (2,1)         244          IF  (.NOT.NPRT2)  CALL ELTIME (2,1)
=   248          STOP  1                                     245          STOP  1                                        =
=   249          END                                         246          END                                            =
```

Table C.5. Comparison of GUCVS with GMCVS (Page 2 of 41)

```
 Unit 0: CVSA(250,856)                                              Unit 1: SUSP:CIGS(247,856)
=    250          SUBROUTINE ADJUST (M,D1)                         247          SUBROUTINE ADJUST (M,D1)                            =
====================================================   5 equal lines ===========================================================
=    256           *             E(3,800), F(5,800),GG(5,800),Y(5  253           *             E(3,800), F(5,800),GG(5,800),Y(5 =
=          ,800),U(5,800),                                               ,800),U(5,800),                                           =
     257           *             H,HPRINT,HS,TPRINT,TSTART,ICNT,I  254           *             H,HPRINT,HS,TPRINT,TSTART,ICNT,I
           DBL,IFLAG,                                                    DBL,IFLAG
     258           *             DMMY(200,25)
=    259          COMMON/COMAIN/ VAR(800),DER(800),DT,HO,HMAX,HM   255          COMMON/COMAIN/ VAR(800),DER(800),DT,HO,HMAX,HM =
=          IN,RSTIME,                                                    IN,RSTIME,                                                =
=================================================== 362 equal lines ===========================================================
=    622   C                                                       618   C                                                         =
     623          L = NSEG + NBAG + 2                              619          L = NGRND + NBAG
     624          K = NSEG + 2                                     620          K = NGRND
=    625          W(L)       = W(K)                                621          W(L)       = W(K)                                 =
===================================================  16 equal lines ===========================================================
=    642       19 SGTEST(I,4,L) = SGTEST(I,4,K)                    638       19 SGTEST(I,4,L) = SGTEST(I,4,K)                     =
     643          NGRND = NSEG + NBAG + 2                          639          NGRND = NGRND + NBAG
=    644          DO 40 J=1,NBAG                                   640          DO 40 J=1,NBAG                                    =
=================================================== 211 equal lines ===========================================================
=    856       49 PYMOUT(J) = CYMOUT(J)                            852       49 PYMOUT(J) = CYMOUT(J)                             =
                                                                   853       50 CONTINUE
                                                                   854       99 CALL ELTIME(2,29)
                                                                   855          RETURN
                                                                   856          END
```

Table C.5. Comparison of GUCVS with GMCVS (Page 3 of 41)

```
Unit 0: CVSA(861,1778)                                             Unit 1: SUSP:CIGS(857,1781)

=   861           SUBROUTINE BELTG (ZA,ZB,ZC,BD)                      857           SUBROUTINE BELTG (ZA,ZB,ZC,BD)                    =
=======================================================   180 equal lines ===========================================================
=  1042           SUBROUTINE BELTRT(I,II,MM,M,NT)                     1038          SUBROUTINE BELTRT(I,II,MM,M,NT)                   =
   1043     C
            REV 20 05/23/80
=  1044     C     THE ROUTINE CALLS SUBROUTINE BELTG TO COMPUTE   .   1039    C     THE ROUTINE CALLS SUBROUTINE BELTG TO COMPUTE   =
=          THE TANGENT POINTS                                               THE TANGENT POINTS                                      =
=======================================================     6 equal lines ===========================================================
=  1051           *              NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJ      1046          *              NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJ  =
=          NTF,NPRT(36)                                                     NTF,NPRT(36)                                            =
   1052           COMMON/SGMNTS/ D(3,3,100) ,WMEG(3,100),WMEGD(3      1047          COMMON/SGMNTS/ D(3,3,100),WMEG(3,100),WMEGD(3,
           ,100),U1(3,100),                                                 100),U1(3,100),
   1053           *    U2(3,100), SEGLP(3,100),SEGLV(3,100),SEGLA     1048          *  U2(3,100),SEGLP(3,100),SEGLV(3,100),SEGLA(3,
           (3,100),NSYM(100)                                                100),NSYM(100)
=  1054           COMMON/CNTSRF/ PL(17,100),BELT(20,8),TPTS(6,8)      1049          COMMON/CNTSRF/ PL(17,100),BELT(20,8),TPTS(6,8) =
=          ,BD(24,110)                                                      ,BD(24,110)                                             =
=  1055           COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(300)      1050          COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(300) =
=          ,NTAB(3000),                                                     ,NTAB(3000),                                            =
   1056           *TAB(9600)                                          1051          *  TAB(9600)
   1057           COMMON/FORCES/ PSF(7,100),BSF(4,20),SSF(10,20)      1052          COMMON/FORCES/PSF(7,100),BSF(4,20),SSF(10,20),
           ,BAGSF(3,20),                                                    BAGSF(3,20),
=  1058           *              PRJNT(7,100),NPANEL(5),NPSF,NBS      1053          *              PRJNT(7,100),NPANEL(5),NPSF,NBS =
=          F,NSSF,NBGSF                                                     F,NSSF,NBGSF                                            =
=======================================================     9 equal lines ===========================================================
=  1068     C                                                         1063    C                                                       =
   1069           MA = MM                                             1064          MA = MOD(MM,100)
   1070           MB = MM                                             1065          MB = MM/100
                                                                      1066          IF (MB.EQ.0) MB=MA
=  1071           CALL DOT31 (D(1,1,MA),BELT(1,M),TA)                 1067          CALL DOT31 (D(1,1,MA),BELT(1,M),TA)               =
=======================================================    87 equal lines ===========================================================
=  1159           CALL DOT31(D(1,1,I),UVB,TTT)                        1155          CALL DOT31(D(1,1,I),UVB,TTT)                      =
   1160           DO 30 K=1,3                                         1156          DO 100 K=1,3
   1161           TTT(K) = TTT(K)+TT(K)                               1157           U1(K,MA) = U1(K,MA) - TT(K)
   1162        30 U1(K,I) = U1(K,I)+TTT(K)                            1158           U1(K,MB) = U1(K,MB) - TTT(K)
                                                                      1159      100  U1(K,I) = U1(K,I)+TTT(K) + TT(K)
=  1163     C                                                         1160    C                                                       =
=======================================================     1 equal line  ===========================================================
=  1165     C                                                         1162    C                                                       =
                                                                      1163           CALL MAT31(D(1,1,MA),TT,ZA)
                                                                      1164           CALL MAT31(D(1,1,MB),TTT,ZB)
                                                                      1165           CALL CROSS(BELT(1,M),ZA,TA)
                                                                      1166           CALL CROSS(BELT(4,M),ZB,TB)
=  1166           CALL CROSS(APA,UVA,TT)                              1167          CALL CROSS(APA,UVA,TT)                            =
=======================================================     1 equal line  ===========================================================
=  1168           DO 40 K=1,3                                         1169          DO 40 K=1,3                                       =
                                                                      1170           U2(K,MA) = U2(K,MA) - TA(K)
                                                                      1171           U2(K,MB) = U2(K,MB) - TB(K)
=  1169        40 U2(K,I) = U2(K,I)+(TT(K)+TTT(K))                    1172        40 U2(K,I) = U2(K,I)+(TT(K)+TTT(K))                =
=======================================================   197 equal lines ===========================================================
=  1367           *              UNITL,UNITM,UNITT,GRAVTY(3)          1370          *              UNITL,UNITM,UNITT,GRAVTY(3)      =
   1368           COMMON/RSAVE/XSG(3,20,3),DPMI(3,3,100),LPMI(10      1371          COMMON/RSAVE/DPMI(3,3,100),LPMI(100)
           0),NSG(7),MSG(20,7)
=  1369           COMMON/TEMPVS/ YPR1(3,100),YPR2(3,100),YPR3(3,      1372          COMMON/TEMPVS/ YPR1(3,100),YPR2(3,100),YPR3(3, =
=          100),YPRPMI(3,100),                                              100),YPRPMI(3,100),                                     =
=======================================================    66 equal lines ===========================================================
=  1436           *    '         UNITL,UNITM,UNITT,GRAVTY(3)          1439          *              UNITL,UNITM,UNITT,GRAVTY(3)      =
   1437           COMMON/RSAVE/XSG(3,20,3),DPMI(3,3,100),LPMI(10      1440          COMMON/RSAVE/DPMI(3,3,100),LPMI(100)
           0),NSG(7),MSG(20,7)
```

Table C.5. Comparison of GUCVS with GMCVS (Page 4 of 41)

```
=  1438            COMMON/TEMPVS/ YPR1(3,100),YPR2(3,100),YPR3(3,        1441            COMMON/TEMPVS/ YPR1(3,100),YPR2(3,100),YPR3(3,  =
=            100),YPRPMI(3,100),                                                    100),YPRPMI(3,100),                                      =
===========================================================   339 equal lines ================================================================
=  1778            END                                                   1781            END                                                 =
```

Table C.5. Comparison of GUCVS with GMCVS (Page :5 of 41)

```
Unit 0: CVSA(1779,3506)                                           Unit 1: SUSP:CIGS(1782,3508)
=  1779          SUBROUTINE CFACTT(A,B,D)                           1782          SUBROUTINE CFACTT(A,B,D)                            =
=======================================================  196 equal lines =======================================================
=  1976          *            E(3,800), F(5,800),GG(5,800),Y(5,80   1979          *            E(3,800), F(5,800),GG(5,800),Y(5,80 =
=          O),U(5,800),                                                     O),U(5,800),                                              =
   1977          *            H,HPRINT,HS,TPRINT,TSTART,ICNT,IDBL   1980          *            H,HPRINT,HS,TPRINT,TSTART,ICNT,IDBL
           ,IFLAG,                                                          ,IFLAG
   1978          *            DMMY(200,25)
=  1979          COMMON/COMAIN/ VAR(800),DER(800),DT,HO,HMAX,HM     1981          COMMON/COMAIN/ VAR(800),DER(800),DT,HO,HMAX,HM =
=          IN,RSTIME,                                                       IN,RSTIME,                                                =
=======================================================    4 equal lines =======================================================
=  1984          CALL DZP (NEQ,VAR,GG,E,FT,M)                       1986          CALL DZP (NEQ,VAR,GG,E,FT,M)                        =
   1985          IF  (NPRT(26).EQ.2)  CALL OUTPUT(0)                1987          IF  (NPRT(4).EQ.2)  CALL OUTPUT(0)
=  1986          CALL PDAUX (VAR,DER,NEQ,K)                         1988          CALL PDAUX (VAR,DER,NEQ,K)                          =
   1987          IF  (NPRT(26).EQ.2)  CALL OUTPUT(1)                1989          IF  (NPRT(4).EQ.2)  CALL OUTPUT(1)
=  1988          RETURN                                             1990          RETURN                                              =
=======================================================  230 equal lines =======================================================
=  2219          IF (NJ2.GT.200) WRITE (6,11) NS,NFLX,NQ,NJNT,N     2221          IF (NJ2.GT.200) WRITE (6,11) NS,NFLX,NQ,NJNT,N =
=          J2                                                               J2                                                        =
   2220       11 FORMAT('ONS=',I6,',NFLX=',I6,',NQ=',I6,',NJNT=     2222       11 FORMAT('ONS=',I6,',NFLX=',I6,',NQ=',I6,',NJNT=
           ',I6,' AND NJ2='I6/                                              ',I6,' AND NJ2=',I6/
=  2221         *' THE VALUE OF NJ2 EXCEEDS THE ARRAY SIZES FOR     2223         *' THE VALUE OF NJ2 EXCEEDS THE ARRAY SIZES FOR =
=           RHS AND IJK IN SUBR                                              RHS AND IJK IN SUBR                                      =
======================================================= 1284 equal lines =======================================================
=  3506          END                                                3508          END                                                 =
```

Table C.5. Comparison of GUCVS with GMCVS (Page 6 of 41)

```
 Unit 0: CVSA(3507,3781)                                      Unit 1: SUSP:CIGS(3509,3782)
=  3507          SUBROUTINE DHHPIN(DD,BN,L,M,N)               3509          SUBROUTINE DHHPIN(DD,BN,L,M,N)               =
=================================================== 43 equal lines ===================================================
=  3551         *          E(3,800), F(5,800),GG(5,800),Y(5,80 3553         *          E(3,800), F(5,800),GG(5,800),Y(5,80 =
=        0),U(5,800),                                                0),U(5,800),                                       =
   3552         *          H,HPRINT,HS,TPRINT,TSTART,ICNT,IDBL 3554         *          H,HPRINT,HS,TPRINT,TSTART,ICNT,IDBL
         ,IFLAG,                                                     ,IFLAG
   3553         *          DMMY(200,25)
=  3554          COMMON/COMAIN/ VAR(800),DER(800),DT,HO,HMAX,HM 3555          COMMON/COMAIN/ VAR(800),DER(800),DT,HO,HMAX,HM =
=        IN,RSTIME,                                                  IN,RSTIME,                                         =
=================================================== 33 equal lines ===================================================
=  3588          ICNT = -2                                    3589          ICNT = -2                                    =
   3589          IF  (ISTEP.EQ.O  .OR. NPRT(26).EQ.2)  CALL OUT 3590          IF  (ISTEP.EQ.O  .OR. NPRT(4).EQ.2)  CALL OUTP
         PUT(O)                                                      UT(O)
=  3590          CALL PDAUX (VAR,DER,NEQ,K)                   3591          CALL PDAUX (VAR,DER,NEQ,K)                   =
   3591          IF  (ISTEP.NE.O .AND. NPRT(26).EQ.2)  CALL OUT 3592          IF  (ISTEP.NE.O .AND. NPRT(4).EQ.2)  CALL OUTP
         PUT(1)                                                      UT(1)
=  3592          DO  14  I=1,NEQ                              3593          DO  14  I=1,NEQ                              =
=================================================== 37 equal lines ===================================================
=  3630          IF  (L.EQ.1)  M = O                          3631          IF  (L.EQ.1)  M = O                          =
   3631          IF  (NPRT(26).NE.2)  CALL OUTPUT(O)          3632          IF  (NPRT(4).NE.2)  CALL OUTPUT(O)
=  3632          CALL CMPUTE (K,M, H)                         3633          CALL CMPUTE (K,M, H)                         =
=================================================== 46 equal lines ===================================================
=  3679      58  IF  (H.LE.HMIN+EPS(8))  GO TO 61             3680      58  IF  (H.LE.HMIN+EPS(8))  GO TO 61             =
   3680          IF  (NPRT(26).EQ.2)  CALL OUTPUT(1)          3681          IF  (NPRT(4).EQ.2)  CALL OUTPUT(1)
=  3681      59  TIME = TSTART                                3682      59  TIME = TSTART                                =
=================================================== 99 equal lines ===================================================
=  3781          END                                          3782          END                                          =
```

Table C.5. Comparison of GUCVS with GMCVS (Page 7 of 41)

```
Unit 0: CVSA(5235,6[illegible])                                 Unit 1: SUSP:CIGS(5257,6304)
= 5235          SUBROUTINE FINPUT                               5257          SUBROUTINE FINPUT                                   =
=====================================================  29 equal lines =====================================================
= 5265        *               IWIND(100),MWSEG(5,100),NFVSEG(6),  5287        *               IWIND(100),MWSEG(5,100),NFVSEG(6),  =
=        NFVNT(5)                                                    NFVNT(5)                                                 =
                                                                5288          COMMON /PLOTRS/ ELYPR(3,100), PLPTS(3,3,100),
                                                                     NELSEG(100), LDNPCS
                                                                5289         1   , IELPS, IELPB, IELP, NNELP, NPLSEG(100)
                                                                5290          REAL*4 PLPTS
= 5266          COMMON/TEMPVS/JTITLE(5,301),NF(5),MS(3),KTITLE  5291          COMMON/TEMPVS/JTITLE(5,301),NF(5),MS(3),KTITLE      =
=        (31)                                                        (31)                                                     =
=====================================================  77 equal lines =====================================================
= 5344       41 FORMAT('O',I7,'-',I3,I11,'-',I3,I8,4I21)        5369       41 FORMAT('O',I7,'-',I3,I11,'-',I3,I8,4I21)        =
  5345          IF (NJ.NE.J) WRITE (6,42)                       5370          IF (NJ.NE.J) GO TO 426
                                                                5371          IF (NJ .GT. 100) GO TO 426
                                                                5372          IF (I .NE. 1) GO TO 422
                                                                5373          NN = IABS(MS(1))
                                                                5374          IF (NN .GT. 100) GO TO 426
                                                                5375          IF (NJ .GT. NPL) GO TO 426
                                                                5376          IF (NPLSEG(NJ) .EQ. 0) NPLSEG(NJ) = NN
                                                                5377          IF (NPLSEG(NJ) .NE. NN) GO TO 426
                                                                5378      422 IF (I .GT. 3) GO TO 429
                                                                5379          IF (I .NE. 3) GO TO 424
                                                                5380          NN = IABS(MS(1))
                                                                5381          IF (NN .GT. 100) GO TO 426
                                                                5382          IF (NN .GT. NSEG .AND. NN .LT. NGRND) GO TO 42
                                                                     6
                                                                5383          IF (NELSEG(NN) .EQ. 0) NELSEG(NN) = NJ
                                                                5384          IF (NELSEG(NN) .NE. NJ) GO TO 426
                                                                5385      424 MM = IABS(MS(3))
                                                                5386          IF (MM .GT. 100) GO TO 426
                                                                5387          M = IABS(MS(2))
                                                                5388          IF (MM .GT. NSEG .AND. MM .LT. NGRND) GO TO 42
                                                                     6
                                                                5389          IF (NELSEG(MM) .EQ. 0) NELSEG(MM) = M
                                                                5390          IF (NELSEG(MM) .EQ. M) GO TO 429
                                                                5391      426 WRITE (6,42)
= 5346       42 FORMAT(' CONTACT INPUT ERROR. PROGRAM TERMINAT  5392       42 FORMAT(' CONTACT INPUT ERROR. PROGRAM TERMINAT  =
=        ED.')                                                       ED.')                                                    =
  5347          IF  (NJ.NE.J)  STOP 14                          5393          STOP 14
  5348          NLT = 1                                         5394      429 NLT = 1
= 5349          DO 43 JJ = 1,31                                 5395          DO 43 JJ = 1,31                                     =
===================================================== 908 equal lines =====================================================
= 6258          END                                             6304          END                                                 =
```

Table C.5. Comparison of GUCVS with GMCVS (Page 8 of 41)

Unit 0: CVSA(6420,6824)

```
  6420         SUBROUTINE HEDING (LINES,LPP)
= 6421   C
=             REV 20 05/18/80
  6422         IMPLICIT REAL*8 (A-H,O-Z)
  6423         COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NV
       EH,NGRND,
  6424         *             NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF
       ,NPRT(36)
  6425         COMMON/JBARTZ/ MNPL( 100),MNBLT( 8),MNSEG( 100
       ),MNBAG(  6),
  6426         *             MPL(3,5,100),MBLT(3,5,8),MSEG(3,5,
       100),MBAG(3,10,6),
  6427         *             NTPL( 5,100),NTBLT( 5,8),NTSEG( 5,
       100)
  6428         COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),B
       DYTTL(5),
  6429         *             BLTTTL(5,8),PLTTL(5,100),BAGTTL(5,
       6),SEG(100),
  6430         *             JOINT(100),CGS(100),JS(100)
  6431         REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BA
       GTTL,SEG,JOINT
  6432         LOGICAL*1 CGS,JS
  6433         COMMON/FORCES/ PSF(7,100),BSF(4,20),SSF(10,20)
       ,BAGSF(3,20),
  6434         *             PRJNT(7,100),NPANEL(5),NPSF,NBSF,N
       SSF,NBGSF
  6435         COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),
  6436         *             UNITL,UNITM,UNITT,GRAVTY(3)
  6437         COMMON/RSAVE/XSG(3,20,3),DPMI(3,3,100),LPMI(10
       0),NSG(7),MSG(20,7)
  6438         COMMON/DAMPER/ APSDM(3,20),APSDN(3,20),ASD(5,2
       0),MSDM(20),MSDN(20)
  6439         COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),
       PLOSS(2,100),
  6440         *             XLONG(20),HTIME(2),IBAR(5,100),NL(
       2,100),
  6441         *             NPTSPB(20),NPTPLY(20),NTHRNS(20),N
       BLTPH(5)
  6442   C     NOTE: SUBROUTINES POSTPR & HEDING SHARE THIS C
       OMMON/TEMPVS/.
  6443   C     SEE COMMENT IN POSTPR ABOUT FIRST DIMENSION OF
        PLDATA.
  6444         REAL    HEAD , PHED , BLANK , PLDATA , USEC , Z
       TTH
  6445         COMMON/TEMPVS/TDATA(14,300),HEAD(20),NOPL(100)
       ,MOPL(100),
  6446         *             PLDATA(97,20) , USEC(45) , ZTTH(14
       ,45,2)
  6447         LOGICAL   LOLD , LNEW
  6448         DIMENSION PHED(5),HEDJ(4,2),HEADJJ(4,2)
  6449         DATA HEDJ/8HIPIN  FL,8HEXURE  A,8HZIMUTH  ,8HT
       ORSION ,
  6450         *       8HIEULER  ,8HPREC.  N,8HUTATION ,8H  SP
       IN  /
```

Unit 1: SUSP:CUTS(835,1122)

```
  835         SUBROUTINE HEDING (KTG, LIN, LPAG)
  836   C                                                      =
             REV 20 05/18/80                                  =
  837         COMMON /HOLDIT/ BAGTTL(5,6), BDYTTL(5), BLTTTL
      (5,8), COMENT(40),
  838        1DATE(3), DT, JOINT(100), INCSML, INCBIG, PLTTL
      (5,100),
  839        2SEG(100), VPSTTL(20), LLFRST, KASTOP, NRNTIM,
  840        3TIMLAS, MULTPL, NPL, NPLT, NRNVAR, LFENKT,
  841        4NBAG, NBLT, LBBAG, LBBELT, LBJNT, LBPL,
  842        5LBSEG, KKNTRL(241), NSD, NSEG, ISPSWT,
  843        6NSTEPS, NVEH, NGRND, NHRNSS, NJNT
  844         COMMON /PRESET/ ISYMB(49), KASLIN(3,16), LDNAR
      Y(3), LPP, NW60
  845        1, MAXKAP
  846         COMMON /FETCHT/ D(3,3,12), DPMI(3,3,100), G, L
      PMI(100), PI,
  847        1SEGLA(3,12), SEGLP(3,12), SEGLV(3,12), WMEG(3,
      12),
  848        2WMEGD(3,12), UNITL, UNITT, UNITM, RADIAN, TIMM
      AX,
  849        3IKASE(12), NKASE, KASKOL, KASPAG, KASFUL
  850         COMMON /HLDCON/ NRNBAS(246), LFIRST(241), KASN
      UM, KATKOL(12),
  851        1KPRELM(3), KATKAS(3,12), KASKTL(5,241), MAXPTS
      ,
  852        2NRN, LSTEP, LASREC(3)
  853         COMMON /IOCNTL/ LDNWRK, KWDPLN, IOTALK
  854         COMMON /FETCHU/ PQUANT(16), KONTLP(6,3), JDTPT
      S(18), IPIN(3),
  855        1KKAS, MSECPT, MTIMPT
  856         COMMON /ZTTHNG/ TIME(6000), ZTTH(6000,11), LIN
      EQT(6000)
  857         DIMENSION CONTLP(6,3), HHEAD(6), HEAD(2,3)
  858         EQUIVALENCE (CONTLP(1,1),KONTLP(1,1)), (HHEAD(
      1), HEAD(1,1))
  859        -DIMENSION PHED(5),HEDJ(8,2),HEADJJ(8,2) , WRTO
      PT(2,2), WRT(2,3)
  860         DATA HEDJ/'IPIN', 4H  FL, 4HEXUR, 4HE  A, 4HZI
      MU, 4HTH   ,
  861        14HTORS, 4HION , 4HIEUL, 4HER   ,'PREC', 4H.  N,
  862        24HUTAT, 4HION , 4H  SP, 4HIN  /
  863         DATA BLANK/'    '/ , PAGEB/O./
  864         DATA PHED/'SPRF','PNL1','PNL2','PNL3','PNL4'/
  865         DATA WRTOPT/4H"W.R, 4H.T.", 4H(W.R, 4H.T.)/
```

Table C.5. Comparison of GUCVS with GMCVS (Page 9 of 41)

```
  6451        DATA BLANK/4H    /
  6452        DATA PHED/4HSPRF,4HPNL1,4HPNL2,4HPNL3,4HPNL4/
  6453        NPRT4 = NPRT(4) + 4
  6454        IF  (NPRT4.LE.O .OR. NPRT4.GT.8)  STOP 40
  6455        GO TO  (11,11,82,12,12,11,11,12) , NPRT4
  6456     11 LOLD = .FALSE.
  6457        LNEW = .TRUE.
  6458        GO TO 13
  6459     12 LOLD = .TRUE.
  6460        LNEW = .FALSE.
  6461     13 MT = 20
  6462        NLINES = MOD(LINES-1,LPP)+1
  6463        XPAGE = 0.01*FLOAT((LINES + LPP-1)/LPP)
  6464  C
  6465  C     NOTE: MT WILL BE THE PAGE OR OUTPUT UNIT COUNT
ER
  6466  C           NT WILL BE THE ACTUAL OUTPUT UNIT NUMBER

  6467  C           IT WILL BE THE INDEX TO THE DATA ARRAY
  6468  C           NLINES WILL BE THE NUMBER OF LINES TO BE
 PRINTED
  6469  C
  6470  C
  6471  C     EVERY LPP LINES PRINT HEADINGS FOR 7 TYPES OF
OUTPUT ABOVE.
  6472  C
  6473        DO 20 K=1,7
  6474        IF (NSG(K).LE.O) GO TO 20
  6475        KSG [illegible]NSG(K)
  6476        J3 [illegible]
  6477        IF ([illegible]Q.7) J3 = 2
  6478        DO 19 J1=1,KSG,J3
  6479        MT = MT + 1
  6480        NT = MT
  6481        IF  (LNEW)  NT = 6

  6482        IT = MT - 20
  6483        PAGE = FLOAT(MT) + XPAGE
  6484        WRITE (NT,21) DATE,PAGE,COMENT,VPSTTL,BDYTTL
  6485        IF (K.EQ.1) WRITE (NT,22)
  6486        IF (K.EQ.2) WRITE (NT,23) UNITL,UNITT
  6487        IF (K.EQ.3) WRITE (NT,24) UNITL
  6488        IF (K.EQ.4) WRITE (NT,25) UNITT
  6489        IF (K.EQ.5) WRITE (NT,26) UNITT
  6490        IF (K.EQ.6) WRITE (NT,27)
  6491        IF (K.EQ.7) WRITE (NT,28)
  6492        J2 = MINO(J1+J3-1,KSG)
  6493        DO 14 J=J1,J2
  6494        KK = MSG(J,K)
  6495        HEAD(J) = SEG(KK)
  6496        IF (K.LT.7) GO TO 14




= 6497        KK = IABS(KK)
  6498        HEAD(J) = JOINT(KK)
  6499        JJ2 = J-J1+1
= 6500        K2 = 1
```

```
 866    C      CURRENT CATEGORY NUMBER
 867           KAT = KTG
 868           IF (KAT .LE. O) GO TO 550
 869           IF (KAT .GT. 14) GO TO 550
 870    C         INITIZLIZE TIME POINT ON PAGE LINE COUNT
 871           LPAG = O
 872    C          FIRST TIME POINT FOR NEW PAGE
 873           LINES = LIN
 874           NPAG = MAXO((LINES + LPP - 2) / (LPP - 1), 1)
 875           IF (LINES .EQ. 1) PAGEB = PAGEB + 1
 876           PAGE = PAGEB + FLOAT(NPAG) / 100.
 877           NT = 6
 878           IF (KAT .GT. 7) GO TO 210
 879           K = KAT
 880           WRITE (NT,10) DATE,PAGE,COMENT,VPSTTL,BDYTTL

 881        10 FORMAT('1',18X,'DATE:',3X,3A4,68X,'PAGE:',F6.2
/
 882          18X,'RUN DESCRIPTION:',3X,20A4/27X,20A4/
 883          23X,'VEHICLE DECELERATION:',3X,20A4/

 884          311X,'CRASH VICTIM:',3X,5A4 )
 885           IF (K.EQ.1) WRITE (NT,20)
 886        20 FORMAT('O', 27X,

 887          1'SEGMENT LINEAR ACCELERATIONS (G''S)')
 888           IF (K.EQ.2) WRITE (NT,30) UNITL,UNITT
 889        30 FORMAT('O', 27X,
 890          1'SEGMENT LINEAR VELOCITIES (',A4,'/',A4,')')
 891           IF (K.EQ.3) WRITE (NT,40) UNITL
 892        40 FORMAT('O', 27X,
 893          1'SEGMENT LINEAR DISPLACEMENTS (',A4,')')
 894           IF (K.EQ.4) WRITE (NT,50) UNITT
 895        50 FORMAT('O'/'O', 27X,
 896          1'SEGMENT ANGULAR ACCELERATIONS (REV/',A4,'**2)
')

 897           IF (K.EQ.5) WRITE (NT,60) UNITT
 898        60 FORMAT('O'/'O', 27X,
 899          1'SEGMENT ANGULAR VELOCITIES (REV/',A4,')')
 900           IF (K.EQ.6) WRITE (NT,70)
 901        70 FORMAT('O'/'O', 27X,
 902          1'SEGMENT ANGULAR DISPLACEMENTS (DEG)')
 903           IF (K.EQ.7) WRITE (NT,80)
 904        80 FORMAT('O', 27X,'JOINT PARAMETERS')
 905           DO 110 L = 1, KKAS
 906           KK = IABS(KONTLP(4, L))
 907           HEAD(1,L) = SEG(KK)
 908           JJ = KONTLP(5, L)
 909           HEAD(2,L) = SEG(JJ)
 910           IF (K .GT. 3) GO TO 90
 911           LL = 1
 912           IF (KONTLP(4,L) .LT. O) LL = 2
 913           WRT(1,L) = WRTOPT(1,LL)
 914           WRT(2,L) = WRTOPT(2,LL)
 915           GO TO 110
 916        90 IF (K.LT.7) GO TO 110
 917           KK = IABS(KK)                                    =
 918           HEAD(1,L) = JOINT(KK)

 919           K2 = 1                                           =
```

Table C.5. Comparison of GUCVS with GMCVS (Page 10 of 41)

```
    IF (MSG(J,K).LT.O) K2 = 2
    DO  35  K1=1,4
 35 HEADJJ(K1,JJ2) = HEDJ(K1,K2)
 14 CONTINUE
    IF (K.LE.3) WRITE (NT,29) (BLANK,(XSG(I,J,K),I
=1,3),J=J1,J2)
    IF (K.LE.6) WRITE (NT,30) (BLANK,MSG(J,K),HEAD
(J),J=J1,J2)
    IF (K.LE.5) WRITE (NT,31) (BLANK,J=J1,J2)

    IF (K.EQ.6) WRITE (NT,32) (BLANK,J=J1,J2)

    IF (K.LT.7) GO TO 15

    WRITE (NT,33) (BLANK,MSG(J,K),HEAD(J),J=J1,J2)

    WRITE (NT,36) (BLANK,UNITL,UNITM,J=J1,J2)
    WRITE (NT,37) (BLANK,(HEADJJ(K1,J),K1=1,4),J=1
,JJ2)
 15 WRITE (NT,38)

    IF  (.NOT.LNEW)  GO TO 19
    IF  (K.EQ.7)  GO TO 17

    JJ = 4*(J2-J1+1)
    DO  16  I=1,NLINES

 16 WRITE (NT,39)  USEC(I),(ZTTH(J,I,IT),J=1,JJ)

    GO TO 19
 17 JJ = 7*(J2-J1+1)

    DO  18  I=1,NLINES
 18 WRITE (NT,40)  USEC(I),(ZTTH(J,I,IT),J=1,JJ)

 19 CONTINUE

 20 CONTINUE
 21 FORMAT('1',18X,'DATE:',3X,3A4,68X,'PAGE:',F6.2
/
   *    8X,'RUN DESCRIPTION:',3X,20A4/27X,20A4/
   *    3X,'VEHICLE DECELERATION:',3X,20A4/
   *    11X,'CRASH VICTIM:',3X,5A4 )
 22 FORMAT(' '/27X,
   * 'SEGMENT LINEAR ACCELERATIONS (G''S) IN LOCAL
 REFERENCE'/)
 23 FORMAT(' '/27X,
   *'SEGMENT LINEAR VELOCITIES (',A4,'/',A4,') IN
VEHICLE REFERENCE'/)
 24 FORMAT(' '/27X,
   *'SEGMENT LINEAR DISPLACEMENTS (',A4,') IN VEHI
CLE REFERENCE'/)
 25 FORMAT(' '//27X,
   *'SEGMENT ANGULAR ACCELERATIONS (REV/',A4,'**2)
 IN LOCAL REFERENCE'
   */)
 26 FORMAT(' '//27X,
   *'SEGMENT ANGULAR VELOCITIES (REV/',A4,') IN VE
HICLE REFERENCE'/)
 27 FORMAT(' '//27X,
```

Table C.5. Comparison of GUCVS with GMCVS (Page 11 of 41)

```
    IF (IABS(IPIN(L)) .EQ. 4) K2 = 2
    DO  100 K1=1,8
100 HEADJJ(K1,L) = HEDJ(K1,K2)
110 CONTINUE
    IF (K.LE.3) WRITE (NT,120) (BLANK,(CONTLP(I,J)
,I=1,3)
   1,J=1,KKAS)

120 FORMAT(9X,3(A4,3X,'POINT (',F6.2,',',F6.2,',',
F6.2,') ON ') )
    IF (K.LE.6) WRITE (NT,130) (BLANK, KONTLP(4,J)
, HEAD(1,J),
   1  (WRT(K,J),K= 1, 2), KONTLP(5,J), HEAD(2,J),
J =1, KKAS)
130 FORMAT('O    TIME ',3(A2,'SEGMENT NO.',I3,'(',
 A4,  ')', 2A4,
   1  I3, '(', A4, ')' ) )
    IF (K.LE.5) WRITE (NT,140) (BLANK,J=1,KKAS)

140 FORMAT('O   (MSEC)',3(A4,5X,'X',8X,'Y',8X,'Z',
7X,'RES',1X)  )
    IF (K.EQ.6) WRITE (NT,150) (BLANK,J=1,KKAS)
150 FORMAT('O   (MSEC)',3(A4,4X,'YAW',5X,'PITCH',5
X,'ROLL',5X,'RES '))
    IF (K.LT.7) GO TO 190
    WRITE (NT,160) (BLANK,KONTLP(4,J),HEAD(1,J),J=
1,KKAS)
160 FORMAT(9X,2(A1,21X,'JOINT NO.',I3,' - ',A4,20X
)  )
    WRITE (NT,170) (BLANK,UNITL,UNITM,J=1,KKAS)
170 FORMAT('O    TIME ',2(A1,'STATE',5X,'JOINT ANG
LES (DEG)',8X,
   1'TOTAL TORQUE (',2A4,') ') )
    WRITE (NT,180) (BLANK,(HEADJJ(K1,J),K1=1,8),J=
1,L)
180 FORMAT('O   (MSEC)',2(A1,8A4,4X,'SPRING  VISCO
US    RES. ') )
190 WRITE (NT,200)
200 FORMAT(1X)

    GO TO 550
```

```
  6541        * 'SEGMENT ANGULAR DISPLACEMENTS (DEG) IN VEHIC
           LE REFERENCE'/)
  6542       28 FORMAT(' '/27X,'JOINT PARAMETERS'/)
  6543       29 FORMAT(9X,3(A4,3X,'POINT (',F6.2,',',F6.2,',',
           F6.2,') ON ') )
  6544       30 FORMAT('    TIME ',3(A4,9X,'SEGMENT NO.',I3,'
           - ',A4,5X)   )
  6545       31 FORMAT('   (MSEC)',3(A4,5X,'X',8X,'Y',8X,'Z',7
           X,'RES',1X)  )
  6546       32 FORMAT('   (MSEC)',3(A4,4X,'YAW',5X,'PITCH',5X
           ,'ROLL',5X,'RES ') )
  6547       33 FORMAT(9X,2(A1,21X,'JOINT NO.',I3,' - ',A4,20X
           )  )
  6548       36 FORMAT('    TIME ',2(A1,'STATE',5X,'JOINT ANGL
           ES (DEG)',8X,
  6549        *                'TOTAL TORQUE (',2A4,') ') )
  6550       37 FORMAT('   (MSEC)',2(A1,4A8,4X,'SPRING  VISCOU
           S    RES. ') )
  6551       38 FORMAT(1X)
  6552       39 FORMAT(F9.3,3(3X,4F9.3) )
  6553       40 FORMAT(F9.3,2(F5.0,3F9.3,2X,3F9.3))
= 6554     C                                                      946     C                                                  =
==================================================  1 equal line  ================================================================
= 6556     C                                                      948     C                                                  =
  6557          MPSF = 0                                          949       210 NGO = KAT - 7
  6558          IF (NPL.EQ.0) GO TO 52                            950           IF (NGO .LT. 1  .OR.  NGO .GT. 7) GO TO 550
  6559          DO  42  J=1,NPL                                   951           GO TO (220, 430, 280, 340, 380, 450, 470), NGO
  6560          IF (MNPL(J).EQ.0) GO TO 42                        952       220 N1 = KONTLP(2,1)
  6561          KPL = MNPL(J)                                     953           N2 = KONTLP(2,2)
  6562          DO 41 I=1,KPL                                     954           M1 = KONTLP(3,1)
  6563          MPSF = MPSF+1                                     955           M2 = KONTLP(3,2)
  6564          NOPL(MPSF) = J                                    956           M1A = KONTLP(4,1)
  6565       41 MOPL(MPSF) = MPL(2,I,J)                           957           M2A = KONTLP(4,2)
  6566       42 CONTINUE                                          958           WRITE (NT,10) DATE,PAGE,COMENT,VPSTTL,BDYTTL
  6567          IF (MPSF.EQ.0) GO TO 52                           959           WRITE (NT,230)
  6568          DO 44 J1=1,MPSF,2                                 960       230 FORMAT(27X,'CONTACT FORCES - VEHICLE PANELS VS
                                                                          . ELLIPSOIDS')
  6569          J2 = MIN0(J1+1,MPSF)                              961     .     IF  (KKAS .EQ. 1)  WRITE  (NT,240)
  6570          MT = MT + 1                                       962          1BLANK,N1,( PLTTL(I,N1),I=1,5),M1A,SEG(M1)
  6571          NT = MT                                           963       240 FORMAT('0', 8X,2(A4,'  PANEL',I3,' (',5A4,') V
                                                                          S. ELLIPSOID',I3,
  6572          IF  (LNEW)  NT = 6                                964          1' (',A4,') ') )
  6573          IT = MT - 20                                      965           IF  (KKAS .EQ. 2)  WRITE  (NT,240)
  6574          PAGE = FLOAT(MT) + XPAGE                          966          1BLANK,N1,( PLTTL(I,N1),I=1,5),M1A,SEG(M1),
  6575          WRITE (NT,21) DATE,PAGE,COMENT,VPSTTL,BDYTTL      967          2BLANK,N2,( PLTTL(I,N2),I=1,5),M2A,SEG(M2)
  6576          WRITE (NT,45)                                     968           WRITE (NT,250) (BLANK,UNITL,J=1,KKAS)
  6577          N1 = NOPL(J1)                                     969       250 FORMAT(' ',8X,A4,'DEFL-  NORMAL  FRICTION RESU
                                                                          LTANT  CONTACT LOCAT
  6578          N2 = NOPL(J2)                                     970          1ION (',A4,')',A2,'DEFL-  NORMAL  FRICTION RESU
                                                                          LTANT  CONTACT LOCAT
  6579          M1 = MOPL(J1)                                     971          2ION (',A4,')')
  6580          M2 = MOPL(J2)                                     972           WRITE (NT,260) (BLANK,J=1,KKAS)
  6581          IF  (J1.EQ.J2)  WRITE  (NT,46)                    973       260 FORMAT('    TIME',2(A4,'ECTION    FORCE    FORC
                                                                          E    FORCE     (VEHI
  6582        *             BLANK,N1,( PLTTL(I,N1),I=1,5),M1,S    974          1CLE REFERENCE)' ) )
           EG(M1)
  6583          IF  (J1.NE.J2)  WRITE  (NT,46)                    975           WRITE (NT,270) (BLANK,UNITL,UNITM,UNITM,UNITM,
                                                                          J=1,KKAS)
  6584        *             BLANK,N1,( PLTTL(I,N1),I=1,5),M1,S    976       270 FORMAT('   (MSEC)',2(A3,'(',A4,')',2X,'(',A4,'
           EG(M1),                                                        )',4X,'(',A4,')',3X,
```

Table C.5. Comparison of GUCVS with GMCVS (Page 12 of 41)

```
  6585          *              BLANK,N2,( PLTTL(I,N2),I=1,5),M2,S       977          1'(',A4,')     X        Y        Z  ') )
        EG(M2)
  6586          WRITE (NT,47) (BLANK,UNITL,J=J1,J2)                      978           WRITE (NT,200)
  6587          WRITE (NT,48) (BLANK,J=J1,J2)                            979           GO TO 550
  6588          WRITE (NT,49) (BLANK,UNITL,UNITM,UNITM,UNITM,J
        =J1,J2)
  6589          WRITE (NT,38)
  6590          IF  (.NOT.LNEW)  GO TO 44
  6591          JJ = 7*(J2-J1+1)
  6592          DO  43  I=1,NLINES
  6593       43 WRITE (NT,50)  USEC(I),(ZTTH(J,I,IT),J=1,JJ)
  6594       44 CONTINUE
  6595       45 FORMAT(27X,'CONTACT FORCES - VEHICLE PANELS VS
        . SEGMENTS' )
  6596       46 FORMAT(' '/8X,2(A4,'  PANEL',I3,' (',5A4,') VS
        . SEGMENT',I3,
  6597          *       ' (',A4,') ') )
  6598       47 FORMAT(' ',8X,A4,'DEFL-  NORMAL  FRICTION RESU
        LTANT  CONTACT LOCAT
  6599          *ION (',A4,')',A2,'DEFL-  NORMAL  FRICTION RESU
        LTANT  CONTACT LOCAT
  6600          *ION (',A4,')')
  6601       48 FORMAT('    TIME',2(A4,'ECTION   FORCE    FORC
        E    FORCE     (VEHI
  6602          *CLE REFERENCE)' ) )
  6603       49 FORMAT('   (MSEC)',2(A3,'(',A4,')',2X,'(',A4,'
        )',4X,'(',A4,')',3X,
  6604          *     '(',A4,')     X        Y        Z  ') )
  6605       50 FORMAT(F9.3,2(F9.3,3F9.2,3F8.3) )
  6606       51 FORMAT(3X,'(MSEC)',4(A1,9X,'X',8X,'Y',8X,'Z',1
        X))
= 6607  C                                                                980     C                                                        =
=====================================================       1 equal line  =========================================================
= 6609  C                                                                982     C                                                        =
  6610       52 MBSF = O                                                 983       280 N1 = KONTLP(2,1)
  6611          IF  (NBLT.EQ.O)  GO TO 83                                984           N2 = KONTLP(2,2)
  6612          DO [illegible] J=1,NBLT                                  985           M1 = KONTLP(3,1)
  6613          IF [illegible]LT(J).EQ.O) GO TO 54                       986           M2 = KONTLP(3,2)
  6614          MBSF = MBSF+1                                            987           WRITE (NT,10) DATE,PAGE,COMENT,VPSTTL,BDYTTL
  6615          NOPL(MBSF) = J                                           988           WRITE (NT,290)
  6616          MOPL(MBSF) = MBLT(2,1,J)                                 989       290 FORMAT('O',26X,'CONTACT FORCES - BELTS VS. SEG
                                                                               MENTS')
  6617       54 CONTINUE                                                 990           IF  (KKAS .EQ. 1)  WRITE  (NT,300)
  6618          IF  (MBSF.EQ.O)  GO TO 83                                991          1BLANK,N1,(BLTTTL(I,N1),I=1,5),M1,SEG(M1)
  6619          DO 56 J1=1,MBSF,2                                        992       300 FORMAT(' ',7X,2(A4,'   BELT',I3,' (',5A4,') VS
                                                                               . SEGMENT',I3,
  6620          J2 = MINO(J1+1,MBSF)                                     993          1' (',A4,') ') )
  6621          MT = MT + 1                                              994           IF  (KKAS .EQ. 2)  WRITE  (NT,300)
  6622          NT = MT                                                  995          1BLANK,N1,(BLTTTL(I,N1),I=1,5),M1,SEG(M1),
  6623          IF  (LNEW)  NT = 6                                       996          2BLANK,N2,(BLTTTL(I,N2),I=1,5),M2,SEG(M2)
  6624          IT = MT - 20                                             997           WRITE (NT,310) (BLANK,J=1,KKAS)
  6625          PAGE = FLOAT(MT) + XPAGE                                 998       310 FORMAT(' ',2X,2(A4,11X,'ANCHOR  POINT  A',14X,
                                                                               'ANCHOR  POINT  B'))
  6626          WRITE (NT,21) DATE,PAGE,COMENT,VPSTTL,BDYTTL             999           WRITE (NT,320) (BLANK,J=1,KKAS)
  6627          WRITE (NT,57)                                           1000       320 FORMAT(4X,'TIME',2(A4,5X,'STRAIN',7X,'FORCE',1
                                                                               2X,
  6628          N1 = NOPL(J1)                                           1001          1'STRAIN',7X,'FORCE', 3X) )
  6629          N2 = NOPL(J2)                                           1002           WRITE (NT,330) (BLANK,UNITL,UNITL,UNITM,UNITL,
                                                                               UNITL,UNITM,J=1,
  6630          M1 = MOPL(J1)                                           1003          1KKAS)
```

Table C.5. Comparison of GUCVS with GMCVS (Page 13 of 41)

```
  6631         M2 = MOPL(J2)                                               1004        330 FORMAT(3X,'(MSEC)',2(A4,2X,'(',A4,'/',A4,')',4
                                                                                   X,'(',A4,')',9X,
  6632         IF  (J1.EQ.J2)  WRITE  (NT,58)                              1005         1'(',A4,'/',A4,')',4X,'(',A4,')',3X) )
  6633         *             BLANK,N1,(BLTTTL(I,N1),I=1,5),M1,S            1006         WRITE (NT,200)
       EG(M1)
  6634         IF  (J1.NE.J2)  WRITE  (NT,58)                              1007         GO TO 550
  6635         *             BLANK,N1,(BLTTTL(I,N1),I=1,5),M1,S
       EG(M1),
  6636         *             BLANK,N2,(BLTTTL(I,N2),I=1,5),M2,S
       EG(M2)
  6637         WRITE (NT,59) (BLANK,J=J1,J2)
  6638         WRITE (NT,60) (BLANK,J=J1,J2)
  6639         WRITE (NT,61) (BLANK,UNITL,UNITL,UNITM,UNITL,U
       NITL,UNITM,J=J1,J2)
  6640         WRITE (NT,38)
  6641         IF  (.NOT.LNEW)  GO TO 56
  6642         JJ = 4*(J2-J1+1)
  6643         DO  55  I=1,NLINES
  6644      55 WRITE (NT,62)  USEC(I),(ZTTH(J,I,IT),J=1,JJ)
  6645      56 CONTINUE
  6646      57 FORMAT('0',26X,'CONTACT FORCES - BELTS VS. SEG
       MENTS')
  6647      58 FORMAT(' ',7X,2(A4,'   BELT',I3,' (',5A4,') VS
       . SEGMENT',I3,
  6648         *        ' (',A4,') ') )
  6649      59 FORMAT(' ',2X,2(A4,11X,'ANCHOR  POINT  A',14X,
       'ANCHOR  POINT  B'))
  6650      60 FORMAT(4X,'TIME',2(A4,5X,'STRAIN',7X,'FORCE',1
       2X,
  6651         *                     'STRAIN',7X,'FORCE', 3X)
        )
  6652      61 FORMAT(3X,'(MSEC)',2(A4,2X,'(',A4,'/',A4,')',4
       X,'(',A4,')',9X,
  6653         *                     '(',A4,'/',A4,')',4X,'
       (',A4,')',3X) )
  6654      62 FORMAT(F9.3,4(F15.6,F12.2,3X) )
=  6655    C                                                             1008      C                                                          =
=====================================================        1 equal line   ===========================================================
=  6657    C                                                             1010      C                                                          =
  6658      83 IF  (NHRNSS.LE.O)  GO TO 91                                1011        340 WRITE (NT,10) DATE,PAGE,COMENT,VPSTTL,BDYTTL
  6659         MBSF = O                                                   1012            WRITE (NT,350)
  6660         J1 = 1                                                     1013        350 FORMAT('O',26X,'HARNESS SYSTEM BELT ENDPOINT F
                                                                                   ORCES')
  6661         K1 = 1                                                     1014            WRITE (NT,360) (BLANK,(KONTLP(I,J),I=1,2),J=1,
                                                                                   KKAS)
  6662         DO  85  I=1,NHRNSS                                         1015        360 FORMAT(9X,2(A4,11X,'BELT NO.',I4,' OF HARNESS
                                                                                   NO.',I3,15X))
  6663         IF  (NBLTPH(I).LE.O)  GO TO 85                             1016            WRITE (NT,370) (BLANK,(KONTLP(I,J),I=3,4),J=1,
                                                                                   KKAS)
  6664         J2 = J1 + NBLTPH(I) - 1                                    1017        370 FORMAT(9X,2(A4,6X,'POINT NO.',I5,16X,'POINT NO
                                                                                   .',I5,6X))
  6665         DO  84  J=J1,J2                                            1018            WRITE (NT,320) (BLANK,J=1,KKAS)
  6666         MBSF = MBSF + 1                                            1019            WRITE (NT,330) (BLANK,UNITL,UNITL,UNITM,UNITL,
                                                                                   UNITL,UNITM,J=1,
  6667         IF  (NPTSPB(J).LE.O)  GO TO 84                             1020             1KKAS)
  6668         K2 = K1 + NPTSPB(J) - 1                                    1021            WRITE (NT,200)
  6669         NOPL(2*MBSF-1) = J                                         1022            GO TO 550
  6670         NOPL(2*MBSF  ) = I
  6671         MOPL(2*MBSF-1) = K1
  6672         MOPL(2*MBSF  ) = K2
```

Table C.5. Comparison of GUCVS with GMCVS (Page 14 of 41)

```
 6673          K1 = K2 + 1
 6674       84 CONTINUE
 6675          J1 = J2 + 1
 6676       85 CONTINUE
 6677          DO  87  J1=1,MBSF,2
 6678          J2 = MINO(J1+1,MBSF)
 6679          MT = MT + 1
 6680          NT = MT
 6681          IF  (LNEW)  NT = 6
 6682          IT = MT - 20
 6683          PAGE = FLOAT(MT) + XPAGE
 6684          WRITE (NT,21) DATE,PAGE,COMENT,VPSTTL,BDYTTL
 6685          WRITE (NT,88)
 6686          WRITE (NT,89) (BLANK,NOPL(2*J-1),NOPL(2*J),J=J
        1,J2)
 6687          WRITE (NT,90) (BLANK,MOPL(2*J-1),MOPL(2*J),J=J
        1,J2)
 6688          WRITE (NT,60) (BLANK,J=J1,J2)
 6689          WRITE (NT,61) (BLANK,UNITL,UNITL,UNITM,UNITL,U
       NITL,UNITM,J=J1,J2)
 6690          WRITE (NT,38)
 6691          IF  (.NOT.LNEW)  GO TO 87
 6692          JJ = 4*(J2-J1+1)
 6693          DO  86  I=1,NLINES
 6694       86 WRITE (NT,62)  USEC(I),(ZTTH(J,I,IT),J=1,JJ)
 6695       87 CONTINUE
 6696       88 FORMAT('O',26X,'HARNESS SYSTEM BELT ENDPOINT F
       ORCES')
 6697       89 FORMAT(9X,2(A4,11X,'BELT NO.',I4,' OF HARNESS
       NO.',I3,15X))
 6698       90 FORMAT(9X,2(A4,6X,'POINT NO.',I5,16X,'POINT NO
       .',I5,6X))
= 6699     C                                                   1023     C                                                                  =
==================================================     1 equal line ====================================================================
= 6701     C                                                   1025     C                                                                  =
  6702        91 IF  (NSD.LE.O)  GO TO 63                      1026       380 M1 = KONTLP(1,1)
  6703           DO  94  J1=1,NSD,4                            1027           N1 = KONTLP(2,1)
  6704           J2 = MINO(J1+3,NSD)                           1028           M2 = KONTLP(3,1)
  6705           MT = MT + 1                                   1029           N2 = KONTLP(4,1)
  6706           NT = MT                                       1030           M3 = KONTLP(1,2)
  6707           IF  (LNEW)  NT = 6                            1031           N3 = KONTLP(2,2)
  6708           IT = MT - 20                                  1032           M4 = KONTLP(3,2)
  6709           PAGE = FLOAT(MT) + XPAGE                      1033           N4 = KONTLP(4,2)
  6710           WRITE (NT,21) DATE,PAGE,COMENT,VPSTTL,BDYTTL  1034           JKAS = 1
  6711           WRITE (NT,95) (BLANK,J,J=J1,J2)               1035           IF (M2 .NE. O) JKAS = 2
  6712           DO  92  J=J1,J2                               1036           IF (M3 .NE. O) JKAS = 3
  6713           M1 = MSDM(J)                                  1037           IF (M4 .NE. O) JKAS = 4
  6714           N1 = MSDN(J)                                  1038           WRITE (NT,10) DATE,PAGE,COMENT,VPSTTL,BDYTTL
  6715     C     POSSIBLE OVERFLOW INTO NOPL ARRAY IS INTENTION  1039           WRITE (NT,390) (BLANK,J,J=1,JKAS)
          AL.
  6716           HEAD(2*J-1) = SEG(M1)                         1040       390 FORMAT('O',26X,'SPRING DAMPER FORCES'/
  6717        92 HEAD(2*J  ) = SEG(N1)                         1041          19X,4(A3,3X,'SPRING DAMPER NO.',I3,4X))
  6718           WRITE (NT,96)(BLANK,MSDM(J),HEAD(2*J-1),MSDN(J  1042           GO TO (400,404,406,408),JKAS
          ),HEAD(2*J),J=J1,J2)
  6719           WRITE (NT,97) (BLANK,J=J1,J2)                 1043      400  WRITE(NT,401) BLANK,M1,SEG(M1),N1,SEG(N1)
  6720           WRITE (NT,98) (BLANK,UNITL,UNITM,J=J1,J2)     1044       401 FORMAT(9X,4(A3,'SEG',I3,'(',A4,') - SEG',I3,'(
                                                                         ',A4,')'))
  6721           WRITE (NT,38)                                 1045           GO TO 409
  6722           IF  (.NOT.LNEW)  GO TO 94                     1046      404  WRITE(NT,401) BLANK,M1,SEG(M1),N1,SEG(N1),
  6723           JJ = 2*(J2-J1+1)                              1047          1BLANK,M2,SEG(M2),N2,SEG(N2)
```

Table C.5. Comparison of GUCVS with GMCVS (Page 15 of 41)

```
  6724          DO  93  I=1,NLINES                               1048          GO TO 409
  6725       93 WRITE (NT,99)  USEC(I),(ZTTH(J,I,IT),J=1,JJ)     1049     406  WRITE(NT,401) BLANK,M1,SEG(M1),N1,SEG(N1),
  6726       94 CONTINUE                                         1050         1BLANK,M2,SEG(M2),N2,SEG(N2),
  6727       95 FORMAT('O',26X,'SPRING DAMPER FORCES'/           1051         2BLANK,M3,SEG(M3),N3,SEG(N3)
  6728        *      9X,4(A3,3X,'SPRING DAMPER NO.',I3,4X))      1052          GO TO 409
  6729       96 FORMAT(9X,4(A3,'SEG',I3,'(',A4,') - SEG',I3,'(   1053     408  WRITE(NT,401) BLANK,M1,SEG(M1),N1,SEG(N1),
            ',A4,')'))
  6730       97 FORMAT(4X,'TIME',1X,4(A3,5X,'LENGTH',7X,'FORCE   1054         1BLANK,M2,SEG(M2),N2,SEG(N2),
            ',4X))
  6731       98 FORMAT(3X,'(MSEC)',4(A3,5X,'(',A4,')',6X,'(',A   1055         2BLANK,M3,SEG(M3),N3,SEG(N3),
            4,')',4X))
  6732       99 FORMAT (F9.3,4(F14.3,F12.2,4X))                  1056         3BLANK,M4,SEG(M4),N4,SEG(N4)
                                                                 1057     409  WRITE (NT,410) (BLANK,J=1,JKAS)
                                                                 1058      410 FORMAT(4X,'TIME',1X,4(A3,5X,'LENGTH',7X,'FORCE
                                                                          ',4X))
                                                                 1059          WRITE (NT,420) (BLANK,UNITL,UNITM,J=1,JKAS)
                                                                 1060      420 FORMAT(3X,'(MSEC)',4(A3,5X,'(',A4,')',6X,'(',A
                                                                          4,')',4X))
                                                                 1061          WRITE (NT,200)
                                                                 1062          GO TO 550
= 6733      C                                                    1063     C                                                     =
===================================================== 1 equal line ==========================================================
= 6735      C                                                  . 1065     C                                                     =
  6736       63 MSSF = O                                         1066      430 N1A = KONTLP(2,1)
  6737          DO  65  J=1,NSEG                                 1067          N1 = KONTLP(1,1)
  6738          IF (MNSEG(J).EQ.O) GO TO 65                      1068          M1A = KONTLP(4,1)
  6739          LSEG = MNSEG(J)                                  1069          M1 = KONTLP(3,1)
  6740          DO 64  I=1,LSEG                                  1070          WRITE (NT,10) DATE,PAGE,COMENT,VPSTTL,BDYTTL
  6741          MSSF = MSSF+1                                    1071          WRITE (NT,440) N1A,SEG(N1),M1A,SEG(M1),UNITL,N
                                                                          1,M1
  6742          NOPL(MSSF) = J                                   1072         1,UNITL,UNITM,UNITM,UNITM
  6743       64 MOPL(MSSF) = MSEG(2,I,J)                         1073      440 FORMAT('O',26X,'CONTACT FORCES - ELLIPSOID NO.
                                                                          ',I3,' (',A4,
  6744       65 CONTINUE                                         1074         1') VS.  ELLIPSOID NO.',I3,' (',A4,')'//
                                                                           13X,'DEFL-  NORMAL
  6745          IF (MSSF.EQ.O) GO TO 70                          1075         2 FRICTION RESULTANT',
                                                                           14X,'CONTACT LOCATI
  6746          DO 67 J=1,MSSF                                   1076         3ON (',A4,')'/
                                                                           4X,'TIME     ECTION'
  6747          MT = MT + 1                                      1077         4,3(3X,'FORCE',1X),
                                                                           2('  SEG.',I3,' LOC
  6748          NT = MT                                          1078         5AL REFERENCE ')/
                                                                           3X,'(MSEC)',3X,'(',
  6749          IF  (LNEW)  NT = 6                               1079         6A4,')', 3(3X,'(',A4,')'),
                                                                           2(5X,'X',7X,'Y',7X,
  6750          IT = MT - 20                                     1080         7'Z',4X)/1X)
  6751          PAGE = FLOAT(MT) + XPAGE                         1081          GO TO 550
  6752          WRITE (NT,21) DATE,PAGE,COMENT,VPSTTL,BDYTTL     1082     C
  6753          N1 = NOPL(J)                                     1083     C     AIRBAG QUANTITIES HEADINGS
  6754          M1 = MOPL(J)                                     1084     C
  6755          WRITE (NT,68) N1,SEG(N1),M1,SEG(M1),UNITL,N1,M   1085      450 WRITE (NT,10) DATE,PAGE,COMENT,VPSTTL,BDYTTL
            1
  6756        *            ,UNITL,UNITM,UNITM,UNITM              1086          WRITE (NT,460) J,(BAGTTL(I,J),I=1,5)
  6757          IF  (.NOT.LNEW)  GO TO 67                        1087      460 FORMAT('O',26X,'PARAMETERS FOR AIRBAG NO.',I2,
                                                                          4X,5A4//
  6758          DO  66  I=1,NLINES                               1088         116X,'SUPPLY CYLINDER   STATIC'/
  6759       66 WRITE (NT, 69)  USEC(I),(ZTTH(JJ,I,IT),JJ=1,10   1089         24X,'TIME',8X,'PRES.',4X,'TEMP.',4X,'PRES.',12X
            )                                                             ,'AIRBAG',        3X,
  6760       67 CONTINUE                                         1090         3'CENTER',14X,'AIRBAG  SEMIAXES',12X,'ORIENTATI
                                                                          ON (DEG.)'/      3X,
```

Table C.5. Comparison of GUCVS with GMCVS (Page 16 of 41)

```
  6761       68 FORMAT('O',26X,'CONTACT FORCES - SEGMENT NO.',    1091        4'(MSEC)',7X,'(PSIG)  (DEG.R)   (PSIG)',8X,'X',
         I3,' (',A4,                                                       8X,'Y',8X,'Z',
  6762          *                      ') VS. SEGMENT NO.',I3,    1092       511X,'A',8X,'B',8X,'C',10X,'YAW',4X,'PITCH',5X,
         ' (',A4,')'//                                                     'ROLL'/ )
  6763          *    13X,'DEFL-  NORMAL  FRICTION RESULTANT',     1093          GO TO 550
  6764          *    14X,'CONTACT LOCATION (',A4,')'/
  6765          *    4X,'TIME    ECTION',3(3X,'FORCE',1X),
  6766          *    2('  SEG.',I3,' LOCAL REFERENCE ')/
  6767          *    3X,'(MSEC)',3X,'(',A4,')', 3(3X,'(',A4,')'
         ),
  6768          *    2(5X,'X',7X,'Y',7X,'Z',4X)/1X)
  6769       69 FORMAT(2F9.3,3F9.2,3F8.3,2X,3F8.3)
= 6770       C                                                    1094       C                                                        =
================================================================  1 equal line  ===================================================
= 6772       C                                                    1096       C                                                        =
  6773       70 IF (NBAG.EQ.O) GO TO 82                           1097         470 DO 520 K = 1, KKAS
  6774          DO  77  J=1,NBAG                                  1098             N1 = KONTLP(1,K)
  6775          IF (MNBAG(J).EQ.O) GO TO 77                       1099             N2 = KONTLP(2,K)
  6776          MT = MT + 1                                       1100             IF (N1 .GT. O) GO TO 480
  6777          NT = MT                                           1101             HEAD(1,K) = PHED(-N1)
  6778          IF  (LNEW)  NT = 6                                1102             GO TO 490
  6779          IT = MT - 20                                      1103         480 HEAD(1,K) = SEG(N1)
  6780          PAGE = FLOAT(MT) + XPAGE                          1104         490 J2 = 3
  6781          WRITE (NT,21) DATE,PAGE,COMENT,VPSTTL,BDYTTL      1105             IF (N2) 500, 520, 510
  6782          WRITE (NT,78) J,(BAGTTL(I,J),I=1,5)               1106         500 HEAD(2,K) = PHED(-N2)
  6783          IF  (.NOT.LNEW)  GO TO 72                         1107             J2 = 4
  6784          DO  71  I=1,NLINES                                1108             GO TO 520
  6785       71 WRITE (NT, 79)  USEC(I),(ZTTH(JJ,I,IT),JJ=1,12    1109         510 HEAD(2,K) = SEG(N2)
         )
  6786       72 KBAG = O                                          1110             J2 = 4
  6787          KP = NPANEL(J)+1                                  1111         520 CONTINUE
  6788          DO 73 K=1,KP                                      1112             J = KONTLP(1,KKAS)
  6789          KBAG = KBAG+1                                     1113             WRITE (NT,10) DATE,PAGE,COMENT,VPSTTL,BDYTTL
  6790       73 HEAD(KBAG) = PHED(K)                              1114             WRITE (NT,530)UNITM,J,(BAGTTL(I,J),I=1,5),(BLA
                                                                           NK,J,(HEAD(I,K),
  6791          KP = MNBAG(J)                                     1115            1I = 1, 2), K = 1, J2)
  6792          DO 74 K=1,KP                                      1116         530 FORMAT('O',26X,'CONTACT FORCES (',A4,') ON AIR
                                                                           BAG NO.',I2,4X,5A4//
  6793          KBAG = KBAG+1                                     1117            1/4X,'TIME',4(A1,11X,'AIRBAG',I2,' VS. ',A4,1X)
                                                                           )
  6794          M = MBAG(2,K,J)                                   1118             WRITE (NT,540) (BLANK,K=1,J2)
  6795       74 HEAD(KBAG) = SEG(M)                               1119         540 FORMAT(3X,'(MSEC)',4(A1,9X,'X',8X,'Y',8X,'Z',1
                                                                           X))
  6796          DO 76 J1=1,KBAG,4                                 1120             WRITE (NT,200)
  6797          J2 = MINO(J1+3,KBAG)                              1121         550 RETURN
  6798          MT = MT + 1
  6799          NT = MT
  6800          IF  (LNEW)  NT = 6
  6801          IT = MT - 20
  6802          PAGE = FLOAT(MT) + XPAGE
  6803          WRITE (NT,21) DATE,PAGE,COMENT,VPSTTL,BDYTTL
  6804          WRITE (NT,80)UNITM,J,(BAGTTL(I,J),I=1,5),(BLAN
         K,J,HEAD(K),K=J1,J2)
  6805          WRITE (NT,51) (BLANK,K=J1,J2)
  6806          WRITE (NT,38)
  6807          IF  (.NOT.LNEW)  GO TO 76
  6808          JJ = 3*(J2-J1+1)
  6809          DO  75  I=1,NLINES
  6810       75 WRITE (NT, 81)  USEC(I),(ZTTH(K,I,IT),K=1,JJ)
  6811       76 CONTINUE
```

Table C.5. Comparison of GUCVS with GMCVS (Page 17 of 41)

```
  6812        77 CONTINUE
  6813        78 FORMAT('0',26X,'PARAMETERS FOR AIRBAG NO.',I2,  .
           4X,5A4//
  6814           *     16X,'SUPPLY CYLINDER   STATIC'/
  6815           *     4X,'TIME',8X,'PRES.',4X,'TEMP.',4X,'PRES.
           ',12X,'AIRBAG',
  6816           *  3X,'CENTER',14X,'AIRBAG  SEMIAXES',12X,'ORIE
           NTATION (DEG.)'/
  6817           *  3X,'(MSEC)',7X,'(PSIG)  (DEG.R)   (PSIG)',8X
           ,'X',8X,'Y',8X,'Z',
  6818           *  11X,'A',8X,'B',8X,'C',10X,'YAW',4X,'PITCH',5
           X,'ROLL'/ )
  6819        79 FORMAT (F9.3,3X,3F9.2,2(3X,3F9.3),3X,3F9.2)
  6820        80 FORMAT('0',26X,'CONTACT FORCES (',A4,') ON AIR
           BAG NO.',I2,4X,5A4//
  6821           *   /4X,'TIME',4(A1,11X,'AIRBAG',I2,' VS. ',A4,
           1X))
  6822        81 FORMAT (F9.3,4(3X,3F9.2))
  6823        82 RETURN
= 6824           END                                         1122          END                                =
```

Table C.5. Comparison of GUCVS with GMCVS (Page 18 of 41)

```
Unit 0: CVSB(166,1501)                                        Unit 1: SUSP:CIGS(6527,7862)
=   166          SUBROUTINE HINPUT                           6527          SUBROUTINE HINPUT                             =
=================================================== 119 equal lines =================================================
=   286          * U2(3,100), SEGLP(3,100),SEGLV(3,100),SEGLA(3,   6647          * U2(3,100), SEGLP(3,100),SEGLV(3,100),SEGLA(3, =
=          100),NSYM(100)                                           100),NSYM(100)                                       =
    287          COMMON/RSAVE/XSG(3,20,3),DPMI(3,3,100),LPMI(10  6648          COMMON/RSAVE/DPMI(3,3,100),LPMI(100)
           0),NSG(7),MSG(20,7)
=   288          COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),  6649          COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100), =
=          PLOSS(2,100),                                           PLOSS(2,100),                                       =
=================================================== 26 equal lines ==================================================
=   315          *   '      POINT   POINT  SEGMENT LENGTH  ENER  6676          *   '      POINT   POINT  SEGMENT LENGTH  ENER =
=          GY LOSS',5X,                                            GY LOSS',5X,                                        =
    316          *      'REFERENCE POINT (',A4,')',13X,'BELT FORC  6677          *      'REFERENCE POINT (',A4,')',13X,'BELT FORC
           ES  ('A4,')',                                           ES  (',A4,')',
=   317          *      9X,'ENERGY LOSS'/                        6678          *      9X,'ENERGY LOSS'/                      =
=================================================== 826 equal lines =================================================
=  1144          *    /'  SEGMENT',11X,'LINEAR POSITION (',A4,')  7505          *    /'  SEGMENT',11X,'LINEAR POSITION (',A4,') =
=          ',                                                      ',                                                  =
   1145          *      14X,'LINEAR VELOCITY ('A4,'/',A4,')'/    7506          *      14X,'LINEAR VELOCITY (',A4,'/',A4,')'/
=  1146          *      '  NO. SEG',2(9X,'X',11X,'Y',11X,'Z',5X) 7507          *      '  NO. SEG',2(9X,'X',11X,'Y',11X,'Z',5X) =
=          )                                                       )                                                   =
=================================================== 4 equal lines ===================================================
=  1151       44 FORMAT('O INITIAL ANGULAR ROTATION AND VELOCIT  7512       44 FORMAT('O INITIAL ANGULAR ROTATION AND VELOCIT =
=          Y',71X,'CARDS G.3'//                                    Y',71X,'CARDS G.3'//                                =
   1152          *      '  SEGMENT',11X,'ANGULAR ROTATION (DEG)'  7513          *      '  SEGMENT',11X,'ANGULAR ROTATION (DEG)'
                                                                   '
=  1153          *      14X,'ANGULAR VELOCITY (DEG/',A4,')'/     7514          *      14X,'ANGULAR VELOCITY (DEG/',A4,')'/   =
=================================================== 347 equal lines =================================================
=  1501          END                                         7862          END                                           =
```

Table C.5. Comparison of GUCVS with GMCVS (Page 19 of 41)

```
Unit 0: CVSB(1502,1830)
= 1502          SUBROUTINE OUTPUT(IJK)
  1503    C
               REV 20 05/18/80
  1504    C     CONTROLS TABULATED OUTPUT ON FORTRAN UNITS (ST
          ARTING WITH NO. 21)
  1505    C     OF SELECTED OPTIONAL SEGMENT LINEAR AND ANGULA
          R ACCELERATIONS,
  1506    C     VELOCITIES AND DISPLACEMENTS, JOINT PARAMETERS
           AND SELECTED DATA
  1507    C     FROM ALL ALLOWED CONTACT FORCE COMPUTATIONS BE
          TWEEN BODY SEGMENTS
  1508    C     AND VEHICLE COMPONENTS.
```

```
Unit 1: SUSP:CIGS(7863,8445)
7863          SUBROUTINE OUTPUT(IJK)                                    =
7864          IMPLICIT REAL*8(A - H,O - Z)
7865          COMMON /CONTRL/ TIME, NSEG, NJNT, NPL, NBLT, N
      BAG, NVEH, NGRND,
7866         1        NS, NQ, NSD, NFLX, NHRNSS, NWINDF, NJNT
      F, NPRT(36)
7867          COMMON /SGMNTS/ D(3,3,100), WMEG(3,100), WMEGD
      (3,100), U1(3,100),
7868         1        U2(3,100), SEGLP(3,100), SEGLV(3,100),
      SEGLA(3,100),
7869         2        NSYM(100)
7870          COMMON /DESCRP/ PHI(3,100), W(100), RW(100), S
      R(3,200), HA(3,200),
7871         1        HB(3,200), RPHI(3,100), HT(3,3,200), SP
      RING(5,300),
7872         2        VISC(7,300), JNT(100), IPIN(100), ISING
      (100), IGLOB(100),
7873         3        JOINTF(100)
7874          COMMON /JBARTZ/ MNPL(100), MNBLT(8), MNSEG(100
      ), MNBAG(6),
7875         1        MPL(3,5,100), MBLT(3,5,8), MSEG(3,5,100
      ), MBAG(3,10,6),
7876         2        NTPL(5,100), NTBLT(5,8), NTSEG(5,100)
7877          COMMON /TITLES/ DATE(3), COMENT(40), VPSTTL(20
      ), BDYTTL(5),
7878         1        BLTTTL(5,8), PLTTL(5,100), BAGTTL(5,6),
       SEG(100),
7879         2        JOINT(100), CGS(100), JS(100)
7880          REAL DATE, COMENT, VPSTTL, BDYTTL, BLTTTL, PLT
      TL, BAGTTL, SEG,
7881         1    JOINT
7882          LOGICAL*1 CGS, JS
7883          COMMON /FORCES/ PSF(7,100), BSF(4,20), SSF(10,
      20), BAGSF(3,20),
7884         1        PRJNT(7,100), NPANEL(5), NPSF, NBSF, NS
      SF, NBGSF
7885          COMMON /CNSNTS/ PI, RADIAN, G, THIRD, EPS(24),
       UNITL, UNITM,
7886         1        UNITT, GRAVTY(3)
7887          COMMON/RSAVE/DPMI(3,3,100), LPMI(100)
7888          COMMON /COMAIN/ VAR(800), DER(800), DT, HO, HM
      AX, HMIN, RSTIME,
7889         1        ISTEP, NSTEPS, NDINT, NEQ, IRSIN, IRSOU
      T
7890          COMMON /DAMPER/ APSDM(3,20), APSDN(3,20), ASD(
      5,20), MSDM(20),
7891         1        MSDN(20)
7892          COMMON /IOCNTL/ LDNWRK, KWDPLN, IOTALK
7893          COMMON /HRNESS/ BAR(15,100), BB(100), BBDOT(10
      0), PLOSS(2,100),
7894         1        XLONG(20), HTIME(2), IBAR(5,100), NL(2,
      100), NPTSPB(20),
7895         2        NPTPLY(20), NTHRNS(20), NBLTPH(5)
7896          COMMON /HLDCON/ LDNARY(3), NRNBAS(246), LFIRST
      (241), KASNUM,
7897         1        KATKOL(12), KPRELM(3), KATKAS(3,12), KA
      SKTL(5,241), MAXPTS,
```

Table C.5. Comparison of GUCVS with GMCVS (Page 20 of 41)

```
=  1509     C
   1510           IMPLICIT REAL*8 (A-H,O-Z)
   1511           COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NV
         EH,NGRND,
   1512          *                NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNT
         F,NPRT(36)
   1513           COMMON/SGMNTS/ D(3,3,100),WMEG(3,100),WMEGD(3,
         100),U1(3,100),
   1514          * U2(3,100), SEGLP(3,100),SEGLV(3,100),SEGLA(3,
         100),NSYM(100)
   1515           COMMON/DESCRP/ PHI(3,100),W(100),RW(100),SR(3,
         200),HA(3,200),
   1516          *HB(3,200),RPHI(3,100),HT(3,3,200),SPRING(5,300
         ),VISC(7,300),
   1517          *              JNT(100),IPIN(100),ISING(100),IGLOB
         (100),JOINTF(100)
   1518           COMMON/JBARTZ/ MNPL( 100),MNBLT( 8),MNSEG( 100
         ),MNBAG(  6),
   1519          *                MPL(3,5,100),MBLT(3,5,8),MSEG(3,5
         ,100),MBAG(3,10,6),
   1520          *                NTPL( 5,100),NTBLT( 5,8),NTSEG( 5
         ,100)
   1521           COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),B
         DYTTL(5),
   1522          *                BLTTTL(5,8),PLTTL(5,100),BAGTTL(5
         ,6),SEG(100),
   1523          *                JOINT(100),CGS(100),JS(100)
   1524           REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BA
         GTTL,SEG,JOINT
   1525           LOGICAL*1 CGS,JS
   1526           COMMON/FORCES/ PSF(7,100),BSF(4,20),SSF(10,20)
         ,BAGSF(3,20),
   1527          *                PRJNT(7,100),NPANEL(5),NPSF,NBSF,
         NSSF,NBGSF
   1528           COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),
   1529          *                UNITL,UNITM,UNITT,GRAVTY(3)
   1530           COMMON/RSAVE/XSG(3,20,3),DPMI(3,3,100),LPMI(10
         0),NSG(7),MSG(20,7)
   1531           COMMON/COMAIN/VAR(800),DER(800),DT,HO,HMAX,HMI
```

```
7898         2        LINES, KASE, KATG, NRN, LSTEP, LASREC(3
      )
7899          COMMON /TEMPVS/ RSTF(27), FDPMI(900)
7900          DIMENSION KSTF(27)
7901          REAL*4 FFIRST(241), FRNBAS(246), FENKAT(51), F
      DPMI, FLPMI(100),
7902         1        RSTF, FKASKL(1205)
7903          EQUIVALENCE (KSTF(1),RSTF(1)), (FFIRST(1),LFIR
      ST(1)),
7904         1              (FRNBAS(1),NRNBAS(1)), (FENKAT(1),
      KATKOL(1)),
7905         2              (FKASKL(1),KASKTL(1,1)), (FLPMI(1)
      ,LPMI(1))
7906          REAL*4 VS, BLANK, HARN1, HARN2, SPDAM1, SPDAM2
      , SPDAM3,
7907         1      SPDAM4, PARM, PNTS
7908          DATA INZ, VS, BLANK, HARN1, HARN2, SPDAM1, SPD
      AM2, SPDAM3, SPDAM4,
7909         1      PARM, PNTS /O, ' VS ', '    ', 'HARN', 'E
      SS', 'SPRI', 'NG D',
7910        2      'AMPE', 'R', 'PARM', 'PNTS'/
7911  C                                                        =
7912          IF (INZ .NE. O) GO TO 10
7913          INZ = 1

7914          LINES = O

7915          KOPY = 1

7916          TPRINT = O.DO

7917       10 IF (IJK .NE. O) GO TO 30
```

Table C.5. Comparison of GUCVS with GMCVS (Page 21 of 41)

```
          N,RSTIME,
   1532        *              ISTEP,NSTEPS,NDINT,NEQ,IRSIN,IRSOU
          T
   1533        COMMON/DAMPER/ APSDM(3,20),APSDN(3,20),ASD(5,2
          0),MSDM(20),MSDN(20)
   1534        COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),
          PLOSS(2,100),
   1535        *              XLONG(20),HTIME(2),IBAR(5,100),NL
          (2,100),
   1536        *              NPTSPB(20),NPTPLY(20),NTHRNS(20),
          NBLTPH(5)
   1537        COMMON/TEMPVS/TDATA(14,300),ACC(7,20),T1(3),T2
          (3),T3(3),T4(9)
   1538        LOGICAL  LTAPE8 , LTHIST
   1539        DATA LINES/0/,LPP/45/
   1540   C
   1541        IF (IJK.NE.0) GO TO 13
=  1542   C                                                       7918    C                                                            =
=====================================================     1 equal line  ================================================================
=  1544   C                                                       7920    C                                                            =
   1545        DO  11  I=1,1740                                   7921         DO 20 I = 1, 1740
   1546     11 PSF(I,1) = 0.0                                     7922      20 PSF(I,1) = 0.0
   1547        GO TO 66                                           7923         GO TO 910
                                                                  7924    C
                                                                  7925    C          IF REGULATED PRINT INTERVAL, SELECT PROPE
                                                                          R TIME
=  1548   C                                                       7926    C                                                            =
   1549   C     LTHIST = TRUE MEANS PRINT LINE OF TIME HISTORY    7927      30 IF (NPRT(4) .NE. 0) GO TO 40
           DATA FOR THIS
   1550   C                          TIME POINT ON EACH OUTPUT    7928         IF (TIME .LE. TPRINT - EPS(8)) GO TO 910
          UNIT (NT).
                                                                  7929         TPRINT = TPRINT + DT
                                                                  7930      40 CALL ELTIME(1, 8)
                                                                  7931         IF (LINES .NE. 0) GO TO 680
=  1551   C                                                       7932    C                                                            =
   1552   C     LTAPE8 = TRUE MEANS WRITE TIME HISTORY DATA ON    7933    C          INITIALIZATION FOR FIRST TIME RECORDED
           TAPE 8.
=  1553   C                                                       7934    C                                                            =
   1554     13 NPRT4 = NPRT(4) + 4                                7935         IOTALK = NPRT(19)
   1555        IF  (NPRT4.LE.0 .OR. NPRT4.GT.8)  STOP 37          7936         LDNWRK = 8
   1556        GO TO  (66,66,66,16,15,14,14,15) , NPRT4           7937         LDNARY(1) = LDNWRK
   1557     14 LTHIST = .FALSE.                                   7938         LDNARY(2) = 10
   1558        LTAPE8 = .TRUE.                                    7939         LDNARY(3) = 11
   1559        GO TO 17                                           7940         KWDPLN = 20
   1560     15 LTHIST = .TRUE.                                    7941         INCSML = 1000000000
   1561        LTAPE8 = .TRUE.                                    7942         INCBIG = 0
   1562        GO TO 17                                           7943         INCCUR = 0
   1563     16 LTHIST = .TRUE.                                    7944         NRN = 0
   1564        LTAPE8 = .FALSE.                                   7945         NRNVAR = 0
   1565     17 CALL ELTIME (1,8)                                  7946         LBBELT = 0
   1566        IF  (LINES.NE.0)  GO TO 21                         7947         LBBAG = 0
   1567        PREVT = -999.0                                     7948         LBJNT = 0
   1568        IF  (IRSIN.NE.0)  GO TO 10                         7949         LBSEG = 0
                                                                  7950         LBPL = 0
                                                                  7951         LLFRST = 0
                                                                  7952         TIMLAS = 0.D0
                                                                  7953         LASREC(1) = 0
                                                                  7954         LASREC(2) = 0
                                                                  7955         LASREC(3) = 0
=  1569   C                                                       7956    C                                                            =
```

Table C.5. Comparison of GUCVS with GMCVS (Page 22 of 41)

```
C      1ST TIME IN ROUTINE, READ CARD INPUT FOR OUTPU
T CONTROL.
= C
C       1. NO. OF SEGMENT LINEAR  ACCELERATIONS,SEGME
NT NOS. AND LOCATION
C       2. NO. OF SEGMENT LINEAR  VELOCITIES   ,SEGME
NT NOS. AND LOCATION
C       3. NO. OF SEGMENT LINEAR  DISPLACEMENTS,SEGME
NT NOS. AND LOCATION
C       4. NO. OF SEGMENT ANGULAR ACCELERATIONS AND S
EGMENT NOS.
C       5. NO. OF SEGMENT ANGULAR VELOCITIES    AND S
EGMENT NOS.
C       6. NO. OF SEGMENT ANGULAR DISPLACEMENTS AND S
EGMENT NOS.
C       7. NO. OF JOINT PARAMETERS AND JOINT NOS.
```

```
C        LAYOUT RECORDING CATEGORIES AND CASES
C                                                  =
C        CATEGORY 1---TIME VALUES
       KATKOL(1) = 3
       KATKAS(1,1) = 1
C        CATEGORY 2---SEGMENT LINEAR KINEMATICS
       KATKOL(2) = 9
       KATKAS(1,2) = NGRND
C      CATEGORY 3---SEGMENT ANGULAR VELOCITIES, ETC.
       KATKOL(3) = 9
       KATKAS(1,3) = NGRND
C        CATEGORY 4---SEGMENT DIRECTION COSINE MATRICE
S
       KATKOL(4) = 9
       KATKAS(1,4) = NGRND
C        CATEGORY 5---JOINT FORCES
       KATKOL(5) = 8
       KATKAS(1,5) = NJNT
C        CATEGORY 6---PLANE-SEGMENT CONTACT FORCES
       KATG = 6
       KASE = 0
       KASNUM = 0
       IF (NPL .LE. 0) GO TO 70
       DO 60 I = 1, NPL
         KPL = IABS(MNPL(I))
         IF (KPL .EQ. 0) GO TO 60
         DO 50 J = 1, KPL
           KASE = KASE + 1
           KASKTL(1,KASE) = KATG + 2
           KASKTL(2,KASE) = MPL(1,J,I)
           KASKTL(3,KASE) = I
           KASKTL(4,KASE) = MPL(2,J,I)
           KASKTL(5,KASE) = MPL(3,J,I)
  50     CONTINUE
  60 CONTINUE
       KATKOL(KATG) = 7
       KATKAS(1,KATG) = KASE
       GO TO 80
  70 KATKOL(KATG) = 0
       KATKAS(1,KATG) = 0
  80 KASNUM = KASNUM + KASE
C        CATEGORY 7---SEGMENT:SEGMENT CONTACT FORCES
       KATG = 7
       KASE = 0
       IF (NSEG .LE. 0) GO TO 110
       DO 100 I = 1, NSEG
         KSG = IABS(MNSEG(I))
         IF (KSG .EQ. 0) GO TO 100
         DO 90 J = 1, KSG
           KASE = KASE + 1
           KASTRU = KASNUM + KASE
           KASKTL(1,KASTRU) = KATG + 2
           KASKTL(2,KASTRU) = MSEG(1,J,I)
```

Table C.5. Comparison of GUCVS with GMCVS (Page 23 of 41)

```
          KASKTL(3,KASTRU) = I
          KASKTL(4,KASTRU) = MSEG(2,J,I)
          KASKTL(5,KASTRU) = MSEG(3,J,I)
   90   CONTINUE
  100 CONTINUE
      IF (KASE .EQ. 0) GO TO 110
      KATKOL(KATG) = 10
      KATKAS(1,KATG) = KASE
      GO TO 120
  110 KATKOL(KATG) = 0
      KATKAS(1,KATG) = 0
  120 KASNUM = KASNUM + KASE
C      CATEGORY 8---BELT FORCES
      KATG = 8
      KASE = 0
      IF (NBLT .LE. 0) GO TO 150
      DO 140 I = 1, NBLT
        KBL = IABS(MNBLT(I))
        IF (KBL .EQ. 0) GO TO 140
        DO 130 J = 1, KBL
          KASE = KASE + 1
          KASTRU = KASNUM + KASE
          KASKTL(1,KASTRU) = KATG + 2
          KASKTL(2,KASTRU) = MBLT(1,J,I)
          KASKTL(3,KASTRU) = I
          KASKTL(4,KASTRU) = MBLT(2,J,I)
          KASKTL(5,KASTRU) = MBLT(3,J,I)
  130   CONTINUE
  140 CONTINUE
      KATKOL(KATG) = 4
      KATKAS(1,KATG) = KASE
      GO TO 160
  150 KATKOL(KATG) = 0
      KATKAS(1,KATG) = 0
  160 KASNUM = KASNUM + KASE
C      CATEGORY 9---HARNESS BELT FORCES
      KATG = 9
      KASE = 0
      IF (NHRNSS .LE. 0) GO TO 190
      J1 = 1
      K1 = 1
      DO 180 I = 1, NHRNSS
        KBL = NBLTPH(I)
        IF (KBL .LE. 0) GO TO 180
        J2 = J1 + KBL - 1
        DO 170 J = J1, J2
          KASE = KASE + 1
          KASTRU = KASNUM + KASE
          KN = NPTSPB(J)
          IF (KN .LE. 0) GO TO 170
          K2 = K1 + KN - 1
          KASKTL(1,KASTRU) = KATG + 2
C      HARNESS BELT NUMBER
          KASKTL(2,KASTRU) = J
C      HARNESS NUMBER
          KASKTL(3,KASTRU) = I
C      FIRST AND LAST POINT NUMBERS
          KASKTL(4,KASTRU) = K1
          KASKTL(5,KASTRU) = K2
          K1 = K2 + 1
```

Table C.5. Comparison of GUCVS with GMCVS (Page 24 of 41)

```
8069        170   CONTINUE
8070                J1 = J2 + 1
8071        180 CONTINUE
8072            KATKOL(KATG) = 4
8073            KATKAS(1,KATG) = KASE
8074            GO TO 200
8075        190 KATKOL(KATG) = 0
8076            KATKAS(1,KATG) = 0
8077        200 KASNUM = KASNUM + KASE
8078      C       CATEGORY 10---SPRING DAMPER FORCES
8079            KATG = 10
8080            KASE = 0
8081            IF (NSD .LE. 0) GO TO 220
8082            DO 210 I = 1, NSD, 2
8083              KASE = KASE + 1
8084              KASTRU = KASNUM + KASE
8085              KASKTL(1,KASTRU) = KATG + 2
8086              KASKTL(2,KASTRU) = MSDM(I)
8087              KASKTL(3,KASTRU) = MSDN(I)
8087.1          KASKTL(4,KASTRU) = 0
8087.2          KASKTL(5,KASTRU) = 0
8088              IF (I .EQ. NSD) GO TO 210
8089              KASKTL(4,KASTRU) = MSDM(I + 1)
8090              KASKTL(5,KASTRU) = MSDN(I + 1)
8091        210 CONTINUE
8092            KATKOL(KATG) = 4
8093            KATKAS(1,KATG) = KASE
8094            GO TO 230
8095        220 KATKOL(KATG) = 0
8096            KATKAS(1,KATG) = 0
8097        230 KASNUM = KASNUM + KASE
8098      C       CATEGORY 11---AIRBAG PARAMETERS
8099            KATG = 11
8100            KASE = 0
8101            IF (NBAG .LE. 0) GO TO 250
8102            NUM = NBAG + NBAG
8103            DO 240 I = 1, NUM
8104              KASE = KASE + 1
8105              KASTRU = KASNUM + KASE
8106              KB = (NUM + 1) / 2
8107              KC = 2 - MOD(NUM,2)
8108              KASKTL(1,KASTRU) = KATG + 2
8109      C       NUMBER OF AIRBAG
8110              KASKTL(2,KASTRU) = KB
8111      C       NUMBER OF PARAMETER GROUP
8112              KASKTL(3,KASTRU) = KC
8113        240 CONTINUE
8114            KATKOL(KATG) = 6
8115            KATKAS(1,KATG) = KASE
8116            GO TO 260
8117        250 KATKOL(KATG) = 0
8118            KATKAS(1,KATG) = 0
8119        260 KASNUM = KASNUM + KASE
8120      C     CATEGORY 12---AIRBAG CONTACT FORCES
8121            KATG = 12
8122            KASE = 0
8123            IF (NBAG .LE. 0) GO TO 290
8124            DO 280 I = 1, NBAG
8125              KPA = NPANEL(I) + 1
8126              KPT = KPA + MNBAG(I)
```

Table C.5. Comparison of GUCVS with GMCVS (Page 25 of 41)

```
        IF (KPT .LE. 0) GO TO 280
        DO 270 J = 1, KPT, 2
      J1 = J
      J2 = MINO(J+1, KPT)
      J5 = J2 - J1 + 1
      IF (J2 .GT. KPA) GO TO 262
      J5 = -J5
      J3 = -J1
      J4 = -J2
      GO TO 266
  262 IF (J1 .GT. KPA) GO TO 264
      J5 = 0
      J3 = -J1
      J4 = MBAG(2, J2-KPA, I)
      GO TO 266
  264 J3 = MBAG(2, J1-KPA, I)
      J4 = MBAG(2, J2-KPA, I)
  266     KASE = KASE + 1
          KASTRU = KASNUM + KASE
          KASKTL(1,KASTRU) = KATG + 2
          KASKTL(2,KASTRU) = I
          KASKTL(3,KASTRU) = J3
          KASKTL(4,KASTRU) = J4
          KASKTL(5,KASTRU) = J5
  270   CONTINUE
  280 CONTINUE
      KATKOL(KATG) = 6
      KATKAS(1,KATG) = KASE
      GO TO 300
  290 KATKOL(KATG) = 0
      KATKAS(1,KATG) = 0
  300 KASNUM = KASNUM + KASE
      DO 320 I = 1, 5
        IF (KATKAS(1,I) .GT. 0) GO TO 310
        KATKAS(2,I) = 0
        KATKAS(3,I) = 0
        GO TO 320
  310   KATKAS(2,I) = 1
        KATKAS(3,I) = KATKAS(1,I)
  320 CONTINUE
      LAST = 0
      DO 340 I = 6, 12
        IF (KATKAS(1,I) .GT. 0) GO TO 330
        KATKAS(2,I) = 0
        KATKAS(3,I) = 0
        GO TO 340
  330   KATKAS(2,I) = LAST + 1
        LAST = LAST + KATKAS(1,I)
        KATKAS(3,I) = LAST
  340 CONTINUE
      WRITE (6,350) KASNUM
  350 FORMAT ('1', 39X, 'RECORDED CATEGORIES AND CAS
ES'/20X,
     1      'NUMBER OF CASES = ', I4/5X, 'CASE', 5X
, 'CATEGORY', 5X,
     2      'N', 8X, 'NN', 10X, 'M', 8X, 'MM')
      DO 510 J = 1, KASNUM
        KAT = KASKTL(1,J)
        N = KASKTL(2,J)
        NN = KASKTL(3,J)
```

Table C.5. Comparison of GUCVS with GMCVS (Page 26 of 41)

```
                                                              8185             M = KASKTL(4,J)
                                                              8186             MM = KASKTL(5,J)
                                                              8187             JJ = KAT - 7
                                                              8188             IF (JJ .LT. 1) GO TO 510
                                                              8189             IF (JJ .GT. 7) GO TO 510
                                                              8190             DO 360 I = 1, 7
                                                              8191               RSTF(I) = BLANK
                                                              8192        360  CONTINUE
                                                              8193             GO TO (370, 390, 400, 420, 430, 440, 460), J
                                                                       J
                                                              8194        370  DO 380 I = 1, 5
                                                              8195               RSTF(I) = PLTTL(I,NN)
                                                              8196        380  CONTINUE
                                                              8197             GO TO 480
                                                              8198        390  RSTF(4) = SEG(N)
                                                              8199             GO TO 480
                                                              8200        400  DO 410 I = 1, 5
                                                              8201               RSTF(I) = BLTTTL(I,NN)
                                                              8202        410  CONTINUE
                                                              8203             GO TO 480
                                                              8204        420  RSTF(3) = HARN1
                                                              8205             RSTF(4) = HARN2
                                                              8206             GO TO 490
                                                              8207        430  RSTF(2) = SPDAM1
                                                              8208             RSTF(3) = SPDAM2
                                                              8209             RSTF(4) = SPDAM3
                                                              8210             RSTF(5) = SPDAM4
                                                              8211             GO TO 490
                                                              8212        440  DO 450 I = 1, 5
                                                              8213               RSTF(I) = BAGTTL(I,N)
                                                              8214        450  CONTINUE
                                                              8215             RSTF(7) = PARM
                                                              8216             GO TO 490
                                                              8217        460  DO 470 I = 1, 5
                                                              8218               RSTF(I) = BAGTTL(I,N)
                                                              8219        470  CONTINUE
                                                              8220             RSTF(7) = PNTS
                                                              8221             GO TO 490
                                                              8222        480  RSTF(6) = VS
                                                              8223             RSTF(7) = SEG(M)
                                                              8224        490  WRITE (6,500) J, KAT, N, NN, M, MM, (RSTF(I)
                                                                       ,I=1,7)
                                                              8225        500  FORMAT (5X, I3, 5X, I7, 3X, I6, 2X, I6, 7X,
                                                                       I6, 2X, I6, 5X, 6A4,
                                                              8226           1         2X, A4)
                                                              8227        510 CONTINUE
                                                              8228            WRITE (6,520) (K,(KATKAS(I,K),I=1,3),KATKOL(K)
                                                                       ,K=1,12)
                                                              8229        520 FORMAT ('0', 40X, ' CASES BY CATEGORY'/5X, 'CA
                                                                       TEGORY', 5X,
                                                              8230          1        'NUMBER OF CASES', 5X, 'STARTING CASE',
                                                                        5X, 'ENDING CASE',
                                                              8231          2        5X, 'RECORDED COLUMNS'/(5X,I6,10X,I6,15
                                                                       X,I6,10X,I6,14X,I6))
                                                              8232            IF (IOTALK .NE. 0) WRITE (6,660)
=  1579    C                                                  8233     C                                                 =
   1580          DO 20 K=1,7                                  8234     C      RECORD TIME INVARIANT INFORMATION ON HOLD FIL
                                                                       E WITH 1ST TIME
=  1581    C                                                  8235     C                                                 =
   1582    C     INPUT CARDS H.(K).(J) FOR K=1,3              8236            NRNVAR = 3
```

Table C.5. Comparison of GUCVS with GMCVS (Page 27 of 41)

```
=   1583      C
    1584            IF (K.LE.3) READ (5,18) KSG,(MSG(J,K),(XSG(I,J
              ,K),I=1,3),J=1,KSG)
    1585            IF (K.LE.3) WRITE(6,88) KSG,(MSG(J,K),(XSG(I,J
              ,K),I=1,3),J=1,KSG)
    1586         18 FORMAT(2I6,3F12.6/(I12,3F12.6))
    1587         88 FORMAT(1X,'CARDS H.(K).(J)',/,1X,2I6,3F12.6/(1
              X,I12,3F12.6))
=   1588      C
    1589      C     INPUT CARDS H.(K)      FOR K=4,7
```

```
8237            LAST = NRNVAR
8238            CALL STASH(VPSTTL, 20, LAST)
8239            CALL STASH(COMENT, 40, LAST)
8240            IF (NBLT .LE. 0) GO TO 530
8241            LBBELT = LAST
8242            CALL STASH(BLTTTL, 5*NBLT, LAST)
8243        530 IF (NBAG .LE. 0) GO TO 540
8244            LBBAG = LAST
8245            CALL STASH(BAGTTL, 5*NBAG, LAST)
8246        540 IF (NJNT .LE. 0) GO TO 550
8247            LBJNT = LAST
8248            CALL STASH(JOINT, NJNT, LAST)
8249        550 IF (NSEG .LE. 0) GO TO 590
8250            LBSEG = LAST
8251            CALL STASH(SEG, NGRND, LAST)
8252            CALL STASH(FLPMI, NSEG, LAST)
8253            LK = 0
8254            DO 580 K = 1, NSEG
8255              DO 570 J = 1, 3
8256                DO 560 I = 1, 3
8257                  LK = LK + 1
8258                  FDPMI(LK) = DPMI(I,J,K)
8259        560     CONTINUE
8260        570   CONTINUE
8261        580 CONTINUE
8262            CALL STASH(FDPMI, LK, LAST)
8263        590 IF (NPL .LE. 0) GO TO 600
8264            LBPL = LAST
8265            CALL STASH(PLTTL, 5*NPL, LAST)
8266        600 LLFRST = LAST
8267      C                                                            =
8268      C         COMPUTE LAYOUT SPACING AND BASE INDICES
8269      C                        FOR INDIVIDUAL CASES
8270      C                                                            =
8271            STPMIN = DMIN1(HMAX,HMIN)*1000.DO
8272            IF (NPRT(4) .EQ. 0) STPMIN = DT*1000.DO
8273            RSTF(1) = STPMIN
8274            TIMMAX = NSTEPS * DT * 1000.DO
8275            MAXPTS = -1
8276            IF (STPMIN .GT. 0.) MAXPTS = TIMMAX / STPMIN +
           1.5
8277      C       LIMIT MAXPTS TO MAXIMUM TIME POINT CAPACITY
          OF CVSOUT
8278            MAXPTS = MIN0(MAXPTS,6000)
8279            LSTEP = (MAXPTS + 2) / 3 + 1
8280            NRNBAS(1) = 0
8281            LFIRST(1) = -1
8282            DO 610 I = 2, 6
8283              ILAS = I - 1
8284              NRNBAS(I) = NRNBAS(ILAS) + KATKAS(1,ILAS) *
          LSTEP
8285              LFIRST(I) = -1
8286              IF (KATKAS(1,ILAS) .LE. 0) NRNBAS(ILAS) = 0
8287        610 CONTINUE
8288            KASTOP = KASNUM + 5
```

Table C.5. Comparison of GUCVS with GMCVS (Page 28 of 41)

```
8289          DO 620 I = 7, KASTOP
8290            NRNBAS(I) = NRNBAS(I - 1) + LSTEP
8291            IF (I .GT. KASNUM) GO TO 620
8292            LFIRST(I) = -1
8293      620 CONTINUE
8294          WRITE (6,630) KASTOP
8295      630 FORMAT ('1', 40X, 'TIME VARIANT QUANTITES RECO
       RDED (NUMBER =', I5,
8296         1        ')'/5X, 'QUANTITY NO.', 4X, 'QUANTITY H
       EADER REC',
8297         2        'ORD NO.', 4X, 'RECORDING CATEGORY', 5X
       , 'NUMBER OF CASES',
8298         3        5X, 'CASE NO.')
8299          DO 650 I = 1, KASTOP
8300          IF (I .GT. 5) GO TO 633
8301          KATG = I
8302          NUM = KATKAS(1,KATG)
8303          KAS = 0
8304          GO TO 636
8305     633    NUM = 1
8306          KAS = I - 5
8307          KATG = KASKTL(1,KAS)
8308     636    WRITE (6,640) I, NRNBAS(I), KATG, NUM, KAS
8309      640   FORMAT (8X, I6, 19X, I6, 19X, I6, 15X, I5, 1
       0X, I6)
8310      650 CONTINUE
8311          WRITE (6,660)
8312      660 FORMAT ('1 ')
8313          CALL STASH(FFIRST(1), KASNUM, LAST)
8314          CALL STASH(FRNBAS, KASTOP, LAST)
8315          CALL STASH(FKASKL, 5*KASNUM, LAST)
8316          LFENKT = 51
8317          KPRELM(1) = LAST + (KWDPLN + LFENKT - 1) / KWD
       PLN + 1
8318          KPRELM(2) = 1
8319          KPRELM(3) = 1
8320          CALL STASH(FENKAT, LFENKT, LAST)
8321          IZERO = 0
8322          RSTF(2) = PI
8323          RSTF(3) = G
8324          RSTF(4) = RADIAN
8325          RSTF(5) = UNITL
8326          RSTF(6) = UNITT
8327          RSTF(7) = UNITM
8328          RSTF(8) = DT * 1000.D0
8329          RSTF(9) = TIMMAX
8330          WRITE (LDNWRK'1) IZERO, IZERO, IZERO, IZERO, I
       ZERO, IZERO,
8331         1        IZERO, IZERO
8332          WRITE (LDNWRK'2) NVEH, NGRND, NSEG, NPL, NJNT,
        NSD, NBAG, NBLT,
8333         1      NHRNSS, MAXPTS, LSTEP, NSTEPS, KASNUM, N
       RNVAR, LBBELT,
8334         2      LBBAG, LBJNT, LBSEG, LBPL, LLFRST
8335          WRITE (LDNWRK'3) DATE, BDYTTL, KASTOP, LFENKT,
        (RSTF(I),I=1,9),
8336         1      NPRT(4)
8337          IF (IOTALK .EQ. 0) GO TO 680
8338          WRITE (6,670) NVEH, NGRND, NSEG, NPL, NJNT, NS
       D, NBAG, NBLT,
```

Table C.5. Comparison of GUCVS with GMCVS (Page 29 of 41)

```
                                                                        8339         1NHRNSS, MAXPTS, LSTEP, NSTEPS, KASNUM, NRNVAR,
                                                                              LBBELT, LBBAG,
                                                                        8340         2LBJNT, LBSEG, LBPL, LLFRST, DATE, BDYTTL, KAST
                                                                             OP, LFENKT,
                                                                        8341         3(RSTF(I),I=1,9), NPRT(4)
                                                                        8342      670 FORMAT ('O RN=2 ', 20I6/'  RN=3 ', 3A4, 5X, 5A
                                                                             4, 2I6, 3G20.10/7X,
                                                                        8343         1        G20.10, 8X, A4, 16X, A4, 16X, A4, 8X, 2
                                                                             G20.10, I3)
=  1590    C                                                            8344    C                                                     =
   1591          IF (K.GT.3) READ (5,19) KSG,(MSG(J,K),J=1,KSG)         8345    C      RECORD TIME VARIANT INFORMATION ON HOLD FILE
                                                                             FOR CURRENT TIME
   1592          IF (K.GT.3) WRITE(6,89) KSG,(MSG(J,K),J=1,KSG)
   1593       19 FORMAT(12I6/(I12,10I6))
   1594       89 FORMAT(1X,'CARDS H.(K)FOR K =4,7',/,1X,2I6,3F1
         2.6/(1X,I12,3F12.6))
   1595          IF (K.NE.7 .OR. KSG.EQ.0) GO TO 20
   1596          DO 12 J=1,KSG
   1597          L = MSG(J,K)
   1598          IF (IABS(IPIN(L)).EQ.4) MSG(J,K) = -L
   1599       12 CONTINUE
   1600       20 NSG(K) = KSG
   1601       10 IF  (.NOT.LTAPE8)  GO TO 21
   1602          WRITE  (8)         NSEG,NJNT,NPL,NBLT,NBAG,NVEH
         ,NGRND,NPANEL,
   1603          *                 MNPL,MNBLT,MNSEG,MNBAG,MPL,MBL
         T,MSEG,MBAG
   1604          WRITE  (8)         DATE,COMENT,VPSTTL,BDYTTL,BL
         TTTL,PLTTL,BAGTTL,
   1605          *                 SEG,JOINT,UNITL,UNITM,UNITT,NS
         G,MSG,XSG,
   1606          *                 NHRNSS,NBLTPH,NPTSPB,NSD,MSDM,
         MSDN
   1607       21 LINES = LINES + 1
   1608          IF  (MOD(LINES,LPP).EQ.1 .AND. LTHIST)  CALL H
         EDING (LINES,LPP)
   1609          NT = 20
   1610          USEC = 1000.0*TIME
=  1611    C                                                            8346    C                                                     =
   1612    C     COMPUTE AND PRINT DATA FOR 7 TYPES OF OUTPUT A         8347      680 LINES = LINES + 1
         BOVE
                                                                        8348         IF (LINES .GT. MAXPTS) GO TO 920
                                                                        8349         USEC = 1000.D0 * TIME
=  1613    C                                                            8350    C                                                     =
   1614          DO 44 K=1,7                                            8351    C      RECORD CURRENT TIME
   1615          IF (NSG(K).LE.0) GO TO 44
   1616          KSG = NSG(K)
   1617          J3  = 3
   1618          IF (K.EQ.7) J3 = 2
   1619          DO 43 J1=1,KSG,J3
   1620          J2 = MIN0(J1+J3-1,KSG)
   1621          NT = NT+1
   1622          DO 38 J=J1,J2
   1623          L = IABS(MSG(J,K))
   1624          GO TO (22,24,26,29,31,34,35),K
=  1625    C                                                            8352    C                                                     =
   1626    C     1. SEGMENT LINEAR ACCELERATIONS IN LOCAL REFER         8353         MULTPL = USEC / STPMIN + .5
         ENCE
   1627    C                                                            8354         RSTF(1) = USEC
   1628       22 CALL CROSS (WMEG(1,L),XSG(1,J,K),T1)                   8355         KSTF(2) = LINES
```

Table C.5. Comparison of GUCVS with GMCVS (Page 30 of 41)

```
      CALL CROSS (WMEG(1,L),T1,T2)
      CALL CROSS (WMEGD(1,L),XSG(1,J,K),T3)
      CALL MAT31(D(1,1,L),SEGLA(1,L),T4)
      DO 23 I=1,3
      ACC(I,J) = (T4(I)+T3(I)+T2(I))/G
   23 T1(I) = ACC(I,J)
      IF  (LPMI(L).NE.O)  CALL DOT31 (DPMI(1,1,L),T1
,ACC(1,J))
      GO TO 33
C
C     2. SEGMENT LINEAR VELOCITIES IN VEHICLE REFERE
NCE
C
   24 CALL CROSS (WMEG(1,L),XSG(1,J,K),T1)
      CALL DOT31(D(1,1,L),T1,T2)
      DO 25 I=1,3
   25 T3(I) = T2(I) + SEGLV(I,L) - SEGLV(I,NVEH)
      GO TO 28
C
C     3. SEGMENT LINEAR DISPLACEMENTS IN VEHICLE REF
ERENCE
C
   26 IF  (LPMI(L).EQ.O)  GO TO 76
      CALL DOT33 (DPMI(1,1,L),D(1,1,L),T4)
      CALL DOT31 (T4,XSG(1,J,K),T1)
      GO TO 77
   76 CALL DOT31 (D(1,1,L),XSG(1,J,K),T1)
   77 DO 27 I=1,3
   27 T3(I) = T1(I) + SEGLP(I,L) - SEGLP(I,NVEH)
   28 CALL MAT31 (D(1,1,NVEH),T3,ACC(1,J))
      GO TO 33
C
C     4. SEGMENT ANGULAR ACCELERATIONS IN LOCAL REFE
RENCE
C
   29 DO 30 I=1,3
      ACC(I,J) = WMEGD(I,L)/(2.O*PI)
   30 T1(I) = ACC(I,J)
      IF  (LPMI(L).NE.O)  CALL DOT31 (DPMI(1,1,L),T1
,ACC(1,J))
      GO TO 33
C
C     5. SEGMENT ANGULAR VELOCITIES IN VEHICLE REFER
ENCE
C
   31 CALL DOT31 (D(1,1,L),WMEG(1,L),T1)
      CALL MAT31 (D(1,1,NVEH),T1,T2)
      DO 32 I=1,3
   32 ACC(I,J) = (T2(I)-WMEG(I,NVEH))/(2.O*PI)
   33 ACC(4,J) = DSQRT(ACC(1,J)**2+ACC(2,J)**2+ACC(3
,J)**2)
      GO TO 38
C
C     6. SEGMENT ANGULAR DISPLACEMENTS IN VEHICLE RE
FERENCE
C
   34 IF  (LPMI(L).EQ.O)  GO TO 36
      CALL DOT33(DPMI(1,1,L),D(1,1,L),T4)

      CALL DOTT33(T4,D(1,1,NVEH),T1)
```

```
      KSTF(3) = MULTPL
      KATG = 1
      KASE = 1
      CALL OUTREC(RSTF(1))
      IF (LINES .EQ. 1) GO TO 690
      KOPY = O
      IF (USEC .LE. TIMLAS) GO TO 690

      KOPY = 1
      INCCUR = (USEC - TIMLAS) / STPMIN + .5
      INCSML = MINO(INCSML,INCCUR)

      INCBIG = MAXO(INCBIG,INCCUR)
  690 IF (NGRND .LE. O) GO TO 720
      DO 710 I = 1, NGRND
        DO 700 J = 1, 3
          RSTF(J) = SEGLP(J,I)
          RSTF(J + 3) = SEGLV(J,I)
          RSTF(J + 6) = SEGLA(J,I)
          RSTF(J + 9) = WMEG(J,I)

          RSTF(J + 12) = WMEGD(J,I)
          RSTF(J + 15) = U1(J,I)
          RSTF(J + 18) = D(J,1,I)
          RSTF(J + 21) = D(J,2,I)
          RSTF(J + 24) = D(J,3,I)
  700   CONTINUE
        KATG = 2
        KASE = I
        CALL OUTREC(RSTF(1))
        KATG = 3
        KASE = I
        CALL OUTREC(RSTF(10))

        KATG = 4
        KASE = I
        CALL OUTREC(RSTF(19))
  710 CONTINUE
  720 IF (NJNT .LE. O) GO TO 750

      KATG = 5
      DO 740 I = 1, NJNT
        DO 730 J = 1, 7

          RSTF(J) = PRJNT(J,I)
  730   CONTINUE
        KSTF(8) = IPIN(I)
        KASE = I
        CALL OUTREC(RSTF(1))
  740 CONTINUE

  750 IF (KASNUM .LE. O) GO TO 890
      DO 880 II = 1, KASNUM
C       KASE = ABSOLUTE CASE NUMBER AMONG ALL TIME D
EPENDANT RECS
        KASE = II
        KATG = KASKTL(1,II) - 2
C       KAS = RELATIVE CASE NUMBER WITHIN RECORDING
CATEGORY
        KAS = KASE - KATKAS(2,KATG) + 1
```

Table C.5. Comparison of GUCVS with GMCVS (Page 31 of 41)

```
1680          GO TO 37
1681       36 CALL DOTT33 (D(1,1,L),D(1,1,NVEH),T1)
1682       37 CALL YPRDEG(T1,ACC(1,J))
1683          TRACE = 0.5*(T1(1)+T2(2)+T3(3)-1.0)
1684          IF (TRACE.GT. 1.0) TRACE =  1.0
1685          IF (TRACE.LT.-1.0) TRACE = -1.0
1686          ACC(4,J) = DARCOS(TRACE)/RADIAN
1687          GO TO 38
1688    C
1689    C     7. JOINT PARAMETERS
1690    C
1691       35 ACC(1,J) = PRJNT(1,L)
1692          ACC(2,J) = PRJNT(2,L)/RADIAN
1693          ACC(3,J) = PRJNT(3,L)/RADIAN
1694          ACC(4,J) = PRJNT(4,L)/RADIAN
1695          ACC(5,J) = DSQRT(PRJNT(5,L))
1696          ACC(6,J) = DSQRT(PRJNT(6,L))
1697          ACC(7,J) = DSQRT(PRJNT(7,L))
1698       38 CONTINUE
1699          IF  (.NOT.LTAPE8)  GO TO 40
1700          KK = 0
1701          I2 = 4
1702          IF  (K.EQ.7)  I2 = 7
1703          DO  39  J=J1,J2
1704          DO  39  I=1,I2
1705          KK = KK+1
1706       39 TDATA(KK,NT-20) = ACC(I,J)
1707       40 IF  (.NOT.LTHIST)  GO TO 43
1708          IF (K.LE.6) WRITE (NT,41) USEC,((ACC(I,J),I=1,
        4),J=J1,J2)
1709       41 FORMAT(F9.3,3(3X,4F9.3) )
1710          IF (K.EQ.7) WRITE (NT,42) USEC,((ACC(I,J),I=1,
        7),J=J1,J2)
1711       42 FORMAT(F9.3,2(F5.0,3F9.3,2X,3F9.3))

1712       43 CONTINUE

1713       44 CONTINUE
1714    C
1715    C     PRINT PLANE FORCES
1716    C
1717          MPSF = O
1718          IF  (NPL.EQ.O)  GO TO 49
1719          DO  45  J=1,NPL
1720       45 MPSF = MPSF + MNPL(J)
1721          IF  (MPSF.EQ.O)  GO TO 49

1722          DO 47 J1=1,MPSF,2
1723          J2 = MINO(J1+1,MPSF)
1724          NT = NT+1
1725          IF  (.NOT.LTAPE8)  GO TO 47
1726          KK = 0
1727          DO  46  J=J1,J2
1728          DO  46  I=1,7
1729          KK = KK+1
1730       46 TDATA(KK,NT-20) = PSF(I,J)
1731       47 IF  (LTHIST)  WRITE (NT,48) USEC,((PSF(I,J),I=
        1,7),J=J1,J2)
1732       48 FORMAT(F9.3,2(F9.3,3F9.2,3F8.3) )
1733    C
```

```
8407             KOL = KATKOL(KATG)
8408             K = KATG - 5
8409             GO TO (760, 780, 820, 800, 810, 850, 840), K
8410      760    DO 770 I = 1, KOL
8411               RSTF(I) = PSF(I,KAS)
8412      770    CONTINUE
8413             GO TO 870
8414      780    DO 790 I = 1, KOL
8415               RSTF(I) = SSF(I,KAS)
8416      790    CONTINUE
8417             GO TO 870
8417.5   800     IF (KATKAS(2,8) .LE. 0) GO TO 820
8418             KAS = KAS + KATKAS(2,9) - KATKAS(2,8)
8419             GO TO 820
8419.5   810     IF (KATKAS(2,8) .LE. 0) GO TO 815
8420             KAS = KAS + KATKAS(2,10) - KATKAS(2,8)
8420.2           GO TO 820
8420.4   815     IF (KATKAS(2,9) .LE. 0) GO TO 820
8420.6           KAS = KAS + KATKAS(2,10) - KATKAS(2,9)
8421      820    DO 830 I = 1, KOL
8422               RSTF(I) = BSF(I,KAS)
8423      830    CONTINUE
8424             GO TO 870
8425      840    KAS = KAS + KATKAS(2,12) - KATKAS(2,11)
8426      850    K = KAS + KAS - 1
8427             DO 860 I = 1, KOL
8428               RSTF(I) = BAGSF(I,K)
8429      860    CONTINUE
8430      870    CALL OUTREC(RSTF(1))

8431      880 CONTINUE
8432     890  RSTF(1) = USEC

8433          WRITE (LDNWRK'1) RSTF(1), LINES, LASREC, MULTP
     L, INCBIG, INCSML
8434          IF (IOTALK .NE. 0) WRITE (6,900) RSTF(1), LINE
     S, LASREC, MULTPL,
8435        1 INCBIG, INCSML
8436      900 FORMAT ('O RN=1 ', G20.10, 7I12)
8437          LAST = LLFRST
8438          CALL STASH(FFIRST(1), KASNUM, LAST)
8439          IF (KOPY .NE. 0) TIMLAS = USEC
8440          CALL ELTIME(2, 8)
8441      910 RETURN
8442      920 WRITE (6,930)
8443      930 FORMAT ('OFATAL ERROR---MORE TIME POINTS THAN
     MIMIMUM STEP.')
8444          STOP 1111
```

Table C.5. Comparison of GUCVS with GMCVS (Page 32 of 41)

```
C     PRINT BELT FORCES
C
   49 MBSF = 0
      IF  (NBLT.EQ.0)  GO TO 67
      DO  50  J=1,NBLT
   50 MBSF = MBSF + MNBLT(J)
      IF  (MBSF.EQ.0)  GO TO 67
      DO 52 J1=1,MBSF,2
      J2 = MINO(J1+1,MBSF)
      NT = NT+1
      IF  (.NOT.LTAPE8)  GO TO 52
      KK = 0
      DO  51  J=J1,J2
      DO  51  I=1,4
      KK = KK+1
   51 TDATA(KK,NT-20) = BSF(I,J)
   52 IF  (LTHIST)  WRITE (NT,53) USEC,((BSF(I,J),I=
1,4),J=J1,J2)
   53 FORMAT(F9.3,4(F15.6,F12.2,3X) )
C
C     PRINT HARNESS-BELT ENDPOINT FORCES (STORED IN
BSF ARRAY).
C
   67 IF  (NHRNSS.LE.0)  GO TO 71
      MBSF1 = MBSF + 1
      DO  68  I=1,NHRNSS
   68 MBSF = MBSF + NBLTPH(I)
      DO  70  J1=MBSF1,MBSF,2
      J2 = MINO(J1+1,MBSF)
      NT = NT+1
      IF  (.NOT.LTAPE8)  GO TO 70
      KK = 0
      DO  69  J=J1,J2
      DO  69  I=1,4
      KK = KK+1
   69 TDATA(KK,NT-20) = BSF(I,J)
   70 IF  (LTHIST)  WRITE (NT,53) USEC,((BSF(I,J),I=
1,4),J=J1,J2)
C
C     PRINT SPRING DAMPER FORCES (STORED IN BSF ARRA
Y).
C
   71 IF  (NSD.LE.0)  GO TO 54
      MBSF1 = MBSF + 1
      MBSF = MBSF + (NSD+1)/2
      DO  73  J1=MBSF1,MBSF,2
      J2 = MINO(J1+1,MBSF)
      NT = NT+1
      IF  (.NOT.LTAPE8)  GO TO 73
      KK = 0
      DO  72  J=J1,J2
      DO  72  I=1,4
      KK = KK+1
   72 TDATA(KK,NT-20) = BSF(I,J)
   73 IF  (LTHIST)  WRITE (NT,74) USEC,((BSF(I,J),I=
1,4),J=J1,J2)
   74 FORMAT (F9.3,4(F14.3,F12.2,4X))
C
C     PRINT SEGMENT CONTACT FORCES
C
```

```
   54 MSSF = 0
      DO  55  J=1,NSEG
   55 MSSF = MSSF + MNSEG(J)
      IF  (MSSF.EQ.0)  GO TO 59
      DO 57 J=1,MSSF
      NT = NT+1
      IF  (.NOT.LTAPE8)  GO TO 57
      DO  56  I=1,10
   56 TDATA(I,NT-20) = SSF(I,J)
   57 IF  (LTHIST)  WRITE (NT,58) USEC,(SSF(I,J),I=1
 .10)
   58 FORMAT(2F9.3,3F9.2,3F8.3,2X,3F8.3)
C
C     PRINT AIRBAG FORCES
C
   59 IF (NBAG.EQ.0) GO TO 65
      K1 = 1
      DO 64 J=1,NBAG
      IF (MNBAG(J).EQ.0) GO TO 64
      KBAG = MNBAG(J)+NPANEL(J)+5
      DO 63 J1=1,KBAG,4
      J2 = MINO(J1+3,KBAG)
      K2 = K1+J2-J1
      NT = NT+1
      IF  (.NOT.LTAPE8)  GO TO 61
      KK = 0
      DO  60  K=K1,K2
      DO  60  I=1,3
      KK = KK+1
   60 TDATA(KK,NT-20) = BAGSF(I,K)
   61 IF  (.NOT.LTHIST)  GO TO 63
      IF  (J1.EQ.1)  WRITE (NT,75) USEC,((BAGSF(I,K)
 .I=1,3),K=K1,K2)
      IF  (J1.NE.1)  WRITE (NT,62) USEC,((BAGSF(I,K)
 .I=1,3),K=K1,K2)
   75 FORMAT (F9.3,3X,3F9.2,2(3X,3F9.3),3X,3F9.2)
   62 FORMAT(F9.3,4(3X,3F9.2))
   63 K1 = K2+1
   64 CONTINUE
   65 NT = NT-20
      IF  (LTAPE8)  WRITE (8) NT,USEC,((TDATA(I,J),I
 =1,14),J=1,NT)
      PREVT = TIME
      CALL ELTIME(2,8)
   66 RETURN
      END                                                     8445          END
```

Table C.5. Comparison of GUCVS with GMCVS (Page 34 of 41)

```
Unit 0: CVSB(2564,2879)                                                      Unit 1: SUSP:CIGS(9161,9474)
=  2564            SUBROUTINE PRINT(SUB)                                     9161          SUBROUTINE PRINT(SUB)                                     =
====================================================================  29 equal lines ==========================================================================
=  2594          *                 UNITL,UNITM,UNITT,GRAVTY(3)               9191       *              UNITL,UNITM,UNITT,GRAVTY(3)                   =
   2595            COMMON/RSAVE/XSG(3,20,3),DPMI(3,3,100),LPMI(1C            9192          COMMON/RSAVE/DPMI(3,3,100),LPMI(100)
              O),NSG(7),MSG(20,7)
=  2596            COMMON/TEMPVS/ YPR(3),T1(3),T2(3),HH(3),T3(3,3            9193          COMMON/TEMPVS/ YPR(3),T1(3),T2(3),HH(3),T3(3,3 =
=             )                                                                      )                                                               =
==================================================================== 276 equal lines ==========================================================================
=  2873            DO 20 J=1,3                                               9470          DO 20 J=1,3                                               =
   2874            F(3,J,I)=E1*F(3,J,I)+T1(J)+EB*X(J,I)                      9471          F(3,J,I)=E1*F(3,J,I)-T1(J)+EB*X(J,I)
   2875            F(4,J,I)=E1*F(4,J,I)+T2(J)+EC*X(J,I)                      9472          F(4,J,I)=E1*F(4,J,I)-T2(J)+EC*X(J,I)
   2876            Y(3,J,I)=E1*Y(1,J,I)+T3(J)-E2*X(J,I)                      9473          Y(3,J,I)=E1*Y(1,J,I)-T3(J)-E2*X(J,I)
   2877         20 Y(4,J,I)=E1*Y(2,J,I)+T4(J)-E2D*X(J,I)                     9474       20 Y(4,J,I)=E1*Y(2,J,I)-T4(J)-E2D*X(J,I)
   2878            RETURN
   2879            END
```

Table C.5. Comparison of GUCVS with GMCVS (Page 35 of 41)

```
 Unit 0: CVSB(2918,4698)                                        Unit 1: SUSP:CIGS(9544,11350)
= 2918         SUBROUTINE ROTATE                                  9544        SUBROUTINE ROTATE                                =
=====================================================  7 equal lines ==========================================================
= 2926         *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNT  9552        *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNT =
=        F,NPRT(36)                                                      F,NPRT(36)                                                =
  2927         COMMON/RSAVE/XSG(3,20,3),DPMI(3,3,100),LPMI(10     9553        COMMON/RSAVE/DPMI(3,3,100),LPMI(100)
         O),NSG(7),MSG(20,7)
= 2928         COMMON/DESCRP/ PHI(3,100),W(100),RW(100),SR(3,     9554        COMMON/DESCRP/ PHI(3,100),W(100),RW(100),SR(3,  =
=        200),HA(3,200),                                                 200),HA(3,200),                                           =
===================================================== 374 equal lines ==========================================================
= 3303         *            E(3,800),FF(5,800),GG(5,800),Y(5,8    9929        *            E(3,800),FF(5,800),GG(5,800),Y(5,8 =
=        OO),U(5,800),                                                   OO),U(5,800),                                             =
  3304         *            H,HPRINT,HS,TPRINT,TSTART,ICNT,IDB    9930        *            H,HPRINT,HS,TPRINT,TSTART,ICNT,IDB
         L,IFLAG,                                                        L,IFLAG
  3305         *            DMMY(200,25)
= 3306   C      NOTE: FF REPLACES F FROM SUBROUTINE DINT.         9931   C      NOTE: FF REPLACES F FROM SUBROUTINE DINT.      =
=====================================================  22 equal lines ==========================================================
= 3329   C 21                                                     9954   C 21                                                      =
  3330         COMMON/RSAVE/XSG(3,20,3),DPMI(3,3,100),LPMI(10     9955        COMMON/RSAVE/DPMI(3,3,100),LPMI(100)
         O),NSG(7),MSG(20,7)
  3331         DIMENSION RC21(450),IC21(177)                      9956        DIMENSION RC21(450),IC21(100)
  3332         EQUIVALENCE (RC21(1),XSG(1,1,1)) , (IC21(1),LP     9957   C *********   IC21 WAS DIMENSIONED 177 NOW 100, CHEC
         MI(1))                                                          K LOGIC BEFORE USE.
                                                                  9958        EQUIVALENCE (RC21(1),DPMI(1,1,1)) , (IC21(1),L
                                                                         PMI(1))
= 3333   C 22                                                     9959   C 22                                                      =
===================================================== 593 equal lines ==========================================================
= 3927   C                                                       10553   C                                                         =
  3928         DATA C21/ 8HXSG    ,8HDPMI    ,8HLPMI    ,8HN     10554        DATA C21/ 8H***    ,8HDPMI    ,8HLPMI    ,8H*
         SG    ,8HMSG    /                                               **    ,8H***    /
= 3929         DATA NC21/3,20,3  ,  3,3,30  ,  30,0,0  ,  7,0    10555        DATA NC21/3,20,3  ,  3,3,30  ,  30,0,0  ,  7,0 =
=        ,0  ,  20,7,0  /                                                ,0  ,  20,7,0  /                                          =
===================================================== 562 equal lines ==========================================================
= 4492         *                UNITL,UNITM,UNITT,GRAVTY(3)      11118        *                UNITL,UNITM,UNITT,GRAVTY(3)     =
                                                                 11119        COMMON /PLOTRS/ ELYPR(3,100), PLPTS(3,3,100),
                                                                         NELSEG(100), LDNPCS
                                                                 11120        1  , IELPS, IELPB, IELP, NNELP, NPLSEG(100)
                                                                 11121        REAL*4 PLPTS
= 4493         COMMON/TEMPVS/ P1(3),P2(3),P3(3),DE(3,3)          11122        COMMON/TEMPVS/ P1(3),P2(3),P3(3),DE(3,3)        =
=====================================================   1 equal line ===========================================================
= 4495         DATA IDYPR/3,2,1/                                 11124        DATA IDYPR/3,2,1/                               =
                                                                 11125        DO 6 I=1,100
                                                                 11126        NELSEG(I) = 0
                                                                 11127        NPLSEG(I) = 0
                                                                 11128        DO 4 J=1,3
                                                                 11129        ELYPR(J,I) = 0.DO
                                                                 11130        DO 2 K=1,3
                                                                 11131        PLPTS(J,K,I) = 0.
                                                                 11132     2  CONTINUE
                                                                 11133     4  CONTINUE
                                                                 11134     6  CONTINUE
= 4496   C                                                       11135   C                                                         =
=====================================================  58 equal lines ==========================================================
= 4555         DO 26 I =1,3                                      11194        DO 26 I =1,3                                    =
                                                                 11195        PLPTS(1,I,J) = P1(I)
                                                                 11196        PLPTS(2,I,J) = P2(I)
                                                                 11197        PLPTS(3,I,J) = P3(I)
= 4556         P2(I) = P2(I)-P1(I)                               11198        P2(I) = P2(I)-P1(I)                             =
=====================================================  38 equal lines ==========================================================
```

Table C.5. Comparison of GUCVS with GMCVS (Page 36 of 41)

```
=  4595        35 IF (NBAG.NE.O) CALL AIRBG1                             11237        35 IF (NBAG.NE.O) CALL AIRBG1                                 =
                                                                         11238           NNELP = NSEG
                                                                         11239           IELP = O
                                                                         11240           IELPB = O
                                                                         11241           IELPS = 101
=  4596           IF (NELP.LE.O) GO TO 51                                11242           IF (NELP.LE.O) GO TO 51                                    =
=======================================================        21 equal lines ========================================================
=  4618           BD(I+3,M) = P2(I)                                      11264           BD(I+3,M) = P2(I)                                          =
                                                                         11265           ELYPR(I,M) = P3(I)
=  4619           DO 45 J=1,3                                            11266           DO 45 J=1,3                                                =
=======================================================         7 equal lines ========================================================
=  4627        45 BD(K+9,M) = SUM2                                       11274        45 BD(K+9,M) = SUM2                                           =
                                                                         11275           IF (M .LE. NSEG) GO TO 50
                                                                         11276           IELPS = MINO(IELPS, M)
                                                                         11277           IELPB = MAXO(IELPB, M)
                                                                         11278           IELP = IELP + 1
=  4628        50 CONTINUE                                               11279        50 CONTINUE                                                   =
                                                                         11280           NNELP = NSEG + IELP
=  4629     C                                                            11281     C                                                                =
=======================================================        41 equal lines ========================================================
=  4671          * 5X,'SEGMENT',9X,'SEGMENT M',16X,'SEGMENT N',1         11323          * 5X,'SEGMENT',9X,'SEGMENT M',16X,'SEGMENT N',1 =
=          5X,                                                                    5X,                                                       =
   4672          * 'SPRING FORCE FUNCTION'12X,'DAMPING FORCE FUN         11324          * 'SPRING FORCE FUNCTION',12X,'DAMPING FORCE FU
           CTION'/                                                                NCTION'/
=  4673          * ' NO.  M   N',2(6X,'X',7X,'Y',7X,'Z',2X),7X,'         11325          * ' NO.  M   N',2(6X,'X',7X,'Y',7X,'Z',2X),7X,' =
=          DO',9X,'A1',11X,                                                       DO',9X,'A1',11X,                                          =
=======================================================        24 equal lines ========================================================
=  4698           END                                                    11350           END                                                       .=
```

Table C.5. Comparison of GUCVS with GMCVS (Page 37 of 41)

```
Unit 0: CVSB(5058)                                            Unit 1: SUSP:CIGS(11649)
=  5058        SUBROUTINE TRIGFS                              11649        SUBROUTINE TRIGFS                                 =
================================================= 3 equal lines =================================================================
=  5062          *          E(3,800), F(5,800),GG(5,800),Y(5,8  11653          *           E(3,800), F(5,800),GG(5,800),Y(5,8 =
=          00),U(5,800),                                              00),U(5,800),                                           =
   5063          *          H,HPRINT,HS,TPRINT,TSTART,ICNT,IDB  11654          *          H,HPRINT,HS,TPRINT,TSTART,ICNT,IDB
           L,IFLAG,                                                  L,IFLAG
   5064          *          DMMY(200,25)
=  5065        BETA = 0.0                                     11655        BETA = 0.0                                        =
================================================= 49 equal lines ================================================================
=  5115          *           NTPL( 5,100),NTBLT( 5,8),NTSEG( 5  11705          *           NTPL( 5,100),NTBLT( 5,8),NTSEG( 5 =
=          ,100)                                                     ,100)                                                  =
   5116        COMMON/RSAVE/XSG(3,20,3),DPMI(3,3,100),LPMI(10  11706        COMMON/RSAVE/DPMI(3,3,100),LPMI(100)
           0),NSG(7),MSG(20,7)
=  5117        COMMON/TEMPVS/ XD(3,3,100),XSEGLP(3,100),XPL(1  11707        COMMON/TEMPVS/ XD(3,3,100),XSEGLP(3,100),XPL(1  =
=          7,100),XBD(24,110),                                       7,100),XBD(24,110),                                    =
================================================= 627 equal lines ===============================================================
=  5745        SUBROUTINE VINPUT                              12335        SUBROUTINE VINPUT                                 =
   5746  C
               REV 20 05/07/80
=  5747  C     PERFORMS CARD INPUT AND COMPUTES DATA AND TABL   12336  C     PERFORMS CARD INPUT AND COMPUTES DATA AND TABL    =
=          ES REQUIRED BY                                            ES REQUIRED BY                                         =
================================================= 10 equal lines ================================================================
=  5758        COMMON/SGMNTS/ D(3,3,100),WMEG(3,100),WMEGD(3,  12347        COMMON/SGMNTS/ D(3,3,100),WMEG(3,100),WMEGD(3,  =
=          100),U1(3,100),                                           100),U1(3,100),                                        =
   5759        * U2(3,100), SEGLP(3,100),SEGLV(3,100),SEGLA(3,  12348        * U2(3,100),SEGLP(3,100),SEGLV(3,100),SEGLA(3,
           100),NSYM(100)                                            100),NSYM(100)
=  5760        COMMON/DESCRP/ PHI(3,100),W(100),RW(100),SR(3,  12349        COMMON/DESCRP/ PHI(3,100),W(100),RW(100),SR(3,  =
=          200),HA(3,200),                                           200),HA(3,200),                                        =
   5761        *HB(3,200),RPHI(3,100),HT(3,3,200),SPRING(5,300  12350        * HB(3,200),RPHI(3,100),HT(3,3,200),SPRING(5,3
           ),VISC(7,300),                                            00),VISC(7,300),
   5762          *         JNT(100),IPIN(100),ISING(100),IGLOB  12351        * JNT(100),IPIN(100),ISING(100),IGLOB(100),JOI
           (100),JOINTF(100)                                         NTF(100)
=  5763        COMMON/VPOSTN/ ZPLT(3),SPLT(3),AXV(3,6),VATAB(  12352        COMMON/VPOSTN/ ZPLT(3),SPLT(3),AXV(3,6),VATAB(  =
=          6,101,6),                                                 6,101,6),                                              =
   5764          *           VTO(6),VDT(6),TIMEV(6),OMEGV(6),N  12353          *            VTO(6),VDT(6),TIMEV(6),OMEGV(6)
           VTAB(6),INDXV(6)                                          ,NVTAB(6),INDXV(6)
=  5765        COMMON/TEMPVS/ XO(3),XDOTO(3),XCOMP(3),XVCOMP(  12354        COMMON/TEMPVS/ XO(3),XDOTO(3),XCOMP(3),XVCOMP(  =
=          3),ANGLE(3),                                              3),ANGLE(3),                                           =
   5766          *           ATAB(15,100),DVEH(3,3),VMEG(3),VM  12355          *            ATAB(15,101),DVEH(3,3),VMEG(3),
           EGD(3),                                                   VMEGD(3),
   5767          *          XACOMP(3),THET(3),AX(3),F(5,100),X  12356          *           XACOMP(3),THET(3),AX(3),F(5,101)
           YZ(102,6),TT(102),                                        ,XYZ(102,6),TT(102),
   5768          *           VIPS,VMPH,ATO,ADT,VTIME,OMEG,NATA  12357          *            VIPS,VMPH,ATO,ADT,VTIME,OMEG,NA
           B                                                         TAB,IDUM
                                                               12358          *            ,SP(5,101,4),Q1(101,4),A1(3),W1(
                                                                     4),QD(4),QC(4)
=  5769        COMMON/INTEST/ SGTEST(3,4,100),XTEST(3,400),SE  12359        COMMON/INTEST/ SGTEST(3,4,100),XTEST(3,400),SE  =
=          GT(400),REGT(400)                                         GT(400),REGT(400)                                      =
================================================= 1 equal line ==================================================================
=  5771        COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),       12361        COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),       =
   5772          *            UNITL,UNITM,UNITT,GRAVTY(3)      12362          *            UNITL,UNITM,UNITT,GRAVTY(3)
=  5773        COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),B  12363        COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),B  =
=          DYTTL(5),                                                 DYTTL(5),                                              =
   5774          *           BLTTTL(5,8),PLTTL(5,100),BAGTTL(5  12364          *            BLTTTL(5,8),PLTTL(5,100),BAGTTL
           ,6),SEG(100),                                             (5,6),SEG(100),
   5775          *           JOINT(100),CGS(100),JS(100)       12365          *            JOINT(100),CGS(100),JS(100)
=  5776        REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BA  12366        REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BA  =
=          GTTL,SEG,JOINT                                            GTTL,SEG,JOINT                                         =
```

Table C.5. Comparison of GUCVS with GMCVS (Page 38 of 41)

```
=====================================================  16 equal lines ====================================================
= 5793          WRITE (6,15) VPSTTL,ANGLE,VIPS,VTIME,XO,NATAB,   12383        WRITE (6,15) VPSTTL,ANGLE,VIPS,VTIME,XO,NATAB, =
=          ATO,ADT,MSEG                                                ATO,ADT,MSEG                                        =
  5794        15 FORMAT('1 VEHICLE DECELERATION INPUTS',91X,'CA   12384     15 FORMAT('1 VEHICLE DECELERATION INPUTS',91X,'CA
           RDS C'//3X,20A4//                                           RDS,C'//3X,20A4//
  5795          *  7X,'YAW',9X,'PITCH',7X,'ROLL',8X,'VIPS',8X,'   12385        *    7X,'YAW',9X,'PITCH',7X,'ROLL',8X,'VIPS',8X
           VTIME',7X,'XO(X)',                                          ,'VTIME',7X,'XO(X)',
  5796          *  7X,'XO(Y)',7X,'XO(Z)',2X,'NATAB',6X,'ATO',9X   12386        *    7X,'XO(Y)',7X,'XO(Z)',2X,'NATAB',6X,'ATO',
           ,'ADT',4X,'MSEG'/                                           9X,'ADT',4X,'MSEG'/
  5797          *  8F12.3,I5,2X,2F12.6,I5)                       12387        *    8F12.3,I5,2X,2F12.6,I5)
= 5798          DA1 = ANGLE(1)*RADIAN                            12388        DA1 = ANGLE(1)*RADIAN                              =
=====================================================  17 equal lines ====================================================
= 5816        17 FORMAT('O PASSENGER COMPARTMENT DISPLACEMENT H   12406     17 FORMAT('O PASSENGER COMPARTMENT DISPLACEMENT H =
=          ISTORY'/                                                    ISTORY'/                                            =
  5817          *       '  ANALYTICAL HALF-SINE WAVE DECELERATION  12407        *       '  ANALYTICAL HALF-SINE WAVE DECELERATI
           '/                                                          ON'/
  5818          *       '  VO=',F8.3,1X,A4,'/',A4,', OBLIQUE ANGL  12408        *       '  VO=',F8.3,1X,A4,'/',A4,', OBLIQUE AN
           ES =',3F7.2,                                                GLES =',3F7.2,
  5819          *       ' DEGREES, TIME DURATION =',F7.3,1X,A4//)  12409        *       ' DEGREES, TIME DURATION =',F7.3,1X,A4/
                                                                       /)
= 5820          GO TO 28                                         12410        GO TO 28                                           =
=====================================================  34 equal lines ====================================================
= 5855          ATAB(J,I  ) = ATAB(J,I-1) + DA2*(5.0*F1+8.0*F2   12445        ATAB(J,I  ) = ATAB(J,I-1) + DA2*(5.0*F1+8.0*F2 =
=          -F3)                                                        -F3)                                                =
  5856        21 ATAB(J,I+1) = ATAB(J,I-1) + DA1*(  F1+4.0*F2+F   12446     21 ATAB(J,I+1) = ATAB(J,I-1) + DA1*(    F1+4.0*F2
           3)                                                          +F3)
= 5857        22 UNITS = 1.0                                     12447     22 UNITS = 1.0                                        =
=====================================================   4 equal lines ====================================================
= 5862        23 FORMAT('O UNIDIRECTIONAL VEHICLE POSITION TABL   12452     23 FORMAT('O UNIDIRECTIONAL VEHICLE POSITION TABL =
=          ES'//                                                       ES'//                                               =
  5863          *   2('      TIME       ACC      VELOCITY      P   12453        *    2('      TIME        ACC        VELOCITY
           OSITION   ')/                                                POSITION   ')/
  5864          * 2('     (MSEC)       (G)     (',A4,'/',A4,')',5   12454        *    2('     (MSEC)       (G)     (',A4,'/',A4,')'
           X,'(',A4,')',4X)/)                                          ,5X,'(',A4,')',4X)/)
= 5865          DO 26 J=1,50                                     12455        DO 26 J=1,50                                       =
=====================================================  24 equal lines ====================================================
= 5890     C      READ LINEAR DECELERATION AND ANGULAR ACCELERAT  12480  C     READ LINEAR DECELERATION AND ANGULAR ACCELERAT =
=          ION TABLES                                                  ION TABLES                                          =
  5891     C      FROM CARDS C.4.                                 12481  C     FROM CARDS  C.4.
= 5892     C                                                      12482  C                                                  =
=====================================================  11 equal lines ====================================================
= 5904          WRITE  (6,36)  LTYPE,LFIT,NPTS,                  12494        WRITE  (6,36)  LTYPE,LFIT,NPTS,                    =
  5905          *              (TT(I),(XYZ(I,J),J=1,6),I=1,LPTS)  12495        *              (TT(I),(XYZ(I,J),J=1,6),I=1,LPTS
                                                                       )
= 5906        36 FORMAT ('O SPLINE FIT TABULAR INPUT'//           12496     36 FORMAT ('O SPLINE FIT TABULAR INPUT'//              =
  5907          *        3X,'LTYPE =',I6,'  LFIT =',I6,'  NPTS ='  12497        *        3X,'LTYPE =',I6,'  LFIT =',I6,'  NPTS
           ,I6//                                                       =',I6//
  5908          *        (F15.6,3X,3F12.3,3X,3F12.3))             12498        *        (F15.6,3X,3F12.3,3X,3F12.3))
= 5909          DO  37  I=1,3                                    12499        DO  37  I=1,3                                      =
  5910          I4 = 4-I
= 5911          XO(I) = XYZ(1,I)                                 12500        XO(I) = XYZ(1,I)                                   =
  5912          IF  (LTYPE.EQ.1)  GO TO 37
= 5913          XDOTO(I) = XYZ(2,I)                              12501        XDOTO(I) = XYZ(2,I)                                =
= 5914          VMEG(I) = XYZ(2,I+3)                             12502        VMEG(I) = XYZ(2,I+3)                               =
  5915        37 ANGLE(I4) = XYZ(1,I+3)                          12503     37 ANGLE(I) = XYZ(1,I+3)
  5916          DO  45  II=1,6                                   12504         IMJ = 6
                                                                 12505         IF(LTYPE.EQ.1)IMJ = 3
                                                                 12506        DO  45  II=1,IMJ
= 5917          CALL SPLINE (TT(LTYPE),XYZ(LTYPE,II),F,NPTS,LF   12507        CALL SPLINE (TT(LTYPE),XYZ(LTYPE,II),F,NPTS,LF =
=          IT)                                                         IT)                                                 =
```

Table C.5. Comparison of GUCVS with GMCVS (Page 39 of 41)

```
=====================================================   1 equal line  ========================================================
=  5919          IF  (II.GT.3)  I = II + 6                     12509          IF  (II.GT.3)  I = II + 6                          =
   5920          IF  (LTYPE.NE.1)  GO TO 38                    12510          IF(LTYPE.EQ.1) XDOTO(I) = F(3,1)
   5921          IF  (II.LE.3)  XDOTO(I) = F(3,1)              12511          UNITS = 1.0
   5922          IF  (II.GT.3)  VMEG(II-3) = F(3,1)
   5923          IF  (II.GT.3)  I = 16-II
   5924       38 UNITS = 1.0
=  5925          IF  (LTYPE.LT.3 .AND. II.LE.3)  UNITS = -1.0/G  12512        IF  (LTYPE.LT.3 .AND. II.LE.3)  UNITS = -1.0/G =
=====================================================   2 equal lines ========================================================
=  5928          TTT = ATO + DFLOAT(J-1)*ADT                   12515          TTT = ATO + DFLOAT(J-1)*ADT                        =
   5929          DO  39  K=K1,NPTS                             12516          DO  39  L=K1,NPTS
   5930          IF  (K.EQ.NPTS)  GO TO 40                     12517           K = L
   5931          IF  (DABS(TTT-F(1,K+1)).LT.EPS(8))  TTT = F(1,  12518         IF  (TTT.LT.F(1,L+1))  GO TO 40
       K+1)
   5932          IF  (TTT.LT.F(1,K+1))  GO TO 40
=  5933       39 CONTINUE                                      12519       39 CONTINUE                                           =
=====================================================   1 equal line  ========================================================
=  5935          DX = TTT - F(1,K)                             12521          DX = TTT - F(1,K)                                  =
   5936          GO TO (41,42,43),LTYPE                        12522          IF (LTYPE-2)  41,42,43
=  5937       41 ACC = 2.0*F(4,K) + 6.0*DX*F(5,K)              12523       41 ACC = 2.0*F(4,K) + 6.0*DX*F(5,K)                   =
=====================================================   6 equal lines ========================================================
=  5944          ISKIP = 1                                     12530          ISKIP = 1                                          =
                                                               12531           IF(LTYPE.NE.1)GO TO 46
                                                               12532      CODE FOR OMEGA ROUTINE: COMPUTE ATAB(I,J),I=10,11,12
                                                                           J = 1,MATAB
                                                               12533          DO 80 I = 1,NPTS
                                                               12534           DO 91 K = 1,3
                                                               12535       91  A1(K) = XYZ(I,K+3)
                                                               12536          CALL QUAT(A1,W1)
                                                               12537           DO 76 K = 1,4
                                                               12538       76 Q1(I,K) = W1(K)
                                                               12539          IF(I.EQ.1)GO TO 80
                                                               12540           TA = 0.0
                                                               12541           TB = 0.0
                                                               12542          DO 77 K = 1,4
                                                               12543          TA = TA + DABS(Q1(I,K) - Q1(I-1,K))
                                                               12544       77 TB = TB + DABS(Q1(I,K) + Q1(I-1,K))
                                                               12545           IF(TA.LE.TB)GO TO 80
                                                               12546           DO 78 K = 1,4
                                                               12547       78  Q1(I,K) = -Q1(I,K)
                                                               12548       80  CONTINUE
                                                               12549           DO 82 K = 1,4
                                                               12550       82  CALL SPLINE(TT,Q1(1,K),SP(1,1,K),NPTS,LFIT)
                                                               12551           DO 90 J = 1,MATAB
                                                               12552           TTT = ATO + DFLOAT(J-1)*ADT
                                                               12553           K1 = 1
                                                               12554           DO 83 L = K1,NPTS
                                                               12555           K = L
                                                               12556           IF(TTT.LT.SP(1,L+1,1))GO TO 84
                                                               12557     83    CONTINUE
                                                               12558       84  K1 = K
                                                               12559           DX = TTT - SP(1,K,1)
                                                               12560           DO 85 L = 1,4
                                                               12561           W1(L) = SP(2,K,L)+DX*(SP(3,K,L)+DX*(SP(4,K,L)
                                                                      +DX*SP(5,K,L)))
                                                               12562           QD(L) = 2.0*SP(4,K,L) + 6.0*DX*SP(5,K,L)
                                                               12563       85  IF(J.EQ.1)QC(L) = SP(3,K,L) + DX*(2.0*SP(4,K,
                                                                      L)+3.0*SP(5,K,L))
                                                               12564           CCC = 2.0/RADIAN
                                                               12565           IF(J.GT.1)GO TO 88
```

Table C.5. Comparison of GUCVS with GMCVS (Page 40 of 41)

```
                                                                       12566           CALL CROSS(QC(2),W1(2),A1)
                                                                       12567            DO 86 K = 1,3
                                                                       12568        86  VMEG(K) = CCC*(W1(1)*QC(K+1) - QC(1)*W1(K+1)
                                                                             + A1(K))
                                                                       12569            CALL DRCQUA(DVEH,W1)
                                                                       12570            CALL YPRDEG(DVEH,ANGLE)
                                                                       12571        88  CALL CROSS(QD(2),W1(2),QC(2))
                                                                       12572            DO 89 K = 2,4
                                                                       12573        89  ATAB(K+8,J) = CCC*(W1(1)*QD(K)-QD(1)*W1(K) +
                                                                             QC(K))
                                                                       12574        90 CONTINUE
=  5945        46 DO  55  J=1,MATAB                                    12575        46 DO  55  J=1,MATAB                                    =
=====================================================      6 equal lines =====================================================
=  5952        48 FORMAT(I1,'  VEHICLE LINEAR TIME HISTORY',3X,2      12582        48 FORMAT(I1,'  VEHICLE LINEAR TIME HISTORY',3X,2 =
=          OA4,3X,                                                           OA4,3X,                                             =
   5953           *          'PAGE NO.',I3//                           12583        *          'PAGE NO.',I3//
   5954           *     4X,'TIME',12X,'LINEAR DECELERATIONS (G''S      12584        *      4X,'TIME',12X,'LINEAR DECELERATIONS (G'
           )',                                                               'S)',
   5955           *               11X,'LINEAR VELOCITIES (',A4,'/      12585        *               11X,'LINEAR VELOCITIES (',A4,
           ',A4,')',                                                         '/',A4,')',
   5956           *               11X,'LINEAR DISPLACEMENTS (',A4      12586        *               11X,'LINEAR DISPLACEMENTS (',
           ,')'    /                                                         A4,')'    /
   5957           *     3X,'(MSEC)',3(11X,'X',11X,'Y',11X,'Z',3X      12587        *      3X,'(MSEC)',3(11X,'X',11X,'Y',11X,'Z',
           )     /)                                                          3X)     /  )
=  5958           ISKIP = 1                                            12588           ISKIP = 1                                          =
=====================================================      5 equal lines =====================================================
=  5964           ATAB(I+6,J) = XO(I)                                . 12594           ATAB(I+6,J) = XO(I)                                =
   5965           ATAB(I+12,J) = VMEG(I)                               12595        50 ATAB(I+12,J) = VMEG(I)
   5966        50 THET(I) = ANGLE(I)*RADIAN
=  5967           CALL DRCYPR (DVEH,ANGLE,IDYPR)                       12596           CALL DRCYPR (DVEH,ANGLE,IDYPR)                     =
=====================================================      9 equal lines =====================================================
=  5977        53 ATAB(I+6,J) = ATAB(I+6,J-1)                          12606        53 ATAB(I+6,J) = ATAB(I+6,J-1)                        =
   5978           *  +ADT*(ATAB(I+3,J-1)-G*ADT/6.0*(2.0*ATAB(I,J-      12607        *    +ADT*(ATAB(I+3,J-1)-G*ADT/6.0*(2.0*ATAB(I,
          1)+ATAB(I,J)))                                                     J-1)+ATAB(I,J)))
=  5979        54 T1 = (ATO + DFLOAT(J-1)*ADT)*1000.0                  12608        54 T1 = (ATO + DFLOAT(J-1)*ADT)*1000.0                =
=====================================================      9 equal lines =====================================================
=  5989        57 FORMAT('1 VEHICLE ANGULAR TIME HISTORY',3X,20A       12618        57 FORMAT('1 VEHICLE ANGULAR TIME HISTORY',3X,20A =
=         4,3X,'PAGE NO.',I3//                                               4,3X,'PAGE NO.',I3//                                =
   5990           *       4X,'TIME', 7X,'ANGULAR ACCELERATIONS (DE     12619        *       4X,'TIME', 7X,'ANGULAR ACCELERATIONS (
          G/',A4,'**2)',                                                     DEG/',A4,'**2)',
   5991           *                  7X,'ANGULAR VELOCITIES (DEG/'     12620        *                  7X,'ANGULAR VELOCITIES (DEG
          ,A4,')',                                                           /',A4,')',
   5992           *                 11X,'ANGULAR DISPLACEMENTS (DE     12621        *                 11X,'ANGULAR DISPLACEMENTS (
          G)' /                                                              DEG)' /
   5993           *     3X,'(MSEC)',2(11X,'X',11X,'Y',11X,'Z',3X)      12622        *     3X,'(MSEC)',2(11X,'X',11X,'Y',11X,'Z',3
          ,                                                                  X),
   5994           *        10X,'YAW',8X,'PITCH',8X,'ROLL' /)           12623        *        10X,'YAW',8X,'PITCH',8X,'ROLL' /)
=  5995        58 IF(J.EQ.1) GO TO 60                                  12624        58 IF(J.EQ.1) GO TO 60                                =
=====================================================     70 equal lines =====================================================
=  6066           NGRND = NVEH+1                                       12695           NGRND = NVEH+1                                     =
   6067           IF  (NGRND.GT.100 .OR. NVH.GT.6) STOP 7              12696           IF  (NGRND.GT.100 .OR. NVH.GT.6)  STOP 7
=  6068           SEG(NGRND) = GRND                                    12697           SEG(NGRND) = GRND                                  =
=====================================================    374 equal lines =====================================================
=  6443           END                                                  13072           END                                                =
```

Table C.5. Comparison of GUCVS with GMCVS (Page 41 of 41)

```
Listing of SUSP:CIGS(3783,3803) at 10:02:36 on AUG 29, 1985 for CCid=SS53                         Page    1
      SUBROUTINE DRCQUA(DC,Q)
COMPUTES DIRECTION COSINE MATRIX FROM QUATERNIONS
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION DC(3,3),Q(4)
      C = Q(1)**2 + Q(2)**2 + Q(3)**2 + Q(4)**2
      E = (Q(1) + Q(1) ) / C
      F = Q(1) * E - 1.DO
      DO 12 I = 1,3
      DO 10 J = 1,3
   10 DC(I,J) = 2.0*Q(I+1)*Q(J+1)  / C
   12 DC(I,I) = DC(I,I) + F
      DO 14 I = 1,3
      J = 1 + MOD(I,3)
      K = 1 + MOD(I+1,3)
      D = E*Q(I+1)
      DC(K,J) = DC(K,J) - D
   14 DC(J,K) = DC(J,K) + D
      DO 18 I =1,3
      DO 18 J = 1,3
   18 IF(DABS(DC(I,J)).GT.1.ODO)DC(I,J) = DSIGN(1.ODO,DC(I,J))
      RETURN
      END
```

Table C.6. Listings of Code Changes to GUCVS (Page 1 of 38)

```
Listing of SUSP:CIGS(8446,8492) at 10:03:18 on AUG 29, 1985 for CCid=SS53          Page    1

      SUBROUTINE OUTREC(ARRAY)
      DIMENSION ARRAY(2)
      COMMON /HLDCON/ LDNARY(3), NRNBAS(246), LFIRST(241), KASNUM,
     1        KATKOL(12), KPRELM(3), KATKAS(3,12), KASKTL(5,241), MAXPTS,
     2        LINES, KASE, KATG, NRN, LSTEP, LASREC(3)
C
C      1. FOR KATG = 1 TO 5:
C         KASE = 1 TO KATKAS(1,KATG)
C         ONLY NRNBAS(KATG) APPLIES
C     2. FOR KATG = 6 TO 12:
C         KASE = KATKAS(2,KATG) TO KATKAS(3,KATG)
C         NRNBAS(KASE+5) AND LFIRST(KASE) APPLY
C      3. KASADJ = NRNBAS INDEX FOR ALL CATEGORIES AND CASES
C
      LEN = KATKOL(KATG)
      IF (KATG .GT. 5) GO TO 10
      LINOFF = 0
      KASADJ = KATG
      LADJST = KASE * LSTEP - LSTEP
      KATABL = KATG
      GO TO 20
   10 LINOFF = LFIRST(KASE)
      KASADJ = KASE + 5
      LADJST = 0
      KATABL = KATG + 2
   20 IF (LINOFF .GE. 0) GO TO 40
      DO 30 I = 1, LEN
        IF (ARRAY(I) .EQ. 0.) GO TO 30
C       RECORD STARTING LINES OFFSET
        LINOFF = LINES - 1
        LFIRST(KASE) = LINOFF
        GO TO 40
   30 CONTINUE
      GO TO 60
   40 LINREL = LINES - LINOFF
      LMOD = MOD(LINREL,3)
      IF (LMOD .EQ. 0) LMOD = 3
      LDN = LDNARY(LMOD)
      LBASE = NRNBAS(KASADJ) + KPRELM(LMOD) + LADJST
      LINREC = (LINREL + 2) / 3
      IF (LINREC .EQ. 1) WRITE (LDN'LBASE) KATABL, KASE, LINES, LINOFF
   50 NRN = LBASE + LINREC
      WRITE (LDN'NRN) (ARRAY(I),I=1,LEN)
      LASREC(LMOD) = MAXO(LASREC(LMOD), NRN)
   60 RETURN
      END
      SUBROUTINE PANEL (DRR,ZR,JB)
```

Table C.6. Listings of Code Changes to GUCVS (Page 2 of 38)

```
      SUBROUTINE PLOTR1
      IMPLICIT REAL*8(A-H, O-Z)
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,
     *               NS,NQ,NSD,NFLX;NHRNSS,NWINDF,NJNTF,NPRT(36)
      COMMON/CNTSRF/ PL(17,100),BELT(20,8),TPTS(6,8),BD(24,110)
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),
     *               BLTTTL(5,8),PLTTL(5,100),BAGTTL(5,6),SEG(100),
     *               JOINT(100),CGS(100),JS(100)
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT
      LOGICAL*1 CGS,JS
      COMMON /PLOTRS/ ELYPR(3,100), PLPTS(3,3,100), NELSEG(100), LDNPCS
     1  , IELPS, IELPB, IELP, NNELP, NPLSEG(100)
      REAL*4 PLPTS
      REAL*4 SOFT(3,100), BOFT(3,6), BBELT(20,8), BBDD(3,100),
     1       ELD(9,100), BL
      COMMON /TEMPVS/ DE(9), SOFT, BOFT, BBELT, BBDD, ELD
      DIMENSION IDYPR(3)
      DATA IDYPR/3,2,1/
C
C----------------------------------------------------------------------
C
      DATA BL/4H    /
      LDNPCS = 9
C
C  BELT DATA
C
      DO 40 I=1, 8
      DO 39 J=1, 6
      BBELT(J, I) = BELT(J, I)
   39 CONTINUE
   40 CONTINUE
C
C  AND SIMILARLY FOR SEMI-MAJOR AXES, TO YIELD THE
C  PLOTCS ARRAY "BD"
C
      IF(NSEG.EQ.O) GOTO 55
      DO 52 I=1, NSEG
      DO 49 J = 1,3
      BBDD(J,I) = BD(J,I)
      SOFT(J,I) = BD(J+3,I)
      IF (NELSEG(I) .EQ. O) NELSEG(I) = I
 49   CONTINUE
      CALL DRCYPR(DE,ELYPR(1,I),IDYPR)
      DO 50 J=1,9
      ELD(J,I) = DE(J)
 50   CONTINUE
   52 CONTINUE
      IF (IELP .LE. O) GO TO 55
      II = NSEG
      DO 56 I = IELPS, IELPB
      IF (BD(1,I) .EQ. O.) GO TO 56
      IF (NELSEG(I) .EQ. O) GO TO 56
      II = II + 1
      NELSEG(II) = NELSEG(I)
      DO 53 J = 1,3
      BBDD(J,II) = BD(J,I)
      SOFT(J,II) = BD(J+3,I)
 53   CONTINUE
```

Table C.6. Listings of Code Changes to GUCVS (Page 3 of 38)

```
      CALL DRCYPR(DE,ELYPR(1,I),IDYPR)
      DO 54 J=1,9
      ELD(J,II) = DE(J)
 54   CONTINUE
 56   CONTINUE
C
C  FINALLY, BOFT -- THE BAG OFFSETS
C
   55 DO 90 I=1, NBAG
      II = NVEH + I
      DO 89 J=4, 6
      BOFT(J-3, I) = BD(J, II)
   89 CONTINUE
   90 CONTINUE
C
C     RECORD TIME INVARIANT PLOTCS VALUES
C
      WRITE (LDNPCS)  NSEG,NJNT,NBLT,NBAG,NPL,NNELP,NGRND,
     1 DATE, COMENT, VPSTTL,  BDYTTL,
     2 (SEG(N),N=1,NGRND), (CGS(J),J=1,NSEG)
      IF (NNELP .LE. 0) GO TO 9110
      WRITE (LDNPCS) ((BBDD(I,J),SOFT(I,J),ELD(I,J),
     1 ELD(I+3,J),ELD(I+6,J),I=1,3),NELSEG(J),J=1,NNELP)
 9110 IF (NJNT .LE. 0) GO TO 9120
      WRITE (LDNPCS) (JOINT(I),I=1,NJNT),(JS(J),J=1,NJNT)
 9120 IF (NBLT .LE. 0) GO TO 9130
      WRITE (LDNPCS) ((BLTTTL(I,J),I=1,3),(BBELT(K,J),
     1 K=1,6),J=1,NBLT)
 9130 IF (NBAG .LE. 0) GO TO 9140
      WRITE (LDNPCS) ((BAGTTL(I,J),I=1,5),(BOFT(K,J),
     1 K=1,3),J=1,NBAG)
 9140 IF (NPL .LE. 0) GO TO 9150
      WRITE (LDNPCS) ((PLTTL(I,J),I=1,5),J=1,NPL)
C
 9150 RETURN
      END
      SUBROUTINE PLOTR2
      IMPLICIT REAL*8(A-H, O-Z)
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,
     *                NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)
      COMMON/SGMNTS/ D(3,3,100),WMEG(3,100),WMEGD(3,100),U1(3,100),
     * U2(3,100), SEGLP(3,100),SEGLV(3,100),SEGLA(3,100),NSYM(100)
      COMMON/VPOSTN/ ZPLT(3),SPLT(3),AXV(3,6),VATAB(6,101,6),
     *             VTO(6),VDT(6),TIMEV(6),OMEGV(6),NVTAB(6),INDXV(6)
      COMMON/CNTSRF/ PL(17,100),BELT(20,8),TPTS(6,8),BD(24,110)
      COMMON /PLOTRS/ ELYPR(3,100), PLPTS(3,3,100), NELSEG(100), LDNPCS
     1 , IELPS, IELPB, IELP, NNELP, NPLSEG(100)
      REAL*4 PLPTS, TYME
      REAL*4 TTPTS(6,8), DD(3,3,100), SSEGLP(3,100), BBDD(3,8)
      REAL*4 PPLTS(4,3,100)
      COMMON /TEMPVS/ TTPTS, DD, SSEGLP, BBDD, PPLTS
C
C-----------------------------------------------------------------------
C
      TYME = TIME
      IF (NBLT .LE. 0) GO TO 25
      DO 20 I=1, NBLT
      DO 19 J=1, 6
```

Table C.6. Listings of Code Changes to GUCVS (Page 4 of 38)

```
      TTPTS(J, I) = TPTS(J, I)
   19 CONTINUE
   20 CONTINUE
C
 25   DO 30 I=1, NGRND
      DO 29 J=1, 3
      DO 28 K=1, 3
      DD(K, J, I) = D(K, J, I)
   28 CONTINUE
      SSEGLP(J, I) = SEGLP(J, I)
   29 CONTINUE
   30 CONTINUE
C
      IF (NBAG .LE. O) GO TO 50
      II = O
      IBAGA = NVEH + 1
      IBAGB = NVEH + NBAG
      DO 40 I= IBAGA, IBAGB
      DO 39 J=1, 3
      II = II + 1
      BBDD(J, II) = BD(J+3, I)
   39 CONTINUE
   40 CONTINUE
C
 50   IF (NPL .LE. O) GO TO 85
      DO 80 I=1, NPL
      DO 70 J=1, 3
      PPLTS(4, J, I) = PLPTS(1,J,I)
      PPLTS(1, J, I) = PLPTS(3,J,I)
      PPLTS(3, J, I) = PLPTS(2,J,I)
      PPLTS(2, J, I) = PLPTS(2,J,I) + PLPTS(3,J,I) - PLPTS(1,J,I)
   70 CONTINUE
      NN = NPLSEG(I)
      IF (NN .LE. O) NN = NVEH
      IF (NN .GE. NGRND) GO TO 80
      DO 75 IJ = 1, 4
      CALL PLOTRA(IJ,PPLTS(1,1,I),NN)
 75   CONTINUE
   80 CONTINUE
C
C      RECORD TIME VARIANT PLOTCS VALUES FOR CURRENT TIME
C
 85   WRITE (LDNPCS)  TYME,
     2 ((SSEGLP(I, J), I=1, 3), J=1, NGRND),
     4 (((DD(K, I, J), K=1, 3), I=1, 3), J=1, NGRND)
      IF (NPL .LE. O) GO TO 90
      WRITE (LDNPCS)(((PPLTS(I,J,K),I=1,4),J=1,3),K=1,NPL)
 90   IF (NBLT .LE. O) GO TO 100
      WRITE (LDNPCS) ((TTPTS(I, J), I=1, 6), J=1, NBLT)
 100  IF (NBAG .LE. O) GO TO 110
      WRITE (LDNPCS) ((BBDD(I, J), I=1, 3), J= 1, II)
C
 110  RETURN
      END
      SUBROUTINE PLOTRA(I, XYZ, N)
      IMPLICIT REAL*8(A-H, O-Z)
      REAL*4 XYZ
      COMMON/SGMNTS/ D(3,3,100),WMEG(3,100),WMEGD(3,100),U1(3,100),
```

Table C.6. Listings of Code Changes to GUCVS (Page 5 of 38)

```
Listing of SUSP:CIGS(8853,9039) at 10:04:07 on AUG 29, 1985 for CCid=SS53                                   Page    4
 9027           * U2(3,100), SEGLP(3,100),SEGLV(3,100),SEGLA(3,100),NSYM(100)
 9028             DIMENSION XYZ(4,3), X(3), T1(3)
 9029             II = I
 9030             NN = N
 9031             X(1) = XYZ(II,1)
 9032             X(2) = XYZ(II,2)
 9033             X(3) = XYZ(II,3)
 9034             CALL DOT31(D(1,1,NN), X(1), T1(1))
 9035             DO 10 J=1, 3
 9036             XYZ(II,J) = T1(J) + SEGLP(J,NN)
 9037        10   CONTINUE
 9038             RETURN
 9039             END
```

Table C.6. Listings of Code Changes to GUCVS (Page 6 of 38)

```
Listing of SUSP:CIGS(9477,9505) at 10:04:40 on AUG 29, 1985 for CCid=SS53                                  Page    1
      SUBROUTINE QUAT(ANG,Q)
COMPUTES QUATERNIONS FROM YAW, PITCH, ROLL ANGLES IN DEGREES
      IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION ANG(3),Q(4),R(4),T(3)
      COMMON/CNSNTS/PI,RADIAN
      A = 0.5*ANG(1)*RADIAN
      Q(1) = DCOS(A)
      Q(2) = 0.0
      Q(3) = 0.0
      Q(4) = DSIN(A)
      K = 3
      DO 10 I = 2,3
      A = 0.5*ANG(I)*RADIAN
      R(1) = DCOS(A)
      R(2) = 0.0
      R(3) = 0.0
      R(4) = 0.0
      R(K) = DSIN(A)
      DOT = Q(2)*R(2) + Q(3)*R(3) + Q(4)*R(4)
      CALL CROSS(Q(2),R(2),T)
      DO 5 J = 2,4
    5 Q(J) = Q(1)*R(J) + R(1)*Q(J) + T(J-1)
      Q(1) = Q(1)*R(1) - DOT
   10 K = 2
      SUM = DSQRT(Q(1)**2 + Q(2)**2 + Q(3)**2 + Q(4)**2)
      DO 12 I = 1,4
   12  Q(I) = Q(I)/SUM
       RETURN
      END
```

Table C.6. Listings of Code Changes to GUCVS (Page 7 of 38)

```
Listing of SUSP:CIGS(11620,11648) at 10:05:10 on AUG 29, 1985 for CCid=SS53                      Page     1
 11620          SUBROUTINE STASH (ARRAY,NUMWRD,LASNRN)
 11621          COMMON /IOCNTL/ LDNWRK, KWDPLN, IOTALK
 11622          DIMENSION ARRAY(2)
 11623          NLIN = (NUMWRD + KWDPLN - 1) / KWDPLN - 1
 11624          NLAS = NUMWRD - NLIN * KWDPLN
 11625          IF (IOTALK .NE. 0) WRITE(6,9998) NUMWRD,  NLIN, NLAS,
 11626         1     LASNRN, KWDPLN
 11627     9998 FORMAT (8HOSTASH :, 5I10)
 11628          IPTB = 0
 11629          NRN = LASNRN
 11630          IF (NLIN .LE. 0) GO TO 20
 11631          DO 10 I = 1, NLIN
 11632          IPTA = IPTB + 1
 11633          IPTB = IPTB + KWDPLN
 11634          NRN = NRN + 1
 11635          WRITE(LDNWRK'NRN) (ARRAY(J), J=IPTA,IPTB)
 11636          IF (IOTALK .GT. 1) WRITE(6,9999) NRN, IPTA, IPTB, (ARRAY(J),
 11637         1     J=IPTA,IPTB)
 11638     10   CONTINUE
 11639     20   IPTA = IPTB + 1
 11640          IPTB = IPTB + NLAS
 11641          NRN = NRN + 1
 11642          WRITE(LDNWRK'NRN) (ARRAY(L), L=IPTA,IPTB)
 11643          IF (IOTALK .GT. 1) WRITE(6,9999) NRN, IPTA, IPTB, (ARRAY(J),
 11644         1     J=IPTA,IPTB)
 11645     9999 FORMAT (1HO,3I5/ (5X,6Z20))
 11646          LASNRN = NRN
 11647          RETURN
 11648          END
```

Table C.6. Listings of Code Changes to GUCVS (Page 8 of 38)

```
Listing of SUSP:CUTS(1,584) at 10:06:02 on AUG 29, 1985 for CCid=SS53          Page    1

      COMMON /PLOTIT/ NYP(20), MX(3,20), MY(3,10,20), NX(20), XO(20),
     1       XN(20), XS(20), NY(20), YO(20), YN(20), YL(20), XL(20),
     2       YS(20), XLAB(15,20), NXLAB(20), NYLAB(20), YLAB(15,20),
     3       NYPLT, NPLB1(20), PLB1(15,20), NPLSMB(10,20)
      COMMON /HOLDIT/ BAGTTL(5,6), BDYTTL(5), BLTTTL(5,8), COMENT(40),
     1       DATE(3), DT, JOINT(100), INCSML, INCBIG, PLTTL(5,100),
     2       SEG(100), VPSTTL(20), LLFRST, KASTOP, NRNTIM,
     3       TIMLAS, MULTPL, NPL, NPLT, NRNVAR, LFENKT,
     4       NBAG, NBLT, LBBAG, LBBELT, LBJNT, LBPL,
     5       LBSEG, KKNTRL(241), NSD, NSEG, ISPSWT,
     6       NSTEPS, NVEH, NGRND, NHRNSS, NJNT
      COMMON /PRESET/ ISYMB(49), KASLIN(3,16), LDNARY(3), LPP, NW60
     1        , MAXKAP
      COMMON /FETCHT/ D(3,3,12), DPMI(3,3,100), G, LPMI(100), PI,
     1       SEGLA(3,12), SEGLP(3,12), SEGLV(3,12), WMEG(3,12),
     2       WMEGD(3,12), UNITL, UNITT, UNITM, RADIAN, TIMMAX,
     3       IKASE(12), NKASE, KASKOL, KASPAG, KASFUL
      COMMON /FETCHU/ PQUANT(16), KONTLP(6,3), JDTPTS(18), IPIN(3),
     1      KKAS, MSECPT, MTIMPT
      COMMON /HLDCON/ NRNBAS(246), LFIRST(241), KASNUM, KATKOL(12),
     1       KPRELM(3), KATKAS(3,12), KASKTL(5,241), MAXPTS,
     2       NRN, LSTEP, LASREC(3)
      COMMON /ZTTHNG/ TIME(6000), ZTTH(6000,11), LINEQT(6000)
      COMMON /IOCNTL/ LDNWRK, KWDPLN, IOTALK
      COMMON /AVDCON/ RSTF(27), FDPMI(900), XSG(3,50,3), NSG(7),
     1      MSG(50,7), MMSG(50,6)
      DIMENSION IORDER(16), NEEDK(16), NTS(16), KIK(2,7),
     1       CONTLP(6,3), KSTF(27), FKASKL(1205),
     2       FFIRST(241), FRNBAS(246), FENKAT(51), FLPMI(100)
      EQUIVALENCE (KSTF(1),RSTF(1)), (FFIRST(1),LFIRST(1)),
     1             (FRNBAS(1),NRNBAS(1)), (FENKAT(1),KATKOL(1)),
     2             (FKASKL(1),KASKTL(1,1)), (FLPMI(1),LPMI(1)),
     3             (CONTLP(1,1),KONTLP(1,1))
      REAL*4 JOINT
C
C         INITIALIZATION FOR OUT PROCESSOR
C
      DO 10 I = 1, 16
        IORDER(I) = 0
        NEEDK(I) = 0
        NTS(I) = 0
   10 CONTINUE
      DO 12 I = 1, 7
      KIK(1,I) = 0
      KIK(2,I) = 0
 12   CONTINUE
      MSECPT = 0
      MTIMPT = 0
      LBBELT = 0
      LBBAG = 0
      LBJNT = 0
      LBSEG = 0
      LBPL = 0
      LLFRST = 0
      IOTALK = 0
      LDNWRK = LDNARY(1)
C
C        READ FIRST CONTROL DATA CARD FOR OUT PROCESSOR
```

Table C.6. Listings of Code Changes to GUCVS (Page 9 of 38)

```
C
      READ (5,15) BEGT, FINT, TINC, IOTALK
   15 FORMAT (3E8.0, I5)
C
C       READ TIME INVARIANT INFORMATION FROM HOLD FILE
C
      READ (LDNWRK'1) TIMLAS, LINES, LASREC, MULTPL, INCBIG, INCSML
      READ (LDNWRK'2) NVEH, NGRND, NSEG, NPL, NJNT, NSD, NBAG, NBLT,
     1     NHRNSS, MAXPTS, LSTEP, NSTEPS, KASNUM, NRNVAR, LBBELT, LBBAG,
     2     LBJNT, LBSEG, LBPL, LLFRST
      READ (LDNWRK'3) DATE, BDYTTL, KASTOP, LFENKT, STPMIN, PI, G,
     1     RADIAN, UNITL, UNITT, UNITM, DT, TIMMAX, ISPSWT
      LAST = NRNVAR
      CALL PICKUP(VPSTTL, 20, LAST)
      CALL PICKUP(COMENT, 40, LAST)
      IF (NBLT .LE. 0) GO TO 20
      LAST = LBBELT
      CALL PICKUP(BLTTTL, 5*NBLT, LAST)
   20 IF (NBAG .LE. 0) GO TO 30
      LAST = LBBAG
      CALL PICKUP(BAGTTL, 5*NBAG, LAST)
   30 IF (NJNT .LE. 0) GO TO 40
      LAST = LBJNT
      CALL PICKUP(JOINT, NJNT, LAST)
   40 IF (NSEG .LE. 0) GO TO 50
      LAST = LBSEG
      CALL PICKUP(SEG, NGRND, LAST)
      CALL PICKUP(FLPMI, NSEG, LAST)
      LK = 9 * NSEG
      CALL PICKUP(DPMI, LK, LAST)
C         COMPLETE SEGMENT REPORTING AXIS ORIENTATIONS
 50   DO 58 I = 1, NGRND
      IF (I .GT. NSEG) GO TO 52
      IF (LPMI(I) .NE. 0) GO TO 58
 52   DO 56 J = 1, 3
      DO 54 K = 1, 3
      DPMI(K, J, I) = 0.
 54   CONTINUE
 56   CONTINUE
      DPMI(1, 1, I) = 1.
      DPMI(2, 2, I) = 1.
      DPMI(3, 3, I) = 1.
      LPMI(I) = 0
 58   CONTINUE
      IF (NPL .LE. 0) GO TO 60
      LAST = LBPL
      CALL PICKUP(PLTTL, 5*NPL, LAST)
   60 LAST = LLFRST
      CALL PICKUP(FFIRST, KASNUM, LAST)
      CALL PICKUP(FRNBAS, KASTOP, LAST)
      CALL PICKUP(FKASKL, 5*KASNUM, LAST)
      CALL PICKUP(FENKAT, LFENKT, LAST)
C
C       READ USER CONTROL DATA FOR THIS RUN OF THE OUT PROCESSOR
C
      KBEG = BEGT / STPMIN + .5
      KINC = TINC / STPMIN + .5
      KFIN = FINT / STPMIN + .5
```

Table C.6. Listings of Code Changes to GUCVS (Page 10 of 38)

```
      KSTP = DT / STPMIN + .5
      KSADJ = MAXO(KSTP - 1,O)
C       GET ALL POINTS IF TINC < DT
      IF (KINC .LT. KSTP) KINC = O
C       MAKE TINC MULTIPLE OF DT
      KINC = KSTP * ((KINC + KSADJ)/KSTP)
C       BEGT MUST BE POSITIVE
      IF (KBEG .LT. O) KBEG = O
      KBEG = KSTP * ((KBEG + KSADJ)/KSTP)
C       IF FINT TOO SMALL, SET FOR END OF ALL POINTS
      IF (KFIN .LE. O) KFIN = MULTPL
      KMINSP = MAXO(KINC,KSTP)
      IF (KFIN .LE. KBEG) KFIN = KBEG + KMINSP
      IF (KFIN .GT. MULTPL) KFIN = MULTPL
C  NOW REDEFINE KFIN TO BE A MULTIPLE OF KMINSP
      KFIN = KBEG + ((KFIN-KBEG+MAXO(KMINSP-1,O)) / KMINSP) * KMINSP
      READ (5,90) IORDER
   90 FORMAT (16I4)
      KATDO = O
      DO 100 I = 1, 16
        N = IORDER(I)
        IF (N .EQ. O) GO TO 102
        IF (N .LT. O) GO TO 100
        IF (N .GT. 16) GO TO 100
        KATDO = KATDO + 1
        NEEDK(N) = I
  100 CONTINUE
 102  IF (KATDO .GT. O) GO TO 110
      DO 104 I = 1, 16
      IORDER(I) = I
      NEEDK(I) = I
 104  CONTINUE
      KATDO = 16
C
C       READ CASE CONTROLS FOR PRINT CATEGORIES 8 TO 14
C          THESE ARE ALL CONTACT FORCE RELATED
C
  110 READ (5,120) NTS
  120 FORMAT (16I4)
      NKNTRL = O
      DO 130 I = 1, 16
        N = NTS(I)
        IF (N .LE. O) GO TO 135
        IF (N .GT. KASNUM) GO TO 130
        KATG = KASKTL(1,N)
        IF (NEEDK(KATG) .EQ. O) GO TO 130
      IF (NKNTRL .GE. 241) GO TO 135
        NKNTRL = NKNTRL + 1
        KKNTRL(NKNTRL) = N
        IF (NEEDK(KATG) .GT. O) NEEDK(KATG) = -NEEDK(KATG)
        IF (NKNTRL .EQ. 241) GO TO 135
  130   CONTINUE
        GO TO 110
C
C     WRITE PROLOGUE ON PRIMARY OUTPUT UNIT.
C
 135    WRITE (6,140)
        WRITE (6,150)
```

Table C.6. Listings of Code Changes to GUCVS (Page 11 to 38)

```
  140   FORMAT ('1', 21X, '*********************************************
     1*', '***************************************', /, 22X,
     2          '*******************************************',
     3          '******************************************')
  150   FORMAT (3(22X,'******',73X,'******',/), 22X, '******', 9X,
     1          'GGGGGGG', 6X, 'UU   UU', 6X, 'CCCCCCC', 6X, 'VV   VV',
     2          6X, 'SSSSSSS', 5X, '******', /, 22X, '******', 9X,
     3          'GG    G', 6X, 'UU   UU', 6X, 'CCCCCCC', 6X, 'VV   VV',
     4          6X, 'SSSSSSS', 5X, '******', /, 22X, '******', 9X,
     5          'GG     ', 6X, 'UU   UU', 6X, 'CC     ', 6X, ' VV VV ',
     6          6X, 'SS     ', 5X, '******', /, 22X, '******', 9X,
     7          'GG  GGG', 6X, 'UU   UU', 6X, 'CC     ', 6X, ' VV VV ',
     8          6X, 'SSSSSSS', 5X, '******', /, 22X, '******', 9X,
     9          'GG   GG', 6X, 'UU   UU', 6X, 'CC     ', 6X, '  VVV  ',
     *          6X, '     SS', 5X, '******', /, 22X, '******', 9X,
     1          'GG   GG', 6X, 'UUUUUUU', 6X, 'CCCCCCC', 6X, '  VVV  ',
     2          6X, 'SSSSSSS', 5X, '******', /, 22X, '******', 9X,
     3          'GGGGG G', 6X, 'UUUUUUU', 6X, 'CCCCCCC', 6X, '   V   ',
     4          6X, 'SSSSSSS', 5X, '******')
        WRITE (6,160) DATE, COMENT
  160   FORMAT (3(22X,'******',73X,'******',/), 22X, '******', 17X,
     1          ' GMR/UMTRI OUT PROCESSOR ', 2X, 3A4, 17X, '******', /,
     2          22X, '******', 73X, '******', /, 2(22X,
     3
     4'*****************************************************************',
     5          '***********************',/), ///, 1X, 20A4, /, 1X, 20A4)
        BEGT = KBEG * STPMIN
        TINC = KINC * STPMIN
        FINT = KFIN * STPMIN
        IF (TINC .EQ. 0.) GO TO 190
        WRITE (6,180) BEGT, FINT, TINC
  180   FORMAT ('0', 10X, 'PRINT QUANTITIES BEGINNING AT TIME=', F10.3,
     1          ' THROUGH TIME =', F10.3, ' AT EVERY INCREMENT OF ',
     2          F10.3)
        GO TO 210
  190   WRITE (6,200) BEGT, FINT
  200   FORMAT ('0', 10X, 'PRINT QUANTITIES FOR ALL RECORDED TIMES',
     1          ' BEGINNING AT TIME=', F10.3, ' THROUGH TIME=', F10.3)
  210   WRITE (6,220) (I,I=1,16), (IORDER(J),J=1,16), (NEEDK(K),K=1,16)
  220   FORMAT ('0', 10X, 'CATEGORIES IN ORDER AND ORDER FOR CATEGORIES'
     1          , ' TO BE PRINTED'/11X, 'POSITION/CATEGORY', 7X,
     2          16I5/11X, 'CATEGORY FOR POSITION', 3X, 16I5/11X,
     3          'POSITION FOR CATEGORY', 3X, 16I5)
        IF (NKNTRL .LE. 0) GO TO 340
        WRITE (6,230)
  230   FORMAT (35X, '(POSITIVE INDICATES ALL CASES FOR CATEGORY',
     1          ' PRINTED IN ASSIGNED ORDER;'/35X,'NEGATIVE THAT CASES',
     2      ' AND ORDER HAVE BEEN SPECIFIED FOR CATEGORY.)'/
     3          '1 ', 42X, 'INDIVIDUALLY SPECIFIED CONTACT CASES'/
     4          10X, 'NUMBER', 10X, 'CASE NO.', 10X, 'CATEGORY NO.',
     5          10X, 'CASE OF CATG', 7X, 'N', 9X, 'NN', 10X, 'M',
     6          9X, 'MM')
C
C     SORT SPECIFIED CASES IN PRINT ORDER BY CATEGORIES
C     KEEPING PRESCRIBED ORDER WITHIN CATEGORIES
C
        NSAM = 0
        IPNT = 1
```

Table C.6. Listings of Code Changes to GUCVS (Page 12 to 38)

```
  240   JPNT = IPNT + 1
        IKAS = KKNTRL(IPNT)
        IKAT = KASKTL(1,IKAS)
        IORD = IABS(NEEDK(IKAT))
  250   JKAS = KKNTRL(JPNT)
        JKAT = KASKTL(1,JKAS)
        JORD = IABS(NEEDK(JKAT))
        IF (IORD - JORD) 300, 270, 260
  260   NSAM = O
        GO TO 280
  270   NSAM = NSAM + 1
  280   KOPA = IPNT + NSAM
        KOPB = JPNT - 1
        KTOP = KOPA + KOPB
        IF (KOPA .GT. KOPB) GO TO 300
        DO 290 JJ = KOPA, KOPB
          J = KTOP - JJ
          KKNTRL(J + 1) = KKNTRL(J)
  290   CONTINUE
      KKNTRL(KOPA) = JKAS
        IF (IORD .EQ. JORD) GO TO 300
        IKAS = JKAS
        IKAT = JKAT
        IORD = JORD
      NSAM = O
  300   JPNT = JPNT + 1
        IF (JPNT .LE. NKNTRL) GO TO 250
        IPNT = IPNT + NSAM + 1
        NSAM = O
        IF (IPNT .LT. NKNTRL) GO TO 240
C
C       NOW PRINT SPECIFIED CASES AS REORDERED
C
        KNUM = O
        KLAS = O
        DO 330 I = 1, NKNTRL
          II = I
          IKAS = KKNTRL(I)
          IKAT = KASKTL(1,IKAS)
          IORD = IABS(NEEDK(IKAT))
          IF (IKAT .EQ. KLAS) GO TO 310
      LKIK = IKAT - 7
      IF (LKIK .GT. O  .AND.  LKIK .LT. 8) GO TO 306
      WRITE(6,303) LKIK, II, IKAT, IKAS, IORD
 303  FORMAT ('OWARNING---ILLEGAL LKIK=', 5I6)
      GO TO 330
 306  KIK(1, LKIK) = II
          KLAS = IKAT
          KNUM = O
  310     KNUM = KNUM + 1
      KIK(2, LKIK) = II
          WRITE (6,320) II, IKAS, IKAT, KNUM, (KASKTL(J,IKAS),J=2,5)
  320     FORMAT (8X, I6, 11X, I6, 13X, I6, 17X, I6, 7X, I6, 5X, I6,
     1        5X, I6, 5X, I6)
  330   CONTINUE
      WRITE(6,335)
 335  FORMAT(1HO)
C
```

Table C.6. Listings of Code Changes to GUCVS (Page 13 to 38)

```
C      READ PRINTED OUTPUT CONTROLS    (H CARDS)
C
  340   DO 370 K = 1, 3
          IF (NEEDK(K) .EQ. 0) GO TO 370
          READ (5,350,ERR=430,END=450) KSG, MMSG(1,K), MSG(1,K),
     1    (XSG(I,1,K),I=1,3)
  350     FORMAT (I6, 2I3, 3F12.6)
      WRITE(6,353) K, KSG
 353  FORMAT('0CATEGORY', I3, ' CONTROLS:', I5, ' CASES')
          IF (KSG .GT. 50) GO TO 410
          NSG(K) = KSG
      IF (KSG-1) 352, 363, 355
 352  NEEDK(K) = 0
      GO TO 370
 355      READ (5,360,ERR=430,END=450) (MMSG(J,K),MSG(J,K),(XSG(I,J,K),
     1    I=1,3),J=2,KSG)
  360     FORMAT (6X, 2I3, 3F12.6)
 363  WRITE(6,365) (J,MMSG(J,K),MSG(J,K),(XSG(I,J,K),I=1,3),J=1,KSG)
 365  FORMAT (26X, 3I5, 3G20.10)
  370   CONTINUE
        DO 400 K = 4, 6
          IF (NEEDK(K) .EQ. 0) GO TO 400
          READ (5,380,ERR=430,END=450) KSG, (MMSG(J,K),MSG(J,K),J=1,11)
  380     FORMAT (I6, 22I3)
      WRITE(6,353) K, KSG
          IF (KSG .GT. 50) GO TO 410
          NSG(K) = KSG
      IF (KSG .GT. 0) GO TO 385
      NEEDK(K) = 0
      GO TO 400
 385      IF (KSG .LT. 12) GO TO 393
          READ (5,390,ERR=430,END=450) (MMSG(J,K),MSG(J,K),J=12,KSG)
  390     FORMAT (6X, 22I3)
 393  WRITE(6,395) (J, MMSG(J,K), MSG(J,K), J = 1, KSG)
 395  FORMAT (26X, 3I5)
  400   CONTINUE
      IF (NEEDK(7) .EQ. 0) GO TO 480
      READ (5,402,ERR=430,END=450) KSG, (MSG(J,7), J=1, 11)
 402  FORMAT (12I6)
      K = 7
      WRITE (6,353) K, KSG
      IF (KSG .GT. 50) GO TO 410
      NSG(7) = KSG
      IF (KSG .GT. 0) GO TO 404
      NEEDK(7) = 0
      GO TO 480
 404  IF (KSG .LT. 12) GO TO 408
      READ (5,406,ERR=430,END=450) (MSG(J,7),J=12, KSG)
 406  FORMAT (6X, 11I6)
 408  WRITE (6,409) (J, MSG(J,7), J=1, KSG)
 409  FORMAT (26X, I5, 5X, I5)
        GO TO 480
  410   WRITE (6,420)
  420   FORMAT ('0FATAL ERROR---MORE THAN 50 PRINT REQUESTS SPECIFIED.')
        GO TO 470
  430   WRITE (6,440) K, J
  440   FORMAT ('0FATAL ERROR---READ ERROR FOR K, J=', 2I6)
        GO TO 470
```

Table C.6. Listings of Code Changes to GUCVS (Page 14 to 38)

```
  450   WRITE (6,460) K, J
  460   FORMAT ('0FATAL ERROR---EOF FOR K, J=', 2I6)
  470   CALL ERROR
C
C     READ INPUT CARD H.8.A TO CONTROL COMPUTATION OF HIC, HSI & CSI.
C
  480   IF (NEEDK(15) .EQ. 0) GO TO 510
        READ (5,520) JDTPTS
        WRITE (6,490) JDTPTS
  490   FORMAT ('0CATEGORY 15 CONTROLS:', 9(2I4, 2X))
        NDPT = 0
        DO 500 KDT = 1, 18
  500   IF (JDTPTS(KDT) .NE. 0) NDPT = NDPT + 1
      IF (NDPT .EQ. 0) NEEDK(15) = 0
C
C     READ INDICES OF VARIABLES TO BE PLOTTED AND
C     ARGUMENTS TO SUBROUTINE SLPLOT ON CARDS I.
C
C         INPUT CARD I.1
C
  510   IF (NEEDK(16) .EQ. 0) GO TO 560
        READ (5,520) NPLT, (NYP(K),K=1,NPLT)
  520   FORMAT (18I4)
        WRITE (6,523) NPLT
 523  FORMAT ('0CATEGORY 16 CONTROLS:', I5, ' CASES')
      IF (NPLT .GT. 0) GO TO 525
      NEEDK(16) = 0
      GO TO 560
 525    DO 550 K = 1, NPLT
          NYPLT = NYP(K)
      WRITE(6,528) K, NYPLT
 528  FORMAT(10X, 'FOR PLOT NO.', I3, ', NO. OF CURVES = ', I3)
C
C         INPUT CARD I.2.K
C
          READ (5,529) MX(1,K), MX(2,K), MX(3,K), (MY(1,J,K),MY(2,J,K),
     1    MY(3,J,K), NPLSMB(J,K), J = 1, NYPLT)
 529  FORMAT (19I3/ (9X, 16I3))
          WRITE (6,529) MX(1,K), MX(2,K), MX(3,K), (MY(1,J,K),MY(2,J,K),
     1    MY(3,J,K), NPLSMB(J,K), J = 1, NYPLT)
C
C         INPUT CARD I.3.K
C
          READ (5,530) NX(K), XO(K), XN(K), XL(K), XS(K)
  530     FORMAT (I4, 4X, 4F8.0)
C
C         INPUT CARD I.4.K
C
          READ (5,530) NY(K), YO(K), YN(K), YL(K), YS(K)
C
C         INPUT CARD I.5.K
C
          READ (5,540) NXLAB(K), (XLAB(I,K),I=1,NW60)
  540     FORMAT (I4, 4X, 15A4)
C
C     NOTE - ABOVE FORMAT ASSUMES 4 ALPHANUMERIC CHARACTERS FOR SINGLE
C     PRECISION WORDS ON IBM 360 AND 370 COMPUTERS. THE 15A4 TERM IN THE
C     FORMAT WILL HAVE TO BE CHANGED ON NON-IBM COMPUTERS TO PRODUCE A
```

Table C.6. Listings of Code Changes to GUCVS (Pag 15 to 38)

```
C     CONTINUOUS STRING OF 60 CHARACTERS IN CORE MEMORY.
C
C         INPUT CARD I.6.K
C
          READ (5,540) NYLAB(K), (YLAB(I,K),I=1,NW60)
C
C         INPUT CARD I.7.K
C
          READ (5,540) NPLB1(K), (PLB1(I,K),I=1,NW60)
  550   CONTINUE
  560   IF (LINES .LE. 0) GO TO 740
C
C       READ IN SELECTED TIMES FOR COMMON REFERENCE
C
        KATG = 1
        KASE = 1
        KPT = 0
        KWANT = KBEG
        LASTM = -1
        DO 590 I = 1, LINES
          CALL INREAD(KATG, KASE, I, RSTF(1))
          KMUL = KSTF(3)
          IF (KINC .LE. 0) GO TO 580
          IF (KMUL - LASTM) 590, 585, 565
 565      IF (KWANT .GT. KMUL) GO TO 590
          IF (KWANT .NE. KMUL) WRITE (6,570) KWANT, KMUL
  570     FORMAT ('0WARNING---EXPECTED NEXT TIME MISSING=', I6,
     1            ' REPLACED BY ', I6)
  580     KPT = KPT + 1
          TIME(KPT) = RSTF(1)
 585      LINEQT(KPT) = KSTF(2)
          IF (KWANT .EQ. KFIN) GO TO 600
          LASTM = KMUL
          KWANT = MIN0(KWANT + KINC,KFIN)
  590   CONTINUE
C
C        PRINT DESIRED CATEGORIES
C
  600   IF (KATDO .LE. 0) GO TO 800
C
C      LOOP OVER DESIRED CATEGORIES
C
      DO 780 IA = 1, KATDO
C      GET GENERAL CATEGORY CONTROLS
      NKAT = IORDER(IA)
      IF (NEEDK(NKAT) .EQ. 0) GO TO 780
C  ZERO PRINTING CONTROLS FOR THIS CATEGORY TO INITIALIZE.
      DO 604 L = 1, 3
      DO 602 M = 1, 6
      KONTLP(M,L) = 0
 602  CONTINUE
 604  CONTINUE
      KASKOL = KASLIN(1,NKAT)
      KASPAG = KASLIN(2,NKAT)
      KASFUL = KASLIN(3,NKAT)
C         GET RANGE OF CASE DRIVER INDICES FOR CURRENT CATEGORY
      IF (NKAT .GT. 7) GO TO 610
      KASTAR = 1
```

Table C.6. Listings of Code Changes to GUCVS (Page 16 of 38)

```
      KASTOP = NSG(NKAT)
      GO TO 660
  610 IF (NKAT - 15) 620, 640, 650
  620 IF (NEEDK(NKAT) .LT. 0) GO TO 630
      ITS = NKAT - 2
      IF (KATKAS(1,ITS) .LE. 0) GO TO 780
      KASTAR = KATKAS(2,ITS)
      KASTOP = KATKAS(3,ITS)
      GO TO 660
  630 ITS = NKAT - 7
      KASTAR = KIK(1, ITS)
      KASTOP = KIK(2, ITS)
      GO TO 660
  640 KASTAR = 1
      KASTOP = NDPT
      GO TO 660
  650 KASTAR = 1
      KASTOP = NPLT
  660 IF (KASTAR .GT. KASTOP) GO TO 780
C
C        LOOP OVER DESIRED CASES WITHIN CURRENT CATEGORY:
C               ONE PAGE AT A TIME
C
C        SET UP FETCH CONTROLS FOR CURRENT PAGE
C
      DO 770 IB = KASTAR, KASTOP, KASPAG
      KETCHA = IB
      KETCHB = MINO(IB + KASPAG - 1, KASTOP)
      KKAS = 0
      DO 730 KKI = KETCHA, KETCHB
      KK = KKI
      KKAS = KKAS + 1
      KONTLP(6,KKAS) = KKI
      IF (NKAT .GT. 7) GO TO 680
      MG = 0
      IF (NKAT .EQ. 7) GO TO 668
      MG = MMSG(KK, NKAT)
      IF (MG) 662, 666, 668
  662 IF (MG .EQ. -2) GO TO 664
      MG = NGRND
      GO TO 668
  664 MG = NVEH
      GO TO 668
  666 MG = NVEH
      IF (NKAT .EQ. 1  .OR.  NKAT .EQ. 4) MG = IABS(MSG(KK, NKAT))
  668 KONTLP(5,KKAS) = MG
      KONTLP(4,KKAS) = MSG(KK, NKAT)
      IF (NKAT .GT. 3) GO TO 670
      CONTLP(1,KKAS) = XSG(1, KK, NKAT)
      CONTLP(2,KKAS) = XSG(2, KK, NKAT)
      CONTLP(3,KKAS) = XSG(3, KK, NKAT)
      GO TO 725
  670 KONTLP(1,KKAS) = 0
      KONTLP(2,KKAS) = 0
      KONTLP(3,KKAS) = 0
      GO TO 725
  680 IF (NKAT - 15) 690, 710, 720
  690 IF (NEEDK(NKAT) .GT. 0) GO TO 700
```

Table C.6. Listings of Code Changes to GUCVS (Page 17 of 38)

```
      KK = KKNTRL(KKI)
      KONTLP(6, KKAS) = KK
  700 KONTLP(1,KKAS) = KASKTL(2,KK)
      KONTLP(2,KKAS) = KASKTL(3,KK)
      KONTLP(3,KKAS) = KASKTL(4,KK)
      KONTLP(4,KKAS) = KASKTL(5,KK)
      KONTLP(5,KKAS) = 0
      GO TO 725
  710 CALL HICCSI(KK, KPT)
      GO TO 725
  720 CALL SLPLOT(KK, KPT)
  725 IF (IOTALK .EQ. 0) GO TO 730
      WRITE (6,728) NKAT, KETCHA, KK, KKAS, (KONTLP(I,KKAS),I=1,6),
     1    (CONTLP(J,KKAS),J=1,6)
  728 FORMAT ('0KONTLP:', 10I10/ 8X, 6G20.10)
  730 CONTINUE
      IF (KKAS .LE. 0) GO TO 770
      IF (NKAT .GT. 14) GO TO 770
C
C       SET UP TIME LOOP FOR CURRENT PAGE
C
      NTIM = 0
      LINPPG = 0
      NTITL = 1
  740 NTIM = NTIM + 1
      DO 745 I = 1, 12
      IKASE(I) = 0
  745 CONTINUE
      NKASE = 0
      KOLBEG = 1 - KASKOL
      DO 750 II = 1, KKAS
      KOLBEG = KOLBEG + KASKOL
      CALL FETCH(NKAT, II, NTIM, KOLBEG)
  750 CONTINUE
  760 IF (NTITL .NE. 0) CALL HEDING(NKAT, NTIM, LINPPG)
      NTITL = 0
      LINPPG = LINPPG + 1
      CALL PRTLIN(NKAT)
      IF (NTIM .GE. KPT) GO TO 770
      IF (LINPPG .LT. LPP) GO TO 740
      NTITL = 1
      GO TO 760
  770 CONTINUE
  780 CONTINUE
C
C     INSERT ANY PLOT TERMINATION CODE REQUIRED BY YOUR SYSTEM HERE.
C
  800   RETURN
        END
      BLOCK DATA
      COMMON /PRESET/ ISYMB(49), KASLIN(3,16), LDNARY(3), LPP, NW60
     1       , MAXKAP
      COMMON /IOCNTL/ LDNWRK, KWDPLN, IOTALK
      DATA ISYMB /1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,
     11HL,1HM,1HN,1HO,1HP,1HQ,1HR,1HS,1HT,1HU,1HV,1HW,
     21HX,1HY,1HZ,1H<,1H(,1H+,1H|,1H&,1H$,1H),1H;,1H¬,1H-,
     31H/,1H,,1H%,1H_,1H>,1H?,1H:,1H#,1H@,1H',1H=,1H",1H*/
      DATA KASLIN/4,3,12,4,3,12,4,3,12,4,3,12,4,3,12,4,3,12,7,2,14,
```

Table C.6. Listings of Code Changes to GUCVS (Page 18 of 38)

```
     1   7,2,14,10,1,10,4,2,8,4,2,8,4,2,8,6,2,12,6,2,12,0,1,0,0,1,0/
      DATA LPP, NW60, MAXKAP, KWDPLN / 41, 15, 6000, 20/
      DATA LDNARY / 8, 10, 11/
      END
```

Table C.6. Listings of Code Changes to GUCVS (Page 19 of 38)

```
      SUBROUTINE POSTPR(PRDT)
C                                                            REV 20 05/18/80
C     CONTROLS GENERATION OF PRINTED TABULAR TIME HISTORIES
C     AND PLOTS BY THE VALUE OF NPRT(4) AS FOLLOWS:
C
C      VALUE OF       TIME
C      NPRT(4)     HISTORIES   PLOTS
C
C        +4           **        NO
C        +3          YES         YES
C        +2          YES          NO
C        +1           **       YES
C         0           **        NO
C        -1           NO         YES
C        -2          YES          NO
C        -3          YES         YES
C
C      **TIME HISTORIES WERE PRINTED BY SUBROUTINE OUTPUT.
C
      COMMON/CDINT/JDTPTS(18) , Z(6000,25)
C     NOTE: THIS OVERWRITES COMMON /CDINT/.
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,
     *              NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)
      REAL*8 TIME
      COMMON/FORCES/ PSF(7,100),BSF(4,20),SSF(10,20),BAGSF(3,20),
     *              PRJNT(7,100),NPANEL(5),NPSF,NBSF,NSSF,NBGSF
      REAL*8 PSF;BSF,SSF,BAGSF,PRJNT
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),
     *              BLTTTL(5,8),PLTTL(5,100),BAGTTL(5,6),SEG(100),
     *              JOINT(100),CGS(100),JS(100)
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT
      LOGICAL*1 CGS,JS
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),
     *              UNITL,UNITM,UNITT,GRAVTY(3)
      REAL*8 PI,RADIAN,G,THIRD,EPS,UNITL,UNITM,UNITT,GRAVTY
      COMMON/JBARTZ/ MNPL( 100),MNBLT( 8),MNSEG( 100),MNBAG(  6),
     *             MPL(3,5,100),MBLT(3,5,8),MSEG(3,5,100),MBAG(3,10,6),
     *             NTPL( 5,100),NTBLT( 5,8),NTSEG( 5,100)
      COMMON/DAMPER/ APSDM(3,20),APSDN(3,20),ASD(5,20),MSDM(20),MSDN(20)
      REAL*8 APSDM,APSDN,ASD
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),
     *              XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),
     *              NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)
      REAL*8 BAR,BB,BBDOT,PLOSS,XLONG,HTIME
      COMMON/RSAVE/XSG(3,20,3),DPMI(3,3,100),LPMI(100),NSG(7),MSG(20,7)
      REAL*8 XSG,DPMI,TDATA,UMSEC,PRDT,TEST
C     NOTE: SUBROUTINES POSTPR & HEDING SHARE THIS COMMON/TEMPVS/.
C     THE FIRST DIMENSION OF XLAB,YLAB,PLB1 AND PLB2 SHOULD BE THE SAME
C     AS THE VALUE ASSIGNED TO NW60 WHICH IS THE NUMBER OF WORDS THAT
C     IS NECESSARY TO CONTAIN 60 CONSECUTIVE CHARACTERS DEPENDING ON THE
C     COMPUTER SYSTEM THIS PROGRAM IS OPERATING ON. THE VALUE OF NW60
C     SHOULD BE 15 ON IBM 360 AND 370, 10 ON UNIVAC 1108, 6 ON CDC 6600.
C     THE LAST TERM IN FORMAT 13 BELOW SHOULD BE 15A4(IBM), 10A6(UNIVAC)
C     OR 6A10(CDC). ALSO, THE FIRST DIMENSION OF PLDATA IN SUBROUTINE
C     HEDING SHOULD BE 97(IBM), 77(UNIVAC) OR 53(CDC).
C
      COMMON/TEMPVS/TDATA(14,300),HEDATA(220),
     *             XO(20),XN(20),XL(20),XS(20),XLAB(15,20),PLB1(15,20),
```

Table C.6. Listings of Code Changes to GUCVS (Page 20 of 38)

```
     *              YO(20),YN(20),YL(20),YS(20),YLAB(15,20),PLB2(15,20),
     *              NYP(20),MX(2,20),MY(2,10,20),NX(20),NY(20),
     *              NXLAB(20),NYLAB(20),NPLB1(20),NPLB2(20),
     *              USEC(45) , ZTTH(14,45,2)
      COMMON /IPLSMB/NPLSMB(100)
      LOGICAL LTABH,LPLOT
      DATA  LPP/45/ , NZD1/6000/
      DATA NW60/15/
      LTABH = .FALSE.
      LPLOT = .FALSE.
      NPRT4 = IABS(NPRT(4))
      IF  (NPRT4.EQ.1)  LPLOT = .TRUE.
      IF  (NPRT4.EQ.3)  LPLOT = .TRUE.
      IF  (NPRT4.EQ.2)  LTABH = .TRUE.
      IF  (NPRT4.EQ.3)  LTABH = .TRUE.
      DO 2 KKK =1,100
    2 NPLSMB(KKK)=0
C
C     READ INPUT CARD H.8.A TO CONTROL COMPUTATION OF HIC, HSI & CSI.
C
      READ  (5,11)  JDTPTS
      WRITE (6,91) JDTPTS
   91 FORMAT(1X,18I4)
      NDPT = 0
      DO 4 KDT =1,18
    4 IF  (JDTPTS(KDT).NE.0)  NDPT = NDPT + 1
      IF  (.NOT.LPLOT .AND. .NOT.LTABH .AND. NDPT.EQ.0) GO TO 99
      CALL ELTIME (1,36)
      IF  (.NOT.LPLOT)  GO TO 20
C
C     READ INDICES OF VARIABLES TO BE PLOTTED AND
C     ARGUMENTS TO SUBROUTINE SLPLOT ON CARDS I.
C
C         INPUT CARD I.1
C
      READ  (5,11)  NPLT , (NYP(K),K=1,NPLT)
      WRITE (6,91) NPLT,(NYP(K),K=1,NPLT)
   11 FORMAT (18I4)
      IF  (NPLT.LE.0)  LPLOT = .FALSE.
      IF  (.NOT.LPLOT)  GO TO 20
      DO  15  K=1,NPLT
      NYPLT = NYP(K)
C
C         INPUT CARD I.2.K
C
      READ  (5,11)  MX(1,K), MX(2,K), (MY(1,J,K), MY(2,J,K),NPLSMB(J)
     1,J=1,NYPLT)
      WRITE (6,11) MX(1,K),MX(2,K),   (MY(1,J,K),MY(2,J,K), NPLSMB(J)
     1,J=1,NYPLT)
C
C         INPUT CARD I.3.K
C
      READ  (5,12)  NX(K), XO(K), XN(K), XL(K), XS(K)
   12 FORMAT (I4 , 4X , 4F8.0 )
C
C         INPUT CARD I.4.K
C
      READ  (5,12)  NY(K), YO(K), YN(K), YL(K), YS(K)
```

Table C.6. Listings of Code Changes to GUCVS (Page 21 of 38)

```
C
C         INPUT CARD I.5.K
C
      READ  (5,13)  NXLAB(K), (XLAB(I,K),I=1,NW60)
   13 FORMAT (I4 , 4X , 15A4)
C
C     NOTE - ABOVE FORMAT ASSUMES 4 ALPHANUMERIC CHARACTERS FOR SINGLE
C     PRECISION WORDS ON IBM 360 AND 370 COMPUTERS. THE 15A4 TERM IN THE
C     FORMAT WILL HAVE TO BE CHANGED ON NON-IBM COMPUTERS TO PRODUCE A
C     CONTINUOUS STRING OF 60 CHARACTERS IN CORE MEMORY.
C
C         INPUT CARD I.6.K
C
      READ  (5,13)  NYLAB(K), (YLAB(I,K),I=1,NW60)
C
C         INPUT CARD I.7.K
C
   15 READ  (5,13)  NPLB1(K), (PLB1(I,K),I=1,NW60)
C
C
C     READ TIME HISTORY DATA FROM TAPE 8.
C
   20 NPTS = 0
      LINES = 0
      IF  (NPRT(4).GT.0)  REWIND 8
      READ  (8,END=29)  NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,NPANEL,
     *                  MNPL,MNBLT,MNSEG,MNBAG,MPL,MBLT,MSEG,MBAG
      READ  (8,END=29)  DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,
     *                  SEG,JOINT,UNITL,UNITM,UNITT,NSG,MSG,XSG,
     *                  NHRNSS,NBLTPH,NPTSPB,NSD,MSDM,MSDN
   21 READ  (8,END=29)  NT, UMSEC, ((TDATA(I,J),I=1,14),J=1,NT)
      NPTS = NPTS + 1
      Z(NPTS,1) = UMSEC
      IF  (NDPT.EQ.0)  GO TO 22
C
C     STORE DATA FOR HIC, HSI AND CSI.
C
      JJ = 1
      DO  61  I=1,18
      IF  (JDTPTS(I).EQ.0)  GO TO 61
      JJ = JJ +1
      JD = JDTPTS(I) - 1
      JE = 4*MOD(JD,3) + 4
      JP = JD/3 + 1
      Z(NPTS,JJ) = TDATA(JE,JP)
   61 CONTINUE
      JHIC= JJ
   22 IF  (.NOT.LPLOT)  GO TO 25
C
C     STORE DATA FOR PLOTTING
C
      JY = NDPT + 1
      DO  24  K=1,NPLT
      JE = IABS(MX(2,K))
      IF  (JE.EQ.0)  GO TO 23
      JY = JY + 1
      JP = MX(1,K) - 20
      Z(NPTS,JY) = TDATA(JE,JP)
```

Table C.6. Listings of Code Changes to GUCVS (Page 22 of 38)

```
   23 NYPLT = NYP(K)
      DO  24  J=1,NYPLT
      JY = JY + 1
      JP = MY(1,J,K) - 20
      JE = IABS(MY(2,J,K))
      Z(NPTS,JY) = UMSEC
   24 IF  (JE.NE.0)  Z(NPTS,JY) = TDATA(JE,JP)
   25 IF  (.NOT.LTABH)  GO TO 21
C
C     STORE DATA TO PRINT TABULAR TIME HISTORIES
C
      TEST = DMOD(UMSEC,PRDT)
      TEST = DMIN1(TEST,DABS(PRDT-TEST))
      IF  (NPRT(26).EQ.0 .AND. TEST.GT.EPS(4))  GO TO 21
      LINES = LINES + 1
      NTTH = MOD(LINES-1,LPP) + 1
      USEC(NTTH) = UMSEC
      DO  26  J=1,NT
      DO  26  I=1,14
   26 ZTTH(I,NTTH,J) = TDATA(I,J)
      IF  (NTTH.EQ.LPP)  CALL HEDING (LINES,LPP)
      GO TO 21
   29 IF  (.NOT.LTABH .OR. LINES.EQ.0)  GO TO 30
      IF  (NTTH.NE.LPP)  CALL HEDING (LINES,LPP)
   30 IF  (NDPT.NE.0)  CALL HICCSI(NPTS,JHIC)
      IF  (.NOT.LPLOT)  GO TO 98
C
C     PLOT DATA VIA SUBROUTINE SLPLOT.
C
C
      CALL COMPRS
      JZ = NDPT+1
      DO  50  K=1,NPLT
      JX = 1
      IF  (MX(2,K).EQ.0)  GO TO 42
      JZ = JZ + 1
      JX = JZ
      IF  (Z(1,JX).EQ.0.0 .OR. MX(2,K).GE.0)  GO TO 42
      DO  41  I=2,NPTS
   41 Z(I,JX) = Z(I,JX) - Z(1,JX)
      Z(1,JX) = 0.0
   42 NYPLT = NYP(K)
      DO  44  J=1,NYPLT
      JY = JZ + J
      IF  (Z(1,JY).EQ.0.0 .OR. MY(2,J,K).GE.0)  GO TO 44
      DO  43  I=2,NPTS
   43 Z(I,JY) = Z(I,JY) - Z(1,JY)
      Z(1,JY) = 0.0
   44 CONTINUE
      NXK = NX(K)
      NYK = NY(K)
      XOK = XO(K)
      YOK = YO(K)
      XNK = XN(K)
      YNK = YN(K)
      XLK = XL(K)
      YLK = YL(K)
      XSK = XS(K)
```

Table C.6. Listings of Code Changes to GUCVS (Page 23 of 38)

```
      YSK = YS(K)
      NXLABK = NXLAB(K)
      NYLABK = NYLAB(K)
      NPLB1K = NPLB1(K)
      NPLB2K = NPLB2(K)
      CALL SLPLOT(Z(1,JX  ), NXK, XOK, XNK, XLK, XSK, XLAB(1,K), NXLABK,
     *          Z(1,JZ+1), NYK, YOK, YNK, YLK, YSK, YLAB(1,K), NYLABK,
     *           NPTS,NYPLT,NZD1,PLB1(1,K),NPLB1K,PLB2(1,K),NPLB2K,K)
C
C     INSERT ANY CODE REQUIRED BY YOUR SYSTEM TO ADVANCE PLOT PAGES HERE
C
   50 JZ = JZ + NYPLT
      CALL DONEPL
C
C     INSERT ANY PLOT TERMINATION CODE REQUIRED BY YOUR SYSTEM HERE.
C
   98 CALL ELTIME (2,36)
   99 RETURN
      END
```

Table C.6. Listings of Code Changes to GUCVS (Page 24 of 38)

```
      SUBROUTINE FETCH(KATG, KASPT, NLIN, KOLB)
      COMMON /HOLDIT/ BAGTTL(5,6), BDYTTL(5), BLTTTL(5,8), COMENT(40),
     1     DATE(3), DT, JOINT(100), INCSML, INCBIG, PLTTL(5,100),
     2     SEG(100), VPSTTL(20), LLFRST, KASTOP, NRNTIM,
     3     TIMLAS, MULTPL, NPL, NPLT, NRNVAR, LFENKT,
     4     NBAG, NBLT, LBBAG, LBBELT, LBJNT, LBPL,
     5     LBSEG, KKNTRL(241), NSD, NSEG, ISPSWT,
     6     NSTEPS, NVEH, NGRND, NHRNSS, NJNT
      COMMON /PRESET/ ISYMB(49), KASLIN(3,16), LDNARY(3), LPP, NW60
     1    , MAXKAP
      COMMON /FETCHT/ D(3,3,12), DPMI(3,3,100), G, LPMI(100), PI,
     1     SEGLA(3,12), SEGLP(3,12), SEGLV(3,12), WMEG(3,12),
     2     WMEGD(3,12), UNITL, UNITT, UNITM, RADIAN, TIMMAX,
     3     IKASE(12), NKASE, KASKOL, KASPAG, KASFUL
      COMMON /IOCNTL/ LDNWRK, KWDPLN, IOTALK
      COMMON /FETCHU/ PQUANT(16), KONTLP(6,3), JDTPTS(18), IPIN(3),
     1    KKAS, MSECPT, MTIMPT
      COMMON /HLDCON/ NRNBAS(246), LFIRST(241), KASNUM, KATKOL(12),
     1     KPRELM(3), KATKAS(3,12), KASKTL(5,241), MAXPTS,
     2     NRN, LSTEP, LASREC(3)
      COMMON /ZTTHNG/ TIME(6000), ZTTH(6000,11), LINEQT(6000)
      DIMENSION CONTLP(6,3), V(3,12), FM(3,3,3), ISWT(2), LSWT(2)
     1    , KQUANT(14)
      EQUIVALENCE (CONTLP(1,1), KONTLP(1,1)), (KQUANT(1),PQUANT(1))
      DATA TWOPI/6.283185/
      LIN = LINEQT(NLIN)
      KPT = KASPT
      KAT = KATG
      KOL = KOLB
      MSECPT = NLIN
      MTIMPT = LIN
      KAS = KONTLP(6, KPT)
      IF (KAT .LE. 0) GO TO 900
      IF (KAT - 7) 5, 700, 800
 5    ITO = KONTLP(5, KPT)
      ISWT(2) = ITO
      IFRM = IABS(KONTLP(4, KPT))
      ISWT(1) = IFRM
      DO 90 J = 1, 2
      II = J
      LSWT(II) = 0
      DO 10 I = 1, 12
      IF (IKASE(I) .LE. 0) GO TO 20
      IF (IKASE(I) .NE. ISWT(II)) GO TO 10
      LSWT(II) = I
      GO TO 90
 10   CONTINUE
 20   JJ = NKASE + 1
      IF (NKASE .LT. 12) GO TO 40
      DO 30 I = 1, 12
      IF (IKASE(I) .EQ. NVEH) GO TO 30
      IF (IKASE(I) .EQ. NGRND) GO TO 30
      JJ = I
      GO TO 40
 30   CONTINUE
      WRITE(6,9999) IKASE, NKASE, ISWT, LSWT, LIN, KPT, KAT, KOL
 9999 FORMAT('0FATAL ERROR---FETCH CURRENT CASE SPACE ',
     1   'FILLED WITH VECH/GRND.'/(5X, 12I10))
```

Table C.6. Listings of Code Changes to GUCVS (Page 25 of 38)

```
      STOP 99
   40 IF (NKASE .LT. 12) NKASE = JJ
      IKASE(JJ) = ISWT(II)
      LSWT(II) = JJ
C  RECORDING CATEGORY 2
      CALL INREAD(2, ISWT(J), LIN, ZTTH(1,1))
      DO 60 M = 1, 3
      SEGLP(M, JJ) = ZTTH(M,1)
      SEGLV(M, JJ) = ZTTH(M+3,1)
      SEGLA(M, JJ) = ZTTH(M+6,1)
   60 CONTINUE
C  RECORDING CATEGORY 3
      CALL INREAD(3,ISWT(J),LIN,ZTTH(1,1))
      DO 70 M = 1, 3
      WMEG(M, JJ) = ZTTH(M,1)
      WMEGD(M, JJ) = ZTTH(M+3,1)
   70 CONTINUE
C  RECORDING CATREGORY 4
      CALL INREAD(4, ISWT(J), LIN, D(1,1,JJ))
   90 CONTINUE
      JFRM = LSWT(1)
      JTO  = LSWT(2)
      IF (IOTALK .EQ. 0) GO TO 95
      WRITE(6,93) KAT, KPT, LIN, NLIN, KOL, ISWT, LSWT
   93 FORMAT ('OFETCH:', 11I10)
   95 IF (KAT .LT. 4) CALL MAT31(DPMI(1,1,IFRM), CONTLP(1,KPT), V(1,12))
      GO TO (100, 200, 300, 400, 500, 600), KAT
C
C     LINEAR ACCELERATION OF POINTS ATTACHED TO SEGMENTS
C
  100 CALL CROSS(WMEG(1,JFRM),V(1,12), V(1,1))
      CALL CROSS(WMEG(1,JFRM), V(1,1), V(1,2))
      CALL CROSS(WMEGD(1,JFRM), V(1,12), V(1,3))
      CALL VECADD(V(1,2), V(1,3), V(1,4))
      CALL DOT31(D(1,1,JFRM), V(1,4), V(1,5))
      CALL VECADD(V(1,5), SEGLA(1,JFRM), V(1,6))
      IF (KONTLP(4,KPT) .GT. 0) GO TO 110
      CALL DOT31(D(1,1,JFRM), V(1,12), V(1,11))
      CALL VECADD(SEGLP(1,JFRM), V(1,11), V(1,10))
      CALL VECSUB(V(1,10), SEGLP(1,JTO), V(1,9))
      CALL MAT31(D(1,1,JTO), V(1,9), V(1,2))
      CALL DOT31(D(1,1,JFRM), V(1,1), V(1,3))
      CALL VECADD(V(1,3), SEGLV(1,JFRM), V(1,4))
      CALL VECSUB(V(1,4), SEGLV(1,JTO), V(1,5))
      CALL CROSS(WMEG(1,JTO), V(1,2), V(1,3))
      CALL VECSUB(V(1,5), V(1,3), V(1,1))
      CALL VECSUB(V(1,6), SEGLA(1,JTO), V(1,7))
      CALL MAT31(D(1,1,JTO), V(1,7), V(1,8))
      CALL CROSS(WMEG(1,JTO), V(1,1), V(1,3))
      CALL SCALED(V(1,3), .5, V(1,1))
      CALL VECSUB(V(1,8), V(1,1), V(1,6))
      CALL CROSS(WMEG(1,JTO), V(1,2), V(1,3))
      CALL CROSS(WMEG(1,JTO), V(1,3), V(1,4))
      CALL CROSS(WMEGD(1,JTO), V(1,2), V(1,1))
      CALL VECADD(V(1,1), V(1,4), V(1,5))
      CALL VECSUB(V(1,6), V(1,5), V(1,7))
      GO TO 120
  110 CALL MAT31(D(1,1,JTO), V(1,6), V(1,7))
```

Table C.6. Listings of Code Changes to GUCVS (Page 26 of 38)

```
 120  CALL DOT31(DPMI(1,1,ITO), V(1,7), V(1,8))
      CALL SCALED(V(1,8), G, PQUANT(KOL))
      GO TO 550
C
C     LINEAR VELOCITIES OF POINTS ATTACHED TO SEGMENTS
C
 200  CALL CROSS(WMEG(1,JFRM), V(1,12), V(1,1))
      CALL DOT31(D(1,1,JFRM), V(1,1), V(1,2))
      CALL VECADD(V(1,2), SEGLV(1,JFRM), V(1,3))
      CALL VECSUB(V(1,3), SEGLV(1,JTO), V(1,4))
      CALL MAT31(D(1,1,JTO), V(1,4), V(1,5))
      LL = 5
      IF (KONTLP(4,KPT) .GT. 0) GO TO 210
      CALL DOT31(D(1,1,JFRM), V(1,12), V(1,11))
      CALL VECADD(V(1,11), SEGLP(1,JFRM), V(1,10))
      CALL VECSUB(V(1,10), SEGLP(1,JTO), V(1,9))
      CALL MAT31(D(1,1,JTO), V(1,9), V(1,8))
      CALL CROSS(WMEG(1,JTO), V(1,8), V(1,7))
      CALL VECSUB(V(1,5), V(1,7), V(1,6))
      LL = 6
 210  CALL DOT31(DPMI(1,1,ITO), V(1,LL), PQUANT(KOL))
      GO TO 550
C
C    LINEAR DISPLACEMENTS OF POINTS ATTACHED TO SEGMENTS
C
 300  CALL DOT31(D(1,1,JFRM), V(1,12), V(1,1))
      CALL VECADD(V(1,1), SEGLP(1,JFRM), V(1,2))
      CALL VECSUB(V(1,2), SEGLP(1,JTO), V(1,3))
      IF (KONTLP(4,KPT) .GT. 0) GO TO 310
      CALL MAT31(D(1,1,JFRM), V(1,3), V(1,4))
      CALL SCALED(V(1,4), -1., PQUANT(KOL))
      GO TO 550
 310  CALL MAT31(D(1,1,JTO), V(1,3), PQUANT(KOL))
      GO TO 550
C
C      ANGULAR ACCELERATIONS OF SEGMENTS
C
 400  CALL DOT31(D(1,1,JFRM), WMEGD(1,JFRM), V(1,2))
      CALL MAT31(D(1,1,JTO), V(1,2), V(1,3))
      CALL VECSUB(V(1,3), WMEGD(1,JTO), V(1,4))
      CALL DOT31(DPMI(1,1,ITO), V(1,4), V(1,5))
      CALL SCALED(V(1,5), TWOPI, PQUANT(KOL))
      GO TO 550
C
C       ANGULAR VELOCITIES OF SEGMENTS
C
 500  CALL DOT31(D(1,1,JFRM), WMEG(1,JFRM), V(1,1))
      CALL MAT31(D(1,1,JTO), V(1,1), V(1,2))
      CALL VECSUB(V(1,2), WMEG(1,JTO), V(1,3))
      CALL SCALED(V(1,3), TWOPI, V(1,4))
      CALL DOT31(DPMI(1,1,ITO), V(1,4), PQUANT(KOL))
 550  PQUANT(KOL+3) = SQRT(PQUANT(KOL)**2 + PQUANT(KOL+1)**2 +
     1      PQUANT(KOL+2)**2)
      GO TO 900
C
C      EULER ANGLE ORIENTATION OF SEGMENTS
C
 600  CALL DOT33(DPMI(1,1,IFRM), D(1,1,JFRM), FM(1,1,1))
```

Table C.6. Listings of Code Changes to GUCVS (Page 27 of 38)

```
      CALL DOTT33(FM(1,1,1),D(1,1,JTO),FM(1,1,2))
      CALL MAT33(FM(1,1,2),DPMI(1,1,ITO),FM(1,1,3))
      CALL YPRDEG(FM(1,1,3), PQUANT(KOL))
      TRACE = .5*(FM(1,1,3) + FM(2,2,3) + FM(3,3,3) - 1.0)
      IF (TRACE .GT. 1.0) TRACE = 1.0
      IF (TRACE .LT. -1.0) TRACE = -1.0
      PQUANT(KOL+3) = ARCOS(TRACE) / RADIAN
      GO TO 900
C
C           JOINT PARAMETERS
C
  700 CALL INREAD(5, KAS, LIN, PQUANT(KOL))
      PQUANT(KOL+1) = PQUANT(KOL+1) / RADIAN
      PQUANT(KOL+2) = PQUANT(KOL+2) / RADIAN
      PQUANT(KOL+3) = PQUANT(KOL+3) / RADIAN
      PQUANT(KOL+4) = SQRT(PQUANT(KOL+4))
      PQUANT(KOL+5) = SQRT(PQUANT(KOL+5))
      PQUANT(KOL+6) = SQRT(PQUANT(KOL+6))
      IPIN(JJ) = KQUANT(KOL+7)
      GO TO 900
C
C         ALL OTHER FORCE PRODUCERS
C
  800 CALL INREAD(KAT,KAS,LIN,PQUANT(KOL))
  900 IF (IOTALK .EQ. 0) GO TO 1000
      KOLZ = KOL + KASKOL - 1
      WRITE(6,910) (PQUANT(I), I=1, KOLZ)
  910 FORMAT(7X, 6G20.10)
 1000 RETURN
      END
```

Table C.6. Listings of Code Changes to GUCVS (Page 28 of 38)

```
Listing of SUSP:CUTS(1123,1134) at 10:07:23 on AUG 29, 1985 for CCid=SS53          Page   1
      SUBROUTINE HICCSI(KASE, NUMTIM)
      DATA NUM /O/
C         DUMMY HICCSI SUBPROGRAM
C*********
C     ORIGINAL ROUTINE FOUND IN FILE "CVSHICCSI".
C     THIS ROUTINE NEEDS MODIFICATION BEFORE IT CAN BE USED.
C*********
      NUM = NUM + 1
      WRITE (6,9999) NUM, KASE, NUMTIM
 9999 FORMAT ('OHICCSI CALL NO. ', I5, '; ARGS=', 2I10)
      RETURN
      END
```

Table C.6. Listings of Code Changes to GUCVS (Page 29 of 38)

```
Listing of CVSB(62,165) at 10:09:19 on AUG 29, 1985 for CCid=SS53                                Page    1
      SUBROUTINE HICCSI(NPTS,JJ)
C                                                                     13/06/83
C
C     HIC, HSI AND CSI FOR ODIAC1 PROGRAM FOR MULTIPLE OCCUPANTS
C
C     ASSUMES Z ARRAY CONTAINS
C     Z(I,1 ),I=1,NPTS : TIME POINTS (SECONDS)
C     Z(I,JH),I=1,NPTS :  HEAD RESULTANT ACCELERATIONS (G'S)
C     Z(I,JC),I=1,NPTS : CHEST RESULTANT ACCELERATIONS (G'S)
C
C     NOTE: FOR THE FIRST OCCUPANT,
C     IF JDTPTS(1)=O,  HEAD RESULTANT IS NOT AVAILABLE (JH=NULL,JC=2).
C     IF JDTPTS(2)=O, CHEST RESULTANT IS NOT AVAILABLE (JH=2,JC=NULL).
C
      COMMON/CDINT/ JDTPTS(18),Z(6000,25)
      DIMENSION AREA(6000)
      IF  (NPTS.LT.25)  GO TO 25
      WRITE  (6,14)
   14 FORMAT (1H1, '    HIC, HSI AND CSI RESULTS')
      JOCC= O
      JSET= JJ/2
      DO 30 IND = 1,JJ,2
      JH= IND+1
      JC= IND+2
      JH1= JH-1
      JC1= JC-1
      JOCC=JOCC+1
      IF  (JDTPTS(JH1).EQ.O)  JC = JC-1
      CSI = O.O
      HSI = O.O
      HIC = O.O
      CMX = Z(1,JC)
      HMX = Z(1,JH)
      IF  (JDTPTS(JC1).EQ.O)  GO TO 16
C
C     COMPUTE CSI - CHEST SEVERITY INDEX
C
      H1  = SQRT(Z(1,JC)) * Z(1,JC)**2
      DO  15  I=2,NPTS
      H2  = SQRT(Z(I,JC)) * Z(I,JC)**2
      DT  = Z(I,1) - Z(I-1,1)
      CSI = CSI + O.5*DT*(H1+H2)
      IF  (CMX.GT.Z(I,JC))  GO TO 15
      CMX = Z(I,JC)
      CMT = Z(I,1)
   15 H1  = H2
      CSI = O.OO1*CSI
   16 IF  (JDTPTS(JH1).EQ.O)  GO TO 23
C
C     COMPUTE HSI - HEAD SEVERITY INDEX - AND AREA TABLE
C
      AREA(1) = O.O
      H1  = SQRT(Z(1,JH)) * Z(1,JH)**2
      DO  17  I=2,NPTS
      H2  = SQRT(Z(I,JH)) * Z(I,JH)**2
      DT  = O.5*(Z(I,1) - Z(I-1,1))
      AREA(I) = AREA(I-1) + DT*(Z(I-1,JH)+Z(I,JH))
      HSI = HSI + DT*(H1+H2)
```

Table C.6. Listings of Code Changes to GUCVS (Page 30 of 38)

```
      IF  (HMX.GT.Z(I,JH))  GO TO 17
      HMX = Z(I,JH)
      HMT = Z(I,1)
   17 H1  = H2
      HSI = 0.001*HSI
C
C      COMPUTE HIC - HEAD INJURY CRITERION - AND TIME DURATION HT1,HT2
C
      DO  19  K=2,NPTS
      DO  18  L=K,NPTS
      DT  = Z(L,1)  -  Z(K-1,1)
      DH  = AREA(L) - AREA(K-1)
      HT  = DH/DT
      HM  = DT*SQRT(HT)*HT**2
      IF  (HM.LE.HIC)  GO TO 18
      HIC = HM
      HT1 = Z(K-1,1)
      HT2 = Z(L,1)
      HA2 = Z(L,JH)
      HA1 = Z(K-1,JH)
      AVE = HT
   18 CONTINUE
   19 CONTINUE
      HIC = 0.001*HIC
      WRITE (6,31) JOCC
   31 FORMAT(1HO,'*******************************************************',
     1 /,'    OCCUPANT  NO. ',I4,/,1X,
     2            '*******************************************************')
      WRITE  (6,21)  HIC,HT1,HT2,HA1,HA2,AVE
   21 FORMAT (1HO, '    HEAD INJURY CRITERION'//
     *       '       HIC = ', F8.2,
     *          9X, 'TIME DURATION = ', F9.3, '  TO ', F9.3, ' MSEC'/
     *  20X, 'WITH HEAD RESULTANTS = ', F9.3, ' AND ', F9.3, ' G''S'//
     *14X,'AVERAGE HEAD RESULTANT FOR TIME DURATION = ', F9.3, ' G''S')
      WRITE  (6,22)  HSI,HMX,HMT
   22 FORMAT (1HO, '    HEAD SEVERITY INDEX'//
     *       '      HSI = ', F8.2//
     * '     MAX  HEAD RESULTANT = ', F9.3, ' G''S AT ', F9.3, ' MSEC')
   23 IF  (JDTPTS(JC1).EQ.O)  GO TO 30
      WRITE  (6,24)  CSI,CMX,CMT
   24 FORMAT (1HO, '    CHEST SEVERITY INDEX'//
     *       '      CSI = ', F8.2//
     * '     MAX CHEST RESULTANT = ', F9.3, ' G''S AT ', F9.3, ' MSEC')
   30 CONTINUE
   25 RETURN
      END
```

Table C.6. Listings of Code Changes to GUCVS (Page 31 of 38)

```
Listing of SUSP:CUTS(1135,1189) at 10:07:36 on AUG 29, 1985 for CCid=SS53                Page   1
      SUBROUTINE INREAD(KAT, KAS, LIN, ARRAY)
      DIMENSION ARRAY(2)
      COMMON /PRESET/ ISYMB(49), KASLIN(3,16), LDNARY(3), LPP, NW60
     1     , MAXKAP
      COMMON /HLDCON/ NRNBAS(246), LFIRST(241), KASNUM, KATKOL(12),
     1       KPRELM(3), KATKAS(3,12), KASKTL(5,241), MAXPTS,
     2       NRN, LSTEP, LASREC(3)
C
C      NOTE: THERE EXISTS A DUAL ROLE FOR CATEGORIES IN INREAD.
C            CATEGORIES 1-5 ARE RECORDING CATEGORIES,
C            CATEGORIES 8-14 ARE PRINTING CATEGORIES, AND
C            ALL OTHER CATEGORY NUMBERS ARE SKIPPED.
C
C      1. FOR KATG = 1 TO 5:
C         KASE = 1 TO KATKAS(1,KATG)
C         ONLY NRNBAS(KATG) APPLIES
C      2. FOR KATG = 8 TO 14:
C         KASE = KATKAS(2,KATG-2) TO KATKAS(3,KATG-2)
C         NRNBAS(KASE+5) AND LFIRST(KASE) APPLY
C      3. KASADJ = NRNBAS INDEX FOR ALL CATEGORIES AND CASES
C
      KATG = KAT
      IF (KATG .LT. 1) GO TO 60
      IF (KATG .GT. 14) GO TO 60
      KASE = KAS
      LINES = LIN
      IF (KATG .GT. 7) GO TO 10
      IF (KATG .GT. 5) GO TO 60
      LINOFF = 0
      KASADJ = KATG
      LADJST = KASE * LSTEP - LSTEP
      KATABL = KATG
      GO TO 20
   10 LINOFF = LFIRST(KASE)
      KASADJ = KASE + 5
      LADJST = 0
      KATABL = KATG - 2
 20   LEN = KATKOL(KATABL)
      IF (LINOFF .LT. 0) GO TO 30
      IF (LINOFF .LT. LINES) GO TO 50
 30   DO 40 I=1, LEN
      ARRAY(I) = 0.
 40   CONTINUE
      GO TO 60
   50 LINREL = LINES - LINOFF
      LMOD = MOD(LINREL,3)
      IF (LMOD .EQ. 0) LMOD = 3
      LDN = LDNARY(LMOD)
      LBASE = NRNBAS(KASADJ) + KPRELM(LMOD) + LADJST
      LINREC = (LINREL + 2) / 3
      NRN = LBASE + LINREC
      READ (LDN'NRN) (ARRAY(I),I=1,LEN)
      LASREC(LMOD) = MAX0(LASREC(LMOD), NRN)
   60 RETURN
      END
```

Table C.6. Listings of Code Changes to GUCVS (Page 32 of 38)

```
      SUBROUTINE PICKUP(ARRAY,NUMWRD,LASNRN)
      COMMON /IOCNTL/ LDNWRK, KWDPLN, IOTALK
      DIMENSION ARRAY(2)
      NLIN = (NUMWRD + KWDPLN - 1) / KWDPLN - 1
      NLAS = NUMWRD - NLIN * KWDPLN
      IF (IOTALK .NE. 0) WRITE(6,9998) NUMWRD,  NLIN, NLAS,
     1     LASNRN, KWDPLN
 9998 FORMAT (8HOPICKUP:, 5I10)
      IPTB = 0
      NRN = LASNRN
      IF (NLIN .LE. 0) GO TO 20
      DO 10 I = 1, NLIN
      IPTA = IPTB + 1
      IPTB = IPTB + KWDPLN
      NRN = NRN + 1
      READ(LDNWRK'NRN) (ARRAY(J), J=IPTA,IPTB)
      IF (IOTALK .GT. 1) WRITE(6,9999) NRN, IPTA, IPTB, (ARRAY(J),
     1     J=IPTA,IPTB)
   10 CONTINUE
   20 IPTA = IPTB + 1
      IPTB = IPTB + NLAS
      NRN = NRN + 1
      READ(LDNWRK'NRN) (ARRAY(L), L=IPTA,IPTB)
      IF (IOTALK .GT. 1) WRITE(6,9999) NRN, IPTA, IPTB, (ARRAY(J),
     1     J=IPTA,IPTB)
 9999 FORMAT (1H0,3I5/ (5X,6Z20))
      LASNRN = NRN
      RETURN
      END
      SUBROUTINE PRTLIN(NKAT)
      COMMON /HOLDIT/ BAGTTL(5,6), BDYTTL(5), BLTTTL(5,8), COMENT(40),
     1        DATE(3), DT, JOINT(100), INCSML, INCBIG, PLTTL(5,100),.
     2        SEG(100), VPSTTL(20), LLFRST, KASTOP, NRNTIM,
     3        TIMLAS, MULTPL, NPL, NPLT, NRNVAR, LFENKT,
     4        NBAG, NBLT, LBBAG, LBBELT, LBJNT, LBPL,
     5        LBSEG, KKNTRL(241), NSD, NSEG, ISPSWT,
     6        NSTEPS, NVEH, NGRND, NHRNSS, NJNT
      COMMON /PRESET/ ISYMB(49), KASLIN(3,16), LDNARY(3), LPP, NW60
     1         , MAXKAP
      COMMON /FETCHT/ D(3,3,12), DPMI(3,3,100), G, LPMI(100), PI,
     1        SEGLA(3,12), SEGLP(3,12), SEGLV(3,12), WMEG(3,12),
     2        WMEGD(3,12), UNITL, UNITT, UNITM, RADIAN, TIMMAX,
     3        IKASE(12), NKASE, KASKOL, KASPAG, KASFUL
      COMMON /HLDCON/ NRNBAS(246), LFIRST(241), KASNUM, KATKOL(12),
     1        KPRELM(3), KATKAS(3,12), KASKTL(5,241), MAXPTS,
     2        NRN, LSTEP, LASREC(3)
      COMMON /FETCHU/ PQUANT(16), KONTLP(6,3), JDTPTS(18), IPIN(3),
     1       KKAS, MSECPT, MTIMPT
      COMMON /ZTTHNG/ TIME(6000), ZTTH(6000,11), LINEQT(6000)
      KAT = NKAT
      JJ = KKAS * KASKOL
      NT = 6
      IF (KAT .GT. 7) GO TO 50
      IF (KAT .EQ. 7) GO TO 30
      WRITE (NT,20) TIME(MSECPT), (PQUANT(J),J=1,JJ)
   20 FORMAT (F9.3, 3(3X,4F9.3))
      GO TO 190
   30 WRITE (NT,40) TIME(MSECPT), (PQUANT(J),J=1,JJ)
```

Table C.6. Listings of Code Changes to GUCVS (Page 33 of 38)

```
   40 FORMAT (F9.3, 2(F5.0,3F9.3,2X,3F9.3))
      GO TO 190
   50 NGO = KAT - 7
      IF (NGO .LT. 1 .OR. NGO .GT. 7) GO TO 190
      GO TO (60, 130, 80, 100, 110, 150, 170), NGO
   60 WRITE (NT,70) TIME(MSECPT), (PQUANT(J),J=1,JJ)
   70 FORMAT (F9.3, 2(F9.3,3F9.2,3F8.3))
      GO TO 190
   80 WRITE (NT,90) TIME(MSECPT), (PQUANT(J),J=1,JJ)
   90 FORMAT (F9.3, 4(F15.6,F12.2,3X))
      GO TO 190
  100 WRITE (NT,90) TIME(MSECPT), (PQUANT(J),J=1,JJ)
      GO TO 190
 110  JJ = 2
      IF (KONTLP(3,1) .NE. 0) JJ = 4
      IF (KONTLP(1,2) .NE. 0) JJ = 6
      IF (KONTLP(3,2) .NE. 0) JJ = 8
      WRITE (NT,120) TIME(MSECPT), (PQUANT(J),J=1,JJ)
  120 FORMAT (F9.3, 4(F14.3,F12.2,4X))
      GO TO 190
  130 WRITE (NT,140) TIME(MSECPT), (PQUANT(J),J=1,10)
  140 FORMAT (2F9.3, 3F9.2, 3F8.3, 2X, 3F8.3)
      GO TO 190
  150 WRITE (NT,160) TIME(MSECPT), (PQUANT(J),J=1,12)
  160 FORMAT (F9.3, 3X, 3F9.2, 2(3X,3F9.3), 3X, 3F9.2)
      GO TO 190
  170 WRITE (NT,180) TIME(MSECPT), (PQUANT(J),J=1,JJ)
  180 FORMAT (F9.3, 4(3X,3F9.2))
  190 RETURN
      END
      SUBROUTINE SCALED(VA,D,VB)
      DIMENSION VA(3),VB(3)
      DO 10 I = 1, 3
      VB(I) = VA(I) / D
   10 CONTINUE
      RETURN
      END
```

Table C.6. Listings of Code Changes to GUCVS (Page 34 of 38)

```
Listing of SUSP:CUTS(1307,1318) at 10:08:21 on AUG 29, 1985 for CCid=SS53                                        Page    1

  1307           SUBROUTINE SLPLOT(KASE, NUMTIM)
  1308           DATA NUM /0/
  1309      C          DUMMY SLPLOT SUBPROGRAM
  1310      C*********
  1311      C        ORIGINAL ROUTINE FOUND IN FILE "CVSSLPLOT".
  1312      C        THIS ROUTINE NEEDS MODIFICATION BEFORE IT CAN BE USED.
  1313      C*********
  1314           NUM = NUM + 1
  1315           WRITE (6,9999) NUM, KASE, NUMTIM
  1316       9999 FORMAT ('OSLPLOT CALL NO. ', I5, '; ARGS=', 2I10)
  1317           RETURN
  1318           END
```

Table C.6. Listings of Code Changes to GUCVS (Page 35 of 38)

Listing of CVSB(4699,4788) at 10:09:37 on AUG 29, 1985 for CCid=SS53 Page 1

```
      SUBROUTINE  SLPLOT(X, NX, XO, XN, XL, XSIZE, XLAB, NXLB,
     *                   Y, NY, YO, YN, YL, YSIZE, YLAB, NYLB,
     *                 NPTS, NYY, NDY, PLAB1, NPLB1, PLAB2, NPLB2,KPLT)
C                                                                06/02/83
C
C     ARGUMENTS:
C        X(NPTS)     - ARRAY OF NPTS ABSCISSAS TO BE PLOTTED.
C        Y(NDY,NYY)  - ARRAY OF NPTS*NYY ORDINATES TO BE PLOTTED.
C        NX,NY       - 0 OR  POSITIVE - LINEAR PLOTS ON POSITIVE AXIS
C                      NEGATIVE - LOGARITHMIC PLOT ON NEGATIVE AXIS.
C                      NX,NY ARE GRID DIVISIONS ALONG X- AND Y-AXES.
C        XO,YO       - MINM VALUES OF X AND Y.
C        XN,YN       - MAXM VALUES OF X AND Y.
C        XL,YL       - LENGTH (INCHES) OF X,Y AXES.
C        XSIZE,YSIZE - PAPER SIZE (INCHES) IN X,Y DIRECTIONS.
C        XLAB,YLAB   - X,Y AXES LABELS (ALPHANUMERIC ARRAYS).
C        NXLB,NYLB   - NO. OF CHARACTERS IN X,Y LABELS.
C        NPTS        - NO. OF POINTS IN X ARRAY AND EACH Y ARRAY.
C        NYY         - NO. OF Y ARRAYS TO BE PLOTTED VS. X ARRAY.
C        NDY         - FIRST DIMENSION OF Y ARRAY IN CALLING ROUTINE.
C                      (NDY MUST BE .GE. NPTS)
C        PLAB1,PLAB2 - 1ST & 2ND LINES OF PLOT ID LABELS (ALPHANUMERIC).
C        NPLB1,NPLB2 - NO. OF CHARACTERS IN PLOT ID LABELS.
C        KPLT        - PLOT NUMBER
C        NPLSMB      - SYMBOL NO. TO BE USED FOR EACH CURVE
C
C     NOTE: PLOTS WILL BE TRUNCATED AS FOLLOWS:
C      NX,NY POSITIVE - XO,YO .LE. X,Y .LE. XN,YN
C      NX,NY NEGATIVE - XO,YO .LE. X,Y .LE. XN,YN
C
      COMMON/IPLSMB/NPLSMB(100)
      DIMENSION X(NPTS),Y(NDY,NYY),XLAB(1),YLAB(1),PLAB1(1),PLAB2(1)
      DIMENSION XPL(5000),YPL(5000)
C
C
C     WRITE (6,4) NDY,NYY,NPTS,Y
C   4 FORMAT(1X,' NDY ',I6,5X,' NYY ',I6,5X,' NPTS ',I6,/,(1X,6E15.5))
      CALL BGNPL(KPLT)
      CALL PAGE(XSIZE,YSIZE)
      CALL NOBRDR
      CALL NOCHEK
C
C         CHECK TYPE OF PLOT AXIS
C
      CALL HEIGHT(0.18)
      CALL INTAXS
      CALL TITLE(PLAB1,NPLB1,XLAB,NXLB,YLAB,NYLB,XL,YL)
      CALL FRAME
      YMESS=YL+0.25
      XMESS=XL-2.5
      CALL HEIGHT(0.16)
      IF (NX.LT.0) XCYCL=ALOG10(XN/XO)
      IF (NY.LT.0) YCYCL=ALOG10(YN/YO)
      IF (NX.GT.0) XSTEP=(XN-XO)/XL
      IF (NY.GT.0) YSTEP=(YN-YO)/YL
      IF (NY.GE.0 .AND.NX.GE.0) CALL GRAF(XO,'SCALE',XN,YO,'SCALE',YN)
      IF (NX.LT.0 .AND. NY.LT.0) CALL LOGLOG(XO,XCYCL,YO,YCYCL)
      IF (NX.GE.0 .AND. NY.LT.0) CALL YLOG(XO,XSTEP,YO,YCYCL)
```

Table C.6 Listings of Code Changes to CUCVS (Page 36 of 38)

```
Listing of CVSB(4699,4788) at 10:09:37 on AUG 29, 1985 for CCid=SS53                          Page    2
4757           IF (NX.LT.O .AND. NY.GE.O) CALL XLOG(XO,XCYCL,YO,YSTEP)
4758           IF (NX.GT.O) IGRX=NX
4759           IF (NX.LT.O) IGRX=-NX
4760           IF (NY.GT.O) IGRY=NY
4761           IF (NY.LT.O) IGRY=-NY
4762           IF (NX.EQ.O) IGRX=1
4763           IF (NY.EQ.O) IGRY=1
4764           CALL GRID(IGRX,IGRY)
4765           DO  40  J=1,NYY
4766           JJP=NPLSMB(J)
4767     C     WRITE (6,52) XO,YO,XN,YN,XL,YL,JJP,XSTEP,YSTEP,J,NYY
4768        52 FORMAT(1X,' XO, YO ',2F15.5,5X,' XN, YN ',2F15.5,/,1X,' XL, YL ',
4769          12F15.5,5X,' NPLSMB ',I5,/,1X,' XSTEP, YSTEP ',2F15.5,
4770          25X,'  J  ',I4,5X,' NYY ',I4)
4771           DO  39  I=1,NPTS
4772           XPL(I)=X(I)
4773           YPL(I)=Y(I,J)
4774       39  CONTINUE
4775           IF (JJP.LT.O) CALL DASH
4776           IF (JJP.LT.O) JJP1=JJP+1
4777           IF (JJP.GE.O) JJP1=JJP
4778           CALL CURVE(XPL,YPL,NPTS,JJP1)
4779           IF (JJP.LT.O) CALL RESET('DASH')
4780       40  CONTINUE
4781           CALL HEIGHT(O.15)
4782           CALL MESSAG('ODIAC1 / GMR',12,XMESS,YMESS)
4783           YMS1=YMESS-O.15
4784           XMS1=XMESS-O.05
4785           CALL MESSAG('.............$',100,XMS1,YMS1)
4786           CALL ENDPL(KPLT)
4787           RETURN
4788           END
```

Table C.6. Listings of Code Changes to GUCVS (Page 37 of 38)

```
Listing of SUSP:CUTS(1319,1332) at 10:08:43 on AUG 29, 1985 for CCid=SS53                    Page    1
 1319          SUBROUTINE VECADD(VA,VB,VC)
 1320          DIMENSION VA(3), VB(3), VC(3)
 1321          DO 10 I = 1, 3
 1322          VC(I) = VA(I) + VB(I)
 1323     10   CONTINUE
 1324          RETURN
 1325          END
 1326          SUBROUTINE VECSUB(VA, VB, VC)
 1327          DIMENSION VA(3), VB(3), VC(3)
 1328          DO 10 I = 1, 3
 1329          VC(I) = VA(I) - VB(I)
 1330     10   CONTINUE
 1331          RETURN
 1332          END
```

Table C.6. Listings of Code Changes to GUCVS (Page 38 of 38)