TECHNICAL REPORT NO. 11789 (LL 143)

THE AMC '71 MOBILITY MODEL

U.S. ARMY TANK-AUTOMOTIVE COMMAND LABORATORIES

VOLUME II

APPENDICES A, B and C

by The Staffs of the Mobility & Environmental Division, U.S. Army Engineer Waterways Experiment Station, & the Surface Mobility Laboratory, U.S. Army Tank-Automotive Command.

TACOM

MOBILITY SYSTEMS LABORATORY

U.S. ARMY TANK AUTOMOTIVE COMMAND Warren, Michigan

Distribution of this document is unlimited.

APPENDIX A

DESCRIPTION OF TERRAIN FOR MOBILITY MODELING

This appendix contains the description of quantitative characteristics of terrains necessary for the operation of the AMC '71 Vehicle Mobility Model.

Methods for Describing Terrain

Areal and Linear Terrains:

Areal terrain units can be represented on a map as an area bordered by an irregular closed line. Linear terrain units appear on the map as a line because their width is relatively small compared to their length. A ravine or a river is a linear terrain unit.

The "WES Terrain Description System" was used to characterize areal and linear terrain data for the ground mobility model. Only a brief explanation of this system is given in this appendix. A more complete explanation can be found in Volume 1 of Reference 5 (listed at the end of the main text of this report).

The terms and values used to describe both areal and linear terrains are defined in Table A1. Each attribute of a terrain that is considered to affect mobility is called a terrain factor. Related factors are grouped in factor families, which are: surface composition, surface geometry, vegetation and hydrologic geometry.

Each terrain factor can be quantitatively characterized in terms of the terrain factor classes given in Table A1. A terrain unit is then described by an array of terrain factor class numbers. This array is designated by a terrain unit number. The final product of the system is a terrain map and a table that shows all the factor complex numbers for each terrain unit.

Areal Terrain Maps:

The following procedures are followed to form an areal terrain map legend: One factor at a time is mapped to form factor maps by depicting areas within which the terrain factor class number is constant; factor maps are then overlaid to form factor family maps and the factor family maps are overlaid to form a terrain factor complex map. Terrain factor class numbers are then replaced by terrain unit numbers on the terrain factor complex map, and a legend relating the terrain unit number to the respective terrain factor class numbers is prepared. Examples of an areal terrain map and legend are shown in Figures A1 and A2, respectively.

The areal terrain data are entered directly into the computer in the form shown in the terrain map legend. The terrain factor values which correspond to the terrain factor class numbers (Table A1) are a permanent part of the AMC Mobility Model.

Linear Terrain Maps:

Linear terrain maps are prepared in much the same way as areal terrain maps, except that a single line representing a linear feature is overlaid successively with a factor map until all the factors are overlaid. The factor complex number is then replaced by a terrain unit number, and a legend relating the terrain unit numbers and terrain factor numbers is prepared. Examples of a linear terrain map and legend are shown in Figures A3 and A4, respectively.

The only features mapped as linear terrains at present are drainage features. Other linear features, such as road embankments, will be added at a later date.

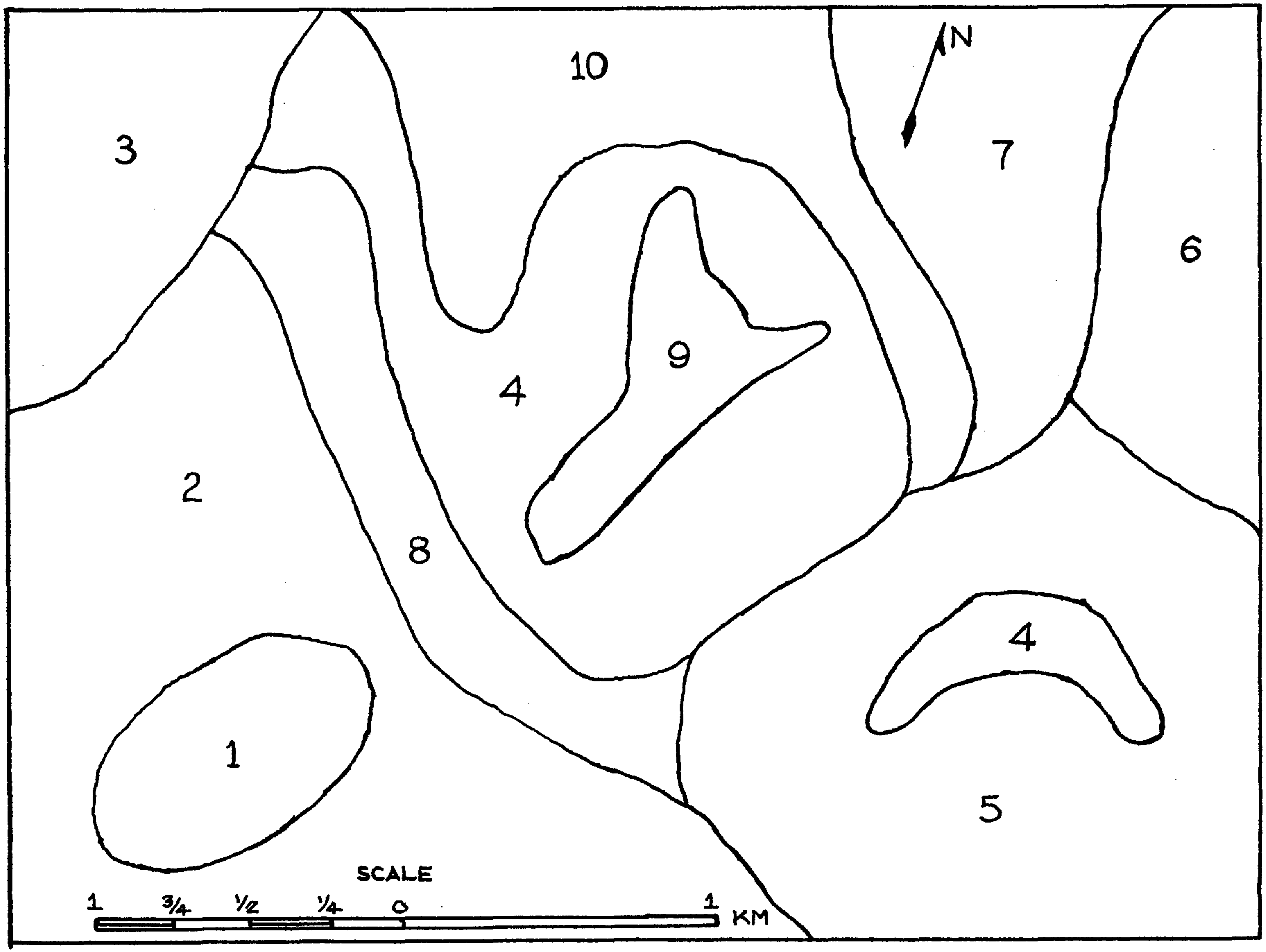


FIG. A1. Example of a Terrain Factor Complex Map

Terrain Unit No.	Terrain Factor Class No.																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
	Surface Type	Surface Strength (Dry Season)	Surface Strength (Average Season)	Surface Strength (Wet Season)	Slope	Obstacle Angle	Obstacle Vertical Magnitude	Obstacle Base Width	Obstacle Length	Obstacle Spacing	Obstacle Spacing Type	Surface Roughness	Stem Spacing of Stem Diam Class 1	Stem Spacing of Stem Diam Class 2	Stem Spacing of Stem Diam Class 3	Stem Spacing of Stem Diam Class 4	Stem Spacing of Stem Diam Class 5	Stem Spacing of Stem Diam Class 6	Stem Spacing of Stem Diam Class 7	Stem Spacing of Stem Diam Class 8	Recognition
1	1	1	2	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	3	4	5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	5	6	7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	7	8	9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	9	10	11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	2	1	2	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	2	7	8	9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	2	9	10	11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	3	1	2	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	3	7	8	9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

FIGURE A2. Example of a Terrain Map Legend

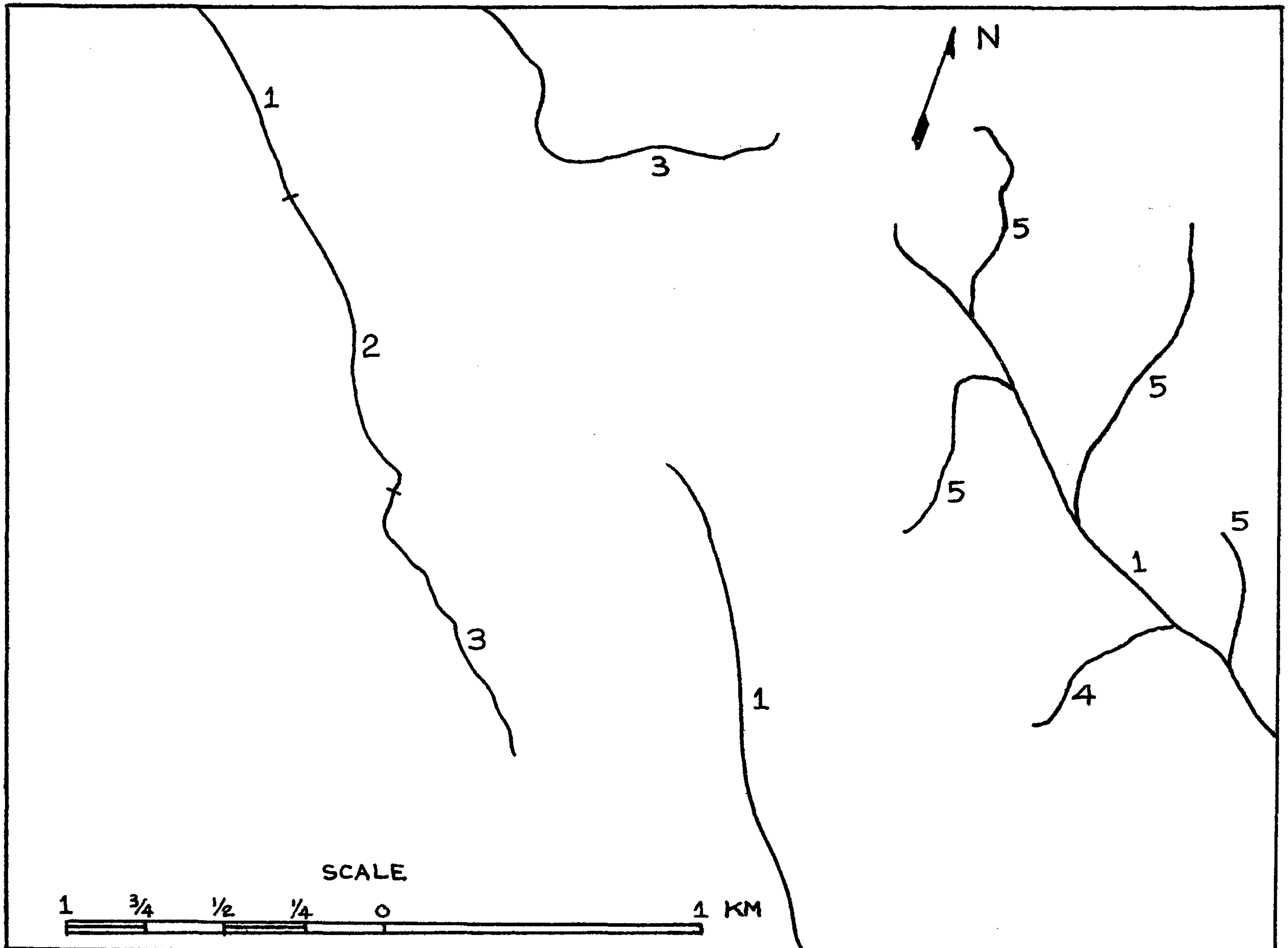


FIG. A3. Example of a Linear Terrain Map

Terrain Unit No.	Terrain Factor Complex No.					
	Left Gap Side Slope	Differential Bank Height	Right Gap Side Slope	Water Width	Water Depth	Stream Velocity
1	1	11	1	2	1	2
2	7	1	7	3	2	2
3	8	1	8	2	1	2
4	10	1	10	10	2	2
5	11	1	8	90	4	1

FIGURE 4A. Example of Linear Terrain Map Legend

Traverses:

The AMC '71 Mobility Model may be run without submodel ROUTE. In this case, one calculates the times-intervals needed to cross consecutive terrain units along a path consisting of a continuous sequence of straight line segments. The additional input data necessary to calculate the total time and the average speed associated with the preselected path consist of pairs of numbers representing the terrain unit code number and the length of the path segment in the terrain unit.

These data can form the basis for computing a great variety of significant output data. For example, one can calculate the average speed along a path and then the average speed obtained when the worst 5%, 10%, 15%, etc., of the terrain units are removed from consideration. This way one can reflect the fact that a driver would avoid the most difficult terrain units. To cite another example, one can show the percent of terrain units that each vehicle must avoid in order to attain a given average speed.

In its original form, however, the AMC '71 Model was only geared to find the best route and the speed made good across a large area.

The details for the necessary data preparation are spelled out below:

a. The terrain strip is divided into sections. Evenly spaced points are placed on each boundary between sections, as shown in Figure A5a. (The number of points is five in this example.)

b. Each point is connected to all points on the opposite boundary of the section (to form 25-path segments). (Figure A5b.)

c. For each path segment, the distance in feet through every areal terrain unit encountered is measured.

d. Data are then prepared for the computer, for each path-segment in each section, in the form illustrated below for Section 1, path-segment 4-3, presented in Figure A5c. (This information is contained in the "line number".)

Line No.	No. of Terrain Units Crossed	Terrain Unit No.	Distance Ft.
01430	7	219	510
		10	240
		55	390
		10	230
		47	140
		91	198
		1061	1820

e. For each path-segment, the linear features encountered are noted and prepared for the computer in the following form:

Line No.	No. of Terrain Units Crossed	Terrain Unit No.	Terrain Unit No.
01430	2	7	19

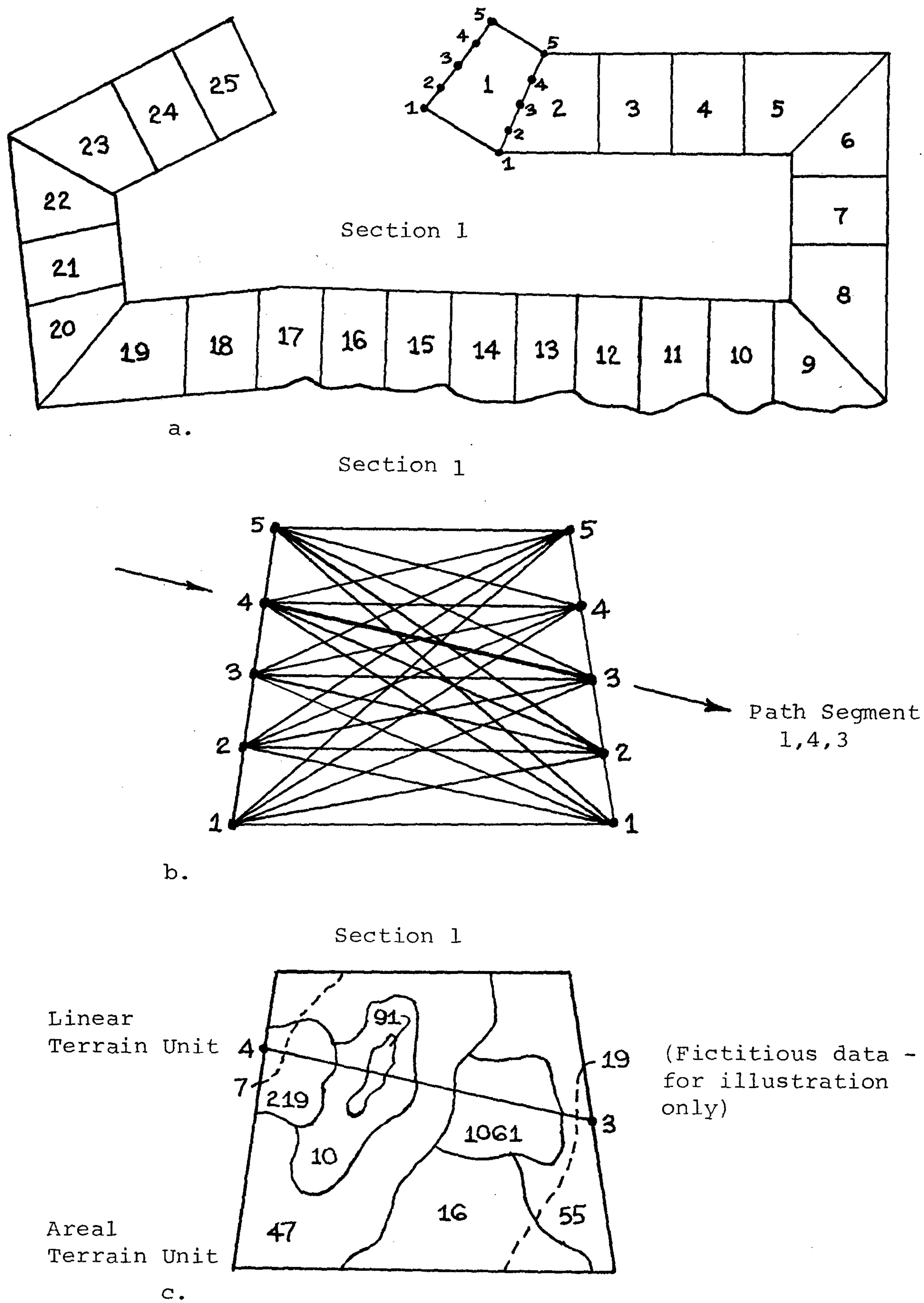


FIG A5. Method of Preparing Traverse Data for AMC Mobility Model.

Puerto Rico Terrain Data

The terrain selected as representative of Puerto Rico was a "traverse strip" (defined as a band or zone of a country 3 to 4 km wide and about 40 km long, not necessarily straight). The location of the traverse strip in Puerto Rico is shown in Figure A6. The areal terrain map for Puerto Rico is presented as plate 1, located at the end of the report, and a representative sample of the map legend is given in Table A2.

The terrain data for all areal terrains were mapped as previously discussed. Lakes or marshes were mapped as areal features, and water depth was added as a terrain factor to the group of factors shown in Table A3. Soil strength classes were mapped as the same class for all seasons for marshes and lakes.

The linear terrain map for Puerto Rico is presented in plate 2, and the map legend is given in Table A4. Stream gradient and roughness coefficient were added to the terrain factor complex number, but are not used by the AMC Mobility Model.

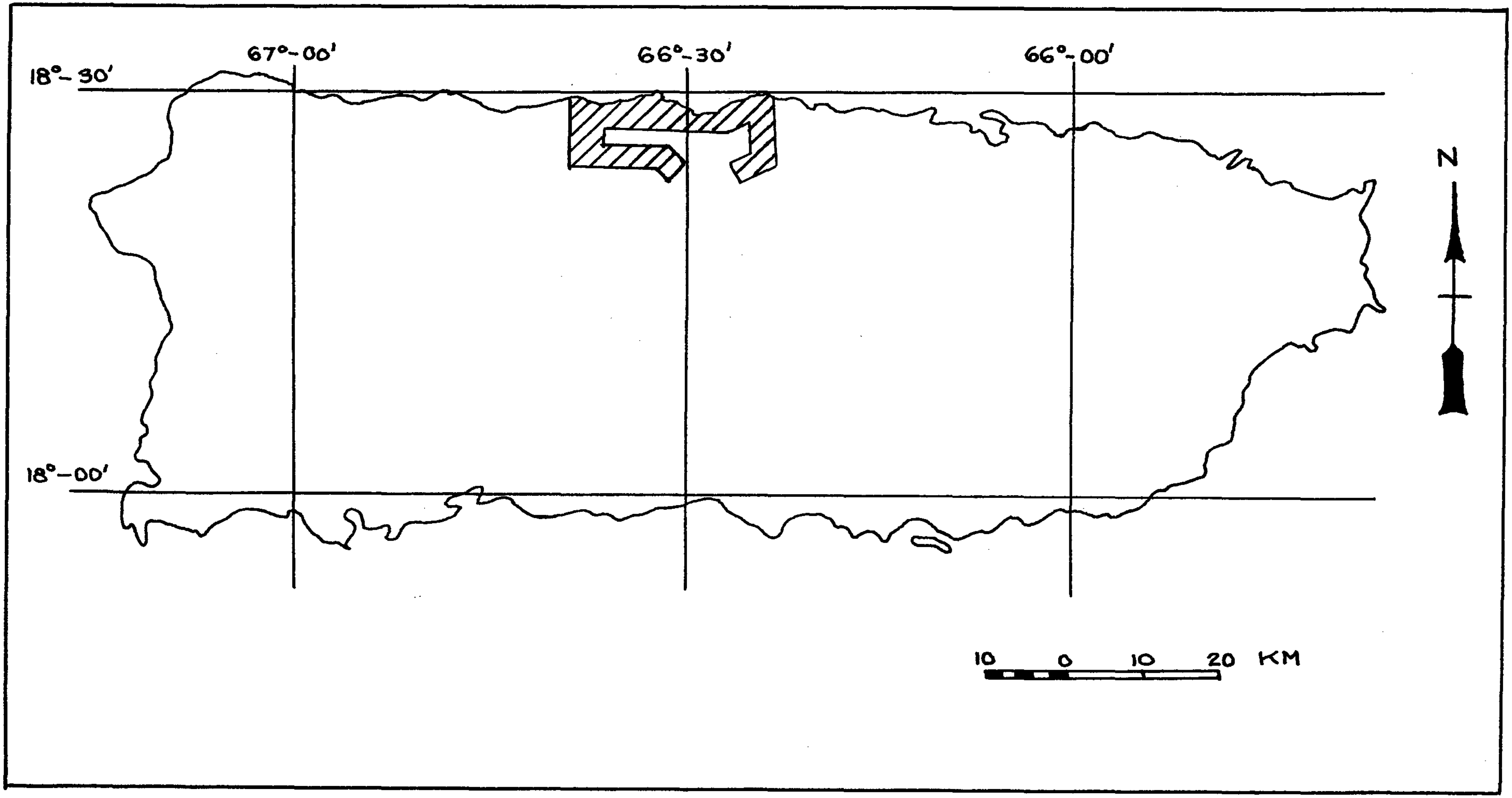


FIG. A6. Location of Traverse Strip in Puerto Rico

TABLE A1

TERRAIN DESCRIPTION

The terms and values used in describing terrains for the AMC '71 Model are given in this table.

Terms Used to Describe Terrain Data

The definitions of the important terms used in describing terrain data are as follows:

A. General Terrain Terms:

1. Areal terrains - Terrains that can be delineated on a terrain map as a patch with both length and width. For example, a forest is an areal terrain.

2. Linear terrains - Terrains that appear on a terrain map as lines due to their extensive length and narrow width. For example, a river, highway embankment, etc., are linear features.

3. Terrain country - A terrain country is an imaginary or geographic area containing two or more terrain units.

4. Terrain unit - A terrain unit is a patch (areal or linear) of terrain described by a specific terrain unit number.

5. Terrain factor complex number - A terrain factor complex number is a combination of two or more terrain factor class numbers chosen for a specific purpose.

6. Terrain factor class number - A terrain factor class number is a number assigned to a terrain

TABLE A1 (cont'd)

factor class range. For mobility purposes, the terrain factor class numbers were assigned in order of increasing severity of effect on vehicle performance.

7. Terrain factor class (class range) - A specific range of factor values established for a specific purpose. For example, a range of slope from 0 to 1.5 deg.

8. Terrain factor value (value) - A terrain factor value is a specific occurrence of a terrain factor. For example, 1.5 deg is a factor value of the terrain factor, slope.

9. Terrain factor - A terrain factor is any attribute of the terrain that can adequately be described at any point (or instant of time) by a single measurable value; for example, slope and plant stem diameter.

10. Terrain factor family - A terrain factor family is two or more terrain factors grouped together. The terrain factor families used to describe terrains are: surface composition, surface geometry, vegetation and hydrologic geometry.

B. Surface Composition Terms:

1. Fine-grained soil - A soil of which more than 50 percent of the grains, by weight, will pass a No. 200 U.S. standard sieve (smaller than 0.074 mm in diameter).

2. Coarse-grained soil - A soil of which more than 50 percent of the grains, by weight, will be retained on a No. 200 sieve (larger than 0.074 mm in diameter).

TABLE A1 (cont'd)

3. <u>Organic soils (muskeg)</u> - A terrain surface composed of a living organic mat of mosses, sedges and/or grasses with or without tree or shrub growth. Underneath the surface there is a mixture of partially decomposed and disintegrated organic material, commonly known as "peat" or "muck".

4. <u>Cone index (CI)</u> - An index of shearing resistance of soil obtained with the cone penetrometer. The value represents the resistance of the soil to penetration of a 30-degree cone of 0.5 sq-in base or projected area.

5. <u>Rating cone index (RCI)</u> - Product of CI and remolding index (RI). RI is the ratio of remolded soil strength to original strength. RCI expresses the soil strength rating of a soil subjected to vehicular traffic.

C. <u>Surface Geometry Terms</u>:

1. <u>Slope (slope)</u> - The angular deviation of a surface from the horizontal, measured perpendicular to the topographic contours (see sketch).

2. <u>Obstacle approach angles (A)</u> - The angle formed by the inclines at the base of a positive or top of a negative vertical obstacle that a vehicle must sense in surmounting the obstacle (see sketch).

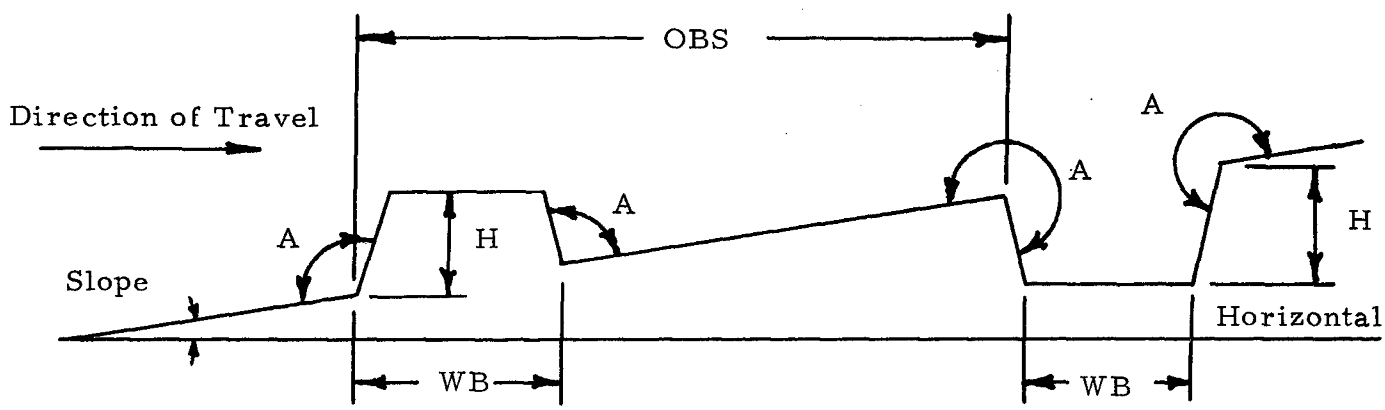
3. <u>Obstacle base width (WB)</u> - The distance across the bottom of the obstacle (centimeters).

4. <u>Obstacle spacing (OBS)</u> - The horizontal distance between contact edges of vertical obstacles (see sketch).

TABLE A1 (cont'd)

5. Obstacle Vertical Magnitude (H) - The vertical distance from the base of a vertical obstacle to the crest of the obstacle (centimeters).

6. Obstacle Length (OBL) - The length of the long axis of the obstacle, measured perpendicularly to the plane of the paper (dimension:meter).



D. Vegetation Terms:

1. Stem Diameter - The diameter of the tree stems at breast height or at 4 feet above the ground. This value is introduced to the model in centimeters.

2. Stem Spacing - The average distance (meters) between tree stems. This value is computed from the number of stems per unit area, assuming that the stems are arranged in a hexagonal pattern.

TABLE A1 (cont'd)

3. Recognition Distance - The distance a vehicle driver can see and recognize objects that may be hazardous to his vehicle or himself in meters.

E. Hydrologic Geometry Terms:

1. Differential Bank Height (BD) - The difference in elevation of the two banks in meters (see sketch).

2. Gap Side Slope (THI, RBA) - The angle formed by the bounding incline at the top of the hydrologic feature. The angle is measured with respect to the horizontal (see sketch).

3. Water Depth (WD) - Maximum depth of water in channel in centimeters (see sketch).

4. Water Width (RW) - The width of the stream in meters at water level (see sketch).

5. Water Velocity (WS) - The maximum velocity of water in a channel (meter/second).

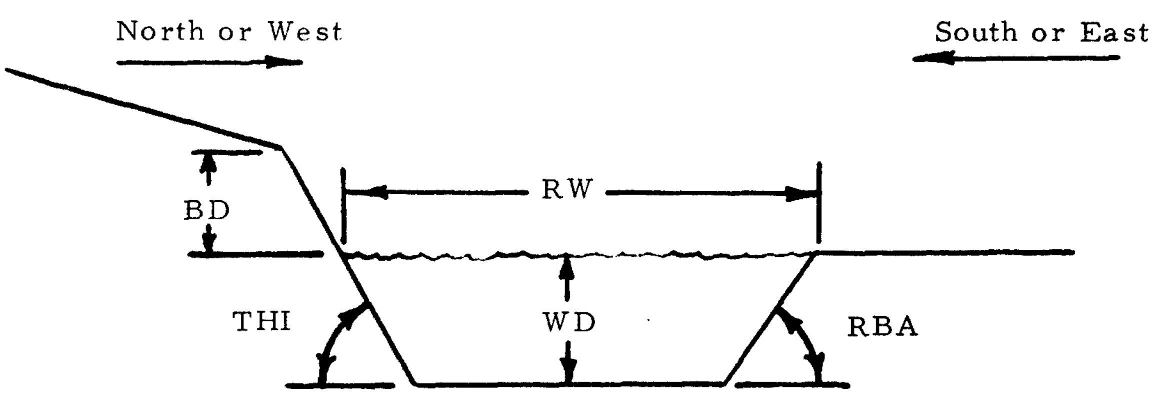


TABLE A1 (cont'd)

Numerical Values for Describing Terrain Units

The terrain factor values, terrain factor class ranges and terrain factor class numbers used to describe a terrain unit are as follows:

A. Surface Composition: Surface composition is described in terms of type of surface material and the strength of the surface material.

1. Surface Type - The surface types of material are:

Code No.	Material Type
1	Fine-grained soil
2	Coarse-grained soil
3	Organic soil

2. Soil Strength - Soil strength is described in terms of cone index (CI) or rating cone index (RCI) of the 0- to 6-in. layer. RCI is used to describe the strength of type 1 and type 3 materials. The classes and values used to describe soil strength are:

Class No.	Class Range	Value Selected for Prediction
1	> 280	300
2	221-280	250
3	161-220	190
4	101-160	130
5	61-100	80
6	41-60	50

TABLE A1 (cont'd)

Class No.	Class Range	Value Selected for Prediction
7	33-40	36
8	26-32	29
9	17-24	20
10	11-16	14
11	0-10	5

The preceding class numbers of soil strength are normally used to describe the soil strength for a terrain unit during the dry, the wet, or the average season. However, a different class number may be required to describe the soil strength during different seasons. For example, for a given terrain unit, fine-grained soils, Class No. 6, may be required to describe the wet season strength and Class No. 2, the dry season strength.

B. Surface Geometry: Surface geometry is subdivided into macrogeometry and microgeometry. Macrogeometry is described by slope angle and is usually considered as a slope length that is greater than the vehicle length. Terrain factors used to describe surface features identified as microgeometry are separated into two categories. One category includes those surface features such as boulders, stumps, logs, dikes, potholes, etc., that a vehicle will override slowly or circumvent, and the other category includes surface irregularities that are overridden and that excite the vehicle in the vertical direction. The latter category is pertinent to the ride problem. Terrain features in category 1 are described in terms of approach and departure angle, vertical magnitude, base width, length, spacing and spacing type. Surface features in category 2 are described as a continuous profile (approximately 500 feet long) in sufficient detail for a valid power spectral density to be obtained.

TABLE A1 (cont'd)

1. Macrogeometry - The classes and values used to describe slope (macrogeometry) are:

Class No.	Class Range %	Value Selected for Prediction %
1	0-2	1
2	2.1-5	3.5
3	5.1-10	7.5
4	10.1-20	15.0
5	20.1-40	30.0
6	40.1-60	50.0
7	60.1-70	65.0
8	> 70	72.0

2. Microgeometry (Category 1) - The classes and values used to describe obstacle approach and departure angle, obstacle vertical magnitude, obstacle base width, obstacle length, obstacle spacing and obstacle spacing type are:

a. Obstacle Approach and Departure Angle

Class No.	Class Range Deg.	Value Selected for Prediction, deg
1	178.6-180	179
2	180.0-181.5	181
3	175.6-178.5	177
4	181.5-184.5	183
5	170.1-175.5	173
6	184.5-190	187
7	158.1-170	164
8	190.1-202	196
9	149.1-158	154
10	202.1-211	206
11	135.1-149	142
12	211.1-225	218
13	90.0-135	112
14	> 225	225

TABLE A1 (cont'd)

b. Obstacle Vertical Magnitude

Class No.	Class Range cm	Value Selected for Prediction, cm
1	0-15	8
2	16-25	20
3	26-35	30
4	36-45	40
5	46-60	53
6	60-85	72
7	>85	85

c. Obstacle Base Width

Class No.	Class Range cm	Value Selected for Prediction, cm
1	>120	120
2	91-120	106
3	61-90	76
4	31-60	46
5	0-30	15

d. Obstacle Length

Class No.	Class Range m	Value Selected for Prediction, m
1	0-0.3	0.2
2	0.4-1.0	0.7
3	1.1-2.0	1.6
4	2.1-3.0	2.6
5	3.1-6.0	4.6
6	6.1-150	78.0
7	> 150	150.0

TABLE A1 (cont'd)

e. Obstacle Spacing

Class No.	Class Range m	Value Selected for Prediction, m
1	Bare	60.0
2	20.1-60	40.0
3	11.1-20	15.6
4	8.1-11	9.6
5	5.6-8	6.8
6	4.1-5.5	4.8
7	2.6-4.0	3.3
8	0-2.5	1.2

f. Obstacle Spacing Type

Code No.	Description
2	Linear
1	Random

3. Microgeometry (Category 2) - The data required for category 2 microgeometry is a terrain profile in sufficient detail for valid power spectral density to be obtained. An example of this terrain description is as follows:

Surface Roughness Profile Class	RMS Range	Value Selected for Prediction
1	0-0.5	0.25
2	0.6-1.5	1
3	1.6-2.5	2
4	2.6-3.5	3
5	3.6-4.5	4
6	4.6-5.5	5

TABLE A1 (cont'd)

Surface Roughness Profile Class	RMS Range	Value Selected for Prediction
7	5.6-6.5	6
8	6.6-7.5	7
9	> 7.5	8

C. Vegetation: Vegetation is described in terms of stem diameter and stem spacing. For convenience, visibility is also included as a part of the vegetation factor family since it is often closely related. Those stems that can be overridden by a vehicle are identified as longitudinal obstacles and those that must be avoided by a vehicle are identified as lateral obstacles. The classes and values used to describe stem diameter, stem spacing and visibility are as follows:

1. Stem Diameter

Class No.	Value, Cm
1	> 0
2	> 2.5
3	> 6.0
4	> 10.0
5	> 14.0
6	> 18.0
7	> 22.0
8	> 25.0

2. Stem Spacing

Class No.	Class Range m	Value Selected for Prediction, m
1	Bare	100.0
2	> 20	20.0
3	11.1-20	15.5

TABLE A1 (cont'd)

Class No.	Class Range m	Value Selected for Prediction, m
4	8.1-11	9.5
5	5.6-8	6.8
6	4.1-5.5	4.8
7	2.6-4	3.3
8	0-2.5	1.2

3. Visibility or Recognition Distance Classes at 1.5 Feet Above Ground

Class No.	Class Range m	Value Selected for Prediction, m
1	>50	50.0
2	24.1-50	37.0
3	12.1-24	18.0
4	9.1-12	10.6
5	6.1-9.0	7.5
6	4.6-6.0	5.3
7	3.1-4.5	3.8
8	1.6-3.0	2.3
9	0-1.5	0.8

NOTE: The surface code number and obstacle spacing code number are used in the same manner as terrain factor class numbers to form the terrain factor complex number.

D. Hydrologic geometry factors are primarily used to describe linear features that transport water. One hydrologic geometry factor, water depth, is also used as a part of the description of areal bodies of water such as lakes, marshes, or swamps. Other hydrologic geometry factors are differential bank height, gap side slope, water width and water velocity. The classes and values used to describe each of these factors are as follows:

TABLE A1 (cont'd)

1. Differential Bank Height

Class No.	Class Range	Value Selected for Prediction, m
1	0	0
2	N/W bank (0.1-1) higher than S/E	0.5
3	N/W bank (1.1-2) higher than S/E	1.5
4	N/W bank (2.1-4) higher than S/E	3.0
5	N/W bank (>4)	4.0
6	S/E bank (0.1-1) higher than N/W	0.5
7	S/E bank (1.1-2) higher than N/W	1.5
8	S/E bank (2.1-4) higher than N/W	3.0
9	S/E bank (>4) higher than N/W	4.0

2. Gap Side Slope

Class No.	Class Range, deg	Value Selected for Prediction, deg
1	180-185	182.5
2	185.1-190	187.5
3	190.1-200	195.0
4	200.1-210	205.0
5	210.1-220	215.0
6	220.1-230	225.0
7	230.1-250	240.0
8	250.1-260	255.0
9	260.1-265	262.5
10	265.1-270	267.5

3. Water Depth

Class No.	Class Range, cm	Value Selected for Prediction, cm
1	0-100	50
2	101-200	150
3	201-500	350
4	> 500	500

TABLE A1 (cont'd)

4. Water Velocity

Class No.	Class Range, mps	Value Selected for Prediction, mps
1	No water	NA
2	0	0
3	0-1	0.5
4	1.1-2	1.5
5	2.1-3.5	2.8
6	> 3.5	3.5

5. Water Width

Class No.	Class Range m	Value Selected for Prediction m	Class No.	Class Range m	Value Selected for Prediction m
1	No water	0	46	200.1-205	202.5
2	0.1-3	1.5	47	205.1-210	207.5
3	3.1-6	4.5	48	210.1-215	212.5
4	6.1-9	7.5	49	215.1-220	217.5
5	9.1-12	10.5	50	220.1-225	222.5
6	12.1-15	13.5	51	225.1-230	227.5
7	15.1-18	16.5	52	230.1-235	232.5
8	18.1-21	19.5	53	235.1-240	237.5
9	21.5-24	22.5	54	240.1-245	242.5
10	24.1-27	25.5	55	245.1-250	247.5
11	27.1-30	28.5	56	250.1-255	252.5
12	30.1-35	32.5	57	255.1-260	257.5
13	35.1-40	37.5	58	260.1-265	262.5
14	40.1-45	42.5	59	265.1-270	267.5
15	45.1-50	47.5	60	270.1-275	272.5
16	50.1-55	52.5	61	275.1-280	277.5
17	55.1-60	57.5	62	280.1-285	282.5
18	60.1-65	62.5	63	285.1-290	287.5

TABLE A1 (cont'd)

Class No.	Class Range m	Value Selected for Prediction m	Class No.	Class Range m	Value Selected for Prediction m
19	65.1-70	67.5	64	290.1-295	292.5
20	70.1-75	72.5	65	295.1-300	297.5
21	75.1-80	77.5	66	300.1-305	302.5
22	80.1-85	82.5	67	305.1-310	307.5
23	85.1-90	87.5	68	310.1-315	312.5
24	90.1-95	92.5	69	315.1-320	317.5
25	95.1-100	97.5	70	320.1-325	322.5
26	100.1-105	102.5	71	325.1-330	327.5
27	105.1-110	107.5	72	330.1-335	332.5
28	110.1-115	112.5	73	335.1-340	337.5
29	115.1-120	117.5	74	340.1-345	342.5
30	120.1-125	122.5	75	345.1-350	347.5
31	125.1-130	127.5	76	350.1-355	352.5
32	130.1-135	132.5	77	355.1-360	357.5
33	135.1-140	137.5	78	360.1-365	362.5
34	140.1-145	142.5	79	365.1-370	367.5
35	145.1-150	147.5	80	370.1-375	372.5
36	150.1-155	152.5	81	375.1-380	377.5
37	155.1-160	157.5	82	380.1-385	382.5
38	160.1-165	162.5	83	385.1-390	387.5
39	165.1-170	167.5	84	390.1-395	392.5
40	170.1-175	172.5	85	395.1-400	397.5
41	175.1-180	177.5	86	400.1-405	402.5
42	180.1-185	182.5	87	405.1-410	407.5
43	185.1-190	187.5	88	410.1-415	412.5
44	190.1-195	192.5	89	415.1-420	417.5
45	195.1-200	197.5	90	420.1-425	422.5

TABLE A2

AREAL TERRAIN UNITS

There are 1061 different terrain units in the Puerto Rico transect. A table depicting 1061 sets of terrain class numbers is too voluminous for complete reproduction.

Therefore, only a sample is given in the following:

	<<<<<<<<<<<<<<<< FACTOR COMPLEX																			>>>>>>>>>>>>>>
						< OBSTACLE >														
	<< SOIL >>																			
													<< SPACING OF STEMS EQUAL TO OR GREATER THAN GIVEN DIAMETER, CENTIMETERS >							
MAP UNIT	SURFACE TYPE	DRY STRENGTH	AVG, STRENGTH	WET STRENGTH	TOPO, SLOPE	APPROACH ANGLE	VERT, MAGNITUDE	BASE WIDTH	LENGTH	SPACING	TYPES	SURFACE ROUGHNESS	0	2,5	6	10	14	18	22	25	RECOG, DIST,
1	1,	3,	4,	5,	1,	1,	1,	5,	1,	1,	1,	1,	1,	1,	1,	1,	1,	1,	1,	1,	2
2	1,	3,	4,	5,	1,	1,	1,	5,	1,	1,	1,	1,	8,	2,	2,	2,	2,	2,	2,	2,	2
3	1,	3,	4,	5,	1,	1,	1,	5,	1,	1,	1,	3,	8,	3,	3,	3,	2,	2,	2,	2,	2
4	1,	3,	4,	5,	1,	1,	1,	5,	1,	1,	1,	2,	8,	4,	3,	3,	3,	3,	3,	3,	2
5	1,	3,	4,	5,	1,	1,	1,	5,	1,	1,	1,	3,	8,	5,	3,	3,	3,	3,	3,	3,	2
6	1,	3,	4,	5,	1,	1,	1,	5,	1,	1,	1,	3,	8,	5,	4,	4,	3,	3,	3,	3,	2
7	1,	3,	4,	5,	1,	1,	1,	5,	1,	1,	1,	3,	8,	6,	2,	2,	2,	2,	2,	2,	2
8	1,	3,	4,	5,	1,	1,	1,	5,	1,	1,	1,	1,	8,	6,	4,	4,	2,	2,	2,	2,	2
9	1,	3,	4,	5,	1,	1,	1,	5,	1,	1,	1,	2,	8,	6,	4,	4,	4,	3,	3,	3,	2
10	1,	3,	4,	5,	1,	1,	1,	5,	1,	1,	1,	3,	8,	7,	3,	2,	2,	2,	2,	2,	2
11	1,	3,	4,	5,	1,	1,	1,	5,	1,	1,	1,	3,	8,	7,	4,	4,	2,	2,	2,	2,	2
12	1,	3,	4,	5,	1,	1,	1,	5,	1,	1,	1,	2,	8,	7,	5,	4,	2,	2,	2,	2,	4
13	1,	3,	4,	5,	1,	1,	1,	5,	1,	1,	1,	1,	8,	7,	5,	5,	2,	2,	2,	2,	6
14	1,	3,	4,	5,	1,	1,	1,	5,	1,	1,	1,	1,	8,	8,	2,	2,	2,	2,	2,	2,	2

APPENDIX B

VEHICLE CHARACTERISTICS

Appendix B presents the vehicle characteristics and other related data required for the AMC '71 Vehicle Mobility Model.

A number and a computer symbol were assigned to each characteristic, which may be grouped into four categories:

a. General characteristics
b. Dynamic characteristics
c. Power train characteristics
d. Geometric characteristics

Most of the data required for groups a and d are listed (for military vehicles) in military standard characteristics or vehicle data sheets, published by the U.S. Army Tank-Automotive Command. Some of the data listed in group c are also shown on data sheets, but net engine torque, transmission characteristic curves, power train losses and other similar descriptors are only available at TACOM's Propulsion Systems Division, or must be obtained from manufacturers. The dynamic characteristics (group b) are not readily available; their establishment requires special tests or laborious calculations.

Table B1 contains the identification of the characteristics with the corresponding numbers, computer symbols and dimensions used in this study. Table B2 lists the numerical values of the characteristics for the four vehicles simulated in the initial application of the AMC '71 Mobility Model (M60, M113, M35 and M151). These values are referenced by number to Table B1.

TABLE B1

VEHICLE CHARACTERISTICS NECESSARY FOR THE AMC '71 MOBILITY MODEL

Characteristics No.	Identification	Dimensions	Computer Symbol
	General Characteristics		
1	Vehicle type (NVEH = 0 for tracked); (NVEH = 1 for 4x4, 2 for 6x6, 3 for 8x8)		NVEH
2	Gross vehicle weight	lb	GVW
3	Track type (NFL = 0 for nonflexible; NFL = 1 for flexible)		NFL
4	Grouser height for tracks; number of tires for wheeled	in	GT
5	Tire ply rating		TPLY
6	Maximum force the pushbar can withstand on the vehicle's leading edge	lb	PBF
7	Vehicle swimming speed	mph	VSS
8	Vehicle fording speed	mph	VFS
9	Maximum braking force the vehicle can develop on hard pavement	lb	XBR
10	Auxiliary water propulsion factor (no auxiliary propulsion system = .5; and propulsion system on vehicle = .8)		AWPKF
11	Vehicle rated horsepower per ton (net)		HPT
12	Number of people in the vehicle on a normal mission		NCREW

TABLE B1 (cont'd)

Characteristics No.	Identification	Dimensions	Computer Symbol
13	Vehicle winch capacity	lb	WC
14	Transmission variety (hydraulic = 0; mechanical = 1)		ITVAR
	Input Data Produced By Vehicle Ride Dynamics Subprogram		
15	Number of point pairs in array VOOB (in curve)		NC4
16	Array containing vehicle velocity vs obstacle height at 2.5 g vertical acceleration		VOOB(I,J)
17	Number of points in array VRIDE		NC5
18	Limited speed due to vibration at the driver's seat for surface roughness Class I		VRIDE(I)
	Geometric Characteristics		
19	Vehicle width	in	W
20	Vehicle length	in	VL
21	Vehicle ground clearance at the center of the greatest wheel span	in	GC
22	Rear end clearance (vertical clearance of vehicle's trailing edge)	in	REC
23	Vehicle departure angle	deg	VDA

TABLE B1 (cont'd)

Characteristics No.	Identification	Dimensions	Computer Symbol
24	Vertical clearance of vehicle's leading edge	in.	FEC
25	Vehicle approach angle (AV in FIVEYP); (VAA in OBSTCL, INPUT)	deg.	AAV
26	Track width or wheel width	in.	WID
27	Length of track on ground or wheel diameter	in.	DL
28	Wheel rim diameter	in.	RDIAM
29	Loaded wheel radius	in.	RW
30	Tire pressure	psi	TPSI
31	Ground contact area	$in.^2$	GCA
32	Height of vehicle pushbar or leading edge	in.	PBHT
33	Area of one track shoe (tracked) or number of axles (wheeled)	$in.^2$	A
34	Number of bogies (tracked) or chain indicator (wheeled); (0 = no chains; 1 = chains)		NBC
35	Distance between the first and last wheel centerlines	in.	XLT
36	Horizontal distance from the C.G. to the front wheel centerline	in.	CGF
37	Vertical distance from the C.G. to the road wheel centerline	in.	CGH

TABLE B1 (cont'd)

Characteristics No.	Identification	Dimensions	Computer Symbol
38	Maximum span between adjacent wheel centerlines (DWX in FIVEYP, RIVER)	in.	GWS
39	Angle between a line parallel to the ground surface and the line connecting the C.G. and the center of the rear wheel (road wheel or idler). The wheel is used to determine departure angle	deg.	ACG
40	Distance from the C.G. to the center of the rear wheel (road wheel or idler). The wheel is the one used to determine departure angle (in.)	in.	DCG
41	Vertical distance from the ground to the center of the rear wheel (road wheel or idler). The wheel is the one used to determine departure angle (wheeled = RW)	in.	HC
42	Track thickness plus the radius of the rear wheel (road wheel or idler). The wheel is the one used to determine departure angle (wheeled = RW)	in.	RWW
43	Maximum vertical step the vehicle can climb	in.	HS
44	Ingress swamp angle of the vehicle (THD in DIG)	deg.	SAI
45	Fording depth or draft height	in.	FD
46	Rolling radius of tire or sprocket pitch radius	in.	RR

TABLE B1 (cont'd)

Characteristics No.	Identification	Dimensions	Computer Symbol
	Power Train Characteristics		
47	Transmission type (ITRAN = 0 for manual; ITRAN = 1 for automatic)		ITRAN
48	Final drive gear ratio		FDR
49	Final drive gear efficiency		FDREF
50	Number of gear ratios in transmission		NG
51	Gear ratio of Ith gear		GR(I)
52	Transmission efficiency		EFF
53	Gear ratio from engine to torque converter		ENTCG
54	Denotes presence of a torque converter lockup; no = 0; yes = 1		LOKUP
55	Input torque at which the torque converter curves were measured	ft-lb	TC
56	Number of point pairs in array TNE1		NC1
57	Array containing torque converter speed vs converter speed ratio curve		TNE1(I,J)
58	Number of point pairs in array TTM		NC2
59	Array containing torque converter torque multiplying coefficient vs converter speed ratio curve		TTM(I,J)
60	Number of point pairs in array TTE		NC3
61	Array containing net engine torque vs engine speed curve		TTE(I,J)

TABLE B1 (cont'd)

Characteristics No.	Identification	Dimensions	Computer Symbol
	Characteristics Required for Vehicle Ride Dynamics Subprogram*		
62	Mass of main frame	$lb\text{-}sec^2/in.$	FMASS
63	Mass of wheel or bogie Assembly I	$lb\text{-}sec^2/in.$	MASS(I)
64	Pitch moment of inertia	$lb\text{-}sec^2/in.$	INRTIA
65	Horizontal distance from C.G. to wheel or bogie Assembly I	in.	LEN(I)
66	C.G. to driver distance	in.	DRVLEN

*For further details, see Appendix C.

Characteristics No.	Identification	Dimensions	Computer Symbol
	Initial Displacements		
67	Vertical C.G.	in.	VAR (1)
68	Pitch	radian	VAR (2)
69	Axle 1	in.	VAR (3)
	Axle 2	in.	VAR (4)
	Axle 3	in.	VAR (5)
	Axle 4	in.	VAR (6)
	Axle 5	in.	VAR (7)
	Axle 6	in.	VAR (8)
70	Horizontal C.G.	in.	VAR (9)
71	Threshold height of wheel segment I	in.	THRSH(I)
72	Segmented wheel spring constant for vertical component of segment I ($KCOS\emptyset_I$)	lb/in	GAMMA(I)

TABLE B1 (cont'd)

Characteristics No.	Identification	Dimensions	Computer Symbol
73	Segmented wheel spring constant for horizontal component of segment I (K SIN$\emptyset_I$)	lb/in.	SIGMA(I)
74	Feeler threshold heights (to portray leading portion of track)	in.	TH(I)
75	Track tension spring constant (between bogies)	lb/in.	
76	Track tension spring constant (feelers ahead of first bogie)	lb/in.	

TABLE B2

INPUT PARAMETER DATA FOR SIMULATED VEHICLES

Characteristic No.	M151	M35A2 Mod	M113A1	M60A1
		General Characteristics		
1	1.0	2.0	0.0	0.0
2	3,180.0	18,230	23,410	104,000
3	0.0	0.0	1.0	1.0
4	4.0	6.0	1.0	1.5
5	6.0	12.0		
6	3,180.0	18,230	55,000	500,000
7	0.0	0.0	3.5	0.0
8	2.0	2.0	5.0	2.0
9	2,560.0	15,040	19,120	83,200
10	0.5	0.5	0.5	0.5
11	46.2	15.4	17.9	11.5
12	1.0	2.0	2.0	4.0
13	0.0	10,000	0.0	0.0
14	1.0	1.0	0.0	0.0

Characteristics Produced by the Vehicle Ride Dynamics Subprogram

Characteristic No.	M151 (a)	M151 (b)	M35A2 Mod (a)	M35A2 Mod (b)	M113A1 (a)	M113A1 (b)	M60A1 (a)	M60A1 (b)
15	26		17		8		11	
16	1.0	50.0	0.0	50.0	0.0	50.0	0.0	60.0
	2.0	16.6	5.0	50.0	8.0	50.0	9.0	60.0
	3.0	10.0	7.0	35.0	9.0	31.0	10.0	12.2
	4.0	7.1	8.0	26.8	11.5	10.0	11.0	6.9
	5.0	5.5	9.0	21.2	15.0	5.0	12.0	6.0
	6.0	4.5	10.0	16.7	20.0	2.0	13.0	5.6
	7.0	3.8	11.0	15.8	27.5	0.5	14.0	5.4
	8.0	3.3	12.0	10.5	40.0	0.0	15.0	5.3
	9.0	2.9	13.0	7.8			19.0	5.1
	10.0	2.6	14.0	6.5			29.0	4.9
	11.0	2.3	15.0	5.0			40.0	4.9
	12.0	2.1	16.0	3.7				
	13.0	2.0	17.0	2.8				

TABLE B2 (cont'd)

Characteristic No.	Vehicles: M151		M35A2 Mod		M113A1	M60A1
	14.0	1.8	18.0	2.1		
	15.0	1.7	19.0	1.5		
	16.0	1.6	20.0	1.2		
	17.0	1.5	40.0	1.0		
	18.0	1.4				
	19.0	1.3				
	20.0	1.2				
	21.0	1.2				
	22.0	1.1				
	23.0	1.1				
	24.0	1.0				
	25.0	1.0				
	40.0	1.0				
17	9.0		9.0		9.0	9.0
18	30.0		40.0		40.0	30.0
	30.0		25.0		32.0	30.0
	20.0		19.0		15.0	30.0
	13.5		14.0		7.0	30.0
	10.0		11.3		4.9	24.3
	7.5		9.0		3.7	20.0
	6.3		7.5		3.1	16.8
	5.5		6.3		3.0	14.2
	5.0		5.2		3.0	12.0
			Geometric Characteristics			
19	62.25		96.0		105.0	143.0
20	132.0		280.6		192.0	273.0
21	11.4		19.0		16.0	18.0
22	16.0		33.5		20.0	40.0
23	37.0		42.0		40.0	60.0
24	19.0		36.5		23.0	45.0
25	66.0		42.0		70.0	90.0
26	7.1		11.5		15.0	28.0
27	30.8		43.6		105.0	167.0
28	16.0		20.0			

TABLE B2 (cont'd)

Characteristic No.	Vehicles M151	M35A2 Mod	M113A1	M60A1
29	14.5	20.0	12.0	13.0
30	20.0	35.0		
31	116.0	600.0	3150.0	9336
32	19.0	39.0	30.0	45.0
33	2.0	3.0	90.0	194.0
34	0.0	0.0	10.0	12.0
35	85.0	178.0	105.0	167.0
36	46.9	109.0	50.7	77.76
37	11.1	25.0	27.5	41.25
38	85.0	130.0	26.25	33.0
39	16.25	17.8	15.5	6.25
40	39.66	80.5	82.3	119.5
41	14.5	20.0	17.0	41.25
42	14.5	20.0	14.0	17.0
43	14.5	18.4	24.0	36.0
44	0.0	0.0	90.0	90.0
45	60.0	72.0	75.0	69.0
46	14.5	20.0	9.81	12.25
		Power Train Characteristics		
47	0.0	0.0	1.0	1.0
48	4.86	6.27	3.93	5.08
49	0.90	0.90	0.95	0.95
50	4.0	10.0	3.0	2.0
51	5.172	9.94	3.81	3.497
	3.179	5.5	1.936	1.256
	1.674	3.2	1.0	
	1.0	1.98		
		1.56		
		5.02		
		2.78		
		1.62		
		1.0		
		0.79		
52	0.90	0.90	0.95	0.95
53			1.0	0.862
54			1.0	0.0
55			275	900

TABLE B2 (cont'd)

Characteristics No.	M151	M35A2 Mod	Vehicles M113A1		M60A1	
56				24.0		12.0
57			0.00	2340	0.0	1875
			0.05	2320	0.1	1850
			0.10	2300	0.2	1825
			0.15	2280	0.3	1815
			0.20	2260	0.4	1830
			0.25	2250	0.5	1895
			0.30	2240	0.6	1970
			0.35	2230	0.7	2030
			0.40	2230	0.8	2130
			0.45	2240	0.85	2210
			0.50	2250	0.90	2500
			0.55	2270	1.0	50000
			0.60	2300		
			0.65	2340		
			0.70	2400		
			0.75	2490		
			0.80	2620		
			0.85	2840		
			0.90	3160		
			0.91	3280		
			0.92	3400		
			0.93	3600		
			0.94	4000		
			1.00	5000		
58				21.0		12.0
59			0.0	3.31	0.0	3.66
			0.05	3.16	0.1	3.125
			0.10	2.99	0.2	2.65
			0.15	2.80	0.3	2.28
			0.20	2.58	0.4	1.95
			0.25	2.38	0.5	1.67
			0.30	2.19	0.6	1.42
			0.35	2.02	0.7	1.22
			0.40	1.87	0.8	1.05
			0.45	1.73	0.85	0.98
			0.50	1.60	0.9	0.97

TABLE B2 (cont'd)

Characteristics No.	M151		M35A2 Mod		M113		M60A1	
					0.55	1.49	1.0	0.97
					0.60	1.38		
					0.65	1.28		
					0.70	1.18		
					0.75	1.07		
					0.80	0.98		
					0.85	0.98		
					0.90	0.98		
					0.95	0.97		
					1.00	0.97		
60	10.0		9.0		12.0		13.0	
61	800	92	1000	2915	600	158.9	1200	1610
	1200	95	1200	2985	800	309.4	1300	1645
	1600	103	1400	297	1000	379.4	1400	1670
	2000	104.3	1600	288.5	1200	410.3	1500	1682
	2400	101.2	1800	283.5	1400	419.9	1600	1680
	2800	96.7	2000	280	1600	417.0	1700	1675
	3200	89.4	2200	268.5	1800	406.7	1800	1655
	3600	83.0	2400	254	2000	391.7	1900	1630
	4000	71.8	2600	238	2200	374.1	2000	1600
	4400	60.0			2400	355.1	2100	1560
					2600	335.6	2200	1515
					2800	316.1	2300	1470
							2400	1420
62	2.58		18.8		27.7		125.0	
63	0.27		1.191		1.29		3.68	
	0.27		2.08		1.29		3.68	
			2.05		1.29		3.68	
					1.29		3.68	
					1.29		3.68	
							3.68	
64	3282.0		90876.0		68000.0		581700.0	
65	44.3		113.0		50.7		77.76	
	40.7		39.0		24.4		44.42	
			24.0		-1.8		11.08	
					-28.1		-22.26	
					-54.3		-55.60	
							-88.94	

TABLE B2 (cont'd)

Characteristics No.	M151	M35A2 Mod	Vehicles M113A1	M60A1
66	0.0	0.0	40.0	60.0
67	-4.303	-2.627	-3.75	-5.79
68	0.00342	0.006	-0.0087	-0.0089
69	-0.81656	-1.038	-0.76	-0.966
	-0.8377	-1.552	-0.78	-0.97
		-1.658	-0.76	-0.942
			-0.73	-0.913
			-0.68	-0.884
				-0.850
70	--	--	--	0.0
71	6.5	7.5	3.2	3.5
	2.7	4.5	0.9	1.0
	0.8	2.1	0.0	0.0
	0.0	0.6	0.9	1.0
	0.8	0.0	3.2	3.5
	2.7	0.6		
	6.0	2.1		
		4.5		
		7.5		
72	420.0	581.0	1500.0	3885.0
	565.0	716.0	2000.0	4715.0
	655.0	817.0	3500.0	5000.0
	685.0	878.0	2000.0	4715.0
	655.0	900.0	1500.0	3885.0
	565.0	878.0		
	420.0	817.0		
		716.0		
		581.0		
73	--	--	1500.0	3145.0
			700.0	1670.0
			0.0	0.0
			-700.0	-1670.0
			-1500.0	-3145.0
74	--	--	12.0	12.0
			10.0	10.0
			8.0	8.0
			6.0	6.0
75	--	--	175.0	375.0
76	--	--	300.0	300.0

APPENDIX C

COMPUTER PROGRAM

Appendix C contains a complete description of the computer program for the AMC model for predicting cross-country vehicle performance. A chart outlining the calling sequency of subroutines (Figure C1) and a master glossary of variable names are followed by descriptions of the main program and each subroutine in order, as listed below. Each subroutine description is followed by a flow chart (numbered figure) and a computer listing.

1. Calling Sequence
2. Glossary
3. Main Program FIVEYP
4. Subroutine INPUT
5. Power Train Submodel
    a. Subroutine STICK
    b. Subroutine AUTØF
6. Subroutine KURVE
7. Subroutine CURVE
8. Subroutines FØIL and CØIL
    a. Subroutine FØIL
    b. Subroutine CØIL
9. Subroutine PATCH
10. Subroutine MARSH
11. Subroutine HILL
12. Subroutine VISIØN
13. Subroutine AREAØ
14. Subroutine ØBSTCL
15. Subroutine ØBSF
16. Subroutine AREAV
17. Subroutine VEGF
18. Subroutine AREAT
19. Subroutine RIVER
20. Subroutine DIG
21. Subroutine VWRT
22. Subroutine RØUTE
23. Subroutine Data Files
24. Vehicle Ride Dynamics Submodel

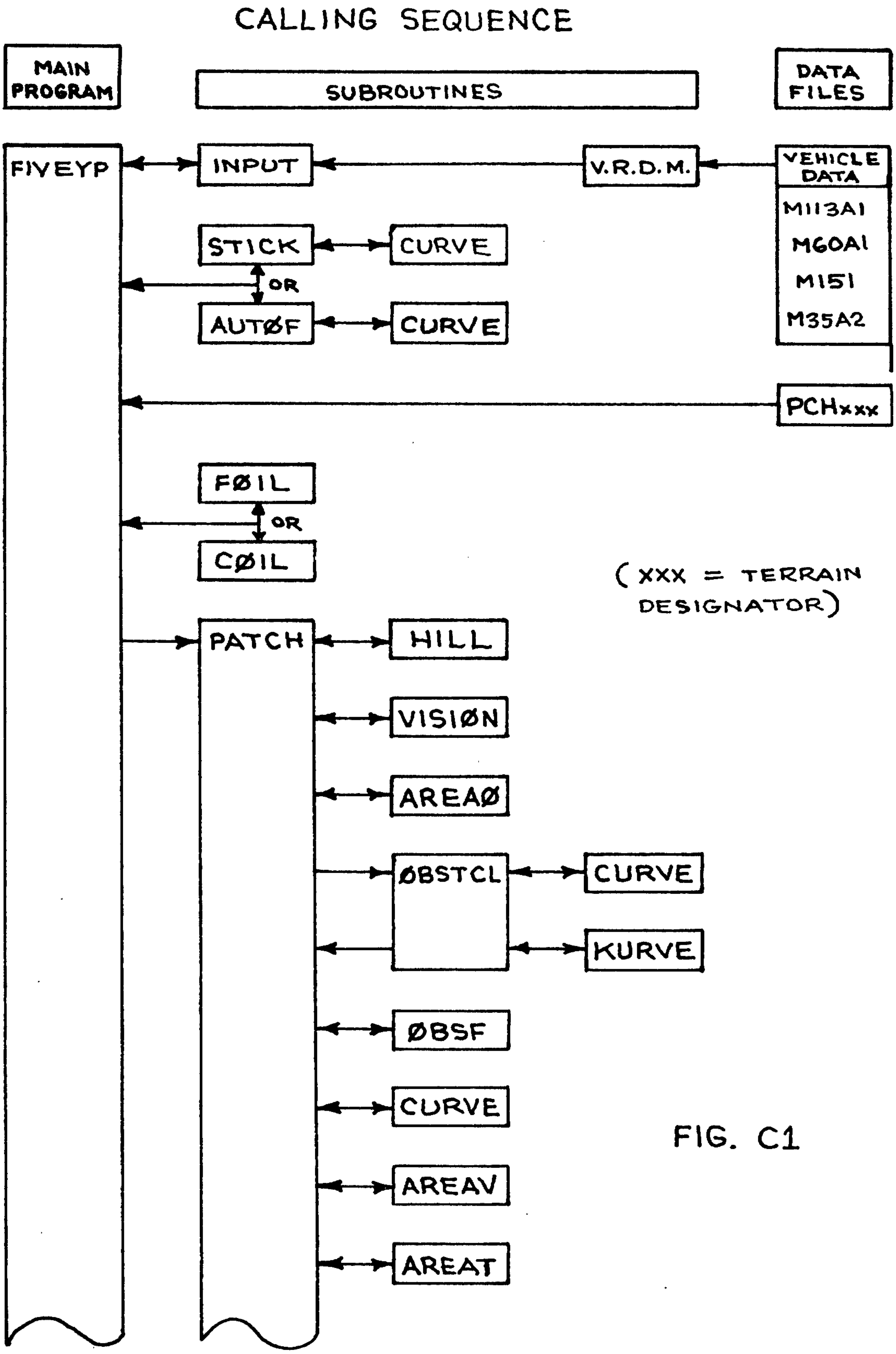

CALLING SEQUENCE
MAIN PROGRAM
SUBROUTINES
DATA FILES
FIVEYP
INPUT
V.R.D.M.
VEHICLE DATA
M113A1
M60A1
M151
M35A2
STICK
CURVE
OR
AUTØF
CURVE
PCHxxx
FØIL
OR
CØIL
(XXX = TERRAIN DESIGNATOR)
PATCH
HILL
VISIØN
AREAØ
ØBSTCL
CURVE
KURVE
ØBSF
CURVE
AREAV
AREAT

FIG. C1

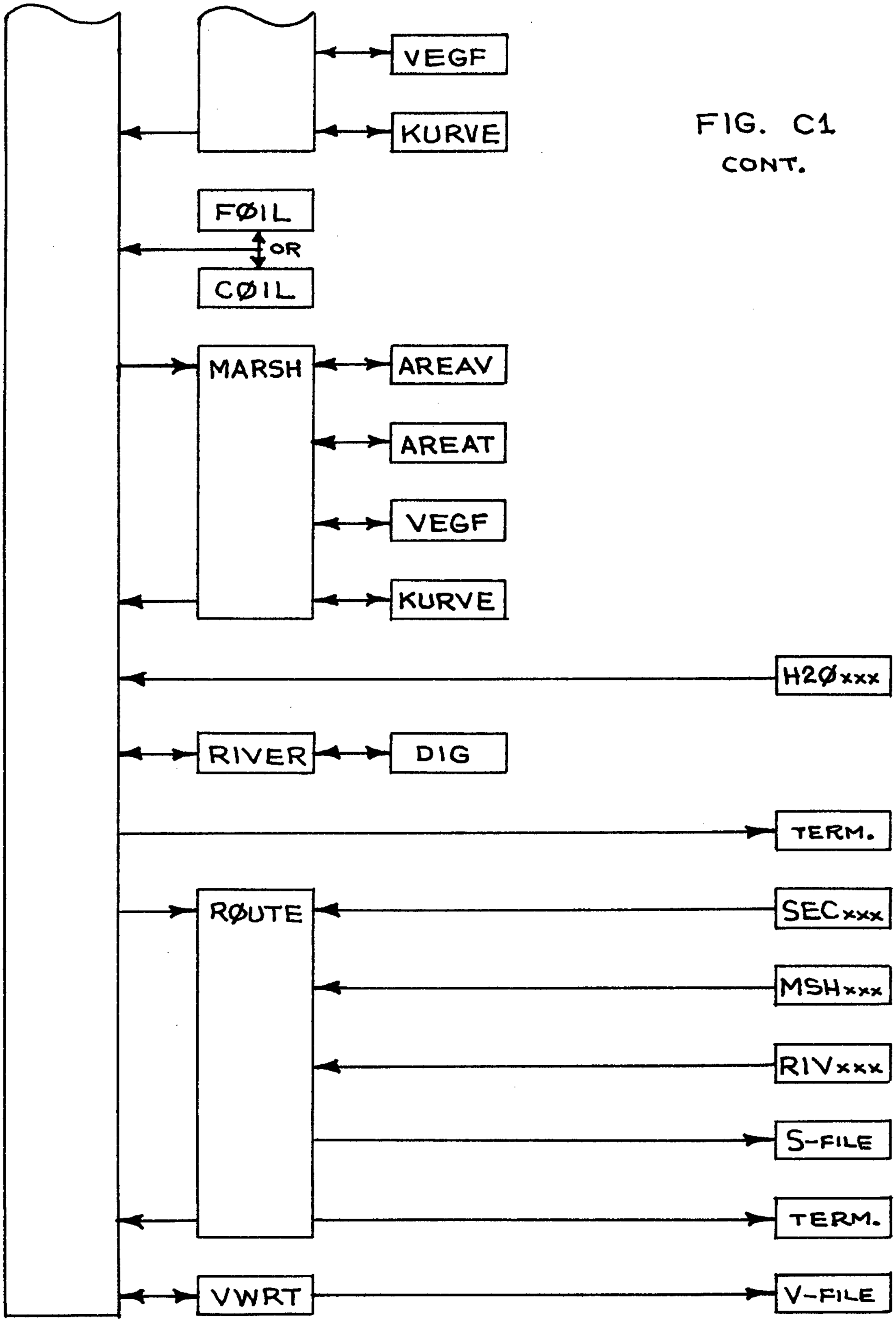

VEGF
KURVE
FØIL
OR
CØIL
MARSH
AREAV
AREAT
VEGF
KURVE
H2Øxxx
RIVER
DIG
TERM.
RØUTE
SECxxx
MSHxxx
RIVxxx
S-FILE
TERM.
VWRT
V-FILE

FIG. C1
CONT.

MASTER GLOSSARY

Variable Name	Used in Subroutine	Definition
A	ØBSTCL AREAØ	Obstacle approach angle (deg, rad) (≡ ØBAA in PATCH)
A	INPUT FØIL	Tracked: Area of one track shoe (in.^{2}) Wheeled: Number of axles
AA(I)	FIVEYP PATCH	Midpoint of obstacle approach angle Class I (deg)
AAV	RIVER	Vehicle approach angle (rad) (≡ AV in FIVEYP) (≡ in ØBSTCL, INPUT)
AAV1	RIVER	Vehicle approach angle plus 5 degrees (rad) (≡ THD in DIG)
ACC	VISIØN	Maximum vehicle acceleration (ft/sec^2)
ACCEL	PATCH	Total tractive force (lb) then changed to: vehicle acceleration (mph/sec).
ACG	ØBSTCL INPUT	Angle between a line parallel to the ground surface and the line connecting the CG and the center of the rear wheel (roadwheel or idler). The wheel is the one used to determine departure angle (rad).
ADØ	PATCH MARSH AREAØ AREAT	Percentage of area denied by obstacles
ADØ1	AREAØ	Area denied by one obstacle (ft^2)
ADT	AREAT	Percentage of area denied by both obstacles and vegetation
ANGLE	HILL	Slope angle (rad)
AREA(I)	RØUTE	Percentage of course area in which the vehicle can achieve speed range I I = 1, 2, 3, 4, 5, 6 RANGE (MPH) = 0-2, 2-4, 4-6, 6-8, 8-10, >10

Variable Name	Used in Subroutine	Definition
AREAØ	RØUTE	Percentage of course area in which the vehicle is immobilized.
ATM	ØBSTCL	= ATAN(MU)
ATP	RIVER DIG	Time penalty for excavating a river bank to allow egress (min)
AV	FIVEYP	Vehicle approach angle (rad) (≡ AAV in RIVER) (≡ VAA in ØBSTCL, INPUT)
AV, AV2, AV3	ØBSTCL	Vehicle angle with respect to level (rad) (see analysis)
AVGV	RØUTE	Average vehicle velocity over the course (mph)
AWPKF	INPUT RIVER	Auxilliary water propulsion factor - no = .5, yes = .8
A1	ØBSTCL	The maximum obstacle flank angle that the vehicle can climb (rad) (if less than A, the vehicle is immobilized in traction)
BA	PATCH	Maximum vehicle breaking deceleration (mph/sec)
BCA	ØBSTCL	Belly clearance angle (rad)
BD	FIVEYP	River bank differential height (ft)
BDC(I)	FIVEYP	Midpoint of river bank differential height class I (ft)
BH	DIG	River bank height (ft) (≡ BHI, ESLH in RIVER)
BHI	FIVEYP RIVER	River bank height (ft) (≡ BH in DIG)
BRFØR	PATCH	Maximum braking force (lb) (≡ TRØF in VISIØN)
C	FIVEYP RIVER	Soil cohesion
CA	ØBSTCL	= COS(A)
CAF	CØIL	Contact area factor used in mobility index calculation

Variable Name	Used in Subroutine	Definition
CAV	ØBSTCL	= COS(AV)
CA2	ØBSTCL	= COS(A/2.)
CF	FØIL CØIL	Correction factor used in slip calculation
CGF	ØBSTCL INPUT	Horizontal distance from the CG to the front wheel centerline (in)
CGH	ØBSTCL INPUT	Vertical distance from the CG to the road-wheel centerline (in)
CGR	ØBSTCL	Horizontal distance from the CG to the rear wheel centerline (in)
CLF	FØIL	Clearance factor used in mobility index calculation
CØNF1	PATCH	Conversion factor = 15./22. fps mph
CØNF2	PATCH	Conversion factor = 22./15. mph fps
CØURSE	RØUTE	Alphanumeric variable containing course file name (e.g.: = SEGPR1)
CPF	FØIL CØIL	Contact pressure factor used in mobility index calculation
CURV	RIVER	Raft capacity curve limit (lb)
CX	FØIL	Maximum $\frac{DP}{W_{20}}$ for given conditions
CXP	FØIL	Assymptote of $\frac{DP}{W_{100}}$ versus slip curve
DCG	ØBSTCL INPUT	Distance from the CG to the center of the rear wheel (roadwheel or idler). The wheel is that one used to determine departure angle (in.)
DFW	ØBSTCL	Horizontal distance from the front wheel centerline to the leading edge of the vehicle (in.)

Variable Name	Used in Subroutine	Definition
DIST	RØUTE	Total length of the course along one edge (inside edge if the course strip is folded) (ft)
DISTM	RØUTE	Total length of the course along one edge (inside edge if the course strip is folded) (miles)
DL	INPUT FØIL	Tracked: Length of track on the ground (in) Wheeled: Wheel diameter (in)
DØW	FØIL CØIL	Drawbar pull to weight ratio
DOW20	CØIL	Drawbar pull to weight ratio at 20 percent slip
DP(I)	RØUTE	Distance across the Ith patch traversed along a path segment
DR	VISIØN	Recognition or stopping distance (ft) (≡ RD(I) in PATCH)
DRW	ØBSTCL	Horizontal distance from the rear wheel centerline to the trailing edge of the vehicle (in)
DS	ØBSTCL	The greatest top trench width that the vehicle can bridge (in). Tracked = TI Wheeled = 2.*RW
DWX	FIVEYP RIVER	Maximum span between adjacent wheel center-lines (in) (≡ GWS in INPUT, ØBSTCL)
DW100	CØIL	Drawbar pull to weight ratio at 100 percent slip.
D1 thru D5	ØBSTCL	Critical distances (see analysis)
EA	ØBSTCL	The least of the vehicle angles of approach and departure (rad) min (VAA, VDA)

Variable Name	Used in Subroutine	Definition
EBH	RIVER	Effective height of first exit slope (ft).
EC	ØBSTCL	= FEC if VAA VDA = REC if VAA VDA
ED	ØBSTCL	= DFW if VAA VDA = DRW if VAA VDA
EF	FØIL	Engine factor used in mobility index calculation 10 hp/ton = 1. 10 hp/ton = 1.05
EFF	INPUT AUTØF STICK ØUTPT	Transmission efficiency
ENTCE	AUTØF	Engine to torque converter efficiency
ENTCG	INPUT AUTØF	Gear ratio from engine to torque converter
ESL	RIVER	Effective slope (rad) (≡ THN in <u>DIG</u>)
ESLH	RIVER	Effective slope height (ft) (≡ BH in <u>DIG</u>)
FACT	KURVE CURVE	Denotes whether the dependent variable increases or decreases as the independent variable increases
FAC7	CØIL	Tire factor used in vehicle cone index calculation
FAT	PATCH MARSH VEGF	Average force to override multiple trees (lb)
FAT1	PATCH MARSH VEGF	Average force to fell a single tree (lb)
FD	INPUT FIVEYP RIVER	Fording depth or draft height (in.)

Variable Name	Used in Subroutine	Definition
FDR	INPUT AUTØF STICK ØUTPT	Final drive gear ratio
FDREF	INPUT AUTØF STICK ØUTPT	Final drive gear efficiency
FEC	INPUT ØBSTCL	Vehicle front end clearance (in.) (vertical clearance of vehicle's leading edge)
FLAGM	AUTØF	Temporary minimum engine speed used in finding engine operating point
FLAGP	AUTØF	Temporary maximum engine speed used in finding engine operating point
FMT	PATCH MARSH VEGF	Maximum force to override a single tree (lb)
FØM	PATCH ØBSF	Average force to override obstacles (lb)
FØRCE(1,I)	AUTØF STICK FIVEYP FØIL	The Ith tractive force component (lb)
FØRCE(2,I)	CØIL ØUTPT	The Ith velocity component on the tractive effort versus vehicle velocity curve (mph) I = 1,101 (0-50 mph in 1/2 mph increment)
FØRCR(1,I)	FØIL CØIL PATCH MARSH	The Ith velocity component on the soil-dependent tractive effort versus vehicle velocity curve for level ground (mph)
FØRCR(2,I)		The Ith tractive force component for level ground (mph)

Variable Name	Used in Subroutine	Definition
FØRCR(3,I)	FØIL CØIL PATCH MARSH	The Ith velocity component on the soil-dependent tractive effort versus vehicle velocity curve on a slope (mph)
FØRCR(4,I)		The Ith tractive force component on a slope (lb) (≡ TABLE(J,I) in KURVE)
FØRK	FØIL CØIL	Tractive force (temporary variable)
FØRMX(I)	FØIL CØIL PATCH MARSH ØBSTCL	Maximum tractive force on slope I (max of array FORCR) I = 1 downhill 2 level ground 3 uphill (Limited by vehicle capacity and soil failure)
FT	AREAV	Temporary variable carrying value of XNT
FX	CURVE KURVE	The value of the dependent variable interpolated from the entering array
G	PATCH MARSH	Acceleration of gravity = 32.16 ft/sec^2
GC	INPUT FØIL RIVER ØUTPT	Vehicle ground clearance at the center of the greatest wheel span (in.) (= 1000. for tracked vehicles) (≡ BC in ØBSTCL)
GCA	INPUT RIVER	Ground contact area (in.^{2})
GF	FØIL	Grouser factor used in mobility index calculation: Wheeled: w/o chains = 1. w/chains = 1.05 Tracked: 1.5 in. high = 1 1.5 in. high = 1.1

Variable Name	Used in Subroutine	Definition
GR(I)	INPUT AUTØF STICK ØUTPT	Gear ratio of the Ith gear
GRADE	PATCH HILL FØIL CØIL	Percent grade (slope) (≡ SLC(I) in FIVEYP)
GRADI	PATCH	Percent grade (slope) (temporary variable)
GT	INPUT FØIL CØIL	Tracked: Grouser height (in.) Wheeled: Number of tires
GVW	INPUT FØIL CØIL PATCH MARSH RIVER ØBSTCL HILL ØBSF ØUTPT	Gross vehicle weight (lb)
GWS	INPUT ØBSTCL	Maximum span between adjacent wheel centerlines (in.) (≡ DWX in FIVEYP, RIVER)
H	PATCH ØBSTCL ØBSF	Obstacle height (in.) (≡ X in CURVE)
HC	ØBSTCL INPUT	Vertical distance from the ground to the center of the rear wheel (roadwheel or idler). The wheel is the one used to determine departure angle (in.). Wheeled = RW
HFT	ØBSF	Obstacle height (ft)

Variable Name	Used in Subroutine	Definition
HPT	INPUT FØIL ØUTPT	Vehicle rated horsepower per ton
HS	INPUT RIVER	Maximum vertical step that the vehicle can climb (in.)
H1 thru H9	ØBSTCL	Critical distances (see analysis)
I	RØUTE	The starting point of a path sgement through a course section (see J,K)
IAVE	FIVEYP	Alphanumeric variable containing "AVE"
IBEG	KURVE	Index denoting the first non-zero point in the column (of the entering array) designated as the dependent variable
IDRY	FIVEYP	Alphanumeric variable containing "DRY"
IFX	KURVE CURVE	Index designating which column in the entering array represents the dependent variable
IFØR	FØIL CØIL	Index denoting force component (see: FØRCE, FØRCR)
IGØ	FIVEYP FØIL CØIL PATCH ØBSTCL	Denotes "go" condition; no go = 0; go = 1 (negative values are used to indicate "no go" for various reasons)
IGR	PATINP FIVEYP PATCH	Percent slope class for given patch type
IND	PATCH MARSH AREAV VEGF	Temporary index
INDEX	RØUTE	Temporary index denoting velocity range

Variable Name	Used in Subroutine	Definition
IØBAA	PATINP FIVEYP	Obstacle approach angle class for given patch type
IØBH	PATINP PATCH	Obstacle height class for given patch type
IØBL	PATINP PATCH	Obstacle length class for given patch type
IØBS	PATINP FIVEYP PATCH	Obstacle spacing class for given patch type
IØBW	PATINP PATCH	Obstacle width class for given patch type
IØST	PATINP PATCH ØBSF AREAØ	Obstacle spacing type class for given patch type - random = 1, linear = 2
IP(I)	RØUTE	Type number of the Ith patch traversed along a path segment
IPR	PATINP PATCH	Microprofile type number for given patch type
IPX	RØUTE	Temporary index denoting patch type
IR	FIVEYP	Temporary index denoting RCI class for given season for given patch type
IRCI(I)	PATINP FIVEYP	RCI class for season I for given patch type
IREC	PATINP PATCH	Recognition distance class for given patch type
IS(I)	PATINP PATCH MARSH	Stem spacing class corresponding to stem diameter class I

Variable Name	Used in Subroutine	Definition
ISEAS	FIVEYP	Alphanumeric variable containing season name ("DRY", "AVE", "WET")
ISLØP	FØIL CØIL	Temporary index denoting slope
ISNI	FIVEYP	Temporary index denoting season DRY = 1 AVE = 2 WET = 3
ISRC(I)	RIVEYP PATCH	Surface roughness class corresponding to microprofile type I
IST	PATINP FIVEYP	Soil type class for given patch type fine-grained = 1 coarse-grained = 2
ITIME	RØUTE	Denotes which data file is to be read: 0 = SEGPR1 1 = MAPPR 2 = RIVPR
ITRAN	INPUT FIVEYP ØUTPT	Transmission type: stick = 0 automatic = 1
ITVAR	INPUT FØIL	Transmission variety: hydraulic = 0 mechanical = 1
IVEL	FØIL CØIL	Index denoting velocity component (see: FØRCE, FØRCR)
IVEL	AUTØF STICK PATCH MARSH KURVE ØUTPT	The number of point pairs in the velocity versus tractive force arrays (FØRCE and FØRCR) (_ 101)
IWET	FIVEYP	Alphanumeric variable containg "WET"
IX	KURVE CURVE	Index designating which column in the entering array represents the independent variable
J	RØUTE	The ending point of a path segment through a course section (see: I, K)

Variable Name	Used in Subroutine	Definition
JQ	RØUTE	The line number to be written into an external data file containing array "s" for the given vehicle
JX(I)	RØUTE	The number of path segment termination points on the line between course sections I-1 and I
JI	AUTØF STICK	Temporary index denoting the particular point pair in the tractive force versus velocity array (FØRCE)
K	RØUTE	The course section being traversed (see: I,J)
LØKUP	INPUT	Denotes presence of a torque converter lockup - no = 0 yes = 1
MD	VEGF	Maximum stem diameter class to be overridden
MSD	AREAV	Minimum stem diameter class to be avoided
MDSM1	AREAV	= MSD - 1
MU real	ØBSTCL	Coefficient of rolling friction
N	KURVE CURVE	Denotes which columns in the entering array are to be designated as dependent and independent variables (see: IX, IFX)
N	RØUTE	The number of patches traversed on a path-segment
NBC	FØIL CØIL INPUT	Tracked: Number of bogies Wheeled: Denotes presence of chains (no = 0, yes = 1)
NBDC	FIVEYP	River bank differential height class for given river type

Variable Name	Used in Subroutine	Definition
NCREW	INPUT RIVER DIG	Number of people in the vehicle on a normal situation
NC1	INPUT	Number of point pairs in array TNE1
NC2	INPUT	Number of point pairs in array TTM
NC3	INPUT AUTØF STICK	Number of point pairs in array TTE
NC4	INPUT PATCH	Number of point pairs in array VØØB (≡ in CURVE)
NC5	INPUT	Number of points in array VRIDE
NE real	AUTØF	Engine speed (rpm) (≡ in CURVE)
NEI real	AUTØF	Average of current values of FLAGM & FLAGP (rpm)
NEMAX real	AUTØF STICK	Maximum engine speed from engine torque curve (rpm)
NEMIN real	AUTØF STICK	Minimum engine speed from engine torque curve (rpm)
NESC	FIVEYP	River egress bank angle class for given river type
NEX	FIVEYP RIVER	Number of distinct slopes on the egress bank of a river (= 1 for this generation)
NE1 real	AUTØF	Temporary engine speed (≡ FX in CURVE)
NFL	INPUT CØIL	Track type: not flexible = 0, flexible = 1

Variable Name	Used in Subroutine	Definition
NG	INPUT AUTØF STICK ØUTPT	Number of gear ratios in the transmission
NGEAR	AUTØF STICK	Temporary index denoting transmission gear ratio
NISC	FIVEYP	River ingress bank angle class for given river class
NØDE(I,J)	RØUTE	The point on line I+1 (next line toward the destination) along the best route from point J on line I to the destination
NØDEF(I)	RØUTE	The point on line I along the finally selected best route through the course
NØP	RØUTE	Number of points on a line of the grid overlay for the course map
NØS	RØUTE	The number of sections into which the grid overlay for the course map is divided
NØSM1	RØUTE	= NOS - 1
NØSM2	RØUTE	= NOS - 2
NPAT	PATINP FIVEYP	The type number of the particular patch or river being traversed
NPATCH	FIVEYP	Total number of patch types (including marshes)
NRIV	FIVEYP	Total number of river types
NRWC	FIVEYP	River width class for given river type
NSDC	FIVEYP PATINP PATCH MARSH AREAV	Number of stem diameter classes

Variable Name	Used in Subroutine	Definition
NSDCM	PATCH MARSH	= NSDC - 1
NSDCP	PATCH MARSH	= NSDC + 1
NSSC	FIVEYP PATCH MARSH	Number of stem spacing classes
NT real	AUTØF STICK	Transmission input speed (rpm)
NT	ØBSTCL	Denotes obstacle type: ridge = 1 trench = 2
NV	ØBSTCL	= NVEH + 1
NVEH	INPUT FØIL CØIL ØBSTCL RIVER	Denotes vehicle type: tracked = 0, 4x4 = 1, 6x6 = 2, 8x8 = 3
NWDC	FIVEYP	Water depth class for given river type
NWV	FIVEYP	Water speed class for given river type
ØBAA	PATCH	Obstacle approach angle (deg) (≡ A in ØBSTCL)
ØBL	PATCH AREAØ ØBSF	Obstacle length (ft)
ØBS	PATCH AREAØ ØBSF	Obstacle spacing (ft)
ØBW	PATCH AREAØ	Cross-sectional width at the bottom of an obstacle (ft) (≡WB in ØBSTCL)

Variable Name	Used in Subroutine	Definition
ØH(I)	FIVEYP PATCH	Midpoint of obstacle height class I (in)
ØL(I)	FIVEYP PATCH	Midpoint of obstacle length class I (ft)
ØS(I)	FIVEYP PATCH	Midpoint of obstacle spacing class I (ft)
ØW(I)	FIVEYP PATCH	Midpoint of obstacle width class I (ft)
P	RØUTE	Total time of the best route through the course (min) (see: SUMTØT)
PAV	PATCH MARSH AREAV AREAT	Percentage of area denied by vegetation
PBF	INPUT PATCH MARSH	Maximum force that the vehicle pushbar can withstand (lb)
PBHT	INPUT PATCH MARSH VEGF	Height of vehicle pushbar (in.)
PC(I)	FIVEYP	Percentage of patches in which the vehicle can achieve speed range I I = 1 2 3 4 5 6 7 Range (mph) = 0 0-2 2-4 4-6 6-8 8-10 10
PI	AUTØF STICK RIVER AREAØ ØBSF VEGF	= 3.14159265

Variable Name	Used in Subroutine	Definition
PID	INPUT ØBSTCL	Conversion factor = PI/180 (deg rad)
PØDE(I,J)	RØUTE	The time along the best route from point J on line I to the destination (min)
P1	RØUTE	Temporary variable containing route time (min)
RBA(I)	FIVEYP RIVER	River bank angle I (consecutively up the bank) (deg)
RBH(I)	FIVEYP RIVER	River bank height I (consecutively up the bank) (ft)
RCI	FIVEYP FØIL CØIL	Rating cone index of the soil
RCIC(I)	FIVEYP	Midpoint of soil RCI class I
RCIX	FØIL	Excess RCI (above one-pass vehicle cone index)
RD(I)	FIVEYP PATCH	Midpoint of recognition distance class I (ft) (≡ DR in VISIØN)
RDIAM	INPUT CØIL	Wheel rim diameter (in.)
REC	ØBSTCL	Rear end clearance (in) (vertical clearanc of vehicle's trailing edge)
RGU(I)	PATCH HILL	Total resisting force due to soil and slop for slope I (lb) I = 1 downslope = 2 level ground = 3 upslope
RR	INPUT AUTØF STICK	Tracked: Sprocket pitch radius (in.) Wheeled: Tire rolling radius (in.)

Variable Name	Used in Subroutine	Definition
RT	PATCH MARSH FØIL CØIL HILL	Soil resistance (lb)
RTØW	FØIL CØIL	Resistance to weight ratio
RTS	PATCH HILL	Soil resistance on slope (lb)
RW	ØBSTCL	Tracked: Road wheel radius plus track thickness (in.) Wheeled: Tire rolling radius (in.)
RW	FIVEYP RIVER	River width (ft)
RWC	ØBSTCL	= RW * (1.-COS(A))
RWC(I)	FIVEYP	Midpoint of river width class I
RWT	ØBSTCL	= RW * TAN (A/2.)
RWW	ØBSTCL INPUT	Track thickness plus the radius of the rear wheel (roadwheel or idler). The wheel is the one used to determine departure angle (in.). Wheeled = RW.
S(I)	PATCH MARSH AREAV	Mean spacing of all stems in stem diameter class I or larger (ft)
S(I,J,K)	RØUTE	Time required to traverse the path segment from point I on the starting line to point J on the ending line of course section K (min.)
SA	ØBSTCL	= SIN (A)

Variable Name	Used in Subroutine	Definition
SAI	INPUT RIVER	Ingress swamp angle of vehicle (deg, rad) (≡ THD in DIG)
SAV	ØBSTCL	= SIN(AV)
SA2	ØBSTCL	= SIN(A/2.)
SD(I)	FIVEYP PATCH MARSH AREAV VEGF	Midpoint of stem diameter class I (in.)
SDA	AREAV	Average stem diameter to be avoided (in.)
SDL(I)	FIVEYP PATCH MARSH VEGF	Upper limit of stem diameter class I (in.)
SDM	VEGF	Maximum stem diameter to be overridden (in.)
SDS(I)	PATCH MARSH VEGF	Mean spacing for stem diameter class I (ft)
SF	CØIL	Strength factor used in vehicle cone index calculations
SF	RIVER	River bank severity factor (ft) (see: SFI)
SF	RØUTE	Scale factor (= 1. this generation) (to be changed if data is given in meters instead of ft)
SFI	RIVER	River bank severity factor (in.) (see: SF)
SLC(I)	FIVEYP PATCH	Midpoint of slope class I (percent slope) (≡ GRADE in FØIL, CØIL)
SLIP	FØIL CØIL	Percent slip of the vehicle tractive element in the soil
SR	AUTØF	Torque converter speed ratio (≡ X in CURVE)

Variable Name	Used in Subroutine	Definition
SR1	AUTØF	Torque converter speed ratio
SRF	PATCH MARSH AREAT	Speed reduction factor due to maneuvering
SUMI	AREAV	Sum of diameters of all trees in an area containing one tree of the largest size (in.)
SUMT	AREAV	Number of all trees in an area containing one tree of the largest size
SUMTØT	RØUTE	Total time of the best route through the course (hr) (see: P)
SV(I)	FIVEYP PATCH MARSH	Midpoint of stem spacing class I (ft)
T	RØUTE	Total time penalty associated with river crossings and exits on one path segment (min)
TA	ØBSTCL	= TAN(A)
TABLE (I,J)	CURVE KURVE	The set of point pairs on the curve to be interpolated
TAD	PATCH	Time available for deceleration (sec)
TANP	FIVEYP RIVER	Tangent of the angle of repose of the soil (ø)
TAV	ØBSTCL	= TAN(AV)
TA2	ØBSTCL	= TAN(A/2.)
TC	INPUT AUTØF	Input torque at which the torque converter curves were measured (ft-lb)

Variable Name	Used in Subroutine	Definition
TDIST	PATCH	Distance that the vehicle has traveled since the last obstacle encounter (ft)
TDIST	RØUTE	Total area of the course (ft) (sum of lengths of all path segments whose starting point, I, equals the ending point, J)
TE	AUTØF STICK	Engine torque from net engine torque curve (ft-lb) (≡ FX in CURVE)
TEF	ØUTPT	Total power train efficiency
TF	FØIL	Track factor or tire factor used in mobility index calculation
TFAT	VEGF	Summation of the work required to override all diameters of trees to be run over
TFØR(1)	FØIL CØIL PATCH	Maximum tractive force downhill (lb)
TFØR(2)		Maximum tractive force on level ground (lb)
TFØR(3)		Maximum tractive force uphill (lb) (limited by soil failure only) (see: FØRMX)
TFØR	VISIØN	Maximum braking force (lb) (≡ BRFØR in PATCH)
TFØW	CØIL	Tractive force to weight ratio
THD	DIG	Angle of bank to be excavated to permit egress (rad) (≡ AAV, SAI, THM, THEM in RIVER)
THEM	RIVER	Traction-limited slope (rad) (≡ THD in DIG)
THI	FIVEYP RIVER	River ingress bank angle (rad) (≡ THN in DIG)

Variable Name	Used in Subroutine	Definition
THIC(I)	FIVEYP	Midpoint of river bank angle class I (deg)
THM	RIVER	Maximum drop-off angle before belly hang-up (rad) (≡ THD in DIG)
THN	DIG	River bank angle (to be excavated) (≡ THI, ESL in RIVER)
TI	AUTØF	Torque input to converter necessary to produce desired vehicle speed (ft-lb)
TI	ØBSTCL	(T inside) = GWS (wheeled) = min (CGF, CGH) (tracked) (see: TØ)
TL	INPUT ØBSTCL	Distance between first and last wheel centerlines (in.)
TN	FIVEYP RIVER	Time required to cross a river (swimming, fording, or rafting) excluding ingress and egress time (min) (≡ VR(I) in FIVEYP)
TND	PATCH	Time needed for deceleration (sec)
TNE1(I,J)	INPUT AUTØF	Array containing torque converter input speed versus converter speed ratio curve
TØ	ØBSTCL	(T outside) = TL (see: TI)
TØS	AUTØF STICK	Transmission output speed (rpm)
TØT	FIVEYP	Total number of patches in which the vehicle can achieve a speed greater than 0 but less than 10 mph
TØT	AUTØF STICK	Transmission output torque (ft-lb)
TP	FIVEYP RIVER	Total time penalty for river ingress and egress (min) (≡ TPR(I) in FIVEYP)

Variable Name	Used in Subroutine	Definition
TPLY	INPUT CØIL	Tire ply rating
TPR(I)	FIVEYP RØUTE	Total time penalty for ingress and egress from river type I (min) (≡ TP in RIVER)
TPSI	INPUT CØIL	Tire pressure (psi)
TRAT	RIVER	Traction based on c and ø of slope
TRF	AUTØF STICK	Tractive force that the vehicle can produce (ft-lb)
TRFU	PATCH MARSH	Total motion resistance due to soil, slope, obstacles and vegetation (lb)
TTE(I,J)	INPUT AUTØF STICK	Array containing net engine torque versus engine speed curve (ft-lb, rpm)
TTIME	PATCH	Time that the vehicle has traveled since the last obstacle encounter (sec)
(TTM(I,J))	INPUT AUTØF	Array containing torque converter torque multiplying coefficient versus converter speed ratio curve
TV	RIVER	Speed made good in crossing a river (corrected for downstream drift, mph)
TVEL1	PATCH	Current speed during speed-up/slow-down model (mph)
TVEL2	PATCH	Temporary variable containing velocity (mph)
TXF	FØIL	Transmission factor used in mobility index calculation: hydraulic = 1, mechanical = 1.05

Variable Name	Used in Subroutine	Definition
T1	RØUTE	Total time required for crossing all rivers encountered on a given path segment (excluding ingress and egress time, min)
T2	RØUTE	Total time penalty for river ingress and egress on all rivers encountered on a given path segment (min)
V(I)	FIVEYP RØUTE	The finally-selected limiting velocity on patch type I (fps)
VAA	INPUT ØBSTCL	Vehicle approach angle (deg, rad) (≡ AV in FIVEYP) (≡ AAV in RIVER)
VCI1	FØIL CØIL	One-pass vehicle cone index
VDA	ØBSTCL INPUT	Vehicle departure angle (rad)
VEH	ØUTPT	Alphanumeric variable containing the vehicle name (e.g.: M113A1, M60A1, M115A1, M35A2) (≡ VEHICL in FIVEYP)
VEHICL	FIVEYP INPUT	Alphanumeric variable containing the vehicle name (e.g.: M113A1, M60A1, M115A1, M35A2) (≡ VEH in ØUTPT)
VEL	AUTØF STICK	Temporary variable containing vehicle velocity (= 0. mph to 50. mph in .5 mph steps)
VELØ(I)	PATCH	The finally-selected best-achievable velocity on the given patch type with slope I (mph) I = 1 downslope = 2 level ground = 3 upslope
VELØC	FIVEYP PATCH MARSH	The finally-selected best-achievable velocity on the given patch type (assuming that equal distances will be traversed downslope, level and upslope, mph)

Variable Name	Used in Subroutine	Definition
VELV	VISIØN	Velocity limited by visibility (mph) (≡ VELV(I) in PATCH)
VELV(I)	PATCH	Velocity limited by visibility on slope I (mph) (≡ VELV in VISIØN)
VF	ØBSTCL	Velocity limited by soil and slope resistance (mph)
VFS	INPUT RIVER	Vehicle fording speed (mph)
VL	INPUT PATCH ØBSTCL ØUTPT	Vehicle length (ft) (in)
VM	PATCH MARSH VISIØN	Vehicle mass (lb-sec^2/ft)
VMTEM	PATCH MARSH	Most limiting speed between obstacles (mph)
VØLA	PATCH ØBSTCL	Limiting speed over an obstacle limited by 2.5g acceleration (mph)
VØØB(I,J)	INPUT PATCH	Array containing vehicle velocity versus obstacle height at 2.5g vertical acceleration (mph, in.)
VR(I)	FIVEYP RØUTE	Time required to cross river type I (swimming, fording, or rafting) excluding ingress and egress time (min) (≡ TN in RIVER)
VRID	PATCH	Limiting speed due to vibration at the driver's seat for a given patch type (mph)
VRIDE(I)	INPUT PATCH	Limiting speed due to vibration at the driver's seat for surface roughness class I (mph)
VSS	INPUT FIVEYP RIVER	Vehicle swimming speed (mph)

Variable Name	Used in Subroutine	Definition
VTEM(I,J,K)	PATCH	Final limiting velocity: I - slope type, J - over or around obstacles, K - over or around trees (mph)
VTT	PATCH	Force limited speed (mph)
VV	ØBSTCL	Finally-selected limiting velocity used to calculate the vehicle's kinetic energy when encountering an obstacle (mph) = Min (VF, VØLA)
VW	FØIL CØIL	Vehicle weight normal to the ground surface (lb)
VX	DIG	Volume of dirt to be excavated (ft^3)
W	INPUT PATCH MARSH AREAV AREAØ VEGF ØBSF RIVER DIG ØUTPT	Vehicle width (in.)
WA	AREAØ	Width of an obstacle at ground level (ft) (mound = bottom width; trench = top width)
WB	ØBSTCL	Cross-sectional width at the bottom of an obstacle (ft) (≡ ØBW in PATCH)
WBI	ØBSTCL	Cross-sectional width at the bottom of an obstacle (in.) (see: WB)
WC	INPUT RIVER	Vehicle winch capacity (lb)
WD	FIVEYP RIVER	Water depth of a river (ft)
WDC	FIVEYP	Midpoint of water depth class I (ft)

Variable Name	Used in Subroutine	Definition
WDF	CØIL	Wheel diameter factor used in vehicle cone index calculation
WF	FØIL	Weight factor used in mobility index calculation
WID	INPUT FØIL CØIL	Track width or wheel width (in.)
WLØRBF	FØIL	Wheeled: Wheel load factor Tracked: Bogie factor Used in mobility index calculation
WR	FØIL	Vehicle weight range (lb)
WS	FIVEYP RIVER	Water speed of a river (fps)
WSC(I)	FIVEYP	Midpoint of river water speed class I (fps)
WX	FØIL	Vehicle weight range (kip)
X	KURVE CURVE	The value of the independent variable to be used in interpolating the entering array
XBR	INPUT PATCH	Maximum braking force that the vehicle can develop (lb)
XJH	FIVEYP	Upper limit of speed range I (mph) (see: PC(I))
XJL	FIVEYP	Lower limit of speed range I (mph) (see: PC(I))
XMI	FØIL	Vehicle mobility index
XNT(I)	PATCH MARSH AREAV VEGF	Number of trees of stem diameter class I in an area containing one tree from class NSDC (the largest class)

Variable Name	Used in Subroutine	Definition
XX	FØIL	Temporary variable used in calculating drawbar pull to weight ratio
X1 thru X8	ØBSTCL	Critical distances (see analysis)
Y	FØIL	Temporary variable used in slip calculation

VARIABLES USED IN RIDE DYNAMICS SUBMODEL (VRDM)

FORCW(I)		Resultant vertical force at I^{th} axle due to profile input to segmented wheel
FORCH(I)		Resultant horizontal forces at I^{th} axle due to profile input to segmented wheel
FORCT(I)		Track interconnecting vertical force between the $(I-1)^{th}$ and the I^{th} axles
FORCK(I)		Suspension spring vertical force of the I^{th} axle
SPDEF(I)		Deflection of the suspension spring of the I^{th} axle
DSPDF(I)		Time derivative of SPDEF or relative velocity of I^{th} suspension
THRESH(K)		Threshold height of the K^{th} wheel segment, $KK\sin\theta_K$
SIGMA(K)		Horizontal spring constant for the K^{th} wheel segment, $KK\sin\theta_K$
GAMMA(K)		Vertical spring constant for the K^{th} wheel segment, $KK\cos\theta_K$
VAR(N)		Solution of N^{th} first order vehicle differential equation
Y		Array containing those interpolated profile elevations beneath the vehicle
PASTP		Array containing old input profile time history
PROFIL		Array containing new input profile time history

Variable	Description
PWRVAR(N)	Solution of N^{th} absorbed power first order differential equation
DAMP(I)	Damping force of I^{th} axle suspension
ACCISS(M)	Acceleration in in/sec^2 (except pitch) of the M^{th} output variable
ACCGS(M)	Acceleration in g's (except pitch) of the M^{th} output variable
ACCMAX(M)	Maximum acceleration of the M^{th} output variable
ACCMIN(M)	Minimum acceleration of M^{th} output variable
SUMRMS(M)	$\sum [ACCGS(M)]^2$ for RMS
RMS(M)	$\sqrt{\sum [ACCGS(M)]^2/t}$ RMS of M^{th} output variable
LEN	Read in values for various length parameters of vehicle. These lengths represent different dimensions in various vehicles. (See the individual vehicle for the use of a particular LEN)
MASS(I)	Mass of the I^{th} axle
H	RKG step size (seconds)
T	Time (sec)
DELTAT	Increment of time (sec)
DELTAL	Increment of length between profile points (in.)
VELIPS	Vehicle velocity (ips)
VELMPH	Vehicle velocity (mph)
NSTEPS	Number of steps in RKG for each DELTAT ($\Delta t/H$)

Variable	Description
YIN	Next profile input (in.)
DRVMAX	Maximum vertical acceleration of driver
DRVMIN	Minimum vertical acceleration of driver
ABSPWR	Absorbed power (watts) of driver
DRIVER(1)/DISDRV	Displacement of driver (in.)
DRIVER(2)/VELDRV	Velocity of driver (in.)
DRIVER(3)/ACCDRV	Acceleration of driver (g's)
DRIVER(4)/RMSDRV	RMS of driver (g's)
IOPT(1)/IFPWR	"Yes" for absorbed power
IOPT(2)/IFFILE	"Yes" for output file
IOPT(3)/IFPACC	"Yes" for peak acceleration
IOPT(4)/IFDRV	"Yes" for driver motions
IOPT(5)/IFRMS	"Yes" for RMS
IOPT(6)/IFINPT	"Yes" for external input
NY	Number of profile points needed to run complete length of vehicle. $\Delta L*NY$ = length of vehicle in inches.
IDF	Number of degrees of freedom (number of second order differential equations)
NAXLES	Number of axles for vehicle
NSEGS	Number of segments in each segmented wheel
IFHORZ	0 = No horizontal equation, 1 = horizontal equation
FNAME	Name of output file
FMASS	Mass of main frame of vehicle

Variable	Description
INRTIA	Pitch inertia of sprung mass
HORMOM	Sum of the moments due to axles about center of gravity
VEHQID	Name of vehicle being run
FID	Identification of input
IOPT	Program options (see individual options)
DRIVER	Driver variables (see individual variables)

THE FOLLOWING VARIABLES ARE USED BY THE MAIN PROGRAM ONLY

IYES	1HY
NO	1HN
IBELL	Rings TTY bell upon completion of test
SDVRMS	$\sum(\text{ACCDRV})^2$, summation to compute driver rms
JSTOP	1 = go, 2 = stop main program
NSTOP	Program terminates after NSTOP steps subsequent to JSTOP becoming 2
TPRINT	Controls TTY printout time

Main Program FIVEYP (Fig. C2)

The program which controls the calling of all subroutines is called FIVEYP. As the main program starts, the mid-points of the various stem-diameter classes of vegetation are calculated. This is necessary because the data presented in the data blocks consist only of the maximum points of each of these classes. (It was necessary to have the data in this form because they are used in this form later in subroutines AREAV and VEGF.) When this is completed, the program proper begins.

First, a call is made to the terminal to determine the vehicle, the geographic area and the season for which calculations will be performed. Next, the subroutine INPUT is entered. Subroutine INPUT calls the particular vehicle data file and loads all its data into the appropriate variables for later use. Then, the variable ISNI is created; it has the value 1, 2 or 3, depending upon whether the season is dry, average or wet. Next, a check is made to see if the vehicle has a standard or an automatic transmission. If it is standard, subroutine STICK is entered, and an array FØRCE is calculated. This array contains vehicle speed versus tractive effort on a solid surface, and has 101 elements, representing 0 to 50 mph in 0.5-mph increments. If the transmission is automatic, subroutine AUTØF is entered instead of subroutine STICK, and the array FØRCE is calculated.

Next, the major part of this program is entered. In this portion, the data for each patch are determined, and several other subroutines are called to determine the speed over that patch. Each patch is handled independently.

First, the data relating to class intervals for one patch are read. The variable IGR is checked. If this variable is equal to 0, the patch is a marsh; if it is greater than 0, the patch is "normal". If the patch is normal, the calculation proceeds and a check is made on variable IST. This is the soil type, which can be either fine-grained or coarse-grained. Depending on this, either subroutine FØIL or subroutine CØIL is called. A check is first made to determine if the soil and slope are the same as for the previous patch; if so, this call is bypassed.

Within these subroutines the array FØRCE is modified by slippage of the track or wheels in the soil, and the new array FØRCR is created. This array contains the actual vehicle speed versus soil-dependent tractive effort. Once this array is determined, another check is made within the subroutine to determine whether the soil is immobilizing the vehicle; i.e., whether the maximum tractive effort is positive. This is stored in variable IGØ. If IGØ returns as 0, the vehicle is immobilized, and the speed in this patch is set to 0. If IGØ is 1, the vehicle can successfully negotiate the soil, and a call to subroutine PATCH is made. Subroutine PATCH calls several other subroutines to calculate various resistances, avoidances, overriding of obstacles, vegetation, slopes, vision and other elements. Returning from PATCH is a limiting speed for this patch type in variable VELØC. This is loaded into array V and a return is made to read data for the next patch.

If the patch is a marsh, instead of a normal patch, control is transferred to another portion of the program. The soil-dependent tractive effort versus speed curve must be determined here too; and depending upon variable IST, the soil type, either subroutine FØIL or subroutine CØIL is called, and this array is calculated. Also, variable IGØ returns from FØIL or CØIL. If IGØ is 0, the vehicle is immobilized in the soil, and array V is set to 0 and a return is made. If the vehicle is not immobilized, subroutine MARSH is called. As with PATCH, subroutine MARSH calculates the resistances, avoidances, etc., for the terrain elements in this marsh, and the variable VELØC is returned containing the limiting speed. This is loaded into array V, and a return is made to read additional patch data.

The last line in the patch data file contains a check that allows an exit from this file. The check is made on variable IØBAA. If this variable is negative, the patch file is closed. This last line will be the only one in that file with a negative element in this location.

The river data file is then opened. A loop is entered to calculate the time penalties associated with crossing each river type. First, the various river data are calculated from the class intervals indicated in the data file. These data are: water speed, WS; water depth, WD: bank angle, THI;

bank differential height, BD; bank height in ingress, BHI; and bank height in egress, RBH. Also, the egress bank anble, RBA, is calculated, and the number of distinct slopes on the egress bank, NEX. For the AMC 71 Model, NEX is always 1. Finally, the river width, RW, is calculated. Then subroutine RIVER is entered. The time penalties associated with crossing the river with ingress and egress os the river are calculated within this subroutine. These return from the subroutine as variables TP and TN, and they are loaded into arrays VR for river-crossing time and TPR for river ingress and egress time penalty. This is performed successively for each river type. The last line in the river data file is an exit line; and when this is reached, an exit is performed to a later part of the main program.

Three arrays are filled now: V for the average speed calculated for each patch type, VR for the time penalty associated with crossing each river, and TPR for the time penalty associated with ingress and egress from each river.

The next part of the program calculates the percentage of patches in which the vehicle is constrained to certain speed ranges. These ranges are: 0 (immobilization), 0 to 2, 2 to 4, 4 to 6, 6 to 8, 8 to 10, and greater than 10 mph. This information is loaded into array PC, and this array is printed by the terminal.

Subroutine ØUTPT is entered next. This prints the various vehicle data that were calculated in earlier subroutines of the program. Next, subroutine RØUTE is entered. This subroutine takes the arrays V, VR, and TPR and calculates the best route through the map. Subroutine VWRT is then entered and the arrays V, VR, and TPR are written into an external file for later manipulation, if required. This file is named VELFIL within the program. Later, when the program has been run, this file will be called up and given a new name to identify it with the specific vehicle. This is the last thing accomplished in the main program; and at this point, the run is complete.

MAIN PROGRAM

VARIABLES INITIATED BY DATA BLOCK: IDRY, IAVE, IWET, RD(9), NSDC, NSSC, SDL(8), SV(8), ØH(7), ØW(5), ØL(7), ØS(8), AA(14), RCIC(11), SLC(8), WSC(5), WDC(4), THIC(10), BDC(9), RWC(24), ISRC(9), IFEAT(5)

VARIABLES ENTERING THROUGH COMMON: VSS, WD, ITRAN, AV, DWX

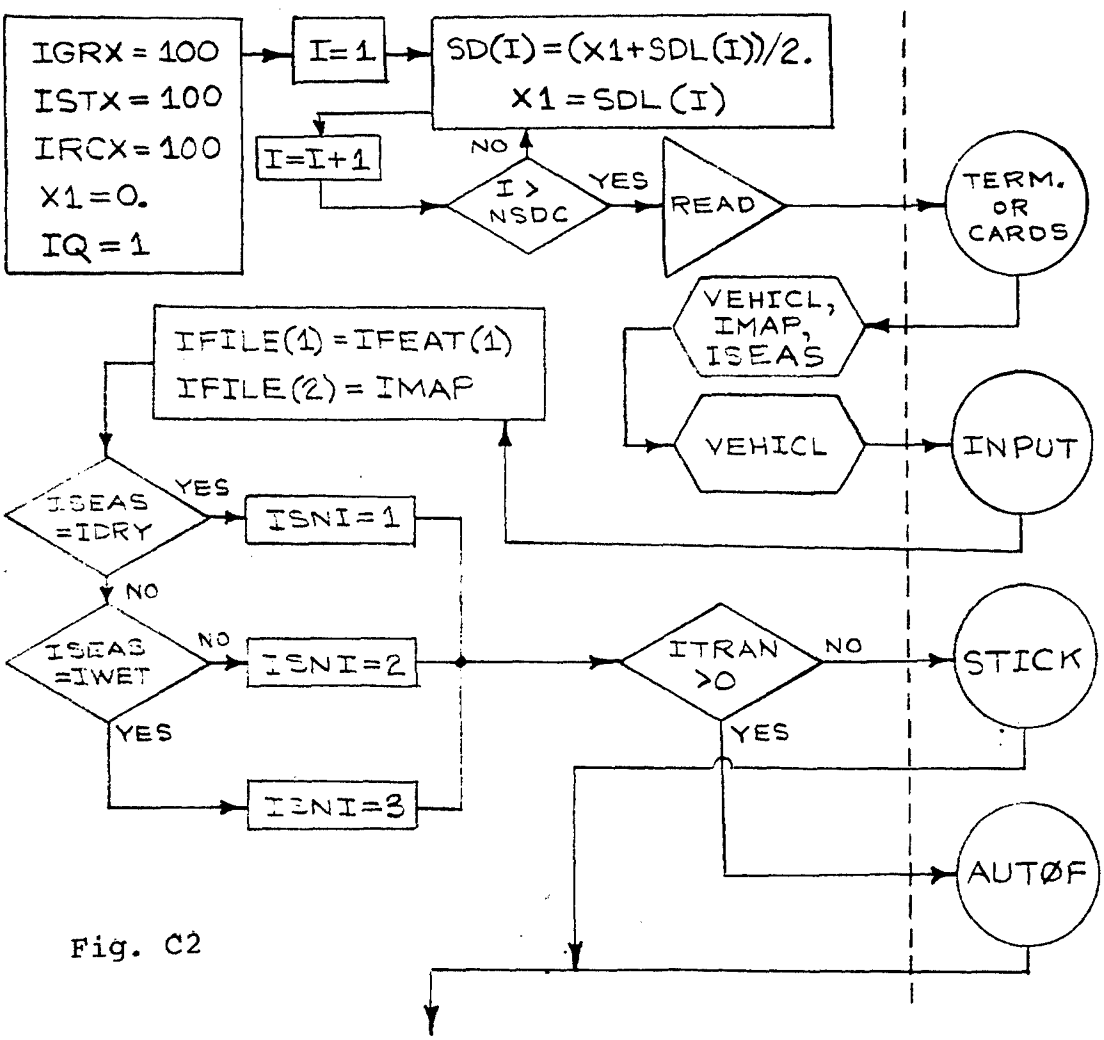


Fig. C2

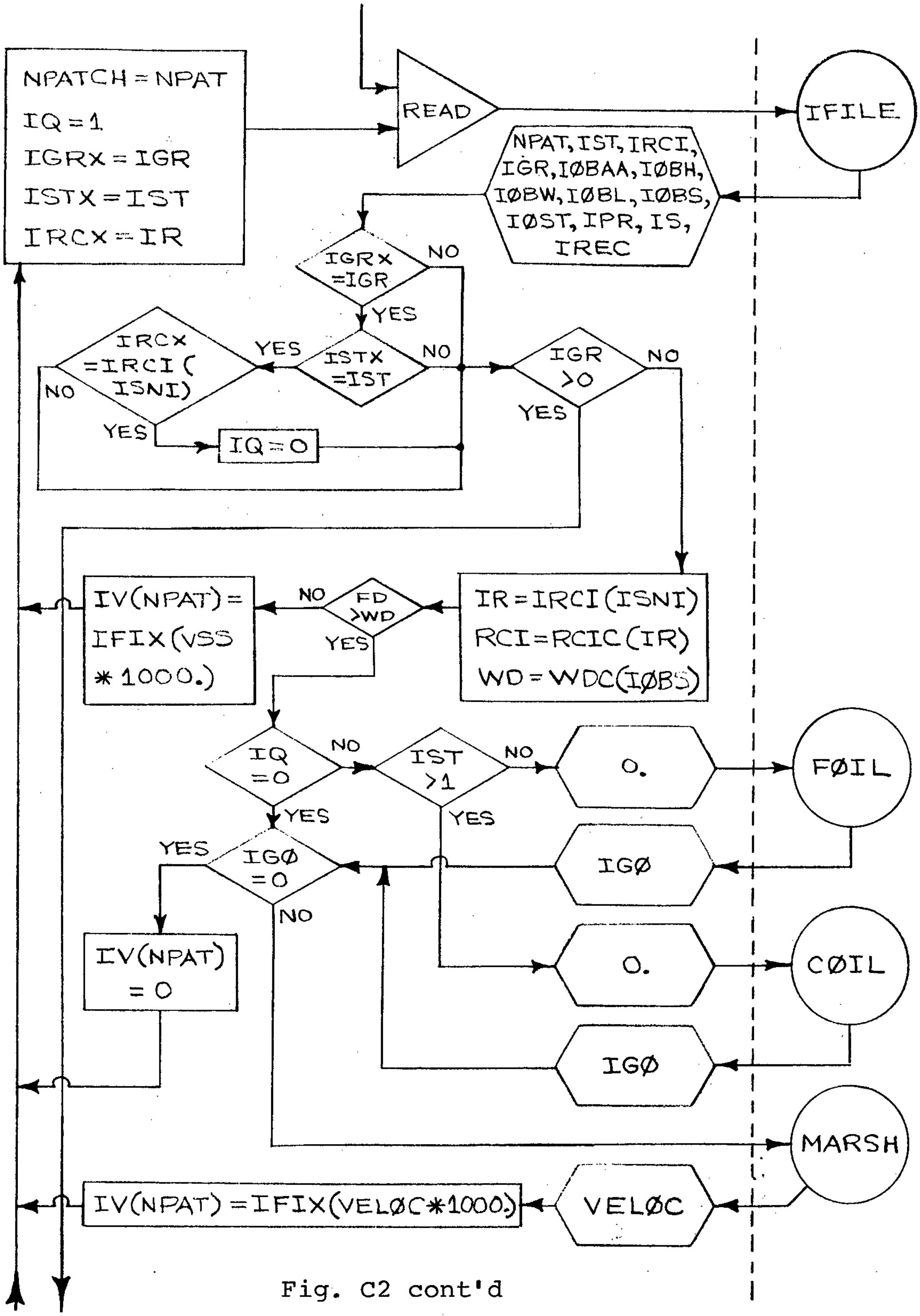
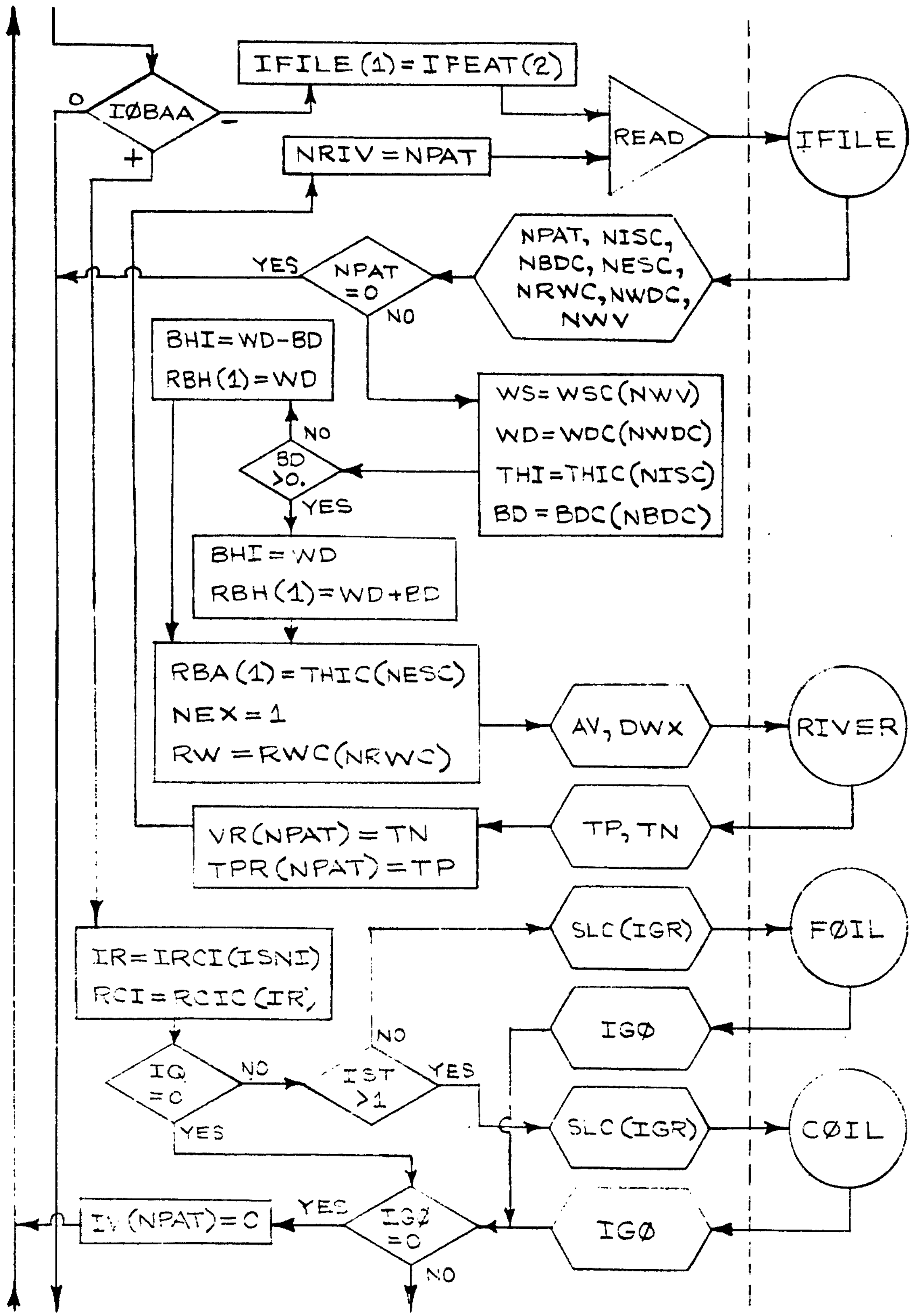

NPATCH = NPAT
IQ = 1
IGRX = IGR
ISTX = IST
IRCX = IR
READ
IFILE
NPAT, IST, IRCI, IGR, IØBAA, IØBH, IØBW, IØBL, IØBS, IØST, IPR, IS, IREC
IGRX =IGR
NO
YES
ISTX =IST
IRCX =IRCI(ISNI)
IQ = 0
IGR >0
IR = IRCI (ISNI)
RCI = RCIC (IR)
WD = WDC(IØBS)
FD >WD
IV(NPAT) = IFIX(VSS * 1000.)
IQ =0
IST >1
O.
FØIL
IGØ =0
IGØ
IV(NPAT) = 0
CØIL
MARSH
VELØC
IV(NPAT) = IFIX(VELØC*1000.)

Fig. C2 cont'd



IFILE(1)=IFEAT(2)
IØBAA
0
-
+
NRIV=NPAT
READ
IFILE
NPAT, NISC, NBDC, NESC, NRWC, NWDC, NWV
YES
NPAT =0
NO
BHI=WD-BD
RBH(1)=WD
WS=WSC(NWV)
WD=WDC(NWDC)
THI=THIC(NISC)
BD=BDC(NBDC)
NO
BD >0.
YES
BHI=WD
RBH(1)=WD+BD
RBA(1)=THIC(NESC)
NEX=1
RW=RWC(NRWC)
AV, DWX
RIVER
TP, TN
VR(NPAT)=TN
TPR(NPAT)=TP
SLC(IGR)
FØIL
IR=IRCI(ISNI)
RCI=RCIC(IR)
IGØ
NO
IQ =0
NO
IST >1
YES
SLC(IGR)
CØIL
YES
IV(NPAT)=0
YES
IGØ =0
IGØ
NO

Fig. C2 cont'd

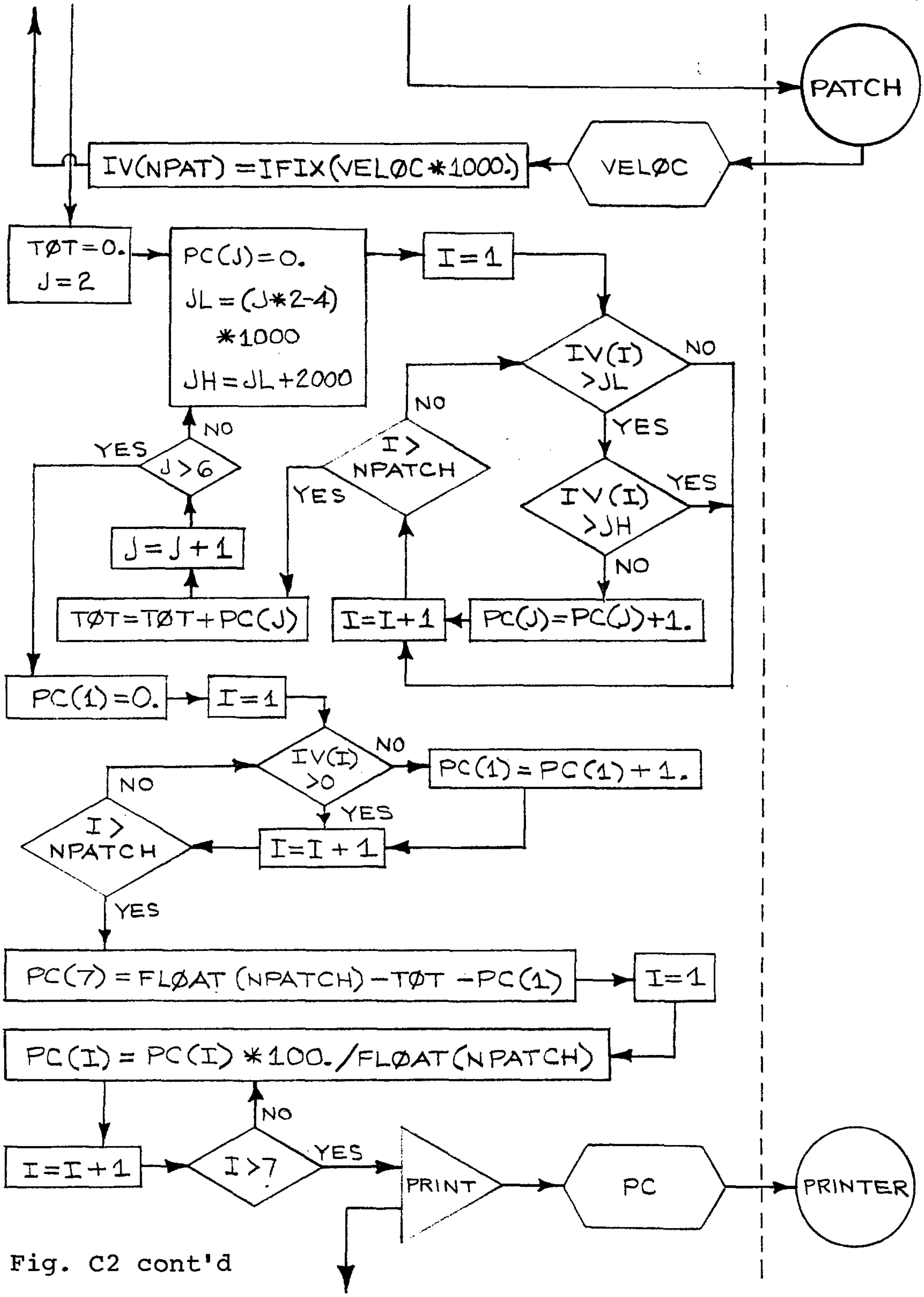
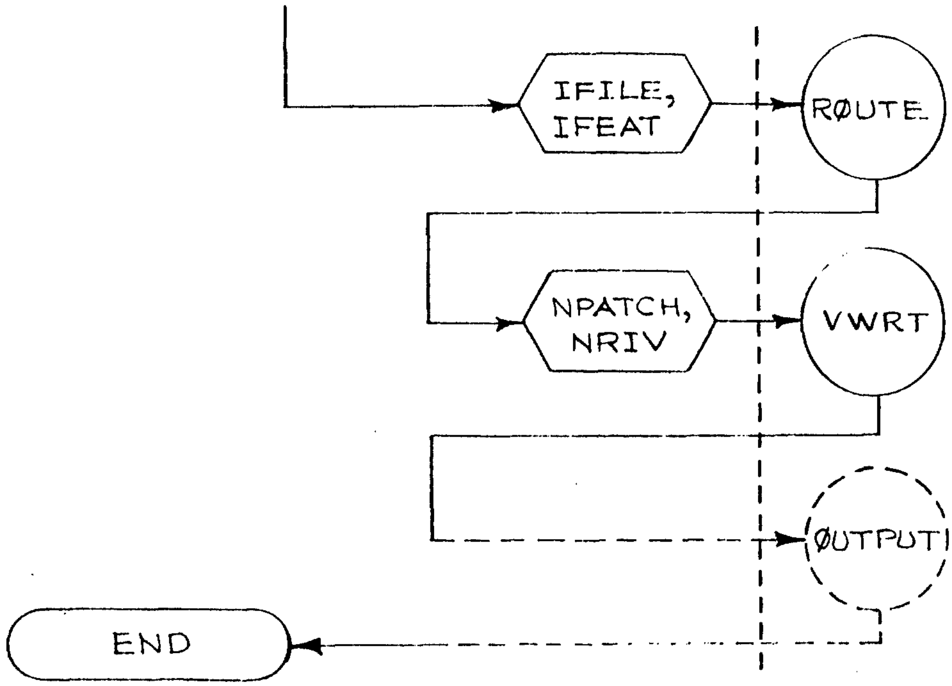

PATCH
VELØC
IV(NPAT) =IFIX(VELØC*1000.)
TØT=0.
J=2
PC(J)=0.
JL=(J*2-4)
*1000
JH=JL+2000
I=1
IV(I)
>JL
NO
YES
IV(I)
>JH
YES
NO
I>
NPATCH
NO
YES
J>6
YES
NO
J=J+1
TØT=TØT+PC(J)
I=I+1
PC(J)=PC(J)+1.
PC(1)=0.
I=1
IV(I)
>0
NO
YES
PC(1)=PC(1)+1.
I>
NPATCH
NO
YES
I=I+1
PC(7)=FLØAT (NPATCH)-TØT -PC(1)
I=1
PC(I)=PC(I)*100./FLØAT(NPATCH)
I=I+1
I>7
NO
YES
PRINT
PC
PRINTER

Fig. C2 cont'd



VARIABLES LEAVING THROUGH COMMON: SV(10), RD(10), SD(10), NSDC, NSSC, SDL(10), ØS(10), AA(20), ØW(10), ØH(10), ØL(10), SLC(10), ISRC(20), RCI, WS, WD, THI, BHI, RW, TANP, C, NEX, RBH(5), RBA(5), IV(1080), VR(110), TPR(1.10)

Fig. C2 cont'd

FR EP

```
1C   AMC 71 MØBILITY MØDEL -- MAIN PRØGRAM
2C
3C   FØR INFØRMATIØN CØNTACT:
4C
5C          JØHN EILERS
6C          U S ARMY TANK-AUTØMØTIVE CØMMAND
7C          AMSTA-RURV
8C          WARRDN, MICHIGAN 48090        PHØNE: 1-313-573-1445
9C
10$ØVR,INPUT
11$ØVR,AUTØF
12$ØVR,STICK
13$ØVR,FØIL
14$ØVR,CØIL
15$ØVR,PATCH
16$ØV$,MARSH
17$ØVR,RIVER
18$ØVR,RØUTE
19$ØVR,VWRT
110        CØMMØN SV(10),RD(10),SD(10),NSDC,SDL(10),NSSC,ØS(10),
120&  AA(20),ØW(10),ØH(10),ØL(10),SLC(10),ISRC(20),IS(10),IREC,
130&  IØBL,IØBW,IØBS,IØBH,IØBAA,IGR,IPR,IRCI(3),IST,IØST,SDS(10),
140&  XNT(10),S(10),FØRCE(2,101),FØRCR(4,101),FØRMX(3),TFØR(3),
150&  RT,RCI,NVEH,NFL,GVW,DL,WID,GT,A,NBC,GC,HPT,ITVAR,RDIAM,
160&  TPSI,TPLY,HS,WC,SAI,AWPKF,GCA,VSS,NCREW,FD,VFS,TNE1(2,30),
170&  TTM(2,30),TTE(2,30),GR(10),NG,TC,RR,FDR,EFF,FDREF,ITRAN,
180&  IVEL,NC1,NC2,NC3,ENTCG,LØKUP,VØØB(2,30),VRIDE(20),W,PBHT,
190&  PBF,VL,NC4,NC5,H,ØEW,ØBAA,XLT,HB,AV,RDC,VDA,CGF,CGH,DWX,
200&  RW1,ACG,DCG,HC,RWW,WS,WD,THI,BHI,RW,TANP,C,NEX,RBH(5),
210&  RBA(5),IV(1080),VR(110),TPR(110),XBR
220        DIMENSIØN RCIC(15),WSC(10),WDC(10),THIC(10),BDC(10)
230        DIMENSIØN PC(10),RWC(25)
235        DIMENSIØN IFILE(2),IFEAT(5)
240        INTEGER VEHICL(2)
250C
260        DATA IDRY/"DRY"/
261        DATA IAVE/"AVE"/
262        DATA IWET/"WET"/
270        DATA (RD(I),I=1,9) /164.,121.,59.,34.8,24.6,17.4,12.5,7.5,
280&  2.6/
290        DATA NSDC,NSSC,(SDL(I),I=1,8)
300&  /8,8,.98,2.36,3.94,5.51,7.09,8.66,9.84,15./
310        DATA (SV(I),I=1,8) /300.,65.6,51.2,31.5,22.3,15.7,10.8,3.9/
320        DATA (ØH(I),I=1,7) /3.15,7.87,11.81,15.75,20.87,28.35,
330&  33.46/
340        DATA (ØW(I),I=1,5) /11.8,3.48,2.49,1.51,.49/
350        DATA (ØL(I),I=1,7) /.66,2.36,5.25,8.53,15.09,256.,492./
360        DATA (ØS(I),I=1,8) /197.,131.,51.2,31.5,22.3,15.7,10.8,3.9/
370        DATA (AA(I),I=1,14) /179.,181.,177.,183.,173.,187.,164.,196.,
380&  154.,206.,142.,218.,112.,248./
```

This Page omitted
(C-44)

FREP CØNTINUED

```
      DATA (RCIC(I),I=1,11) /300.,250.,190.,130.,80.,50.,36.,29.,
&  20.,14.,5./
      DATA (SLC(I),I=1,8) /1.,3.4,7.5,15.,30.,50.,65.,72./
      DATA (WSC(I),I=1,5) /0.,1.64,4.92,9.02,11.48/
      DATA (WDC(I),I=1,4) /1.64,4.92,11.48,16.4/
      DATA (THIC(I),I=1,10) /2.5,7.5,15.,25.,35.,45.,62.5,
&  75.,82.5,87.5/
      DATA (BDC(I),I=1,9) /0.,-1.64,-4.92,-9.84,-13.12,
&  1.64,4.92,9.84,13.12/
      DATA (RWC(I),I=1,24)/0.,4.9,14.8,24.6,34.4,44.3,54.1,64.,
&  73.8,83.7,93.5,106.6,123.,139.,156.,172.,189.,205.,
&  221.5,237.9,254.3,270.7,287.,303.5/
      DATA (ISRC(I),I=1,9) /1,2,3,4,5,6,7,8,9/
      DATA IFEAT/"PCH","H2Ø","SEC","MSH","RIV"/
C
      NPATCH=0
      NRIV=0
      IGRX=100
      ISTX=100
      IRCX=100
      X1=0.
      DØ 4 I=1,NSDC
      SD(I)=(X1+SDL(I))/2.
      X1=SDL(I)
    4 CØNTINUE
    3 CØNTINUE
      PRINT 201
      READ 100,VEHICL
CALL LINK(3,"OINPUT")
      CALL INPUT(VEHICL)
      PRINT 500
  500 FØRMAT(" ENTER GEØGRAPHIC AREA"/)
      READ 501,IMAP
  501 FØRMAT(A3)
      IFILE(1)=IFEAT(1)
      IFILE(2)=IMAP
      CALL ØPENF(2,IFILE)
      PRINT 202
      READ 100,ISEAS
      IF (ISEAS-IDRY)91,90,91
   90 ISNI=1
      GØ TØ 94
   91 IF (ISEAS-IWET)93,92,93
   92 ISNI=3
      GØ TØ 94
   93 ISNI=2
   94 IF (ITR@N)1,1,2
  317 FØRMAT (4I4,F6.0,F4.2,F9.0)
1 CALL LINK(4,"ØSTICK")
  CALL STICK
```

FREP   CØNTINUED

```
      GØ TØ 22
 2 CALL LINK(5,"OAUTØF")
   CALL AUTØF
      GØ TØ 22
   21 NPATCH=NPAT
      IQ=1
      IGRX=IGR
      ISTX=IST
      IRCX=IR
   22 CONTINUE
      READ (2,101) NPAT,IST,(IRCI(J),J=1,3),IGR,IØBAA,IØBH,
&  IØBW,IØBL,IØBS,IØST,IPR,(IS(I),I=1,NSDC),IREC
      IF(IGRX.EQ.IGR.AND.ISTX.EQ.IST.AND.IRCX.EQ.IRCI(ISNI))IQ=0
  101 FØRMAT(I4,4I2,I1,I2,15I1)
      IF (IGR.GT.0) GØ TØ 99
      IR=IRCI(ISNI)
      RCI=RCIC(IR)
      WD=WDC(IØBS)
      IF (FD.GT.WD) GØ TØ 63
      IV(NPAT)=IFIX(VSS*1000.)
      GØ TØ 21
   63 IF(IQ.EQ.0)GØTØ 66
      IF(IST.GT.1)GØTØ 64
 CALL LINK(3,"OFØIL")
      CALL FØIL(0.0,IGØ)
      GØ TØ 66
   64 CALL LINK(3,"OCØIL")
      CALL CØIL(0.0,IGØ)
   66 IF (IGØ)67,67,68
   67 IV(NPAT )  =  0
      GØTØ 21
   68 IF(IQ.EQ.0)GØTØ 110
 CALL LINK(3,"OMARSH")
  110 CALL MARSH(VELØC)
      IV(NPAT)=IFIX(VELØC*1000.)
      GØ TØ 21
   99 IF (IØBAA)20,10,30
 20 CALL CLØSEF(2)
 CALL LINK(3,"ORIVER")
      IFILE(1)=IFEAT(2)
      CALL ØPENF(2,IFILE)
      GØTØ 50
   58 NRIV=NPAT
   50 READ (2,59) NPAT,NISC,NBDC,NESC,NRWC,NWDC,NWV
      IF (NPAT)10,10,60
   59 FØRMAT(I3,I2,I1,2I2,2I1)
   60 WS=WSC(NWV)
      WD=WDC(NWDC)
      THI=THIC(NISC)
      BD=BDC(NBDC)
```

FREP CØNTINUED

```
1150        IF (BD)51,51,52
1160     51 BHI=WD-BD
1170        RBH(1)=WD
1180        GØ TØ 53
1190     52 BHI=WD
1200        RBH(1)=WD+BD
1210     53 RBA(1)=THIC(NESC)
1220        NEX=1
1230        RW=RWC(NRWC)
1240        CALL RIVER(AV,DWX,TP,TN)
1250        VR(NPAT    =   TN
1260        TPR(NPAT)=TP
1270        GØTØ 58
1280     30 IR=IRCI(ISNI)
1290        RCI=RCIC(IR)
1295        IF(IQ.EQ.0)GØTØ 38
1300        IF(IST.GT.1)GØTØ 37
1310        CALL LINK(3,"OFØIL")
1320        CALL FØIL(SLC(IGR),IGØ)
1330        GØ TØ 38
1340     37 CALL LINK(3,"OCØIL")
1345        CALL CØIL(SLC(IGR),IGØ)
1350     38 IF (IGØ)32,32,33
1360     32 IV(NPAT)=0
1370        GØ TØ 178
1375     33 IF(IQ.EQ.0)GØTØ 111
1380   CALL LINK(3,"OPATCH")
1390    111 CALL PATCH(VELØC)
1400        IV(NPAT)=IFIX(VELØC*1000.)
1410    178 CØNTINUE
1430        GØ TØ 21
1440    201 FØRMAT(" ENTER VEHICLE NAME"/)
1450     10 TØT=0.0
1460        DØ 76 J=2,6
1470        PC(J)=0.0
1480        JL=(J*2-4)*1000
1490        JH=JL+2000
1500        DØ 75 I=1,NPATCH
1510        IF(IV(I).LE.JL)GØTØ 75
1520        IF(IV(I).GT.JH)GØTØ 75
1530        PC(J)=PC(J)+1.
1540     75 CØNTINUE
1550        TØT=TØT+PC(J)
1560     76 CØNTINUE
1570        PC(1)=0.0
1580        DØ 77 I=1,NPATCH
1590        IF(IV(I))78,78,77
1600     78 PC(1)=PC(1)+1.
1610     77 CØNTINUE
1620        PC(7)=FLØAT(NPATCH) - TØT -PC(1)
```

FREP CØNTINUED

```
      DØ 79 I=1,7
      PC(I)=PC(I)/FLØAT(NPATCH) *100.
   79 CØNTINUE
      PRINT 80,(PC(I),I=1,7)
  211 FØRMAT (I5,4X,F7.1)
  100 FØRMAT(2A3)
      CALL CLØSEF(2)
CALL LINK(3,"ORØUTE")
      CALL RØUTE(IFILE,IFEAT)
CALL LINK(3,"OVWRT")
      CALL VWRT(NPATCH,NRIV)
   80 FØRMAT(///" PERFØRMANCE BY NUMBER ØF PATCHES"//
&  7(2X,F5.1," % ")/3X,"0.0",6X,"0 TØ 2",4X,"2 TØ 4",4X,
&  "4 TØ 6",4X,"6 TØ 8",3X,"8 TØ 10",5X,"> 10",
&  /15X,"VELØCITY RANGE--MPH")
  202 FØRMAT(" ENTER SEASØN: DRY, AVE, ØR WET?"/)
      END
```

Subroutine INPUT (Fig. C3)

Subroutine INPUT reads in the vehicle data required by the program from the vehicle data files. See the discussion of data files for details.

SUBROUTINE INPUT

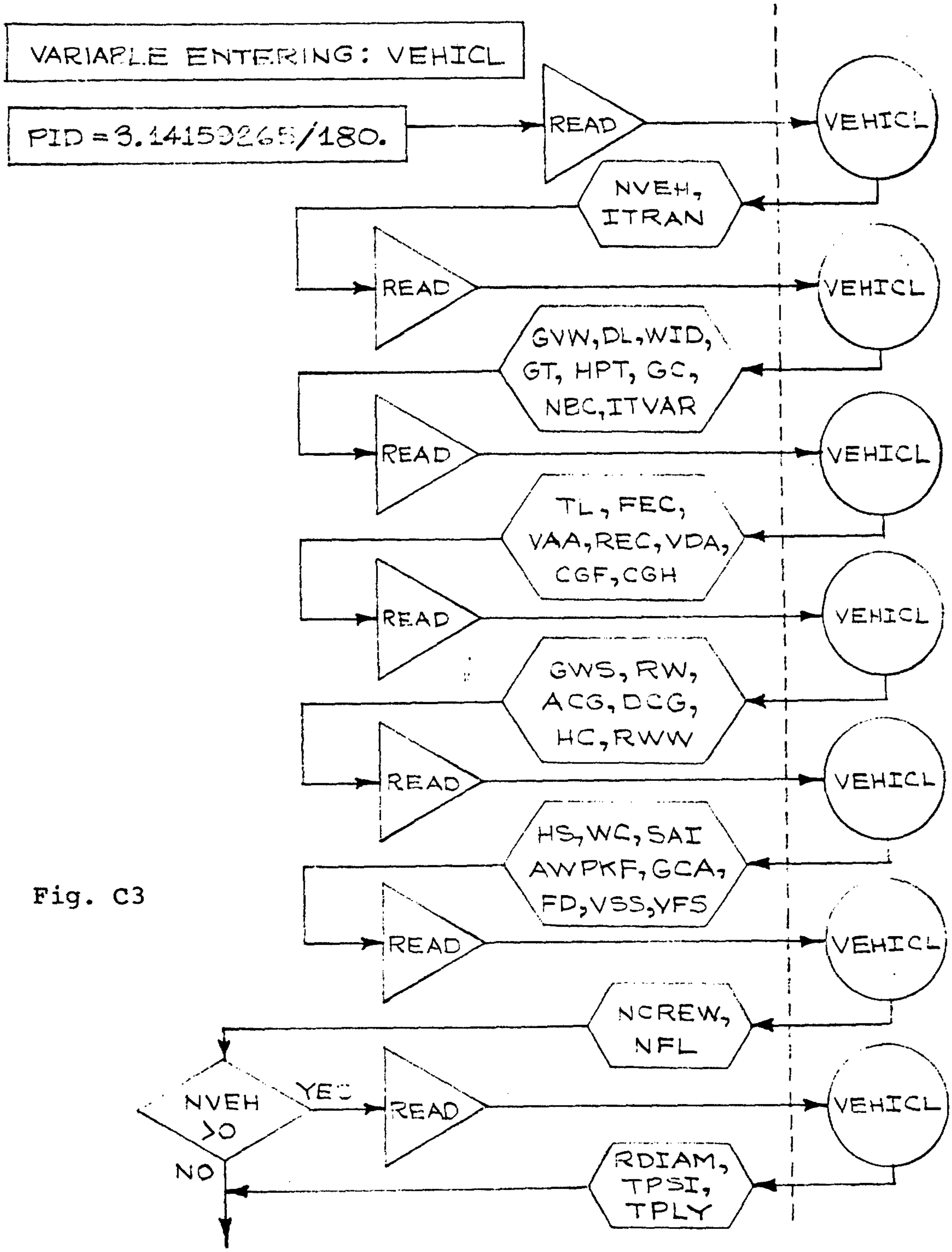


Fig. C3

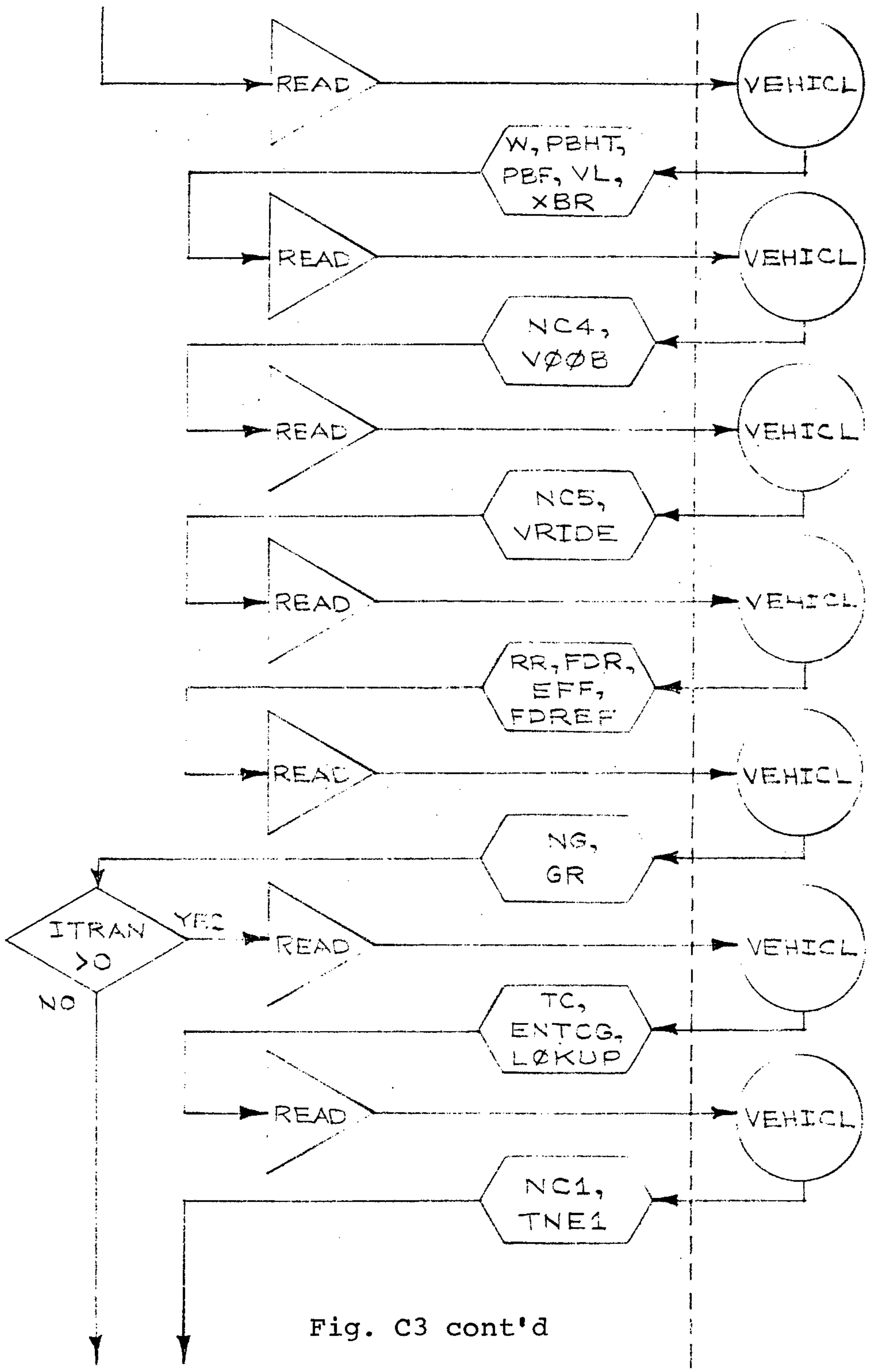
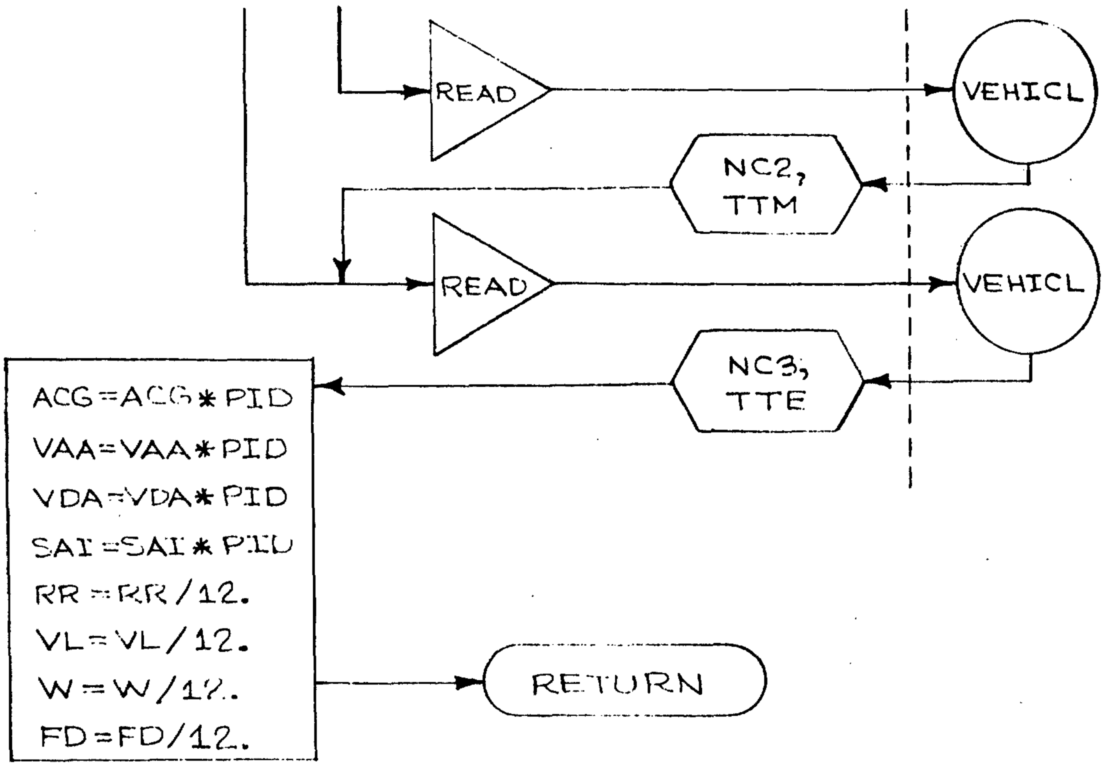

READ
VEHICL
W, PBHT, PBF, VL, XBR
READ
VEHICL
NC4, VØØB
READ
VEHICL
NC5, VRIDE
READ
VEHICL
RR, FDR, EFF, FDREF
READ
VEHICL
NG, GR
ITRAN >0
YES
NO
READ
VEHICL
TC, ENTCG, LØKUP
READ
VEHICL
NC1, TNE1

Fig. C3 cont'd



VARIABLES LEAVING: NVEH, NFL, GVW, DL, WID, GT, A, NBC, GC, HPT, ITVAR, TPSI, TPLY, HS, WC, SAI, AWPKF, GCA, VSS, NCREW, FD, VFS, TNE1(2,30), TTM(2,30), TTE(2,30), GR(10), NG, TC, RR, FDR, EFF, FDREF, ITRAN, NC1, NC2, NC3, ENTCG, LØKUP, VØØB(2,30), VRIDE(20), W, PEHT, PBF, VL, NC4, NC5, TL, FEC, VAA, RFC, VDA, CGF, CGH, GWS, RW, ACG, DCG, HC, RWW, XBR, RDIAM

Fig. C3 cont'd

INPUT

```
      SUBRØUTINE INPUT(VEHICL)
      CØMMØN IPATCH(325),FØRCE(2,101),FØRCR(4,101),FØRMX(3),
     &TFØR(3),RT,RCI,NVEH,NFL,GVW,DL,WID,GT,A,NBC,GC,HPT,ITVAR,
     &RDIAM,TPSI,TPLY,HS,WC,SAI,AWPKF,GCA,VSS,NCRDW,FD,VFS,
     &TNE1(2,30),TTM(2,30),TTE(2,30),GR(10),NG,TC,RR,FDR,EFF,
     &FDREF,ITRAN,IVEL,NC1,NC2,NC3,ENTCG,LØKUP,VØØB(2,30),
     &VRIDE(20),W,PBHT,PBF,VL,NC4,NC5,H,WB,AA,TL,FEC,VAA,REC,
     &VDA,CGF,CGH,GWS,RW,ACG,DCG,HC,RWW,IDUMMY(1555),XBR
      INTEGER VEHICL(2)
      PID=3.14159265/180.
      CALL ØPENF(1,VEHICL)
      READ (1,100) NVEH,ITRAN
      READ (1,101) GVW,DL,WID,GT,A,HPT,GC,NBC,ITVAR
      READ (1,102) TL,FEC,VAA,REC,VDA,CGF,CGH
      READ (1,102) GWS,RW,ACG,DCG,HC,RWW
      READ (1,102) HS,WC,SAI,AWPKF,GCA,FD,VSS,VFS
      READ (1,100) NCREW,NFL
      IF(NVEH)4,4,3
    3 READ (1,102) RDIAM,TPSI,TPLY
    4 READ (1,103) W,PBHT,PBF,VL,XBR
      READ (1,104) NC4,(VØØB(1,I),VØØB(2,I),I=1,NC4)
      READ (1,105) NC5,(VRIDE(I),I=1,NC5)
      READ (1,102) RR,FDR,EFF,FDREF
      READ (1,106) NG,(GR(I),I=1,NG)
      IF(ITRAN)5,5,6
    6 READ (1,107) TC,ENTCG,LØKUP
      READ (1,104) NC1,(TNE1(1,I),TNE1(2,I),I=1,NC1)
      READ (1,104) NC2,(TTM(1,I),TTM(2,I),I=1,NC2)
    5 READ (1,104) NC3,(TTE(1,I),TTE(2,I),I=1,NC3)
      CALL CLØSEF (1)
      ACG=ACG*PID
      VAA=VAA*PID
      RR=RR/12.
      VL=VL/12.
      W=W/12.
      FD=FD/12.
      VDA=VDA*PID
      SAI=SAI*PID
  100 FØRMAT(2I3)
  101 FØRMAT(F7.0,6F7.2,2I3)
  102 FØRMAT(8F7.2)
  103 FØRMAT(5F9.1)
  104 FØRMAT(I3/(2F00.3))
  105 FØRMAT(I3/9F7.2)
  106 FØRMAT(I3/(5F7.4))
  107 FØRMAT(2F7.3,I3)
      RETURN
      END
```

Power Train Submodel

Basic limitations to vehicle performance are set by the maximum tractive force that can be transferred by the driving wheels to the ground. In direct drive, which accounts for the majority of current applications, the engine is coupled through a transmission directly to the driving axle. The transmission can be mechanical, as with a gear shift, or hydraulic. A hydraulic transmission can also include a torque converter.

The gasoline engine starts to run smoothly at a certain minimum idle speed, NEMIN, and produces excess power at speeds above this point. Optimum combustion quality, and therefore maximum effective pressure, is reached at a medium engine speed where, as a result, maximum torque is developed. As speed increases further, brake mean effective pressure deteriorates because of the rapidly growing losses in the air induction manifolds. Torque, therefore, starts to decline. Power output is in the nearly straight-line proportion with speed up to the point of maximum torque. Beyond this point, the rate of power increase falls off until the maximum power output is reached. Engine speed increase beyond this point results in a fast decline in power output, and the maximum permissible speed, NEMAX, is reached quickly. In vehicle applications, this point is usually set just above the maximum power output speed. Vehicles designed for traction, however, are designed to operate at much lower engine speeds, since maximum torque, and not power output, determines performance limits.

The function of the transmission is to transform the torque-speed relation of engine output into a form that corresponds more closely to actual driving demands. The transformation is performed by the following means:

1. By the transmission alone, in the case of a manual gear-shift transmission, or by the transmission plus a torque converter. There can also be a gear reduction stage between the engine and the torque converter.

2. On vehicles requiring extremely high torque at low speeds, additional gear reduction stages are usually placed at the driving wheels.

The power transmission between the engine output shaft and the driving wheels involves the following additional factors as power consumption elements:

1. ENTCE: Engine-to-torque converter efficiency. This power consumption originates in the friction between the gears, and a constant value of 97 percent is used.

2. EFF: This is the transmission power consumption originating in the friction between the gears and oil churning losses. A value of EFF corresponding to the particular vehicle under investigation is used.

3. FDREF: Final drive gear efficiency.

Differences in the torque characteristics of standard and torque converter transmissions require modification of techniques for calculating tractive forces. The gear-shift transmission provides positive ratio coupling between engine speed and vehicle speed, except when the vehicle starts from a standstill. During this part of the operational range, the clutch slips and the exact speed ratio is unknown. Transmission and final drive gear reductions multiply engine torque. Subroutine STICK is used for the case of a vehicle with a standard transmission.

For an automatic transmission, the torque ratio of the converter reaches maximum at stall output speed, and the ratio gradually falls off as output speed increases. The converter eventually acts as a hydraulic coupling with a 1-to-1 torque ratio. Speed ratio of a torque converter is zero at stall conditions - when the vehicle is stationary and the engine is working at a certain predetermined design speed. As the vehicle begins to move, the engine speeds up, first very slowly, then at an increasing rate until the converter becomes a coupling. Characteristics of a typical converter appear in Figure C4. When combined with the reduction in the geared stages, the plot leads to a complete graphical equivalence between vehicle speed, engine speed, converter torque ratio, and engine torque output. Subroutine AUTØF is used for the case of a vehicle with an automatic transmission and torque converter.

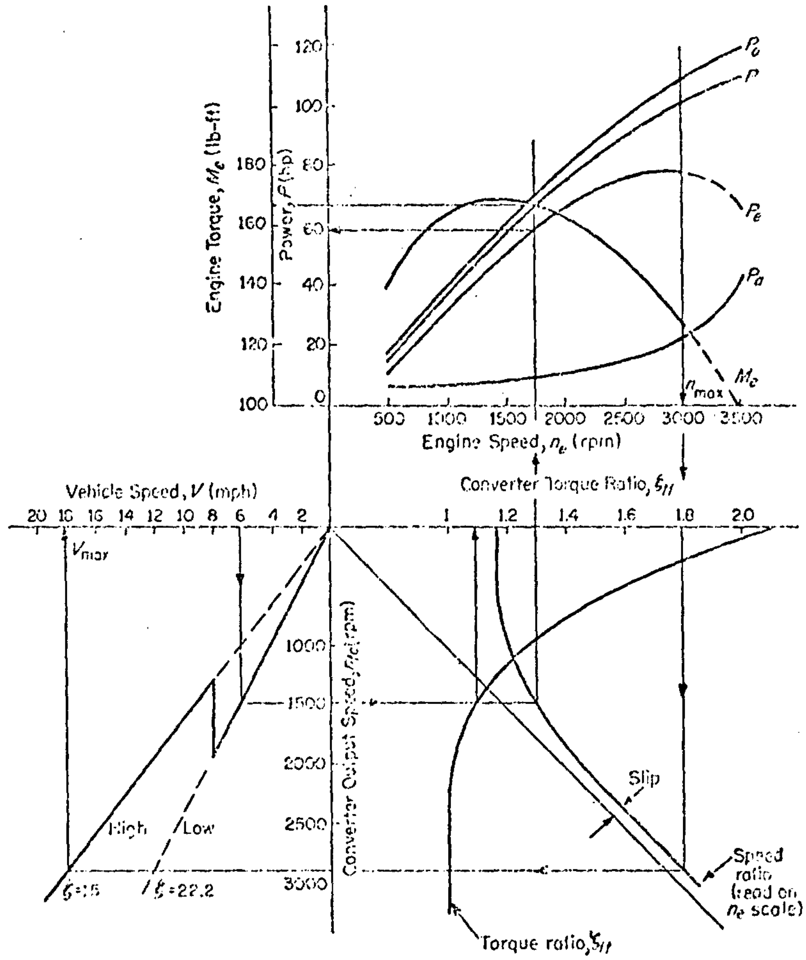


Fig. C4 - Powerplant and torque - converter characteristics for truck installation. At converter stall speed (= 0), torque ratio peaks at 2.1:1 and engine is 1400 rpm. Plot at lower left shows speed reduction in mechanical stages of the transmission. Transition between low and high gear ratios is performed at vehicle speed of 8 mph.

Subroutine STICK (Fig. C5)

Subroutine STICK calculates the curve of theoretical vehicle velocity versus tractive force for a vehicle with a manual transmission. Note that this tractive force is defined by the torque transferred from the engine through the transmission to the final drive axle. Limitations due to the soil's ability to support horizontal tractive forces will be considered later. The variables entering the subroutine are as follows:

1. Array TTE: This array carries points on the curve of engine speed versus engine output torque.

2. NC3: Indicates the number of points on the above curve.

3. NG: The number of gear ratios in the transmission.

4. GR(I) (I = 1, 2. . . NG): The values of the gear ratios.

5. EFF: Transmission efficiency.

6. FDR: Final drive ratio.

7. FDREF: Final drive efficiency.

8. RR: Rolling radius of the wheel for a wheeled vehicle, or the radius of the road wheel plus the track thickness for a tracked vehicle.

Variables NEMAX and NEMIN represent the values of engine speed at the highest and lowest points on the incoming array TTE.

The subject subroutine produces array FØRCE (I, J) as will be explained below, which will carry the tractive force versus theoretical vehicle velocity curve. I = 1 indicates tractive force; I = 2 indicates vehicle velocity. J is incremented from 1 to 101. This represents vehicle velocities from 0 to 50 mph in 0.5 mph increments. These values of velocity are now loaded into the second column of the array FØRCE. The values in the first column of array FØRCE (the column representing tractive effort) are initiated at zero. This column of the array will carry the values of tractive force corresponding to each velocity. These tractive force values are now calculated in the rest of the subroutine.

The balance of the subroutine consists of one long loop, the index of which is variable NGEAR. This is indexed from 1 to NG, NG being the number of gear ratios in the transmission. The tractive force value is calculated for each gear ratio. The best value is selected and overrides any previous value calculated in an earlier pass through the loop. On the first pass, all calculated values are accepted since the previous values had been initiated at zero. On subsequent passes, new calculated values may or may not be larger than previous values.

An inner loop is now begun, which increments velocity from 0 to 50 mph in 0.5 mph increments. This velocity is carried in the variable VEL. For each value of VEL, a value of NT is calculated. NT represents engine speed corresponding to vehicle velocity VEL, corrected for the gear ratios of the transmission and the final drive, and corrected for the rolling radius of the wheel or road wheel. NT is then checked to see if it lies outside of the range of available engine speeds (i.e., if it is greater than NEMAX or less than NEMIN). If it is greater than NEMAX, a return is made to the top of the inner loop, and VEL is incremented upward by 0.5 mph. If it lies within the range of available engine speed, subroutine CURVE is called. Returning from CURVE is variable TE. This is the value of engine output torque corresponding to engine speed NT. It is derived by interpolation from the array TTE.

The next variables calculated are: TØS, the transmission output speed (the engine speed NT, divided by the transmission gear ratio); and TØT, the transmission output torque (the engine output torque TE, times the gear ratio, times the transmission efficiency). Next, the corresponding tractive force at the ground is calculated. This is variable

TRF; it is equal to the transmission output torque times the final drive ratio, times the final drive efficiency, divided by RR, the rolling radius.

Next, J1 is calculated. This indicates the current location within array FØRCE. It corresponds to the specific value of velocity VEL. TRF is now compared with the previously calculated value of FØRCE (1, J1). If it is greater, it overrides this previous value and is loaded into FØRCE (1, J1). If it is smaller, the value of FØRCE (1, J1) calculated in a previous pass through the outside loop is retained. When the calculations for each gear ratio are completed, the array FØRCE contains the best tractive effort for each increment of vehicle velocity.

Finally, the variable IVEL is set equal to the value of J1, which is the last value used in the outside loop. It represents the maximum vehicle velocity that can be achieved and is dependent upon the maximum engine speed available in the highest gear ratio. It will be used in calculations in subsequent subroutines as the upper limit of array FØRCE. At this point, a return is made to the calling program.

SUBROUTINE STICK

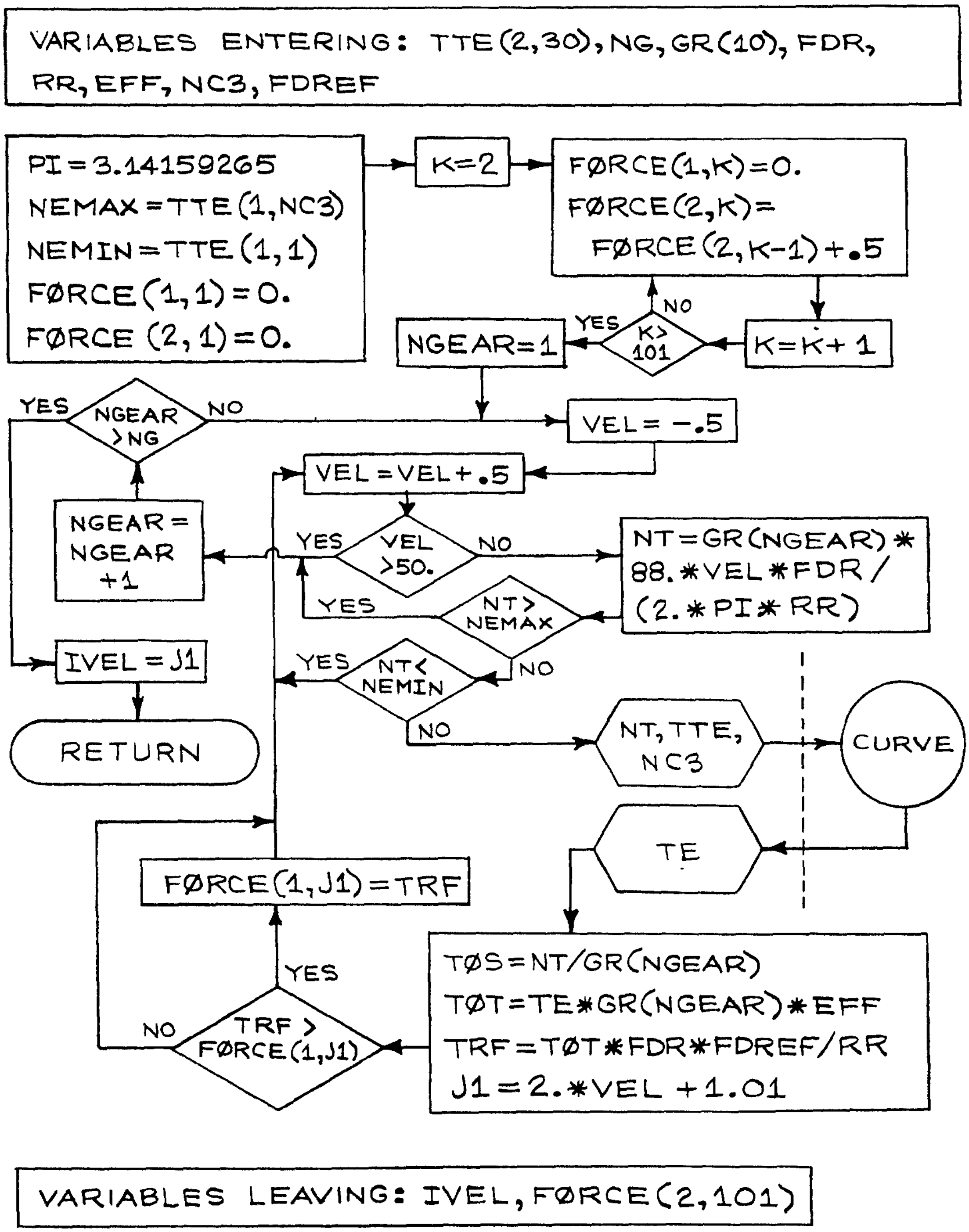


Fig. C5

STICK

```
      SUBRØUTINE STICK
      DIMENSIØN TNE1(2,30),TTM(2,30),TTE(2,30),GR(10),FØRCE(2,101)
      REAL NE,NT,NEMIN,NEMAX
      CØMMØN IPATCH(325),FØRCE,ISØIL(865),TNE1,TTM,TTE,GR,
&    NG,TC,RR,FDR,EFF,FDREF,ITRAN,IVEL,NC1,NC2,NC3,ENTCG,LØKUP
      FØRCE(1,1)=0.
      FØRCE(2,1)=0.
      DØ 71 K=2,101
      FØRCE(1,K)=0.
      FØRCE(2,K)=FØRCE(2,K-1)+0.5
   71 CØNTINUE
      NEMAX=TTE(1,NC3)
      NEMIN=TTE(1,1)
      PI=3.1415926
      DØ 140 NGEAR=1,NG
      VEL=-0.5
  161 VEL=VEL+0.5
      IF (VEL-50.)162,162,140
  162 NT=GR(NGEAR)*88.*VEL*FDR/(2.*PI*RR)
      IF (NT-NEMAX)120,120,140
  120 IF (NT-NEMIN)161,121,121
  121 CALL CURVE(NT,TE,TTE,30)
      TØS=NT/GR(NGEAR)
      TØT=TE*GR(NGEAR)*EFF
      TRF=TE*GR(NGEAR)*EFF*FDR*FDREF/RR
      J1=2.*VEL+1.01
      IF (TRF-FØRCE(1,J1))161,161,160
  160 FØRCE(1,J1)=TRF
      GØ TØ 161
  140 CØNTINUE
      IVEL=J1
      RETURN
      END
```

Subroutine AUTØF (Fig. C6)

Subroutine AUTØF calculates the curve of tractive force versus theoretical vehicle velocity for a vehicle with an automatic transmission and torque converter. The variables entering the subroutine are as follows:

1. Array TTE: This array contains the curve of engine speed versus engine output torque.

2. Array TNE1: This array contains the curve of torque converter-speed ratio versus converter input speed.

3. Array TTM: This array contains the curve of torque converter-speed ratio versus torque multiplying coefficient.

4. NG: The number of gears in the transmission.

5. GR(I): The values of the gear ratios.

6. ENTCG: The gear ratio between the engine and torque converter. This pair of gears is sometimes present to match the optimum operating point of the engine with the optimum operating point of the torque converter.

7. TC: The input torque at which the torque converter curves are measured.

8. EFF: Transmission efficiency.

9. FDR: Final drive ratio.

10. FDREF: Final drive efficiency.

11. RR: Rolling radius of the wheel or road wheel.

12. LØKUP: A variable denoting whether a torque converter lockup is present.

The first thing checked in the program is variable ENTCG. If this variable has a value other than 1, there is a gear ratio between the engine and torque converter, ENTCG.

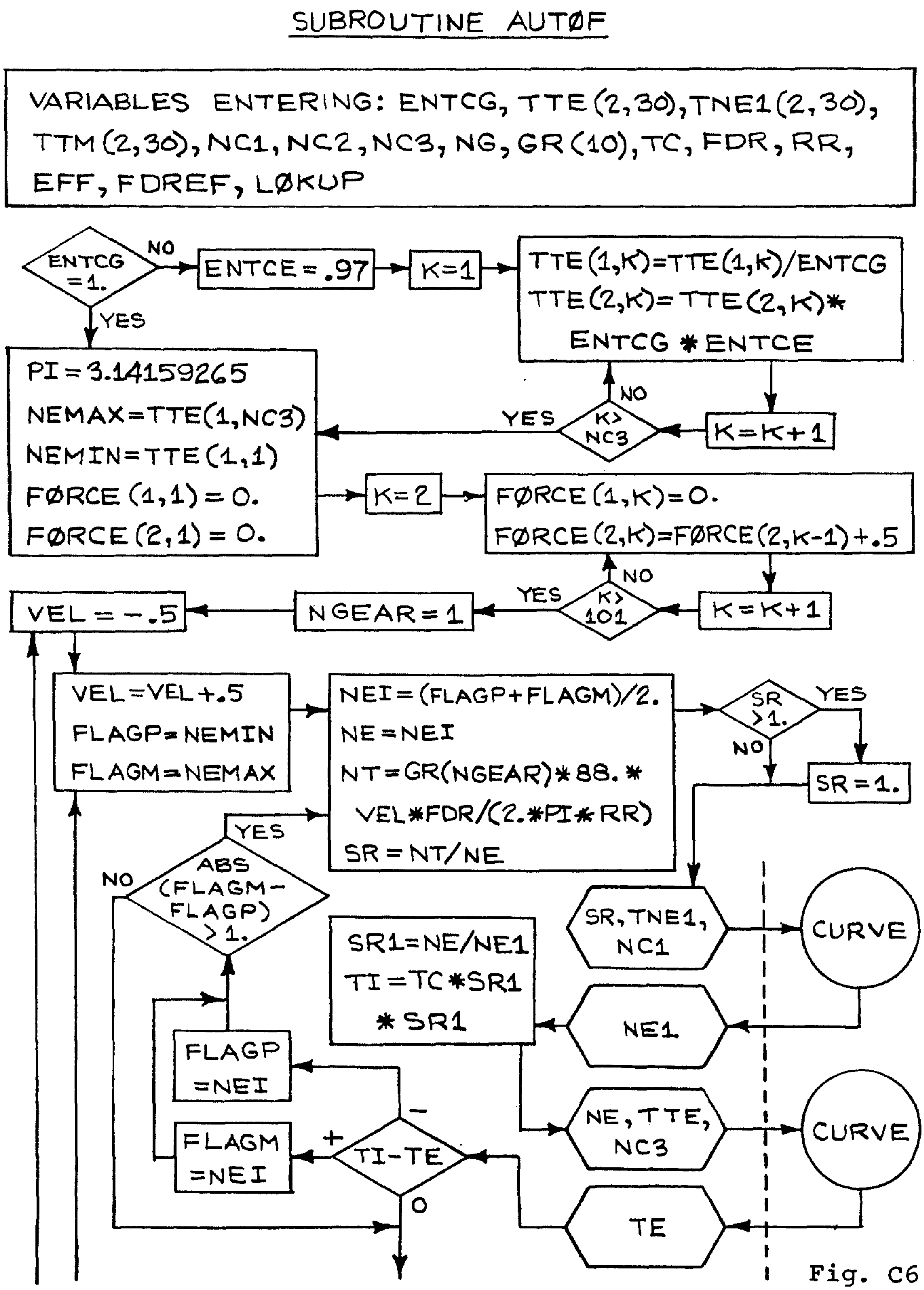
The efficiency of this gear is arbitrarily set to 0.97. The engine speed versus engine output torque curve, TTE, is modified to reflect this gear ratio. The first column carrying engine output torque is multiplied by this ratio and by the efficiency. Array TTE now represents the speed versus torque relation at the output shaft of this gear set. Next NEMAX and NEMIN, the highest and lowest speeds of the input shaft of the torque converter, are set equal to the highest and lowest points of the first column of array TEE. Array FØRCE is now initiated. The first column of this array will carry the values of tractive force calculated by the subroutine; these values are initiated at zero. The second column will carry the corresponding vehicle velocities, incremented from 0 to 50 mph in 0.5 mph increments. These velocities are now loaded into the second column of the array.

A set of three nested loops is now entered. Within these loops the tractive force values corresponding to each velocity increment will be calculated, if the torque converter is assumed to be in operation. The outer loop has index NGEAR, and will run through the transmission gear ratios. The second loop increments velocity from 0 to 50 mph in 0.5 increments; this value is stored in variable VEL. Next, initial values of two temporary variables are established: FLAGM and FLAGP, set equal to the minimum and maximum shaft speeds, respectively (there were NEMIN and NEMAX). Now the third loop is begun. This loop will zero in on the appropriate engine speed corresponding to vehicle velocity VEL, by narrowing the range between FLAGP and FLAGM until the difference is less than one unit. First, variable NEI is set equal to the midpoint of this range, and variable NE is set equal to NEI. NE now represents the temporary value of engine speed. Next, NT is calculated; this is the value of engine speed corresponding to vehicle velocity VEL as reflected through the torque converter and transmission gear. The torque converter-speed ratio, SR, is set equal to NT divided by NE. If the value of SR is greater than 1, it is set back to 1, since the speed ratio of the torque converter cannot exceed 1. Next, SR is sent into subroutine CURVE; this subroutine will interpolate array TNE1, which contains the speed ratio versus input speed curve for the torque converter. Returning from the subroutine is variable NE1, the input speed corresponding to speed ratio SR. Next, a corrected ratio SR1 is set equal to NE divided by NE1, and a temporary value of torque converter input torque TI is set equal to TC times $SR1^2$. NE is then sent to subroutine CURVE. This time the subroutine will interpolate array TTE, the array containing the engine speed versus engine output torque curve. Returning from the subroutine is variable TE, the engine output torque corresponding to engine speed NE. Torque TE is compared against TI; if the difference is negative, FLAGP is reset to NE1; if the difference is positive, FLAGM is set equal to NE1. The two values FLAGP and FLAGM represent the maximum and minimum engine speeds within which the appropriate value lies. A return is now made to the top of the third inner loop, and new values are calculated for torque converter input torque and engine output torque. A comparison of these two will determine how the range between FLAGP and FLAGM is narrowed. When this difference is less than one unit, the values are taken to be established, and an exit is made from this loop.

The speed ratio of the torque converter, SR, is then sent into subroutine CURVE. This time the subroutine will interpolate array TTM, the array containing the torque converter-speed ratio versus torque multiplying coefficient curve. Returning from the subroutine is variable TM, the torque multiplying coefficient corresponding to speed ratio SR. Next, variable TØS, transmission output speed, is calculated. TØS equals the input speed NT divided by the gear ratio. The transmission output torque TØT is set equal to TI, the torque converter input torque times the converter multiplying coefficient TM, times the gear ratio of the transmission, times the transmission efficiency. Next, the torque at the ground, TRF, is set equal to the transmission output torque TØT, times the final drive ratio FDR, times the final drive efficiency FDREF, divided by the wheel rolling radius RR. Variable J1 is then calculated. This denotes the current location within array FØRCE corresponding to the particular velocity presently being calculated. TRF is checked against the value of FØRCE (I, J1). If it is greater, it is loaded into this location in the array. If it is not greater, it is discarded, and the previously calculated value is retained. A return is now made to the top of the second loop, and a new value of VEL is calculated.

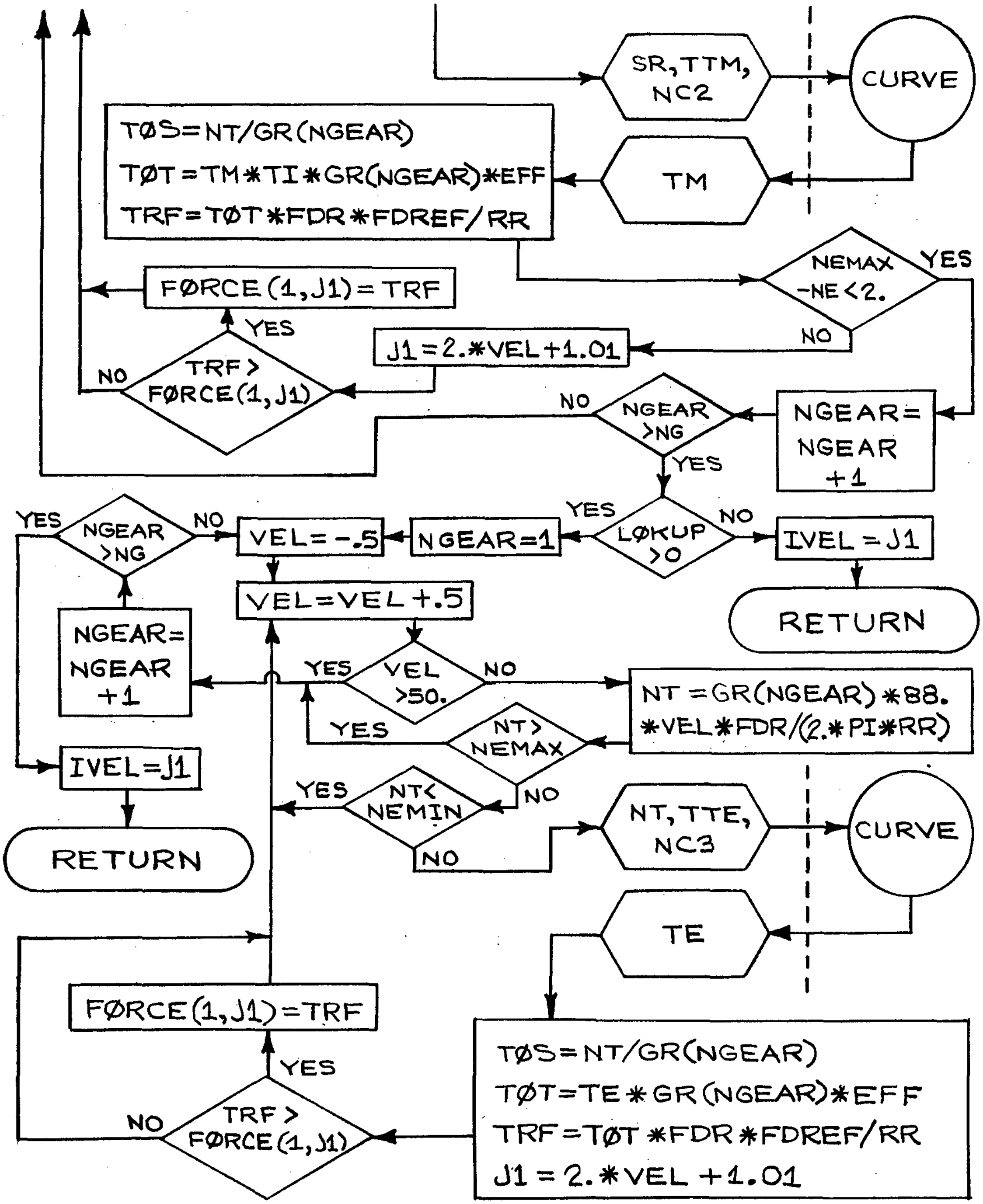
When NEMAX minus NE, the current value of engine speed, is less than 2, a return is made to the top of the outer loop, and calculations are performed for another gear ratio. When all of these calculations are completed, the array FØRCE is filled for operations with a torque converter.

A check is made next to see if a torque converter lockup is present. If there is, the system is identical to a manual transmission, and a loop is entered that is identical to the loop in subroutine STICK. Here, tractive force values are calculated as though the torque converter were a simple 1:1 ratio. If any are greater than the previously calculated tractive force values for the particular velocities, they are superimposed in array FØRCE at the appropriate places. The final step in the subroutine is to set variable IVEL euqal to J1, which represents the highest point reached in array FØRCE. In subsequent subroutines, IVEL will represent the maximum value in this array. A return is now made to the calling program.



SUBROUTINE AUTØF
VARIABLES ENTERING: ENTCG, TTE(2,30),TNE1(2,30), TTM(2,30), NC1, NC2, NC3, NG, GR(10), TC, FDR, RR, EFF, FDREF, LØKUP
ENTCG =1.
NO
YES
ENTCE=.97
K=1
TTE(1,K)=TTE(1,K)/ENTCG
TTE(2,K)=TTE(2,K)*
ENTCG * ENTCE
PI=3.14159265
NEMAX=TTE(1,NC3)
NEMIN=TTE(1,1)
FØRCE (1,1)=0.
FØRCE (2,1)=0.
K> NC3
K=K+1
K=2
FØRCE(1,K)=0.
FØRCE(2,K)=FØRCE(2,K-1)+.5
K> 101
VEL=-.5
NGEAR=1
VEL=VEL+.5
FLAGP=NEMIN
FLAGM=NEMAX
NEI=(FLAGP+FLAGM)/2.
NE=NEI
NT=GR(NGEAR)*88.*
VEL*FDR/(2.*PI*RR)
SR=NT/NE
SR >1.
SR=1.
ABS (FLAGM-FLAGP) >1.
SR, TNE1, NC1
CURVE
SR1=NE/NE1
TI=TC*SR1
*SR1
NE1
FLAGP =NEI
FLAGM =NEI
TI-TE
+
−
0
NE, TTE, NC3
CURVE
TE

Fig. C6



VARIABLES LEAVING: IVEL, FØRCE(2,101)

Fig. C6 cont'd

AUTØF

```
      SUBRØUTINE AUTØF
      DIMENSIØN FØRCE(2,101)
      DIMENSIØN TNE1(2,30),TTM(2,30),TTE(2,30),GR(10)
      REAL NE,NT,NEI,NE1,NEMIN,NEMAX
      CØMMØN IPATCH(325),FØRCE,ISØIL(865),TNE1,TTM,TTE,GR,
&    NG,TC,RR,FDR,EFF,FDREF,ITRAN,IVEL,NC1,NC2,NC3,ENTCG,LØKUP
      IF (ENTCG-1.)3,4,3
    3 ENTCE=0.97
      DØ 2 K=1,NC3
      TTE(1,K)=TTE(1,K)/ENTCG
      TTE(2,K)=TTE(2,K)*ENTCG*ENTCE
    2 CØNTINUE
    4 FØRCE(1,1)=0.
      FØRCE(2,1)=0.
      DØ 71 K=2,101
      FØRCE(1,K)=0.
      FØRCE(2,K)=FØRCE(2,K-1)+0.5
   71 CØNTINUE
      NEMAX=TTE(1,NC3)
      NEMIN=TTE(1,1)
      PI=3.1415926
      DØ 50 NGEAR=1,NG
      VEL=-0.5
   61 VEL=VEL+0.5
      FLAGP=NEMIN
      FLAGM=NEMAX
   80 CØNTINUE
      NEI=(FLAGP+FLAGM)/2.
      NE=NEI
      NT=GR(NGEAR)*88.*VEL*FDR/(2.*PI*RR)
      SR=NT/NE
      IF (SR.GE.1.) SR=1.
      CALL CURVE(SR,NE1,TNE1,30)
      SR1=NE/NE1
      TI=TC*SR1*SR1
      CALL CURVE(NE,TD,TTE,30)
      IF (TI-TE)30,21,40
   30 FLAGP=NEI
      GØ TØ 90
   40 FLAGM=NEI
   90 IF (ABS(FLAGM-FLAGP)-1.)21,21,80
   21 CØNTINUE
       CALL CURVE(SR,TM,TTM,30)
      TØS=NT/GR(NGEAR)
      TØT=TM*TI*GR(NGEAR)*EFF
      TRF=TM*TI*GR(NGEAR)*EFF*FDR*FDREF/RR
      IF (NEMAX-NE-2.)57,58,58
   58 J1=2.*VEL+1.01
       IF (TRF-FØRCE(1,J1))61,61,60
   60 FØRCE(1,J1)=TRF
```

AUTØF CØNTINUED

```
      GØ TØ 61
   57 CØNTINUE
   50 CØNTINUE
      IF (LØKUP)5,5,6
    6 DØ 140 NGEAR=1,NG
      VEL=-0.5
  161 VEL=VEL+0.5
      NT=GR(NGEAR)*88.*VEL*FDR/(2.*PI*RR)
      IF (NT-NEMIN)161,170,170
  170 IF (NT-NEMAX)121,121,140
  121 CALL CURVE(NT,TE,TTE,30)
      TØS=NT/GR(NGEAR)
      TØT=TE*GR(NGEAR)*EFF
      TRF=TE*GR(NGEAR)*EFF*FDR*FDREF/RR
      J1=2.*VEL+1.01
      IF (TRF-FØRCE(1,J1))161,161,160
  160 FØRCE(1,J1)=TRF
      GØ TØ 161
  140 CØNTINUE
    5 IVEL=J1
      RETURN
      END
```

Subroutine KURVE (Fig. C7)

Subroutine KURVE is an interpolation routine. In all cases, the entering array to be interpolated is array FØRCR, the soil-dependent tractive force versus velocity array. It has four columns: the first contains a velocity component on level ground, the second the tractive force component on level ground, the third the velocity component on a slope, and the fourth the tractive force component on a slope. Obviously, columns one and two correspond, and columns three and four correspond. One of the variables entering the subroutine is N, which determines the columns that are to be used as dependent and independent variables for the calculation. If N enters as +2, the independent variable will be the velocity on level ground, and the dependent variable will be tractive force on level ground. If N enters as +1, the independent variable will be velocity on slope, and the dependent variable will be tractive force on slope. If N enters as -1, the independent variable will be tractive force on level ground, and the dependent variable will be velocity on level ground. If N enters as -2, the independent variable will be tractive force on a slope, and the dependent variable will be velocity on a slope.

The first operation performed is the location of the first non-zero element in the entering array. This first non-zero element is identified with index IBEG. Next, determination is made as to whether the dependent variable increases or decreases as the independent variable increases. This information is stored in FACT. A search is then made through the array starting at index IBEG, the first non-zero element in the array, and continuing to the last point of the array, identified by index IVEL. The location of the incoming value of the independent variable in the array is established, and the corresponding value of the dependent variable is calculated from the appropriate column in the array. This value is then returned to the calling program.

SUBRØUTINE KURVE

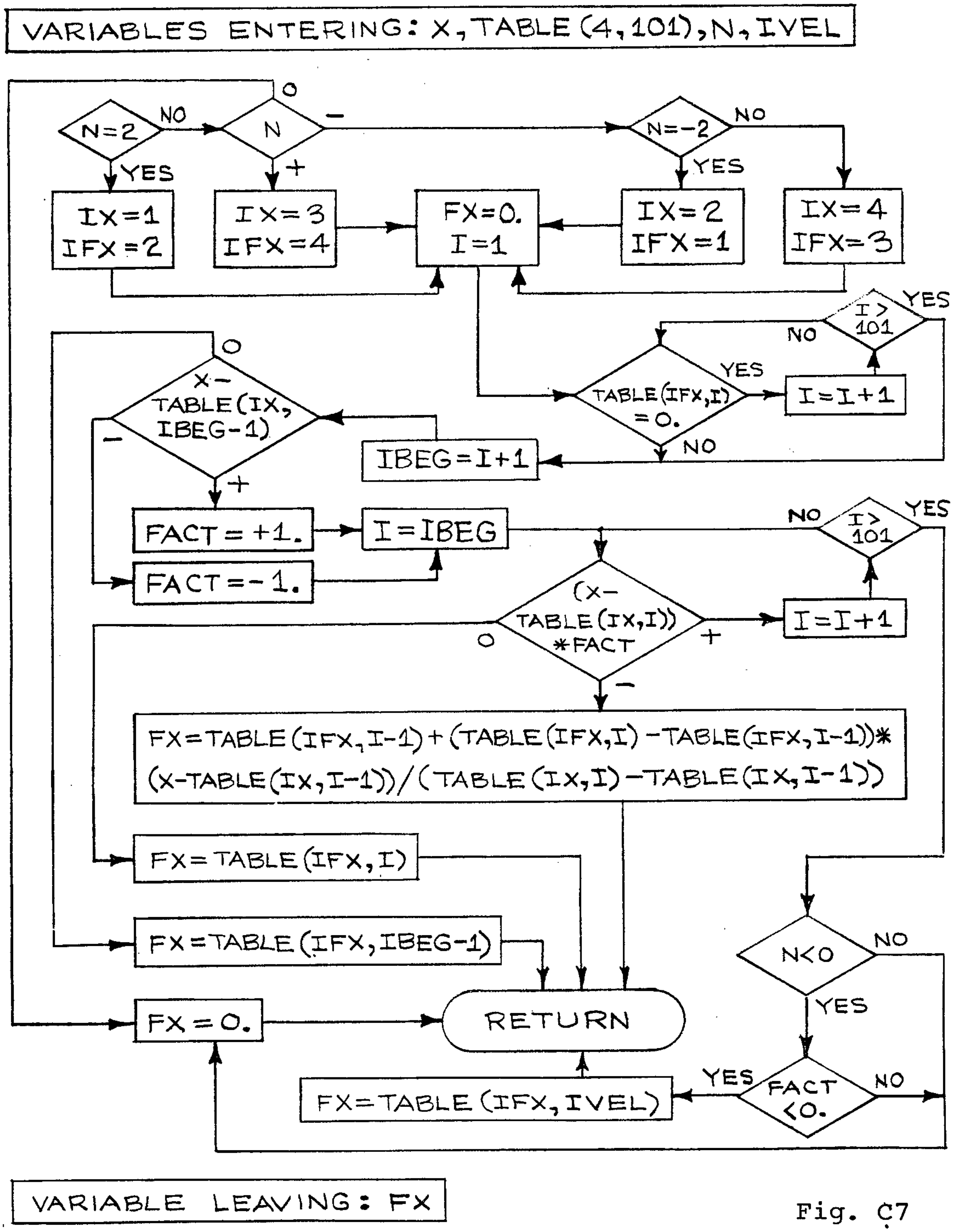


Fig. C7

KURVE

```
      SUBRØUTINE KURVE(X,FX,TABLE,N,IVEL)
      DIMENSIØN TABLE(4,101)
      IF (N-2)11,4,11
   11 IF (N)13,15,8
   13 IF (N+2)7,3,7
    3 IX=2
      IFX=1
      GØ TØ 6
    8 IX=3
      IFX=4
      GØ TØ 6
    4 IX=1
      IFX=2
      GØ TØ 6
    7 IX=4
      IFX=3
    6 FX=0.
      DØ 9 I=1,101
      IF (TABLE(IFX,I))14,9,14
    9 CØNTINUE
   14 IBEG=I+1
      IF (X-TABLE(IX,IBEG-1))30,12,40
   30 FACT=-1.
      GØ TØ 50
   40 FACT=1.
   50 DØ 10 I=IBEG,IVEL
      IF ((X-TABLE(IX,I))*FACT)5,2,10
    5 FX=TABLE(IFX,I-1) + (TABLE(IFX,I)-TABLE(IFX,I-1))
&  *(X-TABLE(IX,I-1))/(TABLE(IX,I)-TABLE(IX,I-1))
      RETURN
    2 FX=TABLE(IFX,I)
      RETURN
   10 CØNTINUE
      IF (N)16,15,15
   16 IF (FACT)17,15,15
   17 FX=TABLE(IFX,IVEL)
      RETURN
   15 CØNTINUE
      FX=0.
      RETURN
   12 FX=TABLE(IFX,IBEG-1)
      RETURN
      END
```

Subroutine CURVE: (Fig. C8)

Subroutine CURVE is also an interpolation routine. In this case, the entering array is array VØØB, or one of the three arrays from the engine model. Array VØØB consists of two columns: the first is obstacle height, and the second is vehicle velocity over an obstacle of this height, limited by 2.5-g vertical acceleration at the driver's seat. The variable N entering the array determines which of these columns is the dependent variable and which is the independent variable. If N enters as +1, the independent variable is obstacle height, and the dependent variable is the associated vehicle velocity. If N enters as -1, the independent variable is vehicle velocity, and the dependent variable is obstacle height. A check is made to determine whether the dependent variable increases or decreases as the independent variable increases. This information is stored in variable FACT. Next, the location of the incoming value of the independent variable is established by searching the array, and the corresponding value of the dependent variable is calculated by interpolation. This value is returned to the calling program.

SUBROUTINE CURVE

VARIABLES ENTERING: X, TABLE (2,101), N

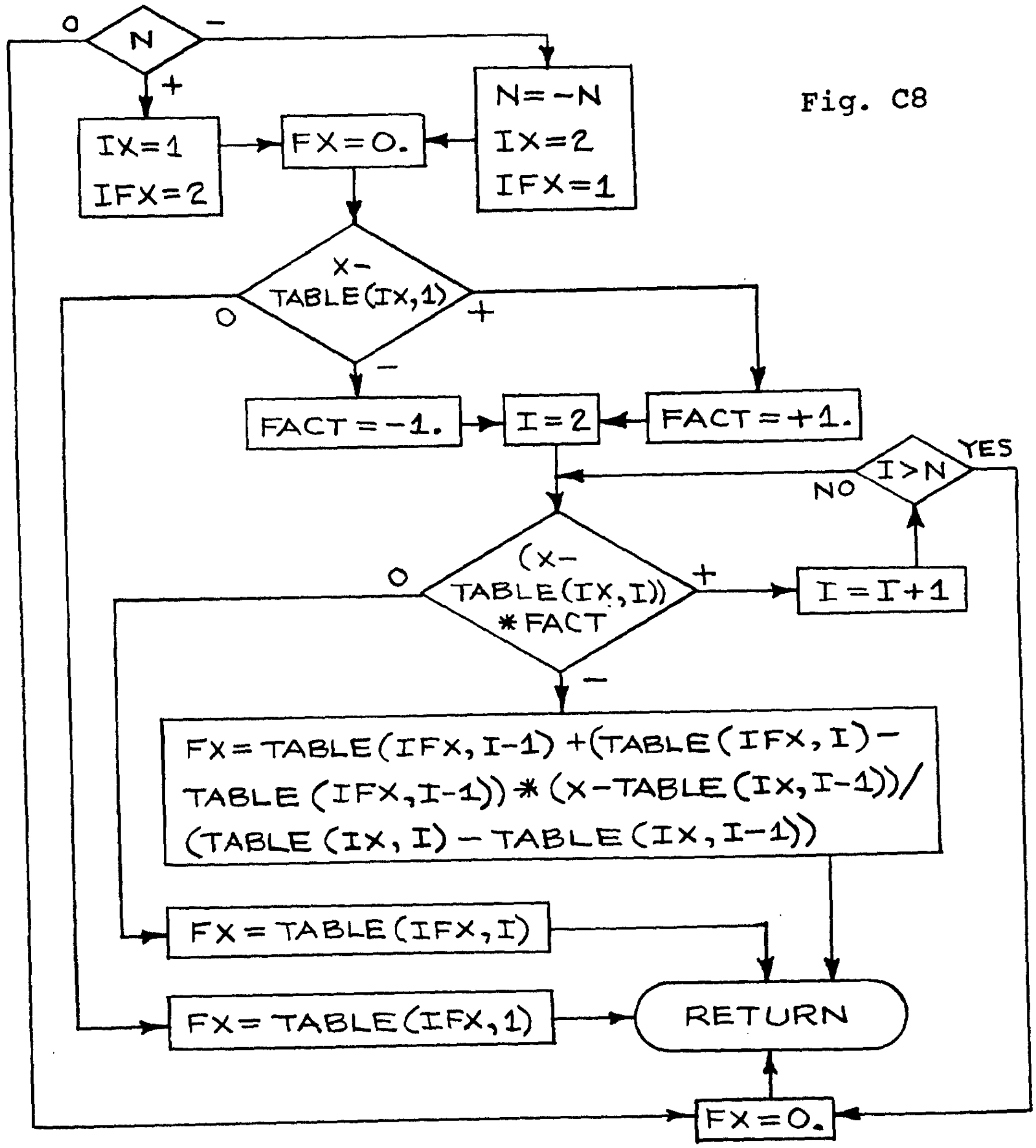


Fig. C8

VARIABLE LEAVING: FX

CURVE

```
      SUBRØUTINE CURVE(X,FX,TABLE,N)
      DIMENSIØN TABLE(2,101)
      IF (N)3,15,4
    3 IX=2
      IFX=1
      N=-N
      GØ TØ 6
    4 IX=1
      IFX=2
    6 FX=0.
      IF (X-TABLE(IX,1))30,12,40
   30 FACT=-1.
      GØ TØ 50
   40 FACT=1.
   50 DØ 10 I=2,N
      IF ((X-TABLE(IX,I))*FACT)5,2,10
    5 FX=TABLE(IFX,I-1) + (TABLE(IFX,I)-TABLE(IFX,I-1))
&    *(X-TABLE(IX,I-1))/(TABLE(IX,I)-TABLE(IX,I-1))
      RETURN
    2 FX=TABLE(IFX,I)
      RETURN
   10 CØNTINUE
   15 FX=0.
      RETURN
   12 FX=TABLE(IFX,1)
      RETURN
      STØP
      END
```

Soil Subroutines FØIL and CØIL:

The standard WES cone penetrometer equations, with their supporting concepts, were used for the soil subroutines since field data for the six geographic sites specified by the Army Materiel Command were available in terms of cone index values only. The standard WES field instrument utilizes a 30-deg cone having a base area of 0.5 sq. in. The strength of fine-grained soils is expressed as the average penetration resistance for a "critical layer", selected according to the size and weight of the vehicle with which the number is to be used. For military vehicles, the 0- to 6-in. or the 0- to 12-in. layer, depending on weight and type of vehicle and the soil profile, is usually considered critical.

Penetration resistance is measured by the cone penetrometer and is expressed in terms of cone index. Since the strength of a soil may increase or decrease when loaded or disturbed, remolding tests are necessary to measure the gain or loss of soil strength to be expected under traffic. The result is the rating cone index which is a soil dependent parameter. A comparison of the rating cone index with the vehicle cone index (a vehicle performance characteristic) indicates whether the vehicle can negotiate the given soil condition. Reference contains a detailed account of these concepts.

Subroutine FØIL: (Fig. C9)

Subroutine FØIL calculates the soil-dependent tractive force versus vehicle velocity array for fine-grained soil. The program is divided into three parts: the first part calculates the mobility index and vehicle cone index, the second part calculates drawbar pull-to-weight ratio and resistance-to-weight ratio, and the third part uses the incoming array FØRCE, which is the tractive force versus theoretical vehicle velocity curve, and the previous calculations to create a new array FØRCR. This array has four columns: the first represents vehicle velocity on level ground corrected for slip in the soil, the second the corresponding value of vehicle tractive effort, the third the vehicle velocity on a slope corrected for soil slippage, and the fourth the corresponding vehicle tractive effort. The calculations, following the WES soil model, use cone indexes. The first part of the program calculates mobility index, XMI, which depends on a number of factors and is calculated differently for tracked and wheeled vehicles. For a tracked vehicle, the contact pressure factor, CPF, is set equal to the vehicle weight divided by the quantity two times the track length times the track width. Next, the weight factor, WF, is calculated: if the vehicle weight is less than 50,000 lb., WF = 1; if it weighs between 50,000 and 70,000 lb., WF = 1.2; if it weighs between 70,000 and 100,000 lb., WF = 1.4; and if it weighs more than 100,000 lb., WF = 1.8. Then, the track factor, TF, is set equal to track width divided by 100. Next, the grouser factor is set equal to 1 if the grouser height is less than 1.5 in., and to 1.1 if the grouser height is greater than 1.5 in. The bogie factor is then set equal to gross vehicle weight divided by the quantity 10 times the number of bogies times the area of one track shoe.

For a wheeled vehicle, the contact pressure factor, CPF, is set equal to two times the gross vehicle weight divided by the quantity nominal wheel width times nominal wheel diameter times number of tires. Next, the weight factor, WF, is calculated. This depends on the weight range, WR, which is equal to gross weight in pounds divided by the number of axles. (The weight range is also expressed as WX, the vehicle weight in kips divided by the number of axles.)

If WR is less than 2,000 lb., WF = .553 * WX;

If WR is between 2,000 and 13,500 lb., WF = .033 * WX + 1.05;

If WR is between 13,500 and 20,000 lb., WF = .142 * WX - .42;

and, if WR is greater than 20,000 lb., WF = .278 * WX - 3.115.

The tire factor, TF, is equal to 10 plus the nominal tire width divided by 100. The grouser factor, GF, is set to 1 if the vehicle is not supplied with chains, and to 1.05 if the vehicle has chains. Next, the wheel load factor is set to WX divided by 2.

For both wheeled and tracked vehicles, the following factors are calculated: the clearance factor, CLF, is set equal to the ground clearance divided by 10; and the engine factor, EF, is set equal to 1 if the horsepower per ton is greater than 10, and to 1.05 otherwise. The transmission factor, TFX, is set equal to 1 if the transmission is hydraulic, and to 1.05 if the transmission is mechanical. Finally, for both wheeled and tracked vehicles, the mobility index, XMI, is calculated as follows:

$$XMI = \left[\frac{CPF * WF}{TF * GF} + WL\emptyset RBF - CLF \right] * EF * TFX$$

The one-pass vehicle cone index, variable VCI, is calculated next. For a tracked vehicle:

$$VCI1 = 7. + .2 * XMI - \frac{39.2}{XMI + 5.6}$$

and for a wheeled vehicle:

$$VCI1 = 11.48 + .2 * XMI - \frac{39.2}{XMI + 3.74}$$

Then, it is determined if there is excess RCI available. This variable, RCIX, is set equal to the incoming soil RCI minus the one-pass vehicle cone index. If this value is greater than zero, calculation proceeds; if not, no further calculation is performed within the subroutine. A variable IGØ is set equal to 0, indicating immobilization in the soil, and a return is made to the calling program. If there is excess RCI, the next variables calculated are: DØW, the drawbar pull-weight ratio; CX, maximum 20-pass drawbar pull-to-weight ratio; and CXP, maximum 100-pass drawbar pull-to-weight ratio. If the vehicle is tracked, and contact pressure factor is less than 4,

$$XX = .544 + .0463 * RCIX$$

$$D\emptyset W = XX \doteq SQRT(XX * XX - .0702 * RCIX)$$

$$CX = .758$$

$$CXP = .71$$

If the vehicle is tracked, and the contact pressure factor is greather than 4:

XX = .4554 + .0392 * RCIX

DØW = XX - SQRT(XX * XX - .0526 * RCIX)

CX = .671

CXP = .71

If the vehicle is wheeled, and the contact pressure factor is less than 4:

XX = .3885 + .0265 * RCIX

DØW = XX - SQRT(XX * XX - .0358 * RCIX)

CX = .674

CXP = .76

If the vehicle is wheeled, and the contact pressure factor is greater than 4:

XX = .379 + .0219 * RCIX

DØW = XX - SQRT(XX * XX - .0257 * RCIX)

CX = .585

CXP = .655

Next, the resistance-to-weight ratio, RTØW, is calculated. If the vehicle is wheeled, and the contact pressure factor is less than 4:

$$RTØW = \frac{.861}{RCIX + 3.249} + .035$$

If the vehicle is wheeled, and the contact pressure factor is greater than 4, or if the vehicle is tracked:

$$RTØW = \frac{2.3075}{RCIX + 6.5} + .045$$

The total soil resistance:

RT = RTØW * VW

and the correction factor used in later slip calculations:

CF = RTØW + DØW - CX

are then calculated.

The last part of the subroutine calculates the soil-dependent tractive force versus vehicle velocity array FØRCR. This part is passed through twice. The first time, columns 1 and 2 of the array are calculated: column 1 contains the vehicle velocity on level ground corrected for slippage in the soil, and index IVEL is set to 1; column 2 contains the corresponding tractive force, and index IFØR is set to 2. On the second pass, columns 3 and 4 are calculated: column 3 contains the vehicle velocity on a slope corrected for soil slippage, and index IVEL is set to 3; column 4 contains the corresponding values of tractive force, and index IFØR is set to 4. Within the loop, the first variable calculated is TRØR, which contains the maximum tractive force that can be derived from the soil. TRØR has an index from 1 to 3, representing downslope, level, and upslope, in that order. Next, a loop is begun with index I, which goes from 1 to 101; this is the number of points in array FØRCE. First, a variable FØRK is established. This variable will temporarily hold the value of FØRCE (1, I), which is the incoming tractive force value. If this tractive force is zero, the corresponding locations for velocity and tractive force in array FØRCR are set to zero, and a return is made to the top of the loop. If the force is not zero, calculation proceeds. Next, FØRK is checked against TFØR. If it is greater than TFØR, it is set equal to TFØR. This will produce a flat spot at the top of the final tractive force curve and assure that this curve will not carry values that exceed the force that can be generated in the soil. Next, the slippage in the soil is calculated. If the vehicle is tracked, and the contact pressure factor is less than 4:

$$Y = FØRK/VW - CF$$

$$SLIP = .0257 * Y - .0161 + \frac{.01519}{.8353 - Y}$$

If the vehicle is tracked, and the contact pressure factor is greater than 4:

$$Y = FØRK/VW - CF$$

$$SLIP = .0733 * Y - .0063 + \frac{.00734}{.7177 - Y}$$

If the vehicle is wheeled, and the contact pressure factor is less than 4:

$$Y = FØRK/VW - CF$$

$$SLIP = .0621 * Y - .021 + \frac{.01888}{.7794 - Y}$$

If the vehicle is wheeled, and the contact pressure factor is greater than 4:

$$Y = FØRK/VW - CF$$

$$SLIP = .084 * Y - .016 + \frac{.01414}{.6697 - Y}$$

Finally, the values of velocity and force are loaded into array FØRCR:

FØRCR(IVEL, I) = FØRCE (2, I) * (1. - SLIP)

FØRCR(IFØR, I) = FØRK;

and a return is made to the top of the loop. When this loop is completed, another small loop is started. This loop generates values of FØRMX(I), where I = 1 to 3 for downslope, level and upslope. FØRMX carries the maximum force that the vehicle can generate, depending both on the soil and on the capability of the vehicle. Last, the gross vehicle weight is corrected for slope, and a return is made for the second pass through the outside loop. Here, all calculations are repeated for the vehicle on a slope. When this is finished, a return is made to the calling program.

SUBRØUTINE FØIL

VARIABLES ENTERING: GRADE, RCI, NVEH, GVW, DL, WID, GT, A, NBC, GC, HPT, ITVAR, FØRCE (2, 101)

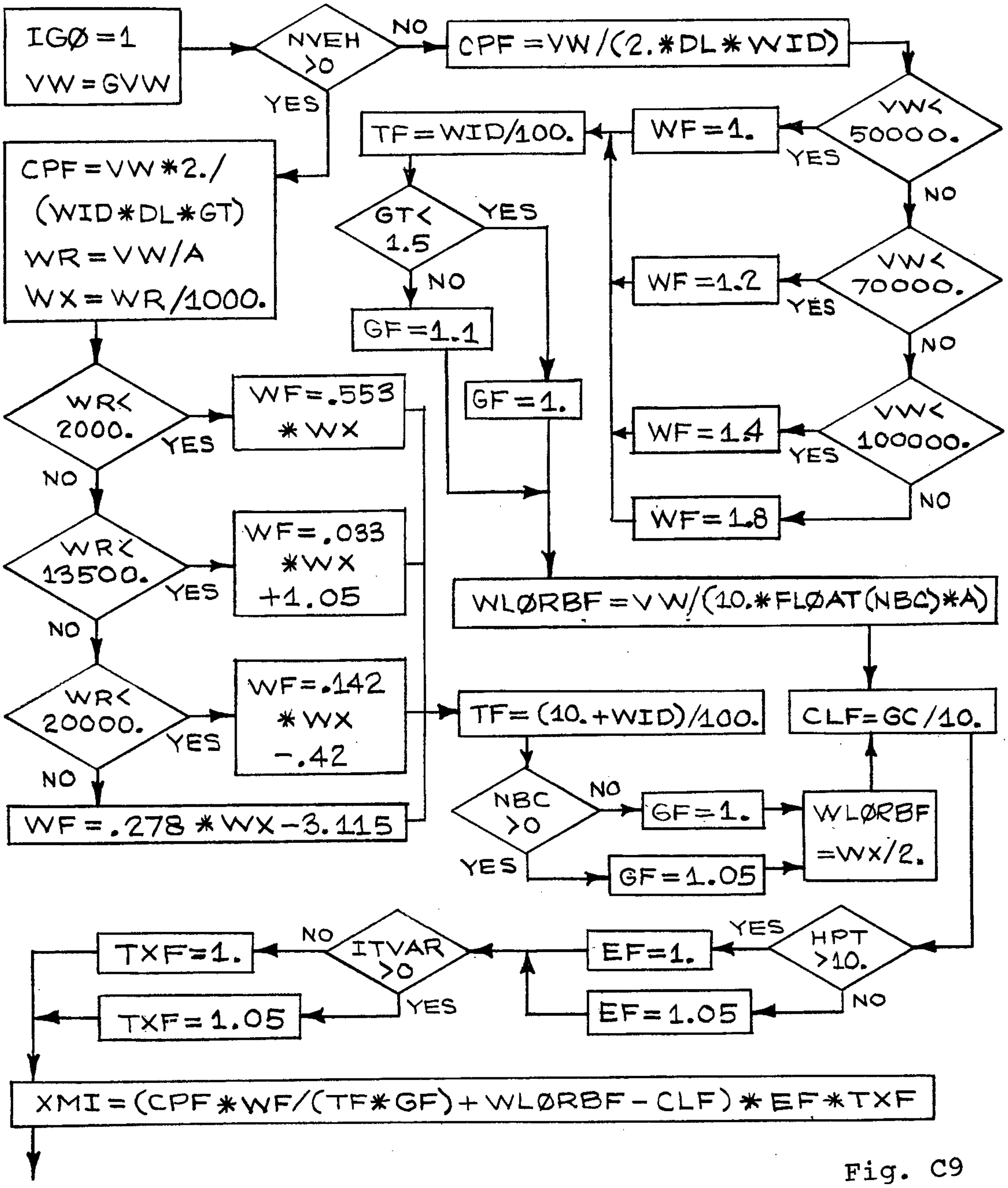


Fig. C9

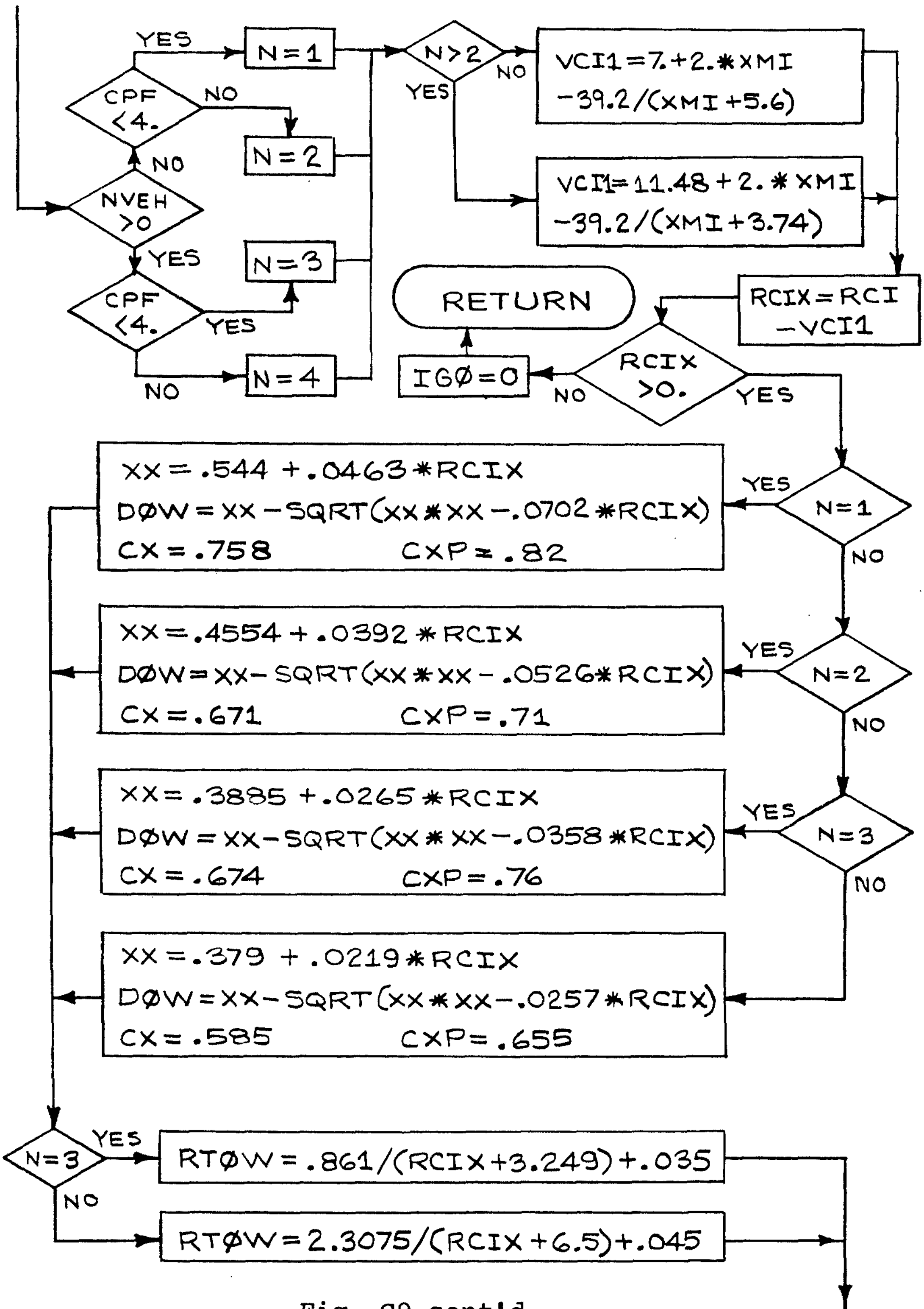
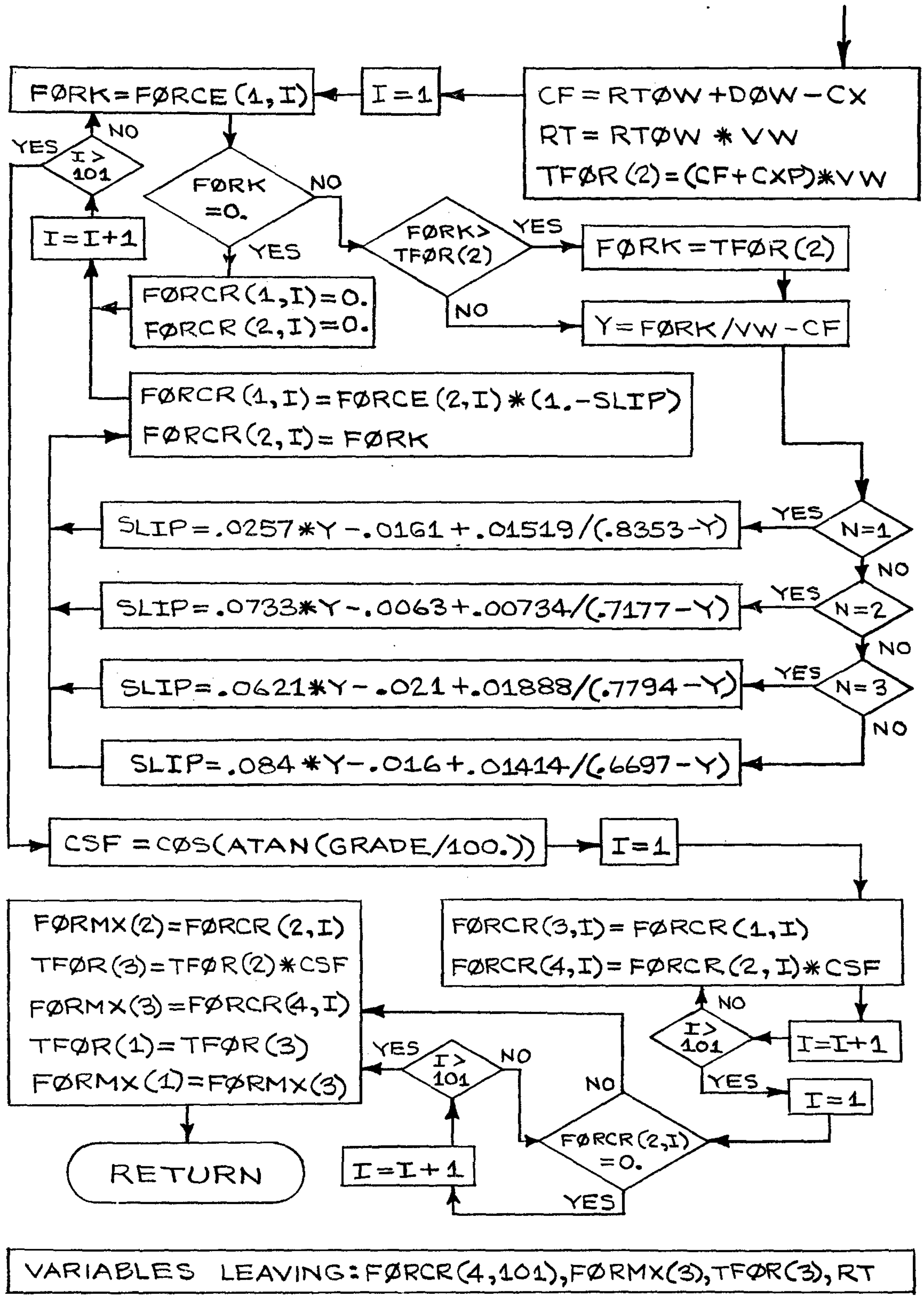

NVEH >0
YES
NO
CPF <4.
YES
NO
N=1
N=2
CPF <4.
YES
NO
N=3
N=4
N>2
YES
NO
VCI1=7.+2.*XMI -39.2/(XMI+5.6)
VCI1=11.48+2.*XMI -39.2/(XMI+3.74)
RCIX=RCI -VCI1
RCIX >0.
NO
YES
IGØ=0
RETURN
N=1
YES
NO
XX=.544+.0463*RCIX
DØW=XX-SQRT(XX*XX-.0702*RCIX)
CX=.758 CXP=.82
N=2
YES
NO
XX=.4554+.0392*RCIX
DØW=XX-SQRT(XX*XX-.0526*RCIX)
CX=.671 CXP=.71
N=3
YES
NO
XX=.3885+.0265*RCIX
DØW=XX-SQRT(XX*XX-.0358*RCIX)
CX=.674 CXP=.76
XX=.379+.0219*RCIX
DØW=XX-SQRT(XX*XX-.0257*RCIX)
CX=.585 CXP=.655
N=3
YES
NO
RTØW=.861/(RCIX+3.249)+.035
RTØW=2.3075/(RCIX+6.5)+.045

Fig. C9 cont'd



CF = RTØW + DØW − CX
RT = RTØW * VW
TFØR (2) = (CF + CXP)*VW
I = 1
FØRK = FØRCE (1,I)
FØRK =0.
NO
YES
FØRK> TFØR(2)
YES
NO
FØRK = TFØR (2)
Y = FØRK / VW − CF
FØRCR (1,I) = 0.
FØRCR (2,I) = 0.
I > 101
YES
NO
I = I+1
FØRCR (1,I) = FØRCE (2,I) * (1.-SLIP)
FØRCR (2,I) = FØRK
N=1
YES
NO
SLIP = .0257*Y −.0161 + .01519/(.8353−Y)
N=2
SLIP = .0733*Y −.0063 + .00734/(.7177−Y)
N= 3
SLIP = .0621*Y − .021 + .01888/(.7794−Y)
SLIP = .084*Y −.016 + .01414/(.6697−Y)
CSF = CØS(ATAN (GRADE/100.))
I=1
FØRCR (3,I) = FØRCR (1,I)
FØRCR (4,I) = FØRCR (2,I)*CSF
I=I+1
I> 101
I = 1
FØRCR (2,I) =0.
I = I+1
FØRMX (2) = FØRCR (2,I)
TFØR (3) = TFØR (2)*CSF
FØRMX (3) = FØRCR (4,I)
TFØR (1) = TFØR (3)
FØRMX (1) = FØRMX (3)
RETURN
VARIABLES LEAVING: FØRCR(4,101), FØRMX(3), TFØR(3), RT

Fig. C9 cont'd

FØIL

```
      SUBRØUTINE FØIL(GRADE,IGØ)
      DIMENSIØN FØRCE(2,101),FØRCR(4,101)
      CØMMØN IPATCH(325),FØRCE,FØRCR,FØRMX(3),TDØR(3),RT,RCI,
&  NVEH,NFL,GVW,DL,WID,GT,A,NBC,GC,HPT,ITVAR,RDIAM,TPSI,TPLY,
&  HS,WC,SAI,AWPKF,GCA,VSS,NCREW,FD,VFS
      IGØ=1
      VW=GVW
    1 IF(NVEH.NE.0) GØTØ 200
      CPF=VW/(2.*DL*WID)
      IF(VW.LT.50000.)WF=1.0
      IF(VW.GE.50000..AND.VW.LT.70000.)WF=1.2
      IF(VW.GE.70000..AND.VW.LT.100000.)WF=1.4
      IF(VW.GE.100000.)WF=1.8
      TF=WID/100.
      IF(GT.LT.1.5)GF=1.0
      IF(GT.GE.1.5)GF=1.1
      WLØRBF=VW/(10.*FLØAT(NBC)*A)
      GØTØ 210
200   CPF=VW*2./(WID*DL*GT)
      WR]=]T
      WX=WR/1000.
      IF(WR.LT.2000.)WF=.553*WX
      IF(WR.GE.2000..AND.W=Y;FM00.)WF=.033*WX+1.05
      IF(WR.GE.13500..AND.WR.LT.20000.)WF=.142*WX-.42
      IF(WR.GE.20000.)WF=.278*WX-3.115
      TF=(10.+WID)/100.
      IF(NBC.EQ.0)GF=1.0
      IF(NBC.DQ.1)GF=1.05
      WLØRBF=WX/2.
210   CLF=GC/10.
      IF(HPT.GT.10.)EF=1.0
      IF(HPT.LE.10.)EF=1.05
      IF(ITVAR.EQ.0)TXF=1.0
      IF(ITVAR.EQ.1)TXF=1.05
      XMI=(CPF*WF/(TF*GF)+WLØRBF-CLF)*EF*TXF
      IF(NVEH.EQ.0.AND.CPF.LT.4.)N=1
      IF(NVEH.EQ.0.AND.CPF.GE.4.)N=2
      IF(NVEH.GT.0.AND.CPF.LT.4.)N=3
      IF(NVEH.GT.0.AND.CPF.GE.4.)N=4
      GØTØ(220,220,230,230),N
220   VCI1=7.+.2*XMI-39.2/(XMI+5.6)
      GØTØ 240
230   VCI1=11.48+.2*XMI-39.2/(XMI+3.74)
240   RCIX=RCI-VCI1
      IF (RCIX.LE.0.) GØ TØ 600
      GØTØ(250,260,270,280),N
250   XX=.544+.0463*RCIX
      DØW=XX-SQRT(XX*XX-.0702*RCIX)
      CX=.758
      CXP=.82
```

```
      GØTØ 290
  260 XX=.4554+.0392*RCIX
      DØW=XX-SQRT(XX*XX-.0526*RCIX)
      CX=0.671
      CXP=.71
      GØTØ 290
  270 XX=.3885+.0265*RCIX
      DØW=XX-SQRT(XX*XX-.0358*RCIX)
      CX=.674
      CXP=.76
      GØTØ 290
  280 XX=.379+.0219*RCIX
      DØW=XX-SQRT(XX*XX-.0257*RCIX)
      CX=.585
      CXP=.655
  290 GØ TØ (310,310,300,310),N
  300 RTØW=0.861/(RCIX+3.249) + 0.035
      GØTØ 320
  310 RTØW=2.3075/(RCIX+6.5) + 0.045
  320 CF=RTØW+DØW-CX
      RT=RTØW*VW
      TFØR(2)=(CF+CXP)*VW
      DØ 420 I=1,101
      FØRK=DØRCE(1,I)
      IF (FØRCE(1,I))330,410,330
  330 IF(FØRCE(1,I).LE.TFØR(2))GØTØ 350
      FØRK=TFØR(2)
  350 Y=FØRK/VW - CF
      GØ TØ (360,370,380,390),N
  360 SLIP=0.0257*Y-0.0161+0.01519/(0.8353-Y)
      GØ TØ 400
  370 SLIP=0.0733*Y-0.0063+0.00734/(0.7177-Y)
      GØ TØ 400
  380 SLIP=0.0621*Y-0.021+0.01888/(0.7794-Y)
      GØ TØ 400
  390 SLIP=0.084*Y-0.016+0.01414/(0.6697-Y)
  400 FØRCR(1,I)=FØRCE(2,I)*(1.-SLIP)
      FØRCR(2,I)=FØRK
      GØ TØ 420
  410 FØRCR(1,I)=0.
       FØRCR(2,I)=0.
  420  CØNTINUE
       CSF=CØS(ATAN(GRADE/100.))
       DØ 1000 I=1,101
       FØRCR(3,I)=FØRCR(1,I)
 1000  FØRCR(4,I)=FØRCR(2,I)*CSF
  520  DØ 530 I=1,101
       IF(FØRCR(2,I).NE.0.)GØTØ 540
  530  CØNTINUE
  540  FØRMX(2)=FØRCR(2.I)
       TFØR(3)=TFØR(2)*CSF
       FØRMX(3)=FØRCR(4,I)
       TFØR(1)=TFØR(3)
       FØRMX(1)=FØRMX(3)
       RETURN
  600  IGØ=0
       RETURN
       END
```

Subroutine CØIL (Fig. C10)

Subroutine CØIL calculates the soil-dependent tractive force versus vehicle velocity array for a coarse-grained soil. If the vehicle is tracked and has a flexible track, the drawbar pull-to-weight ratio is set equal to 0.568, and the resistance-to-weight ratio is set equal to 0.074. If the vehicle has a non-flexible track, the drawbar pull-to-weight ratio is set equal to 0.695, and the resistance-to-weight ratio is set equal to 1. For either type of track, the tractive force-to-weight ratio, TFØW, is set equal to drawbar pull-to-weight ratio plus resistance-to-weight ratio.

If the vehicle is wheeled, the calculation is somewhat more complicated and depends on several factors. First, if the ratio of nominal wheel width to wheel rim diameter is greater than 2.4, the tire factor, FAC7, is set equal to 5. If this ratio is less than 2.4, FAC7 is set equal to 2. Then, the wheel diameter factor, WDF, is calculated as follows:

WDF = FAC7 * WID + RDIAM

The contact pressure factor, CPF, is defined by:

CPF = .607 * TPSI + 1.35 * (117. * TPLY/WDF) - 4.93

The contact area factor, CAF, is defined by:

CAF = log (VW/CPF)

The strength factor, SF, is defined by:

SF = .0526 * GT + .0211 * TPSI - .35 * CAF + 1.587

And the one-pass cone index, VCI1, becomes:

VCI1 = 10. ** SF

Finally, it is determined if there is any excess RCI available. The variable RCIX is set equal to the incoming soil RCI minus VCI1. If there is no excess RCI, variable IGØ is set equal to zero, indicating that the vehicle is immobilized in the soil, and a return is made to the calling program. If there is excess RCI, calculation proceeds.

First, a new strength factor, SF, is set equal to the log of RCI. The maximum towing force is calculated:

TFM = (28.87 * SF + 10.1 * CAF - 1.52 * GT
      - .61 * TPSI - 43.82)/100.

The 20-pass drawbar pull-to-weight ratio is set to 0.56, and the 100-pass drawbar pull-to-weight ratio is set equal to 0.57475. Next, the resistance-to-weight ratio is calculated:

RTØW = (22.2 + .92 * TPSI - (8. + .37 * TPSI) * SF)/
       100.

The correction factor, CF, used in later slip calculations, is also calculated:

CF = RTØW + TFM - DØW 20

Finally, the tractive force-to-weight ratio is calculated:

TFØW = CF + DW 100

For both wheeled and tracked vehicles, the total soil resistance becomes:

RT = RTØW * VW

The last part of the subroutine fills in array FØRCR, the soil-dependent tractive force versus vehicle velocity array. This calculation is identical to the calculation in subroutine FØIL, with the following exceptions. The maximum tractive force available from the soil, TFØR, is here set equal to tractive force-to-weight ratio, TFØW, times gross vehicle weight. Also, the slip equations are different. If the vehicle is wheeled,

Y = FØRK/VW - CF

$$SLIP = .0074 * Y - .0061 + \frac{.00374}{.5785 - Y}$$

If the vehicle is tracked and has a flexible track:

```
Y = FØRK/VW - RTØW

SLIP = 1.074 * Y - .72 + SQRT((1.074 * Y - .72) ** 2
       + .09 * Y + .009)
```

If the vehicle has a non-flexible track:

```
Y = FØRK/VW - RTØW
```

$$SLIP = -.0083 + \frac{.005312}{.573 - Y}$$

The rest of this calculation is identical to that in FØIL.

SUBRØUTINE CØIL

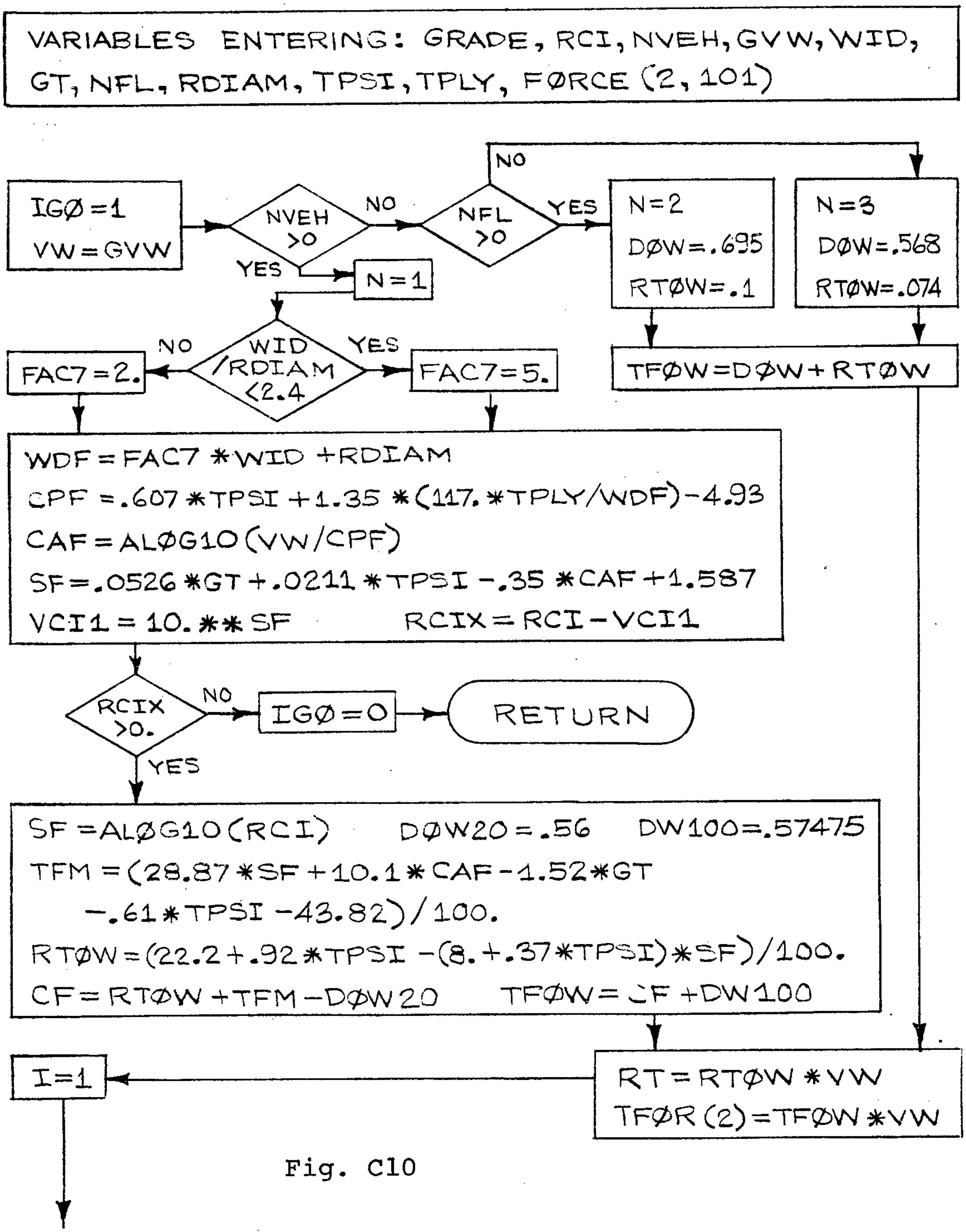


Fig. C10

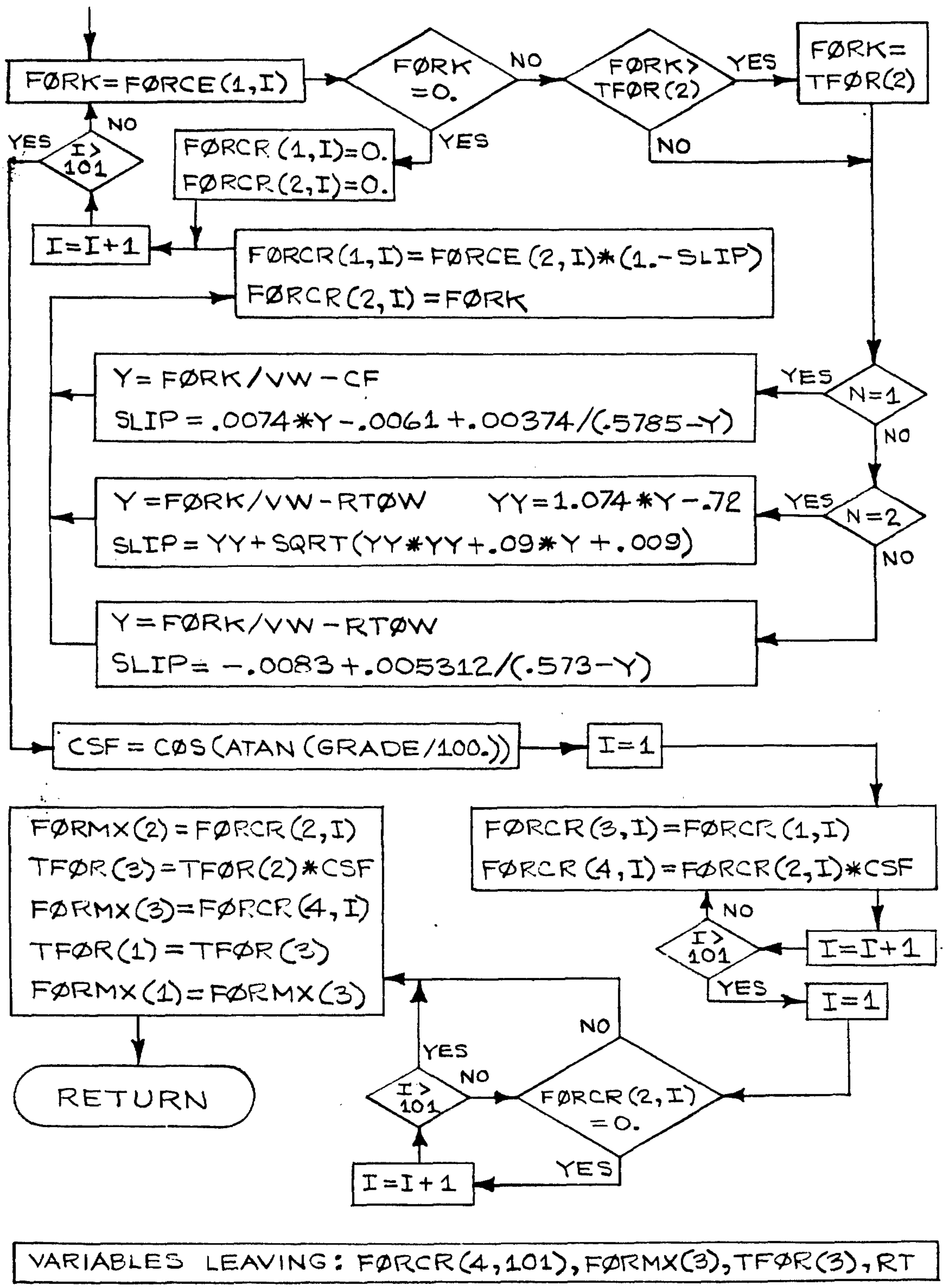


Fig. C10 cont'd

CØIL

```
      SUBRØUTINE CØIL(GRADE,IGØ)
      DIMENSIØN FØRCE(2,101),FØRCR(4,101)
      CØMMØN IPATCH(325),FØRCE,FØRCR,FØRMX(3),TDØR(3),RT,RCI,
     NVEH,NFL,GVW,DL,WID,GT,A,NBC,GC,HPT,ITVAR,RDIAM,TPSI,TPLY,
     HS,WC,SAI,AWPKF,GCA,VSS,NCREW,FD,VFS
      IGØ=1
      VW=GVW
    1 IF (NVEH)2,2,3
    2 IF (NFL)32,32,30
   30 N=2
      DØW=.695
      RTØW=.1
      GØ TØ 34
   32 N=3
      DØW=.568
      RTØW=.074
   34 TFØW=DØW+RTØW
      GØ TØ 18
    3 N=1
      IF (WID/RDIAM-2.4)5,4,4
    4 FAC7=2.0
      GØ TØ 104
    5 FAC7=5.0
  104 WDF=FAC7*WID+RDIAM
      CPF=0.607*TPSI+1.35*(117.*TPLY/WDF)-4.93
      CAF=ALØG10(VW/CPF)
      SF=0.0526*GT+0.0211*TPSI-0.35*CAF+1.587
      VCI1=10.**SF
    7 RCIX=RCI-VCI1
    8 IF (RCIX)9,9,10
    9 IGØ=0
      RETURN
   10 SF=ALØG10(RCI)
      TFM=(28.87*SF+10.1*CAF-1.52*GT-0.61*TPSI-43.82)/100.
   13 DØW20=0.56
   14 DW100=0.57475
   15 RTØW=(22.2+0.92*TPSI-(8.+0.37*TPSI)*SF)/100.
   16 CF=RTØW+TDM-DØW20
   17 TFØW=CF+DW100
   18 RT=RTØW*VW
      TFØR(2)=TFØW*VW
      DØ 420 I=1,101
      FØRK=FØRCE(1,I)
      IF (FØRCE(1,I))410,410,330
  330 IF(FØRCE(1,I).LE.TFØR(2))GØTØ 340
      FØRK=TFØR(2)
  340 GØ TØ (24,37,38),N
   24 Y=FØRK/VW-CF
      SLIP=0.0074*Y-0.0061+0.00374/(0.5785-Y)
      GØ TØ 25
```

CØIL CØNTINUED

```
   37 Y=DØRK/VW-RTØW
      SLIP=1.074*Y-.72+SQRT((1.074*Y-.72)**2+.09*Y+.009)
      GØ TØ 25
   38 Y=FØRK/VW-RTØW
      SLIP=.005312/(.573-Y)-.0083
   25 FØRCR(1,I)=FØRCE(2,I)*(1.-SLIP)
      FØRCR(2,I)=FØRK
      GØ TØ 420
  410 FØRCR(1,I)=0.
      FØRCR(2,I)=0.
  420 CØNTINUE
      CSF=CØS(ATAN(GRADE/100.))
      DØ 1000 I=1,101
      FØRCR(3,I)=FØRCR(1,I)
 1000 FØRCR(4,I)=FØRCR(2,I)*CSF
      DØ 530 I=1,101
      IF(FØRCR(2,I).NE.0.)GØTØ 540
  530 CØNTINUE
  540 FØRMX(2)=FØRCR(2,I)
      TFØR(3)=TFØR(2)*CSF
      FØRMX(3)=FØRCR(4,I)
      TFØR(1)=TFØR(3)
      FØRMX(1)=FØRMX(3)
      RETURN
      END
```

Subroutine PATCH (Fig. C11)

In using subroutine PATCH, values of several constants that are used later in the program are calculated first: the acceleration due to gravity at 32.16 ft/sec^2; two conversion factors for changing velocity in miles per hour to feet per second and the reverse (CØNF1 and CØNF2); two values for NSDC, the number of stem diameter classes, that are used as limits on the loops (NSDCM, which is NSDC - 1, and NSDCP, which is NSDC + 1); and the vehicle mass, VM, which is gross vehicle weight divided by g, acceleration due to gravity. Several variables are necessary in the analysis: ØBL, the obstacle length; ØBW, the obstacle width; ØBS, the obstacle spacing; H, the obstacle height; ØBAA, the obstacle approach angle; and GRADI, the slope class for this patch type from the patch data. Also required are: VRID, the velocity limited by ride dynamics or surface roughness; and S(I), the mean spacing of all stems of stem diameter class I or larger. Also, the value of XNT(I) must be set; this is the number of trees of stem diameter class I in an area containing one tree of the largest class. Then, the value of SDS(I), the mean spacing for stem diameter class I, is calculated for all values of I from 1 to NSDC.

Next, there is a loop with index K, whose values 1 to 3 define the slope -- downslope, level, or upslope -- in that order. Within this loop, the forces of resistance on slopes and the velocity limited by vision are calculated. (Also, the values of the variable VELØ(K) are initiated at zero within this loop. This is extraneous to the calculations in this loop, but it is conveniently done here.)

The first subroutine entered in this loop is subroutine HILL, in which the resistance due to the slope and the soil is calculated and stored in variable RGU(K), K being 1, 2 and 3 as before. Return is made to subroutine PATCH, and the braking force, BRFØR, that is necessary in the VISIØN subroutine is calculated; BRFØR is the braking force that can be generated in the soil, and is equal to the maximum tractive force generated in the soil plus the resistance

RGU(K), minus RTS, which is the soil resistance altered by the angle of the slope. BRFØR is compared with the variable XBR, the braking force that the vehicle can produce with its own brakes; and the least of these two values is taken to be the final braking force. This value is sent into subroutine VISIØN.

In subroutine VISIØN, the variable VELV, the velocity limited by visibility, is calculated. This is the initial velocity which, given the braking force available and the resultant deceleration, will bring the vehicle to a velocity of zero within the recognition distance, and is then the maximum velocity due to recognition. This calculation is for "downslope".

Next, the grade is indexed upward, and is set to "level". HILL and VISIØN are gone through again, and new values of RGU(K), BRFØR, and VELV for level ground are calculated. After this is done, the grade is set to "upslope", and the same procedure as before is followed through subroutines HILL and VISIØN. When this loop is completed, there are available RGU(K), the resisting force due to soil and the slope, and VELV(K), the velocity limited by vision. Both of these have three values - downslope, level and upslope.

The next part of the subroutine consists of a series of nested loops, and the final outcome of all calculations performed within these loops is the variable VELØ(K), K again denoting downslope, level and upslope. The variable VELØ is initially set to zero. Now, within these loops, various temporary values of velocity are calculated. These velocities are dependent, first, on whether obstacles are avoided or overridden; this information is carried in index J, which has the values 1 and 2. Secondly, they are dependent on the forces necessary to override vegetation or the area available if the vegetation must be avoided; this is carried by index I, which goes from 1 to the number of stem diameter classes +1. The third index is K, which, as before, carries the values of 1, 2 and 3 for downslope, level and upslope, respectively. The number of stem diameter classes +1 is 9;

determination of whether obstacles are avoided or overridden yields two possible values, and the slope has three possible values; this is a total of 54 temporary velocities that are calculated. At the end of the loops, each of these are compared with the previous value of VELØ; and if it is larger, VELØ is set up to the newly calculated value. If not, a return is made through the loops to calculate the next temporary velocity. Finally, when these loops are completed, VELØ carries the maximum velocity, given all the considerations just described.

The first loop entered is that carrying index J to determine whether obstacles will be avoided or overridden. The first time through, J is checked to see if it is 1. If so, subroutine AREAØ is entered; here, the percentage of the area denied by obstacles, ADØ, is calculated. (Subroutine PATCH is entered every time a new patch is being calculated, so only one obstacle size is used here for the given patch.) If J is 2, subroutines ØBSTCL and ØBSF are entered instead of AREAØ.

In ØBSTCL, the value IGØ, which signifies a go or no-go condition, is calculated. If IGØ is 0 and the vehicle cannot negotiate the obstacle type geometrically, no further calculation is performed, and a return is made to the beginning of the loop. If IGØ is 1, meaning there is no geometric interference, subroutine ØBSF is entered; here, the force required to overcome the obstacle is calculated and stored in variable FØM. Next, subroutine CURVE is entered; here, the maximum velocity limited by 2.5-g vertical acceleration at the driver's seat is calculated and stored in variable VØLA. The two possibilities - whether obstacles are to be avoided or overridden - have now been calculated. The variables returning from this part of the calculation are: ADØ, the percentage of the area denied by obstacles; FØM, the force required to overcome obstacles, and VØLA, the velocity limited by vertical acceleration. For J = 1, ADØ is set to zero, FØM is calculated in subroutine ØBSF, and VØLA is calculated in subroutine CURVE.

Now the second loop, which runs through the stem diameter classes, is begun. The first subroutine entered is AREAV, which calculates the percentage of the area denied by vegetation (assuming the vehicle cannot overcome this vegetation); this percentage is stored in variable PAV. Next, subroutine AREAT is entered. The variables sent into subroutine AREAT are: ADØ, the percentage of the area denied by obstacles, and PAV, the percentage of the area denied by vegetation. Within subroutine AREAT, a total area denied is calculated. The variable returning from AREAT is SRF, a speed reduction factor due to maneuvering. This is used later as a multiplier in calculating total velocity. If SRF is equal to zero, no further calculation on this pass through the loop is performed, and a return is made. If SRF has a real value, the calculation proceeds.

The next subroutine entered is VEGF; here, the forces necessary to override vegetation (trees) are calculated, as follows: FAT1, the force required to override a single tree; FMT, the maximum force required to overcome trees; and FAT, the average force required to overcome trees. Now, two checks are made. First, it must be determined whether the force FMT divided by vehicle weight is greater than 2 (2-g horizontal acceleration). If this is exceeded, no further calculation is performed, and a return is made; if not, a check is made to determine whether this maximum force is less than the pushbar force that the vehicle can stand. If this force is less than the pushbar capability, the calculation proceeds; if not, a return is made.

The third loop, carrying index K, is now entered, K being 1 to 3 for downslope, level and upslope, as already explained. The first calculation is the total force of resistance due to the slope, the soil, obstacles and vegetation. This total resistance force is stored in variable TRFU. A check is made to determine whether TRFU, the resisting force, is larger than the maximum force that the vehicle can generate (which is stored in variable FØRMX). If it is larger, no further calculation is performed in this loop, and a return is made; if not, the calculation proceeds. Subroutine KURVE is now entered in this loop; here, the maximum velocity that the vehicle can manage for the given conditions is calculated. This value is stored in variable VTT.

Now, the maximum velocity that can be attained in the patch under consideration, given what has previously been calculated, is determined. This velocity is stored in variable VMTEM and consists of the minimum of: VTT, the velocity available in the soil; VRID, the maximum velocity for the given surface roughness; and VELV(K), the maximum velocity limited by recognition distance. At this point, the obstacle appraoch angle, ØBAA, is checked. If it is less than 17 deg, VØLA is set equal to VMTEM. (It is assumed that there will be no sudden acceleration on such a gradual slope.) Next, a check is made to determine if VMTEM is larger than VØLA (the velocity limited by 2.5-g vertical acceleration). If it is larger, the calculation proceeds; if it is not, the temporary velocity is set to VMTEM, and an exit is made to a lower part of the calculation. This temporary velocity is VTEM(K, J, I); i.e., it is the temporary velocity with the index value on each of the three loops taken into consideration. If VMTEM is larger than VØLA, another check is made before the speed-up/slow-down model is entered.

If the obstacle spacing type is random, the effective spacing, ØBS, is set equal to the area of a circle whose diameter is the mean spacing divided by the vehicle width. A final check is made to determine whether the spacing between obstacles is larger than two times the vehicle length. If it is not, the temporary velocity VTEM is set to VØLA. If the obstacle spacing is greater than two times the vertical length, the speed-up/slow-down computation is performed. The first thing calculated is the maximum braking deceleration that the vehicle can manage. Then, an initial velocity, TVEL1, is set equal to VØLA, the maximum velocity the vehicle can attain going over the obstacle. An initial distance, TDIST, is set equal to two times the vehicle length; and the initial time, TTIME, is set to two times the vehicle length divided by the velocity VØLA.

Subroutine CURVE is now entered. Returning from CURVE is a variable ACCEL, which at this point carries the maximum force that the vehicle can produce at velocity TVEL1. From this is subtracted the total resisting forces, TRFU, and the result divided by the vehicle mass, VM, to produce the acceleration the vehicle can develop. Then a new velocity, TVEL2, is set equal to the previous velocity, TVEL1, plus this acceleration times the time interval of 1 sec. Since the time is incremented at 1-sec intervals, it does not appear in the equation. If TVEL2 exceeds VMTEM, it is set equal to VMTEM. A new distance is then set equal to the previous distance, TDIST, plus the distance the vehicle has progressed in this 1-sec interval.

Next, with the present velocity given, the time available to decelerate, TAD, and the time needed to decelerate, TND, before the next obstacle encountered are calculated. A check is made to see if the time available is greater than the time needed. If it is, the time allowed for acceleration is incremented upward by 1 sec. The starting velocity of TVEL1 is set to TVEL2 (which was just calculated), and a return to KURVE is made for another calculation. A new force, a new acceleration, and a new time available and time needed to decelerate are calculated. This loop is continued at 1-sec intervals until the time available to decelerate equals the time needed to decelerate. At this point, the maximum velocity that the vehicle can achieve between obstacles has been reached. Then, the average velocity is the distance between obstacles, ØBS, divided by the total of the current time, TTIME, plus the remaining time needed to decelerate, TND. This velocity is stored in VTEM(K, J, I).

Three values have now been established: (a) VTEM has been established for the indexes K, J, and I; (b) if the obstacle spacing was not greater than two times the vehicle length, VTEM was set equal to VØLA; and (c) if VMTEM was not greater than VØLA, VTEM was set equal to VMTEM.

It is now necessary to determine if the total resistances exceed the total forward forces that the vehicle can generate, i.e., the total force FØRMX that the vehicle can generate in the soil plus the force derived from its kinetic energy. If the resistances are greater than the forward force the vehicle can generate, no further calculation is done, and a return is made to the early part of the loop. If the vehicle still has enough force to overcome these resistances, VTEM is taken as being established. It is now reduced by the speed reduction factor due to maneuvering, SRF, calculated in subroutine AREAT.

The bottom of the loops has now been reached. A value of VTEM dependent on all the indexes in these loops is now compared with the previously calculated value of VELØ. If this new velocity VTEM is larger than the old calculated velocity VELØ, VELØ is set equal to this currently calculated value of VTEM. If the current VTEM is not greater than VELØ, as previously calculated, the old value of VELØ is retained, this current value of VTEM is discarded, and a return is made to the top of the loops.

At the end of this procedure, (VELØ(K), with K denoting downslope, level and upslope, has been established and contains the maximum velocity the vehicle can achieve, given all these previous considerations. After the looping is completed, it is then necessary to calculate the maximum velocity the vehicle can attain in this patch type; this is carried in variable VELØC. It is first necessary to determine if VELØ (sub 1, 2 or 3) is equal to zero. If any one of these three is equal to zero (it would, of course, usually be the value when on an upslope), the vehicle would be immobilized in this patch, variable VELØC is set to zero, and a return is made to the calling program. If these three values of VELØ are greater than zero, a calculation is performed; it is assumed that the vehicle travels at each of these velocities for the same distance. This average velocity is stored in VELØC, and a return is made to the calling program.

SUBRØUTINE PATCH

VARIABLES ENTERING: SV(10), RD(10), SD(10), NSDC, SDL(10), NSSC, ØS(10), AA(20), ØW(10), ØH(10), ØL(10), SLC(10), ISRC(20), IS(10), IØBL, IØBW, IPR, IØBH, IØBS, IGR, IREC, IØST, FØRCR(4,101), IVEL, FØRMX(3), TFØR(3), RT, GVW, VØØB(2,30), NC4, VRIDE(20), W, PBHT, PBF, VL, IØBAA

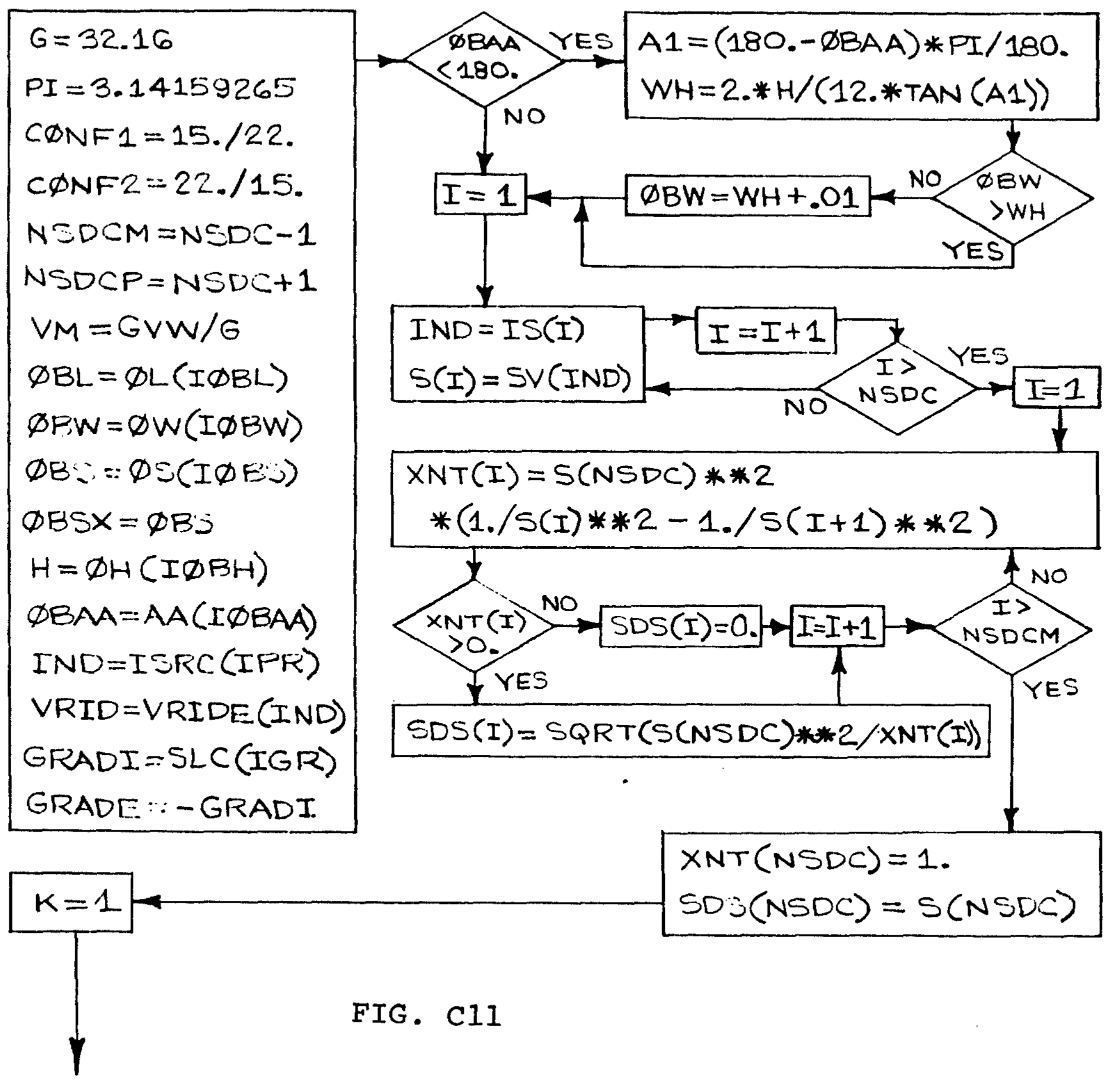


FIG. C11

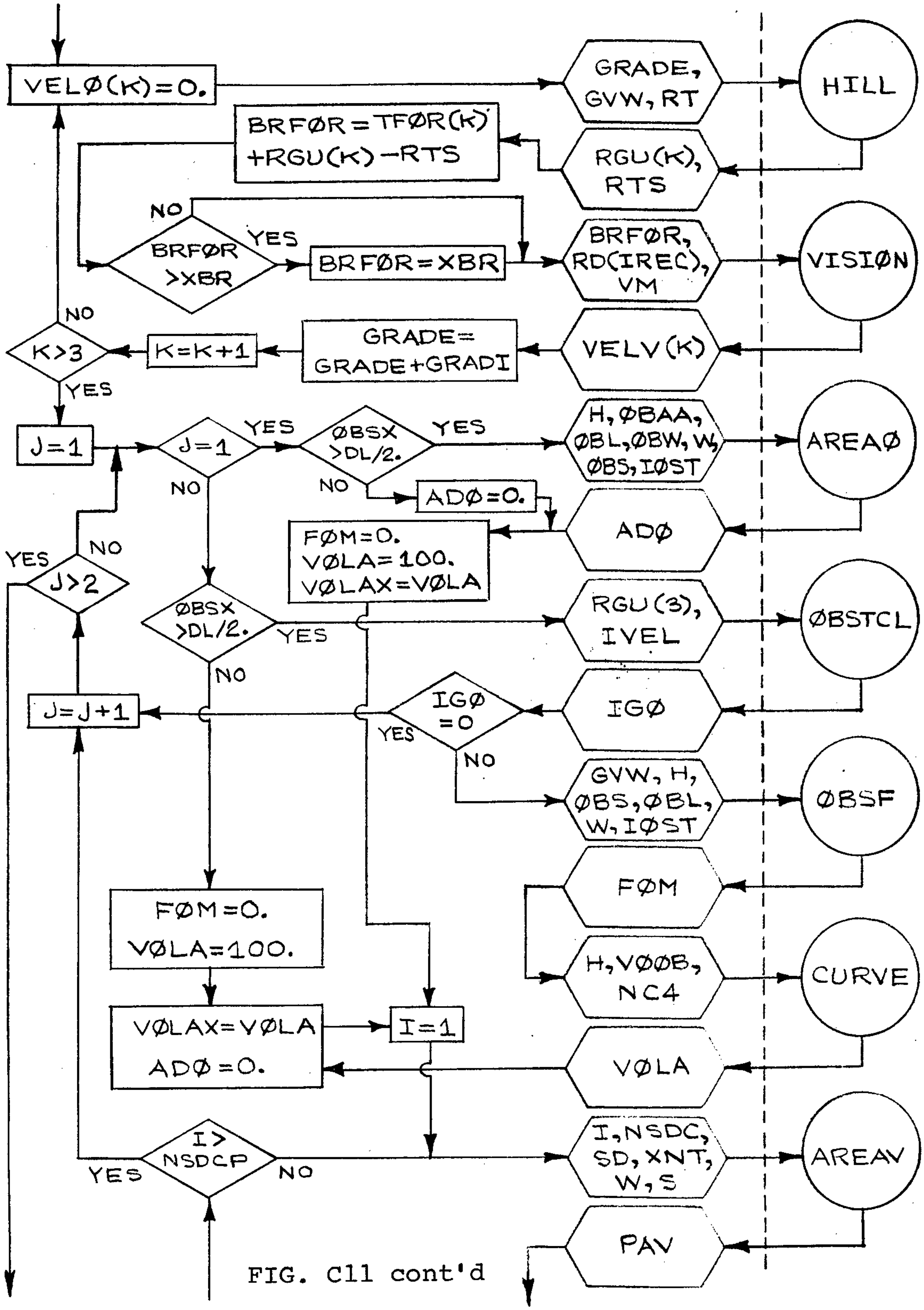


FIG. C11 cont'd

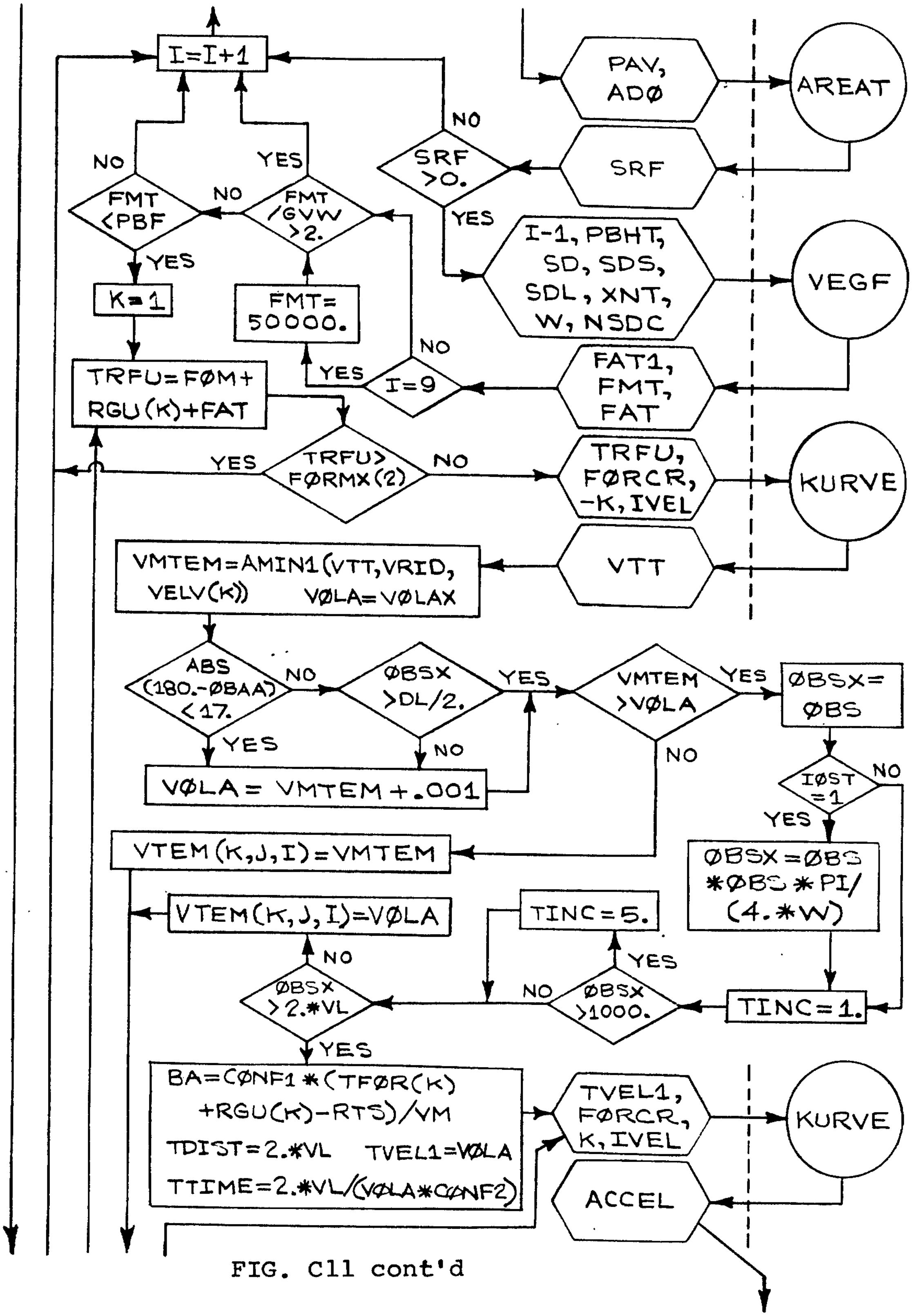
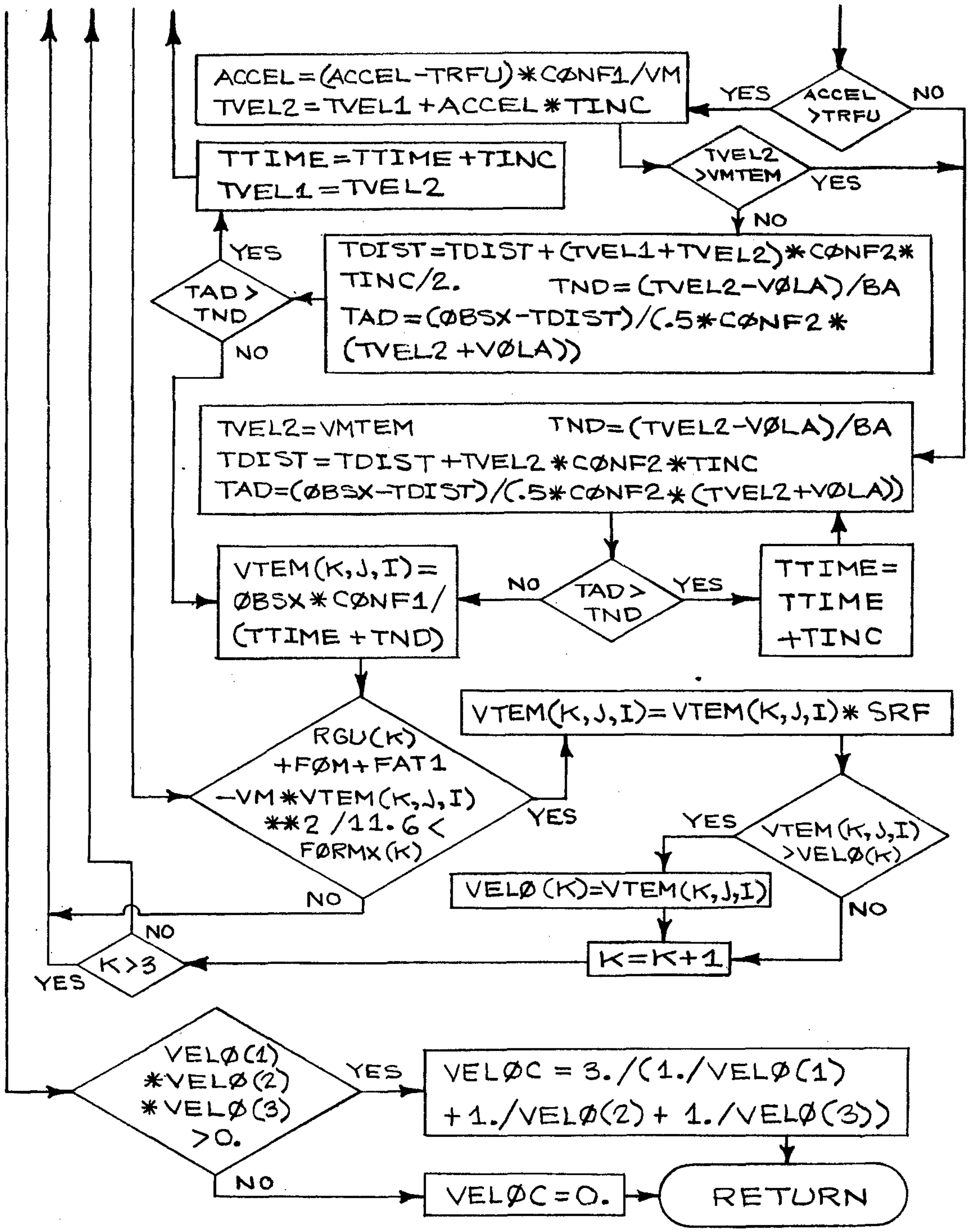


FIG. C11 cont'd



VARIABLES LEAVING: H, ØBW, ØBAA, VELØC

FIG. C11 cont'd

PREP

```
      SUBROUTINE PATCH(VELØC)
      CØMMØN SV(10),RD(10),SD(10),NSDC,SDL(10),NSSC,ØS(10),
&     AA(20),ØW(10),ØH(10),ØL(10),SLC(10),ISRC(20),IS(10),IREC,
&     IØBL,IØBW,IØBS,IØBH,IØBAA,IGR,IPR,IRCI(3),IST,IØST,SDS(10),
&     XNT(10),S(10),FØRCE(2,101),FØRCR(4,101),FØRMX(3),TFØR(3),
&     RT,RCI,NVEH,NFL,GVW,DL,WID,GT,A,NBC,GC,HPT,ITVAR,RDIAM,
&     TPSI,TPLY,HS,WC,SAI,AWPKF,GCA,VSS,NCREW,FD,VFS,TNE1(2,30),
&     TTM(2,30),TTE(2,30),GR(10),NG,TC,RR,FDR,EFF,FDREF,ITRAN,
&     IVEL,NC1,NC2,NC3,ENTCG,LØKUP,VØØB(2,30),VRIDE(20),W,PBHT,
&     PBF,VL,NC4,NC5,H,ØBW,ØBAA,XLT,HB,AV,REC,VDA,CGF,CGH,DWX,RW1,
&     ACG,DCG,HC,RWW,IDUMMY(1555),XBR
      DIMENSIØN VTEM(3,2,11),VELØ(3),VELV(3),RGU(3)
      G=32.16
      CØNF1=15./22.
      CØNF2=22./15.
      NSDCM=NSDC-1
      NSDCP=NSDC+1
      VM=GVW/G
      ØBL=ØL(IØBL)
      ØBW=ØW(IØBW)
      ØBS=ØS(IØBS)
      ØBSX=ØBS
      H=ØH(IØBH)
      ØBAA=AA(IØBAA)
      IF(ØBAA-180.)2000,2001,2001
2000  A1=(180.-ØBAA)*3.14159265/180.
      WH=2.*H*CØS(A1)/(12.*SIN(A1))
      IF(ØBW.GT.WH)GØTØ 2001
      ØBW=WH+.01
2001  IND=ISRC(IPR)
      VRID=VRIDE(IND)
      GRADI=SLC(IGR)
      GRADE=-GRADI
      DØ 100 I=1,NSDC
      IND=IS(I)
      S(I)=SV(IND)
  100 CØNTINUE
      DØ 101 I=1,NSDCM
      XNT(I)=S(NSDC)**2*(1./S(I)**2-1./S(I+1)**2)
      IF (XNT(I))40,40,41
   40 SDS(I)=0.
      GØ TØ 101
   41 SDS(I)=SQRT(S(NSDC)**2/XNT(I))
  101 CØNTINUE
      XNT(NSDC)=1.
      SDS(NSDC)=S(NSDC)
   52 DØ 1001 K=1,3
      VELØ(K)=0.0
      CALL HILL(GRADE,RGU(K),GVW,RT,RTS)
      BRFØR=TFØR(K)+RGU(K)-RTS
```

PREP CØNTINUED

```
      IF(BRFØR.GT.XBR)BRFØR=XBR
      CALL VISIØN(BRFØR,RD(IREC),VM,VELV(K))
      GRADE=GRADE+GRADI
 1001 CØNTINUE
      GRADE=-GRADI
      DØ 1000 J=1,2
      IF (J-1)130,130,140
  130 IF(ØBSX.LE.DL/2.)GØTØ 131
      CALL AREAØ(ØBL,ØBW,ØBS,W,ADØ,IØST,ØBAA,H)
      GØTØ 132
  131 ADØ=0.
  132 FØM=0.0
      VØLA=100.
      VØLAX=VØLA
      GØ TØ 150
  140 IF(ØBSX.LE.DL/2.)GØTØ 148
      CALL ØBSTCL(RGU(3),IVEL,IGØ)
  999 FØRMAT(10X,I3)
      IF (IGØ)1000,1000,145
  145 CALL ØBSF(GVW,H,ØBS,W,ØBL,FØM,IØST)
  149 CALL CURVE(H,VØLA,VØØB,NC4)
      GØTØ 5183
  148 FØM=0.
      VØLA=100.
 5183 VØLAX=VØLA
      ADØ=0.
  150 DØ 300 I=1,NSDCP
      CALL AREAV(I,SD,NSDC,XNT,W,S,PAV)
      CALL AREAT(PAV,ADØ,SRF)
      IF(SRF)300,300,160
  160 CALL VEGF(I-1,PBHT,SD,SDS,SDL,XNT,W,FAT1,FMT,FAT,NSDC)
      IF(I.EQ.9)FMT=500000.
      IF(FMT/GVW-2.)165,165,300
  165 IF(FMT-PBF(170,300,300
  170 DØ 290 K=1,3
      TRFU=RGU(K)+FØM+FAT
      IF (TRFU-FØRMX(K))171,171,300
  171 CALL KURVE(TRFU,VTT,FØRCR,-K,IVEL)
      VMTEM=AMIN1(VTT,VRID,VELV(K))
      VØLA=VØLAX
       IF(ABS(180.-ØBAA).LT.17.)VØLA=VMTEM+.001
       IF(ØBSX.LE.DL/2.)VØLA=VMTEM+.001
       IF (VMTEM-VØLA)190,190,215
   190 VTEM(K,J,I)=VMTEM
       GØTØ 200
   215 ØBSX=ØBS
       IF(IØST.EQ.1)ØBSX=ØBS*ØBS*3.14159265/(4.*W)
       TINC=1.
       IF(ØBSX.GE.1000.)TINC=5.
       IF(ØBSX-2.*VL)220,220,225
```

PREP CØNTINUED

```
 220 VTEM(K,J,I)=VØLA
     GØTØ 200
 225 BA=CØNF1*(TFØR(K)+RGU(K)-RTS)/VM
     TVEL1=VØLA
     TTIME=2.*VL/(VØLA*CØNF2)
     TDIST=2.*VL
 240 CALL KURVE(TVEL1,ACCEL,FØRCR,K,IVEL)
     IF(ACCEL.LE.TRFU)GØTØ 235
     ACCEL=(ACCEL-TRFU)*CØNF1/VM
     TVEL2=TVEL1+ACCEL*TINC
     IF(TVEL2.GT.VMTEM)GØTØ 235
     TDIST=TDIST+(TVEL1+TVEL2)*CØNF2*TINC/2.
     TAD=(ØBSX-TDIST)/(.5*CØNF2*(TVEL2+VØLA))
     TND=(TVEL2-VØLA)/BA
     IF (TAD-TND)250,250,230
 230 TTIME=TTIME+TINC
     TVEL1=TVEL2
     GØ TØ 240
 235 TVEL2=VMTEM
     TDIST=TDIST+TVEL2*CØNF2*TINC
     TAD=(ØBSX-TDIST)/(.5*CØNF2*(TVEL2+VØLA))
     TND=(TVEL2-VØLA)/BA
     IF(TAD-TND)250,250,236
 236 TTIME=TTIME+TINC
     GØTØ 235
 250 VTEM(K,J,I)=ØBSX*CØNF1/(TTIME+TND)
 200 IF(RGU(K)+FØM+FAT1-VM*VTEM(K,J,I)**2/11.6
&    -FØRMX(K))260,300,300
 260 VTEM(K,J,I)=VTEM(K,J,I)*SRF
     IF (VTEM(K,J,I)-VELØ(K))290,290,270
 270 VELØ(K)=VTEM(K,J,I)
 290 CØNTINUE
 300 CØNTINUE
1000 CØNTINUE
     IF (VELØ(1)*VELØ(2)*VELØ(3))400,400,401
 400 VELØC=0.0
     RETURN
 401 VELØC=3.0*VELØ(1)*VELØ(2)*VELØ(3) /
&    (VELØ(2)*VELØ(3)+VELØ(1)*VELØ(3)+VELØ(1)*VELØ(2))
3172 FØRMAT (5F12.2)
3173 FØRMAT (//)
     RETURN
     END
```

Subroutine MARSH (Fig. C12)

Subroutine MARSH follows the general pattern of subroutine PATCH, except, since there are no obstacles in a marshy area, the subroutines associated with obstacles are not called. Checks have already been made in the main program to determine whether the water depth in the marsh is greater than the fording depth of the vehicle, and whether the vehicle is a swimmer and can swim across the area if the water is too deep to ford. When this subroutine is entered, it is already known that the vehicle will be fording. The only elements of this terrain that limit vehicle motion are soft soil (which will be considerably softer than was generally the case in PATCH) and vegetation. The first thing calculated are the various arrays associated with vegetation. These are: S(I), the mean spacing of all stems of stem diameter class I or larger; XNT(I), the number of trees of stem diameter class I in an area containing one tree of the largest size; and SDS(I), the mean spacing of stem diameter class I. Before the loop is entered, variable ADØ, the area denied by obstacles, is set to zero. A loop then is performed over the stem diameter classes +1.

The first subroutine entered is AREAV, and the area denied by vegetation is calculated and stored in variable PAV. PAV and ADØ are then sent to subroutine AREAT. In this subroutine, the total area denied is calculated, and the speed reduction factor, SRF, is returned. If SRF is zero, meaning that the vehicle is immobilized because it cannot maneuver, a return is made to the top of the loop. If not, subroutine VEGF is entered, and the forces associated with overriding vegetation are calculated. The variables are FAT1, the force required to knock over one tree; FMT, the maximum force needed to override trees; and FAT, the average force to override trees. Then a check is made to determine if FMT/GVW is greater than 2-g's, the horizontal acceleration limit the driver can stand. If it is larger than 2-g's, a return is made to the top of the loop, and no further calculations are performed.

A check is then made to see if the maximum force exceeds the pushbar force that the vehicle can withstand. If it does exceed this force, a return is made to the top of the loop; if the vehicle can stand the force, the calculations proceed. Variable TRFU is calculated; it contains the total resisting forces, in this case due to soil and vegetation. In PATCH, this variable also contains resistances due to slopes and obstacles, but neither of these features occur in a marsh. A check is then made to determine if the resisting forces are greather than the maximum force the vehicle can exert in forward motion. If they are, the vehicle is immobilized in traction, and a return is made to the top of the loop; if not, subroutine KURVE is entered, and a velocity (VMTEM) is returned as a starting velocity for subsequent calculations.

Next, a check is made to see if the total resisting forces exceed the sum of the forward forces that the vehicle can exert. These forward forces consist of the maximum force the vehicle can generate in soil, FØRMX, plus the forces associated with the forward kinetic energy, this being based on the velocity just determined. If the resisting forces do not exceed the forward forces, the calculation proceeds. It is next necessary to determine if the velocity, VMTEM, is greater than 2 mph. If it is, it is set back to 2 mph, because the limit due to recognition in an area heavily covered with vegetation is considered to be 2 mph. (When this point in the calculation is reached, it is already known that the area is covered with vegetation.) It is done here, and not in the PATCH subroutine, because in PATCH, it is not known at this point whether or not there is obscuring vegetation. This velocity, VMTEM, then is reduced by the speed reduction factor, SRF, calculated in subroutine AREAT. The velocity is loaded into variable VELØC, and a return to the main program is made.

SUBRØUTINE MARSH

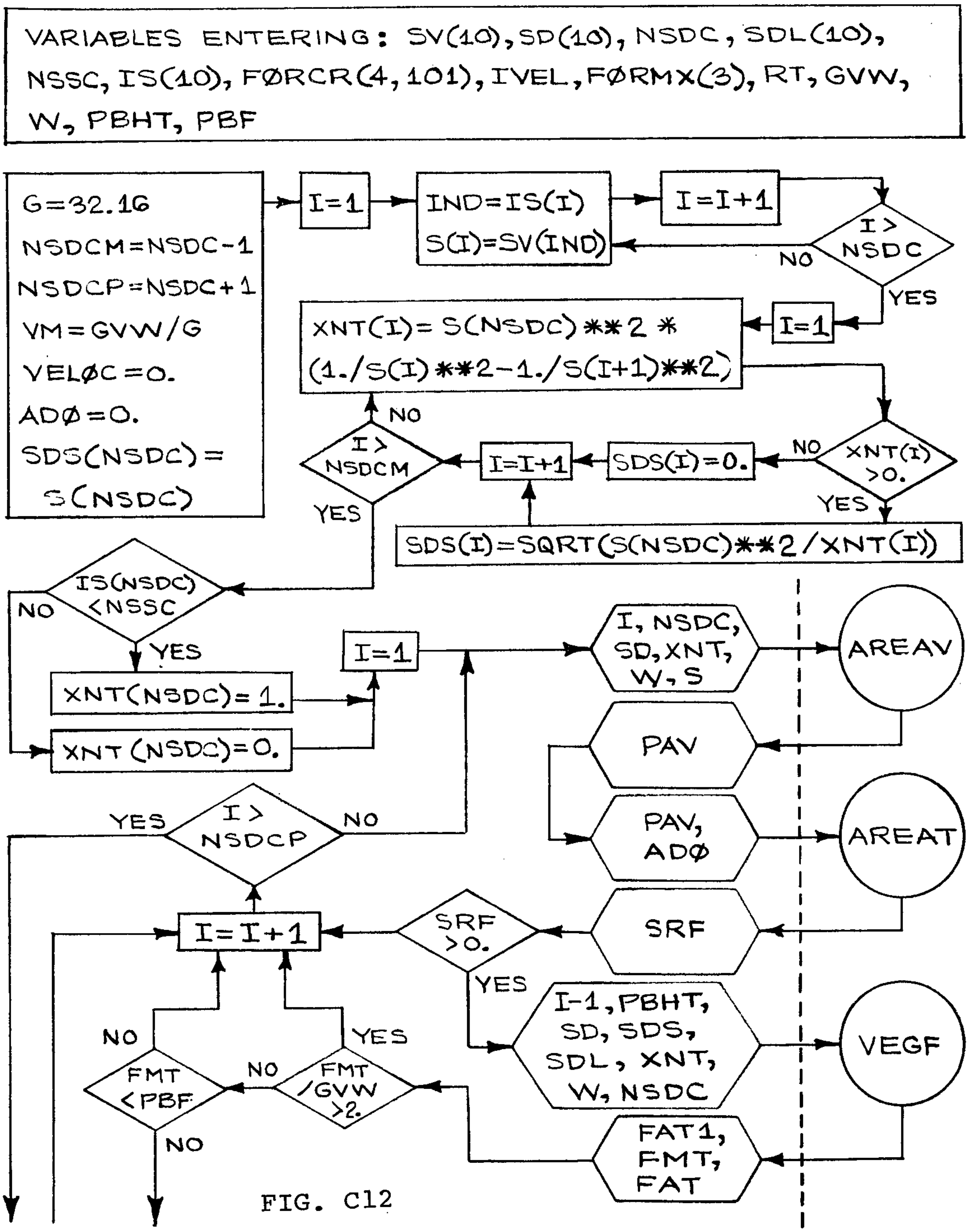


FIG. C12

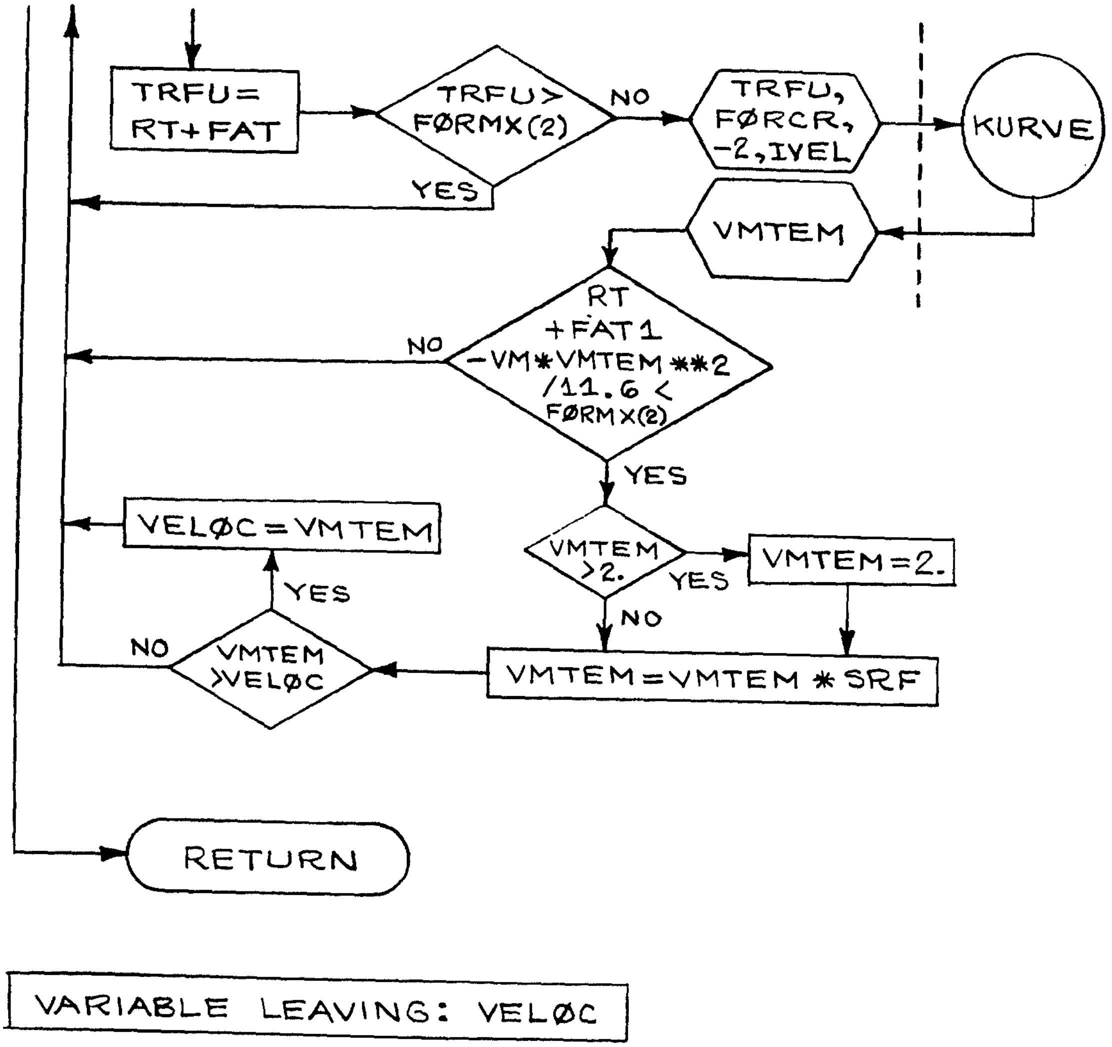


FIG. C12 cont'd

MARSH

```
      SUBRØUTINE MARSH(VELØC)
      CØMMØN SV(10),RD(10),SD(10),NSDC,SDL(10),NSSC,ØS(10),
&  AA(20),ØW(10),ØH(10),ØL(10),SLC(10),ISRC(20),IS(10),IREC,
&  IØBL,IØBW,IØBS,IØBH,IØBAA,IGR,IPR,IRCI(3),IST,IØST,SDS(10),
&  XNT(10),S(10),FØRCE(2,101),FØRCR(4,101),FØRMX(3),TFM(3),
&  RT,RCI,NVEH,NFL,GVW,DL,WID,GT,@,NBC,GC,HPT,ITV@R,RDIAM,
&  TPSI,TPLY,HS,WC,SAI,@WPKF,GCA,VSS,NCREW,FD,VFS,TNE1(2,30),
&  TTM(2,30),TTE(2,30),GR(10),NG,TC,RR,FDR,EFF,FDREF,ITRAN,
&  IVEL,NC1,NC2,NC3,ENTCG,LØKUP,VØØB(2,30),VRIDE(20),W,PBHT,
&  PBF,NC4,NC5
      G=32.16
      CØNF1=15./22.
      CØNF2=22./15.
      NSDCM=NSDC-1
      NSDCP=NSDC+1
      VM=GVW/G
      DØ 100 I=1,NSDC
      IND=IS(I)
      S(I)=SV(IND)
  100 CØNTINUE
      VELØC=0.0
      DØ 101 I=1,NSDCM
      XNT(I)=S(NSDC)**2*(1./S(I)**2-1./S(I+1)**2)
      IF (XNT(I))40,40,41
   40 SDS(I)=0.
      GØ TØ 101
   41 SDS(I)=SQRT(S(NSDC)**2/XNT(I))
  101 CØNTINUE
      IF (IS(NSDCYNSSC)50,51,51
   50 XNT(NSDC)=1.0
      SDS(NSDC)=S(NSDC)
      GØ TØ 52
   51XNT(NSDC)=0.
      SDS(NSDC)=S(NSDC)
   52 ADØ=0.0
  150 DØ 300 I=1,NSDCP
      CALL AREAV(I,SD,NSDC,XNT,W,S,PAV)
      CALL AREAT(PAV,ADØ,SRF)
      IF (SRF)155,155,160
  155 GØ TØ 300
  160 CALL VEGF(I-1,PBHT,SD,SDS,SDL,XNT,W,FAT1,FMT,FAT,NSDC)
      IF (FMT/GVW-2.0)165,165,155
  165 IF (FMT-PBF)170,155,155
  170 TRFU=RT+FAT
      IF (TRFU-FØRMX(2))171,171,300
  171 CALL KURVE(TRFU,VMTEM,FØRCR,-2,IVEL)
      IF(RT+FAT1-VM*VMTEM**2/11.6-FØRMX(2))260,300,300
  260 IF (VMTEM-2.0)265,261,261
  261 VMTEM=2.0
  265 VMTEM=VMTEM*SRF
```

MARSH CØNTINUED

```
      IF (VMTEM-VELØC)300,300,270
  270 VELØC=VMTEM
  300 CØNTINUE
      RETURN
      END
```

Subroutine HILL (Fig. C13)

Subroutine HILL calculates the grade resistance. Entering the subroutine are variables GRADE, the percent slope; GVW, the gross vehicle weight; and RT, the maximum soil resistance on level ground. First, variable ANGLE, which corresponds to percent slope, is calculated; next, RTS, the soil resistance corrected for slope angle; and finally, RGU, the slope resistance plus RTS, the slope-corrected soil resistance. Variables RGU and RTS are sent back to the calling program.

SUBRØUTINE HILL

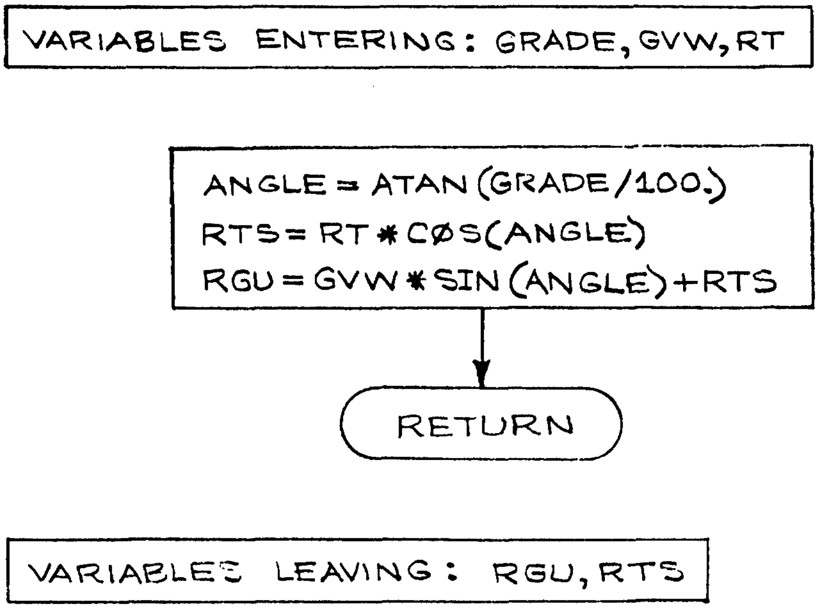


FIG. C13

HILL

```
      SUBRØUTINE HILL(GRADE,RGU,GVW,RT,RTS)
      ANGLE=ATAN(GRADE/100.)
      RTS=RT*CØS(ANGLE)
      RGU=GVW*SIN(ANGLE)+RTS
      RETURN
      END
```

Subroutine VISIØN (Fig. C14)

Subroutine VISIØN calculates the maximum velocity over a given patch type, limited by the recognition distance. It is assumed that the vehicle will be driven at a safe speed, i.e., it can be brought to a stop within the clear-view area ahead. The driver's reaction time is assumed to be 0.5 sec.

The safe distance is the sum of the distance traveled in 0.5 seconds, which is 0.5 x VELV and the distance traveled during the deceleration from VELV to zero speed, which is $(VELV)^2/2a$. Here "a" is the deceleration or the ratio of the braking force and the vehicle's mass, TFM/VM. Thus:

$$DR = 0.5 * VELV + (VELV)^2/(2 \times TFM/VM)$$

The solution of this equation yields:

```
VELV = [SQRT(0.25 * (TFM/VM) ** 2 + 2.0 * DR *
        TFM/VM) - 0.5 * TFM/VM]  * 15./22.
```

where:

VELV = the maximum safe velocity, mph

DR = the recognition or stopping distance, ft

TFM = the maximum braking force, lb (this force is assumed to be the maximum force the vehicle could impart to the soil if enough power were available)

VM = the mass of the vehicle, slugs

15./22 = the conversion factor from ft/sec to mph

If this calculated velocity is less than 2.0 mph, it is set to 2.0 mph, since this is considered the lowest safe speed (based on U.S. Army experience).

SUBRØUTINE VISIØN

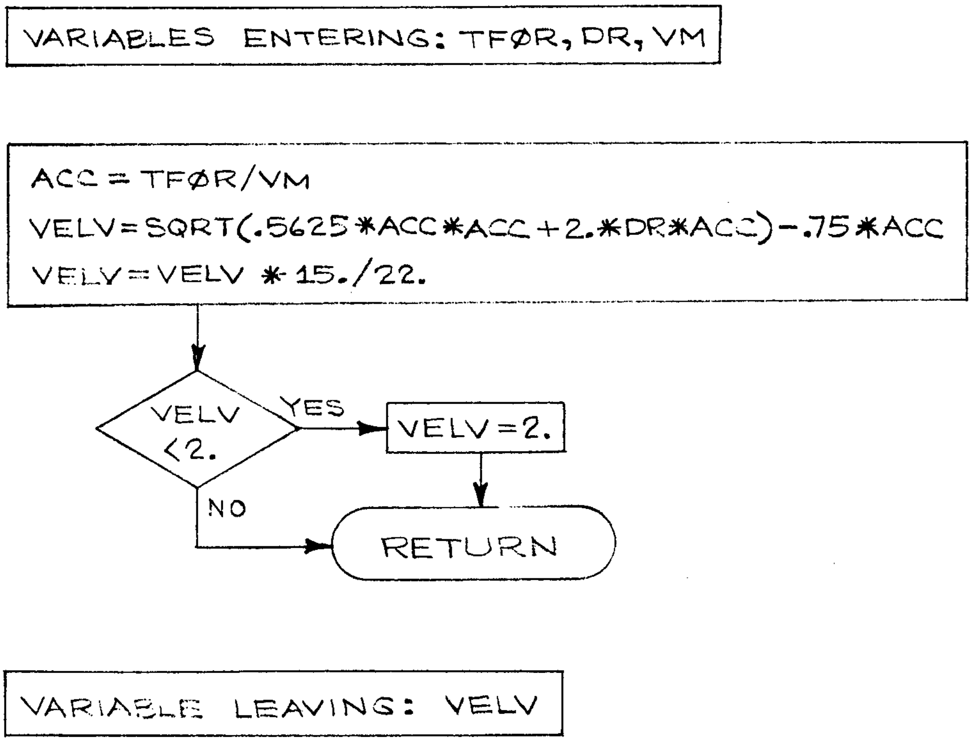


FIG. C14

VISIØN

```
      SUBRØUTINE VISIØN(TFØR,DR,VM,VELV)
      ACC=TFØR/VM
      ARG=.5625*ACC*ACC+2.*DR*ACC
      IF(ARG.LT.0.)GØTØ 1
      VELV=SQRT(ARG)-.75*ACC
      VELV=VELV*15./22.
      IF (VELV. LE. 2.0) VELV=2.0
      RETURN
    1 VELV=2.
      RETURN
      END
```

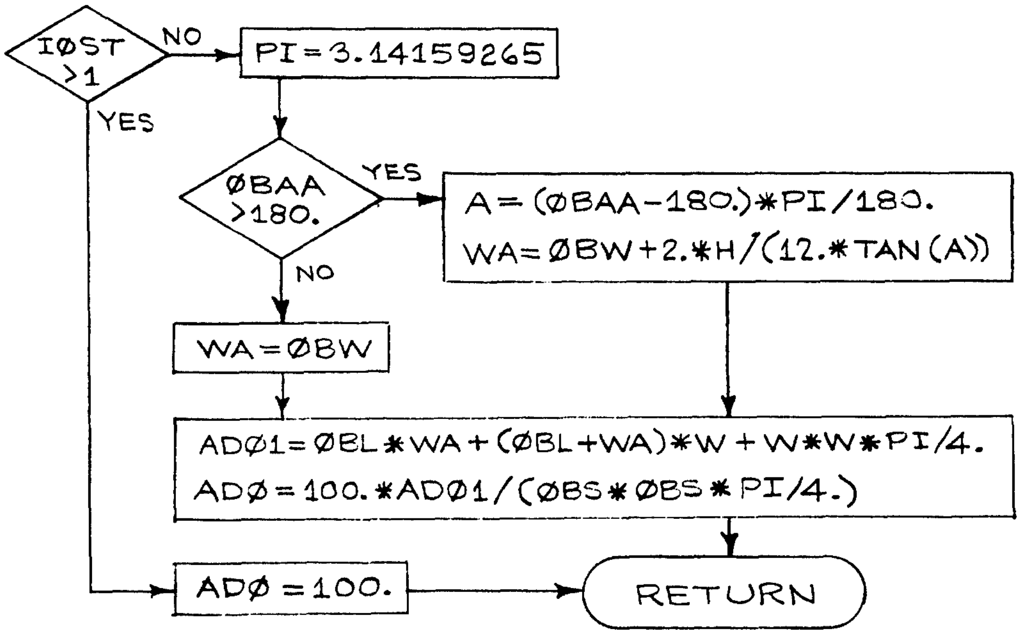
Subroutine AREAØ (Fig. C15)

Subroutine AREAØ calculates the percentage of area denied by obstacles, assuming that they must be avoided. (The avoid-override decision is made in PATCH.) First, the width of the obstacle at ground level is calculated. If the obstacle is a trench, the top is the width; if the obstacle is a mound, the bottom is the width. Next, the area denied by one obstacle is calculated; this is the sum of the obstacle length, times the obstacle width, plus an area created by laying off one-half the vehicle width all around the obstacle. The percentage of the area denied by obstacles is calculated as the area denied by one obstacle, divided by the area of a circle whose diameter is the mean obstacle spacing.

At the beginning of the program, a check was made to see if the obstacle's spacing type was linear. If it was, all obstacles are parallel, are of indefinite length, and lie across the path of vehicle travel. In this case, the percentage of the area denied by obstacles is set to 100 percent, which means that the obstacles cannot be avoided and must be crossed. In either case, variable ADØ contains the percentage and is returned to the calling program.

SUBRØUTINE AREAØ

VARIABLES ENTERING: ØBL, ØBW, ØBS, W, IØST, ØBAA, H



VARIABLE LEAVING: ADØ

FIG. C15

AREAØ

```
      SUBRØUTINE AREAØ(ØBL,ØBW,ØBS,W,ADØ,IØST,ØBAA,H)
      IF(IØST-1)1,1,2
    1 PI=3.14159265
      IF(ØBAA.GT.180.)GØTØ 3
      WA=ØBW
      GØTØ 4
    3 A=(ØBAA-180.)*PI/180.
      WA=ØBW+2.*H*CØS(A)/(12.*SIN(A))
    4 ADØ1=ØBL*WA+(ØBL+WA)*W+W*W*PI/4.
      ADØ=ADØ1/(ØBS*ØBS*PI/4.)*100.
      RETURN
    2 ADØ=100.
      RETURN
      END
```

Subroutine ØBSTCL (Fig. C50)

Subroutine ØBSTCL makes geometric and traction checks to see if the vehicle is immobilized in crossing an obstacle. The equations defining various geometric interferences and traction problems are derived in the following mathematical analysis. In the Figs. C16 and C17, the vehicle and obstacle data used in the analysis are defined by drawings. Next, the vehicle angle with respect to the level is calculated for the three possible configurations of the vehicle on the obstacle Fig. C18; and then, several critical distances are calculated. (Figs. C19-C26) These define the relation between certain dimensions on the vehicle and corresponding dimensions on the obstacle. Next, the geometric interferences possible on a trench are defined in order (Figs. C27-C37), followed by the definitions of possible geometric interferences on a mound. (Figs. C38-C45) The next part of the analysis derives the value of μ, the coefficient of friction used in the traction analysis, and the several cases of the immobilizations caused by lack of traction are derived mathetmatically.

In the program itself, the various critical distances are calculated first; then, all of the interferences in a trench are checked simultaneously. Next, all the interferences on a mound are checked simultaneously; and finally, all of the traction problems are checked. The only variable leaving this subroutine is IGØ. IGØ is zero if any geometric or traction check indicates interference or lack of available traction; if all checks are passed, IGØ leaves the subroutine as 1.

VEHICLE DATA

WHEELED

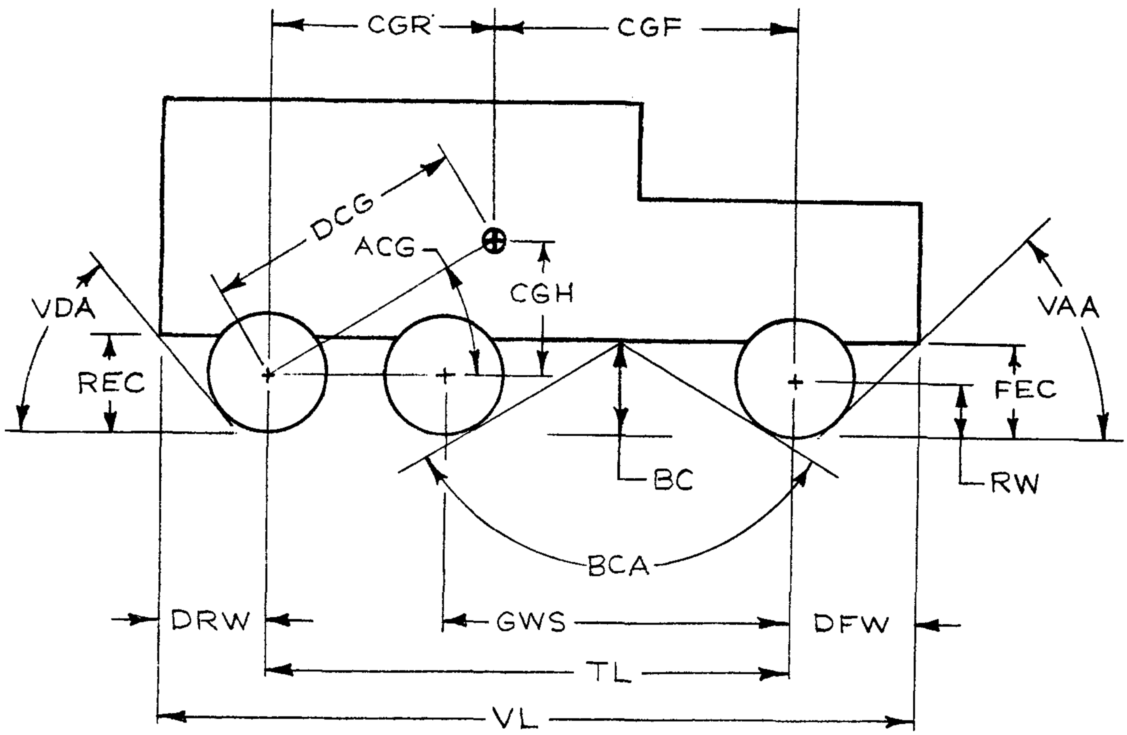


FIG. C16

TRACKED

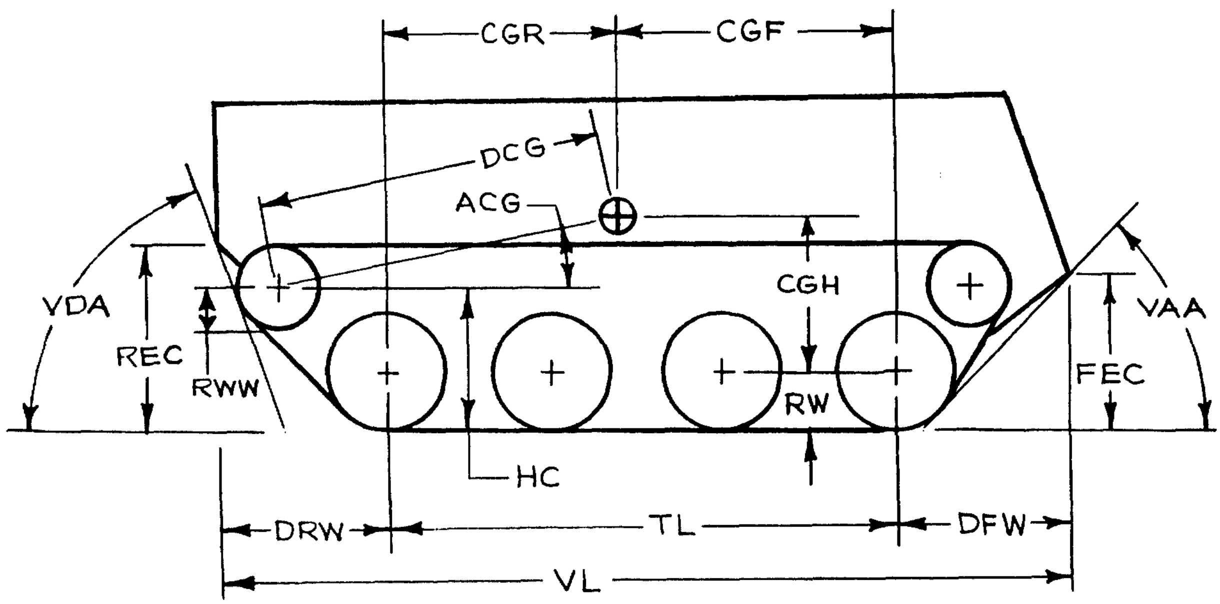
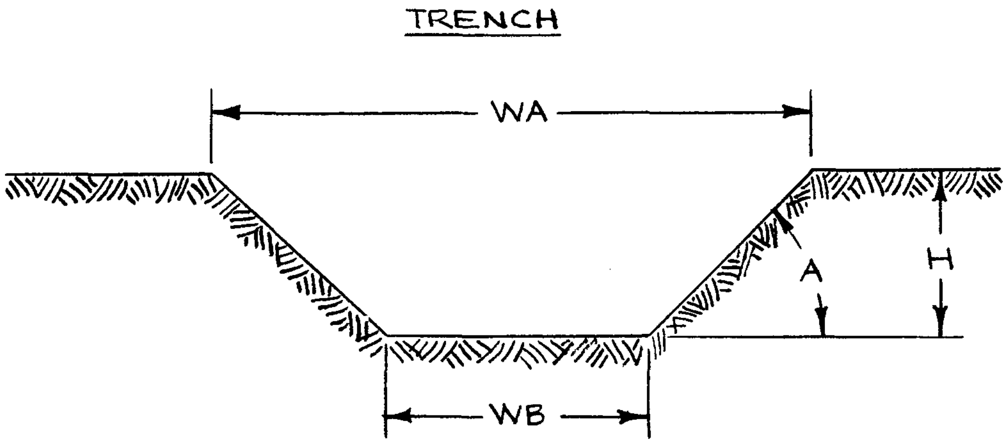


FIG. C16 cont'd

OBSTACLE DATA



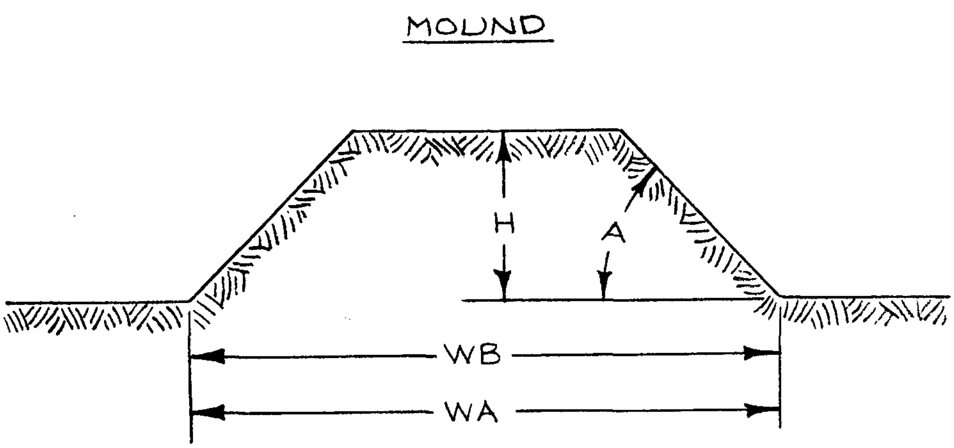
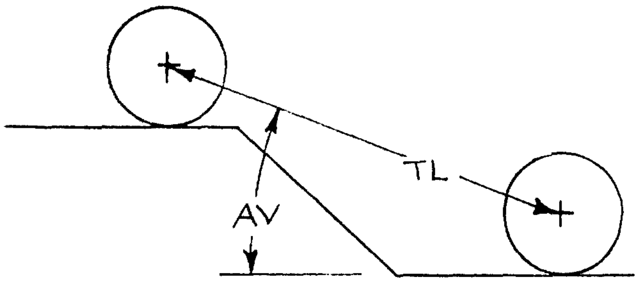
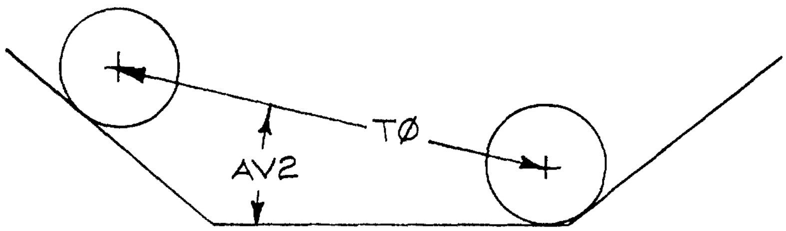


FIG. C17

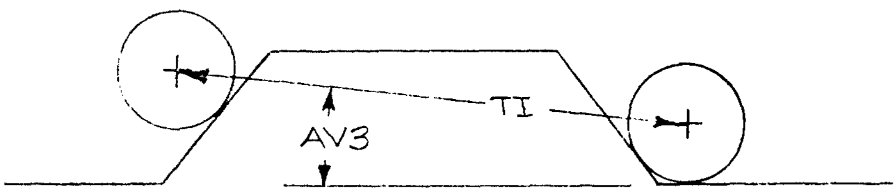
VEHICLE ANGLE



$$AV = \sin^{-1}\left(\frac{H}{TL}\right)$$



$$AV2 = A - \sin^{-1}\left[\frac{WB \sin A - 2 \cdot RW(1 - \cos A)}{TØ}\right]$$



$$AV3 = \sin^{-1}\left[\frac{WB \cdot \sin A + 2 \cdot RW(1 - \cos A)}{TI}\right] - A$$

FIG. C18

CRITICAL DISTANCES

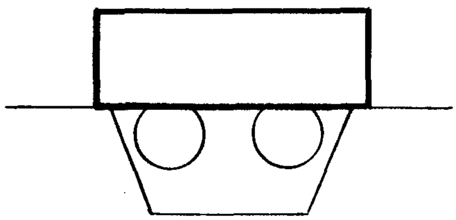


FIG. C19

$$X1 = VL - WB - \frac{2H}{TAN\,A} = 0$$

X1 is positive when the vehicle length is greater than the top width of a trench (negative otherwise).

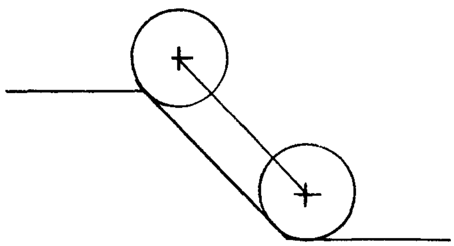


FIG. C20

$$X2 = TI + RW \cdot TAN\,\frac{A}{2} - \frac{H}{SIN\,A} = 0$$

X2 is negative or zero when the vehicle is fully supported on the flank of the obstacle (positive otherwise).

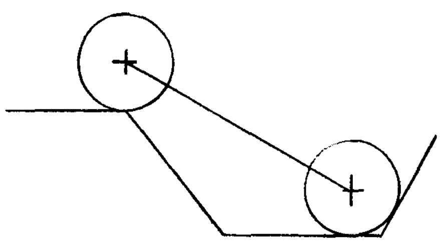


FIG. C21

$$X3 = \sqrt{TØ^2 - H^2} - WB + RW \cdot TAN\,\frac{A}{2} - \frac{H}{TAN\,A} = 0$$

X3 is positive when one wheel has not yet entered a trench and the other wheel is in contact with the bottom and the opposite flank (negative otherwise).

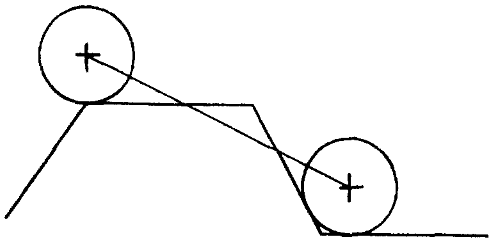


FIG. C22

$$X4 = \sqrt{TI^2 - H^2} - WB - RW \cdot \tan\frac{A}{2} + \frac{H}{\tan A} = 0$$

X4 is positive when one wheel has not yet reached the top of a mound and the other wheel is in contact with the bottom and the opposite flank (negative otherwise).

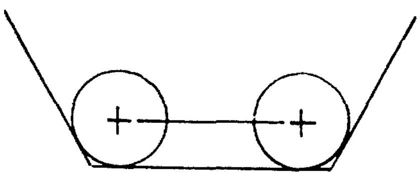


FIG. C23

$$X5 = T\emptyset + 2 \cdot RW \cdot \tan\frac{A}{2} - WB = 0$$

X5 is negative or zero when the vehicle is fully supported on the bottom of a trench (positive otherwise).

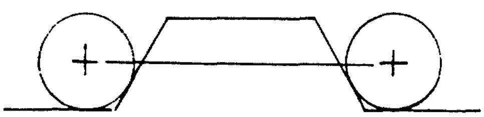


FIG. C24

$$X6 = TI - 2 \cdot RW \cdot \tan\frac{A}{2} - WB = 0$$

X6 is positive or zero when the vehicle is fully supported on the ground on both sides of a mound (negative otherwise).

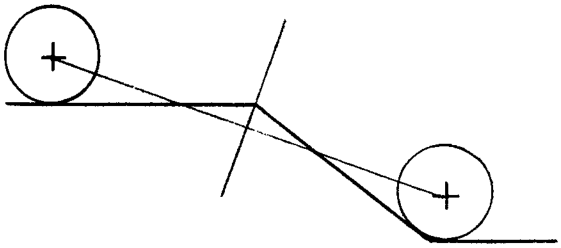


FIG. C25

$$X7 = TI \cdot \sin\frac{A}{2} - H = 0$$

X7 is negative when one wheel is on top of the obstacle (mound or trench), the other wheel is on the flank but not yet in contact with the bottom, and a line perpendicular to the wheelbase exactly divides in half the top corner angle of the obstacle (positive otherwise).

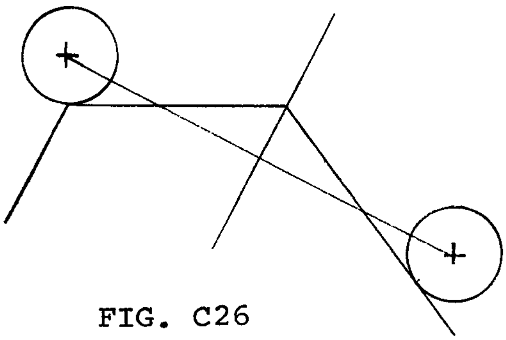


FIG. C26

$$X8 = WB - \frac{2H}{\tan A} - \frac{RW}{\tan\frac{A}{2}} - \frac{1}{\cos\frac{A}{2}}\left[\frac{TI}{2} - \frac{RW}{\sin\frac{A}{2}}\right]$$

X8 is positive when one wheel is on top of a mound (off the flank), the other wheel is on the opposite flank, and a line perpendicular to the wheelbase exactly divides in half the top corner angle of the obstacle (negative otherwise).

GEOMETRIC INTERFERENCES IN A TRENCH

In each case, all inequalities must be satisfied for an interference to occur.

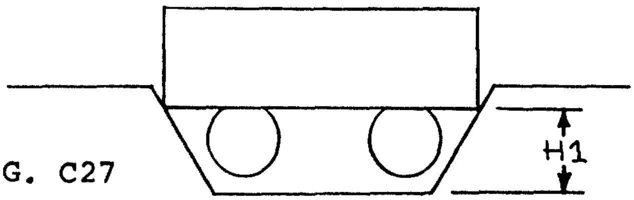


FIG. C27

$$\begin{cases} X1 < 0 \\ EA < A \\ H1 > EC \end{cases}$$

$$H1 = \frac{1}{2}(VL - WB)\tan A$$

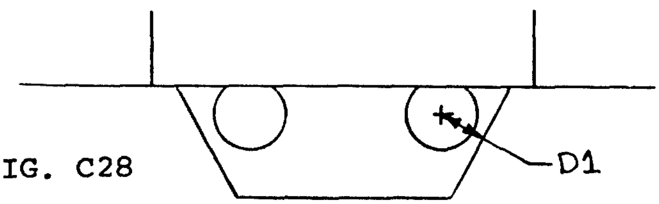


FIG. C28

$$\begin{cases} X1 \geq 0 \\ H > EC \\ D1 > RW \end{cases}$$

$$D1 = \frac{1}{2}(WB - T\emptyset)\sin A + (H - EC + RW)\cos A$$

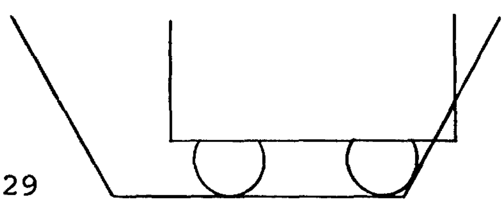


FIG. C29

$$\begin{cases} X5 \leq 0 \\ H > EC \\ EA < A \end{cases}$$

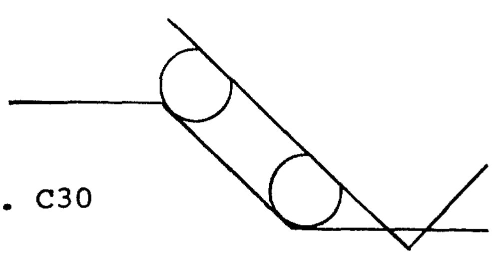


FIG. C30

$$\begin{cases} X2 \leq 0 \\ EA < A \end{cases}$$

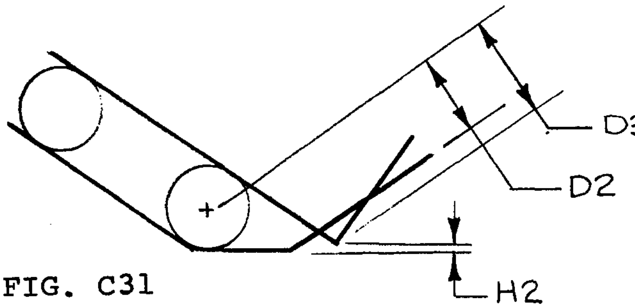


FIG. C31

$$\begin{cases} X2 \le 0 \\ H2 < H \\ EA \ge A \\ D3 > D2 \end{cases}$$

$$D2 = WB \sin A + RW\,(2\cos A - 1)$$
$$D3 = ED \sin(2A) - (EC - RW)\cos(2A)$$
$$H2 = EC \cos A + RW\,(1 - \cos A) - ED \sin A$$

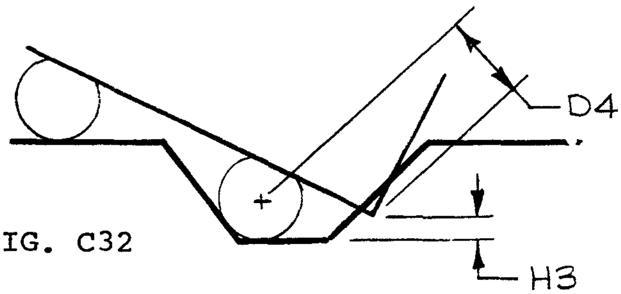


FIG. C32

$$\begin{cases} X2 > 0 \\ H3 < H \\ EA \ge AV \\ D4 > D2 \end{cases}$$

$$D4 = ED \sin(A + AV) - (EC - RW)\cos(A + AV)$$
$$H3 = EC \cos AV + RW\,(1 - \cos AV) - ED \sin AV$$

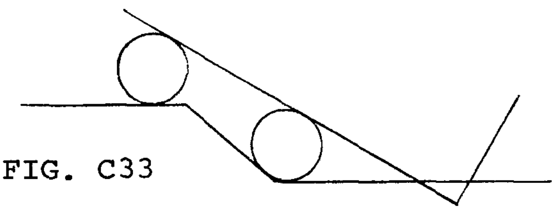


FIG. C33

$$\begin{cases} X2 > 0 \\ EA < AV \end{cases}$$

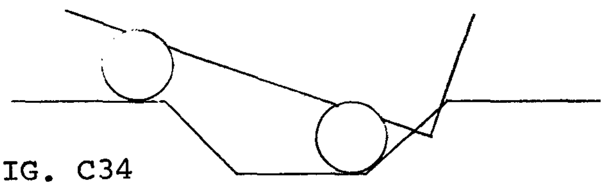
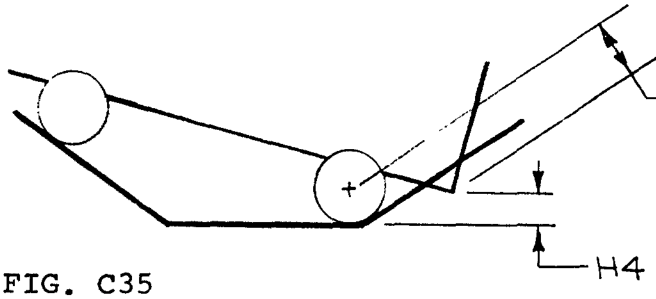


FIG. C34

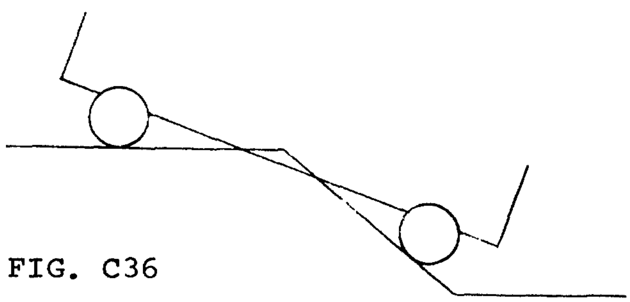
$$\begin{cases} X3 \ge 0 \\ H3 < H \\ D4 > RW \end{cases}$$



$$\begin{cases} X3 < 0 \\ X5 > 0 \\ H4 < H \\ D5 > RW \end{cases}$$

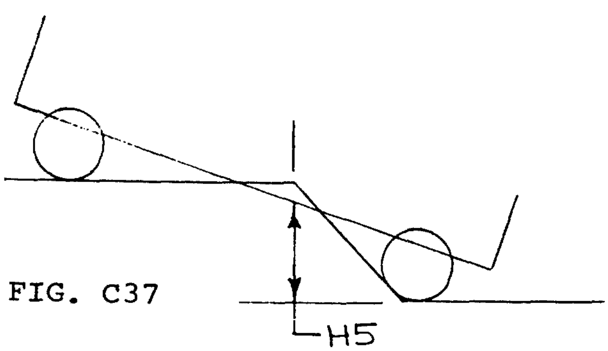
FIG. C35

$$D5 = ED \sin(A+AV2) - (EC-RW)\cos(A+AV2)$$
$$H4 = EC \cos AV2 + RW(1-\cos AV2) - ED \sin AV2$$



$$\begin{cases} BCA > \pi - A \\ X7 \leq 0 \end{cases}$$

FIG. C36



$$\begin{cases} X7 > 0 \\ H > H5 \end{cases}$$

FIG. C37

$$H5 = \left[BC - RW + \left(H \tan AV + \frac{RW}{\tan AV} + \frac{H}{\tan A} + RW \tan\frac{A}{2}\right)\sin AV\right]\cos AV$$

GEOMETRIC INTERFERENCES ON A MOUND

In each case, all inequalities must be satisfied for an interference to occur.

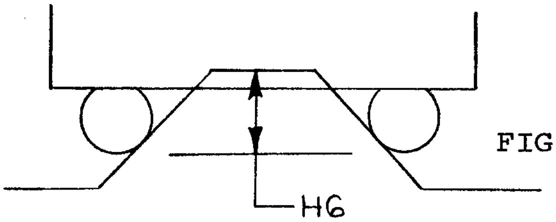


FIG. C38

$$\begin{cases} X6 < 0 \\ H6 > BC \end{cases}$$

$$H6 = H - \left[RW(1-\cos A) - \frac{1}{2}(TI - WB)\sin A \right] \frac{1}{\cos A}$$

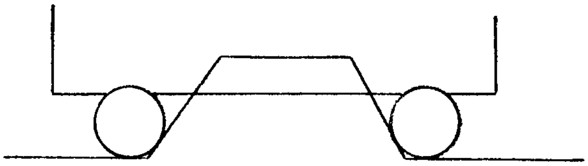


FIG. C39

$$\begin{cases} X6 \geq 0 \\ H > BC \end{cases}$$

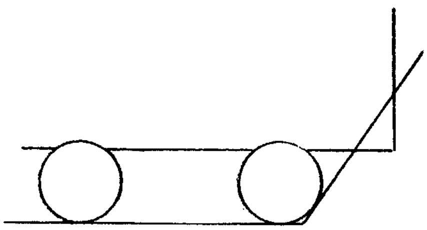


FIG. C40

$$\begin{cases} FEC < H \\ VAA < A \end{cases}$$

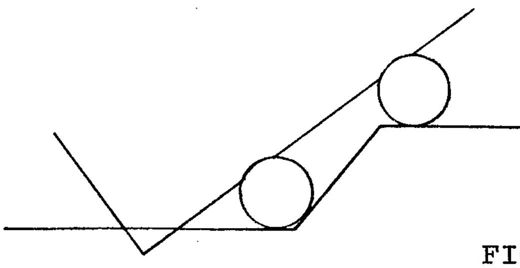
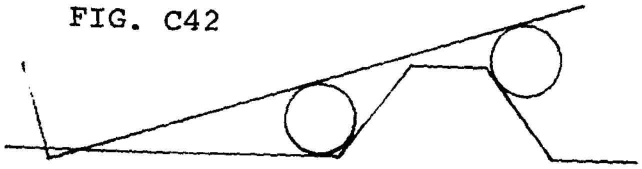


FIG. C41

$$\begin{cases} X2 > 0 \\ X4 \leq 0 \\ EA < AV \end{cases}$$

FIG. C42



$$\begin{cases} X4 > 0 \\ X6 < 0 \\ EA < AV3 \end{cases}$$

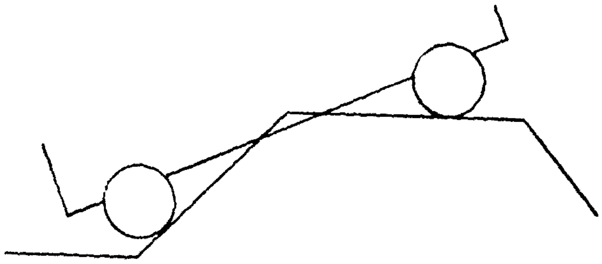


FIG. C43

$$\begin{cases} BCA < \pi - A \\ X7 \leq 0 \\ X8 \geq 0 \end{cases}$$

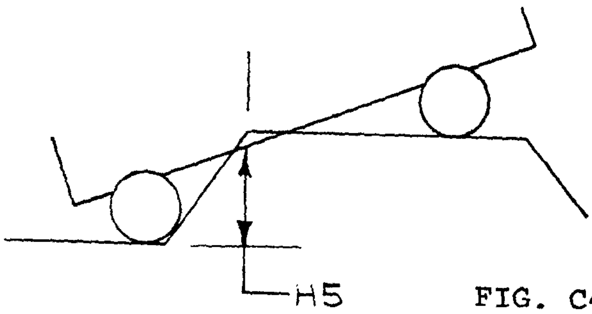
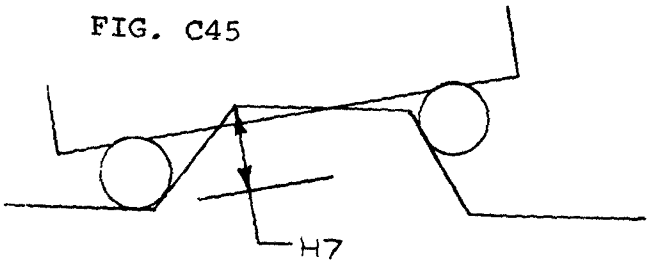


FIG. C44

$$\begin{cases} X7 > 0 \\ X4 \leq 0 \\ H > H5 \end{cases}$$

FIG. C45



$$\begin{cases} X4 > 0 \\ X6 < 0 \\ BC < H7 \end{cases}$$

$$H7 = \frac{H \sin(A - AV3)}{\sin A} - RW \tan\frac{A}{2} \sin AV3$$

CALCULATION OF THE COEFFICIENT OF FRICTION, μ, USED IN TRACTION

The coefficient of friction is assumed to be equal to the maximum force that the vehicle can generate divided by the vehicle weight. The force consists of the maximum traction that the vehicle can produce in the soil plus the force derived from the vehicle's kinetic energy.

$$\mu = \frac{FØRMX(2) + F_{KINETIC}}{GVW}$$

Kinetic energy is dependent on vehicle velocity, VV, which must be the least of:

VF -- Velocity that can be developed in the soil.

VØLA - Velocity over obstacles at 2.5g vertical acceleration.

Then kinetic energy,

$$E_{KINETIC} = \frac{1}{2} VM \cdot VV^2$$

where: $VM = \text{vehicle mass} = \frac{GVW}{G}$
$G = 32.16$
$VV = \text{Min}(VF, VØLA)$

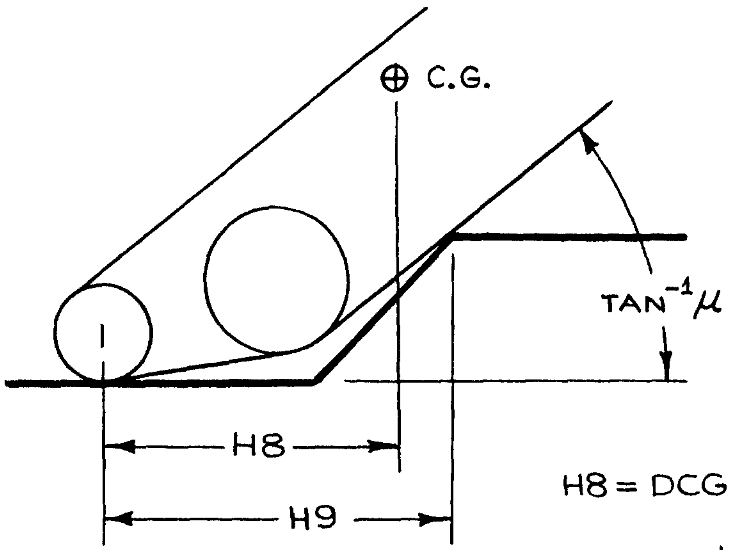
This energy is assumed to act over a distance equal to the vertical projection of the slope height of the obstacle.

$$D = \frac{H}{12. * \tan A} \qquad F_{KINETIC} = \frac{E_{KINETIC}}{D}$$

Finally then: $$\mu = \frac{FØRMX(2)}{GVW} + \frac{6 \cdot VV^2 \tan A}{G \cdot H}$$

IMMOBILIZATION CAUSED BY LACK OF TRACTION

Tracked vehicle on the approach flank of a mound. It is assumed that the largest angle that the vehicle can achieve is equal to $\tan^{-1}\mu$.



$$\begin{cases} X2 > 0 \\ A > \tan^{-1}\mu \\ H9 > H8 \end{cases}$$

$$H8 = DCG \cos\left(ACG + \tan^{-1}\mu\right)$$

$$H9 = \frac{HC}{\sin\left(\tan^{-1}\mu\right)} - \frac{RWW - H}{\mu}$$

FIG. C46

In each of the following cases, a free-body diagram and the equations derived from that diagram are shown. The equations are solved for A1, the obstacle flank-angle producing an equilibrium of forces. If this is less than the actual flank-angle, A, the vehicle is immobilized in traction.

Case 1: A wheeled or tracked vehicle entirely supported on the flank of the obstacle (mound or trench).

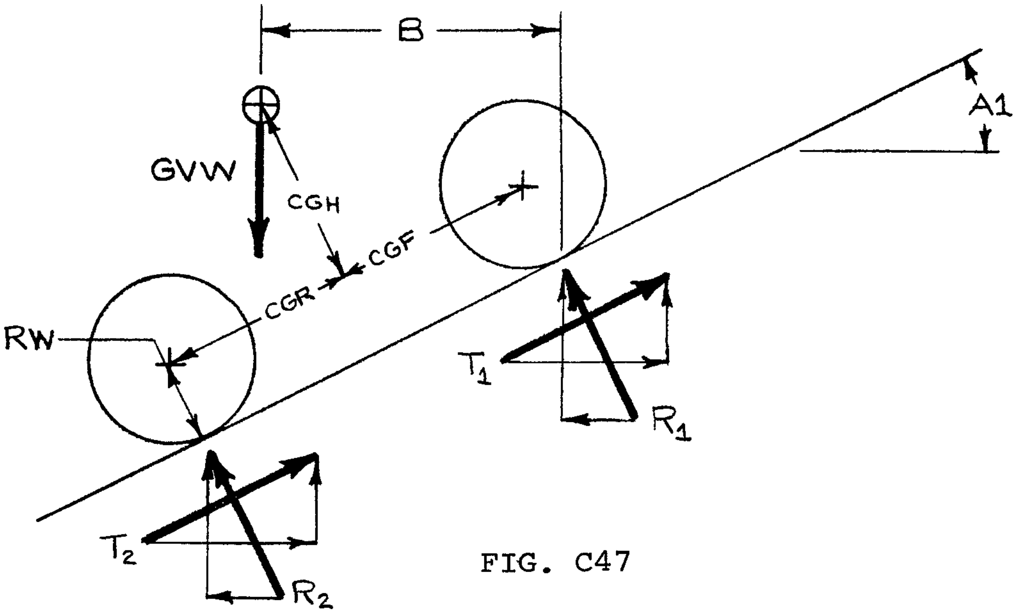


FIG. C47

$$\begin{cases} B = CGF\cos A1 + CGH\sin A1 + RW\sin A1 \\ T_1 = \mu R_1 \\ T_2 = \mu R_2 \\ T_2\cos A1 + T_1\cos A1 - R_1\sin A1 - R_2\sin A1 = 0 \\ T_2\sin A1 + T_1\sin A1 + R_1\cos A1 + R_2\cos A1 - GVW = 0 \\ B\cdot GVW - T\emptyset\cos A1\,(R_2\cos A1 + T_2\sin A1) \\ \qquad + T\emptyset\sin A1\,(T_2\cos A1 - R_2\sin A1) = 0 \\ T\emptyset = CGR + CGF \end{cases}$$

These equations resolve to: $A1 = \tan^{-1}\mu$

Immobilization occurs when: $\begin{cases} X2 \leq 0 \\ A1 < A \end{cases}$

Case 2: A wheeled vehicle with one wheel in contact with the bottom and flank of the obstacle (mound or trench), and the other wheel on the top.

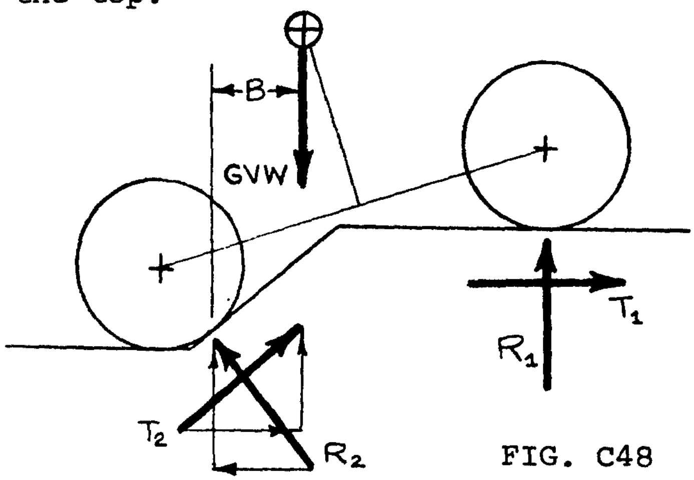


FIG. C48

$$
\begin{cases}
\sin AV = H/T\emptyset \qquad T\emptyset = CGF + CGR \\
B = CGR\cos AV - CGH\sin AV - RW\sin A1 \\
T_1 = \mu R_1 \qquad T_2 = \mu R_2 \\
T_1 + T_2\cos A1 - R_2\sin A1 = 0 \\
R_1 + R_2\cos A1 + T_2\sin A1 - GVW = 0 \\
B\cdot GVW + [H - RW(1-\cos A1)]T_1 - (T\emptyset\cos AV - RW\sin A1)R_1 = 0
\end{cases}
$$

These equations resolve to: $A1 = \sin^{-1}\frac{Z}{\sqrt{X^2+Y^2}} - \tan^{-1}\frac{Y}{X}$

WHERE: $X = AL(1+\mu^2) - Q$

$Y = \mu Q$

$Z = RW\mu^2$

$Q = T\emptyset\cos AV - (H-RW)\mu$

$AL = CGR\cos AV - CGH\sin AV$

Immobilization occurs when: $\begin{cases} x2 > 0 \\ x4 \leq 0 \\ A1 < A \end{cases}$

Case 3: A wheeled or tracked vehicle with one wheel on the top of a trench and the other wheel in contact with the bottom and the opposite flank.

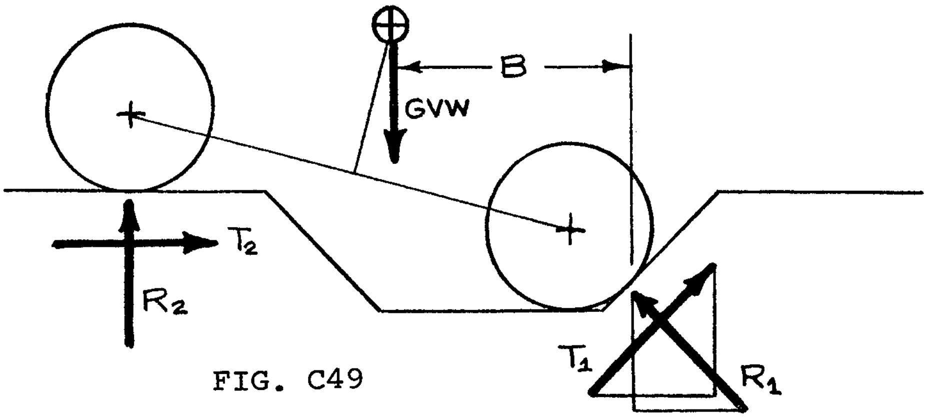


FIG. C49

$$
\begin{cases}
\sin AV = H/T\emptyset \qquad T\emptyset = CGF + CGR \\
B = CGF\cos AV - CGH\sin AV + RW\sin A1 \\
T_1 = \mu R_1 \qquad T_2 = \mu R_2 \\
T_2 + T_1\cos A1 - R_1\sin A1 = 0 \\
R_2 - GVW + T_1\sin A1 + R_1\cos A1 = 0 \\
B\cdot GVW - (T\emptyset\cos AV + RW\sin A1)R_2 - [H - RW(1-\cos A1)]T_2 = 0
\end{cases}
$$

These equations resolve to: $A1 = \sin^{-1}\dfrac{Z}{\sqrt{X^2+Y^2}} - \tan^{-1}\dfrac{Y}{X}$

WHERE:

$$X = AL(1+\mu^2) - Q$$
$$Y = \mu Q$$
$$Z = -RW\mu^2$$
$$Q = T\emptyset\cos AV + (H-RW)\mu$$
$$AL = CGF\cos AV - CGH\sin AV$$

Immobilization occurs when: $\begin{cases} X3 \geq 0 \\ A1 < A \end{cases}$

SUBRØUTINE ØBSTCL

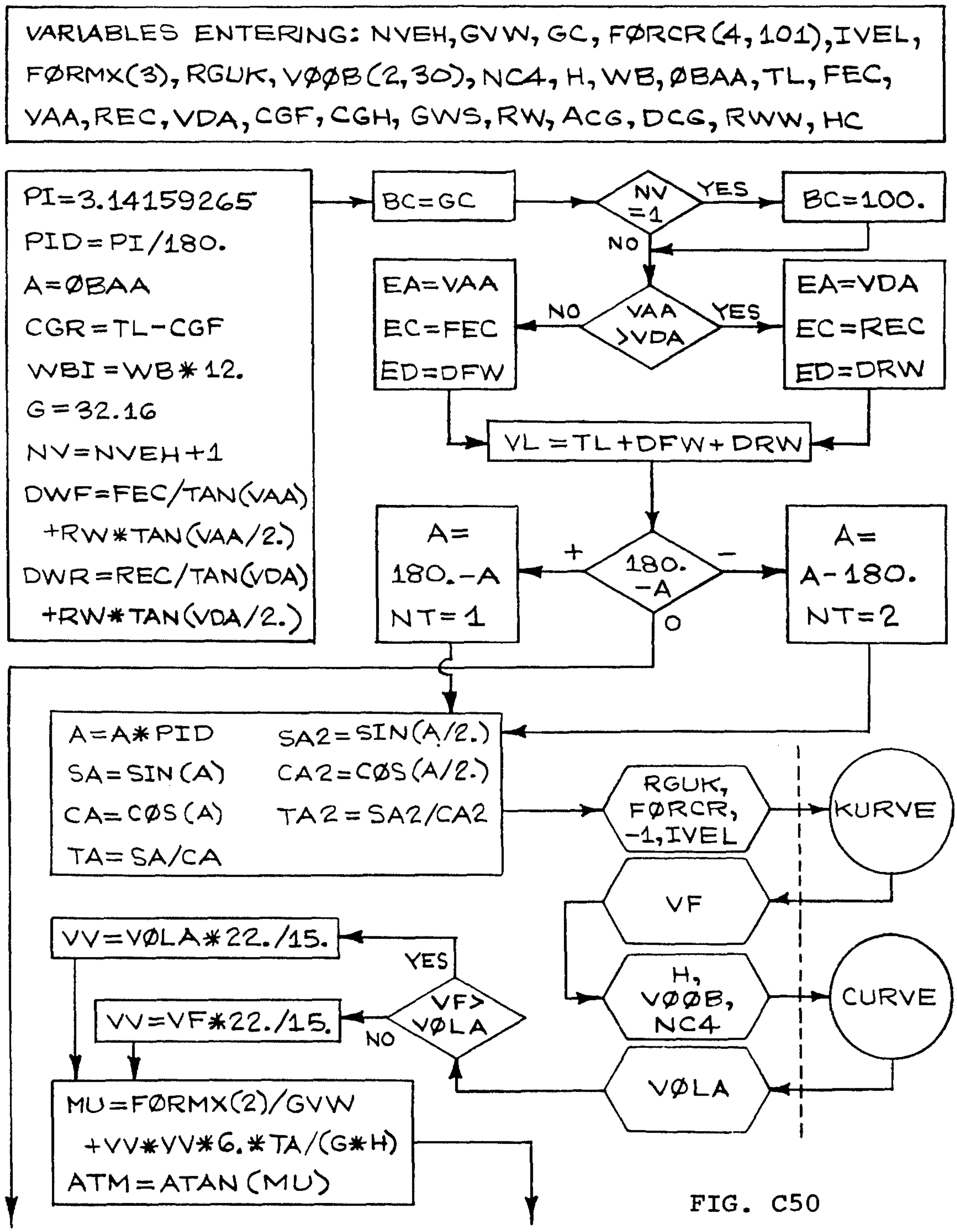


FIG. C50

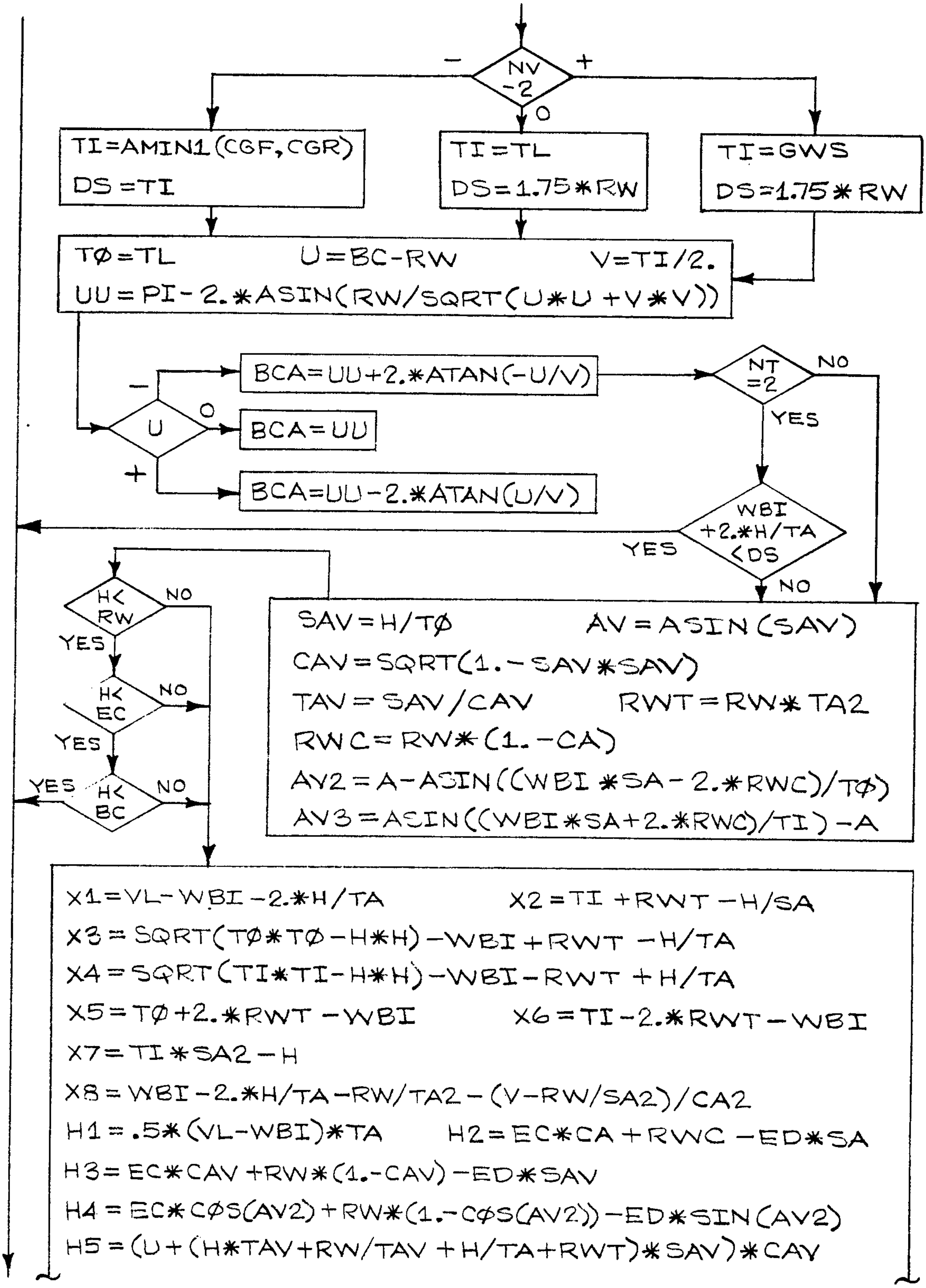


FIG. C50 cont'd

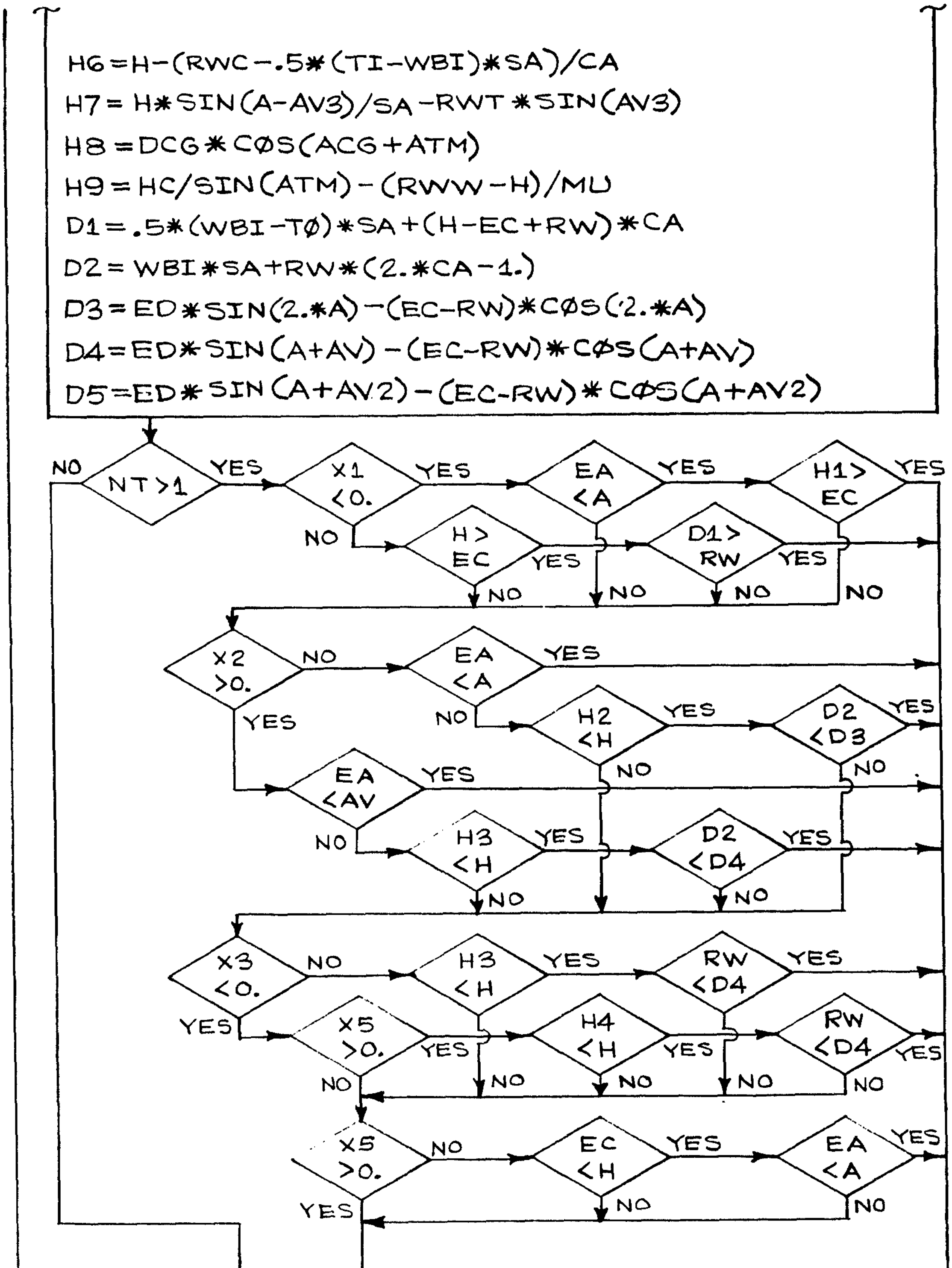


FIG. C50 cont'd

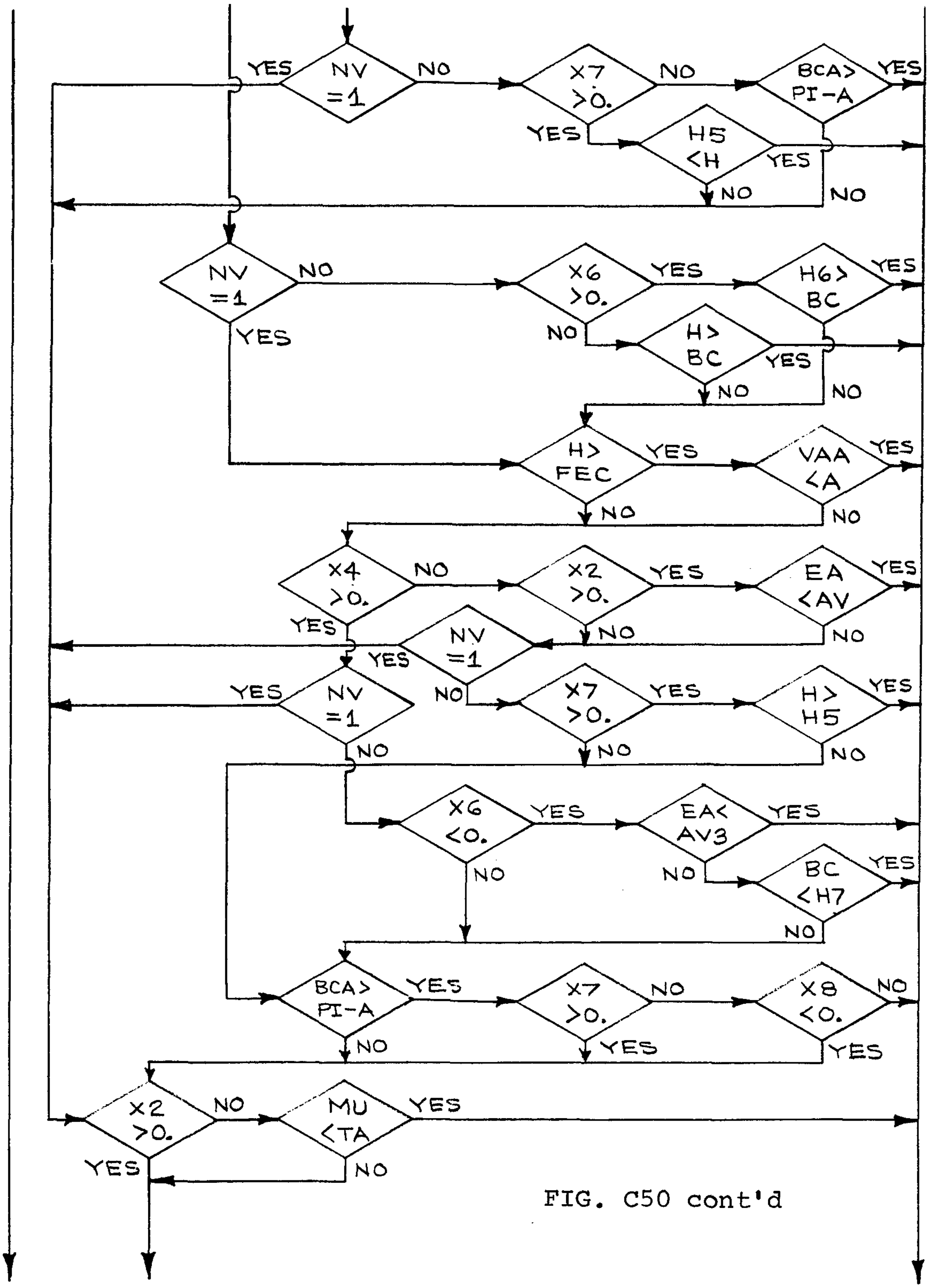


FIG. C50 cont'd

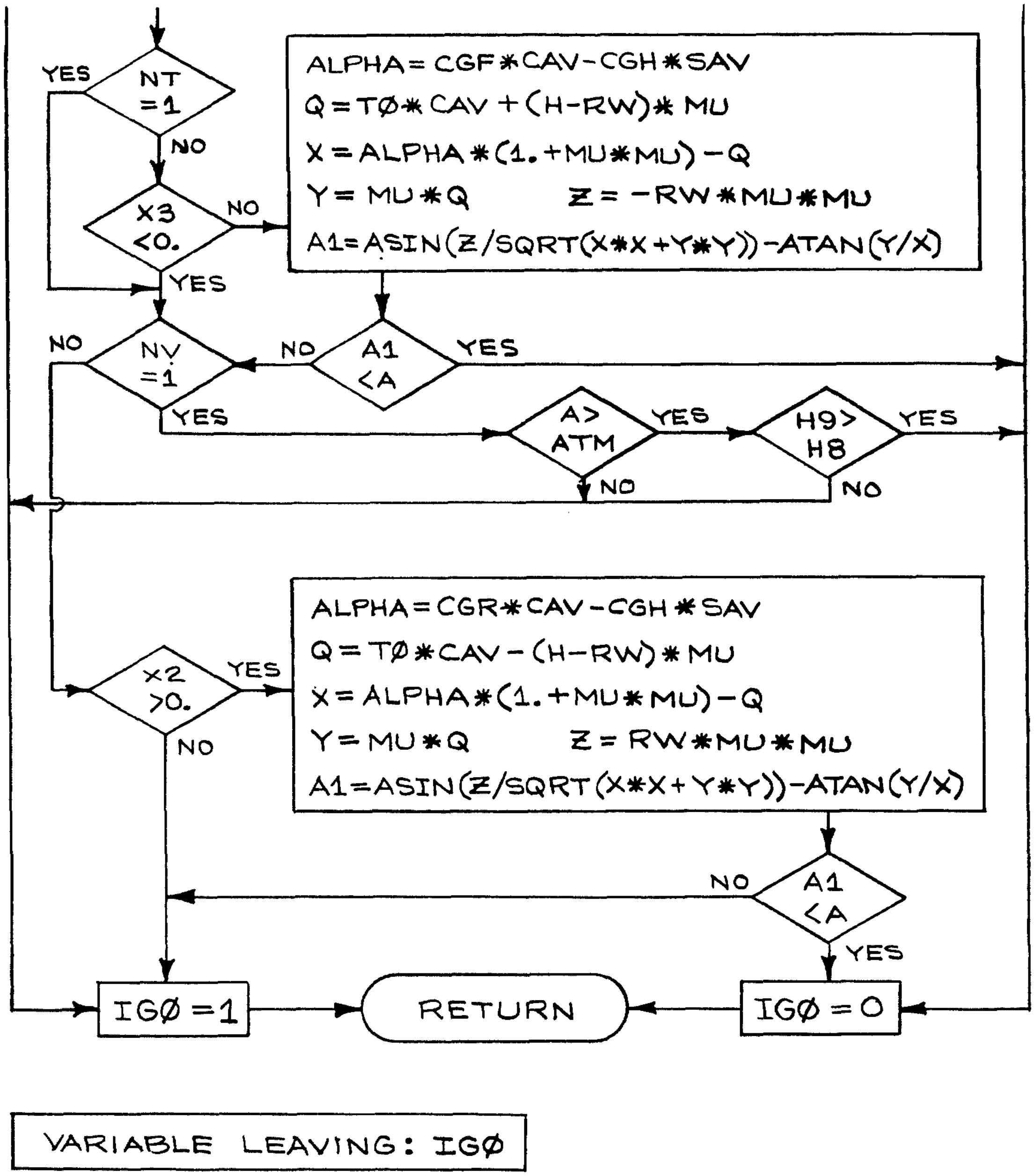


FIG. C50 cont'd

ØBSTCL

```
      SUBRØUTINE ØBSTCL(RGUK,IVEL,IGØ)
      REAL MU
      CØMMØN IPATCH(325),FØRCE(2,101),FØRCR(4,101),FØRMX(3),
& TFØR(3),RT,RCI,NVEH,NFL,GVW,DL,WID,GT,A0,NBC,GC,HPT,ITVAR,
& RDIAM,TPSI,TPLY,HS,WC,SAI,AWPKF,GCA,VSS,NCRDW,FD,VFS,
& ITRACT(399),VØØB(2,30),VRIDE(20),W,PBHT,PBF,VLL,NC4,NC5,
& H,WB,ØBAA,TL,FEC,VAA,REC,VDA,CGF,CGH,GWS,RW,ACG,DCG,HC,RWW
      PI=3.14159265
      PID=PI/180.
      A=ØBAA
      CGR=TL-CGF
      WBI=WB*12.
      G=32.16
      DFW=FEC*CØS(VAA)/SIN(VAA)+RW*SIN(VAA/2.)/CØS(VAA/2.)
      DRW=REC*CØS(VDA)/SIN(VDA)+RW*SIN(VDA/2.)/CØS(VDA/2.)
      NV=NVEH+1
      BC=GC
      IF(NV.EQ.1)BC=100.
      IF(VAA.LE.VDA)GØTØ 10
      EA=VDA
      EC=REC
      ED=DRW
      GØTØ 11
   10 EA=VAA
      EC=FEC
      ED=DFW
   11 VL=TL+DFW+DRW
      IF(180.-A)13,90,12
   12 A=180.-A
      NT=1
      GØTØ 14
   13 A=A-180.
      NT=2
   14 A=A*PID
      SA=SIN(A)
      CA=CØS(A)
      TA=SA/CA
      SA2=SIN(A/2.)
      CA2=CØS(A/2.)
      TA2=SA2/CA2
      CALL KURVE(RGUK,VF,FØRCR,-1,IVEL)
      CALL CURVE(H,VØLA,VØØB,NC4)
      IF(VF.GT.VØLA)GØTØ 1
      VV=VF*22./15.
      GØTØ 2
    1 VV=VØLA*22./15.
    2 MU=FØRMX(2)/GVW+6.*VV*VV*TA/(G*H)
      ATM=ATAN(MU)
      GØTØ(15,16,17),NV
   15 TI=AMIN1(CGF,CGR)
```

ØBSTCL CØNTINUED

```
      DS=TI
      GØTØ 18
   16 TI=TL
      DS=1.75*RW
      GØTØ 18
   17 TI=GWS
      DS=1.75*RW
   18 TØ=TL
      U=BC-RW
      V=TI/2.
      UU=PI-2.*ASIN(RW/SQRT(U*U+V*V))
      IF(U)41,42,43
   41 BCA=UU+2.*ATAN(-U/V)
      GØTØ 44
   42 BCA=UU
      GØTØ 44
   43 BCA=UU-2.*ATAN(U/V)
   44 CØNTINUE
      IF(NT.EQ.2.AND.WBI+2.*H/TA.LT.DS)GØTØ 90
      SAV=H/TØ
      AV=@SIN(SAV)
      CAV=SQRT(1.-SAV*SAV)
      TAV=SAV/CAV
      RWT=RW*TA2
      RWC=RW*(1.-CA)
      AV2=A-ASIN((WBI*SA-2.*RWC)/TØ)
      AV3=ASIN((2.*RWC+WBI*SA)/TI)-A
      IF(H.LT.BC.AND.H.LT.EC.AND.H.LT.RW)GØTØ 90
      X1=VL-WBI-2.*H/TA
      X2=TI+RWT-H/SA
      X3=SQRT(TØ*TØ-H*H)-WBI+RWT-H/TA
      X4=SQRT(TI*TI-H*H)-WBI-RWT+H/TA
      X5=TØ+2.*RWT-WBI
      X6=TI-2.*RWT-WBI
      X7=TI*SA2-H
      X8=WBI-2.*H/TA-RW/TA2-(TI/2.-RW/SA2)/CA2
      H1=.5*(VL-WBI)*TA
      H2=EC*CA+RWC-ED*SA
      H3=EC*CAV+RW*(1.-CAV)-ED*SAV
      H4=EC*CØS(AV2)+RW*(1.-CØS(AV2))-ED*SIN(AV2)
      H5=(BC-RW+(H*TAV+RW/TAV+H/TA+RWT(*SAV)*CAV
      H6=H-(RWC-.5*(TI-WBI)*SA)/CA
      H7=H*SIN(A-AV3)/SA-RWT*SIN(AV3)
      H8=DCG*CØS(ACG+ATM)
      H9=HC/ SIN(ATM)-(RWW-H)/MU
      D0=.5*(WBI-TØ)*SA+(H-EC+RW)*CA
      D2=WBI*SA+RW*(2.*CA-1.)
      D3=ED*SIN(2.*A)-(EC-RW)*CØS(2.*A)
      D4=ED*SIN(A+AV)-(EC-RW)*CØS(A+AV)
      D5=ED*SIN(A+AV2)-(EC-RW)*CØS(A+AV2)
```

ØBSTCL CØNTINUED

```
      GØTØ(50,20),NT
 20   IF(X1)21,22,22
 21   IF(EA.LT.A.AND.H1.GT.EC)GØTØ 91
      GØTØ 23
 22   IF(H.GT.EC.AND.D1.GT.RW)GØTØ 91
 23   IF(X2)24,24,25
 24   IF(EA.LT.A)GØTØ 91
      IF(H2.LT.H.AND.D2.LT.D3)GØTØ 91
      GØTØ 26
 25   IF(EA.LT.AV)GØTØ 91
      IF(H3.LT.H.AND.D2.LT.D4)GØTØ 91
 26   IF(X3)28,27,27
 27   IF(H3.LT.H.AND.RW.LT.D4)GØTØ 91
      GØTØ 29
 28   IF(X5.GT.0..AND.H4.LT.H.AND.RW.LT.D5)GØTØ 91
 29   IF(X5.LE.0..AND.EC.LT.H.AND.EA.LT.A)GØTØ 91
      IF(NV.EQ.1)GØTØ 70
      IF(X7)30,30,31
 30   IF(BCA.GT.PI-A)GØTØ 91
      GØTØ 70
 31   IF(H5.LT.H)GØTØ 91
      GØTØ 70
 50   IF(NV.EQ.1)GØTØ 53
      IF(X6)51,52,52
 51   IF(H6.GT.BC)GØTØ 91
      GØTØ 53
 52   IF(H.GT.BC)GØTØ 91
 53   IF(H.GT.FEC.AND.VAA.LT.A)GØTØ 91
      IF(X4)54,54,55
 54   IF(X2.GT.0..AND.EA.LT.AV)GØTØ 91
      IF(X7.GT.0..AND.H.GT.H5.AND.NV.NE.1)GØTØ 91
      GØTØ 56
 55   IF(X6.GE.0.)GØTØ 56
      IF(EA.LT.AV3.AND.NV.NE.1)GØTØ 91
      IF(BC.LT.H7.AND.NV.NE.1)GØTØ 91
 56   IF(BCA.GT.PI-A.@ND.X7.LE.0..AND.X8.GE.0..AND.NV.NE.1)GØTØ 91
 70   IF(X2.LE.0..AND.MU.LT.TA)GØTØ 91
      IF(NT.EQ.1.ØR.X3.LT.0.)GØTØ 71
      ALPHA=CGF*CAV-CGH*SAV
      Q=TØ*CAV+(H-RW)*MU
      X=ALPHA*(1.+MU*MU)-Q
      Y=MU*Q
      Z=-RW*MU*MU
      A1=ASIN(Z/SQRT(X*X+Y*Y))-ATAN(Y/X)
      IF(A1.LT.A)GØTØ 91
 71   IF(NV.EQ.1)GØTØ 72
      IF(X2.LE.0.)GØTØ 90
      GØTØ 75
 72   IF(A.GT.ATM.AND.H9.GT.H8)GØTØ 91
      GØTØ 90
```

ØBSTCL CØNTINUED

```
75 ALPHA=CGR*CAV-CGH*SAV
   Q=TØ*CAV-(H-RW)*MU
   X=ALPHA*(1.+MU*MU)-Q
   Y=MU*Q
   Z=RW*MU*MU
   A1=ASIN(Z/SQRT(X*X+Y*Y))-ATAN(Y/X)
   IF(A1.LT.A)GØTØ 91
90 IGØ=1
   RETURN
91 IGØ=0
   RETURN
   END
```

Subroutine ØBSF (Fig. C51)

Subroutine ØBSF calculates the average force needed to override a series of vertical obstacles. This force is added to the other resisting forces in determining vehicle speed as limited by power and traction. The force is obtained by dividing the work done to override one obstacle by the distance between encounters of that obstacle type.

First, IØST, which is equal to the obstacle spacing type, is checked. IØST = 1 for random spacing (obstacles arranged at random), and IØST = 2 for linear spacing (obstacles running parallel). If the spacing is random, the following equation is used:

FØM = GVW * HFT/ (PI * ØBS * ØBS)/(4 * W)

Where:

FØM = the average force to override obstacles, lb

GVW = the vehicle weight, lb

HFT = the vertical height of the obstacle, ft

ØBS = the mean spacing between obstacles, ft

W = the width of the vehicle, ft

PI = 3.14159265

If the spacing is linear:

FØM = GVW * HFT/ØBS

Note that:

Work = GVW * HFT

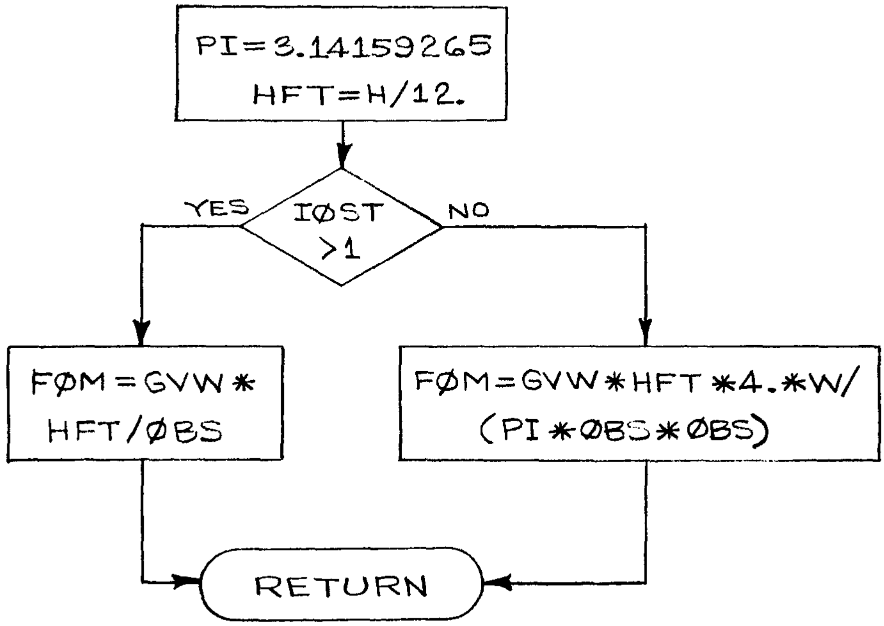
And:

Area for 1 obstacle = (PI * ØBS * ØBS)/4.

The effective distance between encounters of obstacles is equal to the area of a circle whose radius is the average obstacle spacing divided by the vehicle width. (This has been found from past field testing by the WES.) However, if the obstacle spacing type is linear rather than random, the distance between encounters is simply the obstacle spacing.

SUBRØUTINE ØBSF

VARIABLES ENTERING: GVW, H, ØBS, W, ØBL, IØST



VARIABLE LEAVING: FØM

FIG. C51

ØBSF

```
      SUBRØUTINE ØBSF(GVW,H,ØBS,W,ØBL,FØM,IØST)
      PI=3.14159265
      HFT=H/12.
      IF(IØST-1)1,1,2
    1 FØM=GVW*HFT/(PI*ØBS*ØBS)*4.*W
      RETURN
    2 FØM=GVW*HFT/ØBS
      RETURN
      END
```

Subroutine AREAV (Fig. C52)

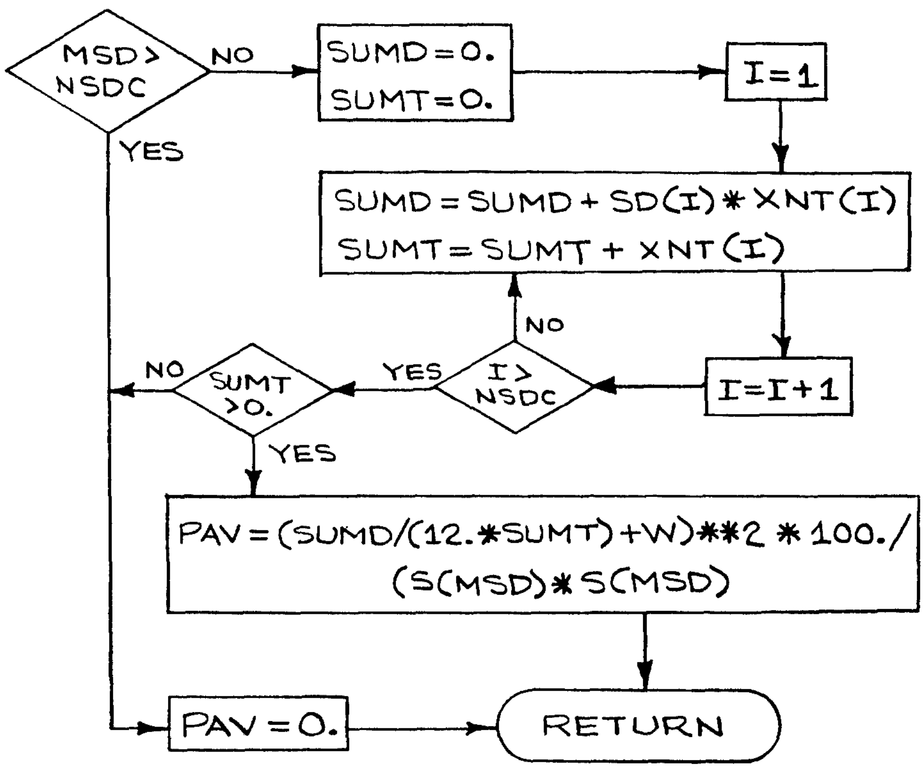
Subroutine AREAV calculates the percentage of the total area denied by trees. The first part calculates the average stem diameter to be avoided, variable SDA, by first accumulating two values: one, the total of the diameters of all stems in the area considered, variable SUMI; and the other, the total number of stems in the area considered, variable SUMT. Variable XNT, which contains the number of stems of each stem diameter in the area, is multiplied by the diameter of the stems, and the values are accumulated in variable SUMI. XNT is also accumulated in variable SUMT, from stem diameter MSD (one of the variables entering the subroutine) to the largest stem diameter in the area, to obtain the total number of stems (trees). Subroutine AREAV is called repeatedly, and MSD is indexed upward by one class each time it is called. The average stem diameter to be avoided is equal to SUMI, the total of the diameters, divided by SUMT, the total of the number of trees in the area.

Finally, the percentage of the area denied by vegetation is calculated as follows, based on this average stem diameter: Two areas are considered; the first is the area of a circle whose radius is the average stem diameter plus the vehicle width, and the second is the area of a circle whose radius is the mean spacing of all stems in diameter class MSD or larger in the area being considered. The first area divided by the second and multiplied by 100 yields the percentage of the total area denied by the trees; this is variable PAV.

If, at the end of the accumulation of stem sizes, it is found that there are no trees in the area (SUMT = 0), the percentage of the area denied, PAV, is set to zero. In either case, an exit is made from this subroutine.

SUBRØUTINE AREAV

VARIABLES ENTERING: MSD, SD(10), NSDC, XNT(10), W, S(10)



VARIABLE LEAVING: PAV

FIG. C52

AREAV

```
      SUBRØUTINE AREAV(MSD,SD,NSDC,XNT,W,S,PAV)
      DIMENSIØN SD(10),SDS(00),XNT(10),S(10)
      IF(MSD.GT.NSDC)GØTØ 2
      SUMD=0.
      SUMT=0.
      DØ 1 I=MSD,NSDC
      SUMD=SUMD+SD(I)*XNT(I)
    1 SUMT=SUMT+XNT(I)
      IF(SUMT.EQ.0.)GØTØ 2
      PAV=(SUMD/(12.*SUMT)+W)**2*100./(S(MSD)*S(MSD))
      RETURN
    2 PAV=0.
      RETURN
      END
```

Subroutine VEGF (Fig. C53)

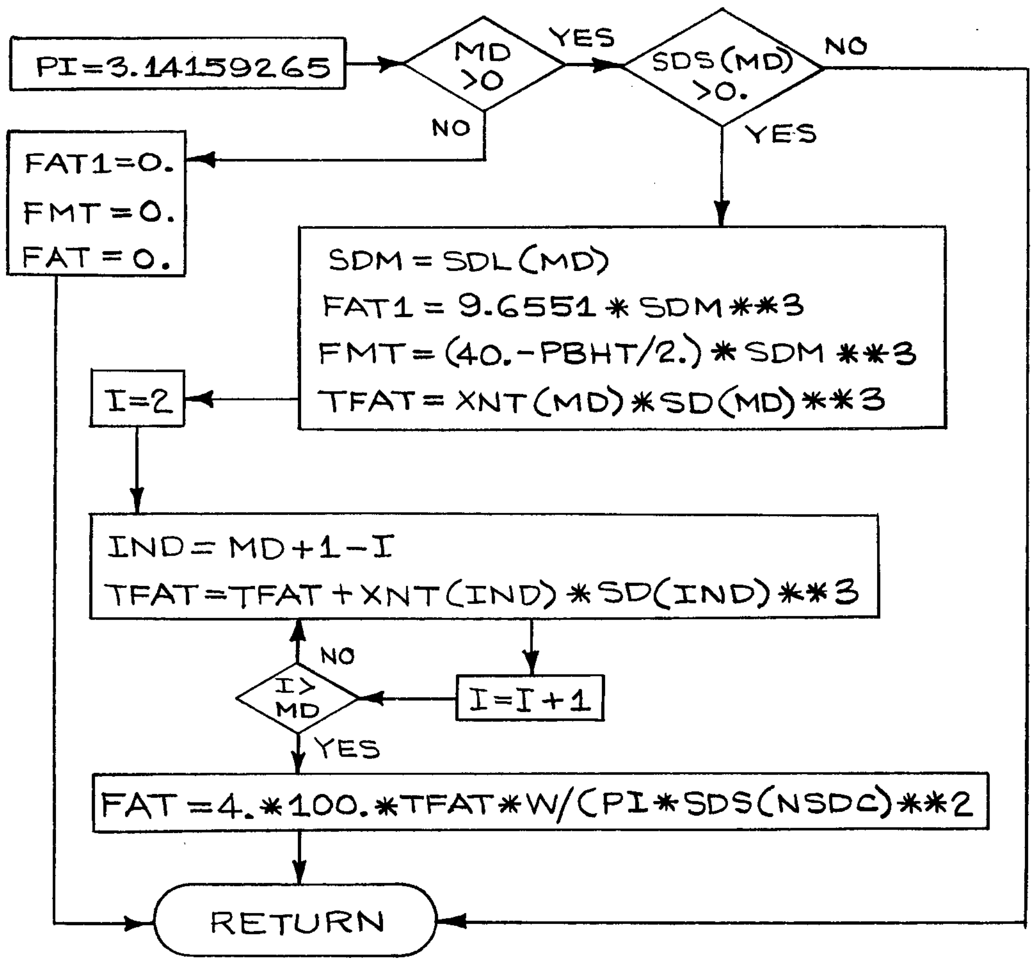
Subroutine VEGF calculates the forces associated with overriding vegetation: FAT1, the force required to override one tree; FMT, the maximum force required to override all trees; and FAT, the average force required to override all trees. This subroutine is called nine times from PATCH. The first time, the incoming variable MD is zero, indicating that all trees are being avoided; therefore, FAT1, FMT and FAT are set to zero, and a return is made. For other values of MD (1 through 8), varying numbers of tree sizes are being overridden. For example, when MD = 1, stem diameters of class 1 are being overridden, and all others are avoided; when MD = 2, stem diameters of classes 1 and 2 are being overridden, and all others are avoided; etc. Next, a check is made to see if the mean spacing of stem diameter class 1 is greater than zero. If it is not, an exit is made from the subroutine. Then, the maximum stem diameter to be overridden is calculated as variable SDM, taken to be the upper limit of the largest stem diameter class to be overridden as indicated by the variable MD.

The force to override one tree, FAT1, based on the maximum stem diameter to be overridden, is calculated next, followed by the calculation of FMT, the maximum forces involved in overriding stems. FMT is based on the pushbar height of the vehicle versus the stem diameter. Variable TFAT is then calculated. This is a summation of the work required to override all diameter classes of trees to be run over.

Finally, a loop is entered in which the stem diameter sizes and the associated forces are accumulated from I = 2 to MD, the largest tree to be overridden; and the variable FAT, the average force to override trees, is calculated. FAT is equal to TFAT, the total work required in overriding trees, divided by a quantity equal to the area of a circle whose radius is the mean spacing for the given stem diameter class divided by the width of the vehicle. The three variables - FAT1, FMT, and FAT - are now returned to the calling program.

SUBRØUTINE VEGF

VARIABLES ENTERING: MD, PBHT, SD(10), SDS(10), SDL(10), XNT(10), W, NSDC



VARIABLES LEAVING: FAT1, FMT, FAT

FIG. C53

VEGF

```
      SUBRØUTINE VEGF(MD,PBHT,RD,SDS,SDL,XNT,W,DAT1,FMT,FAT,
&  NSDC)
      DIMENSIØN SD(10),SDS(10),XNT(10),SDL(10)
      PI=3.14159265
      IF (MD)15,15,5
    5 IF (SDS(MD()7,7,25
    7 RETURN
   25 SDM=SDL(MD)
      FAT1=9.6551*SDM**3
      FMT=(40.-PBHT/2.)*SDM**3
      TFAT=XNT(MD ) *SD(MD)**3
      DØ 10 I=2,MD
      IND=MD+1-I
      TFAT=TFAT+XNT(IND)*SD(IND)**3
   10 CØNTINUE
      FAT=4.*100.*TFAT/(PI*SDS(NSDC)**2)*W
      RETURN
   15 FAT1=0.
      FMT=0.
      FAT=0.
      RDTURN
      END
```

Subroutine AREAT (Fig. C54)

Subroutine AREAT takes the area denied by both vegetation and obstacles and determines a speed reduction factor due to the necessary maneuvering. The speed reduction factor is a fraction, equal to or less than one, that multiplies the vehicle actual speed to obtain the effective speed across a patch. For example, if the vehicle must travel twice as far, due to maneuvering, the speed reduction factor is 0.5.

The percentage of total area denied is:

ADT = ADØ + PAV * (100. - ADØ)/100

where:

ADT = total percentage of area denied

ADØ = percentage of area denied by obstacles

PAV = percentage of area denied by vegetation

If ADT 10:

SRF = 1.0;

If ADT 50,

SRF = 0.0;

Otherwise:

SRF = 1.0 - (ADT - 10.)/40.0

Where:

SRF = speed reduction factor.

Note that, since the trees are assumed evenly spaced, some trees will be in the area denied by obstacles.

SUBRØUTINE AREAT

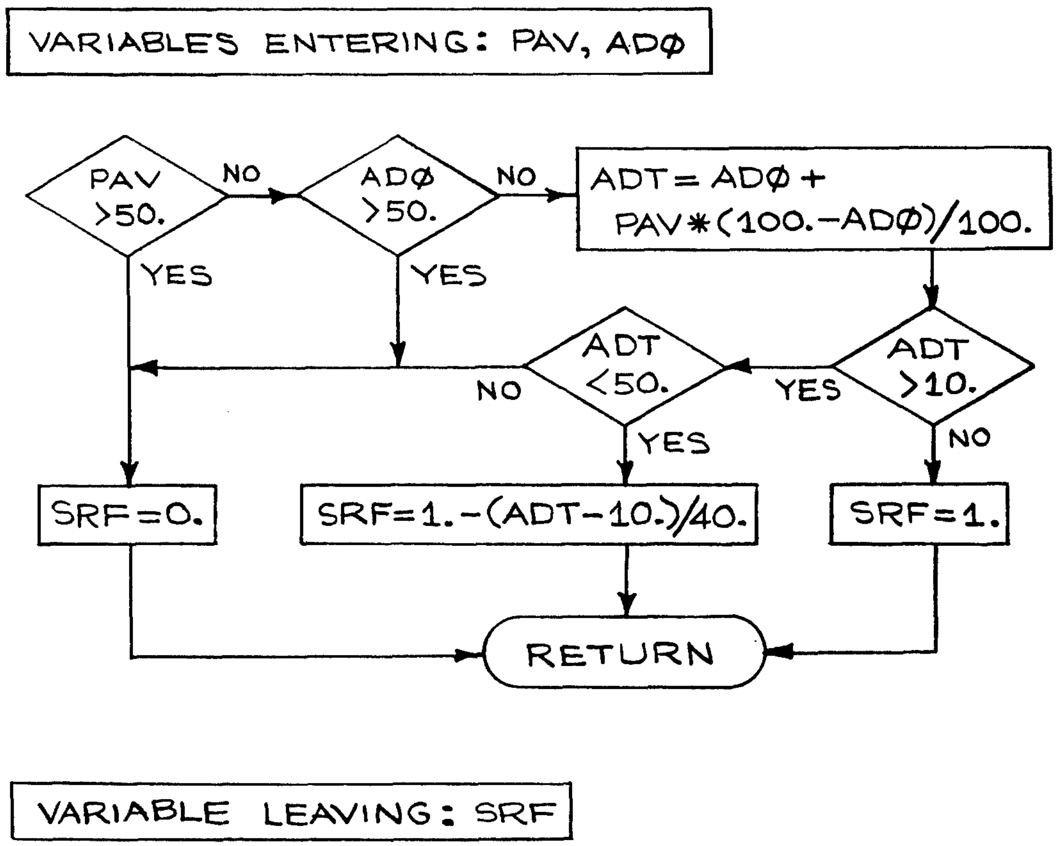


FIG. C54

AREAT

```
      SUBRØUTINE AREAT(PAV,ADØ,SRF)
      IF(PAV.GT.50..ØR.ADØ.GT.50.)GØTØ 6
      ADT=ADØ+PAV*(100.-ADØ)/100.
      IF (ADT-10.)1,1,5
    1 SRF=1.0
      RETURN
    5 IF (ADT-50.)7,6,6
    6 SRF=0.0
      RETURN
    7 SRF=1.0-(ADT-10.0)/40.0
      RETURN
      END
```

Subroutine RIVER (Fig. C55)

Subroutine RIVER calculates the time penalties for crossing a river and for ingress and egress. The program is in four parts: In the first part, the time penalty for fording the river is calculated if the vehicle can ford; in the second, the time penalty for swimming is calculated if the vehicle can swim, and the water is too deep to ford; in the third, a rafting penalty is assigned if the vehicle can neither swim nor ford; and in the fourth, the time penalty for egress is calculated. First, a check is made to determine whether the water depth is greater than the fording depth. If it is, the vehicle cannot ford, and an exit is made to the swimming portion of the program; if the vehicle can ford, a check is made on water speed. If the water speed is greater than 11 mph, the vehicle cannot successfully ford, and an exit is made to the rafting portion of the program; if the water speed is less than 11 mph, a check is made on vehicle type. If the vehicle is tracked, the fording calculation is unnecessary, and an exit is made to a later part of this portion of the program; if the vehicle is wheeled, intermediate calculations must be performed.

First, variable THM, the maximum dropoff angle before belly hangup, is calculated. If this is less than the ingress bank angle, there must be a call to subroutine DIG to determine the time penalty for excavating the ingress bank until the vehicle can successfully enter without belly hangup. This time penalty returns as ATP and is loaded into a variable, TP, which accumulates the ingress and egress penalties. Next, the vehicle approach angle is calculated, and 5 degrees are added to it. This is checked against the bank angle of the river. If this angle is less than the bank angle, a nose-in hang-up would occur, additional excavation is necessary, and another call go DIG is made. The time penalty returns as ATP and is accumulated into variable TP. Next, variable TV, the velocity made good in crossing the river, is calculated. It is simply equal to the vehicle fording speed. Next, variable TN, the time required to cross the river, is calculated from the speed in crossing and the width of the river. An exit is now made to the egress routine.

If the swimming routine has been entered, a check is first made to see if the vehicle swimming speed is greater than zero, i.e., to see if the vehicle can swim. (If it cannot, an exit is made to the rafting portion of the program.) A check is then made to see if the water speed is greater than the vehicle swimming speed times an auxiliary water propulsion factor, which takes into consideration such things as shrouding and water jets. If the water speed is too great, an exit is made to the rafting portion of the program; if not, further computations in the swimming portion of the program are continued. First, the effective bank height, EBH, is calculated. This is the height from the top of the ingress bank to a point below water level equal to the fording depth or draft height of the vehicle. A check is then made against the vehicle ingress swamp angle. If this is less than the bank angle, THI, a call is made to DIG to calculate a penalty for excavation. This penalty returns as ATP and is loaded into variable TP. Next, a check is made on vehicle type. If the vehicle is tracked, most of the calculations in this portion of the program are unnecessary, and control is directed to the last part of the swimming model. If the vehicle is wheeled, a check is made on the belly clearance of the vehicle. If the vehicle will hang up on its underbelly, another call to DIG must be made for further excavation. This time penalty returns as ATP and is accumulated in variable TP. Finally, from the last equations of this portion of the program, TV, the velocity made good in crossing the river, is calculated. This takes into consideration the water speed that will cause the vehicle to travel downstream as it crosses the river. When this velocity is known, the time penalty for crossing is calculated by using the river width. This is loaded into variable TN, and an exit is made to the egress portion of the program.

If the rafting routine has been entered, water speed is first checked to see whether it is greater than 11 mph. If it is, a time penalty of 180 minutes is assigned for constructing a raft; if the water speed is less than 11 mph, rafts can be constructed in shorter periods of time, but this time is dependent upon the water speed. If the vehicle weighs less than 28,000 lbs., the time penalty for constructing a raft is 25 minutes; if it weighs between 28,000 and 42,000 lbs., the time penalty is 45 minutes; and if it weighs more than 42,000 lbs., the time penalty is 90 minutes. After this penalty has been assigned, the velocity for crossing the river is assumed at 0.68 mph, since this is approximately the rate at which a raft can be maneuvered across a river. The time penalty in crossing, TN, is calculated from the river width, and an exit is made to the egress portion of the program.

The egress portion of the program is based on the calculation of the severity factor, SF. This is an empirical factor based on the experience of engineers at TACOM. The severity factor is an equivalent step height assigned to each exit bank. This is compared with the step height the vehicle is capable of climbing. Next, the program accumulates the severity factors for each of the distinct slopes on the bank. In this generation of the model, only one slope is allowed, and this loop is therefore ignored. Next, a check is made to see if the bank severity factor exceeds the step height that the vehicle can negotiate. If it does not, no further time penalties are assigned, and a return is made to the calling program. If it does, a check is made to see if the severity factor is greater than two times the step height the vehicle can manage. If it is not, a time penalty is calculated based on the difference between the severity factor and the step height the vehicle can manage; if it is greater, excavation is necessary; and variable ESL, the effective slope of the bank, and ESLH, the effective height of the bank, are calculated. Then the traction that the vehicle can generate on this bank is calculated, using c, the soil cohesion, and ∅, the angle of the repose of the soil. If the vehicle has a winch, the winch capacity is added to this tractive force. A traction-limited slope angle, THEM, is then calculated. If this traction-limited slope is less than the effective slope of the bank, excavation is necessary, and a call to subroutine DIG is made to reduce this slope to an angle that the vehicle can climb, based upon available traction and winch capacity. This time penalty returns as variable ATP and is accumulated in variable TP, which carries the penalties for ingress and egress. If the vehicle has a winch, an additional 10-minute setup time is added to time penalty TP. The two variables leaving the program are TN, the time for crossing the river either by fording, swimming, or rafting, and TP, the time penalties associated with ingress and egress.

SUBRØUTINE RIVER

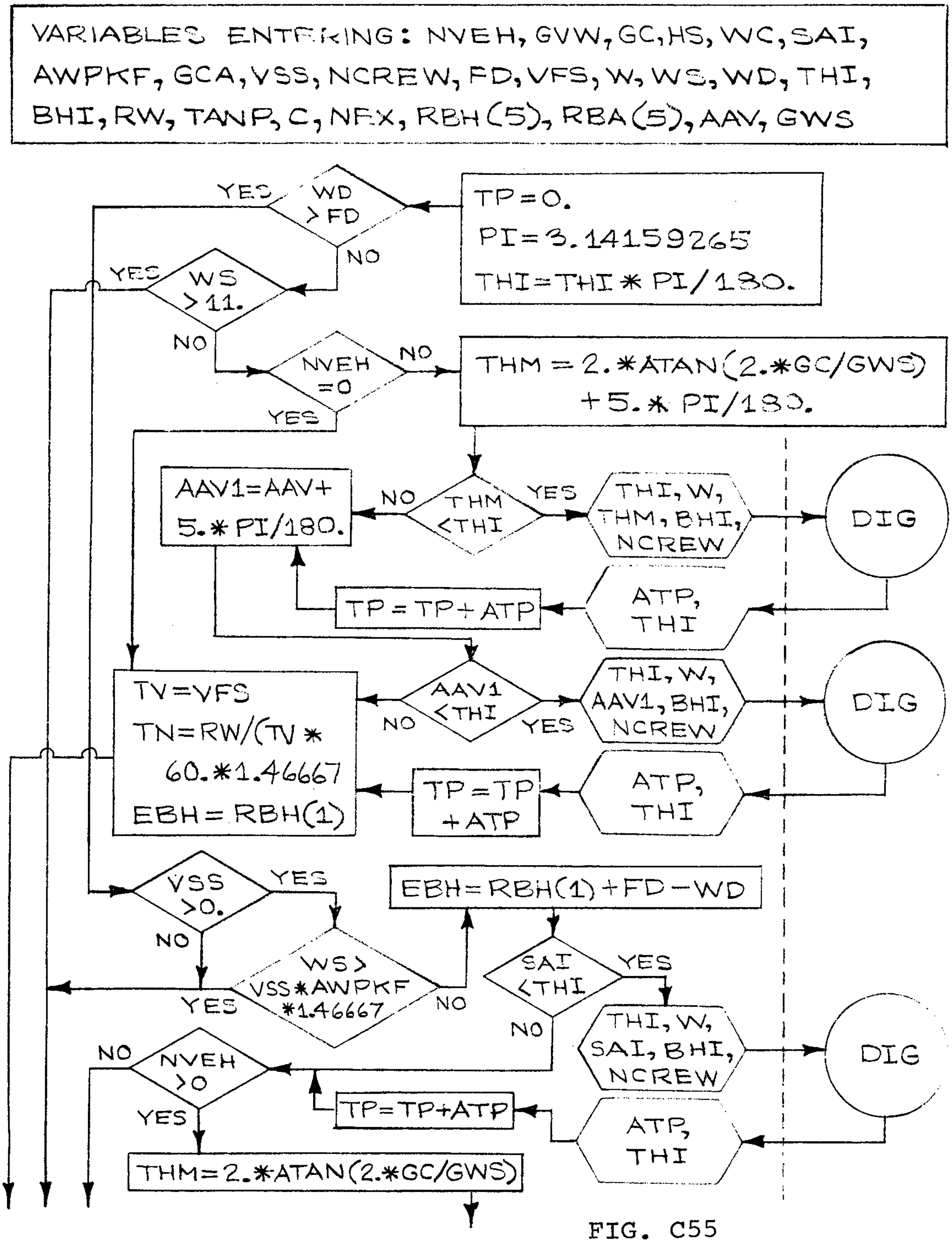


FIG. C55

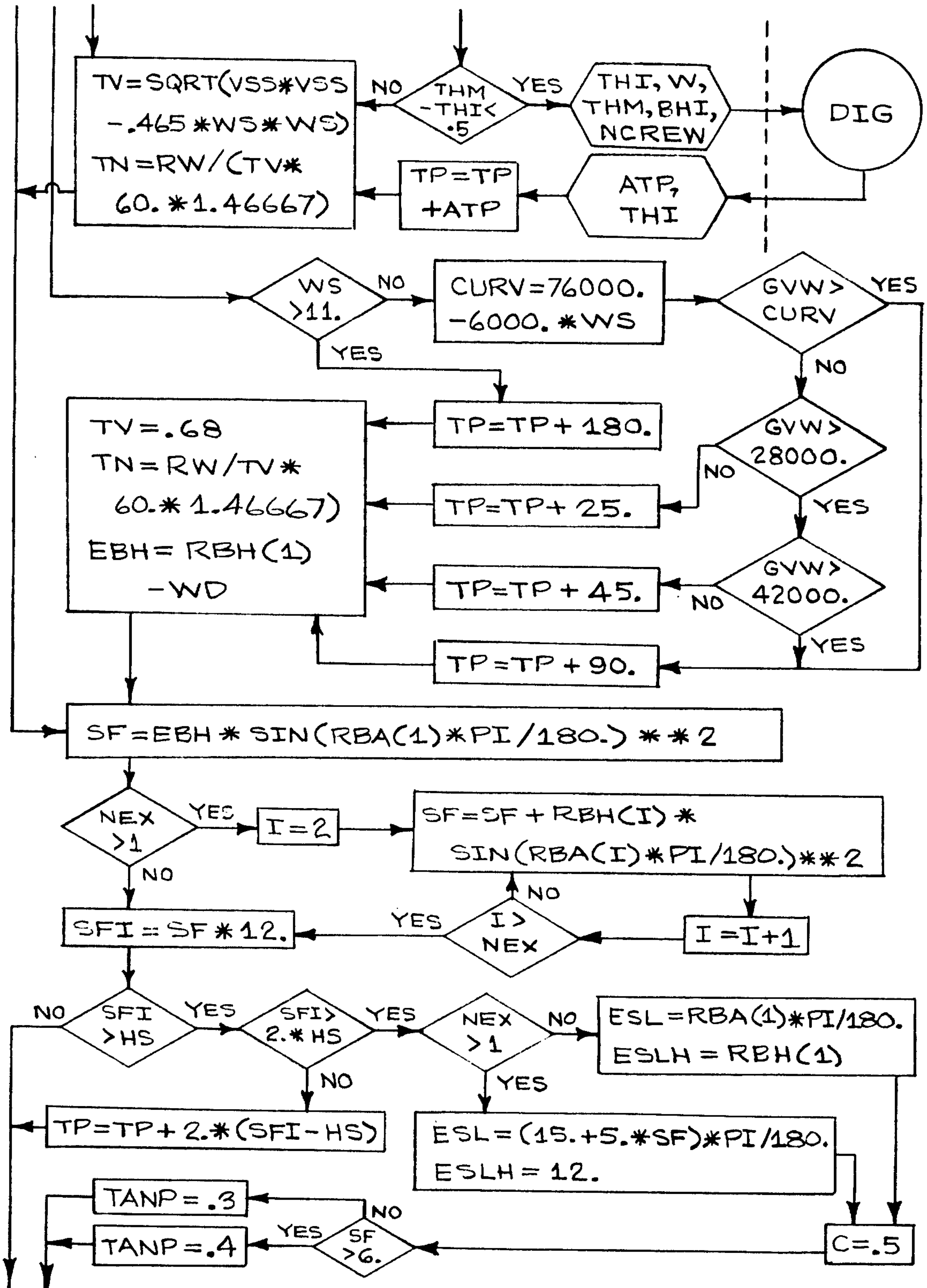


FIG. C55 cont'd

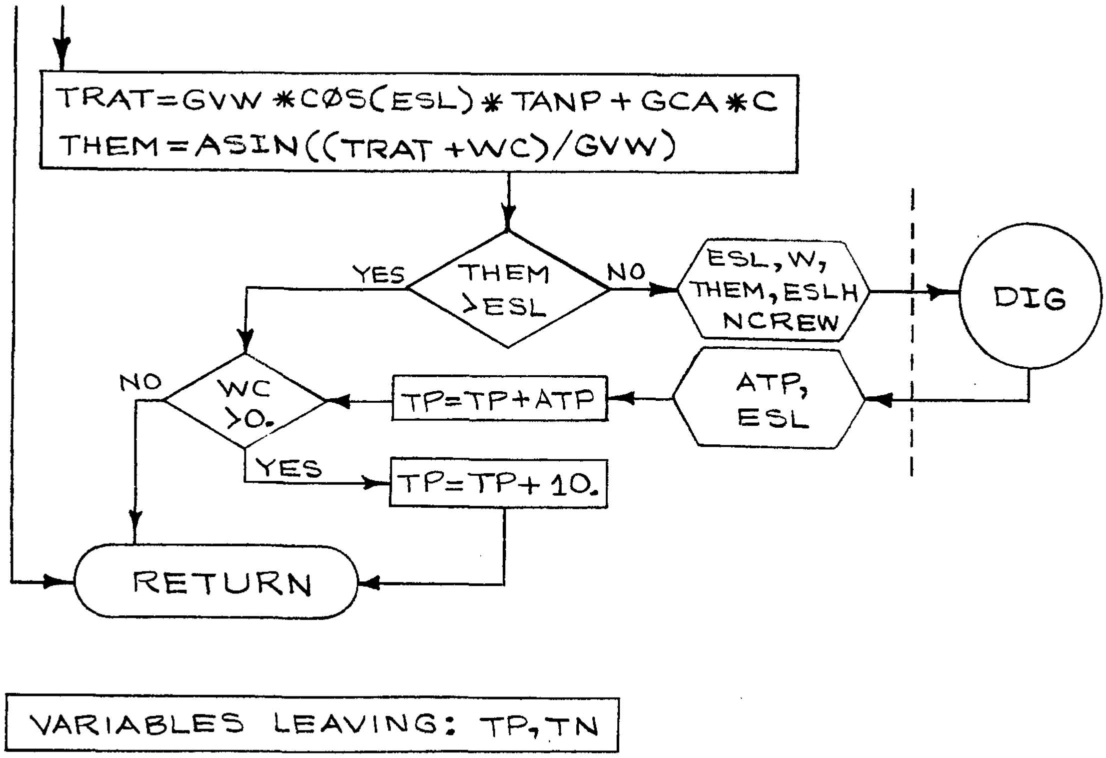


FIG. C55 cont'd

RIVER1

```
      SUBRØUTINE RIVER(AAV,GWS,TP,TN)
      CØMMØN IPATCH(325),FØRCE(2,101),FØRCR(4,101),FØRMX(3),
&     TDM(3),RT,RCI,NVEH,NFL,GVW,DL,WID,GT,A,NBC,GC,HPT,ITVAR,
&     RDIAM,TPSI,TPLY,HS,WC,SAI,AWPKF,GCA,VSS,NCREW,FD,VFS,
&     ITRACT(399),VØØB(2,30),VRIDE(20),W,PBHT,PBF,VL,NC4,NC5,
&     IØBST(32),WS,WD,THI,BHI,RW,TANP,C,NDX,RBH(5),RBA(5)
      TP=0.0
      PI=3.14159265
      THI=THI*PI/180.
      IF (WD-FD)1,1,10
    1 IF (WS-11.0)2,2,90
    2 IF (NVEH.EQ.0) GØ TØ 4
      THM=2.*ATAN(2.*GC/GWS)+5.*PI/180.
      IF (THM-THI)80,3,3
   80 CALL DIG(THI,THM,BHI,W,ATP,NCREW)
      TP=TP+ATP
    3 IF (AAV+5.*PI/180. -THI)81,4,4
   81 CALL DIG(THI,AAV+5.*PI/180.,BHI,W,ATP,NCREW)
      TP=TP+ATP
    4 TV=VFS
      TN=RW/(1.46667*TV*60.)
      EBH=RBH(1)
      GØ TØ 100
   10 IF (VSS)90,90,11
   11 IF (WS.GT.VSS*AWPKF*1.46667) GØ TØ 90
      EBH=RBH(1)+FD-WD
      IF (SAI-THI)12,13,13
   12 CALL DIG(THI,SAI,@HI,W,ATP,NCREW)
      TP=TP+ATP
   13 IF (NVEH)16,16,14
   14 THM=2.*ATAN(2.*GC/GWS)
      IF (THM-THI-5.)15,16,16
   15 CALL DIG(THI,THM,BHI,W,ATP,NCREW)
      TP=TP+ATP
   16 TV=SQRT(VSS*VSS-0.465*WS*WS)
      TN=RW/(1.46667*TV*60.)
      GØ TØ 100
   90 IF (WS-11.0)92,92,91
   91 TP=TP+180.
      GØ TØ 99
   92 CURV=76000.-6000.*WS
      IF (GVW.LE.28000. .AND. GVW.LE.CURV) GØ TØ 95
      IF (GVW.LE.42000. .AND. GVW.LE.CURV) GØ TØ 96
      TP=TP+90.
      GØ TØ 99
   95 TP=TP+25.
      GØ TØ 99
   86 TP=TP+45.
   99 TV=0.68
      TN=RW/(1.46667*TV*60.)
```

RIVER1 CØNTINUED

```
      EBH=RBH(1)-WD
100   SF=EBH*SIN(RBA(1)*PI/180.)**2
      IF (NEX-1)103,103,101
101   DØ 102 I=2,NEX
      SF=SF+RBH(I)*SIN(RBA(I)*PI/180.)**2
102   CØNTINUE
103   SFI=SF*12.
      IF (SFI.GT.HS) GØ TØ 104
      TP=TP+0.0
      GØ TØ 900
104   IF (SFI.GT.2.0*HS) GØ TØ 105
      TP=TP+2.0*(SFI-HS)
      GØ TØ 900
105   IF (NEX-1)108,108,110
110   ESL=(15. + 5.0*SF)*PI/180.
      ESLH=12.
      GØ TØ 111
108   ESL=RBA(1)*PI/180.
      ESLH=RBH(1)
111   IF (SF-6.)112,112,113
112   TANP=0.3
      GØ TØ 114
113   TANP=0.4
114   C=0.5
      TRAT=GVW*CØS(ESL)*TANP+GCA*C
      THEM=ASIN((TRAT+WC)/GVW)
      IF (THEM-ESL)106,106,107
106   CALL DIG(ESL,THEM,ESLH,W,ATP,NCREW)
      TP=TP+ATP
107   IF (WC.GT.0.0) TP=TP+10.
900   CØNTINUE
      RETURN
      END
```

Subroutine DIG (Fig. C56)

Subroutine DIG calculates the time penalty associated with excavating a river bank to permit the egress of a vehicle. First, the volume of a triangular prism of dirt on the exit bank, variable VX, is calculated. This will lower the bank angle to the point where the vehicle is permitted egress. Next, a time penalty for this excavation, variable ATP, is calculated. It is assumed that each member of the crew can excavate 1 cubic foot of dirt in five minutes. Also, the bank angle that entered the program is reduced to the new angle following excavation. This new bank angle and the time penalty associated with the excavation are returned to the calling program. This subroutine is called only from subroutine RIVER.

SUBRØUTINE DIG

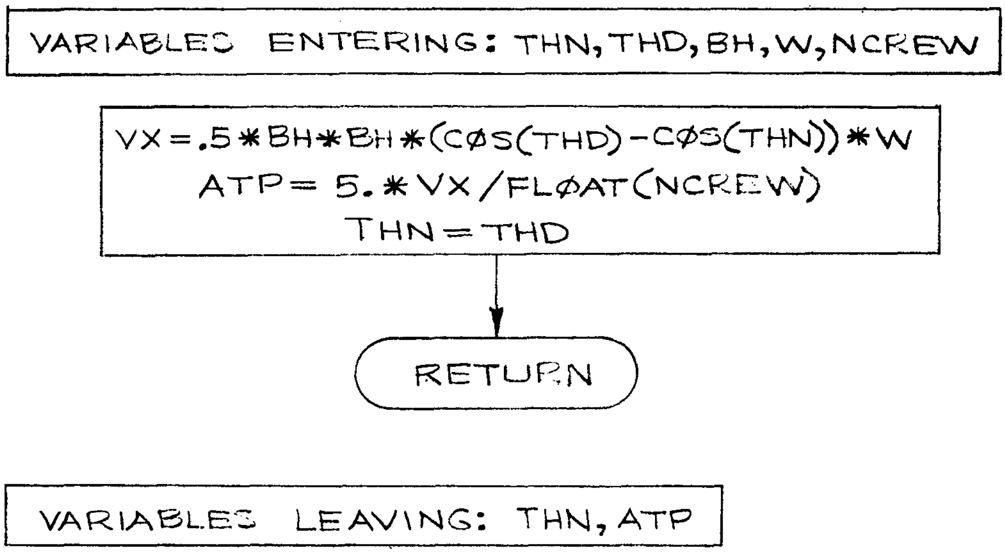


FIG. C56

DIG

```
      SUBRØUTINE DIG(THN,THD,BH,W,ATP,NCREW)
      VX=0.5*BH*BH*(CØS(THD)-CØS(THN))*W
      ATP=5.*VX/FLØAT(NCRDW)
      THN=THD
      RETURN
      END
```

Subroutine RØUTE (Fig. C57)

Subroutine RØUTE consists of two parts. In the first part, the time required to traverse each path-segment of the map is calculated. There are 750 such path-segments, 25 each in 30 sections of the Puerto Rico terrain map. The second part of this subroutine uses these data and calculates, by means of a dynamic programming scheme, the best route through the map. Three large arrays enter this subroutine from the main program, having been previously calculated in the other subroutines. These are: array V with 1080 elements, which contains the velocities the vehicle can manage in each of 1061 normal patches and 19 marshes; array VR, which contains the time required to cross a river for each of 10 rivers; and array TPR, which includes the ingress and egress times.

Another array, S(I, J, K), is calculated in the first part of the program. The first subscript indicates the starting point of a given path-segment, the second indicates the ending point, and the third indicates the section of the map. The first thing done in the subroutine is to initiate all elements of this array to zero, as this is necessary for some machines. Then, the first of three data files is called, data file SECPUE, which contains the data for each path-segment in the map. The data consist of the type number of the patch traversed and the distance traversed on that patch, for each of the patches encountered along that path-segment. The variable N, which contains the number of patches encountered, is also read. Variable DP contains the distance across the patch; variable IP contains the patch type number. There will be N pairs. For each patch crossed, the patch type number, IP, is used as the index in searching array V to determine the velocity in that patch. The distance, DP, is divided by that velocity to produce the time to cross the patch; and these times are accumulated for each patch crossed. If it is found that one of the patches has zero velocity, i.e., if the vehicle is immobilized on that patch, the time to cross it is arbitrarily set at 600,000 seconds. This overrides any other times accumulated and indicates at the end of the calculation that the vehicle is immobilized on this path-segment. The time data are calculated for each path-segment in the map, of which there are 750, and the values are stored in array S. When this is completed for all of the normal patches, data file SECPUE is closed, and data file MSHPUE is opened.

Data file MSHPUE contains the information as to which marshes are crossed on each path-segment. Since the data for marshes are in the same format as the data for patches, the calculation just described is re-entered and performed for each of the marshes. The times for the marshes are also loaded into the corresponding locations in array S. When this is completed for marshes, data file MSHPUE is closed, and data file RIVPUE is opened.

Data file RIVPUE contains the information as to which rivers are crossed on each path-segment. Returning from this file is the variable N, the number of rivers on a given path-segment, and variable IP(I), which contains the type numbers of the particular rivers crossed. There will be N of these. IP is used as an index in arrays VR and TPR to determine the time penalties associated with this river crossing. These time penalties are also loaded into array S as additional times associated with crossing the given path-segment. All of the data as to times on path-segments have now been accumulated in array S, and this array is ready for use in the dynamic programming model.

There are, however, other things calculated in this part of the subroutine. It is necessary to know on what percentage of the area of a map the vehicle is constrained to certain velocity ranges. These ranges are 0 mph, or immobilization; 0-2 mph; 2-4 mph; 4-6 mph; 6-8 mph; 8-10 mph; and more than 10 mph. This information is accumulated in array AREA; this array is printed out to the terminal. Additionally, array S is written into an external file for later manipulation if required. This external file is called VEHNAM (this stands for vehicle name). The name, of course, is temporary, since later, after the program has been run, this file will be called and renamed for the specific vehicle.

Now the dynamic programming model is entered. The object of this model is to select the best route through the map. It was found that it is appropriate to start at the termination of the map and work backward, selecting at each stage the best route to that point. The map was divided into 30 sections, resulting in 31 interfacing lines between sections. In dynamic programming terminology, these are referred to as decision stages. There are five points equally spaced upon each of these lines, including the beginning and ending lines. The calculations begin at stage 30. From each point at stage 30, there are five path-segments proceeding to the five points at stage 31. For each point at stage 30, the best of the five possible path-segments is selected; and the time associated with that path is collected in array PØDE. When this part of the calculation is completed, the best path from each of the points at stage 30 is known and is loaded into this array. The point on line 31 to which this best path proceeds is loaded into array NØDE. Calculations now proceed backward to line 29. From each point on line 29, there are five path-segments proceeding to the five points on line 30. The time associated with the path-segment from the point on line 29 to each point on line 30 is added to the best time from that point on line 30 to line 31. This is done for each of the five lines, and the best of these sums is selected to be the best route from that point on line 29 to the designation stage. This sum is loaded into array PØDE, and the point on line 30 is entered in NØDE. Now the best path from each point on line 29 to the destination is known. Calculations proceed to line 28, and the same operation is repeated until line 1 is reached, at which point the best path from each of the five points on line 1 to the destination, line 31, is known. The calculation than selects the best of these five paths to be the finally selected best route through the map.

The time accumulated in PØDE for the best path is now loaded into variable P. The points along the path that were loaded into array NØDE are now extracted for the best path; these 31 points along this path are loaded into array NØDEF. Now the time for the best route and the individual points along that route are known.

It is next necessary to calculate the speed made good across the map. The distance from one end of the map to the other (the shortest distance if the map is folded) has been collected in variable DIST, in feet; this value is divided by 5280 to produce DISTM, in miles. Variable P contains the best route time in minutes, so it is divided by 60 to yield the best time in hours in variable SUMTØT, which is averaged to yield the average velocity across the map in miles per hour. This value is loaded in variable AVGV. The following information then is printed out to the terminal: the individual points along the best route, the time for this route, and the average velocity for this best route. When this is completed, a return is made to the main program.

SUBRØUTINE RØUTE

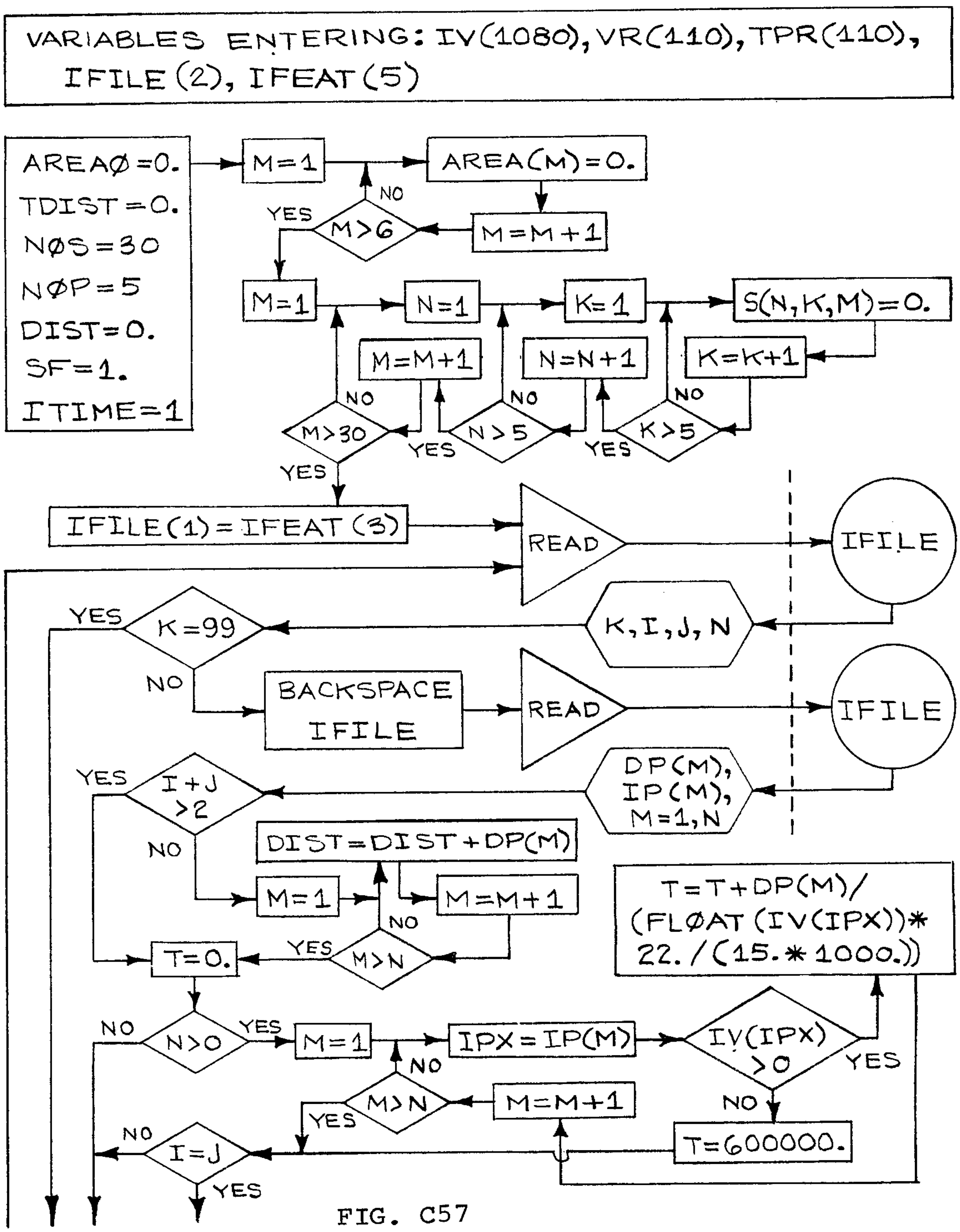


FIG. C57

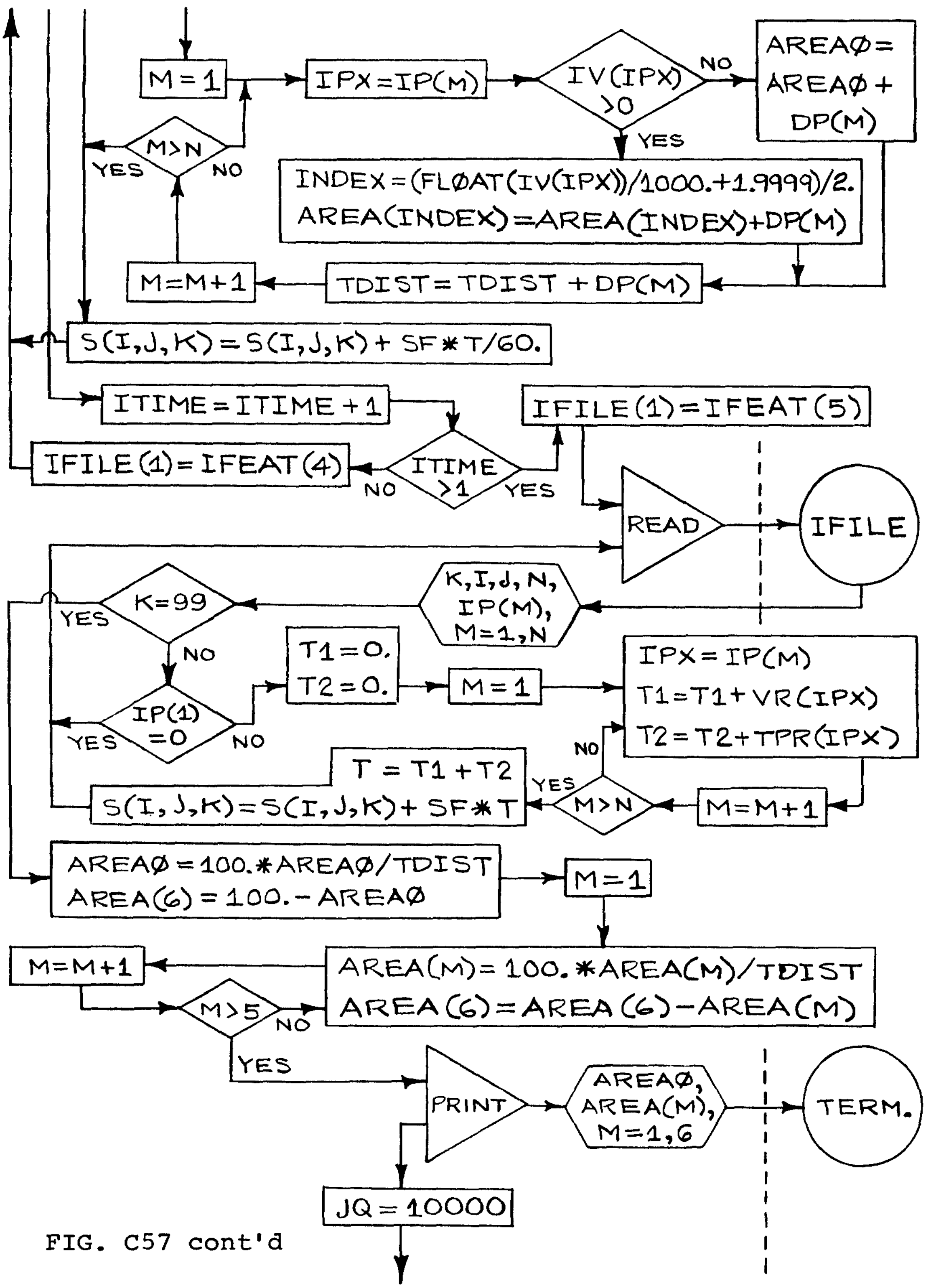


FIG. C57 cont'd

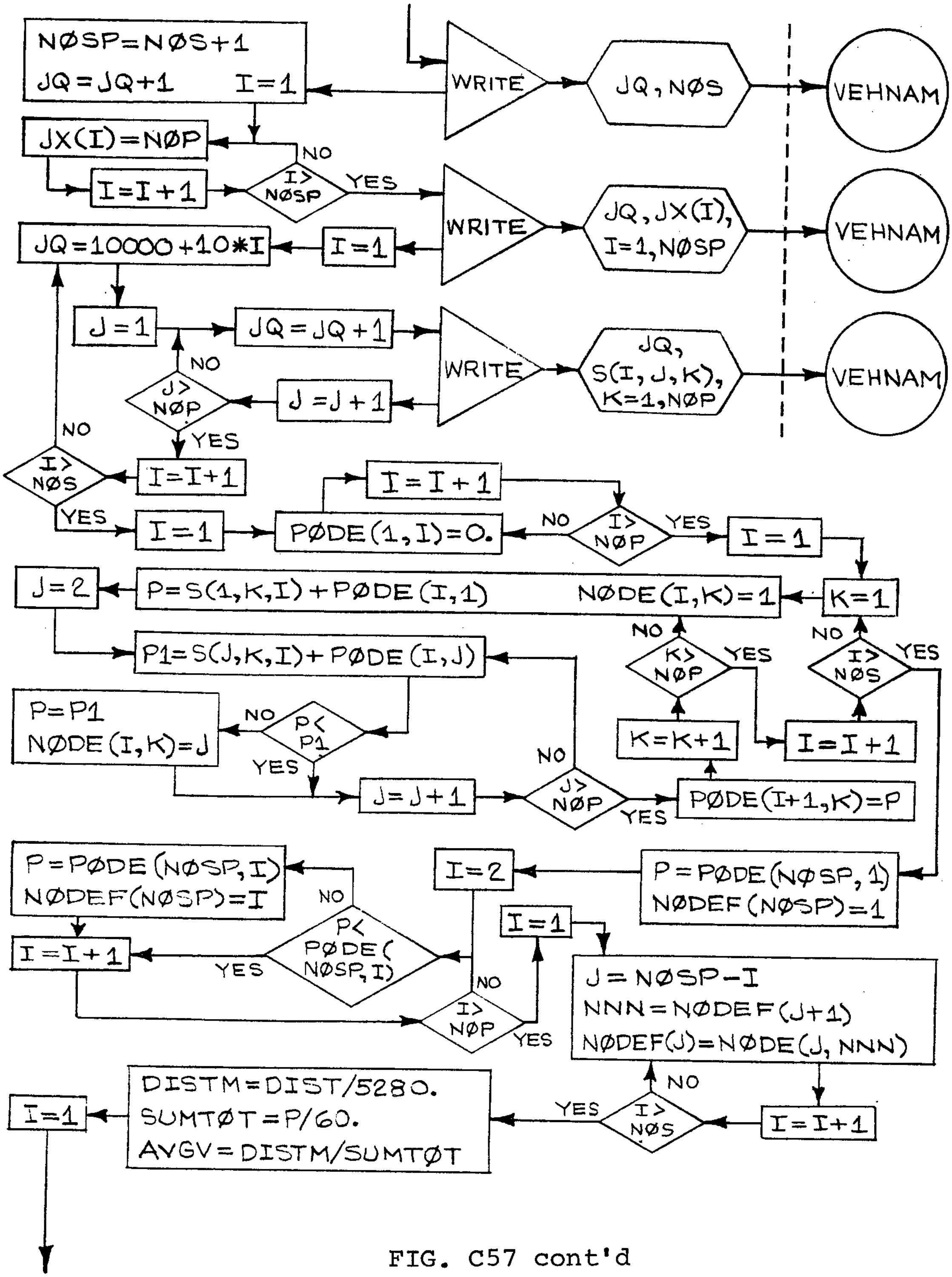


FIG. C57 cont'd

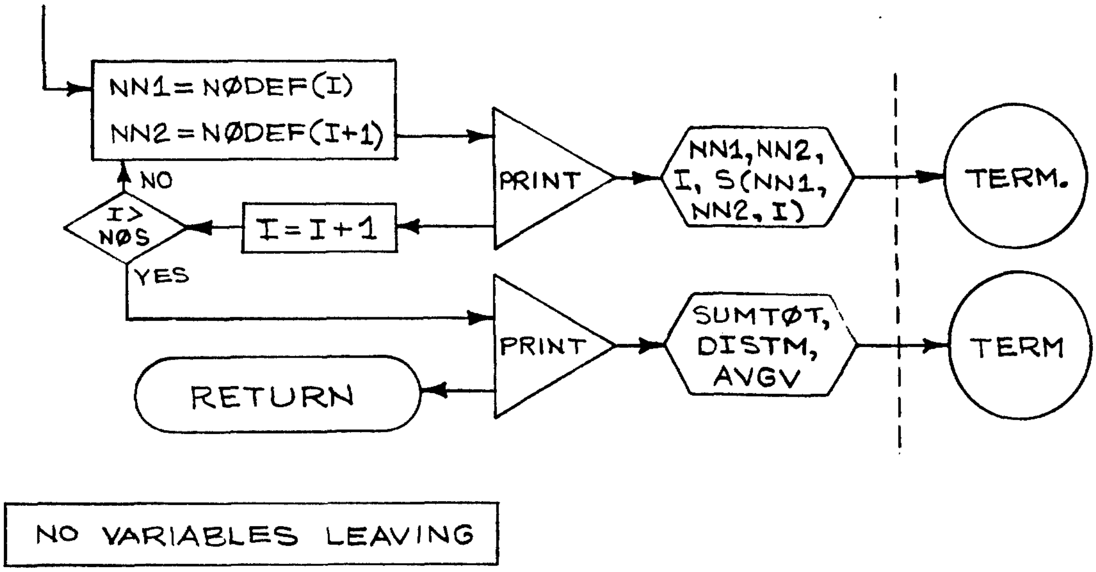


FIG. C57 cont'd

RØUTE

```
      SUBRØUTINE RØUTE(IFILE,IFEAT)
      DIMENSIØN IFILE(2),IFEAT(5)
      CØMMØN IPATCH(2230),IV(1080),VR(110),TPR(110)
      DIMENSIØN S(5,5,40),IP(50),DP(50)
      DIMENSIØN PØDE(31,5),NØDE(30,5),NØDEF(31)
      DIMENSIØN AREA(25)
      DIMENSIØN JX(31)
      AREAØ=0.0
      DØ 61 M=1,25
   61 AREA(M)=0.0
      TDIST=0.0
      NØS=30
      NØP=5
      DIST=0.
      NØSM2=NØS-2
      NØSM1=NØS-1
      SF=1.
      ITIME=0
      DØ 1 M=1,40
      DØ 1 N=1,5
      DØ 1 K=1,5
      S(N,K,M)=0.0
    1 CØNTINUE
      IFILE(1)=IFEAT(3)
      CALL ØPENF(1,IFILE)
C
   20 READ (1;101) K,I,J,N
   16 IF (K.EQ.99) GØ TØ 21
      BACKSPACE 1
      READ (1;102) (DP(M),IP(M),M=1,N)
      IF (ITIME.GT.0) GØ TØ 19
   19 IF (I+J.GT.2) GØ TØ 22
      DØ 32 M=1,N
      DIST=DIST+DP(M)
   32 CØNTINUE
   22 T=0.0
      IF (N)23,26,27
   27 DØ 25 M=1,N
      IPX=IP(M)
      IF(IV(IPX))23,23,24
   23 T=600000.
      GØ TØ 28
   24 T=T+DP(M)/(FLØAT(IV(IPX))*22./(15.*1000.))
   25 CØNTINUE
   28 IF (ITIME.EQ.0 .AND. I.EQ.J) GØ TØ 62
      GØ TØ 26
   62 DØ 5 M=1,N
      IPX=IP(M)
      IF(IV(IPX))2,2,3
    2 AREAØ=AREAØ+DP(M)
```

RØUTE CØNTINUED

```
      GØ TØ 4
    3 INDEX=(FLØAT(IV(IPX))/1000.+1.99999)/2.0
      AREA(INDEX)=AREA(INDEX)+DP(M)
    4 TDIST=TDIST+DP(M)
    5 CØNTINUE
   26 S(I,J,K)=S(I,J,K) + SF*T/60.
      T=T/60.
      GØ TØ 20
   21 CALL CLØSEF(1)
      ITIME=ITIME+1
      GØTØ(58,57)ITIME
   58 IFILE(1)=IFEAT(4)
      CALL ØPDNF(1,IFILE)
      GØ TØ 20
   57 IFILE(1)=IFEAT(5)
      CALL ØPENF(1,IFILE)
C
   50 READ (1;105) K,I,J,N,(IP(M),M=1,N)
   40 IF (K.EQ.99) GØ TØ 56
      IF (IP(1).EQ.0) GØ TØ 50
   52 T1=0.0
      T2=0.0
      DØ 53 M=1,N
      IPX=IP(M)
      T1=T1+VR(IPX)
      T2=T2+TPR(IPX)
   53 CØNTINUE
      T=T1+T2
      S(I,J,K)=S(I,J,K) + SF*T
      GØ TØ 50
  100 FØRMAT (1X,2A5)
  101 FØRMAT(I2,2I1,1X,I3)
  102 FØRMAT(8X,F6.0,I5,F6.0,I5,F6.0,I5,F6.0,I5,F6.0,I5)
  103 FØRMAT (" FRØM",I3," TØ",I3," ØN SEG",I3," IN",F9.2
&    ," MIN")
  105 FØRMAT(I2,2I1,1X,20I3)
  200 FØRMAT (28X," ---------",/,4X,"TØTAL TIME ØF TRAVERSE ",
&    F9.2," MINUTES")
  201 FØRMAT (/////," ENTER NAME ØF CØURSE",/)
   56 CØNTINUE
      CALL CLØSEF(1)
   18 AREAØ=AREAØ/TDIST*100.
        AREA(6)=100.-ARDAØ
        DØ 6 M=1,5
        AREA(M)=AREA(M)/TDIST*100.
        AREA(6)=AREA(6)-AREA(M)
    6 CØNTINUE
        PRINT 80,@REAØ,(AREA(M),M=1,6)
   80 FØRMAT (///," PERFØRMANCE BX AREA",//,
&    7(2X,F5.1," % ")/4X," 0.0",6X,"0 TØ 2",4X,"2 TØ 4",4X,
```

RØUTE CØNTINUED

```
&       "4 TØ 6",4X,"6 TØ 8",3X,"8 TØ 10",5X,"> 10",
&       /15X,"VELØCITY RANGE--MPH"/////)
              CALL ØPENF(1,"VEHNAM")
              JQ=10000
              WRITE (1;1000) JQ,30
              JQ=JQ+1
              DØ 2000 I=1,31
        2000  JX(I)=5
              WRITE (1;1001) JQ,(JX(I),I=1,31)
              DØ 2001 I=1,30
              JQ=10000+10*I
              DØ 2001 J=1,5
              JQ=JQ+1
        2001  WRITE (1;1002) JQ,(S(J,K,I),K=1,5)
        1000  FØRMAT(I5,1X,I3)
        1001  FØRMAT(I5,1X,31I2)
        1002  FØRMAT(I5,1X,5F13.6)
              CALL CLØSEF(1)
C
              DØ 500 I=1,5
         500  PØDE(1,I)=0.
              DØ 302 I=1,30
              DØ 302 K=1,5
              P=S(1,K,I)+PØDE(I,1)
              NØDE(I,K)=1
              DØ 301 J=2,5
              P1=S(J,K,I)+PØDE(I,J)
              IF(P.LT.P1)GØTØ 301
              P=P1
              NØDE(I,K)=J
         301  CØNTINUE
         302  PØDE(I+1,K)=P
              P=PØDE(31,1)
              NØDEF(31)=1
              DØ 303 I=2,5
              IF(P.LT.PØDE(31,I))GØTØ 303
              P=PØDE(31,I)
              NØDEF(31)=I
         303  CØNTINUE
              DØ 304 I=1,30
              J=31-I
              NNN=NØDEF(J+1)
         304  NØDEF(J)=NØDE(J,NNN)
              DISTM=DIST/5280.
              SUMTØT=P/60.
              AVGV=DISTM/SUMTØT
              DØ 198 I=1,30
              NN1=NØDEF(I)
              NN2=NØDEF(I+1)
         198  PRINT 199,NØDEF(I),NØDEF(I+1),I,
```

RØUTE CØNTINUED

```
     S(NN1,NN2,I)
     PRINT 197,SUMTØT,DISTM,AVGV
 199 FØRMAT(5H FRØM,I3,3H TØ,I3,7H ØN SEG,I3,3H IN,F9.2
     ,4H MIN)
 197 FØRMAT(///4X,22HTØTAL TIME ØF TRAVERSE,F11.2,6H HØURS/
     4X,24HTØTAL LENGTH ØF TRAVERSE,F8.2,6H MILES/
     4X,25HAVERAGE SPEED ØF TRAVERSE,F8.2,4H MPH////)
     RETURN
     END
```

Subroutine VWRT (Fig C58)

Subroutine VWRT writes the arrays V, VR and TPR into an external file for later manipulation if required. This file is named VELFIL within the program. Later, when the program has been run, this file will be called and renamed to identify it with the specific vehicle.

SUBRØUTINE VWRT

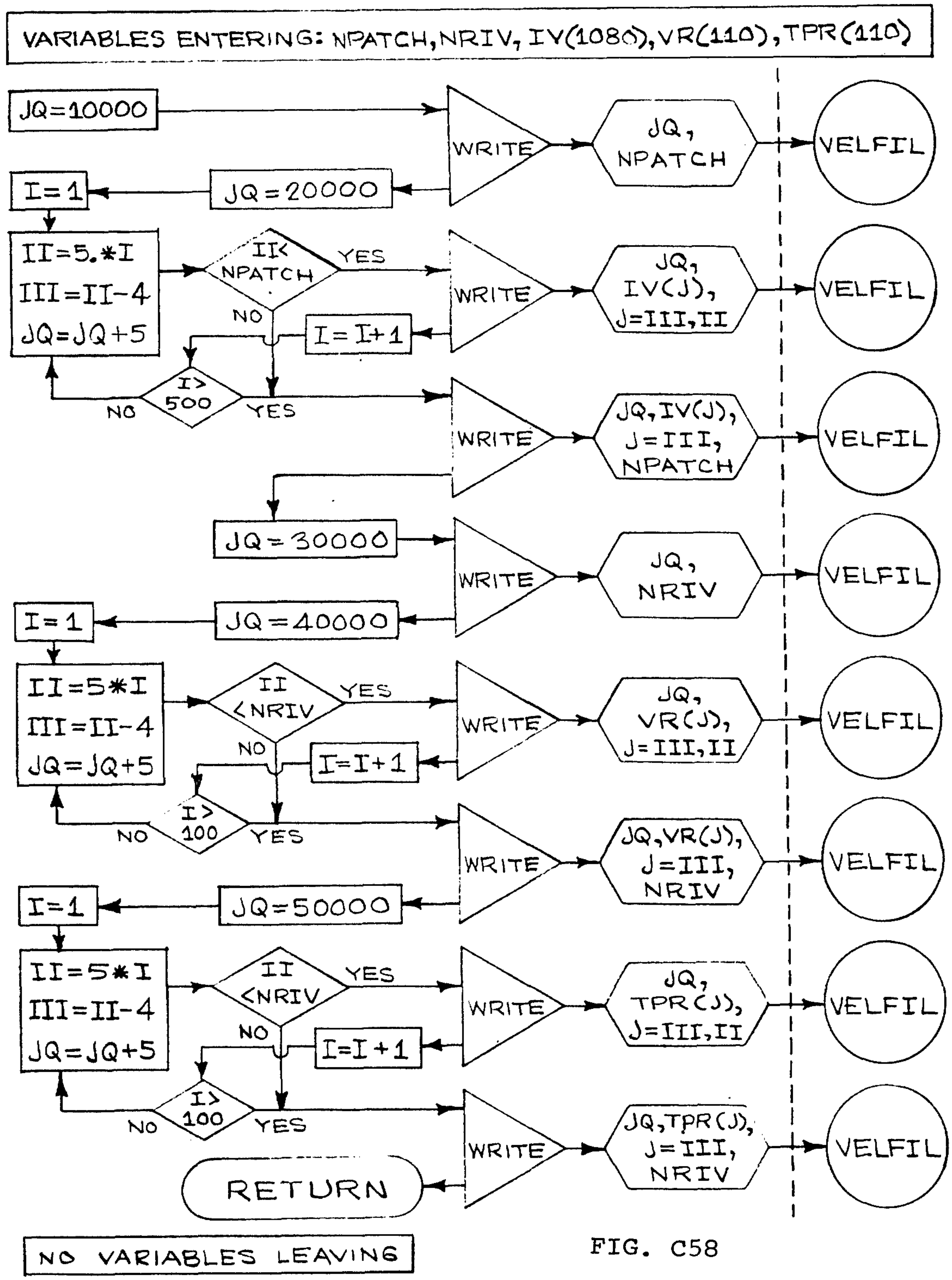


FIG. C58

VWRT

```
      SUBRØUTINE VWRT(NPATCH,NRIV)
      CØMMØN IPATCH(2230),IV(1080),VR(110),TPR(110)
      CALL ØPENF(1,"VELFIL")
      JQ=10000
      WRITE (1;500) JQ,NPATCH
      JQ=20000
      DØ 600 I=1,500
      II=5*I
      III=II-4
      JQ=JQ+5
      IF(II.GE.NPATCH)GØTØ 601
  600 WRITE(1;502) JQ,(IV(J),J=III,II)
  601 WRITE(1;502) JQ,(IV(J),J=III,NPATCH)
      JQ=30000
      WRITE (1;500) JQ,NRIV
      JQ=40000
      DØ 605 I=1,100
      II=5*I
      III=II-4
      JQ=JQ+5
      IF(II.GE.NRIV)GØTØ 606
  605 WRITE (1;501) JQ,(VR(J),J=III,II)
  606 WRITE (1;501) JQ,(VR(J),J=III,NRIV)
      JQ=50000
      DØ 610 I=1,100
      II=5*I
      III=II-4
      JQ=JQ+5
      IF(II.GE.NRIV)GØTØ 611
  610 WRITD (1;501) JQ,(TPR(J),J=III,II)
  611 WRITE (1;501) JQ,(TPR(J),J=III,NRIV)
  500 FØRMAT(I5,1X,I5)
  501 FØRMAT(I5,1X,5F13.6)
  502 FØRMAT(I5,1X,5I10)
      CALL CLØSEF(1)
      RETURN
      END
```

DATA FILES

There are five terrain data files and four vehicle data files. Two of the terrain data files contain information regarding the patches and rivers, respectively; and three map data files contain information regarding the location of the patches in the terrain, the location of marshes and the location of rivers.

The first data file, PCHPUE, contains the data identifying the terrain characteristics of each of 1061 normal patches and 19 marshes for the Puerto Rico site. The data are in the same format for both. The variables read in order are: NPAT, the patch number; IST, the soil type class (either fine-grained or coarse-grained); IRCI(I), the RCI values for the three seasons, I being equal to 1, 2, or 3 for dry, average and wet; IGR, the grade class; IØBAA, the obstacle approach angle class; IØBH, the obstacle height class; IØBW, the obstacle width class; IØBL, the obstacle length class; IØBS, the obstacle spacing class; IØST, the obstacle spacing type class (either linear or random); IPR, the microprofile class; IS(I), which contains the stem spacing class for stem diameter class I (I = 1 to the number of stem diameter classes); and IREC, the recognition distance class. In the case of marshes (which contain no obstacles), variable IØBS is used to identify the water depth class. The second file, HZØPUE, contains river data. Data read from this file in order are: NPAT, the river type number; NISC, the river ingress bank angle class; NBDC, the river bank differential height class; NESC, the river egress bank angle class; NRWC, the river width class; NWDC, the water depth class; and NWV, the water speed class. These two files are read early in the program, and the data are used to calculate velocities in patches and time penalties in rivers.

The other three terrain data files are used late in the program in the route selection mode. The first, SECPUE, contains information about the patches encountered along each path-segment of the map. The first datum read is N, the number of patches encountered, followed by N pairs of data consisting of variable DP(I), the distance across the patch, and IP(I), the patch type number. The next file, MSHPUE, contains information regarding the marshes encountered in each path-segment. The data are in the same format as in SEGPR1. The last terrain data file, RIVPUE, contains information regarding the river types encountered on each path-segment. The first datum is variable N, containing the number of rivers encountered, followed by N river type numbers, IP(I).

The first file, PCHPUE, is read from the main program. The second file, HZØPUE, is called from the main program. The last three files - SECPUE, MSHPUE and RIVPUE - are called from subroutine RØUTE.

In addition to these terrain files, there are four vehicle data files entitled: M151, M35A2M, M60A1 and M113A1. One of them, determined by the operator, is called early in the main program. There are 55 variables in each data file. In order, they are: NVEH, vehicle type (0 for tracked, 1 for 4x4 wheeled, and 2 for 6x6 wheeled); ITRAN, transmission type; GVW, gross vehicle weight; DL, length of track on the ground for tracked vehicle, or wheel diameter for wheeled vehicle; WID, width of the wheel or track; GT, grouser height for tracked vehicle, or the number of tires for wheeled vehicle; A, area of one track shoe for tracked vehicle, or the number of axles for wheeled vehicle; HBT, rated horsepower per ton; GC, ground clearance; NBC, number of bogies for tracked vehicle, or presence of chains for a wheeled vehicle; ITVAR, transmission variety; TL, wheel base; FEC, front-end clearance; VAA, vehicle approach angle; REC, rear-end clearance; VDA, vehicle departure angle; CGF, horizontal distance from the center of gravity (CG) to the center line of the front wheel; CGH, vertical distance from the CG to the wheel center line; GWS, greatest span between adjacent wheel center lines; RR, rolling radius for a wheel, or radius of the road wheel plus track thickness for a track; ACG, angle between a line parallel to the ground and a line between the CG and the center of the rear wheel, used to determine departure angle; DCG, distance from the CG to the center of rear wheel; HC, height of the center of rear wheel above the ground; RWW, rolling radius of rear wheel; HS, maximum step height the vehicle can manage; WC, winch capacity; SAI, ingress swamp angle; AWPKF, auxiliary water propulsion factor; GCA, ground contact area; FD, fording depth; VSS, vehicle swimming speed; VFS, vehicle fording speed; NCREW, number of members in the crew; and NFL, track type. If the vehicle is wheeled, the following three are read: RDIAM, wheel rim diameter; TPSI, tire pressure; and TPLY, tire ply rating. For all vehicles, the following are read: XBR, braking force a vehicle can generate; W, the vehicle width; PBHT, vehicle pushbar height; PBF, force the vehicle pushbar can withstand; and VL, length of the vehicle.

Next in order is NC4, the number of points in array VØØB; array VØØB contains obstacle height versus vehicle velocity at 2.5-g vertical acceleration. Then array VØØB is read, followed by NC5, the number of points in array VRIDE, which contains velocity limited by surface roughness at 6-watts absorbed power for various surface roughness classes. Then array VRIDE is read.

Next read are: RR, sprocket pitch radius for a track, or tire rolling radius for a wheel; FDR, final drive ratio; EFF, transmission efficiency; FDREF, final drive efficiency; and NG, number of gear ratios in the transmission. This is followed by GR(I), and array containing these gear ratios. If the transmission is automatic, the following variables are read: TC, the input torque at which the torque converter curves were measured; ENTCG, gear ratio between the engine and torque converter; LØKUP, denoting the presence of a torque converter lockup; and NC1, the number of points in array TNE1. Then, array TNE1, which contains the torque converter input speed versus converter speed ratio curve, is read. Next read is NC2, the number of points in array TTM, followed by the reading of array TTM, which contains the torque converter torque multiplying coefficient versus converter speed ratio curve. For all vehicles, the following are read: NC3, the number of points in array TTE; then array TTE, which contains the net engine torque versus engine speed curve. These variables are all read for subroutine INPUT, which is called from the main program.

Since these files are very large, it is not desirable to reproduce them here. Anyone desiring these files should contact the authors for punched-tape or cards.

VEHICLE RIDE DYNAMICS SUBMODEL:

The rest of Appendix C presents a synopsis of the Vehicle Ride Dynamics Model used to determine speed as limited by shock and vibration. The vehicle ride dynamics model requires specific terrain and vehicle factors as input and yields as outputs the motions at various parts of the vehicle that allow for determination of the limiting speeds due to shock and vibration in terms of response limits and specific terrain attributes. These limiting speed-terrain attribute relations form inputs to the main program in the form of an array of coordinates and serve as catalogs for shock-limiting and ride-limiting speeds.

The surface geometry features that affect vehicle ride vary from a discrete single perturbation, such as a boulder, rice dike, or log, to gentle undulations as appear in the surface of an open level pasture, and produce effects on vehicles that range from shock to vibration to immobilization, depending on the speed and size of the vehicle in relation to the size and spacing of the surface features. These surface irregularities can be conveniently divided into two basic types that can be associated with either the steady-state or transient solutions of a mechanical system. The first is the type of surface undulations that produce a relatively uniform vibrational activity and is sometimes referred to as stable ground roughness. The second type of activity is the response to singular obstacles. The ride dynamics model is used to predict vehicle responses to both singular obstacles and stable ground roughness and establish meaningful relations among vehicle response, vehicle speed and terrain features. The prime objective of the ride dynamics model with regard to the analytical model is to establish relations between limiting speed and a measure of the pertinent terrain features. Because of the nature of the responses, the singular obstacle problem is treated separately from that of stable ground roughness.

A major problem encountered in determining limiting speed versus terrain feature relations is that of first defining meaningful quantities to describe the terrain features and vibration response level.

Results of past studies have indicated that for the singular obstacle problem, obstacle height is a simple, straightforward, and suitably adequate descriptor for the terrain feature. For describing stable ground roughness, a statistical classification (power spectral density, PSD) that yields information about the amplitude and frequency content of a surface was chosen. It is believed that eventually geographic regions can be suitably related to specific characteristic PSD curves; however, due to the lack of such terrain information at this time, no attention is given to the PSD curves as such. Each terrain profile is assumed to exhibit the same characteristic PSD of the form $p(\Omega) = k\Omega^{-2}$ and only a roughness index (rms elevation) is used to describe the condition of the terrain surface.

The prime response criteria for limiting vehicle speed is that level at which the driver's vertical acceleration reaches 2.5 g (for singular obstacles) or the driver's absorbed power reaches a sustained level of 6 watts (for stable ground roughness).

Vehicle Models

The vehicles in the ride dynamics model are represented in the form of coupled, second-order differential equations that describe the motions of each degree of freedom. The equations derive naturally by applying Newton's second law to the mass-spring-damper elements representing the vehicle's components. The elements comprising the vibratory systems are idealized in the usual sense in that the mass elements are assumed to be rigid bodies, the spring elements are assumed to be of a negligible mass and represent the elastic properties of the structure, and the damping elements have neither mass nor elasticity and represent the dissipative forces or energy losses of the system. Damping forces exist only if there is relative motion between the two ends of the damper. The two types of damping most common to vehicle suspensions are (a) frictional (Coulomb) damping, which is a function of the normal force between the sliding bodies, as well as the materials involved, and is found chiefly in leaf-spring action; and (b) viscous damping, in which the damping force is proportional to the velocity as occurs in shock absorbers.

Although a vehicle is a very complicated vibrational system possessing a number of degrees of freedom, for many types of problems certain motions are unimportant. As a result of a compromise between model complexity, adequate description of the significant motions, and time and cost of computer simulations, two-dimensional models were used to represent the vehicles.

Schematic diagrams showing the manner in which the mass-spring-damper elements are arranged to represent the two wheeled and the two tracked vehicles used in this study are shown respectively in Figures C59 to C62. In Figure C59, a 4-degree of freedom model of the M151 jeep that could actually represent almost any conventional four-wheeled vehicle is shown.

The frame is considered rigid and only the pneumatic tires and suspension elements are considered to contribute to the sprung motion of the frame. All the pertinent non-linearities of the suspension, including jounce and rebound limits, are determined from appropriate measurements and are included in the action of the spring and damper elements.

A schematic of the M35, 2½ ton truck that portrays the vehicle as four mass points -- the axle-wheel assemblies and the body -- is shown in Figure C60. The unsprung mass (axle-wheel assemblies) is connected to the sprung mass (vehicle body) through the springs and dampers representing the suspension compliance. The "walking beam" bogie assembly in the rear is composed of the masses of the two axle-wheel assemblies connected by spring and dampers to a rigid massless bar that is free to pivot about a frictionless point that connects the assembly to the main frame. The tire compliance for both wheeled vehicles is represented by a cluster of radially projecting springs.

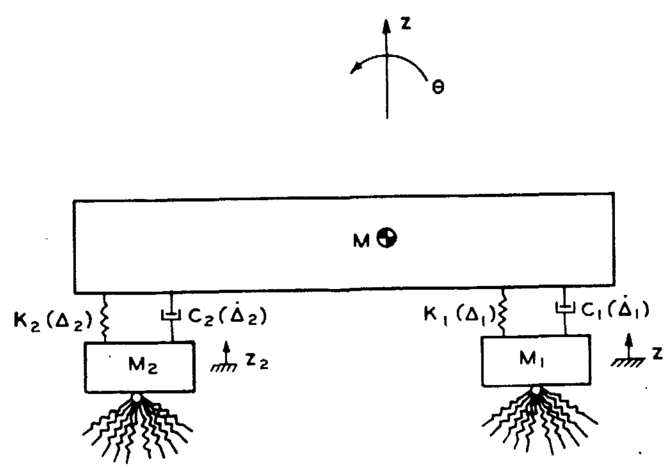


Fig. C59. M151 truck

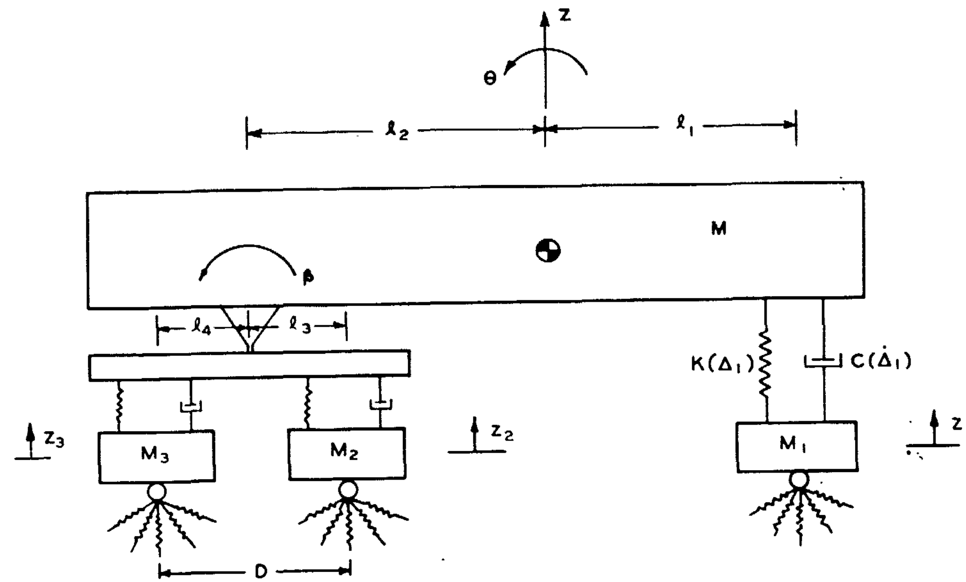


Fig. C60. M35, 2 1/2-ton truck

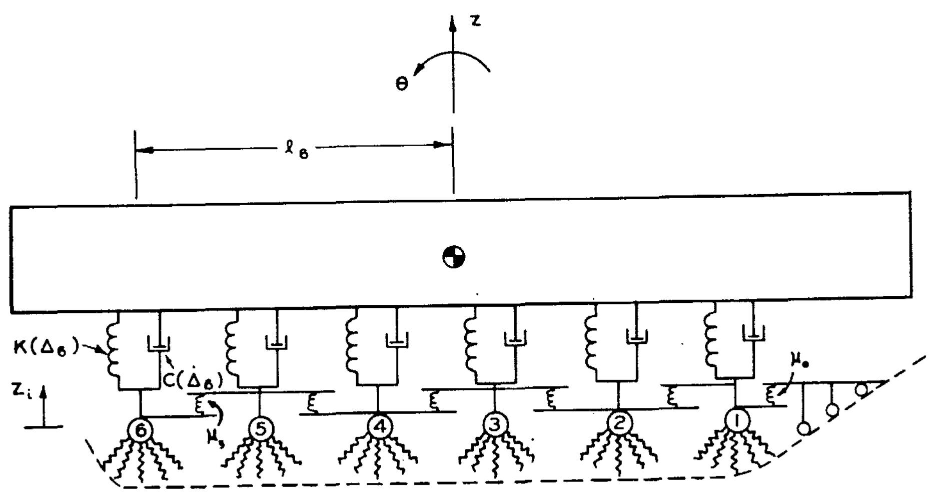


Fig. C61. M60A1 tank

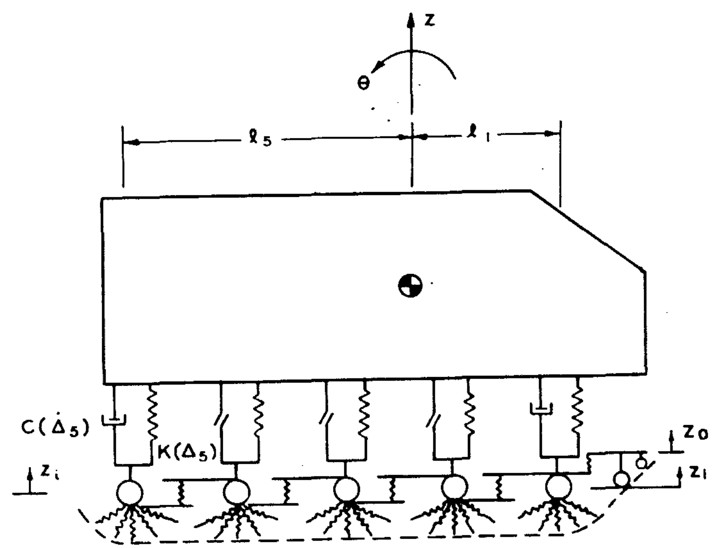


Fig. C62. M113 armored personnel carrier

Schematics of the M60 tank and the M113 armored personnel carrier are presented in Figures C61 and C62, respectively. The structures of these two models are similar, except that the M113 has one less bogie than the M60. These models consist of eight and seven degrees of freedom, respectively, which include the bounce and pitch of the main frame and the vertical motions of each of the bogie wheels. The longitudinal motion is accounted for only in the acceleration determined from the horizontal forces resulting from deflections of the bogie spring segments. No attempt is made to simulate the horizontal motions from a fixed reference. This method of accounting for horizontal accelerations is analogous to supplying the additional traction necessary to balance the resisting forces due to spring deflection, thus enabling the tank to maintain a constant velocity while crossing an obstacle. This additional force required to maintain this constant velocity is then used to determine the longitudinal acceleration.

Here, as in all cases where the driver is located away from the C. G., the motions at the driver compartment are computed. The geometry effects of the bogies are represented by radially projecting stiff springs, and the track compliance by interconnecting springs between the bogies and massless "feelers" appropriately positioned to portray the geometry of the leading portion of the track in front of the first bogie.

The construction of each vehicle model requires the specific values for masses, inertias, suspension, and tire or track compliance and the appropriate dimensions relative to the centers of gravity.

Development of Equations

The differential equations describing the motions of each degree of freedom were developed for each vehicle by first establishing an appropriate set of coordinates and sign convention and then placing each system in a displaced configuration such that each coordinate was non-zero. The relative displacements of the masses cause compressions and extensions in the springs and relative motion of the damper ends that produce forces on each mass, as represented by the free-body diagram for the M60 tank in Figure C63.

EQUATIONS FOR M60A1:

Using Newton's second law of motion and summing first the vertical and longitudinal forces and moments on the main frame and then the vertical forces on each bogie led to the series of equations listed below to describe the M60 tank.

M60 Tank Equations:

a. Forces and moments on main frame

(1) Sum of vertical forces

$$M\ddot{z} = -\left[\sum_{i=1}^{6} k(\Delta_i)\Delta_i + \sum_{i=1}^{6} c(\dot{\Delta}_i)\dot{\Delta}_i + Mg\right]$$

(2) Sum of moments

$$I\ddot{\Theta} = -\left[\sum_{i=1}^{3} k(\Delta_i)\Delta_i \ell_i \cos\Theta + \sum_{i=1}^{3} c(\dot{\Delta}_i)\dot{\Delta}_i \ell_i \cos\Theta - \sum_{i=4}^{6} k(\Delta_i)\Delta_i \ell_i \cos\Theta - \sum_{i=4}^{6} c(\dot{\Delta}_i)\dot{\Delta}_i \ell_i \cos\Theta\right]$$

(3) Sum of horizontal forces

$$M\ddot{x} = \sum_{i=1}^{6} H_i$$

b. Vertical forces on bogies

$$M_1\ddot{z}_1 = k(\Delta_1)\Delta_1 + c(\dot{\Delta}_1)\dot{\Delta}_1 - \mu_o\delta_o + \mu_1\delta_1 - M_1 g + V_1$$

$$M_2\ddot{z}_2 = k(\Delta_2)\Delta_2 + c(\dot{\Delta}_2)\dot{\Delta}_2 - \mu_1\delta_1 + \mu_2\delta_2 - M_2 g + V_2$$

$$M_3\ddot{z}_3 = k(\Delta_3)\Delta_3 + c(\dot{\Delta}_3)\dot{\Delta}_3 - \mu_2\delta_2 + \mu_3\delta_3 - M_3 g + V_3$$

$$M_4\ddot{z}_4 = k(\Delta_4)\Delta_4 + c(\dot{\Delta}_4)\dot{\Delta}_4 - \mu_3\delta_3 + \mu_4\delta_4 - M_4 g + V_4$$

This Page omitted
(C-199)

$$M_5 \ddot{z}_5 = k(\Delta_5)\Delta_5 + c(\dot{\Delta}_5)\dot{\Delta}_5 - \mu_4 \delta_4 + \mu_5 \delta_5 - M_5 g + V_5$$

$$M_6 \ddot{z}_6 = k(\Delta_6)\Delta_6 + c(\dot{\Delta}_6)\dot{\Delta}_6 - \mu_5 \delta_5 - M_6 g + V_6$$

where (for all the above equations)

M, M_i = mass of main frame and i[th] bogie assembly, respectively

$\ddot{z}, \dot{z}, z$ = vertical motions of center of gravity of main frame, i.e. acceleration, velocity, and displacement, respectively

$\ddot{z}_i, \dot{z}_i, z_i$ = vertical motions at center of gravity of the i[th] bogie, i.e. acceleration, velocity, and displacement, respectively

$\ddot{\theta}, \dot{\theta}, \theta$ = angular motion about the center of gravity of the main frame

$\ddot{x}$ = horizontal acceleration at center of gravity of main frame

Δ_i $= z + \ell_i \sin\theta - z_i$ for $1 \leq i \leq 3$

$= z - \ell_i \sin\theta - z_i$ for $4 \leq i \leq 6$

$\dot{\Delta}_i$ $= \dot{z} + \ell_i \dot{\theta} \cos\theta - \dot{z}_i$ for $1 \leq i \leq 3$

$= \dot{z} - \ell_i \dot{\theta} \cos\theta - \dot{z}_i$ for $4 \leq i \leq 6$

ℓ_i = distance from center of gravity of main frame to contact point of i[th] bogie

$k(\Delta_i)$ = force-deflection relation for i[th] bogie suspension

$C(\dot{\Delta}_i)$ = force-velocity relation for i^{th} bogie suspension

g = acceleration of gravity

I = pitch moment of inertia of main frame

Q = moment about the center of gravity of main frame produced by horizontal forces

H_i = resultant horizontal force of spring segments of i^{th} bogie

μ_i = spring constant for i^{th} track spring; in this study, $\mu_o = 600$ lb/in, $\mu_i = 375$ lb/in for $1 \leq i \leq 5$

δ_i = $Z_{i+1} - Z_i$ = relative displacement between adjacent bogies

V_i = resultant vertical force of spring segments of i^{th} bogie

A representative force-deflection and a force-velocity relation for describing suspension compliance is shown in Figures C64 and C65, respectively. Photographs of the tank in different attitudes while crossing an 18-inch high obstacle indicated that the greatest pitch expected for this study would probably be in the order of 9 degrees or less. It is seen that if

$$\theta = 9^\circ, \text{ then } \cos 9^\circ = .988 \approx 1.$$

Also, $9^\circ = 9\pi/180 = .157$ radians

and, $\sin 9^\circ = .156 \approx .157$

Based on these values, the small angle assumption, i.e. $\cos\theta = 1$ $\sin\theta = \theta$, can be employed with less than 2 percent

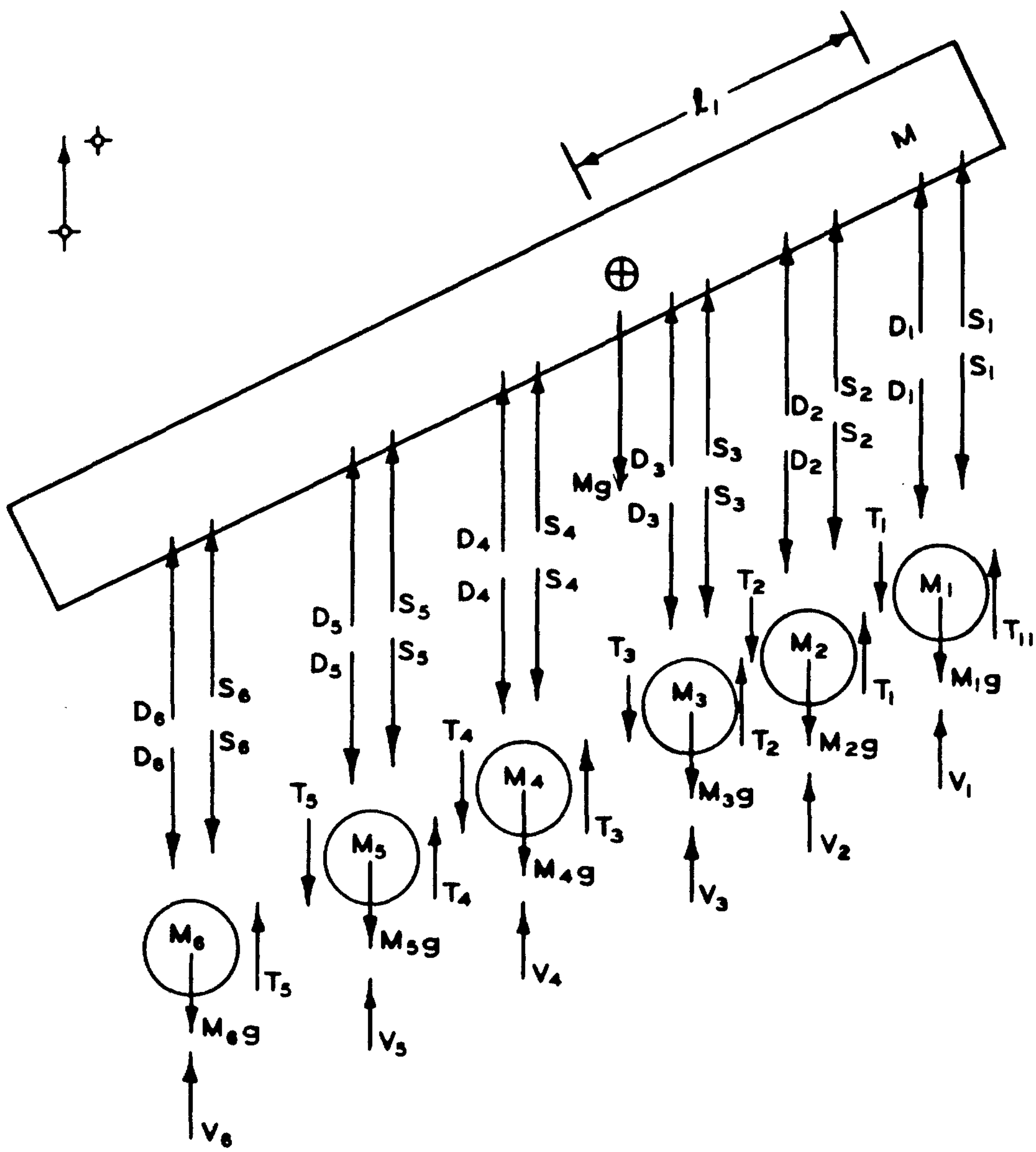


Fig. C63. Vertical Forces Acting on Tank Free Body error. Therefore, to simplify the calculations, the small-angle concept was used in the equations.

Once the motions at the C.G. of the main frame have been determined, the motions in the vicinity of the driver can be found in the usual manner by combining the translational and rotational motions.

Computation of track forces

The track compliance is represented chiefly by interconnecting linear springs between the bogies and massless "feelers" that are connected to the front bogie by a stiff spring. The spring constants portraying the track tension were determined by observing photographs of the vehicles in different positions on an obstacle (Figure C66). From these photos, the influence of displacing a particular bogie on the displacement of the adjacent bogies was estimated. With the approximate mass of each bogie assembly and their displacements relative to each other and the main frame known, an appropriate spring constant could be determined as follows (refer to Figure C63A):

General Equation: $F = K\Delta$

where,

F = applied force

Δ = spring deflection

K = spring constant

$F = F_s + M_i g$

where,

F = total applied force on bogie

F_s = resultant force due to suspension reaction

$M_i g$ = force due to weight of i^{th} bogie

Track-spring constant $K = \frac{F}{\Delta} = \frac{F_s + M_i g}{\Delta}$

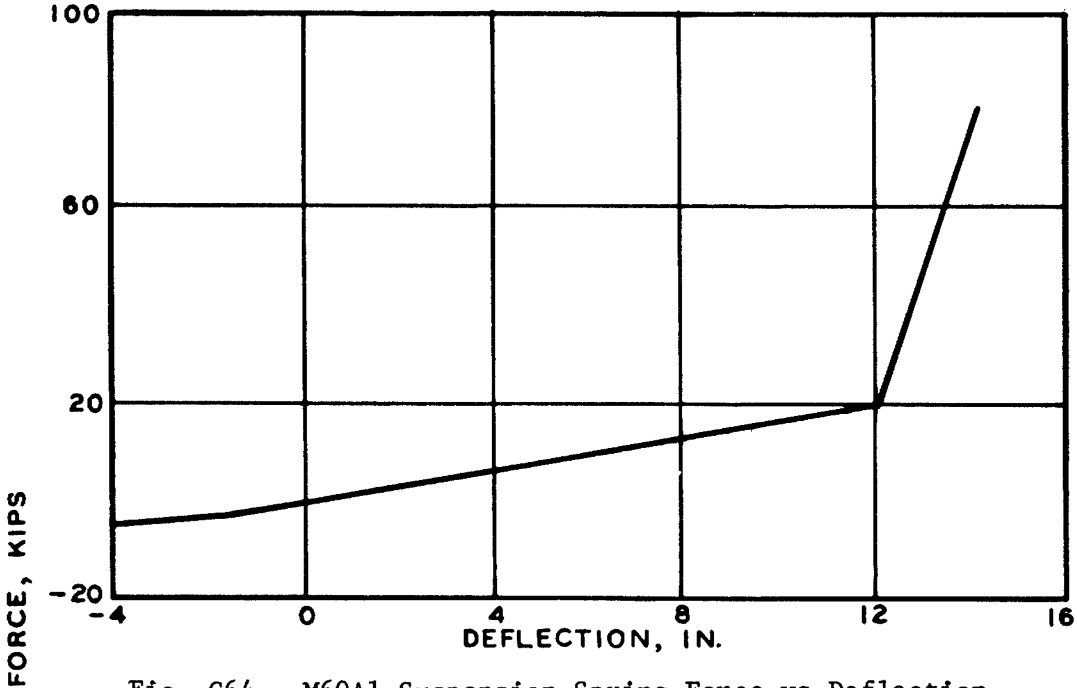


Fig. C64. M60A1 Suspension Spring Force vs Deflection

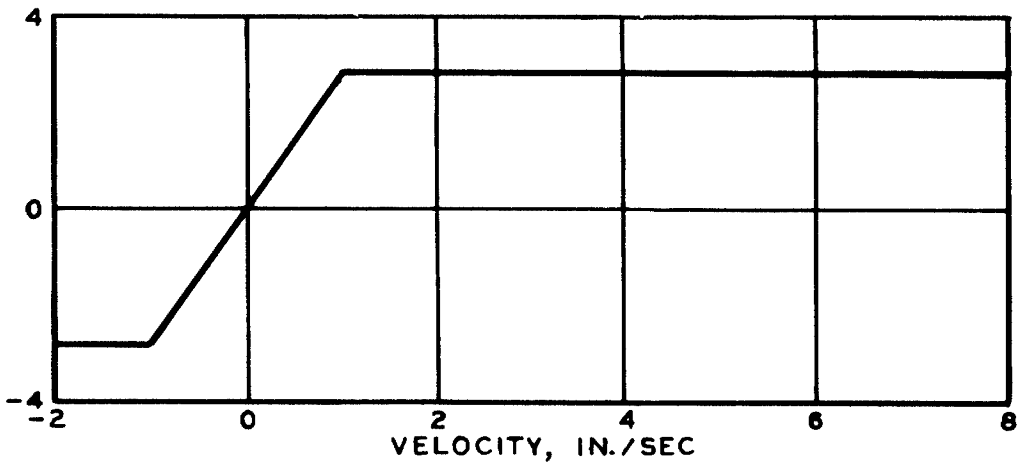


Fig. C65. M60A1 Suspension Damping Force vs Velocity

a. 12-in.-high obstacle; 5 mph

b. 18-in.-high obstacle; 2 mph

Fig. C66. Relative displacements of bogies on M60A1 for computing track spring constants where Δ is relative displacement between adjacent bogies.

NOTE: The track spring is allowed to exert only positive forces, i.e., it cannot exert a downward pull on the bogies.

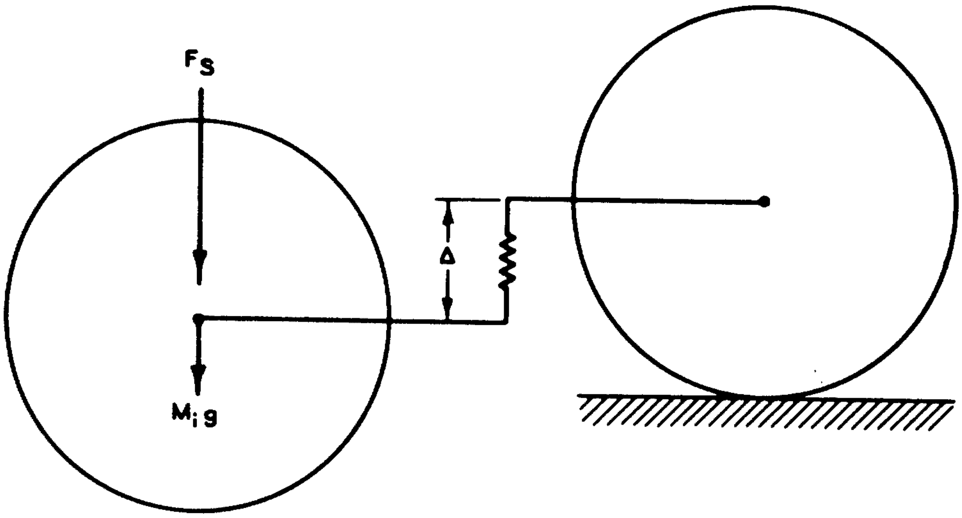


Fig. C63A. Schematic for Use in Determining Track-Spring Constant

Close observation further revealed that for both the M60 and the M113 approaching an obstacle larger than about 6 inches, the initial track-obstacle contact tended to lift the front bogie up and guide it over the obstacle. This lifting has a significant effect on the longitudinal motion. To simulate this effect, massless points were positioned in front of the first bogie each at a different threshold height, to conform to the geometry of the leading portion of the track. The influence of the points in lifting the front bogie depends on the height and shape of the encountered obstacle. At the time of this study, no information was available to enable the determination of an effective spring constant, and an arbitrary value of 600 lb/in. was chosen. This arbitrary value makes the simulation of the longitudinal acceleration the weak point in the system, but since the vibration limits, which were the chief concern, generally occur first in the vertical mode, the longitudinal motions were not of interest in this particular study. However, it is noted that with proper determination of the leading spring constant the longitudinal accelerations could be adequately simulated.

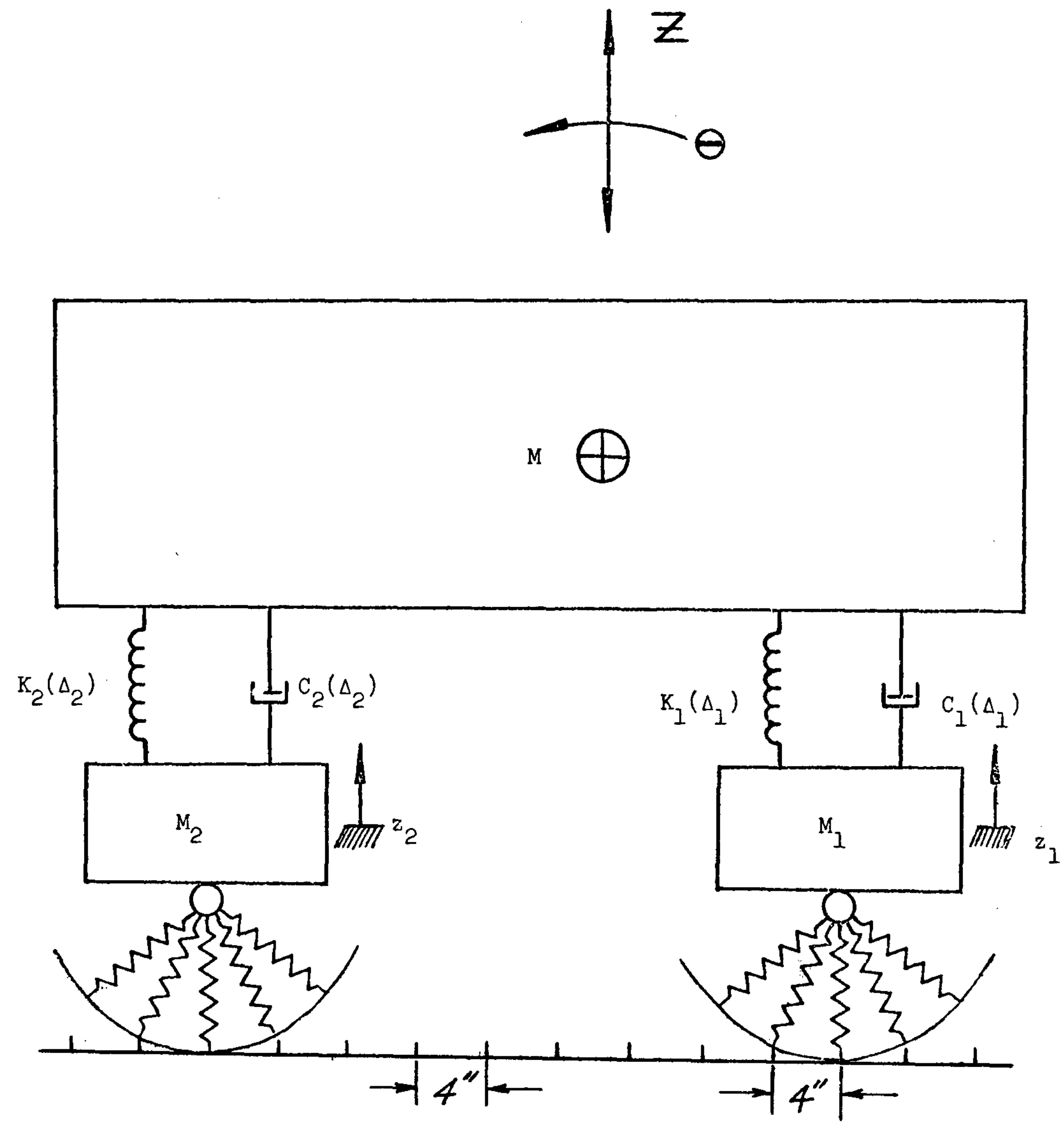


Fig. C67. Schematic Drawing of Truck Model Illustrating Spacing of Profile-Tire Segment Intervals

Tire and bogie spring segments

The segmented wheel concept* was used in both the wheel and track models to (a) provide the flexibility for predicting longitudinal accelerations (if needed), (b) include the important effects of the tire and bogie geometry, and (c) incorporate a means for accounting for the effects of the envelopment characteristics of tires and tracks. Each tire and bogie was divided into an appropriate number of segments with the segments spaced so that their peripheral projections onto the horizontal plane would be spaced at 4-inch intervals (Figure C67).

As a result, all profile points were interpolated to 4-inch spacing prior to input to the models. To account for track thickness 2 inches were added to the radii of each bogie. The spring constants for the pneumatic tire models were obtained by first measuring the load-deflection relation for each tire at the desired inflation pressure for the purpose of selecting a characteristic load-deflection coordinate. For example, the load-deflection relation for the 11.00-20, 12-PR tire at 20 psi inflation pressure (Figure C68) was such that an applied centerline deflection of 1.35 inches produced a force of 3,000 pounds. At this deflection, three spring segments are influenced (Figure C69). The spring constant can be computed from the statics equation:

$$F = \sum_{i=1}^{9} K \cos\phi_i \, \delta_i$$

where, δ_i is the deflection of the i[th] spring defined as:

$$\delta_i = Y_i - T_i - Z \quad \text{for} \quad Y_i - T_i - Z \geq 0$$

$$\delta_i = 0 \quad \text{for} \quad Y_i - T_i - Z < 0$$

---

*Lessem, A.S., "Dynamics of Wheeled Vehicles; A Mathematical Model for the Traversal of Rigid Obstacles by a Pneumatic Tire," Technical Report M-68-1, Report 1, May 1968, U.S. Army Engineer Waterways Experiment Station, CE, Vicksburg, Miss.

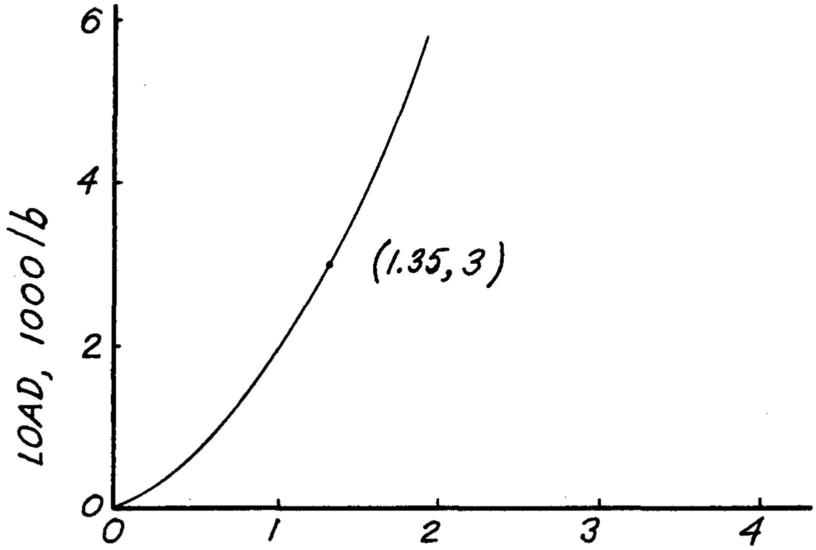


Fig. C68

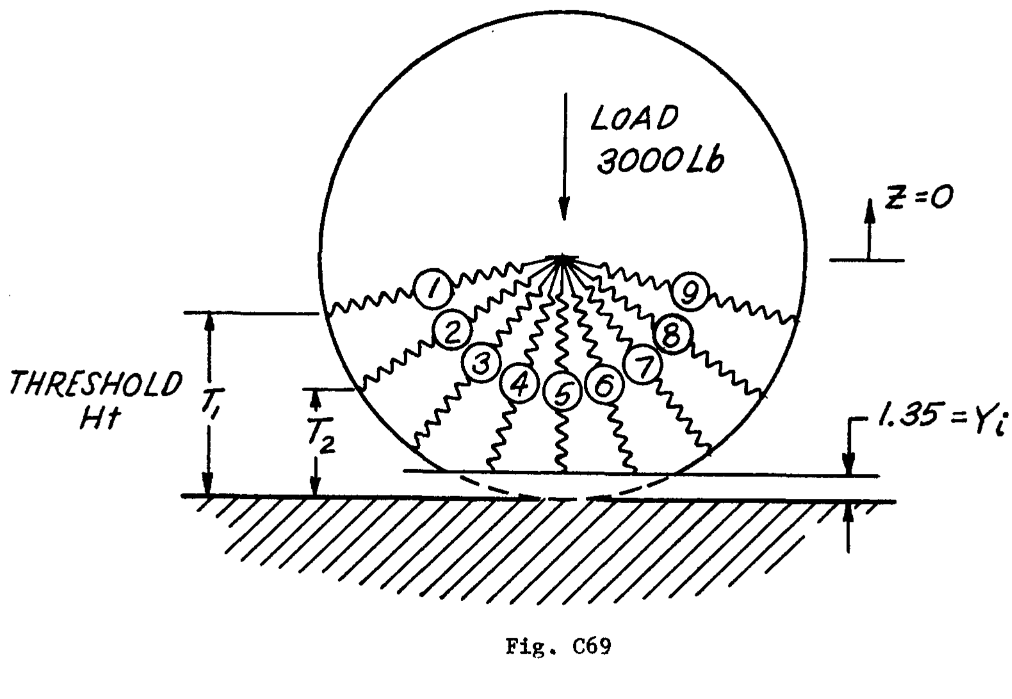


Fig. C69

Y_i = vertical height of terrain profile beneath i^{th} segment. For this case, Y_i = 1.35" = a constant

Z = vertical displacement of axle, in this case Z = 0

T_i = height from the zero reference to the i^{th} spring of the undeflected wheel (see Figure C69)

ϕ_i = angle to the i^{th} bogie measured from a vertical ray through wheel center

For this case, and due to the symmetry of the segments about the center line, the equation reduces to:

$$3000 = K\left[1.35 \cos 0^{o} + 2\ (1.02 \cos 12.5^{o})\right]$$

where the effective radial deflections are:

$\delta_5 = 1.35$ in.

$\delta_4 = \delta_6 = 1.02$ in.

Solving for K yields:

K = 900 lb/in.

Defining GAMMA = k cos ϕ_i = 900 cos ϕ_i yields the following relations for the segments of the front and rear wheels:

Front Wheel		Rear Wheel		
GAMMA (1)	= GAMMA (9)	= GAMMA (10)	= GAMMA (18)	= 581
GAMMA (2)	= GAMMA (8)	= GAMMA (11)	= GAMMA (17)	= 716
GAMMA (3)	= GAMMA (7)	= GAMMA (12)	= GAMMA (16)	= 817
GAMMA (4)	= GAMMA (6)	= GAMMA (13)	= GAMMA (15)	= 878
GAMMA (5) =		GAMMA (14) =		= 900

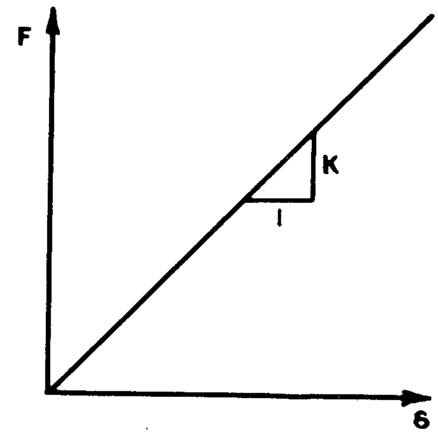


Fig. C70A. Force-Deflection Relation

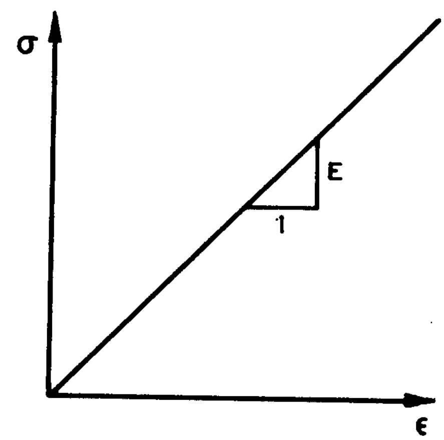


Fig. C70B. Stress-Strain Relation

A similar relation, SIGMA, is defined for the horizontal component, K sin ϕ_i, and is used when calculations of horizontal motions are desired.

The segment deflections are permitted to have positive values only; negative values are replaced by zero. The reference from which vertical displacements are measured is the point that locates the axle when the wheel is imagined to be rigid and in static equilibrium. Static deviations from this reference correspond to static wheel deflections, and superposed on these static deflections are the dynamic obstacle-induced deflections.

The spring constants for the bogie segments were obtained in the following manner. The periphery of the bogies of both the M60 and M113 was encased in a hard, abrasive-resistant rubber shell approximately 2 inches thick. A channel down the center divided this shell into 2 bands each approximately 5 inches wide for the M60 and 2.5 inches wide for the M113. The first step in determining segment constants was to consider the relations between linear force-deflection and stress-strain curves of the type shown in Figure C70. The equation describing force F and deflection δ is:

$$F = K\delta \qquad (1)$$

Likewise, the equation describing stress σ and strain ϵ is:

$$\sigma = E\epsilon \qquad (2)$$

The idea is to obtain a relationship between the constants of proportionality K (spring constant) and E (modulus of elasticity).

Defining $\sigma = \frac{F}{A}$ and $\epsilon = \frac{\Delta L}{L} = \frac{\delta}{L}$, the stress-strain relation (2) can be written as $F = EA \cdot \frac{\delta}{L}$. (3)

Equating the forces in equations (1) and (3) yields the desired relation between K and E.

$$K = \frac{EA}{L} \qquad (4)$$

where K = spring constant in lb/in.

E = modulus of elasticity in psi

A = the area upon which pressure is being applied

L = thickness of rubber casing

The thickness, L, was taken as 2 inches for all the bogies. The effective area was determined to be that portion of the rubber shell beneath the bogie hub upon which pressure was being applied. The areas were computed to be approximately 20 square inches for the M60 and 14 square inches for the M113. A value of 500 psi, obtained from a handbook of material properties for a hard, abrasive-resistant rubber, was used for the modulus of elasticity.

Substituting these values into equation (4) yielded spring constants of 5000 lb/in. and 3500 lb/in. for the M60 and M113 bogies, respectively. The schematics of the bogies in Figure C71 illustrate the manner in which these values were obtained.

No damping was incorporated in the tire or bogie compliance, since in actual vehicles this damping is negligible compared with the damping of the suspensions.

The differential equations describing these vehicles were programmed for solution on a GE-400 series computer by the Runge-Kutta-Gill numerical integration scheme.

Equations for M113, M151 and M35

The equations for the M113 are as follows:

a. Forces and moments on main frame

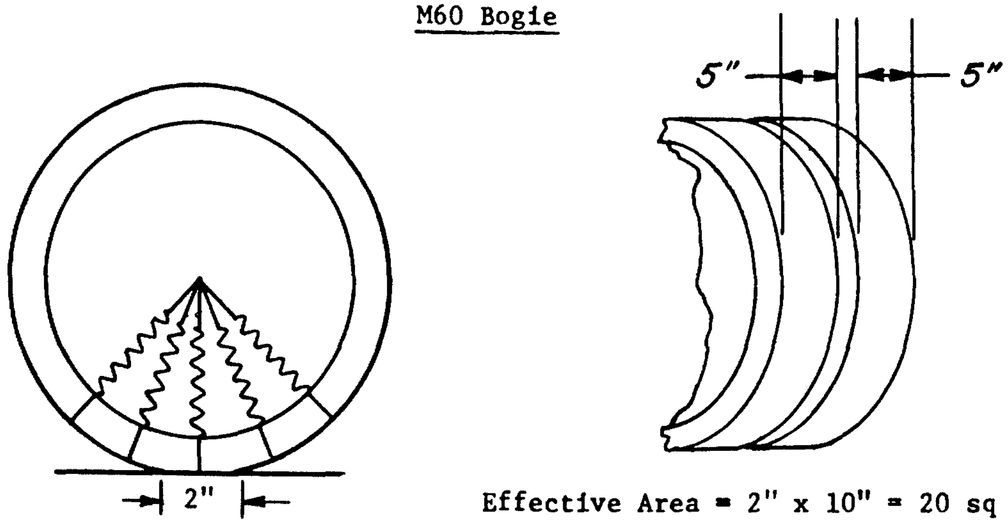
(1) Sum of vertical forces

$$M\ddot{z} = -\left[\sum_{i=1}^{5} k(\Delta_i)\Delta_i + \sum_{i=1}^{5} c(\dot{\Delta}_i)\dot{\Delta}_i + Mg\right]$$

(2) Sum of moments

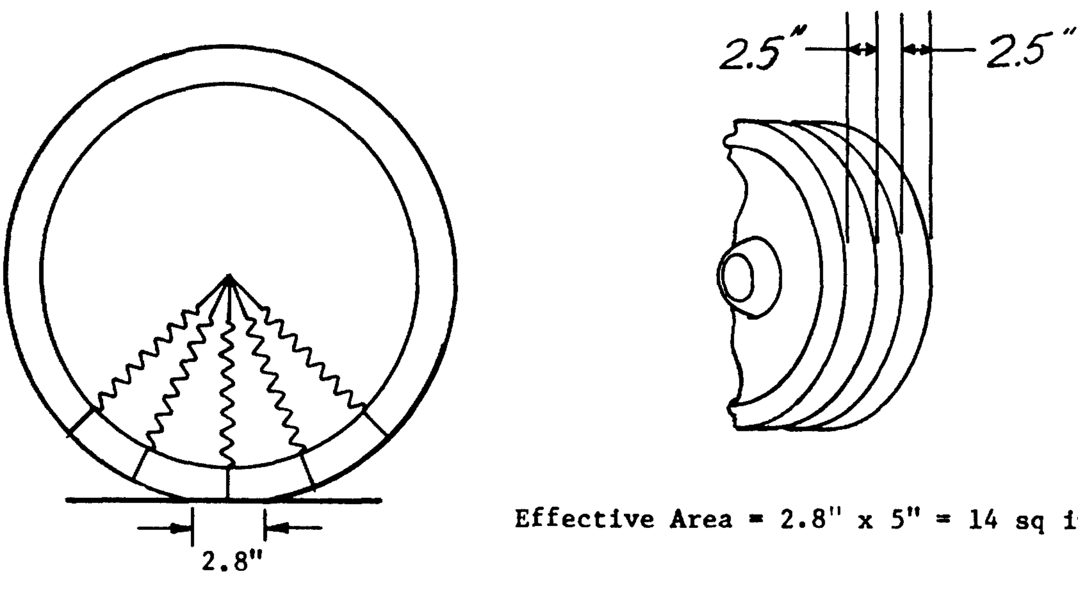
$$I\ddot{\theta} = -\left[\left(\sum_{i=1}^{2} k(\Delta_i)\Delta_i + \sum_{i=1}^{2} c(\dot{\Delta}_i)\dot{\Delta}_i - \sum_{i=3}^{5} k(\Delta_i)\Delta_i - \sum_{i=3}^{5} c(\dot{\Delta}_i)\dot{\Delta}_i\right)\ell_i \cos\theta\right]$$

M60 Bogie



Effective Area = 2" x 10" = 20 sq in.

M113 Bogie



Effective Area = 2.8" x 5" = 14 sq in.

Fig. C71. Schematic Drawings of M60A1 and M113 Bogies Illustrating Effective Areas of Pressure Application

(3) Sum of horizontal forces

$$M\ddot{x} = \sum_{i=1}^{5} H_i$$

b. Vertical forces on bogies

$$M_1\ddot{Z}_1 = k(\Delta_1)\Delta_1 + c(\dot{\Delta}_1)\dot{\Delta}_1 - \mu_o\delta_o + \mu_1\delta_1 - M_1 g + V_1$$

$$M_2\ddot{Z}_2 = k(\Delta_2)\Delta_2 + c(\dot{\Delta}_2)\dot{\Delta}_2 - \mu_1\delta_1 + \mu_2\delta_2 - M_2 g + V_2$$

$$M_3\ddot{Z}_3 = k(\Delta_3)\Delta_3 + c(\dot{\Delta}_3)\dot{\Delta}_3 - \mu_2\delta_2 + \mu_3\delta_3 - M_3 g + V_3$$

$$M_4\ddot{Z}_4 = k(\Delta_4)\Delta_4 + c(\dot{\Delta}_4)\dot{\Delta}_4 - \mu_3\delta_3 + \mu_4\delta_4 - M_4 g + V_4$$

$$M_5\ddot{Z}_5 = k(\Delta_5)\Delta_5 + c(\dot{\Delta}_5)\dot{\Delta}_5 - \mu_4\delta_4 - M_5 g + V_5$$

where for $1 \leq i \leq 2$, $\Delta_i = Z + \ell_i \sin\theta - Z_i$

$$\dot{\Delta}_i = \dot{Z} + \ell_i \dot{\theta}\cos\theta - \dot{Z}_i$$

and for $3 \leq i \leq 5$, $\Delta_i = Z - \ell_i \sin\theta - Z_i$

$$\Delta_i = \dot{Z} - \ell_i \dot{\theta}\cos\theta - \dot{Z}_i$$

The equations for the M151 are as follows:

a. Forces on body

$$M\ddot{Z} = k_1(\Delta_1)\Delta_1 + C_1(\dot{\Delta}_1)\dot{\Delta}_1$$
$$+ k_2(\Delta_2)\Delta_2 + C_2(\dot{\Delta}_2)\dot{\Delta}_2 - Mg$$

$$I\ddot{\theta} = k_1(\Delta_1)a\Delta_1 + c_1(\dot{\Delta}_1)a\dot{\Delta}_1$$
$$- k_2(\Delta_2)b\Delta_2 - c_2(\dot{\Delta}_2)b\dot{\Delta}_2$$

where $\Delta_1 = z_1 - z - a\sin\theta$, $\dot{\Delta}_1 = \dot{z}_1 - \dot{z} - a\dot{\theta}\cos\theta$

$\Delta_2 = z_2 - z + b\sin\theta$, $\dot{\Delta}_2 = \dot{z}_2 - \dot{z} + b\dot{\theta}\cos\theta$

b. Forces on front axle

$$M_1\ddot{z}_1 = -k_1(\Delta_1)\Delta_1 - c_1(\dot{\Delta}_1)\dot{\Delta}_1 - M_1 g + V_1$$

c. Forces on rear axle

$$M_2\ddot{z}_2 = -k_2(\Delta_2)\Delta_2 - c_2(\dot{\Delta}_2)\dot{\Delta}_2 - M_2 g + V_2$$

The equations for the M35 are as follows:

a. Forces and moments on main frame

(1) Sum of vertical forces

$$M\ddot{z} = \sum_{i=1}^{3} k(\Delta_i)\Delta_i + \sum_{i=1}^{3} c(\dot{\Delta}_i)\dot{\Delta}_i - Mg$$

(2) Sum of moments

$$I\ddot{\theta} = [k(\Delta_1)\Delta_1 + c(\dot{\Delta}_1)\dot{\Delta}_1]\ell_1\cos\theta$$
$$-\sum_{i=2}^{3}[k(\Delta_i)\Delta_i + c(\dot{\Delta}_i)\dot{\Delta}_i]\ell_i\cos\theta$$

(3) Sum of forces on i^{th} axle

for i = 1, 2, 3

$$M_i\ddot{z}_i = -k(\Delta_i)\Delta_i - c(\dot{\Delta}_i)\dot{\Delta}_i - M_i g + V_i$$

where (for all the above equations)

$$\Delta_1 = z_1 - (z + \ell_1\sin\theta)$$

$$\Delta_2 = z_2 - (z - \ell_2\sin\theta + \ell_3\sin\beta)$$

$$\Delta_3 = z_3 - (z - \ell_2\sin\theta - \ell_4\sin\beta)$$

$$\dot{\Delta}_1 = \dot{Z}_1 - (\dot{Z} + \ell_1 \dot{\theta} \cos\theta)$$

$$\dot{\Delta}_2 = \dot{Z}_2 - (\dot{Z} - \ell_2 \dot{\theta} \cos\theta + \ell_3 \dot{\beta} \cos\beta)$$

$$\dot{\Delta}_3 = \dot{Z}_3 - (\dot{Z} - \ell_2 \dot{\theta} \cos\dot{\theta} - \ell_4 \dot{\beta} \cos\beta)$$

$$\beta = \arctan(Z_2 - Z_3)/D$$

V_i = vertical force on i^{th} wheel due to wheel segments

Generation of Random Profiles

An essential requirement for this study was the capability to generate artificial terrain profiles composed of random, normally distributed amplitudes and to have control of the frequency content and pertinent statistics. A computer program generated these low-pass, Gaussian profiles with a zero mean and specified rms by the following procedures: A random number chain, generated by the power residue method, was entered at an arbitrary starting point, and 12 uniform, random numbers were computed and summed. By the central limit theorem, a single random, normal number was then computed by the formula:

$$V = (A - 6) \quad (\sigma_n)$$

where

V = the random, normal number

A = the sum of 12 uniform, random numbers

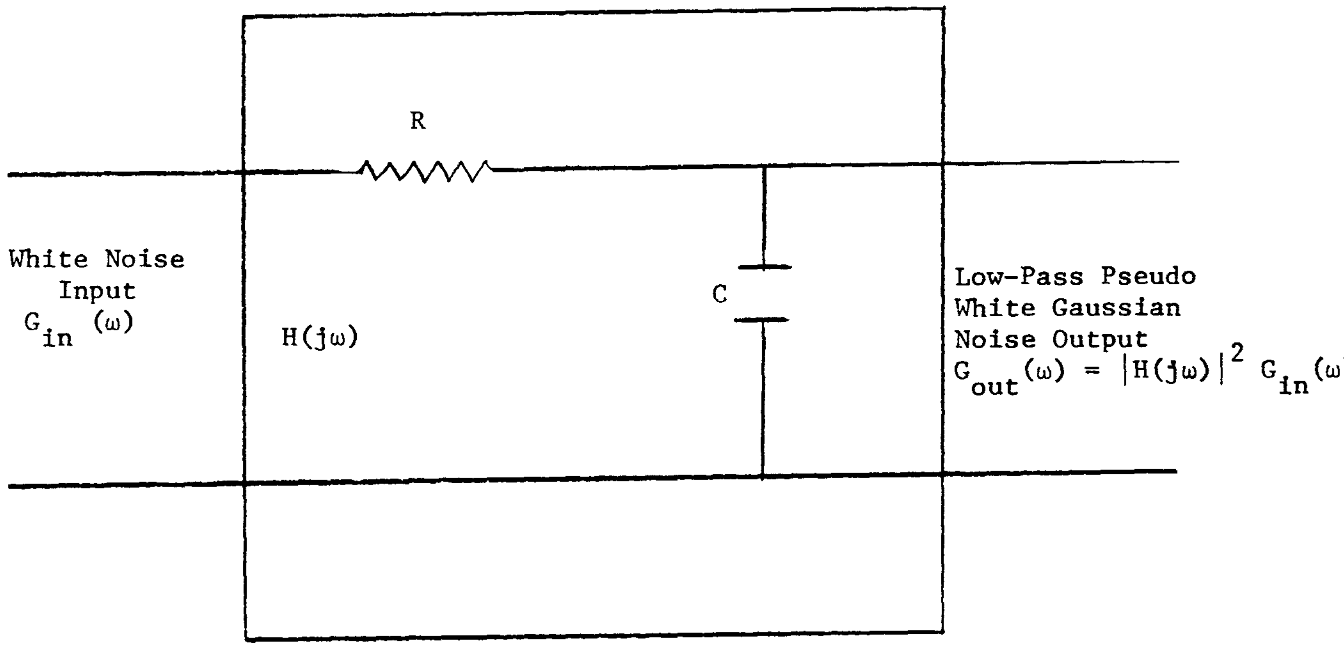


Fig. C72. Low-Pass RC Filter

σ_n = the standard deviation desired for the sequence of random, normal numbers

This technique for computing random, normal numbers is often referred to as the "sum of the uniform deviates" method.

The calculations were repeated until a sequence of 300 random, normal numbers was obtained. Although the mean of the resulting sequence was always very nearly zero, a shifting operation was performed to insure a mean value of absolute zero. The frequency distribution of the sequence at this point approximates that of white Gaussian noise in that its with an average mean square level of σ_n^2. To obtain the desired spectrum, the sequence was then passed through a numerical system that simulated an analog low-pass filter with a certain "time constant" T = RC and cut-off frequency α. The analog equivalent of this system is shown in Figure C72. This system is formulated numerically by the following formula:

$$Y_{i+1} = X_{i+1}\ (0, \sigma_n) = Y_i e^{-\alpha\tau}$$

where

$X\ (0, \sigma_n)$ = sequence of random, normal numbers previously calculated

α = the system's cut-off frequency (temporal or spatial)

τ = the time (or distance) interval between points in the sequence.

(NOTE: The points are assumed to be equidistant)

Y = the resulting sequence

The $\alpha\tau$ product gives complete control over the spectrum shaping. It was determined after several trials that $\alpha\tau \approx 0.055$ gave the best normality condition and a power spectrum of the desired form. The desired roughness is achieved by specifying the appropriate rms of the resulting sequence. This value is obtained by the formula:

$$rms = \frac{\sigma_n}{\sqrt{1-e^{-2\alpha\tau}}}$$

The computer program entitled "NOISE 1" listed later in this Appendix performs the operations for generating the random, normal sequences. If the only change in the profile construction process is the rms level, then the resulting profiles will be similar in all respects, except that their respective elevations will be proportionally related. That is, for rms levels of 1 and 3 (as illustrated in Figure C73) the profiles are similar in structure but one has elevations three times that of the other.

Absorbed Power Calculations

As part of the vehicle ride dynamics model, it is desired to compute absorbed power. The concept of absorbed power was developed by Dr. R. A. Lee of TACOM as a measure of the rate at which vibratory energy is absorbed by humans and is thought to be a meaningful descriptor of human vibration tolerance.

It was decided to develop an algorithm capable of producing time histories of absorbed power. The input to this algorithm was to consist of the on-going time history of acceleration being produced elsewhere in the vehicle dynamics model.

The outcome of efforts at WES to fabricate the absorbed power algorithm resulted in the combination of the following elements:

a. The acceleration-to-force transfer function shown in Figure C74.

b. The flow chart in Figure C75 producing absorbed power from force and acceleration.

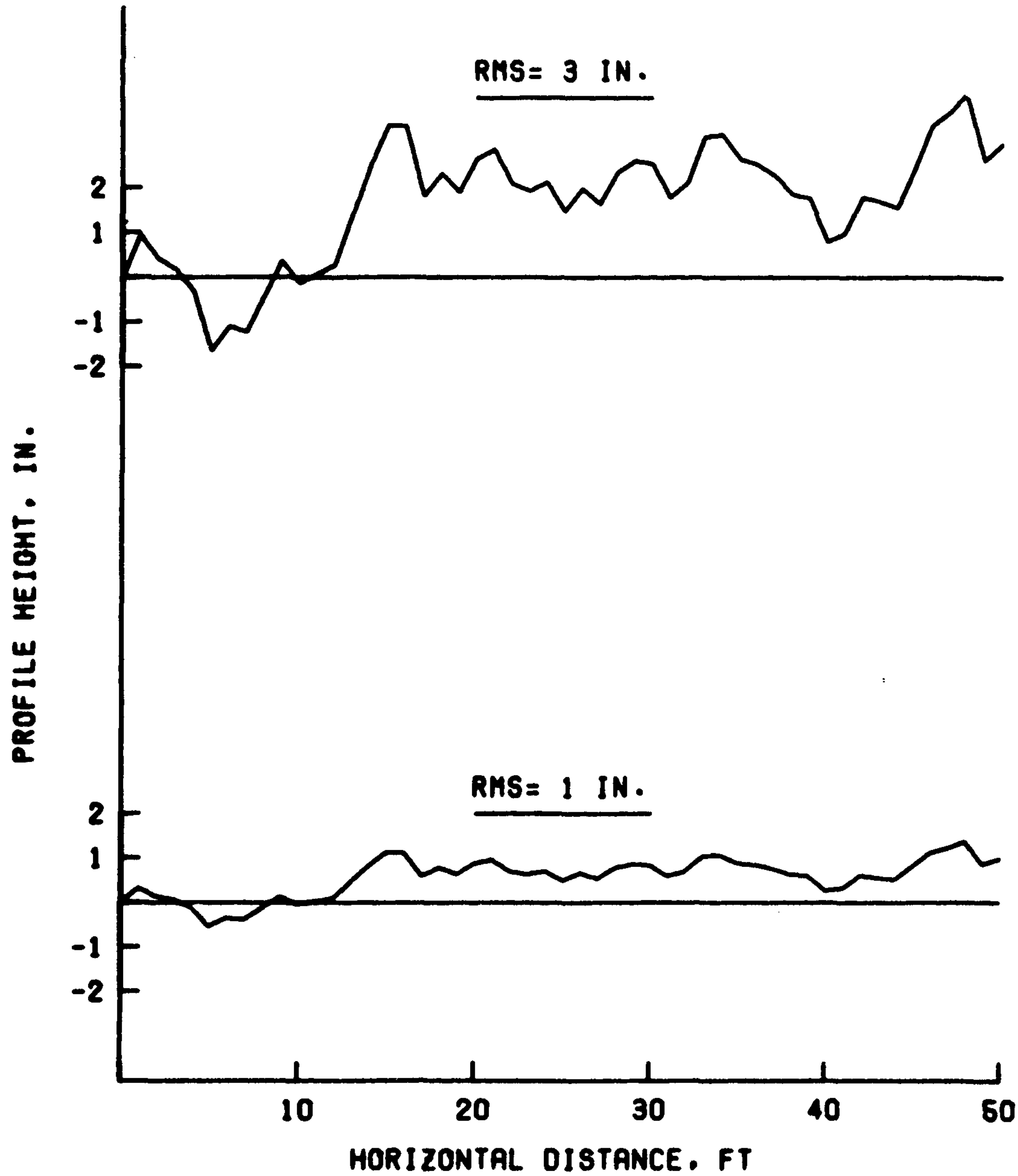


FIG. C73 COMPUTER-GENERATED RANDOM PROFILES

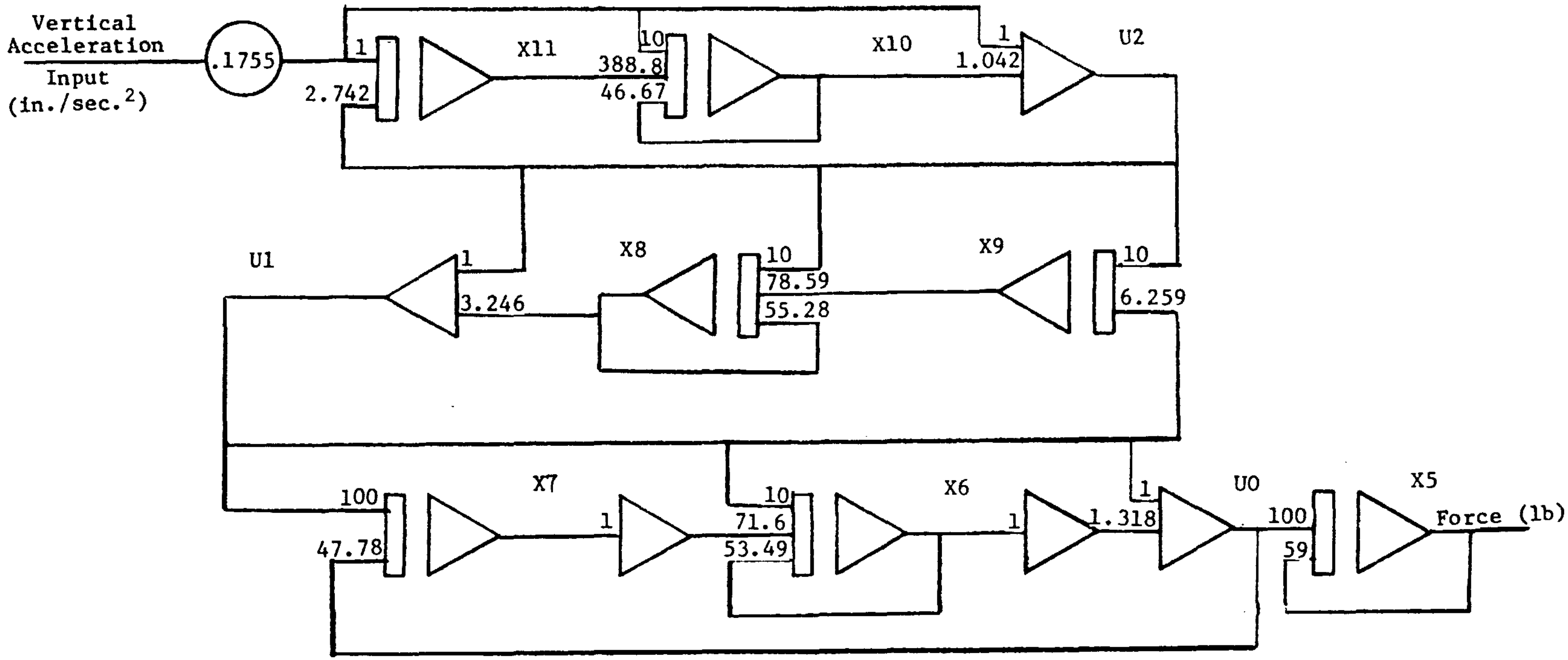


Fig. C74. Analog Circuit for Vertical Transfer Function

The algorithm was converted to digital form and inserted directly into the ride dynamics computer program to provide absorbed power-time histories during each vehicle simulation. The numerical equations were derived from the acceleration-to-force transfer function are given below. The variables and coefficients correspond to those shown on the analog flow diagram. They are as follows:

$$X11 = -0.1755a(t) - 2.742U2$$

$$X10 = -1.755a(5) - 388.8X11 - 46.67X10$$

$$U2 = -0.1755a(t) - 1.042X10$$

$$X9 = -10U2 - 6.259U1$$

$$X8 = -10U2 - 78.59X9 - 55.28X8$$

$$U1 = -U2 - 3.246X8$$

$$X7 = -100U1 - 47.78U0$$

$$X6 = -10U1 + 71.6X7 - 53.49X6$$

$$U0 = -U1 + 1.318X6$$

$$X5 = -100U0 - 59X5$$

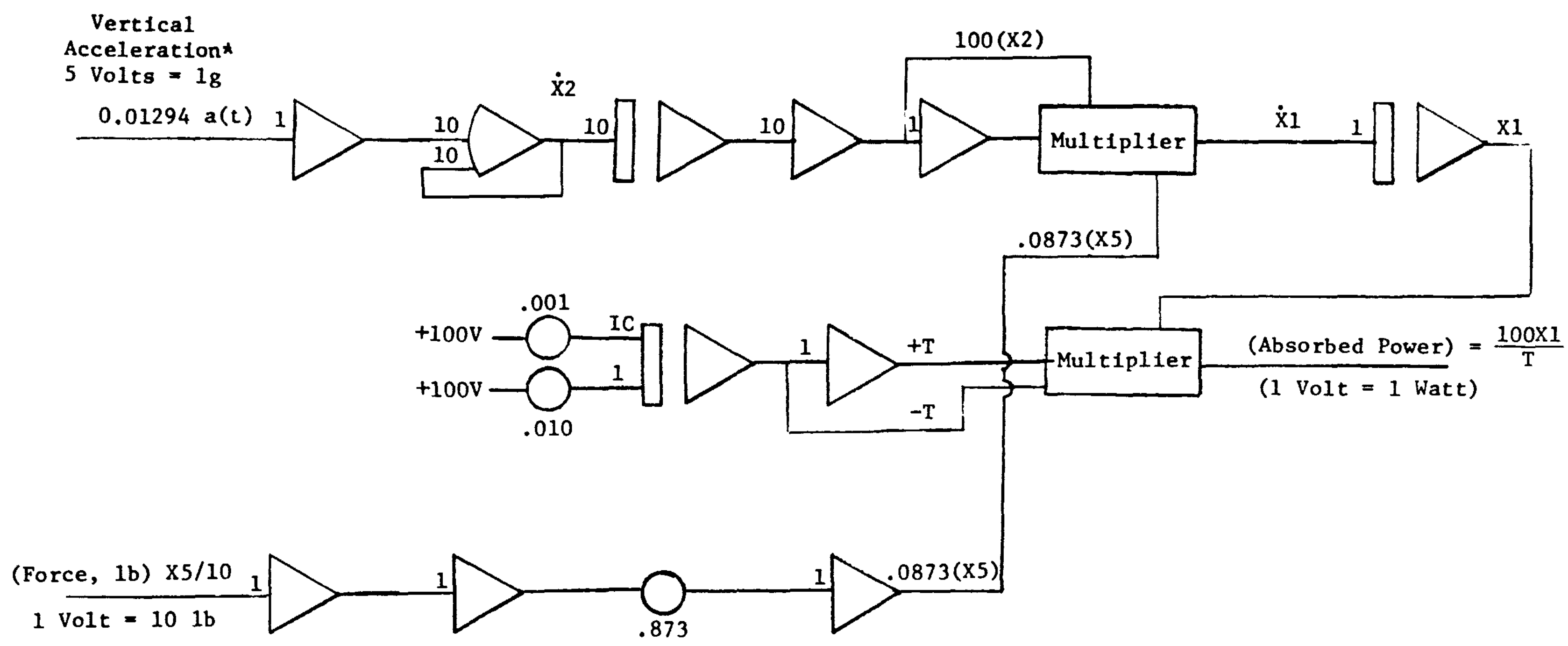
where X5 represents the simulated force and is obtained through integration of X5 by the Runge-Kutta-Gill algorithm.

The equations for producing the absorbed power from the simulated force and acceleration inputs are obtained from the analog flow chart in Figure C75 depicting the integration of acceleration to obtain velocity and the integration and division by time of the force-velocity product as required by the algorithm. These equations are:

$$X2 = -0.01294a(5)$$

$$X1 = 0.0873\ X2X5$$

$$PWR = 100X1/T$$



* To convert the acceleration a(t) from units of in/sec^2 as required by the transfer function to the specified input unit g requires multiplying by $\left(\frac{1}{12}\right)\left(\frac{5}{32.2}\right) = 0.0129$

FIGURE C75. Analog Flow Chart for Absorbed Power Circuit

Lack of knowledge of the voltage scale factors in the acceleration-to-force transfer function required the determination of a factor multiplying the force-velocity product that would produce the proper values on the calibration chart.

The constant comfort curve (Figure C76) developed by TACOM for vertical vibration was used to test the absorbed power algorithm. This was done by taking values of frequency and rms accelerations corresponding to points on the curve, forming sinusoidal acceleration-time histories with these numbers, and inserting these accelerations into the absorbed power equation. The desired outcome was the appearance of an absorbed power of 6 watts, corresponding to the constant comfort curve. The arbitrary factor was determined to be that value that caused the output of the absorbed power algorithm to be suitably near to 6 watts over the frequency range from 0.50 to 20 cps, when acceleration amplitudes corresponding to the TACOM constant comfort curve were supplied as input.

The results of the calibration checkout are shown in Figure C77. Ideally, the absorbed power would be 6 watts for all inputs. However, for frequencies up to about 20 cps, which is the range of interest in vehicle dynamics problems, the deviation from 6 watts is not significant.

Ten cycles of acceleration input were used to compute each value of absorbed power. Figures C78, C79 and C80 show representative comparisons between time domain and frequency domain calculations. Results from processing vehicle accelerations of actual field tests reveal that after a short period of travel over relatively stable surfaces, these transient fluctuations stabilize and converge to a value equivalent to that computed via frequency domain approach.

Determination of Limiting Speed

Singular Obstacles

Obstacle-limiting speeds for a given vehicle can be readily determined either by actual tests or via computer

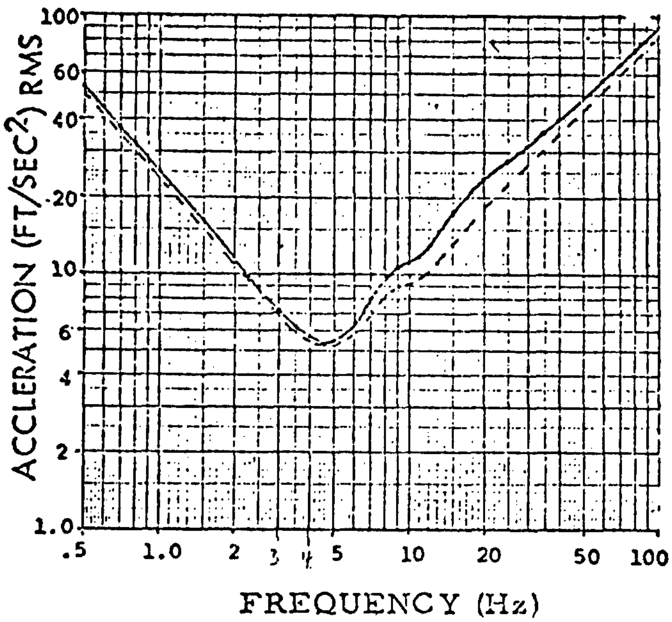


FIGURE C76. Vertical Constant Comfort Curve

RMS ACCEL, FT/SEC2	FREQ, HZ	ABSORBED POWER, WATTS
52.0	0.5	5.03
45.0	0.6	5.75
37.0	0.7	5.51
32.0	0.8	5.55
29.0	0.9	5.93
25.0	1.0	5.59
16.5	1.5	6.19
11.9	2.0	6.39
9.0	2.5	6.21
7.2	3.0	5.95
6.2	3.5	5.89
5.8	4.0	6.15
5.7	5.0	6.26
6.5	6.0	6.26
8.0	7.0	6.23
9.9	8.0	6.56
10.2	9.0	5.71
11.0	10.0	6.04
17.5	15.0	6.54
24.0	20.0	6.89
28.0	25.0	6.85
32.0	30.0	6.97
36.0	35.0	7.07
40.0	40.0	7.11
49.0	50.0	7.43
58.0	60.0	7.81
65.0	70.0	7.81
73.0	80.0	8.09
82.0	90.0	8.54
90.0	100.0	8.73

Fig. C77. Checkout of Absorbed Power Computations for Constant Comfort Levels

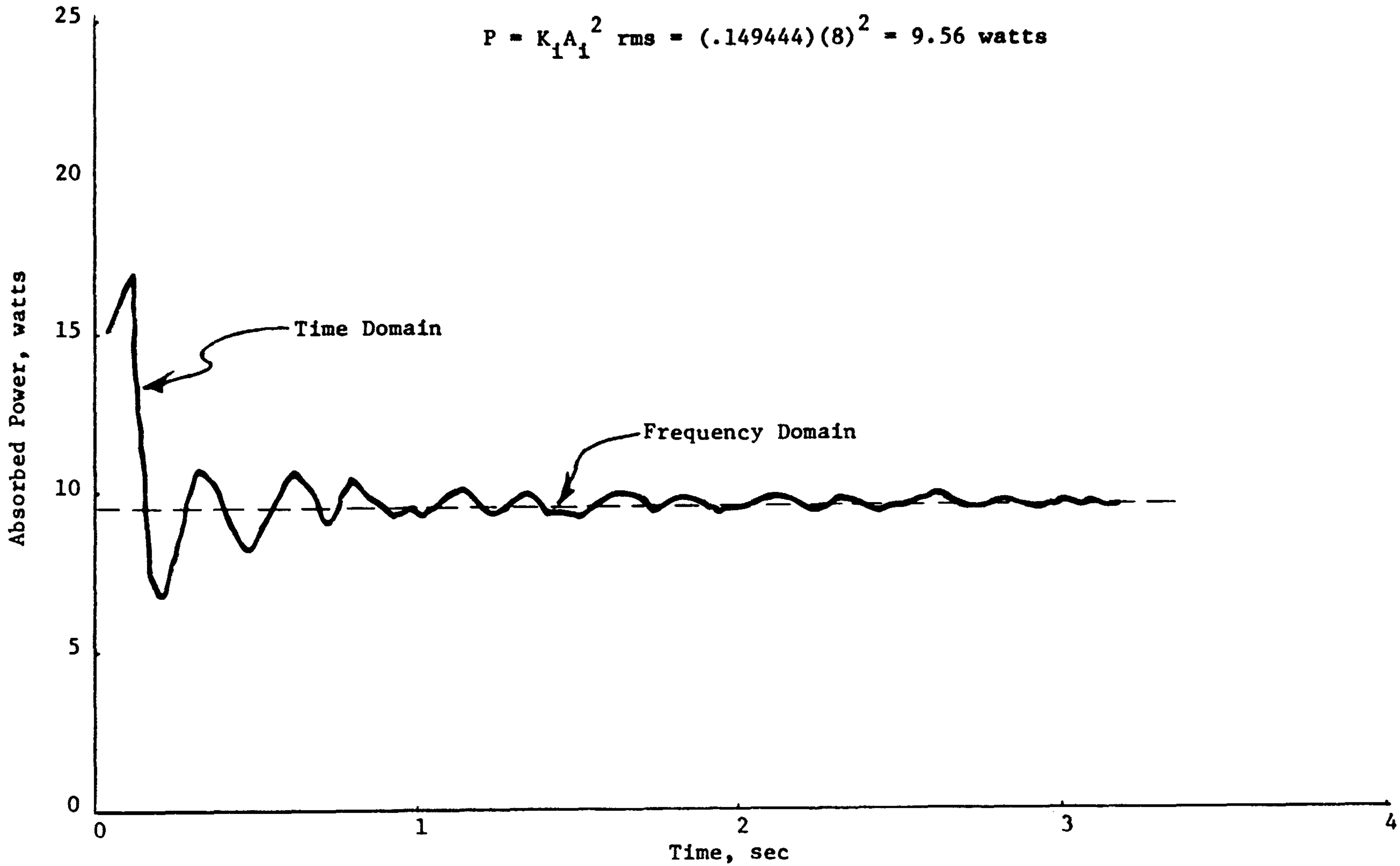


Fig. C78. Absorbed Power-Time History

Input: 6 cps, 8 ft/sec^2 rms

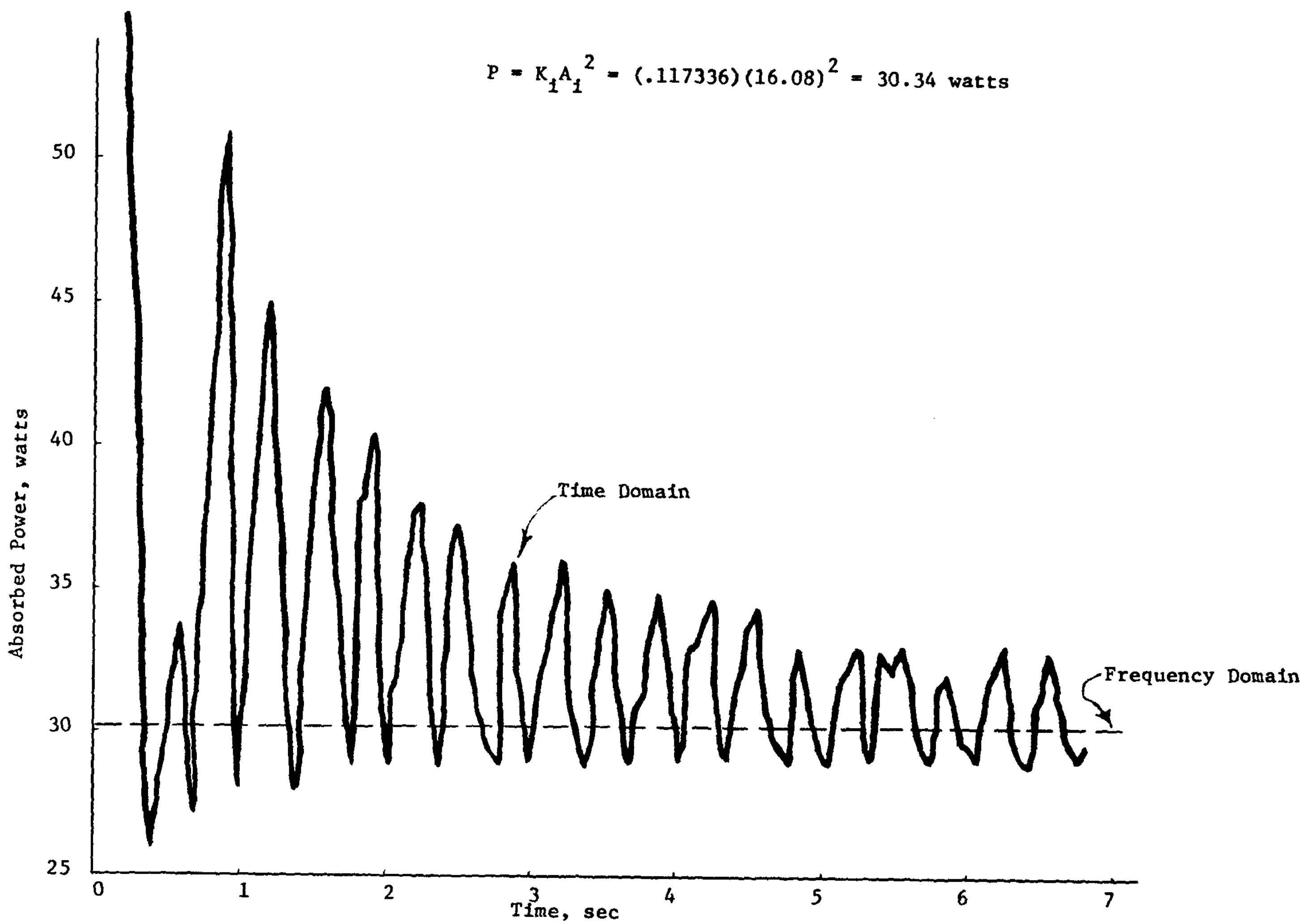


Fig. C79. Absorbed Power-Time History

Input: 3 cps, 16.08 ft/sec^2 rms

$$P = K_1 A_1^{\,2} \text{ rms} = (.203469)(16.08)^2 = 52.61 \text{ watts}$$

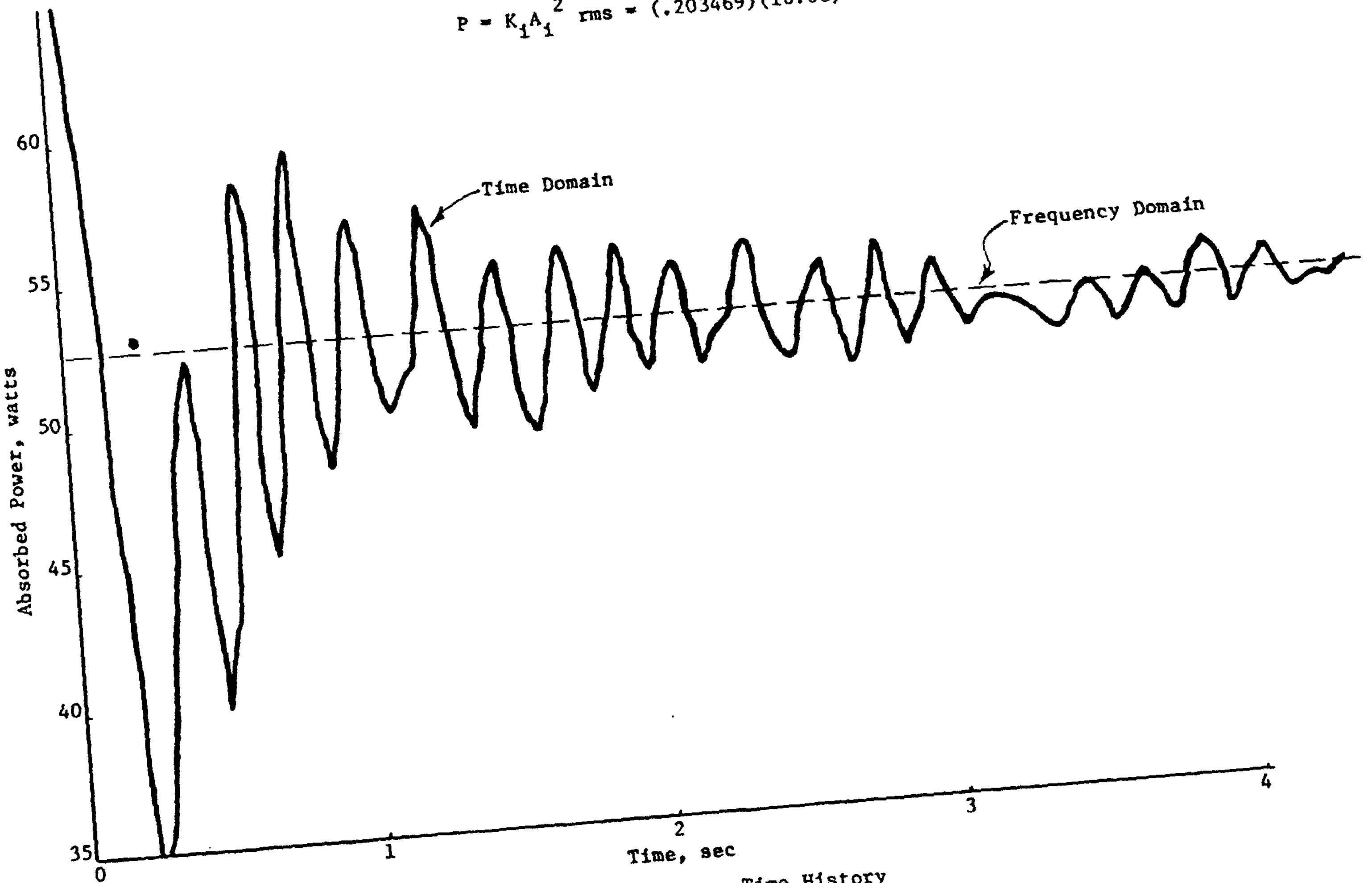


Fig. C80. Absorbed Power-Time History

Input: 4.5 cps, 16.08 ft/sec^2 rms

simulations. The tests or simulations are conducted to determine the speed at which 2.5-g vertical acceleration occurs for a series of obstacle heights. The resulting relation is of the form shown in Figure C81. Discrete coordinates, taken from such a curve, serve as the required input to the analytical model. Similar relations for the four reference vehicles are given in Figure C82.

Stable Ground Roughness

The speed limit of the vehicle is defined by the speed at which the driver's absorbed power reaches a stable level of 6 watts. In order to use the roughness index as an input, a digitally programmed random number generator is employed that provides random profiles with the specified roughness level and PSD characteristics. The vehicle model traverses each profile until a speed at which a stable 6-watt level of absorbed power is reached. Experience has shown that inputs with a high degree of statistical stationarity, such as those produced by the computer process, require approximately 200 feet of travel before stable vehicle responses are attained. For actual terrains, which are generally less "stationary", at least 300 feet of travel is normally required. The profile is then changed to correspond to another roughness level and the simulation repeated. This process is continued until a sufficient range of surface roughness is covered. A plot depicting ride-limiting speed as a function of surface roughness is then constructed in the manner shown in Figure C83. A similar plot establishing the roughness-speed characteristics for the four reference vehicles is given in Figure C84.

Coordinates are selected from the graph to serve as input to the analytical model. Plots of the type shown in Figures C81 and C83 can be readily obtained for each vehicle, and serve as ready catalogs of the speed limitation imposed by human tolerance to shock and vibration environments.

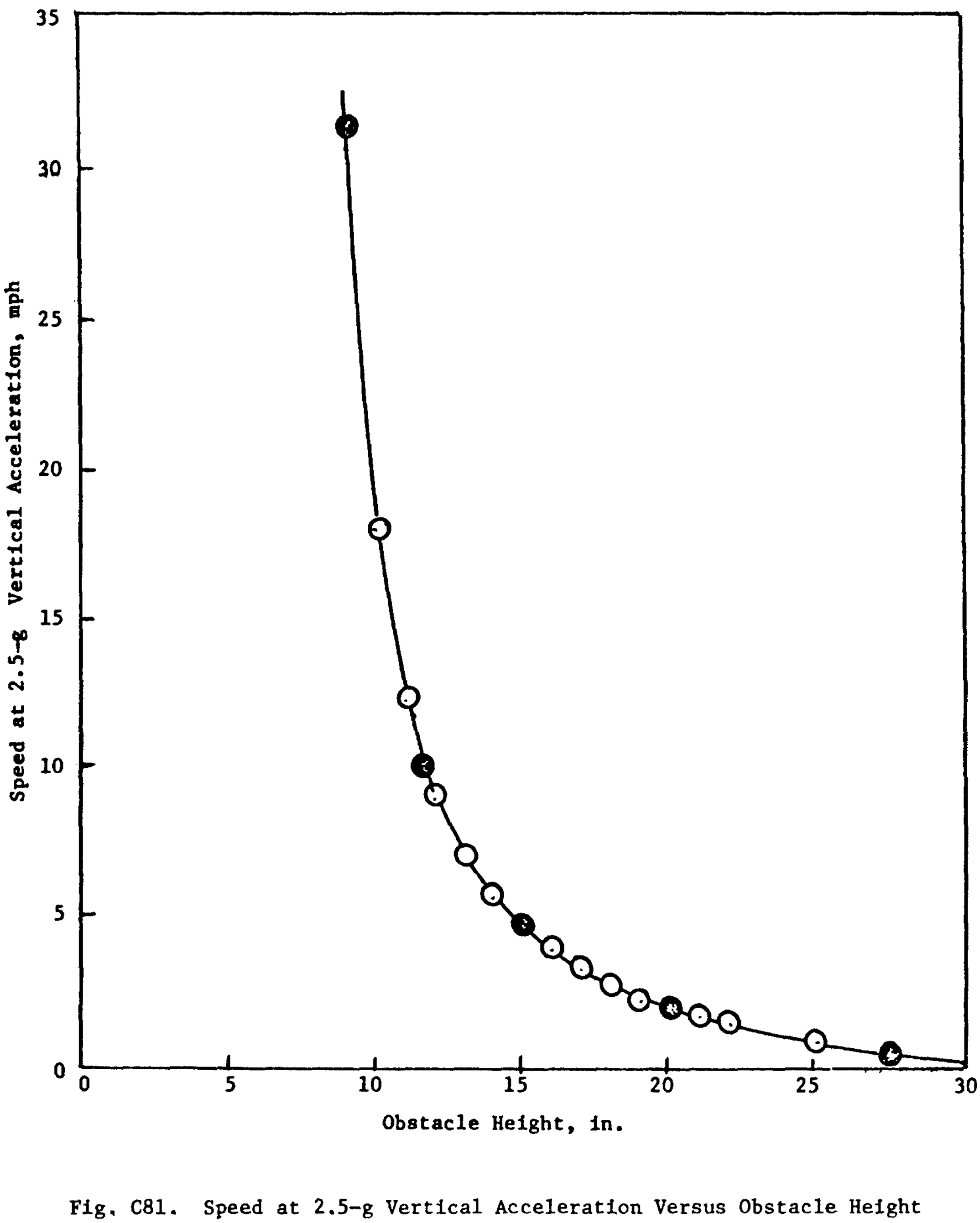


Fig. C81. Speed at 2.5-g Vertical Acceleration Versus Obstacle Height

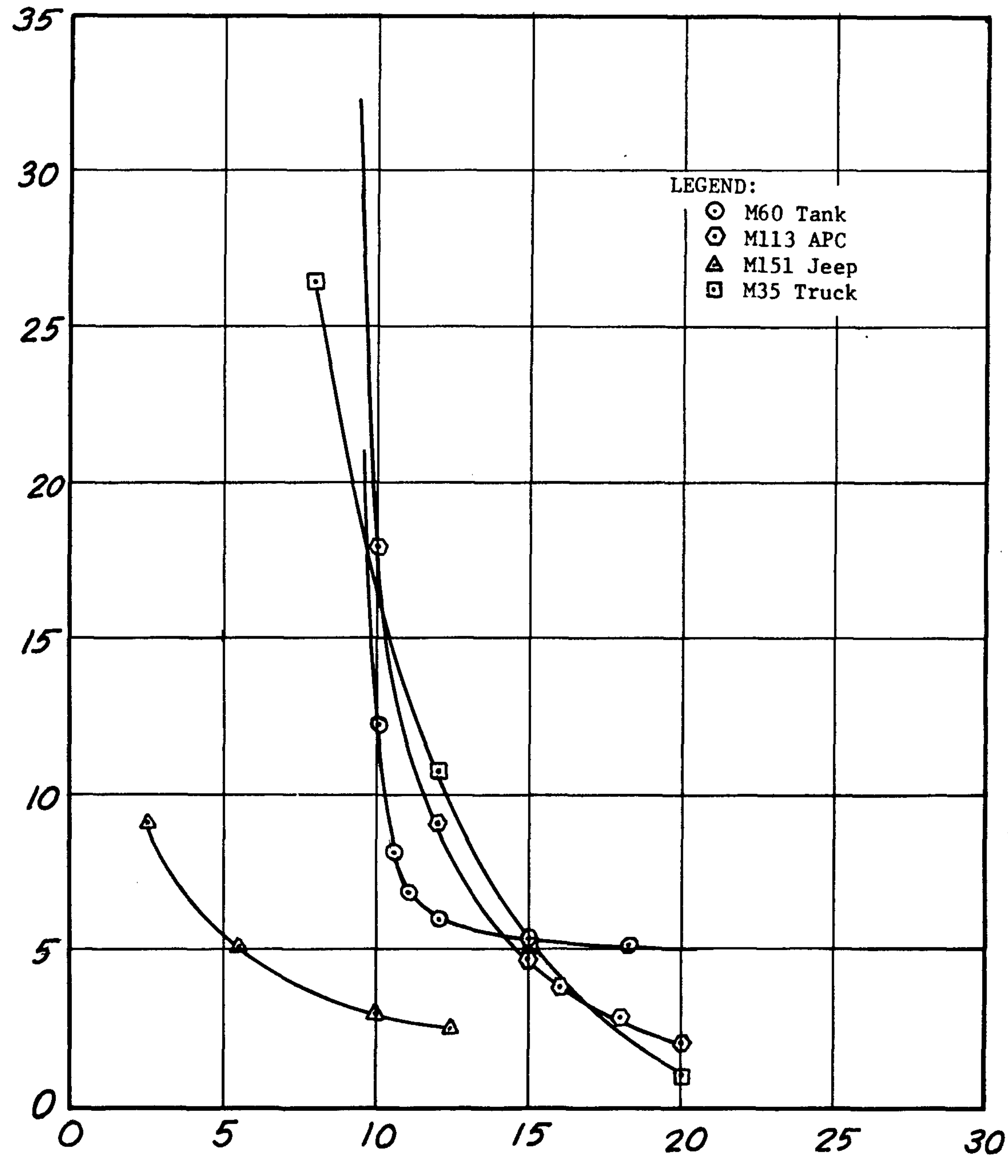


Fig. C82. Speed at 2.5-g Vertical Acceleration Versus Obstacle Height for the M60A1, M113, M35, and M151 Vehicles

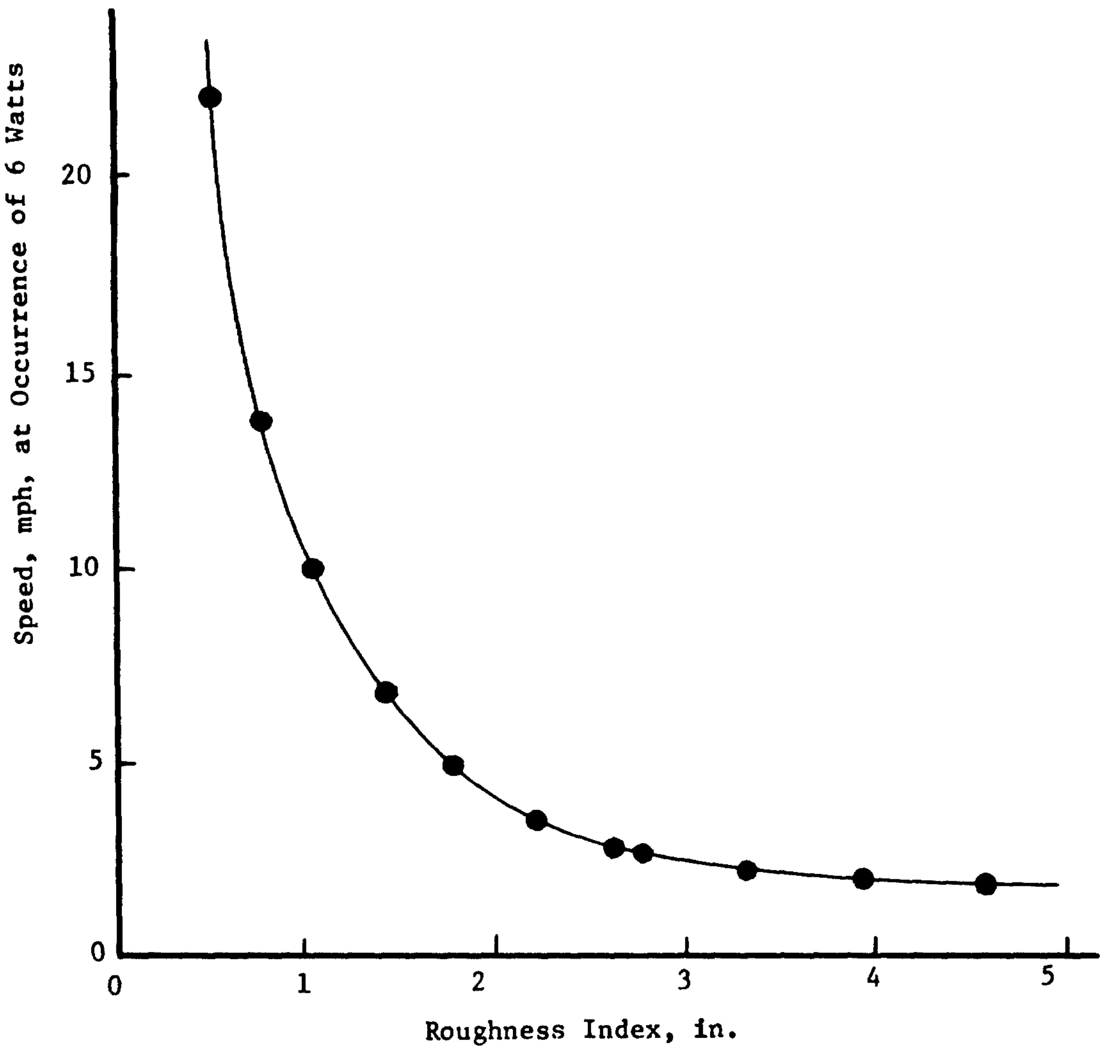


Fig. C83. Maximum Speed Versus Roughness Index

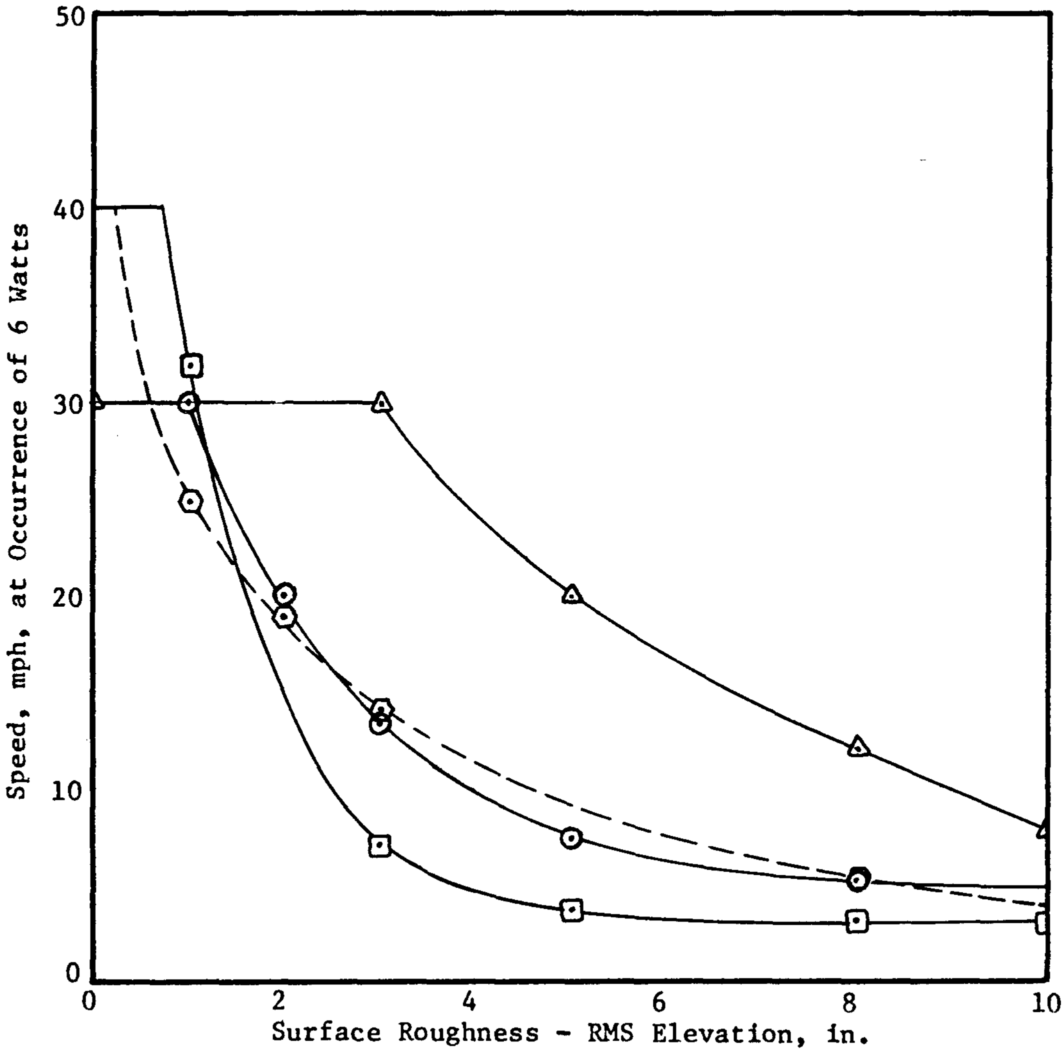


Fig. C84. Ride-Limiting Speed
Speed at 6 Watts Versus Surface Roughness

LEGEND:
- ⊙ M151
- ⊡ M113
- △ M60
- ⬡ M35

Computer Programs

The ride dynamics computer program VDPROG is of such a structure that each specific vehicle is included as a separate entity in the form of subroutines that are controlled by a general executive program. This particular structure permits the most efficient running time per problem and yet is suitably flexible, requiring little programming effort for the inclusion of each new vehicle. It is written in a conversational mode for time-sharing operations.

Program Input

The input to the program can consist of a terrain profile in the form of a file containing equally spaced elevations, or it can be a file describing a discrete obstacle or series of obstacles, or it can be a set of statistical quantities, in which case the program then calls upon a subroutine to generate internally a random profile composed of the specified statistical quantities. A present restriction on the input profile is that the elevation points must be spaced at 4-inch intervals. Therefore, regardless of what spacing an original profile may have, it is first processed by an interpolation program that creates via linear interpolation a file of elevations with 4-inch spacing.

Program Output

Optional outputs are available that provide the user with the response quantities most generally used in the field of vehicle dynamics studies, such as peak accelerations, rms accelerations, driver absorbed power and detailed motion-time histories of each degree of freedom. A limited output is printed on the teletype at a time increment determined by the user. If desired, a complete detailed printout at each time step can be obtained from a file by the high-speed printer at the WES Automatic Data Processing Division.

Operating Procedures

A teletype printout for a discrete obstacle test is shown in Figure C85 to illustrate the questions to be answered at the start of the program and the type of printout for the response. The print interval was chosen as 0.25 sec. The procedures for running this program are as follows:

a. Call the main program - VDPROG.

b. Give name of vehicle to be simulated.

c. Choose the desired output by answering "yes" or "no" to the questions that follow.

d. Upon answering the questions, the program then automatically runs to completion.

Some noteworthy comments are in order to explain certain options. A "yes" answer to A DETAILED OUTPUT FILE? will cause the program to generate a detailed file of the motions of each degree of freedom and the driver, rms of all accelerations, absorbed power, and peak accelerations. The program will ask the user to designate a name for this output file, and it can then be listed on the high-speed printer. A "no" answer precludes the generation of such a file.

The question DRIVER MOTIONS? is for those cases where the driver is positioned away from the C.G. of the sprung mass. A "yes" answer will cause the program to compute the driver motions from the bounce and pitch motions at the C.G. A "no" answer to this question will yield only the motions at the C.G. of each mass in the vehicle model.

A "yes" answer to RMS OF ALL ACCELS? will cause the program to compute the rms for all accelerations; a "no" answer precludes the program from making the calculations.

A "yes" answer to EXTERNAL FILE INPUT? tells the computer that the profile input will be from an external file and will cause the computer to ask for the name of the

```
RUN:VDPROG

VDPROG       09:24       05/10/71

THIS IS THE GENERALIZED TWO-DIMENSIONAL VEHICLE MODEL PROGRAM
NAME OF VEHICLE ? M-151

DO YOU WANT THE FOLLOWING OPTIONS
ABSORBED POWER  ? YES

A DETAILED OUTPUT FILE   ? NO

PEAK ACCELERATIONS ? YES

DRIVER MOTIONS  ? YES

RMS OF ALL ACCELS  ? YES

EXTERNAL FILE INPUT   ? YES

VEHICLE VELOCITY IN MPH. ? 10

TTY PRINTOUT TIME INTERVAL  ? .25

NAME OF INPUT PROFILE FILE
 ? OBS4



VELOCITY=10.00 MPH ( 176.0 IPS)
DELTA-L=4.000   DELTA-T=0.0227
NSTEPS=  22    H=.001033
VEHICLE IS: M-151 JEEP
INPUT PROFILE IS:
A 4 INCH TRAPEZOIDAL OBSTACLE


           DISPL      VELOC      ACCEL      RMSACC

TIME= 0.000 INPUT=  0.000 ABSORBED POWER=  0.000

C-G      -4.30300   0.00000   0.00000   0.00000
PITCH     0.00342   0.00000   0.00000   0.00000
AXLE1    -.31656    0.00000   0.00000   0.00000
AXLE2    -.33770    0.00000   0.00000   0.00000
DRVER    -4.30300   0.00000   0.00000   0.00000
```

Fig. C85. Teletype Printout from Computer Program

TIME= 0.273 INPUT= 0.000 ABSORBED POWER= 2.731

C-G	-1.76834	3.59814	-.34038	0.49184
PITCH	0.06617	-.04019	-2.87888	6.27680
AXLE1	0.70528	20.93096	-2.95146	6.64551
AXLE2	-.80537	-.13143	0.00807	0.06462
DRVER	-1.76834	3.59814	-.34038	0.49184

TIME= 0.523 INPUT= 0.000 ABSORBED POWER= 1.744

C-G	-4.60874	-7.89029	0.28938	0.44105
PITCH	-.00849	-.04715	2.61962	5.32405
AXLE1	-.92796	-.23212	-.16448	4.93000
AXLE2	-.86383	-.63290	0.02296	0.07731
DRVER	-4.60874	-7.89029	0.28938	0.44105

TIME= 0.773 INPUT= 0.000 ABSORBED POWER= 2.788

C-G	-3.26465	25.43113	0.13839	0.43657
PITCH	-.02695	-.56173	-.72271	5.14128
AXLE1	-.83768	0.26636	0.04108	4.05505
AXLE2	3.25439	-59.32187	-6.93707	2.66479
DRVER	-3.26465	25.43113	0.13839	0.43657

TIME= 0.886 INPUT= 0.000 ABSORBED POWER= 1.713

C-G	-1.29009	9.33573	-.40595	0.43849
PITCH	-.06072	-.08965	3.40869	5.15666
AXLE1	-.75357	0.66210	-.02565	3.78621
AXLE2	1.39995	22.44274	-2.69556	3.55318
DRVER	-1.29009	9.33573	-.40595	0.43849

PEAK ACCELERATION VALUES

	MAXIMUM	MINIMUM
C-GACC	0.9799	-.8131
PITCH	11.5535	-11.5919
AXLE1	16.0446	-7.7388
AXLE2	10.7903	-7.9264
DRIVER	0.9799	-.8131

RUNNING TIME: 126.9 SECS I/O TIME : 20.1 SECS

READY

Fig. C85(Continued)

file. A "no" answer indicates the profile is to be generated internally, and the program asks for the inputs to the subroutine.

The VEHICLE VELOCITY question is obvious.

The TTY PRINTOUT TIME INTERVAL? permits the user to specify the time increment of the teletype printout. A zero input here will yield a teletype printout every time step.

The question NAME OF INPUT PROFILE FILE? is simply asking for the name of the file containing the profile elevations.

This program was written with four basic ideas in mind:

a. The program would be run on a GE 430 Time Share system.

b. The program should be as efficient and yet as general as possible.

c. The program should have sufficient I/O options to provide adequate flexibility for dealing with the various types of dynamics simulations.

d. The vehicle parameters could be easily changed, thus permitting the inclusion of any specific vehicle with little programming effort.

A teletype printout illustrating the input requirements for calculating terrain profiles is shown in Figure C86. The variable, TAU, is the desired spacing of the profile points (can be any units of length). RMS is currently just a dummy variable; it is indirectly related to the distribution of the original sequence of random, normal numbers with a flat spectra, that is, before frequency shaping. ALPHA controls the cut-off frequency and is generally given a value such that the (ALPHA) (TAU) product = 0.055. DESRMS is the desired RMS level of the resulting profile. IX is the starting point of the random number chain, and NPTS is the number of points specified in the resulting profile.

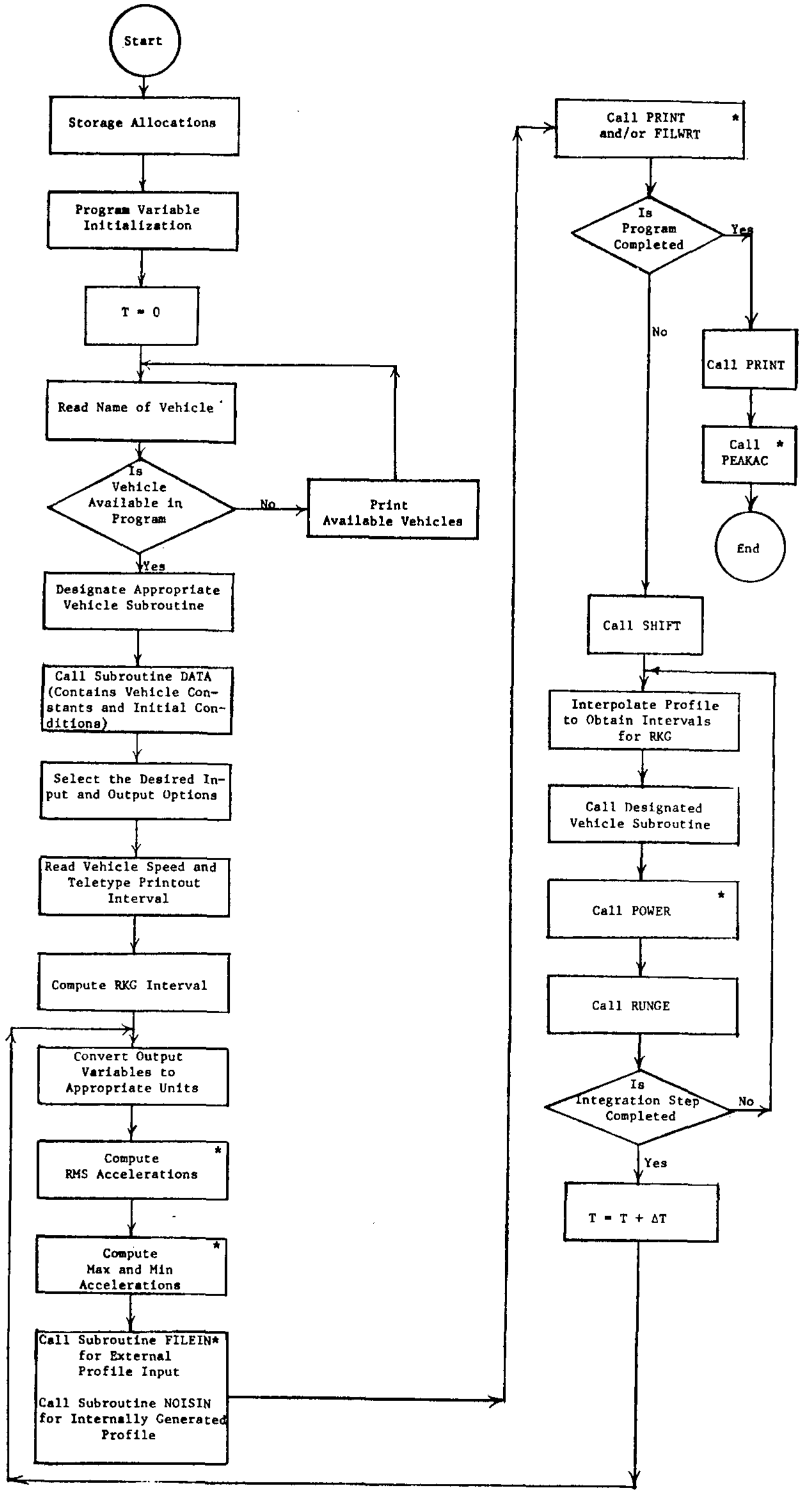
Flow charts illustrating the logic of the programs and subroutines comprising the Ride Dynamics Model are given in Figures C86, C87, C88, C89, C90, C91, C92, C93, C94, C95 and C96.

```
OLD:NOISE1

READY
RUNNH

TAU,RMS,ALPHA,DESRMS,IX,NPTS
 ?12,5,.0046,1,2555,300

SIGMAN=     1.616509563E+00
X-BAR AFTER GAUSS=     1.061495617E-01
X-BAR AFTER SHIFT=     4.608106489E-11
RMS=    1.000000000E+00
FACT=    1.114613877E-01
OUTPUT FILE NAME
 ?RMS1
```

Fig. C86. Teletype Printout from Noise 1 Program Illustrating Input Requirements and the Output Parameters

MAIN PROGRAM: VDPROG

PURPOSE: To simulate responses of vehicles traveling over rough surfaces



*The accomplishment of these actions depends on the selected options

Fig. C87

SUBROUTINE SHIFT

PURPOSE: To shift profile array, point by point, beneath the vehicle

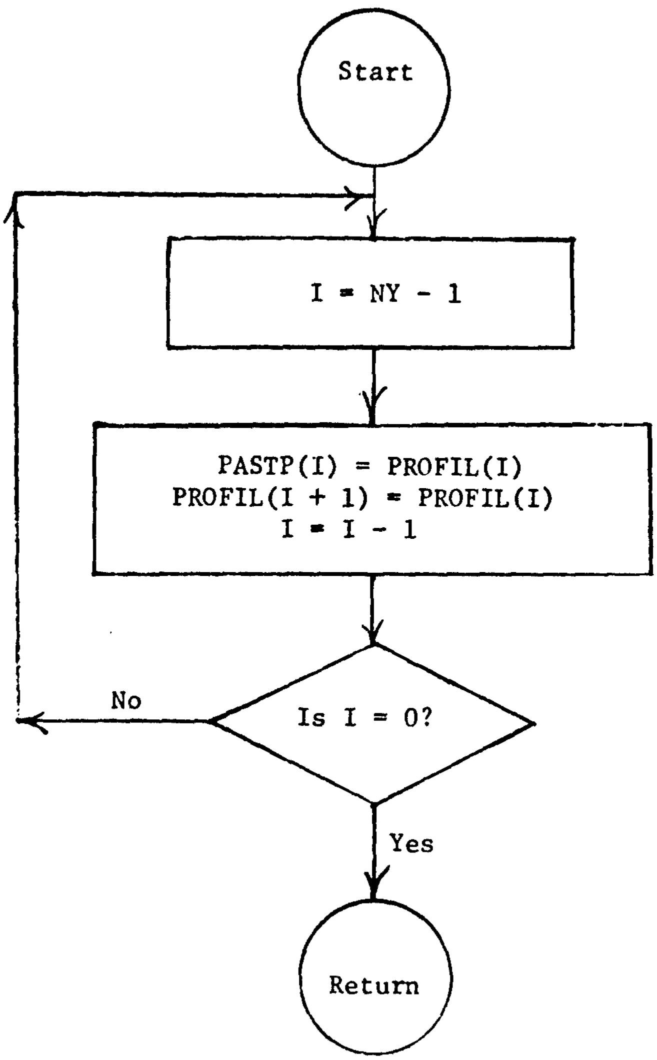


Fig. C88

SUBROUTINE FILIN

PURPOSE: To input a profile point from an external file

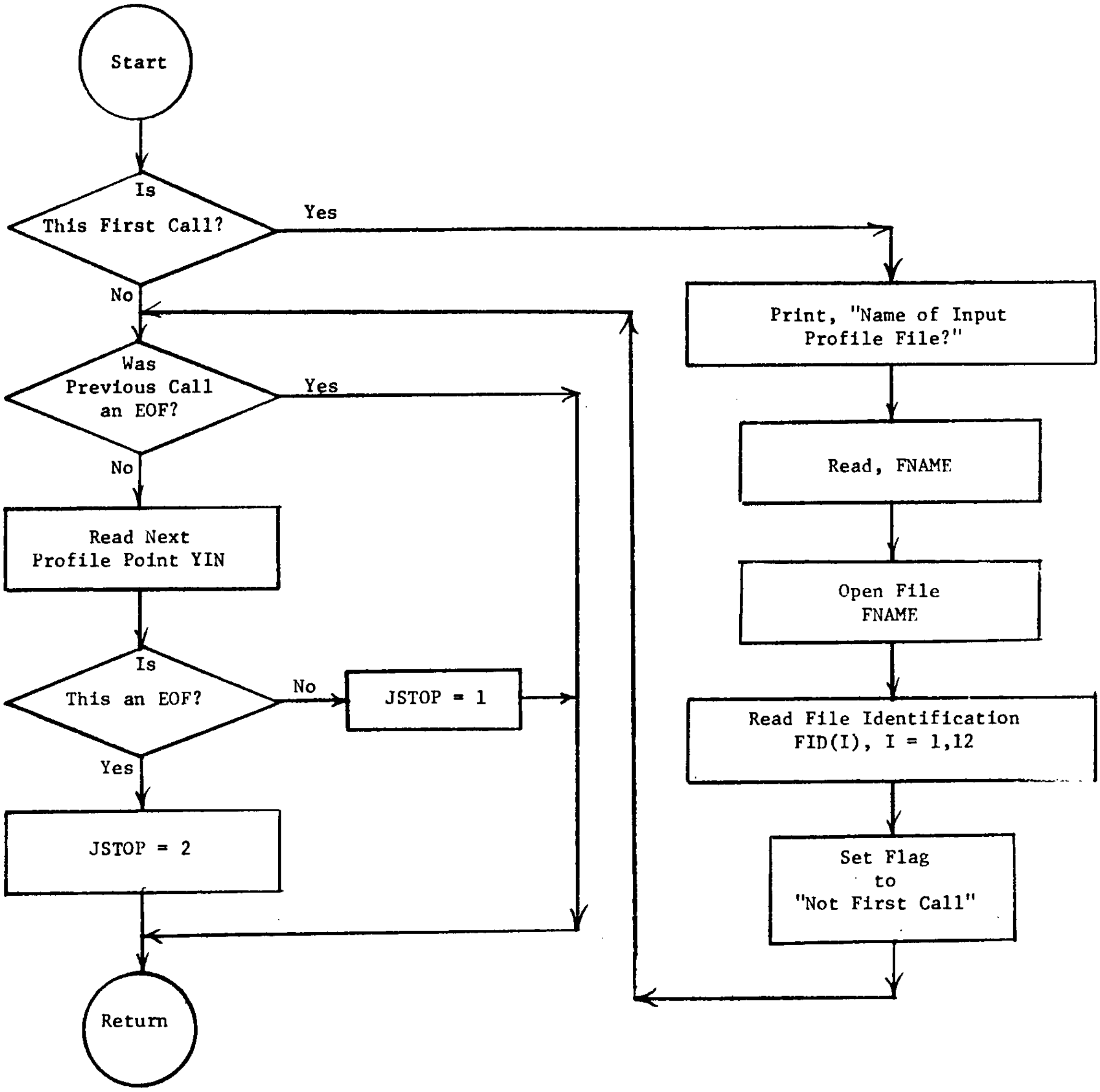
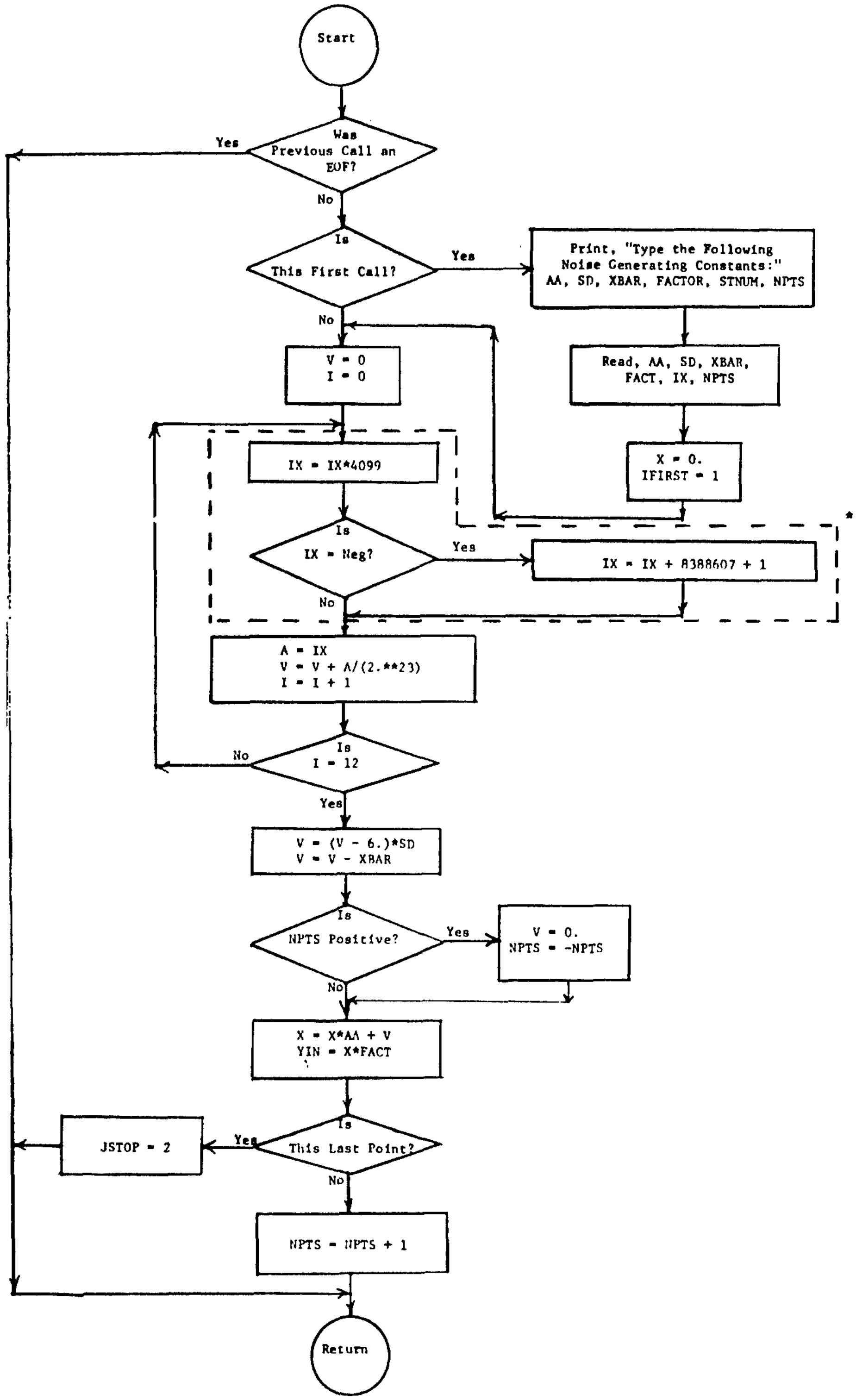


Fig. C89

SUBROUTINE NOISIN

PURPOSE: To internally generate a random profile with normally distributed amplitudes and a specified rms and PSD



*This operation is equivalent to the congruential multiplicative method given by IX = MOD(4099*IX,2**23) but will work only on a machine with 24 bit integer words and which uses two's complement representation

Fig. C90

SUBROUTINE PRINT

PURPOSE: Provides a teletype printout of all pertinent pre-test information, appropriate headings and response quantities.

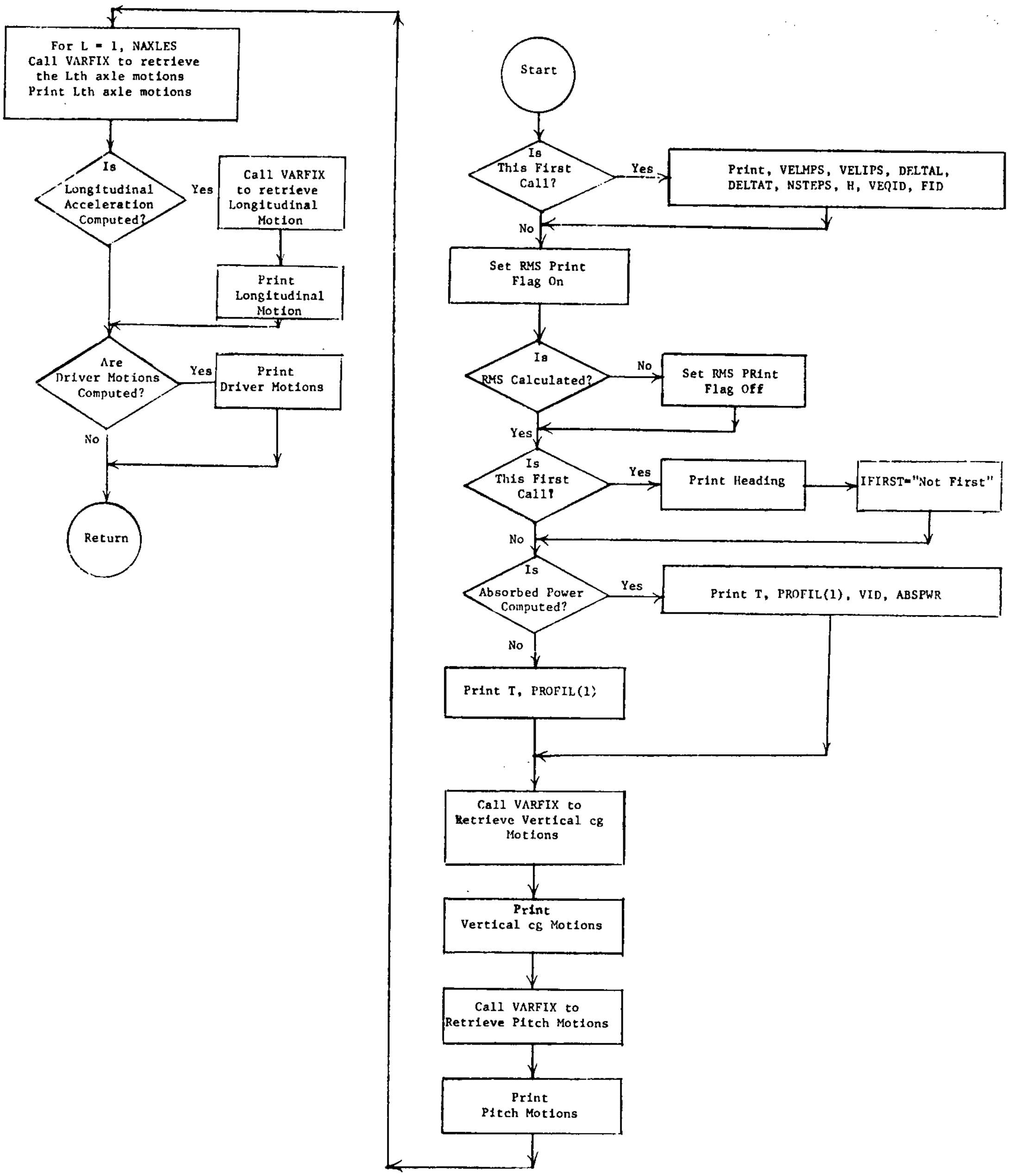


Fig. C91

SUBROUTINE VARFIX

PURPOSE: To arrange the response motions in proper sequence for printout

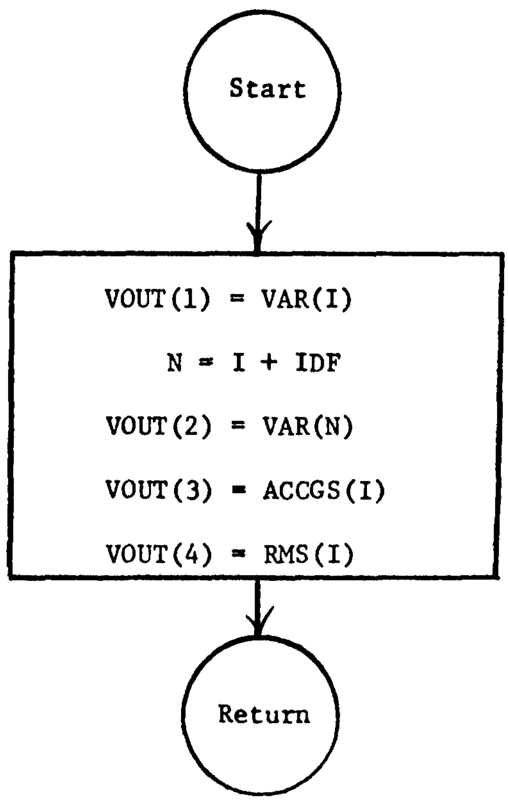


Fig. C92

SUBROUTINE FILWRT

PURPOSE: To write the output to a file in a format suitable for listing on a high speed line printer.

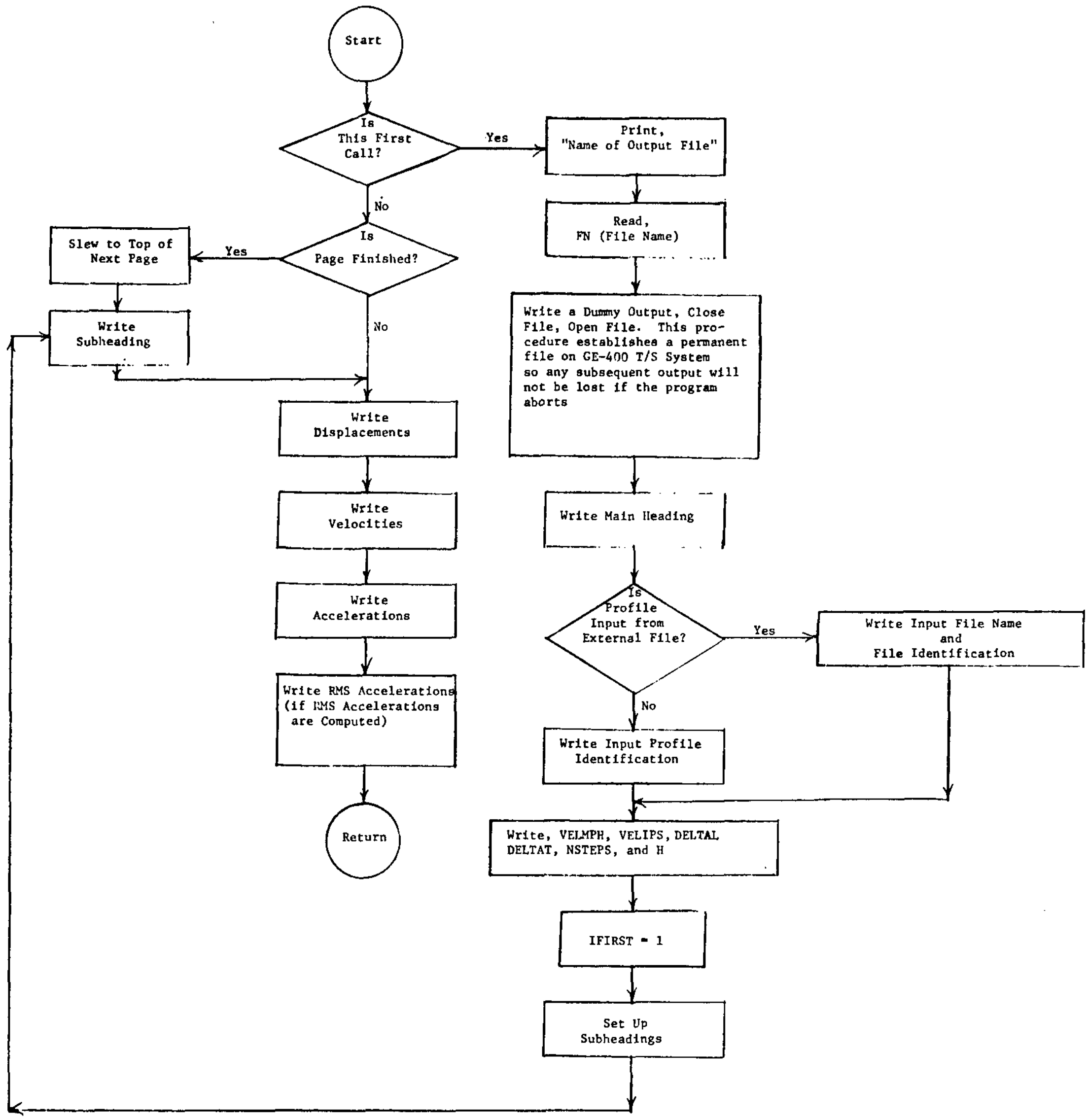


Fig. C93

SUBROUTINE PEAKAC

PURPOSE: To print the maximum and minimum accelerations at the termination of the program

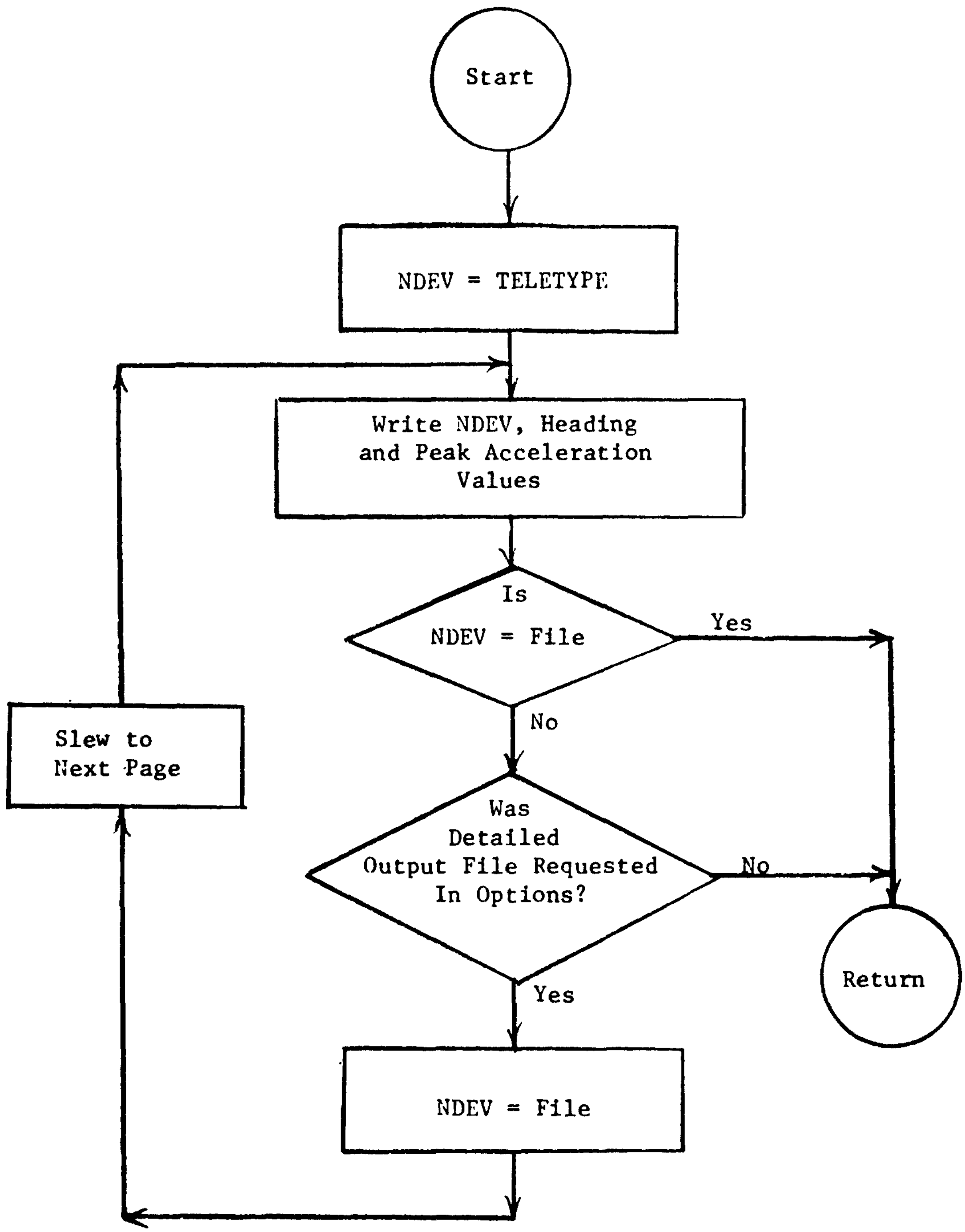


Fig. C94

SUBROUTINE DATA

PURPOSE: To systematically store the vehicle parameters for insertion into appropriate elements of the main program

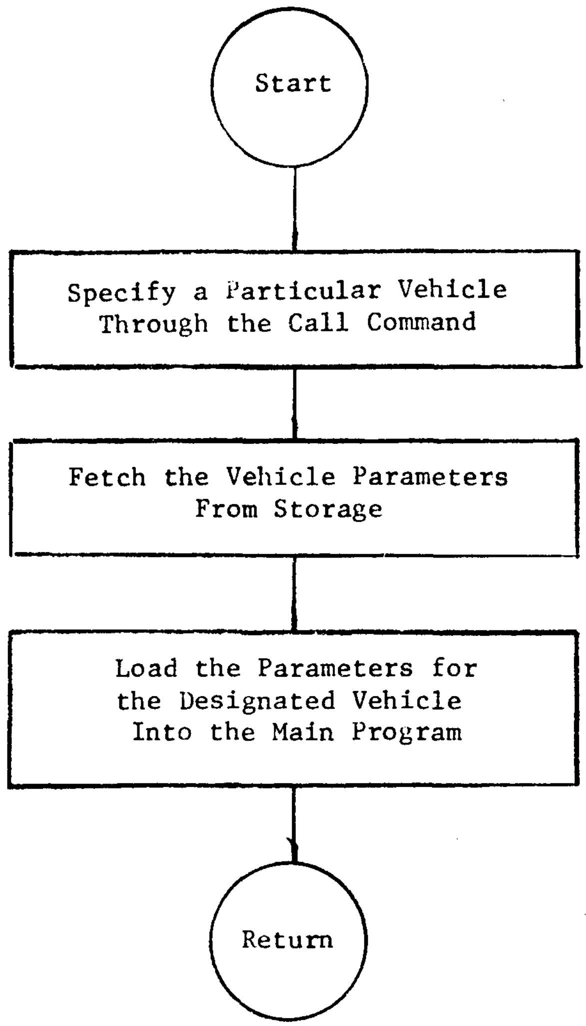


Fig. C95

GENERAL FORM OF A VEHICLE SUBROUTINE

PURPOSE: To compute the forces acting on the vehicle for substituting into the RKG Algorithm

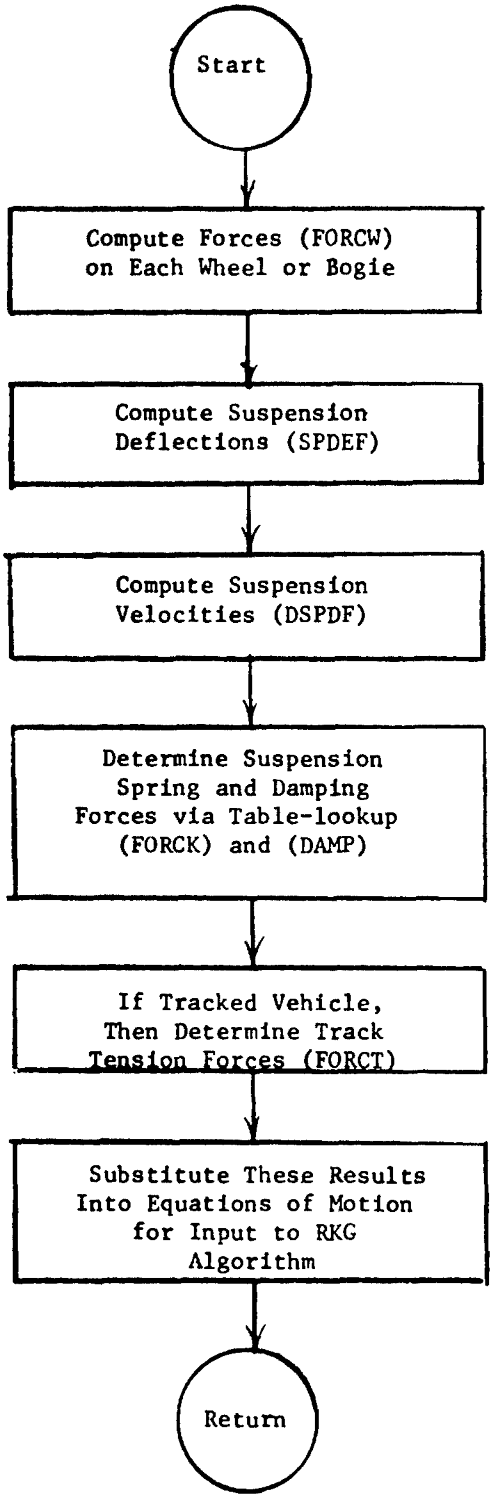


Fig C96

VDPROG

```
$LIB,COST,,F509
$NDM
$RPC
$SAV
C ********STORAGE ALLOCATIONS********
      COMMONFORCH(6),FORCT(7),FORCK(6),FORCW(6)
      COMMONSPDEF(6),DSPDF(6),THRESH(9),SIGMA(9),GAMMA(9)
      COMMONVAR(18),Y(100),PWRVAR(9),DAMP(6)
      COMMONACCISS(9),ACCGS(9),ACCMAX(9),ACCMIN(9)
      COMMONSUMRMS(9),RMS(9),LEN(10),MASS(6)
      COMMONH,T,DELTAT,DELTAL,VELIPS,VELMPH,NSTEPS
      COMMONYIN,DRVMAX,DRVMIN,ABSPWR
      COMMONDISDRV,VELDRV,ACCDRV,RMSDRV
      COMMONIFPWR,IFFILE,IFPACC,IFDRV,IFRMS,IFINPT
      COMMONNY,IDF,NAXLES,NSEGS,IFHORZ,FNAME
      COMMON FMASS, INRTIA, HORMOM, DRVLEN
      COMMONVEHQID(2)
      COMMON PROFIL(100),PASTP(100),INDEX
      DIMENSION DRIVER(4),IOPT(6)
      DIMENSION FID(12),XTNAME(4)
      DIMENSION FK(18),P(18),Q(18),PY(9),PWRFK(9),PP(9),QQ(9)
      INTEGER SETUP(5)
      REALLEN,MASS,INRTIA
      EQUIVALENCE (SETUP,NY)
      EQUIVALENCE(DRIVER(1),DISDRV)
      EQUIVALENCE(IOPT(1),IFPWR)
      DATAFID/36HINTERNALLY GENERATED PROFILE WITH SH,
&     30HAPED PSD AND SPECIFIED RMS.    ,1H /
      DATAIYES/1HY/,NO/1HN/
      DATA XTNAME/5HM-151,4HM-35,4HM-60,5HM-113/
      DATAIBELL/458752/
C ********VARIABLE INITIALIZATION********
      CALL COST
      DO 10 I=1,100
 10   Y(I)=PROFIL(I)=PASTP(I)=0.
      DO20I=1,9
      PWRVAR(I)=0.
      QQ(I)=0.
      ACCISS(I)=0.
      SUMRMS(I)=0.
      Q(I)=0.
      ACCMAX(I)=0.
 20   ACCMIN(I)=0.
      ACCDRV=0.
      DO30I=1,18
 30   VAR(I)=0.
      T=0.
      SDVRMS=0.
      ABSPWR=0.
      DRVMAX=0.
```

VDPROG CONTINUED

```
550      DRVMIN=0.
560      YIN=0.
570C----YIN IS THE PROFILE INPUT POINT
580      H=.001
590C----H IS RKG STEP SIZE
600      DELTAL=4.
610C---DELTAL IS PROFILE SEGMENT SPACING IN INCHES
620      JSTOP=1
630C----JSTOP IS THE STOP FLAG, 1=GO, 2=STOP
640      NSTOP=0
650C----NSTOP IS THE STOPPING VARIABLE
660      TPRINT=0.
670C----TPRINT IS TTY PRINTOUT TIMEKEEPER
680C********VEHICLE CONSTANTS READ-IN********
690      PRINT,"THIS IS THE GENERALIZED TWO-DIMENSIONAL VEHICLE",
700&      " MODEL PROGRAM"
710 12   PRINT,"NAME OF VEHICLE",^*
720      READ1,TNAME
730      DO 51 I=1,4
740      IF(TNAME-XTNAME(I))51,11,51
750 51   CONTINUE
760      PRINT3,XTNAME
770      GOTO12
780 11   GO TO (61,62,63,64)I
790 61   ASSIGN 81 TO ISUB
800      GOTO65
810 62   ASSIGN 82 TO ISUB
820      GOTO65
830 63   ASSIGN 83 TO ISUB
840      GOTO65
850 64   ASSIGN 84 TO ISUB
860 65   CALL DATA(I)
870C********SELECTION OF OPTIONS********
880      DO40I=1,6
890 40   IOPT(I)=IYES
900 50   PRINT,"DO YOU WANT THE FOLLOWING OPTIONS"
910      PRINT,"ABSORBED POWER",^*
920      CALLINPUT(IFPWR)
930      PRINT,"A DETAILED OUTPUT FILE",^*
940      CALLINPUT(IFFILE)
950      PRINT,"PEAK ACCELERATIONS",^*
960      CALLINPUT(IFPACC)
970      PRINT,"DRIVER MOTIONS",^*
980      CALLINPUT(IFDRV)
990      PRINT,"RMS OF ALL ACCELS",^*
1000      CALLINPUT(IFRMS)
1010      PRINT,"EXTERNAL FILE INPUT",^*
1020      CALL INPUT(IFINPT)
1030C********PROGRAM OPTION SET-UP********
1040 70   IRMS=IDF
```

VDPROG CONTINUED

```
      IMINMX=IDF
      IF(IFPACC.EQ.NO)IMINMX=0
      IF(IFRMS.EQ.NO)IRMS=0
C ********VEHICLE RUN VARIABLE INPUT*****
      PRINT,"VEHICLE VELOCITY IN MPH.",^*
      READ,VELMPH
      PRINT,"TTY PRINTOUT TIME INTERVAL",^*
      READ,TIP
C ********TIME STEP & RKG TIME SET-UP********
      VELIPS=VELMPH*17.6
      DELTAT=DELTAL/VELIPS
      NSTEPS=DELTAT/H
      TEMP=NSTEPS
      H=DELTAT/TEMP
C ********OUTPUT SCALING********
  100 DO15I=1,IDF
  15  ACCGS(I)=ACCISS(I)/386.
C----RESET PITCH ACCEL
      ACCGS(2)=ACCISS(2)
      ABSPWR=-100.*PWRVAR(1)/T
      IF(IFDRV-NO)41,42,41
  41  DISDRV=VAR(1)+DRVLEN*VAR(2)
      VELDRV=VAR(IDF+1)+DRVLEN*VAR(IDF+2)
C ********RMS CALCULATION********
  42  DO16I=1,IRMS
      SUMRMS(I)=SUMRMS(I)+ACCGS(I)**2
  16  RMS(I)=SQRT(SUMRMS(I)*DELTAT/T)
      IF(IFDRV-NO)13,14,13
  13  SDVRMS=SDVRMS+ACCDRV**2
      RMSDRV=SQRT(SDVRMS*DELTAT/T)
C ********PEAK ACCELERATION CALCULATION********
  14  DO17I=1,IMINMX
      ACCMAX(I)=AMAX1(ACCMAX(I),ACCGS(I))
  17  ACCMIN(I)=AMIN1(ACCMIN(I),ACCGS(I))
      IF(IFDRV-NO)35,36,35
  35  DRVMAX=AMAX1(DRVMAX,ACCDRV)
      DRVMIN=AMIN1(DRVMIN,ACCDRV)
C ********PROFILE INPUT********
  36  IF(IFINPT-NO)25,26,25
  25  CALLFILIN(FID,JSTOP)
      GOTO27
  26  CALLNOISIN(JSTOP)
C ********PROGRAM OUTPUT********
  27  IF(IFFILE-NO)18,19,18
  18  CALLFILWRT(FID,NPL)
  19  IF(T-TPRINT)21,22,22
  22  CALLPRINT(FID)
      TPRINT=TPRINT+TIP
C ********MAIN PROGRAM**********
  21  IF(NSTOP-NY)23,24,23
```

VDPROG CONTINUED

```
 23   CALLSHIFT
C----SHIFT ADVANCES THE Y PROFILE ARRAY
      IF(JSTOP-2)28,29,28
 29   NSTOP=NSTOP+1
 28   PROFIL(1)=YIN
      INDEX=-NSTEPS
      LDF=2*IDF
 599  DO 199 J=1,IDF
      K=J+IDF
 199  PY(J)=VAR(K)
      DO 999 I=1,NY
 999  Y(I)=PASTP(I)+((INDEX+NSTEPS+1)*(PROFIL(I)-PASTP(I)))/NSTEPS
      DO 299 I=1,4
      GO TO ISUB,(81,82,83,84)
 81   CALL M151(FK)
      GOTO299
 82   CALL M35(FK)
      GOTO 299
 83   CALL M60(FK)
      GOTO 299
 84   CALL M113(FK)
 299  CALLRUNGE(P,Q,VAR,FK,LDF,I)
      DO 399 I=1,IDF
      K=I+IDF
 399  ACCISS(I)=(VAR(K)-PY(I))/H
      ACCDRV=(ACCISS(1)+DRVLEN*ACCISS(2))/386.
      IF(IFPWR-1HN)699,799,699
 699  DO 899 I=1,4
      CALLPOWER(PWRFK)
 899  CALLRUNGE(PP,QQ,PWRVAR,PWRFK,9,I)
 799  INDEX=INDEX+1
      IF(INDEX)599,499,599
 499  CONTINUE
      T=T+DELTAT
      GOTO100
C*******FINAL OUTPUT*******
 24   CALLPRINT(FID)
      IF(IFPACC-NO)31,32,31
 31   CALLPEAKAC(NPL)
 32   PRINT2,(IBELL,I=1,40)
      CALL COST
      CALL EXIT
 1    FORMAT(2A6)
 2    FORMAT(40A1)
 3    FORMAT(28HTHE AVAILIABLE VEHICLES ARE:X,3(A5,1H,),X,1H&,XA5)
      END
      SUBROUTINESHIFT
THIS SUBROUTINE ADVANCES THE PROFILE UNDER THE VEHICLE
      I=NY-1
 1    PASTP(I)=PROFIL(I)
```

VDPROG CONTINUED

```
          PROFIL(I+1)=PROFIL(I)
          I=I-1
          IF(I)1,2,1
    2     RETURN
          END
          SUBROUTINEFILIN(FID,JS)
THIS SUBROUTINE OPENS THE INPUT PROFILE FILE, READS A
NEW INPUT VALUE [YIN], AND CHECKS FOR END OF FILE.
          DIMENSIONFID(12)
          DATAIFIRST/0/
          IF(IFIRST)1,2,1
    2     PRINT,"NAME OF INPUT PROFILE FILE"
          READ5,FNAME
          CALLOPENF(1,FNAME)
          READ(1,5)FID
          IFIRST=1
    1     IF(JS-2)3,4,3
    3     READ(1,)YIN
          CALLEOFTST(1,JS)
    4     RETURN
    5     FORMAT(12A6)
          END
          SUBROUTINENOISIN(J)
THIS SUBROUTINE SUPPLYS THE NEXT INPUT PROFILE POINT
FROM AN INTERNALLY GENERATED RANDOM NUMBER (FROM
GAUSS AND RANDU), AND SHAPES THE RANDOM NOISE
TO A SPECIFIED PSD.
          DATAIFIRST/0/
          IF(J-2)1,2,1
    1     IF(IFIRST)3,4,3
    4     PRINT,"TYPE THE FOLLOWING NOISE GENERATING CONSTANTS:"
          PRINT,"AA,SD,X-BAR,FACTOR,ST-NUM,NPTS"
          READ,AA,SD,XBAR,FACT,IX,NPTS
          X=0.
          IFIRST=1
C***GAUSS & RANDU
    3     V=0.
          I=0.
    10    IX=IX*4099
          IF(IX)30,40,40
    30    IX=IX+8388607+1
    40    A=IX
          V=V+A/(2.**23)
          I=I+1
          IF(I-12)10,20,10
    20    V=(V-6.)*SD
C***NOISE GENERATION
          V=V-XBAR
          IF(NPTS)5,5,6
    6     V=0.
```

VDPROG CONTINUED

```
        NPTS=-NPTS
 5      X=X*AA+V
        YIN=X*FACT
        IF(NPTS)7,8,7
 8      J=2
        RETURN
 7      NPTS=NPTS+1
 2      RETURN
        END
        SUBROUTINEPRINT(FID)
THIS SUBROUTINE HANDLES THE PROGRAM PRINTOUT IN THE
TELETYPE.
        DIMENSIONFID(12),HEAD(4),VID(3)
        DIMENSIONVOUT(4)
        DATAVID/15HABSORBED POWER=/
        DATAHEAD/24HDISPL VELOC ACCEL RMSACC/
        DATAIFIRST/0/,NO/1HN/
 25     IF(IFIRST)2,1,2
 1      PRINT11,VELMPH,VELIPS,DELTAL,DELTAT,NSTEPS,H
        PRINT12,VEHQID,FID
 2      K=4
        IF(IFRMS-NO)9,10,9
 10     K=3
 9      IF(IFIRST)20,21,20
 21     PRINT13,(HEAD(I),I=1,K)
 20     IFIRST=1
        IF(IFPWR-NO)3,4,3
 3      PRINT 14, T, PROFIL(1), VID, ABSPWR
        GOTO5
 4      PRINT 14, T, PROFIL(1)
 5      CALLVARFIX(1,VOUT)
        PRINT15,(VOUT(I),I=1,K)
        CALLVARFIX(2,VOUT)
        PRINT16,(VOUT(I),I=1,K)
        DO22L=1,NAXLES
        N=2+L
        CALLVARFIX(N,VOUT)
 22     PRINT17,L,(VOUT(I),I=1,K)
        IF(IFHORZ)23,24,23
 23     N=N+1
        CALLVARFIX(N,VOUT)
        PRINT18,(VOUT(I),I=1,K)
 24     IF(IFDRV-NO)7,8,7
 7      PRINT19,(DRIVER(I),I=1,K)
 8      PRINT,
        RETURN
 11     FORMAT(///"VELOCITY="F5.2," MPH ("F6.1," IPS)"/
&       "DELTA-L="F5.3,3X,"DELTA-T="F6.4/
&       "NSTEPS="I4,4X,"H="F7.6)
 12     FORMAT(12HVEHICLE IS: 2A6/17HINPUT PROFILE IS:/12A6)
```

VDPROG CONTINUED

```
3050 13    FORMAT(//,6X,4(4X,A6))
3060 14    FORMAT(/,5HTIME=F6.3,X,6HINPUT=F7.3,X,2A6,A3,F7.3)
3070 15    FORMAT(/,3HC-G3X,4F10.5)
3080 16    FORMAT(5HPITCHX,4F10.5)
3090 17    FORMAT(4HAXLEI1,X,4F10.5)
3100 18    FORMAT(5HHORIZX,4F10.5)
3110 19    FORMAT(5HDRVERX,4F10.5)
3120       END
3130       SUBROUTINEVARFIX(I,VOUT)
3140THIS SUBROUTINE IS CALLED BY PRINT TO SELECT THE
3150VARIABLES TO BE PRINTED.
3160       DIMENSIONVOUT(4)
3170       VOUT(1)=VAR(I)
3180       N=I+IDF
3190       VOUT(2)=VAR(N)
3200       VOUT(3)=ACCGS(I)
3210       VOUT(4)=RMS(I)
3220       RETURN
3230       END
3240       SUBROUTINEFILWRT(FID,NPL)
3250THIS SUBROUTINE HANDLES THE OUTPUT TO AN EXTERNAL FILE
3260THAT IS WRITTEN WITH 120 CHARACTER LINES, AND CAN BE
3270LISTED ON THE HIGH-SPEED PRINTER.
3280       DIMENSIONHEAD1(6),HEAD2(2)
3290       DIMENSIONVOUT(10),FID(12)
3300       DATAIFIRST/0/,NPL/15/
3310       DATAHEAD1/35HAXLE1 AXLE2 AXLE3 AXLE4 AXLE5 AXLE6/
3320       DATAHEAD2/11HH,C-G V,DRV/
3330       IF(IFIRST)1,2,1
3340 2     PRINT,"NAME OF OUTPUT FILE",^*
3350       READ11,FN
3360 11    FORMAT(A6)
3370       WRITE(2;16)
3380       CALLCLOSEF(2,FN,7)
3390       CALLOPENF(2,FN,7)
3400       WRITE(2;12)VEHQID
3410 12    FORMAT(41X,37(1H*),/,41X,1H*,35X,1H*,/,41X,1H*,2X,
3420&      2A6,19HPROGRAM OUTPUT FILE2X,1H*,/,
3430&      41X,1H*,35X,1H*,/,41X,37(1H*)//)
3440       IF(IFINPT-1HN)3,4,3
3450 3     WRITE(2;13)FID,FNAME
3460       GOTO5
3470 4     WRITE(2;14)FID
3480 13    FORMAT(17HINPUT PROFILE IS:X,12A6,10X,12H[ FILE NAME ,A6,
3490&      X,1H]/)
3500 14    FORMAT(17HINPUT PROFILE IS:X,12A6,/)
3510 5     WRITE(2;15)VELMPH,VELIPS,DELTAL,DELTAT,NSTEPS,H
3520 15    FORMAT(/,9HVELOCITY=F6.2,X,17HMILES PER HOUR  (,
3530&      F6.1,X,18HINCHES PER SECOND)7X,8HDELTA-L=F5.3,X,6HINCHES,
3540&      9X,8HDELTA-T=F10.8,X,7HSECONDS,//,
```

VDPROG CONTINUED

```
&       35HNUMBER OF STEPS IN RKG INTEGRATION=I4,26X,
&       12HSTEP SIZE H=F12.10//)
        IFIRST=1
        KK=2
        IF(IFDRV-1HN)10,20,10
   20   HEAD2(2)=0
        KK=KK-1
   10   IF(IFHORZ)21,22,21
   22   HEAD2(1)=HEAD2(2)
        KK=KK-1
        GOTO21
    1   IF(NPL-50)6,7,7
    7   IF(NPL-54)8,9,9
    8   WRITE(2;16)
        NPL=NPL+1
        GOTO7
    9   NPL=0
   16   FORMAT(1H )
   21   WRITE(2;17)(HEAD1(I),I=1,NAXLES),(HEAD2(I),I=1,KK)
   17   FORMAT(2X,4HTIME3X,4HY(1)14X,5HV,C-G,4X,5HPITCH,4X,8(A6,3X))
        WRITE(2;16)
        NPL=NPL+2
    6   DO23I=1,IDF
   23   VOUT(I)=VAR(I)
        J=IDF
        IF(IFDRV-1HN)24,25,24
   24   J=J+1
        VOUT(J)=DISDRV
   25   WRITE(2;18) T, PROFIL(1), (VOUT(I),I=1,J)
   18   FORMAT(/,X,F7.4,F6.2,2X,6HDISPL.2X,10F9.4)
        DO26I=1,IDF
        K=I+IDF
   26   VOUT(I)=VAR(K)
        IF(IFDRV-1HN)27,28,27
   27   VOUT(J)=VELDRV
   28   WRITE(2;19)(VOUT(I),I=1,J)
   19   FORMAT(15X,8HVELOCITY10F9.4)
        DO29I=1,IDF
   29   VOUT(I)=ACCGS(I)
        IF(IFDRV-1HN)31,32,31
   31   VOUT(J)=ACCDRV
   32   IF(IFPWR-1HN)33,34,33
   34   WRITE(2;30)(VOUT(I),I=1,J)
        GOTO35
   33   WRITE(2;36)ABSPWR,(VOUT(I),I=1,J)
   35   NPL=NPL+4
   30   FORMAT(16X,6HACCEL.2X,10F9.4)
   36   FORMAT(6HPOWER=F6.2,4X,6HACCEL.2X,10F9.4)
        IF(IFRMS-1HN)37,38,37
   37   DO39I=1,IDF
```

VDPROG CONTINUED

```
39    VOUT(I)=RMS(I)
      IF(IFDRV-IHN)41,42,41
41    VOUT(J)=RMSDRV
42    WRITE(2;40)(VOUT(I),I=1,J)
40    FORMAT(16X,8HRMS.ACC.10F9.4)
      NPL=NPL+1
38    RETURN
      END
      SUBROUTINERUNGE(P,Q,X,FK,M,N)
THIS SUBROUTINE IS THE RUNGE-KUTTA-GILL ALGORITHM.
      DIMENSIONP(1),Q(1),X(1),FK(1),A(4),B(4),C(4)
      DATAA/.5,.2928932188,1.707106781,.1666666667/
      DATAB/2.,1.,1.,2.,/
      DATAC/.5,.2928932188,1.707106781,.5/
      TA=A(N)
      TB=B(N)
      TC=C(N)
      DO10I=1,M
      P(I)=TA*(FK(I)-TB*Q(I))
      X(I)=X(I)+P(I)
10    Q(I)=Q(I)+3.*P(I)-TC*FK(I)
      RETURN
      END
      SUBROUTINEPOWER(FK)
THIS SUBROUTINE IS THE ABSORBED POWER EQUATIONS.
      DIMENSIONFK(9)
2     U2=-67.743*ACCDRV-1.042*PWRVAR(8)
      U1=-U2-3.246*PWRVAR(6)
      U0=-U1+1.318*PWRVAR(4)
      FK(1)=H*(.00873*PWRVAR(2)*PWRVAR(3))
      FK(2)=H*(-49.9484*ACCDRV)
      FK(3)=H*(-100.*U0-59.*PWRVAR(3))
      FK(4)=H*(-10.*U1+71.6*PWRVAR(5)-53.49*PWRVAR(4))
      FK(5)=H*(-100.*U1-47.78*U0)
      FK(6)=H*(-10.*U2-78.59*PWRVAR(7)-55.28*PWRVAR(6))
      FK(7)=H*(-10.*U2-6.259*U1)
      FK(8)=H*(-677.43*ACCDRV-388.8*PWRVAR(9)-46.67*PWRVAR(8))
      FK(9)=H*(-67.743*ACCDRV-2.742*U2)
3     RETURN
      END
      SUBROUTINEPEAKAC(NPL)
THIS SUBROUTINE WRITES THE PEAK ACCELERATION VALUES
      N=66
19    WRITE(N;)"PEAK ACCELERATION VALUES"
      WRITE(N;)"        MAXIMUM  MINIMUM"
      WRITE(N;6)(ACCMAX(I),ACCMIN(I),I=1,2)
      DO15I=1,NAXLES
      J=I+2
15    WRITE(N;7)I,ACCMAX(J),ACCMIN(J)
      IF(IFHORZ)11,12,11
```

VDPROG CONTINUED

```
4550 11    WRITE(N;8)ACCMAX(IDF),ACCMIN(IDF)
4560 12    IF(IFDRV-1HN)13,14,13
4570 13    WRITE(N;9)DRVMAX,DRVMIN
4580 14    IF(N-2)4,2,4
4590 4     IF(IFFILE-1HN)16,2,16
4600 16    IF(NPL-54)17,18,18
4610 17    WRITE(2;5)
4620       NPL=NPL+1
4630       GOTO16
4640 18    N=2
4650       GOTO19
4660 2     RETURN
4670 5     FORMAT(1H )
4680 6     FORMAT(6HC-GACC 2F9.4,/,6HPITCH 2F9.4)
4690 7     FORMAT(4HAXLEI1,X,2F9.4)
4700 8     FORMAT(6HHORIZ 2F9.4)
4710 9     FORMAT(6HDRIVER 2F9.4)
4720       END
4730       SUBROUTINEINPUT(INP)
4740 10    READ1,INP
4750       IF(INP.EQ.1HY.OR.INP.EQ.1HN)RETURN
4760       PRINT,"TYPE YES OR NO",^*
4770       GOTO10
4780 1     FORMAT(A1)
4790       END
4800       SUBROUTINE DATA(N)
4810       DIMENSION DVEHCL(2,4),DTHRSH(9,4),DGAMMA(9,4),DSIGMA(9,4)
4820       DIMENSION DMASS(6,4),DVAR(9,4),DLEN(10,4)
4830      DIMENSION DFMASS(4),DINRTA(4),DDRVLN(4)
4840       INTEGER DSETUP(5,4)
4850       DATA DVEHCL/10HM-151 JEEP,11HM-35  TRUCK,10HM-60  TANK,
4860&      10HM-113 TANK/
4870       DATA DSETUP/34,4,2,7,0,53,5,3,9,0,50,9,6,5,1,36,8,5,5,1/
4880       DATA DTHRSH/6.,2.7,.8,0.,.8,2.7,6.,2*0,
4890&      7.5,4.5,2.1,.6,0.,.6,2.1,4.5,7.5,
4900&      3.5,1.,0.,1.,3.5,4*0.,
4910&      3.2,.9,0.,.9,3.2,4*0./
4920       DATA DGAMMA/420.,565.,655.,685.,655.,565.,420.,2*0.,
4930&      581.,716.,817.,878.,900.,878.,817.,716.,581.,
4940&      3885.,4715.,5000.,4715.,3885.,4*0.,
4950&      1500.,2000.,3500.,2000.,1500.,4*0./
4960       DATA DSIGMA/9*0.,9*0.,3145.,1670.,0.,-1670.,-3145.,4*0.,
4970&      1500.,700.,0.,-700.,-1500.,4*0./
4980       DATA DLEN/44.3,40.7,8*0.,113.,39.,24.,24.,6*0.,
4990&      77.,44.,11.,-22.,-55.,-88.,4*0.,52.,24.,0.,-28.,-65.,5*0./
5000       DATA DMASS/.27,.27,4*0.,1.191,2.08,2.05,3*0.,6*0.,6*0./
5010       DATA DFMASS/2.58,18.8,0.,0./
5020       DATA DINRTA/3282.,90876.,0.,0./
5030       DATA DDRVLN/0.,0.,25.,25./
5040      DATA DVAR/-1.17069,.00076,-.81339,-.84536,5*0.,
```

VDPROG

```
&      -2.627,.006,-1.038,-1.552,-1.658,4*0.,
&      -5.79,-.0089,-.966,-.97,-.942,-.913,-.884,-.856,0.,
&      -3.75,-.0087,-.76,-.78,-.76,-.73,-.68,2*0./
       DO 10 I=1,2
 10    VEHQID(I)=DVEHCL(I,N)
       DO 20 I=1,5
 20    SETUP(I)=DSETUP(I,N)
       DO 30 I=1,NSEGS
       THRESH(I)=DTHRSH(I,N)
       SIGMA(I)=DSIGMA(I,N)
 30    GAMMA(I)=DGAMMA(I,N)
       DO 40 I=1,IDF
       LEN(I)=DLEN(I,N)
 40    VAR(I)=DVAR(I,N)
       DO 50 I=1,NAXLES
 50    MASS(I)=DMASS(I,N)
       FMASS=DFMASS(N)
       INRTIA=DINRTA(N)
       DRVLEN=DDRVLN(N)
       RETURN
       END
       SUBROUTINE M151(FK)
       DIMENSION TEMP(2),FK(8)
C*****ALGEBRAIC UPDATE OF VARIABLES
C--SEGMENTED WHEEL INPUT
       DO10I=1,2
       FORCW(I)=0
       DO10J=1,7
       K=(I-1)*27+J
       II=I+2
       TEMP1=Y(K)-VAR(II)-THRESH(J)
       IF(TEMP1)20,30,30
 20    TEMP1=0
 30    FORCW(I)=FORCW(I)+GAMMA(J)*TEMP1
C--FORCW IS THE RESULTING WHEEL FORCE
 10    CONTINUE
       TEMP1=SIN(VAR(2))
       SPDEF(1)=VAR(3)-VAR(1)-LEN(1)*TEMP1
       SPDEF(2)=VAR(4)-VAR(1)+LEN(2)*TEMP1
C--SPDEF IS THE SUSPENSION SPRING DEFLECTION
       TEMP1=VAR(6)*COS(VAR(2))
       DSPDF(1)=VAR(7)-VAR(5)-LEN(1)*TEMP1
       DSPDF(2)=VAR(8)-VAR(5)+LEN(2)*TEMP1
C--DSPDF IS THE SUSPENSION SPRING DEFLECTION VELOCITY
C*****COMPUTATION OF FRONT SUSPENSION FORCE [FORCK(1)]
 FORCK(1)=1500.*SPDEF(1)
 FORCK(2)= 1500.*SPDEF(2)
C*****COMPUTATION OF FRONT SUSPENSION DAMPING [DAMP(1)]
 DAMP(1)=42.*DSPDF(1)
 DAMP(2)=42.*DSPDF(2)
```

VDPROG CONTINUED

```
C *****DIFFERENTIAL EQUATIONS
C   FK(1&5)--VERT C-G MOTION
C   FK(2&6)--PITCH MOTION
C   FK(3&7)--AXLE1 MOTION
C   FK(4&8)--AXLE2 MOTION
          DO 11 I=1,4
     11   FK(I)=H*VAR(I+4)
          STEMP=0.
          DO 21 I=1,2
          TEMP(I)=FORCK(I)+DAMP(I)
          STEMP=STEMP+TEMP(I)
     21   FK(I+6)=H*(FORCW(I)-TEMP(I)-MASS(I)*386.)/MASS(I)
          FK(5)=H*(STEMP-FMASS*386.)/FMASS
          FK(6)=H*(LEN(1)*TEMP(1)-LEN(2)*TEMP(2))/INRTIA
          RETURN
          END
          SUBROUTINE M35(FK)
          DIMENSION TEMP(3),FK(10)
C ******ALGEBRAIC UPDATE OF VARIABLES
          FORCW(1)=0.
C--FRONT AXLE RESULTING FORCE [FORCW(1)]
          DO900I=1,9
          TEMP0=Y(I)-VAR(3)-THRESH(I)
          IF(TEMP0)910,900,900
    910   TEMP0=0.
    900   FORCW(1)=FORCW(1)+GAMMA(I)*TEMP0
          FORCW(2)=0.
C--SECOND AXLE RESULTING FORCE [FORCW(2)]
          DO920I=1,9
          TEMP0=Y(I+32)-VAR(4)-THRESH(I)
          IF(TEMP0)930,920,920
    930   TEMP0=0.
    920   FORCW(2)=FORCW(2)+GAMMA(I)*TEMP0
          FORCW(3)=0.
C--REAR AXLE RESULTING FORCE [FORCW(3)]
          DO940I=1,9
          TEMP0=Y(I+44)-VAR(5)-THRESH(I)
          IF(TEMP0)950,940,940
    950   TEMP0=0.
    940   FORCW(3)=FORCW(3)+GAMMA(I)*TEMP0
          U=(VAR(4)-VAR(5))/48.
          BETA=ATAN(U)
          DBETA=(VAR(9)-VAR(10))/(48.*(1.+U*U))
          TEMP1=SIN(VAR(2))
          TEMP2=SIN(BETA)
          TEMP3=COS(VAR(2))
          TEMP4=COS(BETA)
C--SUSPENSION SPRING DEFLECTION [SPDEF]
          SPDEF(1)=VAR(3)-VAR(1)-LEN(1)*TEMP1
          SPDEF(2)=VAR(4)-VAR(1)+LEN(2)*TEMP1-LEN(3)*TEMP2
```

DPROG CONTINUED

```
          SPDEF(3)=VAR(5)-VAR(1)+LEN(2)*TEMP1+LEN(4)*TEMP2
C--SUSPENSION SPRING RATE OF DEFLECTION [DSPDF]
          DSPDF(1)=VAR(8)-VAR(6)-LEN(1)*VAR(7)*TEMP3
          DSPDF(2)=VAR(9)-VAR(6)+LEN(2)*VAR(7)*TEMP3-LEN(3)*DBETA*TEMP4
          DSPDF(3)=VAR(10)-VAR(6)+LEN(2)*VAR(7)*TEMP3+LEN(4)*DBETA*TEMP4
C--FRONT SUSPENSION SPRING FORCE [FORCK(1)]
          IF(SPDEF(1)+4.4)420,425,425
     420  FORCK(1)=11771.43*SPDEF(1)+66714.3
          GOTO460
     425  IF(SPDEF(1)+3.65)430,435,435
     430  FORCK(1)=3333.33*SPDEF(1)+7986.65
          GOTO460
     435  IF(SPDEF(1)-3.65)440,445,445
     440  FORCK(1)=1145.2*SPDEF(1)
          GOTO460
     445  IF(SPDEF(1)-4.4)450,455,455
     450  FORCK(1)=3333.33*SPDEF(1)-7986.65
          GOTO460
     455  FORCK(1)=11771.43*SPDEF(1)-66714.3
     460  CONTINUE
C--REAR AXLES SUSPENSION SPRING FORCES [FORCK(2&3)]
          DO510I=2,3
          IF(SPDEF(I)+5.7)470,475,475
     470  FORCK(I)=46000.*SPDEF(I)+243800.
          GOTO510
     475  IF(SPDEF(I)+5.1)480,485,485
     480  FORCK(I)=9333.33*SPDEF(I)+34800.
          GOTO510
     485  IF(SPDEF(I)-5.1)490,495,495
     490  FORCK(I)=2509.8*SPDEF(I)
          GOTO510
     495  IF(SPDEF(I)-5.7)500,505,505
     500  FORCK(I)=9333.33*SPDEF(I)-34800.
          GOTO510
     505  FORCK(I)=46000.*SPDEF(I)-243800.
     510  CONTINUE
C--FRONT SUSPENSION DAMPING [DAMP(1)]
          IF(DSPDF(1)+.6)520,525,525
     520  DAMP(1)=70.*DSPDF(1)-800.
          GOTO540
     525  IF(DSPDF(1)-.6)530,535,535
     530  DAMP(1)=1402.*DSPDF(1)
          GOTO540
     535  DAMP(1)=40.*DSPDF(1)+820.
     540  CONTINUE
C--REAR SUSPENSION SPRINGS DAMPING [DAMP(2&3)]
          DO570I=2,3
          IF(DSPDF(I)+.6)550,555,555
     550  DAMP(I)=-950.
          GOTO570
```

VDPROG CONTINUED

```
555   IF(DSPDF(I)-.6)560,565,565
560   DAMP(I)=1583.*DSPDF(I)
      GOTO570
565   DAMP(I)=950.
570   CONTINUE
C******DIFFERENTIAL EQUATIONS
C  FK(1&6)--VERT C-G MOTION
C  FK(2&7)--PITCH MOTION
C  FK(3&8)--AXLE1 MOTION
C  .   .   .   .   .
C  FK(5&10)-AXLE3 MOTION
      DO1I=1,5
1     FK(I)=H*VAR(I+5)
      STEMP=0.
      DO2I=1,3
      TEMP(I)=FORCK(I)+DAMP(I)
      STEMP=STEMP+TEMP(I)
2     FK(I+7)=H*(FORCW(I)-TEMP(I)-MASS(I)*386.)/MASS(I)
      FK(6)=H*(STEMP-FMASS*386.)/FMASS
      FK(7)=H*(LEN(1)*TEMP(1)-LEN(2)*TEMP(2))/INRTIA
      RETURN
      END
      SUBROUTINE M60(FK)
      DIMENSION TH(4),IY(6),FK(18)
C******ALGEBRAIC UPDATE OF VARIABLES
      DATAIY/4,12,20,29,37,45/
      DATATH/12.,10.,8.,6./
      HORMOM=0.
C--COMPUTATION OF VERTICLE [FORCW] AND HORIZONTAL [FORCH] FORCES
C   RESULTING FROM THE PROFILE INPUT [Y] TO THE SEGMENTED BOGIES.
      DO200I=1,6
      FORCH(I)=0.
      FORCW(I)=0.
      DO100J=1,5
      K=I+2
      L=IY(I)+J
      TEMP=Y(L)-VAR(K)-THRESH(J)
      IF(TEMP)10,20,20
10    TEMP=0.
20    FORCW(I)=FORCW(I)+TEMP*GAMMA(J)
100   FORCH(I)=FORCH(I)+TEMP*SIGMA(J)
C--SUSPENSION SPRING DEFLECTION [SPDEF]
      SPDEF(I)=VAR(1)+LEN(I)*VAR(2)-VAR(K)
C--MOMENT ABOUT THE C-G RESULTING FROM THE HORIZONTAL FORCES [HORMOM]
      HORMOM=HORMOM+FORCH(I)*(46.+SPDEF(I))
      KK=I+11
C--VELOCITY OF THE SUSPENSION [DSPDF] SPRING DEFLECTION
200   DSPDF(I)=VAR(10)+LEN(I)*VAR(11)-VAR(KK)
      VARFEL=0.
C--COMPUTATION OF FORCES FROM THE "FEELER"
```

VDPROG CONTINUED

```
        DO30I=1,4
        TEMP=Y(I)-TH(I)
        IF(TEMP)30,30,40
40      VARFEL=AMAX1(VARFEL,TEMP)
30      CONTINUE
C--SUSPENSION [FORCK] SPRING FORCE COMPUTATION
        DO700I=1,6
        IF(SPDEF(I)-.402)710,710,720
720     SPDEF(I)=.402
        DSPDF(I)=0.
710     IF(SPDEF(I)+12.)730,740,740
730     FORCK(I)=29998.*SPDEF(I)+339972.
        GOTO700
740     FORCK(I)=1667.*SPDEF(I)
700     CONTINUE
C--SUSPENSION SPRING DAMPING [DAMP]
        DO800I=1,6
        IF(ABS(DSPDF(I))-1.)810,820,820
810     DAMP(I)=2750.*DSPDF(I)
        GOTO800
820     DAMP(I)=SIGN(2750.,DSPDF(I))
800     CONTINUE
C--TRACK INTERCONNECTION FORCES [FORCT]
        TEMP=VAR(3)-VARFEL
        IF(TEMP)50,50,60
50      FORCT(1)=300.*TEMP
        GOTO70
60      FORCT(1)=0.
70      DO80I=2,6
        J=I+1
        K=J+1
80      FORCT(I)=375.*(VAR(K)-VAR(J))
C*****DIFFERENTIAL EQUATIONS
C   FK(1&10)--VERT C-G MOTION
C   FK(2&11)--PITCH MOTION
C   FK(3&12)--AXLE1 MOTION
C   .   .   .   .   .   .
C   FK(8&17)--AXLE6 MOTION
C   FK(9&18)--HORIZ C-G MOTION
        FORCT(7)=0.
        DO 11 I=1,9
11      FK(I)=VAR(I+9)*H
        FK(18)=0.
        FK(10)=0.
        FK(11)=0.
        DO 31 I=1,6
        TEMP=FORCK(I)+DAMP(I)
        FK(I+11)=H*(TEMP-FORCT(I)+FORCT(I+1)+FORCW(I)-1420.)*.2717
        FK(11)=FK(11)-LEN(I)*TEMP
        FK(18)=FK(18)+FORCH(I)
```

VDPROG CONTINUED

```
31    FK(10)=FK(10)-TEMP
      FK(10)=H*(FK(10)*.008-386.)
      FK(18)=H*FK(18)*.008
      FK(11)=H*(FK(11)-HORMOM)/581700.
      RETURN
      END
      SUBROUTINE M113(FK)
      DIMENSION TH(4),IY(5),FK(16)
C*****ALGEBRAIC UPDATE OF VARIABLES
      DATAIY/4,11,18,24,31/
      DATATH/12.,10.,8.,6./
      HORMOM=0.
C--COMPUTATION OF VERTICLE [FORCW], HORIZONTAL [FORCH], AND
C  MOMENT [HORMOM] FORCES RESULTING FROM THE PROFILE INPUT [Y]
      DO200I=1,5
      FORCH(I)=0.
      FORCW(I)=0.
      DO100J=1,5
      K=I+2
      L=IY(I)+J
      TEMP=Y(L)-VAR(K)-THRESH(J)
      IF(TEMP)10,20,20
10    TEMP=0.
20    FORCW(I)=FORCW(I)+TEMP*GAMMA(J)
100   FORCH(I)=FORCH(I)+TEMP*SIGMA(J)
C--SUSPENSION SPRING DEFLECTION [SPDEF]
      SPDEF(I)=VAR(1)+LEN(I)*VAR(2)-VAR(K)
      HORMOM=HORMOM+FORCH(I)*(46.+SPDEF(I))
      KK=I+10
C--SUSPENSION SPRING RATE OF DEFLECTION [DSPDF]
200   DSPDF(I)=VAR(9)+LEN(I)*VAR(10)-VAR(KK)
      VARFEL=0.
C--RESULTING DEFLECTION OF "FEELER" [VARFEL]
      DO30I=1,4
      TEMP=Y(I)-TH(I)
      IF(TEMP)30,30,40
40    VARFEL=AMAX1(VARFEL,TEMP)
30    CONTINUE
C--SUSPENSION SPRING FORCE AXLES 1&5 [FORCK(1&5)]
      DO 700 I=1,5
      IF(SPDEF(I)-.4)710,710,720
720   SPDEF(I)=.4
      DSPDF(I)=0.
710   IF(SPDEF(I)+2.7)730,740,740
740   FORCK(I)=740.74*SPDEF(I)
      GO TO 700
730   IF (SPDEF(I)+9.5)741,741,742
742   FORCK(I)=514.71*SPDEF(I)-610.28
      GO TO 700
741   IF(SPDEF(I)+9.6)743,744,744
```

VDPROG CONTINUED

```
743   SPDEF(I)=-9.6
744   FORCK(I)=15672.*SPDEF(I)+143384.
700   CONTINUE
      DO 2000 I=1,5
      IF(ABS(DSPDF(I))-7.42)830,840,840
830   DAMP(I)=316.71*DSPDF(I)
      GO TO 2000
840   DAMP(I)=SIGN(2350.,DSPDF(I))
2000  CONTINUE
C--TRACK INTERCONNECTION FORCES [FORCT]
      TEMP=VAR(3)-VARFEL
      IF(TEMP)50,50,60
50    FORCT(1)=300.*TEMP
      GOTO70
60    FORCT(1)=0.
70    DO80I=2,5
      J=I+1
      K=J+1
80    FORCT(I)=175.*(VAR(K)-VAR(J))
C*****DIFFERENTIAL EQUATIONS
C  FK(1&9)--VERT C-G MOTION
C  FK(2&10)-PITCH MOTION
C  FK(3&11)-AXLE1 MOTION
C  .    .    .    .    .    .
C  FK(7&15)-AXLE5 MOTION
C  FK(8&16)-HORIZ C-G MOTION
      FORCT(6)=0.
      DO 11 I=1,8
11    FK(I)=H*VAR(I+8)
      FK(9)=0.
      FK(10)=0.
      FK(16)=0.
      DO 31 I=1,5
      TEMP=FORCK(I)+DAMP(I)
      FK(I+10)=H*(TEMP-FORCT(I)+FORCT(I+1)+FORCW(I)-500.)*0.772
      FK(10)=FK(10)-LEN(I)*TEMP
      FK(16)=FK(16)+FORCH(I)
31    FK(9)=FK(9)-TEMP
      FK(9)=H*(FK(9)*.036-386.)
      FK(16)=H*FK(16)*.036
      FK(10)=H*(FK(10)-HORMOM)/68000.
      RETURN
      END
```

TABLE C2

```
1$NDM
10 COMMONSD
20 PRINT,"TAU,RMS,ALPHA,DESRMS,IX,NPTS"
30 READ,TAU,X,ALPHA,RMS,NS,N
40 SIGMAN=X*SQRT(1.-EXP(-2.*ALPHA*TAU))
50 PRINT,"SIGMAN=",SIGMAN
60 AA=EXP(-ALPHA*TAU)
70 SD=SIGMAN*SIGMAN
100 XBAR=0.
110 5I=0
120 SUM=0.
130 IX=NS
140 1CALLGAURND(V,IX)
150 V=V-XBAR
160 SUM=SUM+V
170 I=I+1
180 IF(N-I)1,2,1
190 2IF(XBAR)4,3,4
200 3XBAR=SUM/N
210 PRINT,"X-BAR AFTER GAUSS=",XBAR
220 GOTO5
230 4XBARAS=SUM/N
240 PRINT,"X-BAR AFTER SHIFT=",XBARAS
250 FACT=1
260 12Y=0.
270 IX=NS
280 SUM=0.
290 I=0
300 8CALLGAURND(V,IX)
310 I=I+1
320 V=V-XBAR
330 IF(I.EQ.1)V=0.
340 Y=Y*AA
350 Y=Y+V
360 X=Y*FACT
370 IF(FACT-1.)6,7,6
380 6WRITE(1,)X
390 7X=X*X
400 SUM=SUM+X
410 IF(I-N)8,9,8
420 9SUM=SUM/N
430 SUM=SQRT(SUM)
440 IF(FACT-1.)10,11,10
450 11FACT=RMS/SUM
460 GOTO12
470 10PRINT,"RMS=",RMS
480 PRINT,"FACT=",FACT
490 PRINT,"OUTPUT FILE NAME"
500 READ13,FN
510 CALLCLOSEF(1,FN)
```

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* US Army Tank-Automotive Command Warren, Michigan 48090	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

THE AMC '71 MOBILITY MODEL (VOLUME I AND II)

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)* Final Report. Summary Report Volume I, Appendices A, B and C Volume II.

5. AUTHOR(S) *(First name, middle initial, last name)*

The Staffs of the Surface Mobility Division (TACOM) and the Mobility and Environmental Laboratory (WES).

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1973	62 & 282 Appen.	20

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO. 1TA161102B52A	Report No. 11789
c. 1T662601A045	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d. 1T162112A046	LL-143

10. DISTRIBUTION STATEMENT

Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	US Army Materiel Command

13. ABSTRACT

This report presents the AMC '71 Mobility Model, a comprehensive computerized simulation of the interaction of a vehicle, a terrain and an operator. This model represents existing technology (as of 1971) for predicting the performance of wheeled or tracked vehicles across any type of terrain. While the model involves several simplifying assumptions necessitated either by lack of more complete information or by practical limitations on complexity and computer capacity, when used judiciously, it is a useful tool for ground mobility analysis even in its present form.

Following a brief introductory section, input requirements are discussed. Next is presented a narrative description of the model's structure including the simulation of dynamic effects and the crossing of areal terrain and linear terrains such as streams. The basic model output is shown to be a number of predicted speeds for a given single vehicle in each of a number of subunits of the terrains. Speeds in individual terrain subunits can be used for the development of various outputs depending on the needs of the user.

(cont'd on attached sheet)

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Mobility modeling						
Computer simulation						
Mobility						
Modeling						
Vehicle Mobility						
Vehicle Dynamics						
Soil trafficability						
Terrain analysis						
River crossing						
Obstacle crossing						

DD Form 1473:

<u>ABSTRACT</u>: (cont'd)

Principal restrictions and limitations of the model are given. Finally, two important applications are described in order to illustrate some of the possible uses of the model.

Appendix A contains the complete listing and definition of the necessary terrain input data. Appendix B includes the same for the vehicle inputs. Appendix C contains flow charts, program listings and the necessary background information in sufficient detail for a programmer to reproduce the AMC '71 Model.