RPI TECHNICAL REPOꓤT MP-75

INTERPRETATION OF LASER/MULTI-SENSOR DATA FOR SHORT RANGE TERRAIN MODELING AND HAZARD DETECTION

by

Brian S. Messing

A Study Supported by the

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

under

Grant NSG-7369

and by the

JET PROPULSION LABORATORY

under

Contract 954880

School of Engineering
Rensselaer Polytechnic Institute
Troy, New York

August 1980

CONTENTS

LIST OF TABLES AND FIGURES

ACKNOWLEDGEMENT

The author would like to express his deepest and sincere appreciation to Drs. Stephan W. Yerazunis and Dean K. Frederick for their supervision and support throughout this project.

Many special thanks are conveyed to Marjan Krajewski for his patience and guidance at the beginning of my association with the Mars Rover project. His information and suggestions were invaluable.

Finally a salute to all of those connected with the Mars program at Rensselaer without whom this paper would not have been possible.

This research was supported by the National Aeronautics and Space Administration under NSG-7369 and by the Jet Propulsion Laboratories under contract 954880.

ABSTRACT

An autonomous roving vehicle capable of exploring the surface of the planet Mars has been under construction and continual development at Rensselaer Polytechnic Institute. An improved multi-laser/multi-detector triangulation scheme has been proposed to collect data on the terrain in the immediate foreground of the planetary rover. The purpose of this report is to present a terrain modeling algorithm that would reconstruct the sensed ground images formed by the triangulation scheme, and classify as unsafe any terrain feature that would pose a hazard to the vehicle. This modeler greatly reduces quantization errors inherent in a laser/sensing system through the use of a thinning algorithm. Dual filters are employed to separate terrain steps from the general landscape, simplifying the analysis of terrain features. Finally, a crosspath analysis is utilized to detect and avoid obstacles that would adversely affect the roll of the vehicle. Computer simulations of the rover on various terrains examine the performance of the modeler.

PART 1

INTRODUCTION

Mechanical probes and remote sensing devices have long been used to explore regions that are beyond the immediate reach of man. This is true not only on the Earth's surface, but also in deep space research. The first lunar probes were unmanned and were mainly used to gather the initial data on the moons's surface. High powered Earth based telescopes and "fly-bys" were able to pick out major points of interest that a probe could explore. These early probes were stationary; once landed, they could not move.

Considerable effort has also been focused on the exploration of the planet Mars. Unfortunately, Mars is much further away from the Earth than the moon. This of course limits the resolution that may be obtained through telescopes for the purpose of positioning a martian probe. This would not be a problem if many probes were to be placed on the planet, however this would require a great deal of duplicity in the mission. Another alternative would be to have a few mobile vehicles that would roam the surface. Such a vehicle would not be limited to a single location; this eliminates one problem, but creates others.

There is a very large communication delay time (from nine to twenty-five minutes) between Mars and any Earth based control station. If the roving vehicle were dependent upon messages from the Earth, it might take days to move several meters, or more likely, the vehicle would fall into a crater before the command center could prevent such a disaster.

The vehicle then needs to be somewhat autonomous. While the extreme long range hazards could be avoided via the communications link, a short range detection system would also be a necessity. Such a short range detection system would make use of computing facilities on-board the rover itself, eliminating the dangerous time delay of the communications link.

The autonomous rover being developed at Rensselaer Polytechnic Institute uses a laser/phctodetector triangulation system as the short range sensor. A great deal of work has gone into the development of terrain modelers - algorithms that detect and catagorize possible hazards. All previous work in this field has concentrated on slope calculations using area analysis to detect hazards. While these methods do a fairly good job at detecting boulders, negative obstacles such a craters frequently go undetected unless they are very large. Sloping terrain across the path of the vehicle also presented problems to these modelers.

The subject of this report is the development of a new modeler using completely different techniques, breaking away from all earlier modelers. The current version of the modeler using these new concepts draws slightly from pattern recognition ideas, processing data from the sensors before any analysis is performed. Results of tests show vastly improved recognition of both positive and negative hazards; and the problem of crosspath slopes or slopes in any direction has been virtually eliminated.

PART 2

SYSTEM CONFIGURATION

The Mars Rover project has two major divisions, the hardware and software groups. The hardware group is divided into two smaller teams: one is responsible for all of the mechanical aspects of the rover, the other must develop and maintain the electrical components and subsystems of the vehicle. The software group is also subdivided into two teams, Realtime control and Simulation. As its name implies, the realtime group works closely with the hardware group and is responsible for the data received, and for the total control of the rover during an actual run of the vehicle. The simulation team has almost no physical contact with the vehicle, although all of the programs that analyze the incoming data and select the path of the vehicle are created by this group. Through the use of many sophisticated and complex mathematical models, runs of the rover are simulated on computers.

2.1 Mechanical Hardware Description

The heart of the rover (Figure 1) is a box that contains a microprocessor which controls wheel speeds, analyzes vehicle attitude, and interfaces with the telemetry link. All of the power sources for the vehicle and the gyroscopes are on this box. The box or body of the vehicle is supported by four wheels each independently powered by a 1/6 HP motor. In this configuration, the rover has a very good mobility. It has the ability to traverse level ground at approximately 0.2 meters

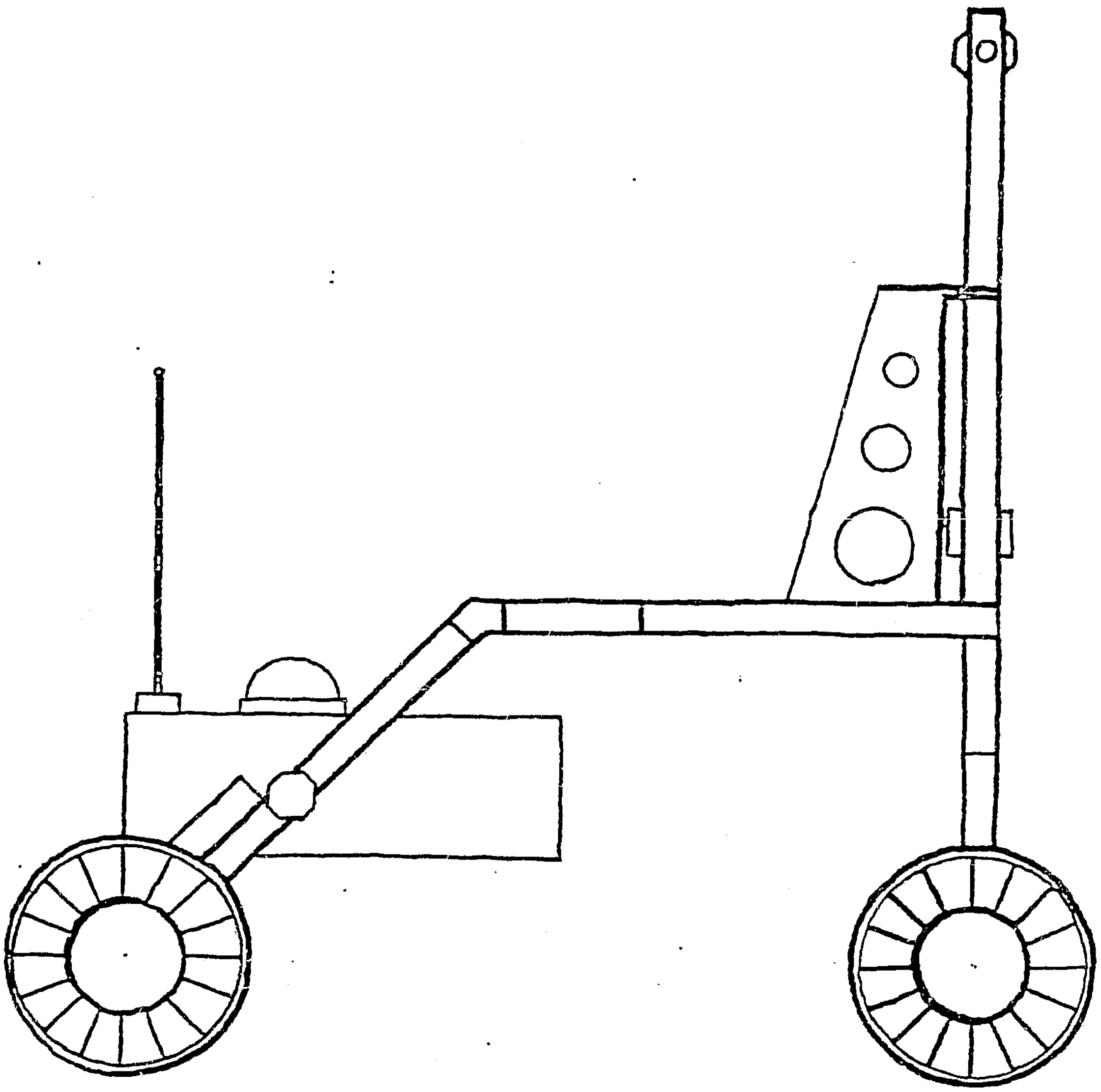


Figure 1. The RPI Mars Rover

per second and can negotiate thirty degree slopes. The wheels were designed for the rover; each is one-half meter in diameter. On level ground this permits the vehicle to handle 0.25 meter steps (on slopes this figure decreases until at a thirty degree slope no obstruction would be tolerated).

The front struts of the vehicle are joined to the main frame such that each of the front wheels may be raised or lowered individually; thus the front wheels do not significantly affect the "tilt" or roll of the vehicle. This is determined solely by the roll of the rear wheels.

Directly between the front wheels is located the elevation/scanning mast. This mast contains the lasers and sensors of the short range detection system. The mast rotates in a contra-clock-wise direction with a period of about two seconds. A mirror at the top of the mast also rotates in an upward direction making a complete revolution twelve times per second.

2.2 Electronic Hardware Description

The "eye" of the vehicle is the laser/sensing scheme located on the front center of the vehicle. All of the hardware associated with this scheme is known as the elevation scanning (laser/multi-detector) mast (Figure 2). At the very top of the mast is a rotating octahedrally sided mirror. This system simulates the firing of up to thirty-two lasers using only one laser and the mirror arrangement.

A gallium arsenide 100 watt laser is pulsed from below the mirror onto one of its faces. As the mirror turns,

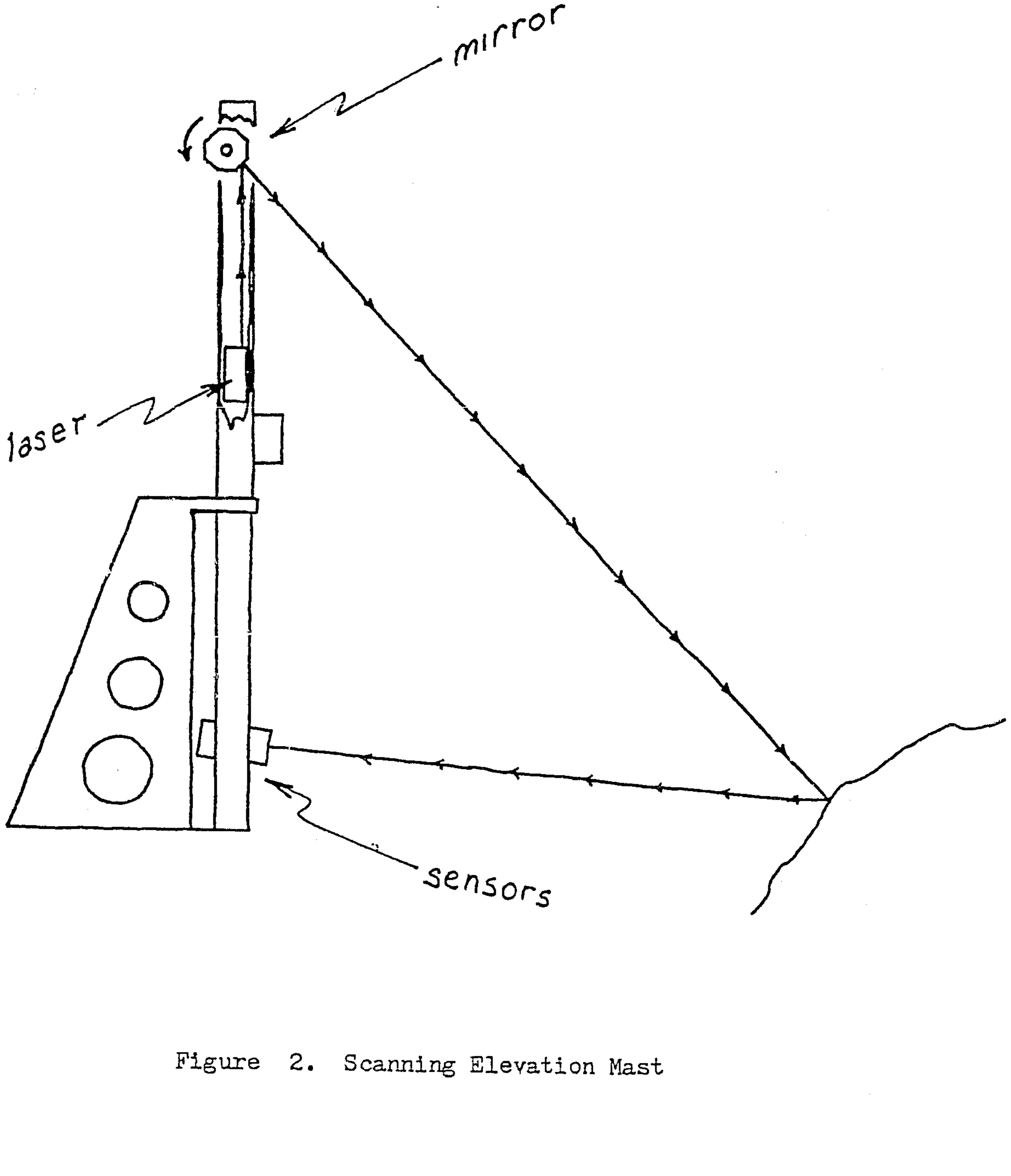


Figure 2. Scanning Elevation Mast

the incident angle of the beam and hence the reflected beam are altered. By controlling the speed of rotation of the mirror and timing the laser pulses, it is possible to fire "lasers" to sweep out thirty-two different elevation angles.

Near the base of the mast will eventually be fourty photo-detectors (there are twenty in place at the present time). The detectors are arranged such that they cover all of the terrain that would be swept out by the lasers on level ground.

While an actual Mars vehicle would have an on-board computer capable of analyzing the elevation scanning mast data, most of the computing power for the RPI rover is provided through the use of a two-way telemetry link to a Prime 650 computer. The terrain laser/sensor data, as well as all of the pertinent vehicle attitude and status information is sent to the Prime, and the vehicle commands are sent along the same link to the rover.

2.3 Realtime Software

The realtime software is concerned with getting the data in from the craft, analyzing it, and sending additional commands back to the rover. The data flows in along the telemetry link. This data is translated through an interface board and placed in buffers for use by other routines. The initial program is one of the most important since it has the direct link to the vehicle.

The data in the buffers is just what was sent by the vehicle: laser/sensor data and vehicle attitude information.

The laser/sensor data is used by the modeler. This is a routine that tries to recreate the terrain in front of the vehicle. Attitude information is used in both the modeler and the path selection algorithm. The output of the modeler and path selection algorithms are used to generate the commands that are sent back to the vehicle. Since the modeler and path selection algorithm are common to both realtime control and simulations, these will be examined in more detail in a later section.

2.4 Simulation Software

The simulation team has a difficult problem: it does not have a vehicle or terrain from which to get data. These must be "created" before anything else is attempted. The Simulator in its present state has evolved over many years and is quite complex. There are routines that create the terrain and simulate the vehicle and its laser/sensor system. Other programs move the "vehicle" over the terrain toward its target. The created landscape may have boulders, cliffs, craters, crevasses, hills, or blocks. The overall terrain itself may be level, sloped, or have a rolling base. In addition to this, noise may be added to corrupt various parameters, simulating rubble or small rocks that may litter the view. The simulated terrain may be made to be very similar to an actual martian landscape. The modelers and path selection algorithms used in realtime control are developed and tested on the Simulator.

2.5 Data Collection and Analysis Path

The new heading commands that are sent to the rover are generated within the path selection algorithm. The path selection system however consists of three separate and distinct subsystems; the data collector (sensing system), the modeler, and the path selection algorithm.

2.5.1 The Sensor

The sensor must employ some means of gathering information about the surrounding terrain. There are several approaches that may be used. A complete video system would be able to see everything, but analysis may prove to be much more difficult than a simple range finder. A video system would probably be suited for the long range hazard detection mentioned earlier. The RPI rover uses the laser/sensor scheme to gather the near field information. The hardware for this was described in section 2.2. The sensors have a set cone or field of vision which samples the line segment formed by the laser beam/sensor cone intersection. The laser/sensor intersection points for a system using thirty lasers and forty detectors are shown in Figure 3.

Analysis of such a system very quickly becomes complex. A three laser, four detector arrangement is shown in Figure 4 to illustrate how this system works. Each laser pulse intersects each detector cone once forming twelve line segments of intersection. If the terrain is present within any segment, it will scatter the laser pulse. The sensor as-

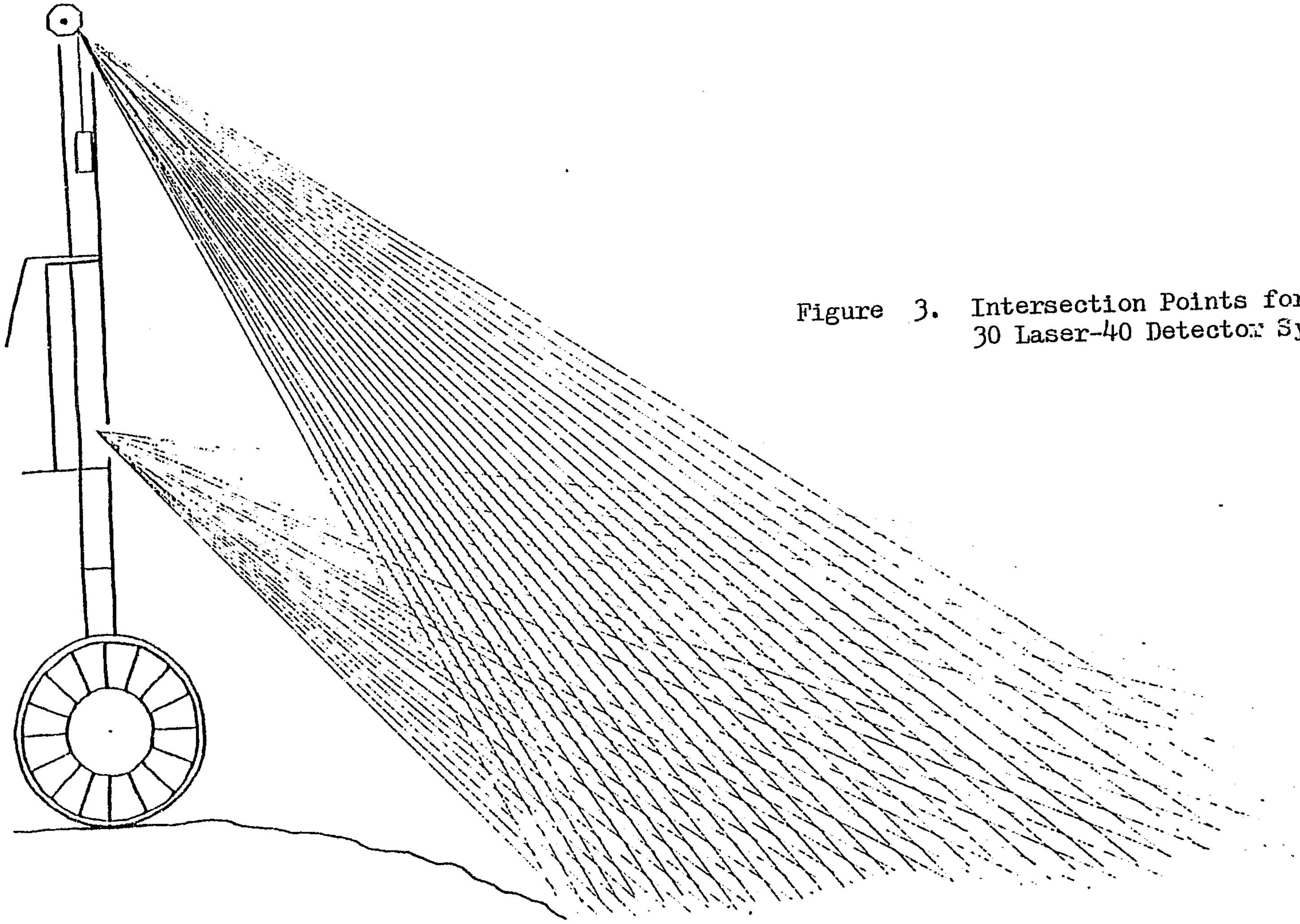


Figure 3. Intersection Points for a 30 Laser-40 Detector System

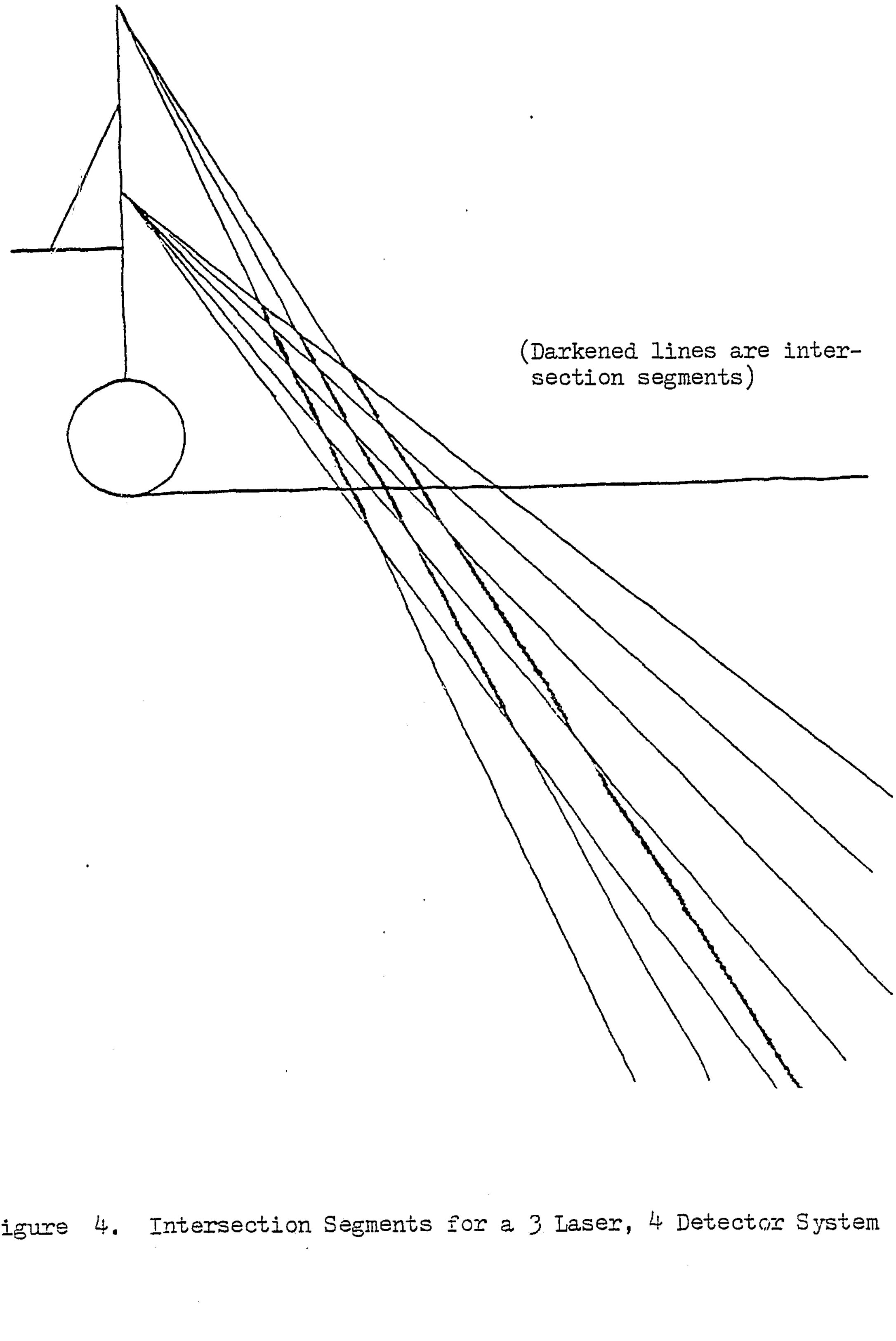


Figure 4. Intersection Segments for a 3 Laser, 4 Detector System

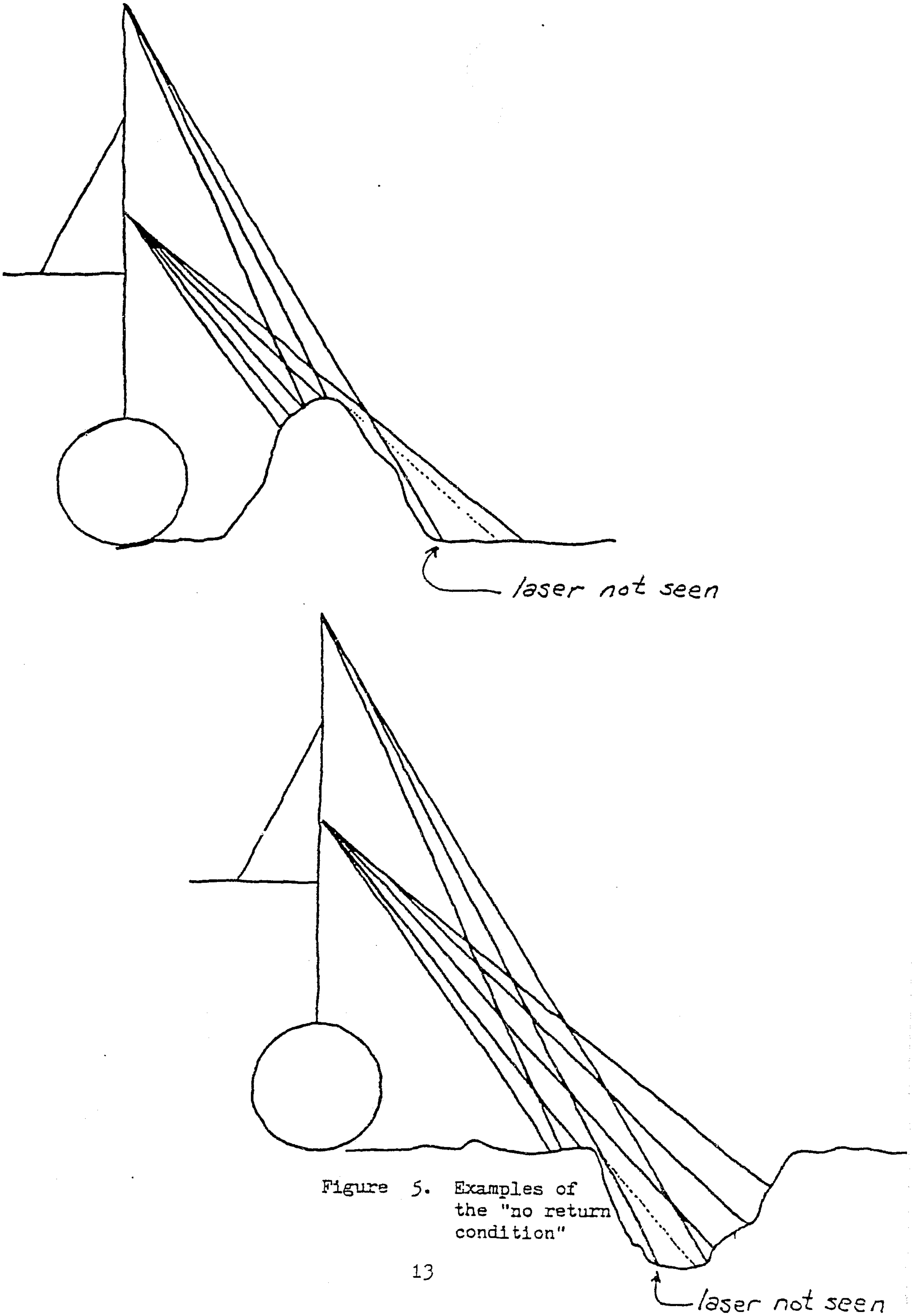


Figure 5. Examples of the "no return condition"

sociated with that segment will detect a portion of the energy from the diffracted beam, provided the path from the laser to the ground and back to the sensor is clear of other obstructions. Ideally, for each laser fired, one sensor will see it. Should none of the sensors detect a pulse, the path from the ground to the sensor must be blocked, and a "no return" condition results (Figure 5).

2.5.2 The Modeler

The data that comes from the sensor mast must then be analyzed. This data enters the terrain modeler - a special processor the "recreates" the landscape in the foreground of the rover. The modeler is perhaps the most important routine of the path selection process; if the terrain is reconstructed improperly, an obstacle may be overlooked, and an unsafe path for the rover to follow may be chosen. Alternatively, spotting hazards that are not present may force the vehicle to detour far from the desired course.

This reconstruction is not as simple as it first appears. In Figure 6, using the three laser, four detector system, several possible terrains are shown that would give identical returns to the sensor configuration used on the vehicle. It is the modeler's responsibility to incorporate vehicle attitude information with the appropriate hazard definitions into the sensor data. The output from the modeler would be the location of hazards to be avoided by the path selection algorithm.

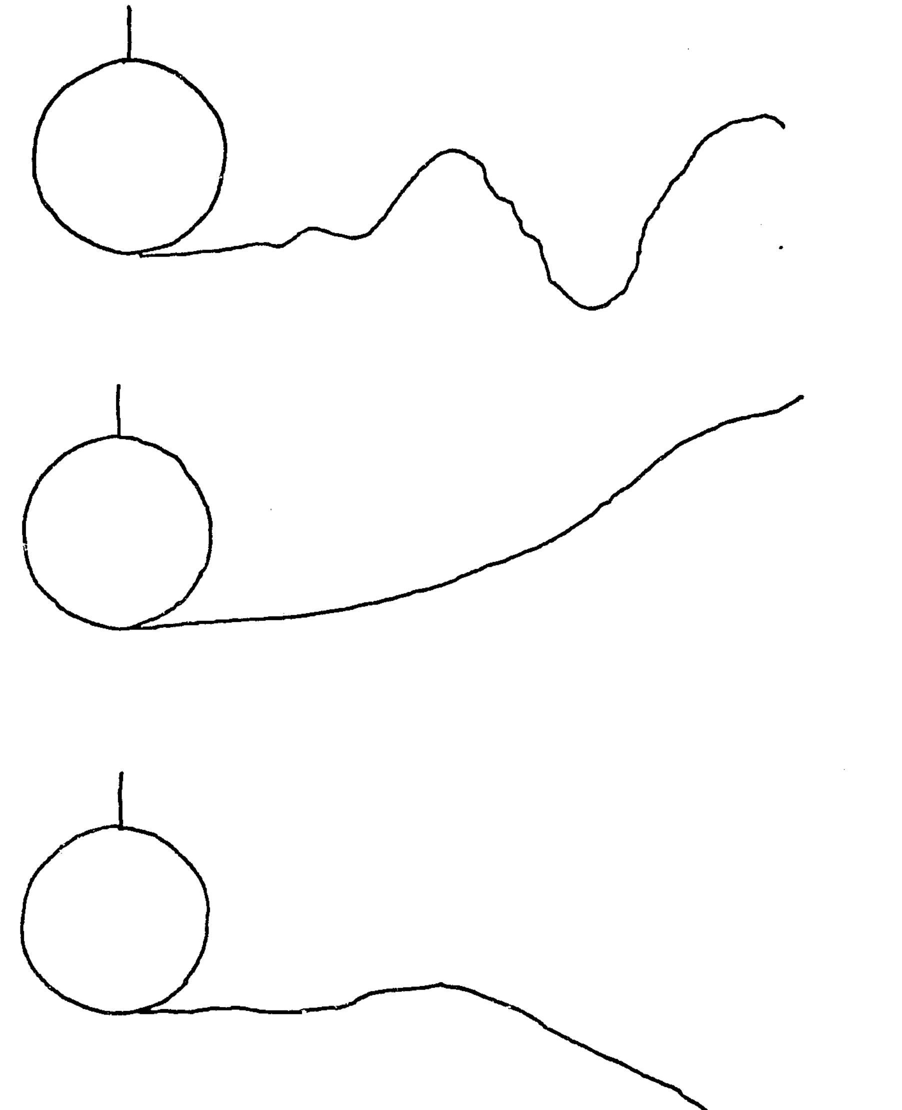


Figure 6. Samples of Terrain that give Identical returns to the Sensors

2.5.3 The Path Selection Algorithm

The path selection algorithm accepts as input the range and location of possible hazards. The path initially calculated is the direct line to the target from the present location of the vehicle - as if no obstacles are in its pathway. If the path were clear, this is the heading that the vehicle would take. If hazards are present, the selection of paths becomes much more difficult. The analysis of steering angles and rear wheel movement as well as other factors must be considered. Without going into detail, the path selection algorithm chooses a heading angle (if one exists) which is nearest the desired unblocked heading that avoids all of the flagged hazards.

While selecting a path, constraints are placed on the future available steering angles based on the location of noted hazards. Until well behind the vehicle, although these obstacles are out of sight, they pose a hazard to the vehicle. The constraints placed upon the steering angles prevent the rover from striking an obstacle located to the side of the vehicle when it is no longer visible.

PART 3

THE TERRAIN MODELER

3.1 Historical Review

In May of 1976 Marjan Krajewski submitted a report[1] that began the initial investigation into the development of a terrain modeler for the planetary rover. In this report three slightly different configurations were compared against each other: a single laser-single detector system, a two laser-single detector system, and a three laser-three detector system. His recommendations were to employ more lasers and sensors, include vehicle attitude information, and develop a more sophisticated pattern recognition scheme.

It was quickly recognized that indeed many lasers and detectors would provide more detail, but that this would require some very complicated interpretive software. The single laser-single detector system was installed on the vehicle for testing other concepts, meanwhile work was begun on better path selection algorithms as well as terrain modelers that would be able to handle a multi-laser/multi-detector system.

A report by Gary Maroon[2] in 1977 presented a numerical technique for the estimation of slopes. His paper was mathematical in nature; it did not present a terrain modeler, but layed the foundation for work that was to be done by Nick Troiani. Most of his work dealt with area-slope calculations; however obstacles such as boulders and craters wreaked havoc on his slope estimations. Early experiments indicated that the characteristics of different terrain features often overlap, creating difficulties in determining hazards.

1978 was the year that Troiani completed his work[3] on a usable modeler using Maroon's study as a framework. Certain calculations were simplified and improved; the sensor geometry was altered to form a "quasi-linearized" array of laser-sensor intersection points. This modeler detected most large positive (boulders and uphill slopes) obstacles, but did not do very well in the classification of negative hazards (craters), nor did it consider cross path obstacles.

Using the same slope equations used by the previous modelers, Erwin Hunter. refined Troiani's modeler and implemented a crosspath analysis to detect very large obstacles that would affect the roll of the vehicle[4]. Negative hazards were still not readily detected, slope estimates were good to within ten degrees, and slopes with obstacles continued to create problems for the area-slope manipulations. Although in the very near field (less than one meter) objects were easily identified, this did not leave the vehicle enough time to avoid small obstacles.

The modeler in this paper for the most part, discards all of the attempts made using the area-slope estimates and starts again from Krajewski's initial work. A thinning algorithm reduces the quantization errors that prevented previous modelers from identifying terrain steps. A dual first order filter separates the problem of slope and height estimates into two parallel paths of analysis. The crosspath detection algorithm eliminates the dangerous possibility of tipping the vehicle to either side.

3.2 Definition of Hazards

Before any further description of the modeler is presented, an examination of exactly what constitutes a hazard would be beneficial. In section 2.1, the wheels of the rover were described to be 0.5 meters in diameter. Because of this, steps greater than 0.25 meters in height are considered to be hazards. Steps of that height cannot be negotiated without forcing the wheels of the vehicle to travel only in a verticle direction (Figure 7).

The slope hazards have been set at thirty degrees for both uphill and downhill travel. The reasons for this depend upon the size of the motors used and the stability of the vehicle. Although more powerful motors could force the rover to negotiate the steeper slopes, the location of the center of mass would be altered, and the stability of the vehicle becomes very low. This would increase the chances of the rover tipping onto its side. If this were to occur on Mars, its mission would be essentially ended since in all likelihood, the vehicle could not right itself. The same reasoning for the inpath slopes also holds true for the crosspath slopes. Thus, the roll of the vehicle must be kept below thirty degrees. Enough of a safety factor has been built into these figures, such that, if these constraints are followed (0.25m step, 30° slopes) the rover should always be in a stable attitude.

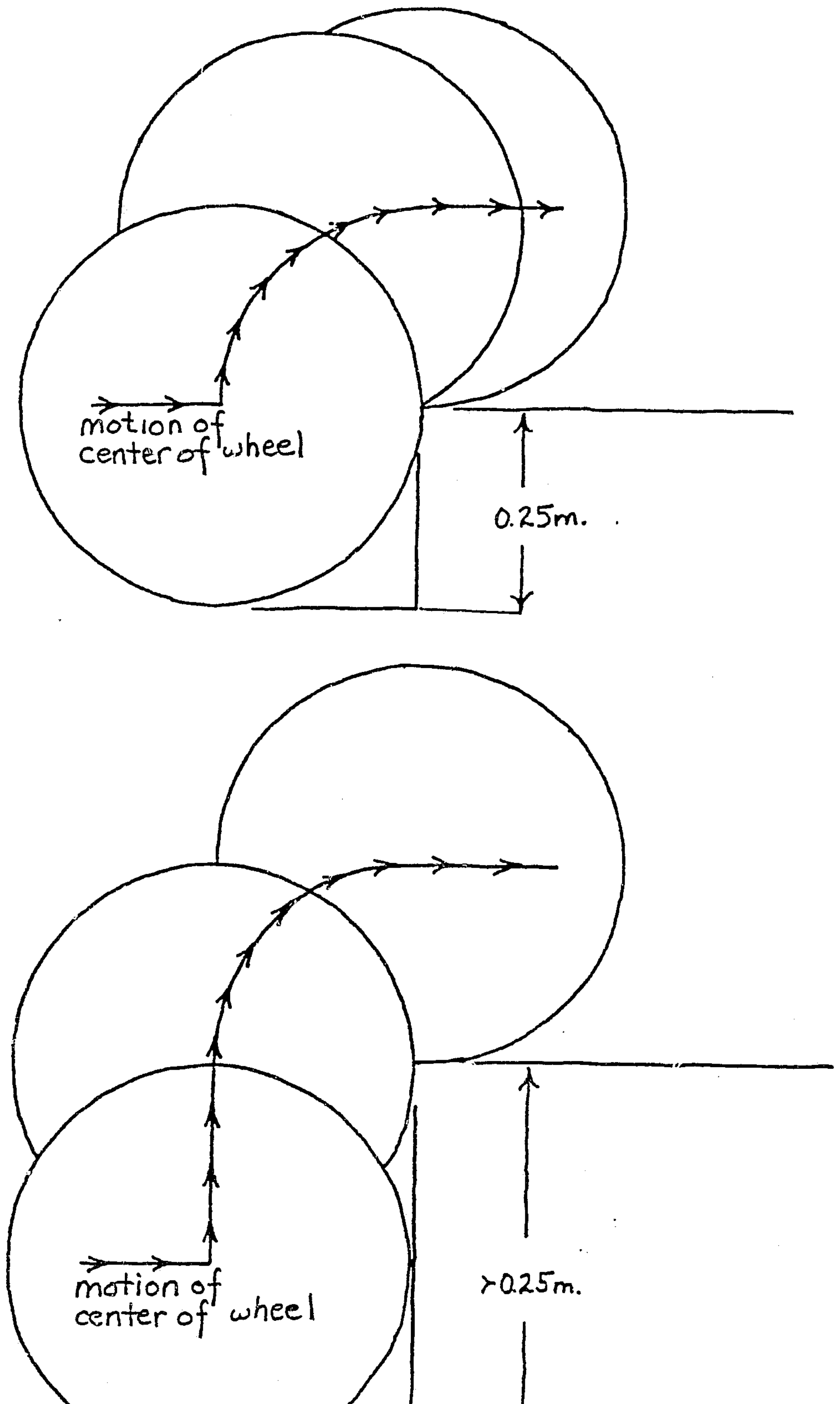


Figure 7. Rationale for Step Hazard Definition

3.3 Description of the New Modeler

The terrain data collected by the sensors is essentially in the form of height/range information. It can be seen from the previous examples of the terrain possible for one given set of returns, that this data is limited in accuracy by the finite dimensions of the intersection line segment formed by the laser beam and the detector's cone of vision. The accuracy of the return is dependent on several variables. As expected, the first of these is the range at which the intersection occurs. In general, the further from the vehicle that the terrain is observed, the larger the intersection segment. The other two variables that affect the accuracy are the separation distance between the laser and detector assemblies, as well as the detector's cone of vision. The job of the terrain modeler is to reduce these uncertainties from the laser/detector system to tolerable proportions, recreating the most probable terrain from the data.

3.3.1 The Filters

There are several types of hazardous terrain, each of these will be discussed in turn. The two major classifications of hazardous terrain are steps and slopes. This modeler recognizes that the steps and slopes can be distinguished by noting that steps would appear over only a few terrain samples. The data samples ordered in increasing range from the rover would have an almost constant height value until the step is encountered. At this point, the height of the terrain would suddenly shift to another value. Slopes on the other hand, would appear over many samples (long slopes would appear over the entire in-path scan). Each succeeding sample from the vehicle would have only a slight change from the previous return.

Recognizing this, the inherent inaccuracies of the individual data samples could be greatly reduced, while preserving the pertinent feature information by passing the height and slope data through separate filters. The estimated terrain height would be obtained by lightly filtering the data. That is, a quickly responding filter would be used to preserve the presence of step changes. The estimated slope of the terrain would be obtained by heavily straining (using a more slowly responding filter) all the data. This would prevent sudden steps or singularities from appearing in the slope estimates unless they were extremely large obstacles (in which case they would be easily spotted by the height analysis). This in itself eliminates a major source of error that was found in the earlier modelers. The output of each of these filters could then be examined for its own particular kind of hazard.

The data from the elevation scanning mast cannot be filtered as it is collected since the raw data is the same for both steps and slopes. Once in the modeler, the data can be filtered by different algorithms. The dual filtering concept is impemented using the following equations (beginning with the second data point in an azimuth):

$$\text{New Height Estimate} = (1-\alpha)X(\text{Current Sensor Sample Height}) + (\alpha)X(\text{Previous Height Estimate})$$

$$\text{Current Computed Slope} = \tan^{-1}\left[\frac{\text{New Height Estimate} - \text{Previous Height Estimate}}{\text{Current Sample Range} - \text{Previous Sample Range}}\right]$$

$$\text{New Slope Estimate} = (1-\beta)X(\text{Current Computed Slope}) + (\beta)X(\text{Previous Slope Estimate})$$

By controlling the parameters α and β (the height and slope filtering coefficients), the degree of filtering may be altered. The equations for both the height and slope filters are identical. In each case a value of 0.0 for the coefficient would essentially remove that filter from the data's path; there would be no filtering and each estimate would take on the current sample value. Conversely, if the coefficient were to have the value 1.0, perfect filtering would occur; the value of the initial sample would be given to each of the successive estimates.

These equations form the first order digital filters which effectively decouple the step and slope hazard identification problem. Although somewhat crude, they do an admirable job estimating the terrain features. Higher order filters were tried, but no real noticable difference was observed. Since these equations were more complicated, they required more calculations (and hence time). The additional calculations performed were deemed unnecessary and outweighed the questionable gain over the first order scheme: they were eliminated from further consideration.

3.3.2 Initial Processing

Before each run, a "look-up table" is generated of every possible laser/sensor cone mid-point intersection point; this table contains the height and range from the vehicle of each intersection point (Figure 8). Since it is not known where within the cone the terrain fell for any given sensor return, the mid-point is used to minimize the error (other modelers used either the top or bottom edge). This look-up table will be used in the modeler.

Every time a laser is fired, each sensor responds with a '0' (no detected pulse) or a '1' (pulse detected). Each laser has been assigned an identification number (from 1 to the number of lasers), as have each of the sensors. When a laser fires, the number of the laser as well as the number of the sensor that saw the return are encoded at the vehicle and sent to the computer. Thus, at the end of one azimuth, a one dimensional array has been formed: the rows are the number of a laser, while the sole entry of a row is the number of the sensor that saw that laser. At the end of an entire scan, there are as many of these one dimensional arrays as there were azimuths. If each of these arrays were placed next to each other, a two dimensional array would be formed: the rows again would be the number of a laser, the columns would be the number of an azimuth. An entry of the array in the ith row, jth column would be the number of the sensor that saw the ith

```
                                        SENSOR NUMBER
LASER NUMBER

               1        2        3        4        5        6        7        8        9       10
1  Z        -0.21    -0.12    -0.04     0.03     0.09     0.15     0.20     0.25     0.30     0.34
   R         1.10     1.05     1.01     0.98     0.95     0.92     0.89     0.87     0.85     0.83

              11       12       13       14       15       16       17       18       19       20
1  Z         0.37     0.41     0.44     0.47     0.50     0.53     0.56     0.58     0.60     0.63
   R         0.81     0.79     0.77     0.76     0.74     0.73     0.72     0.71     0.69     0.68

              21       22       23       24       25       26       27       28       29       30
1  Z         0.65     0.67     0.68     0.70     0.72     0.74     0.75     0.77     0.78     0.80
   R         0.67     0.66     0.65     0.64     0.64     0.63     0.62     0.61     0.61     0.60

              31       32       33       34       35       36       37       38       39       40
1  Z         0.81     0.82     0.84     0.85     0.86     0.87     0.88     0.90     0.91     0.92
   R         0.59     0.58     0.58     0.57     0.57     0.56     0.55     0.55     0.54     0.54

              41
1  Z         0.93
   R         0.53

               1        2        3        4        5        6        7        8        9       10
2  Z        -0.33    -0.23    -0.14    -0.06     0.01     0.08     0.14     0.19     0.24     0.29
   R         1.21     1.16     1.11     1.07     1.03     1.00     0.97     0.94     0.91     0.89

              11       12       13       14       15       16       17       18       19       20
2  Z         0.33     0.37     0.40     0.44     0.47     0.50     0.53     0.55     0.58     0.60
   R         0.87     0.85     0.83     0.81     0.79     0.78     0.76     0.75     0.74     0.73

              21       22       23       24       25       26       27       28       29       30
2  Z         0.62     0.65     0.67     0.69     0.70     0.72     0.74     0.76     0.77     0.79
   R         0.71     0.70     0.69     0.68     0.67     0.66     0.65     0.65     0.64     0.63

              31       32       33       34       35       36       37       38       39       40
2  Z         0.80     0.81     0.83     0.84     0.85     0.87     0.88     0.89     0.90     0.91
   R         0.62     0.62     0.61     0.60     0.59     0.59     0.58     0.58     0.57     0.56

              41
2  Z         0.92
   R         0.56

               1        2        3        4        5        6        7        8        9       10
3  Z        -0.48    -0.36    -0.25    -0.16    -0.08    -0.00     0.07     0.13     0.18     0.23
   R         1.34     1.28     1.22     1.17     1.12     1.08     1.05     1.01     0.98     0.96

              11       12       13       14       15       16       17       18       19       20
3  Z         0.28     0.32     0.36     0.40     0.43     0.47     0.50     0.52     0.55     0.58
   R         0.93     0.91     0.89     0.87     0.86     0.83     0.81     0.80     0.78     0.77

              21       22       23       24       25       26       27       28       29       30
3  Z         0.60     0.62     0.65     0.67     0.69     0.71     0.72     0.74     0.76     0.77
   R         0.76     0.74     0.73     0.72     0.71     0.70     0.69     0.68     0.67     0.66
```

Figure 8. Laser-Cone Intersection Points (Example)

LASER	SENSOR
30	29
29	29
28	28
27	27
26	27
25	26
24	25
23	24
22	24
21	23
20	22
19	21
18	20
17	20
16	19
15	18
14	17
13	16
12	15
11	14
10	13
9	12
8	11
7	10
6	9
5	8
4	7
3	6
2	4
1	3

One Dimensional Array Formed by One Azimuth

LASER	AZIMUTH																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
30	29	29	29	29	29	29	28	27	27	27	28	29	29	29	29	29	29
29	29	29	29	29	29	29	28	27	27	27	28	29	29	29	29	29	29
28	28	28	28	28	28	28	28	27	27	27	28	28	28	28	28	28	28
27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27	27
26	27	27	27	27	27	27	27	27	26	27	27	27	27	27	27	27	27
25	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26
24	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25
23	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
22	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
21	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23
20	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22
19	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21
18	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
17	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
16	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19
15	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18
14	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17
13	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
12	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15
11	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
10	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
9	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
8	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11
7	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
6	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
5	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
4	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
3	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
2	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

Two Dimensional Array Formed by One Entire Scan

Figure 9. Laser-Sensor Returns

laser along the jth azimuth (Figure 9).

The data from the sensors enters the modeler one azimuth at a time. Thus, the information in the first column of Figure 9 is examined before the second column. The modeler now knows which sensor saw which laser. From the look-up table, the approximate range and height of the terrain are found. The data points for an azimuth are then ordered in terms of increasing range from the vehicle.

Before proceeding through the dual filters, a thinning algorithm is used to eliminate closely spaced data points. This reduces the effect of quantization errors frequently encountered near terrain steps. This is especially important when the slope estimates will be made (Figure 10 and the slope filter equations).

A quick test for extremely large obstacles is then performed along each azimuth. If fewer than half of the fired lasers were observed by the sensors, or if the range of the furthest intersection point is less than 1.5 meters distant, there is clearly an unusual and most likely hazardous terrain feature present (the sensors are able to detect a laser pulse up to 5.0 meters away). Should this be the case, the entire azimuth is flagged as hazardous. If an appropriate number of sample points remain after the thinning algorithm, the dual filtering process is begun (a sample of raw data returns and the new data set formed by the filters is given in Figure 11).

The new data set from the height filters is examined solely for sudden steps in both the positive and negative dir-

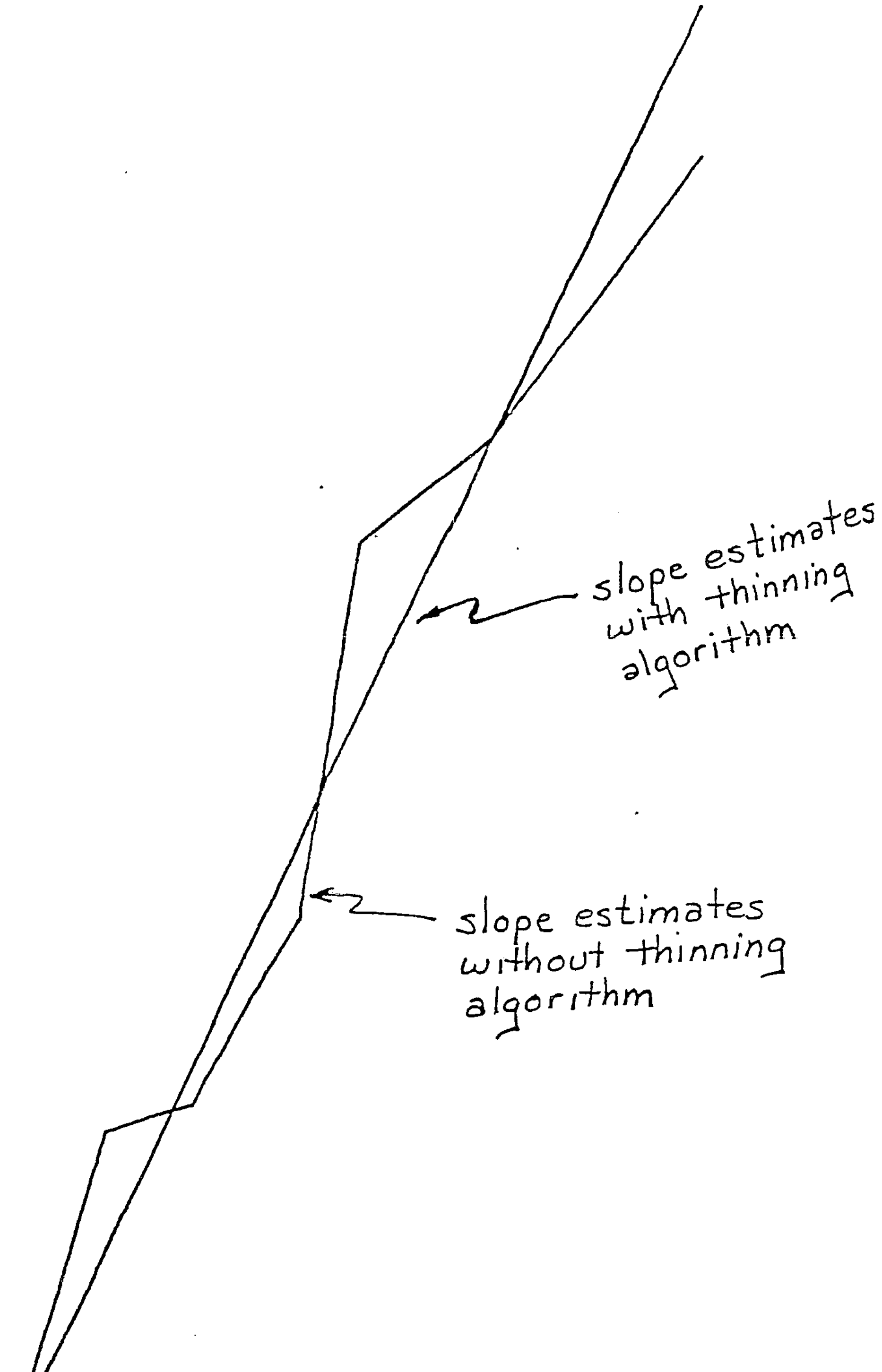


Figure 10. The Effect of the Thinning Algorithm

raw data

```
AZIMUTH =     0.0  DEGREES

SAMPLED TERRAIN POINTS:

RANGE:    1.00  1.05  1.07  1.12  1.17  1.22  1.27  1.33  1.33  1.34  1.34  1.34  1.38  1.38  1.47  1.51  1.55  1.60  2.29  2.35
HEIGHT:  -0.01 -0.02  0.03  0.02  0.01  0.00 -0.00 -0.01  0.13  0.19  0.25  0.31 -0.01  0.32  0.27  0.29  0.30  0.31 -0.03 -0.01

RANGE:    2.40  2.45  2.63  2.68  2.74  2.95
HEIGHT:   0.02  0.05 -0.02  0.01  0.04 -0.03
```

fitered data

```
TERRAIN ESTIMATES:

RANGE:    1.00  1.07  1.12  1.17  1.22  1.27  1.34  1.38  1.47  1.51  1.55  1.60  2.29  2.35  2.40  2.45  2.63  2.68  2.74  2.95
HEIGHT:  -0.01  0.02  0.02  0.01  0.00 -0.00  0.25  0.30  0.28  0.29  0.30  0.31  0.04  0.00  0.01  0.04 -0.01  0.01  0.04 -0.02
SLOPE:    0.9   2.4   1.8   0.7  -0.3  -1.0   6.6  11.3   8.7   8.5   9.1  10.1   6.9   2.9   4.1   6.4   4.2   5.5   8.0   5.8

  1 HAZARD CATALOGED ON THIS AZIMUTH

                                        HAZARD CATALOG

                    AZIMUTH          RANGE           X               Y
                     (DEG)            (M)

                      0.0            1.17           6.17            0.00
```

a hazard detected by modeler

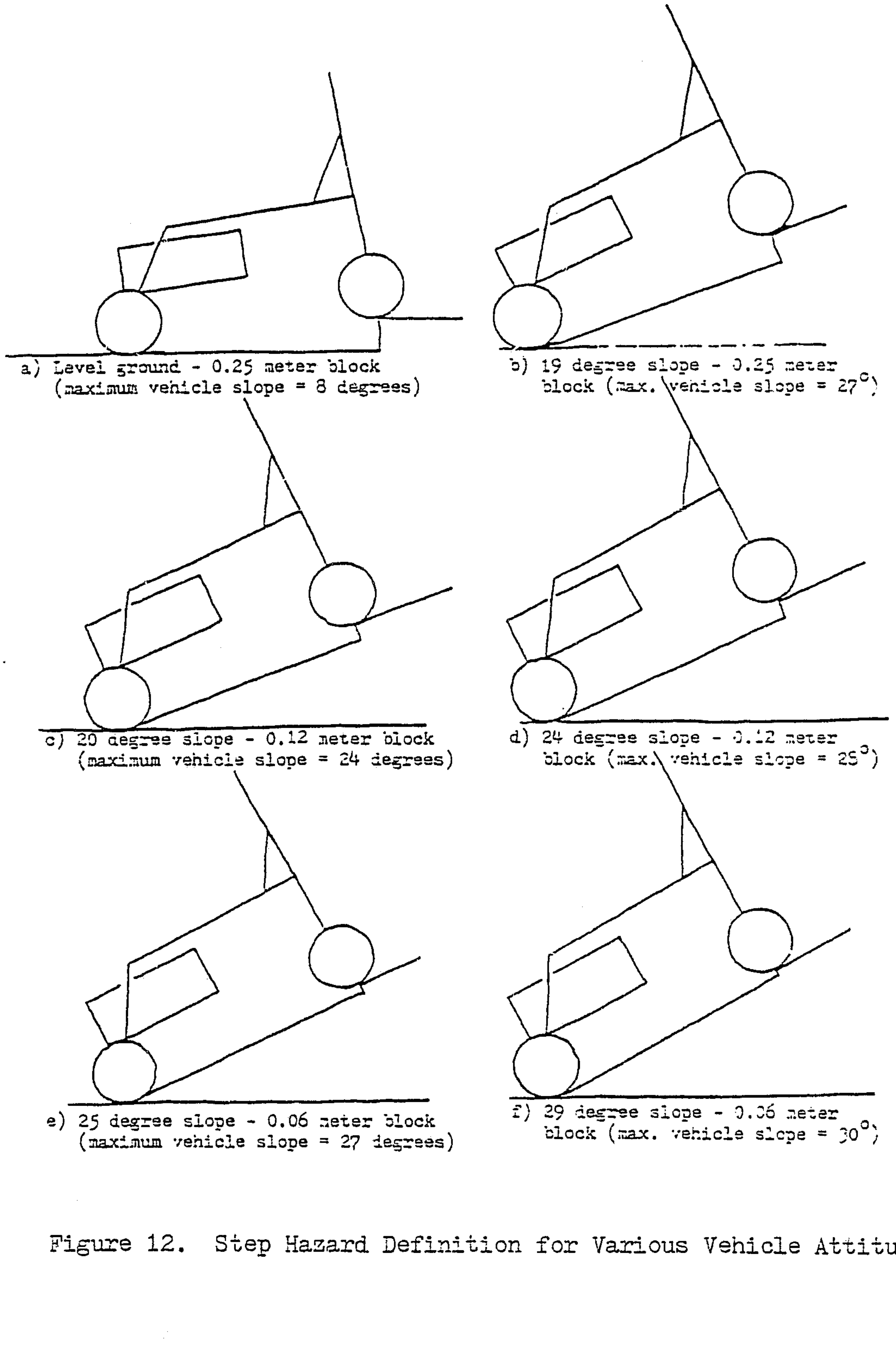
Figure 11. Sample of the Dual Filter (and Modeler) Output

ection. Any hazardous slopes that exceed the safe limits are detected from the output of the slope filter. Before the examination of either data set takes place, the attitude of the vehicle is taken into account.

3.3.3 Inpath Analysis

The attitude of the vehicle is used in the slope estimates to determine what the attitude of the vehicle would be if it were to travel along a given azimuth. The current vehicle attitude also affects the size of the step that the rover may be permitted to take. If the total inpath slope of the vehicle is less than 2/3 the maximum permitted (twenty degrees), then the full step size of 0.25 meters is traversable. If the vehicle is on a slope greater than twenty degrees, but less than twenty five degrees (5/6 maximum) steps of 0.12 meters may be safely handled. Vehicle attitudes of twenty-five to thirty degrees dictate a maximum step size of 0.06 meters. The rationale behind these changes in the hazard definition for steps may be seen in Figure 12.

Data returns from greater than 3.5 meters distant are ignored since the line segments of intersection begin to become large (review Figure 3). Any steps that are larger than the current hazard definition are flagged as dangerous, as are slopes that exceed the thirty degree definition. The azimuth and range at which the hazard is detected are stored in memory for further use.



a) Level ground - 0.25 meter block
(maximum vehicle slope = 8 degrees)

b) 19 degree slope - 0.25 meter
block (max. vehicle slope = 27°)

c) 20 degree slope - 0.12 meter block
(maximum vehicle slope = 24 degrees)

d) 24 degree slope - 0.12 meter
block (max. vehicle slope = 29°)

e) 25 degree slope - 0.06 meter block
(maximum vehicle slope = 27 degrees)

f) 29 degree slope - 0.06 meter
block (max. vehicle slope = 30°)

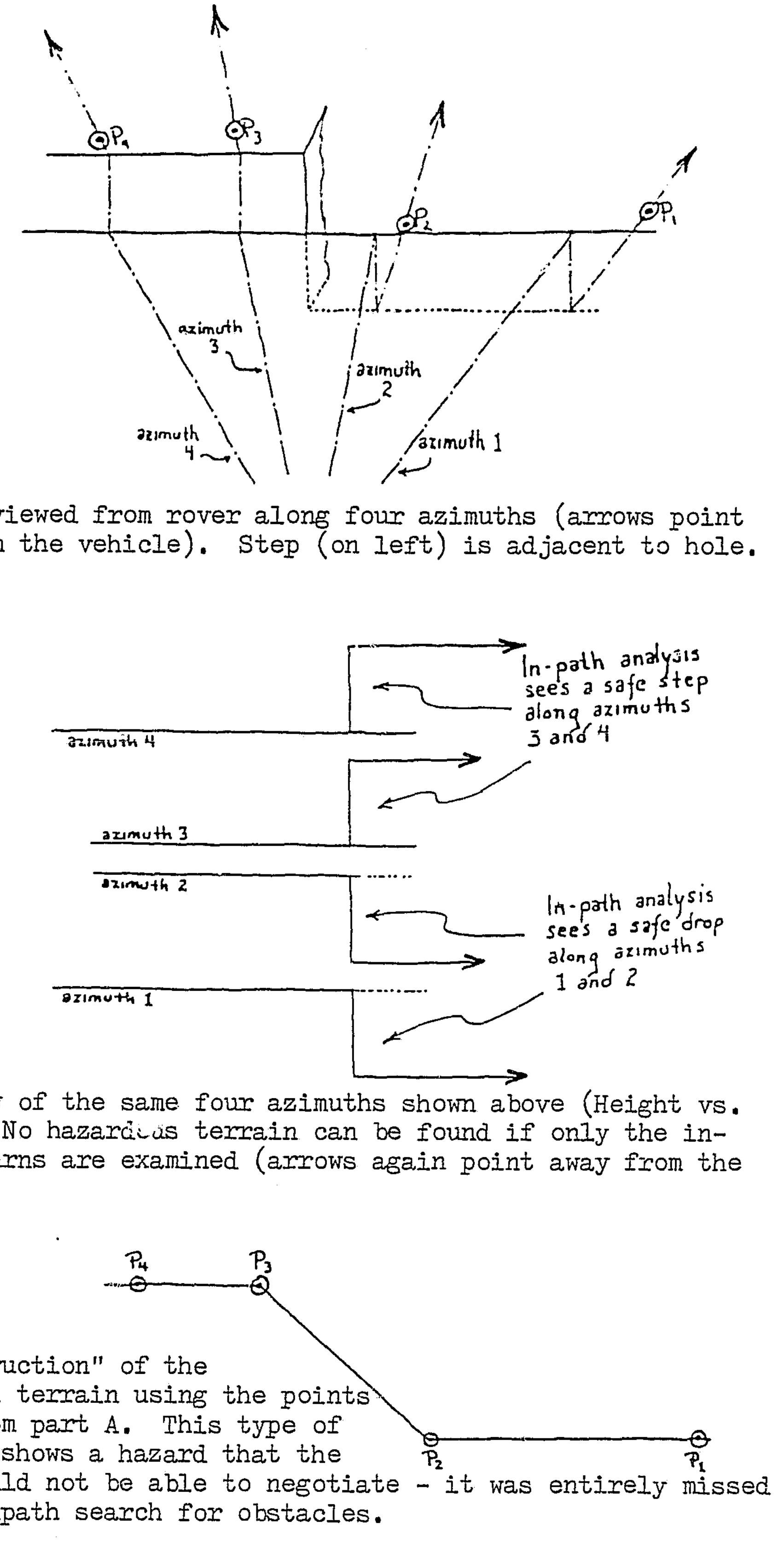
Figure 12. Step Hazard Definition for Various Vehicle Attitude

3.3.4 Crosspath Analysis

The modeler has an algorithm to determine crosspath hazards. All previous modelers had the ability to detect most hazards along an azimuth, but only one even attempted to check for crosspath obstacles. The modeler discussed here uses fewer calculations, is more sophisticated in its hazard determinations, and gets better results than the earlier attempt of crosspath analysis.

The problem of crosspath hazards can be clearly seen in Figure 13. The figure shows a positive step located adjacent to a hole. The inpath scans of the terrain are suggested by the azimuths 1-4. In the first two azimuths the terrain drops 0,20 meters at some distance from the rover. Azimuths three and four show a sudden jump of 0.20 meters at the same range from the vehicle. Each azimuth taken individually indicates that there is safe terrain ahead of the vehicle when, in fact, this is not the case. If the terrain were sampled just after the discontinuities and examined, the terrain as suggested by the crosspath would be reconstructed. Obviously should the rover elect to travel on this hill, it would be extremely unstable and would probably tip over.

Although exaggerated, this figure graphically shows the crosspath problem. As the data is being filtered, various data points are "picked off" for the cross path analysis. These points are spaced at approximately 0.25 meter intervals beginning at 1.25 meters from the vehicle along each azimuth. The overall analysis (both inpath and crosspath) forms a grid


P4
P3
P2
P1
azimuth 3
azimuth 2
azimuth 4
azimuth 1
In-path analysis sees a safe step along azimuths 3 and 4
azimuth 4
azimuth 3
azimuth 2
azimuth 1
In-path analysis sees a safe drop along azimuths 1 and 2
P4
P3
P2
P1

A) Terrain viewed from rover along four azimuths (arrows point away from the vehicle). Step (on left) is adjacent to hole.

B) Side view of the same four azimuths shown above (Height vs. Range). No hazardous terrain can be found if only the in-path returns are examined (arrows again point away from the rover).

C) "Reconstruction" of the crosspath terrain using the points P_1-P_4 from part A. This type of analysis shows a hazard that the rover would not be able to negotiate - it was entirely missed by the inpath search for obstacles.

Figure 13. Crosspath Hazard Illustration

as shown in Figure 14.

A careful examination of the crosspath slopes is of more importance than that of the inpath slopes since it is actually easier to tip the vehicle sideways than to tip it backwards. The analysis however is basically the same. The safe step size is massaged depending on the actual roll or crosspath slope of the vehicle.

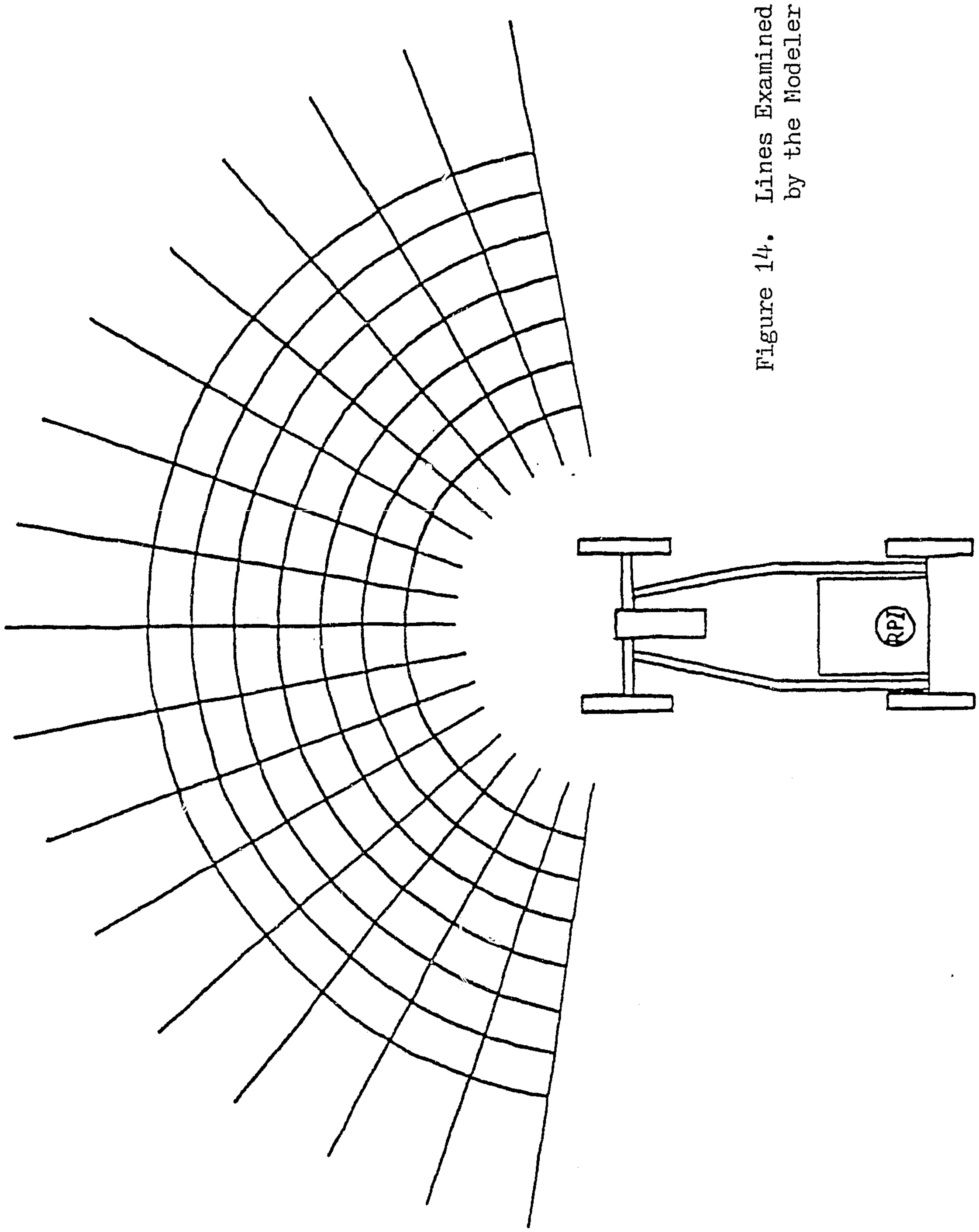


Figure 14. Lines Examined by the Modeler

PART 4

PROPOSED COMPUTER SIMULATIONS

Once the appropriate values of alpha and beta are selected, the results of this modeler are expected to be more descriptive of the terrain in the vicinity of the rover than any other previous modeling scheme. To test the abilities of the modeler, a series of computer simulations are to be run. The first phase of these runs are to be performed under "laboratory" conditions. Exactly how the rover approaches an obsracle will be under "manual control"; that is, no path selection algorithms are to be used. A predetermined course will be followed. The rover will traverse level ground only - there will be no pitch or roll. This type of environment will present repeatable results showing just how the terrain is being interpreted by the modeler.

Although the hazards have been predefined, the rover may not interpret the terrain as expected. From these initial tests, what the rover is seeing will be examined and the parameters of the filtering equations will be determined. The remainder of the runs will pose various problems to the interpretive abilities of the modeler over a range of simulated terrains. The path selection algorithm will be fully active during these more normal or "field" conditions. The rover will be expected to maneuver around obstacles placed in its path.

One of the major additions to the modeler is the capability to detect hazards that are across the path of the vehicle. The vehicle will be given different rolls and/or obstacles that must be examined for crosspath hazards. Having tested how the roll affects the vehicle, the pitch will then be tested using a sinusoidal terrain. The terrain will be designed such that it is always traversable. The most important reason for the existance of the modeler is of course the avoidance of obstacles such as boulders and craters. This then is to form the next series of simulations. Since the roll and pitch have essentially been tested individually, the vehicle is to have both a pitch and a roll during some of these runs. Finally, if all goes according to plan, all of the preceeding tests will be combined into several difficult, realistic terrain simulations. These will be designed to test all of the modeler and its integration with the path selection algorithm, detection, classification, and avoidance of all types of hazards on a surface comparable to the martian landscape. The entire testing sequence is shown in Figure 15.

TESTING SEQUENCE

	Simulations For:	Special Comments	Terrain Features
I	Determination of Filter Coefficients	"manual control"	Level ground, one boulder Level ground, one crater Level ground, positive slope Level ground, negative slope
II	Tests of the Crosspath Algorithm	vehicle will have roll (and possibly pitch)	Level ground, crosspath hazards Sloping ground, crosspath hazards
III	Rolling Terrain	vehicle will have pitch (and possibly roll)	Sinusoidal based terrain (no obstacles) Sinusoidal based terrain (small obstacles)
IV	Obstacle Avoidance	vehicle will have both a pitch and a roll	Level ground, boulder-crater fields Level ground, boulder-crater fields, "rubble"
V	Final Martian Terrains	vehicle will have both a pitch and a roll	Level ground, hills, slopes, boulders, craters, "rubble"

Figure 15. The Proposed Testing Sequence

PART 5

SIMULATION RESULTS

5.1 Determination of Alpha and Beta

The first phase of computer testing examined how the filtering equation parameters affected the performance of the modeler. In each of these tests, the rover would advance along a straight line toward its target. Located immediately behind the target was an obstacle of one particular type: a boulder, crater or slope. Since these obstacles were not in the pathway of the vehicle, they would never prevent the rover from reaching its destination.

At first glance it might appear that such tests have little value; looking at only the final output maps, they would indeed seem to serve no purpose. However, in many cases, the most useful data is not in these final maps, but from the intermediate data and decisions from the modeler or path selection algorithm. By setting certain flags, all of the intermediate data from the modeler was presented for observation. This information included all of the initial heights and ranges (of the laser/sensor intersection points), as well as the output from the dual filters. Careful study of this massive output of data permitted an indepth understanding of how the modeler "saw" the terrain feature as the rover neared the obstacle.

To examine how the height estimate depended upon alpha, the vehicle would approach a boulder of a safe height. Since the height parameter alpha is independent of the slope parameter beta, the latter was fixed to a single value for these tests. The rover would approach the same boulder along the same path for several simulator runs. Only the height parameter in the filtering equations was a variable. Following these simulations, a boulder that was considered unsafe for the rover to climb was substituted for the previous rock. Each of the earlier runs was then repeated. Examination of the intermediate data for all of these tests revealed a range of alpha for which the safe boulder would appear safe, and the unsafe boulder would stand out as a hazard.

The same procedure that was used for boulders was then applied to safe as well as unsafe craters. With alpha fixed to a single value, and beta as the variable, each of the above simulations was repeated for safe and unsafe slopes in both the uphill and downhill directions. Each of these groups presented a range for alpha or beta that would appropriately classify the hazard associated with that parameter.

The range of alpha for boulders was then compared to the range of alpha for craters. From this a smaller range of alpha was found that would properly classify the height obstacles. Similarly, the range of beta for uphill slopes was compared with that of the downhill slopes establishing another range for beta that would classify slopes. The values selected for alpha and beta from these final ranges were 0.25 and 0.90 respectively. These were used in all of the remaining simulations.

$\alpha = 0.1 \quad \beta = 0.8$

```
AZIMUTH =    0.0   DEGREES

SAMPLED TERRAIN POINTS:

RANGE:     0.92  0.96  1.05  1.08  1.16  1.24  1.32  1.33  1.40  1.47  1.54  1.61  1.78  1.84  1.91
HEIGHT:    0.15  0.23  0.22  0.31  0.31  0.32  0.33  0.44  0.47  0.50  0.54  0.58  0.53  0.58  0.64

TERRAIN ESTIMATES:

RANGE:     0.92  0.96  1.05  1.16  1.24  1.32  1.40  1.47  1.54  1.61  1.78  1.84  1.91
HEIGHT:    0.15  0.22  0.22  0.30  0.32  0.33  0.45  0.50  0.53  0.57  0.53  0.58  0.63
SLOPE:      0.0  12.6   9.5  15.0  14.1  13.3  22.3  23.8  24.6  26.0  18.4  21.2  25.1
```

same data

$\alpha = 0.1 \quad \beta = 0.9$

```
AZIMUTH =    0.0   DEGREES

SAMPLED TERRAIN POINTS:

RANGE:     0.92  0.96  1.05  1.08  1.16  1.24  1.32  1.33  1.40  1.47  1.54  1.61  1.78  1.84  1.91
HEIGHT:    0.15  0.23  0.22  0.31  0.31  0.32  0.33  0.44  0.47  0.50  0.54  0.58  0.53  0.58  0.64

TERRAIN ESTIMATES:

RANGE:     0.92  0.96  1.05  1.16  1.24  1.32  1.40  1.47  1.54  1.61  1.78  1.84  1.91
HEIGHT:    0.15  0.22  0.22  0.30  0.32  0.33  0.45  0.50  0.53  0.57  0.53  0.58  0.63
SLOPE:      0.0   6.3   5.4   8.5   8.8   8.9  13.8  15.4  16.7  18.1  15.1  16.9  19.3
```

ACTUAL SLOPE AT THIS POINT
IS 19.27 DEGREES

Figure 16. Beta's Effect on Slope Estimates. The same raw data was used for both runs, the slope was observed from exactly the same place. The terrain height estimates did not change since alpha did not vary. The final slope estimates for that azimuth are circled. Note how close estimate is for beta=0.9

5.2 Obstacle Classification

Using the chosen values of alpha and beta, a table of how the modeler sorts obstacles was established. These are presented in Table 1. The steps are discernable to within a few centimeters and slopes to within one or two degree. A few points should be clarified here concerning Table 1. A 0.24 meter boulder, or a 0.26 meter crater may occasionally be classified as unsafe or safe respectively. This depended upon its exact location as the rover approached. The fault of this is not really due to the modeler, but to the geometry of the laser/sensor system. Two solutions to this problem are possible: a) the rover may be slowed down to allow more scans of the obstacle as it was approached, or b) increase the number of lasers and sensors to get a clearer picture of the obstacle.

Although the unsafe craters were seen for each of these runs, invariably they would be noticed at a closer range than unsafe boulders of similar dimensions. This problem will never be alleviated; it is inherent in any type of terrain detection system. Consider the following analogy: a one meter fence post is located five meters from an observer. Even at that distance, a fairly good approximation could be made of its height. If instead of the post there was a posthole, the observer would have no idea what-so-ever what the depth of the hole might be. The observer would have to move toward the hole to get a feeling for its true depth. Although somewhat extreme, it illustrates the problem of detecting negative hazards.

BOULDERS		CRATERS	
HEIGHT	CLASSIFICATION	DEPTH	CLASSIFICATION
0.20	NO. HAZARD	0.20	NO HAZARD
0.22	NO HAZARD	0.24	NO HAZARD
0.24	OCCASIONAL HAZ.	0.26	OCCASIONAL HAZ.
0.26	HAZARD	0.28	HAZARD
0.30	HAZARD	0.30	HAZARD

POSITIVE SLOPES		NEGATIVE SLOPES	
ANGLE	CLASSIFICATION	ANGLE	CLASSIFICATION
25.0	NO HAZARD	-25.0	NO HAZARD
28.0	SLIGHT HAZARD	-28.0	HAZARD
32.0	HAZARD	-32.0	HAZARD
35.0	HAZARD	-35.0	HAZARD

Table 1.

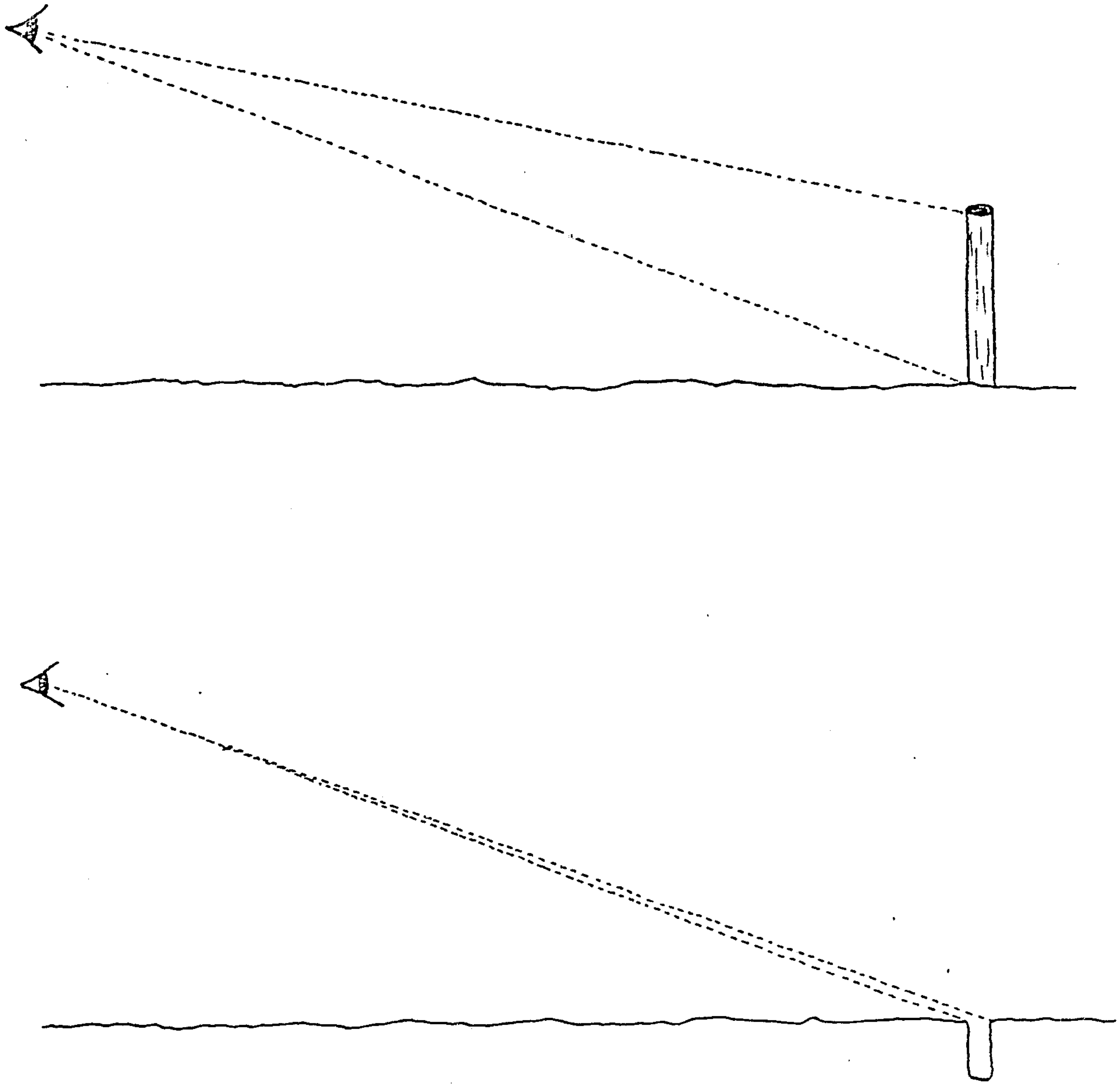


Figure 17. Post/Post-hole Analogy

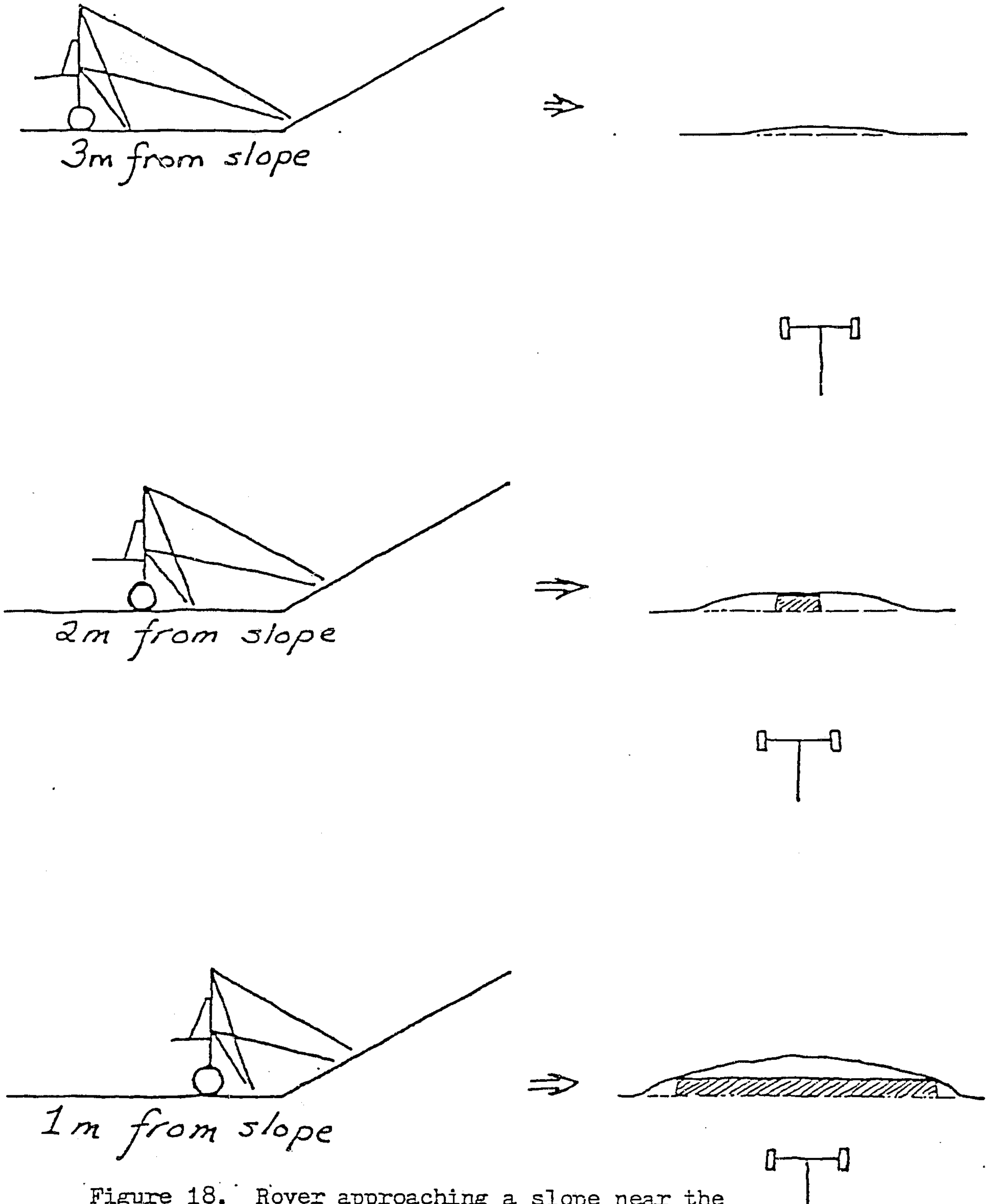


Figure 18. Rover approaching a slope near the hazard definition. With just slightly less information than that needed to determine if the slope is safe, the modeler recreates the terrain to the right - an apparent boulder (shaded area is the step hazard definition)

The classification of "slightly hazardous" terrain for the twenty-eight degree slope can be understood when the data from the sensors is examined as the rover approaches the slope (Figure 18). The terrain is seen as safe up to the final scan that the rover makes just before it gets on the slope. Since at this point it still appears as though there were a large boulder in its path, the modeler declares the path as unsafe, and the path selection algorithm moves to avoid the obstacle. This situation occurs only when the rover nears a sudden very long slope (or as will be seen later, extreme variations in the terrain). Short slopes with the same degree of incline pose no problem. Once entirely on the slope the modeler had no difficulty determining the actual terrain.

5.3 Crosspath Results

While it is clearly important for the rover to be able to see objects in its pathway, certain hazards may be overlooked if only one azimuth at a time is examined. These are collectively known as crosspath hazards. It is entirely possible to have safe steps along the inpath direction which are hazardous in a crosspath sense. Several examples of this problem are depicted in Figure 19. The data from the mast is taken along one azimuth at a time and is processed for inpath hazards. It is only after all of the azimuths have been examined (one full scan) that enough data exists for the crosspath analysis.

The next several pages contain simulation results of obstacles that would present crosspath hazards. Figure 20 is the "control" of Figure 19A. In this run, the ground is level; the rover is aimed toward its target in a manner such that its left wheels would roll into a crater and its right wheels would roll over a boulder if its path were not altered. Since the overall roll of the vehicle was less than the thirty degree hazard, the rover did not change its course and proceeded directly toward the target. If the overall roll of the vehicle was to become greater than thirty degrees along an azimuth (as would be the case if the boulder were higher and the crater deeper), a crosspath hazard would be observed. To avoid an unsafe attitude, the vehicle would be forced to alter its course (Figure 21).

In both of these figures, as in all other maps in this report, the location of the elevation mast (center of front wheels) has been highlighted by a heavy, double line. The motion of the rear wheels has been shown by a dotted and dashed line. Any obstacles have been outined in a single heavy line. A rough cross-sectional view of the terrain has been sketched at the bottom of each page. In order to bring attention to the obstacles and motion of the vehicle, some terrain information may have been painted out of the picture.

Another crosspath hazard was shown in Figure 19B. An initial control of this situation had the rover pointed toward its target with a 0.10 meter block in the path of its right wheels. The initial roll of the vehicle was 0.0 degrees (level ground). As expected, the rover had no difficulty neg-

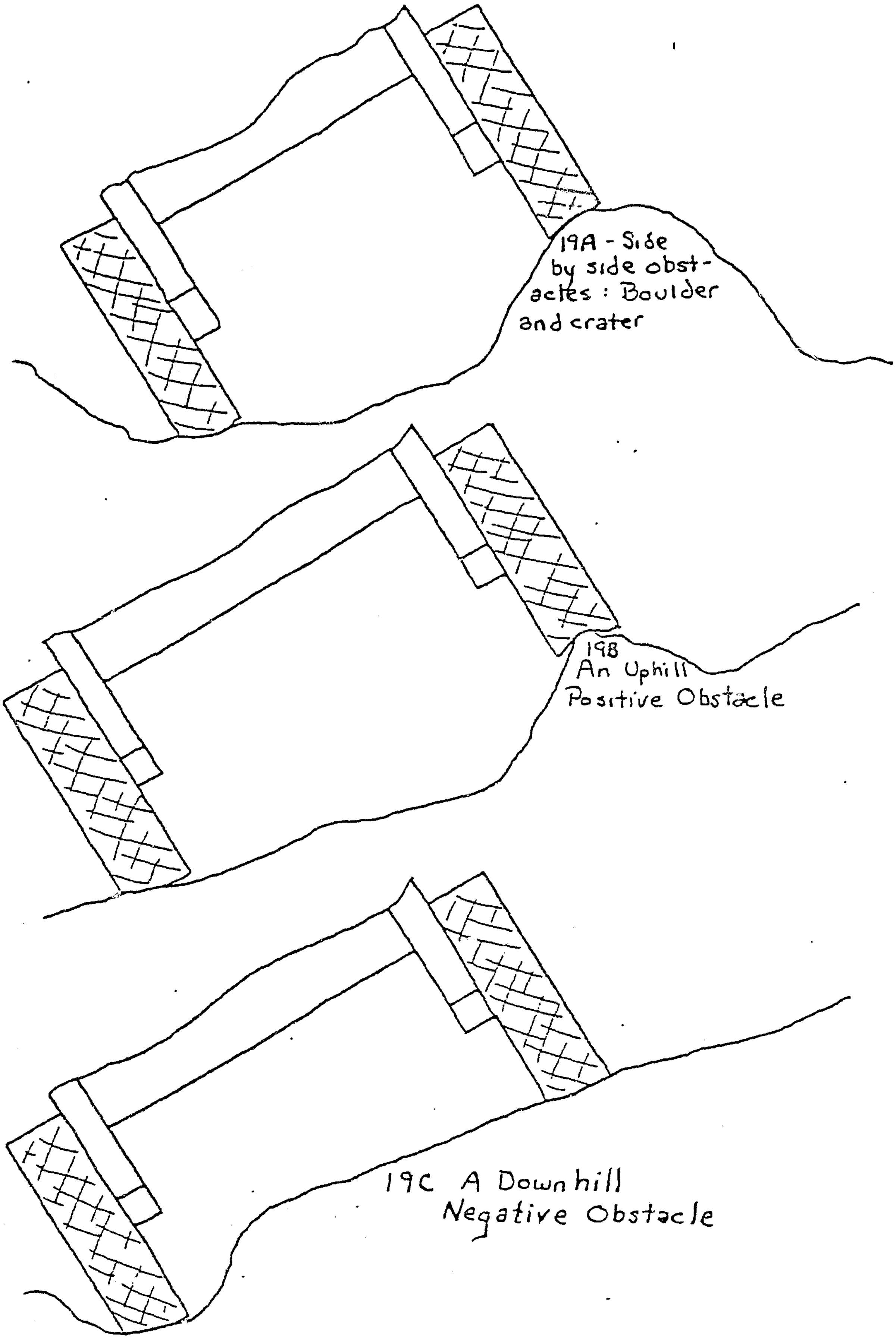

19A - Side by side obstacles : Boulder and crater
19B An Uphill Positive Obstacle
19C A Downhill Negative Obstacle

Figure 19. Crosspath Hazards

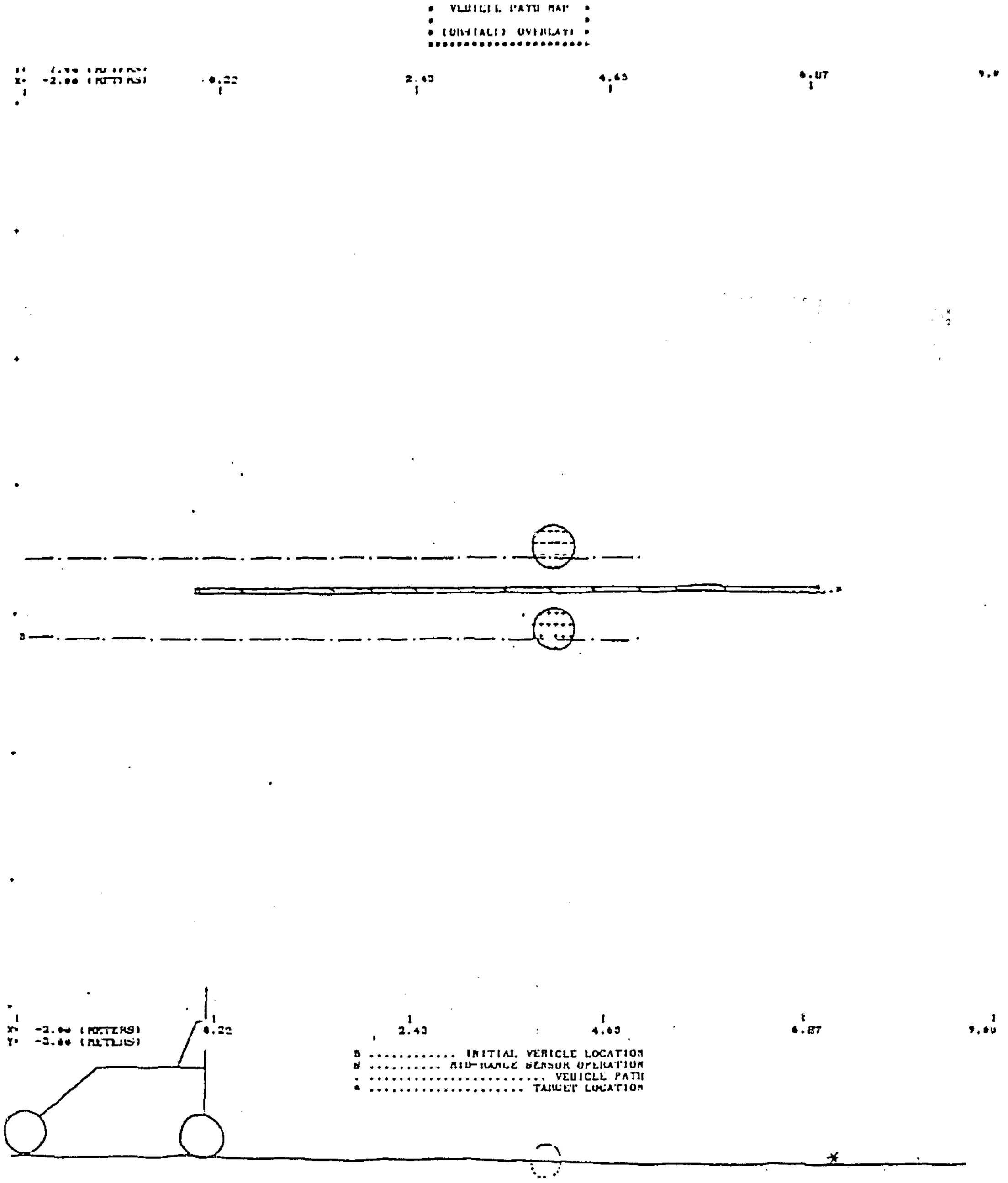


Figure 20. Near Crosspath Hazard of Side-by-Side Obstacles

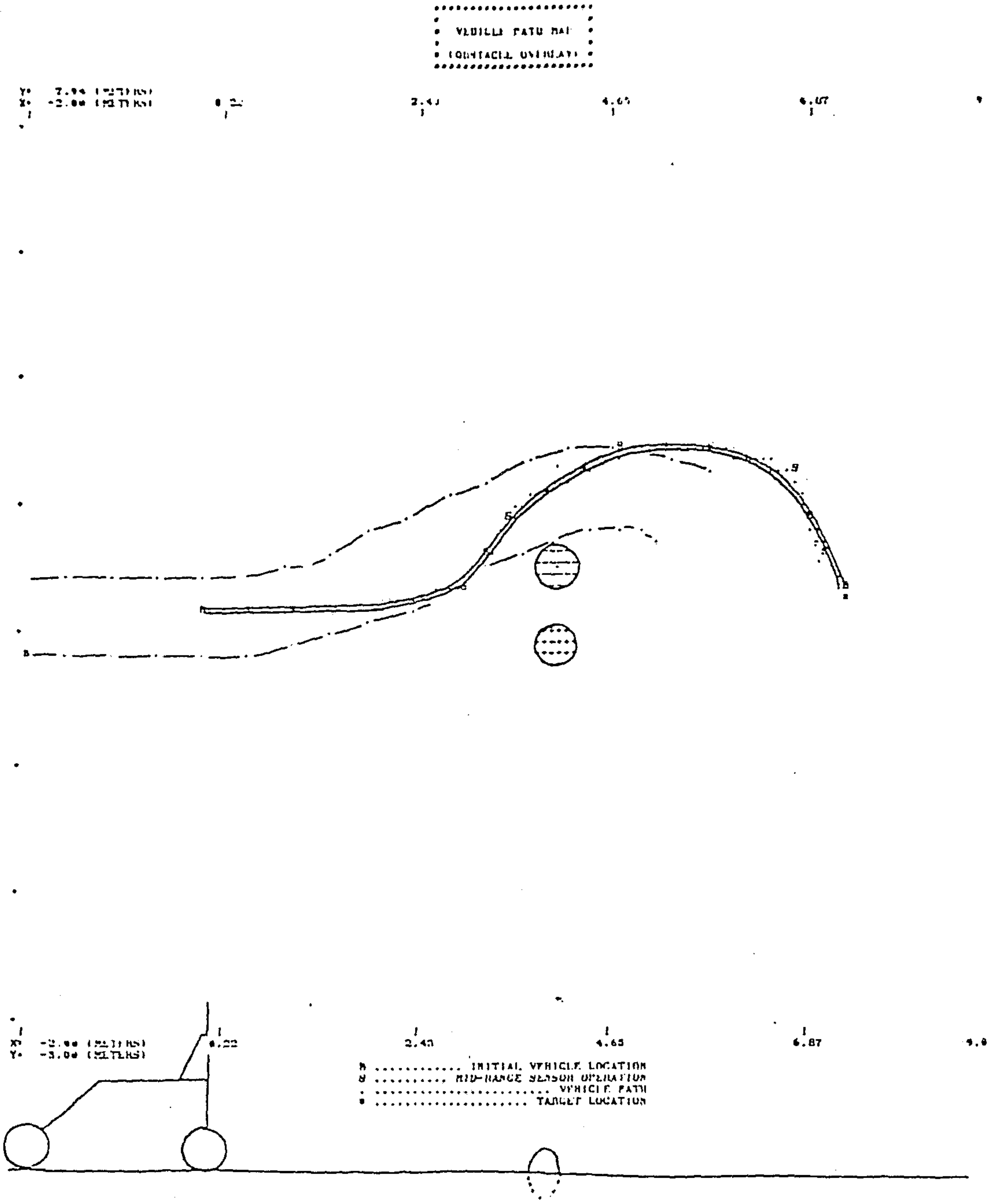


Figure 21. Crosspath Hazard of Side-by-Side Obstacles

otiating the block (Figure 22). This experiment was repeated with the rover on a hillside such that its initial roll was twenty degrees (Figure 23). Again since the overall roll of the vehicle was kept below the hazard level, the rover was able to move directly toward the target. It was only when the vehicle would have had to take on an unsafe roll that the rover was forced to detour around the block (Figure 24).

Although not explicitly shown, it should be clear that any crosspath slope less than that of Figure 24 would be classified as safe unless the obstacles themselves were larger; this would once again force the rover into an unsafe attitude. The crosspath analysis works equally well for slopes in the other direction, and obstacles do not have to be positive steps (Figure 19C).

5.4 Sinusoidal Terrain

Most of the terrain on the planet Mars is not going to consist of perfectly level ground, or abrupt smooth slopes. Instead, some type of rolling, varying hillside witn obstacles is more apt to be found. It is important to know how the rover would handle this type of terrain, or more specifically, how the modeler interprets the data received when the vehicle has a pitch and roll. The next series of simulations was designed to show what can happen if the terrain becomes too rolling or hilly. In these runs the rover will travel over a sinusoidally varying terrain base toward its target. The only variable will be the amplitude of the sinusoidal base.

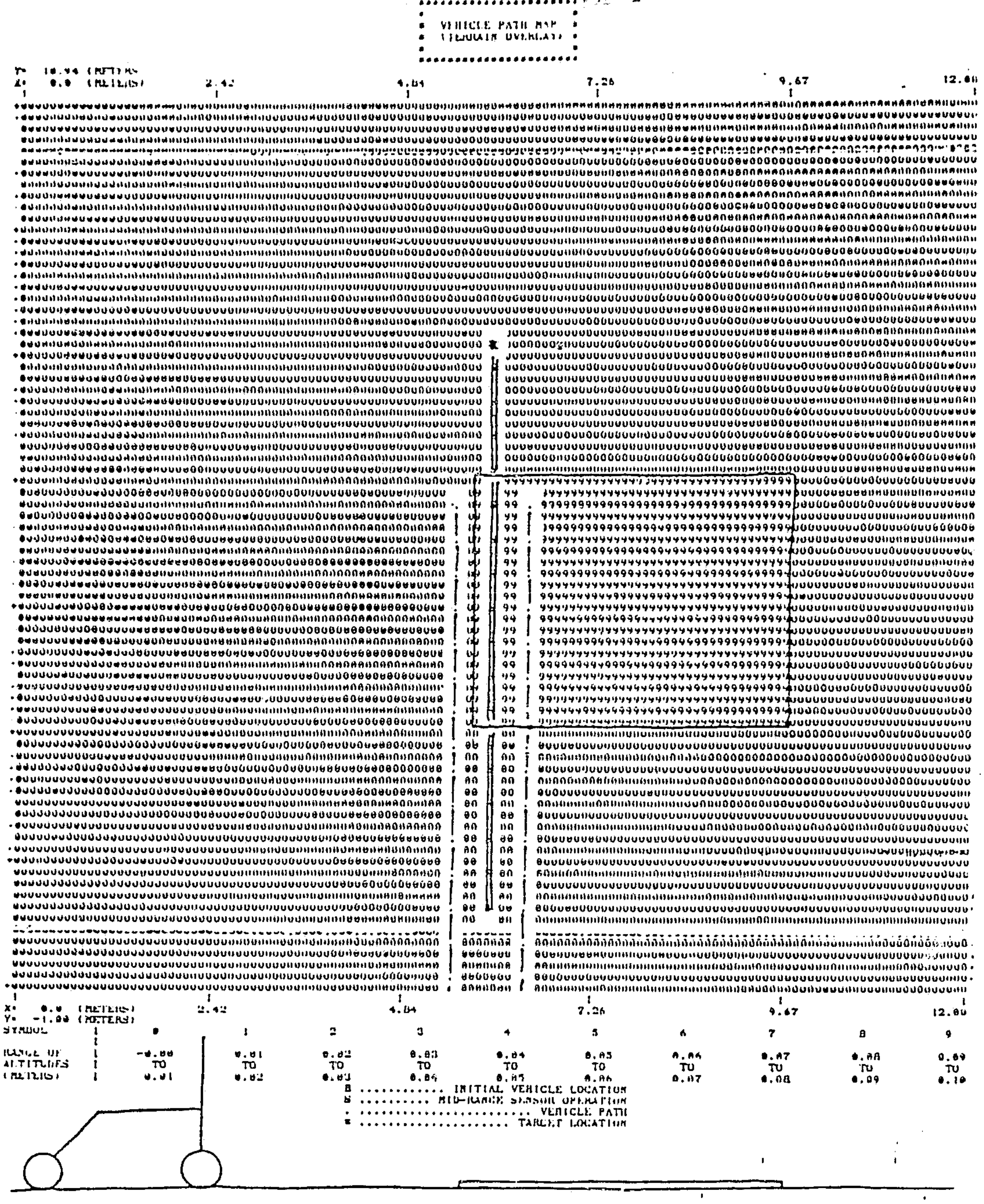


Figure 22. Rover Negotiating 0.10m Block on Level Ground

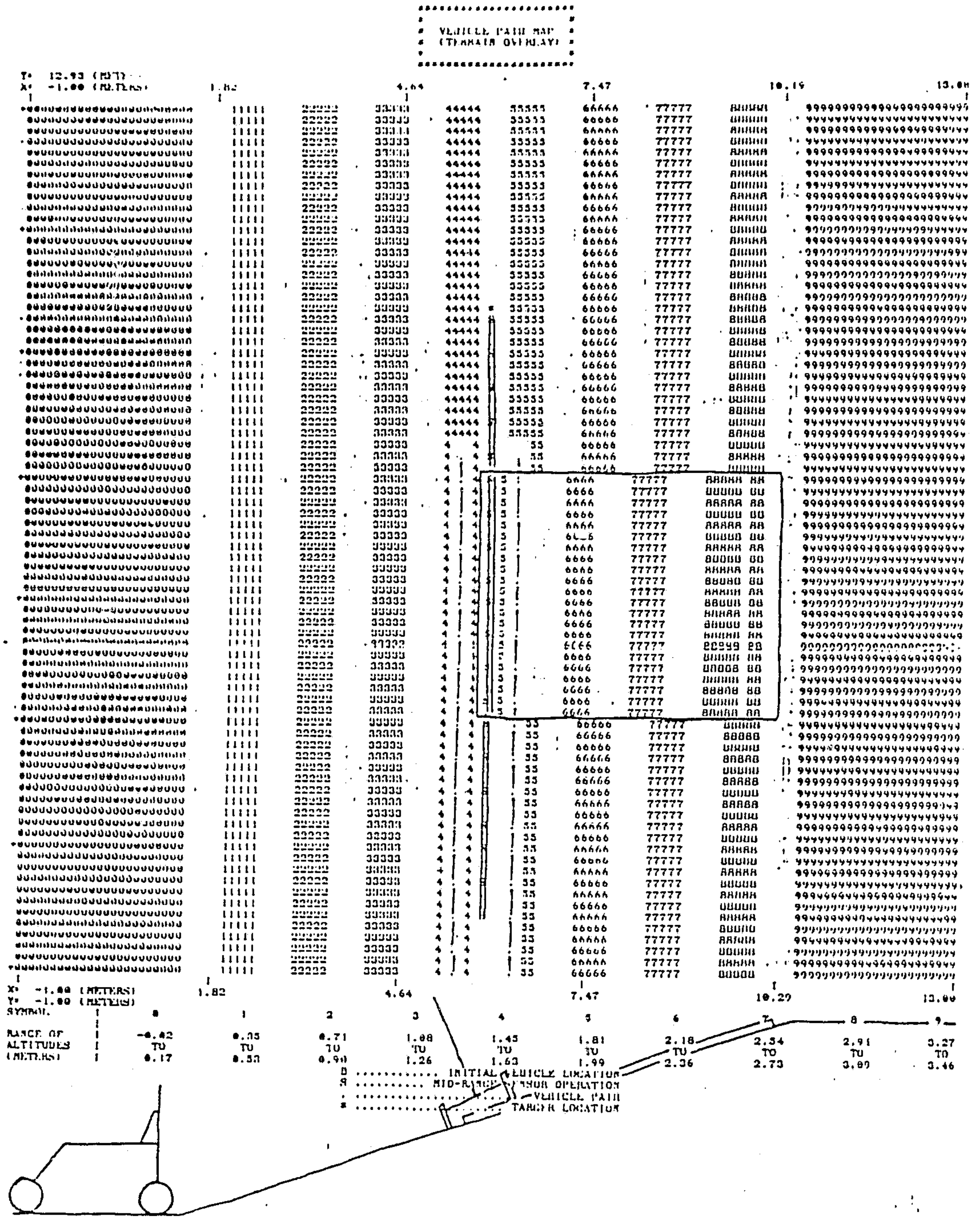


Figure 23. Near Crosspath Hazard of an Uphill Positive Obstacle

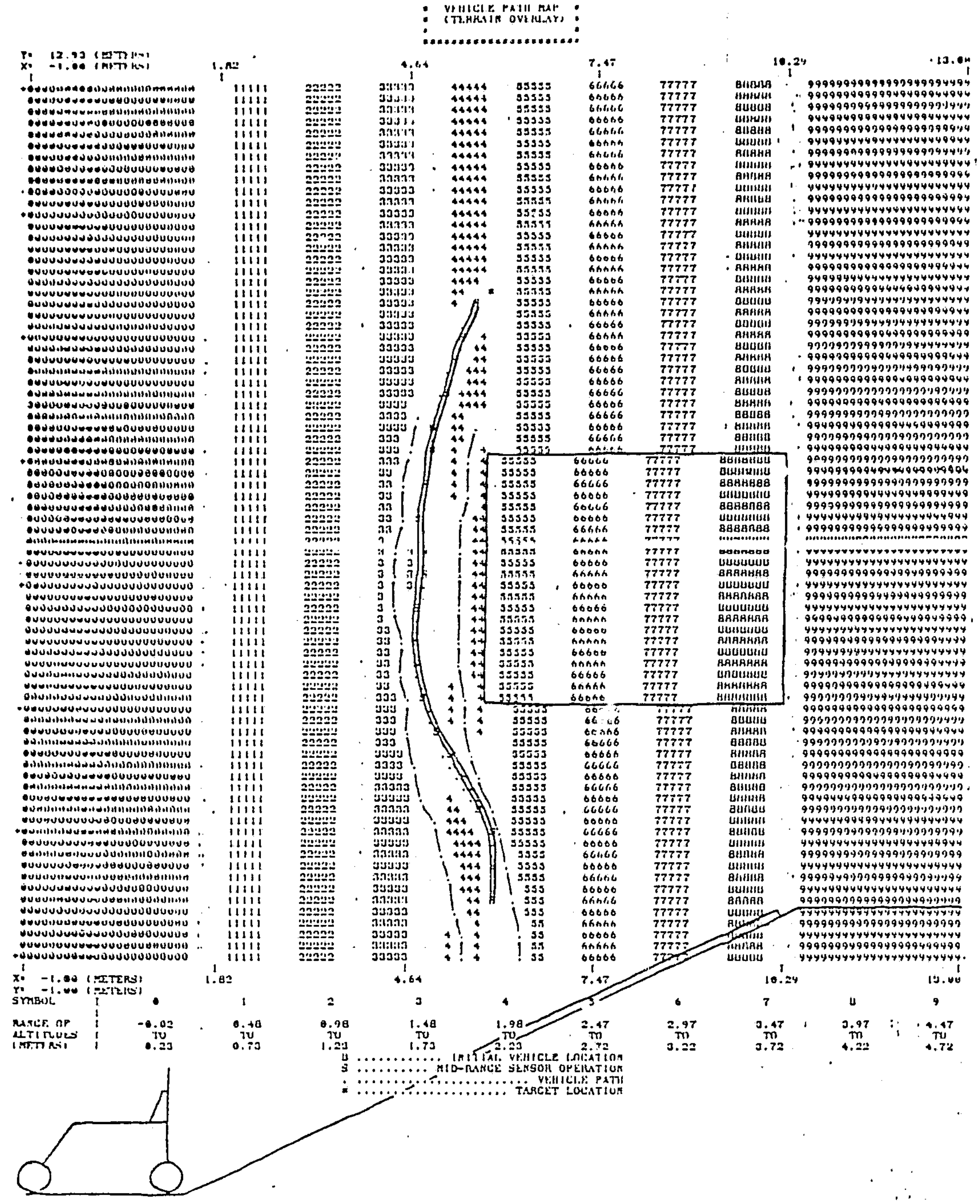


Figure 24. Crosspath Hazard of an Uphill Positive Obstacle

Before these runs were executed, it was expected that the rover would decide not to traverse ground that it should have been able to cover. The reasons for this can be seen in Figure 25. Here the rover is on a sine based terrain that it should be able to roll over without detouring. At various locations below the main figure are drawn the raw data returns from the sensors. At point three, the terrain is declared as hazardous. This is because not enough information exists at that point about the ground (notice that the same terrain has been swept out by the lasers as the vehicle approaches point three, but there is no data on the land between points three and four). As the terrain forces the pitch of the vehicle to become more and more violent, the "obstacles" (missing returns) that the vehicle sees become larger and more hazardous.

Figures 26 - 29 show the results of these simulations. The terrain with a 0.25 meter amplitude is easily traversed by the rover. At a 0.30 meter amplitude, the vehicle first perceives an obstacle (Figure 27). As predicted, as the amplitude variations increased, the sensors detected larger hazards which the vehicle needed to avoid, until at a 0.40 meter amplitude (not shown), the rover become "lost". The path followed is the same as that for Figure 29, except once the rover moved off the top of the page, it never returned to the target.

Returning to Figure 27 (0.30m. amplitude), it would be interesting to note how the modeler might interpret a few small obstacles placed in its pathway; this was done in the next few simulations.

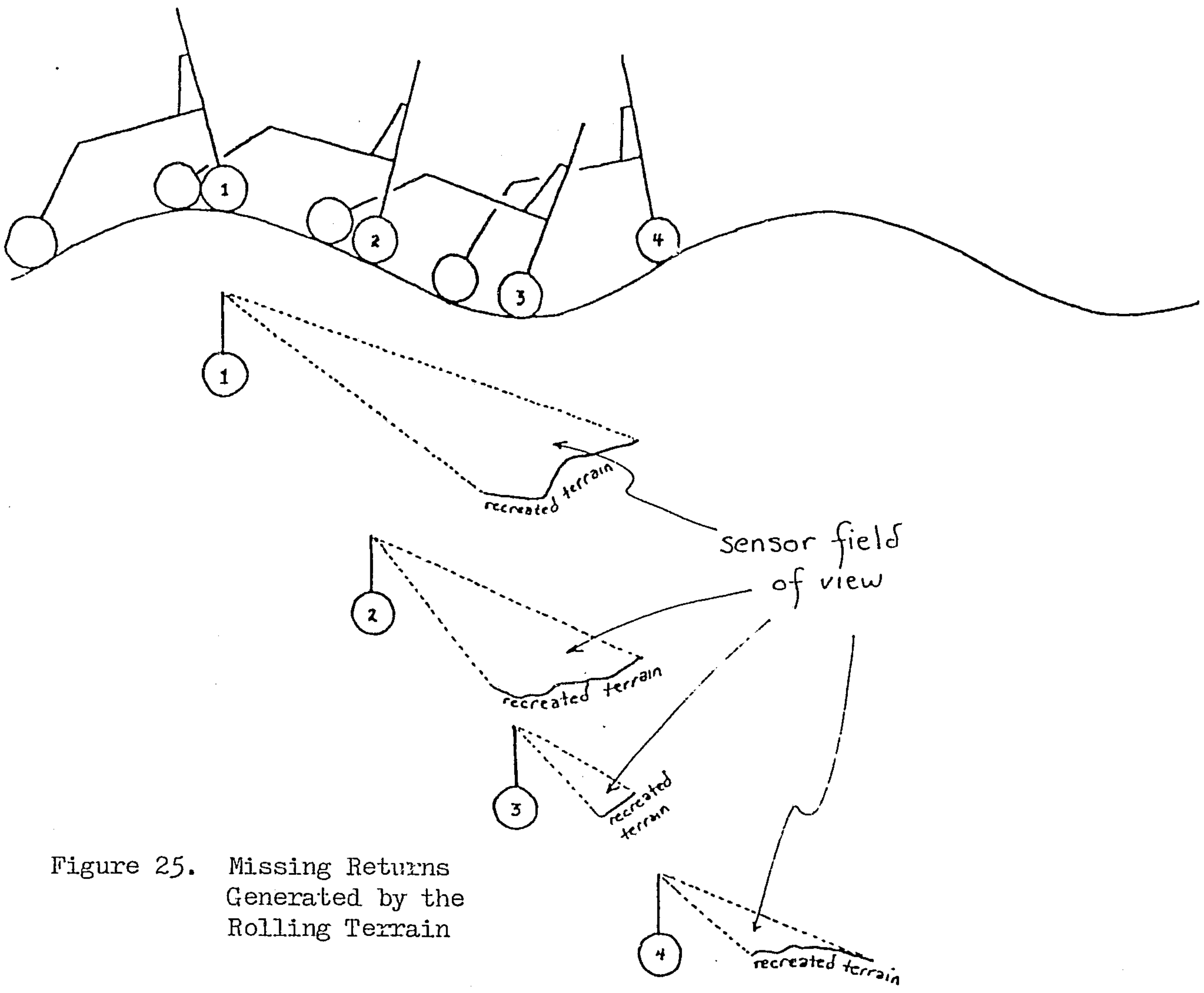
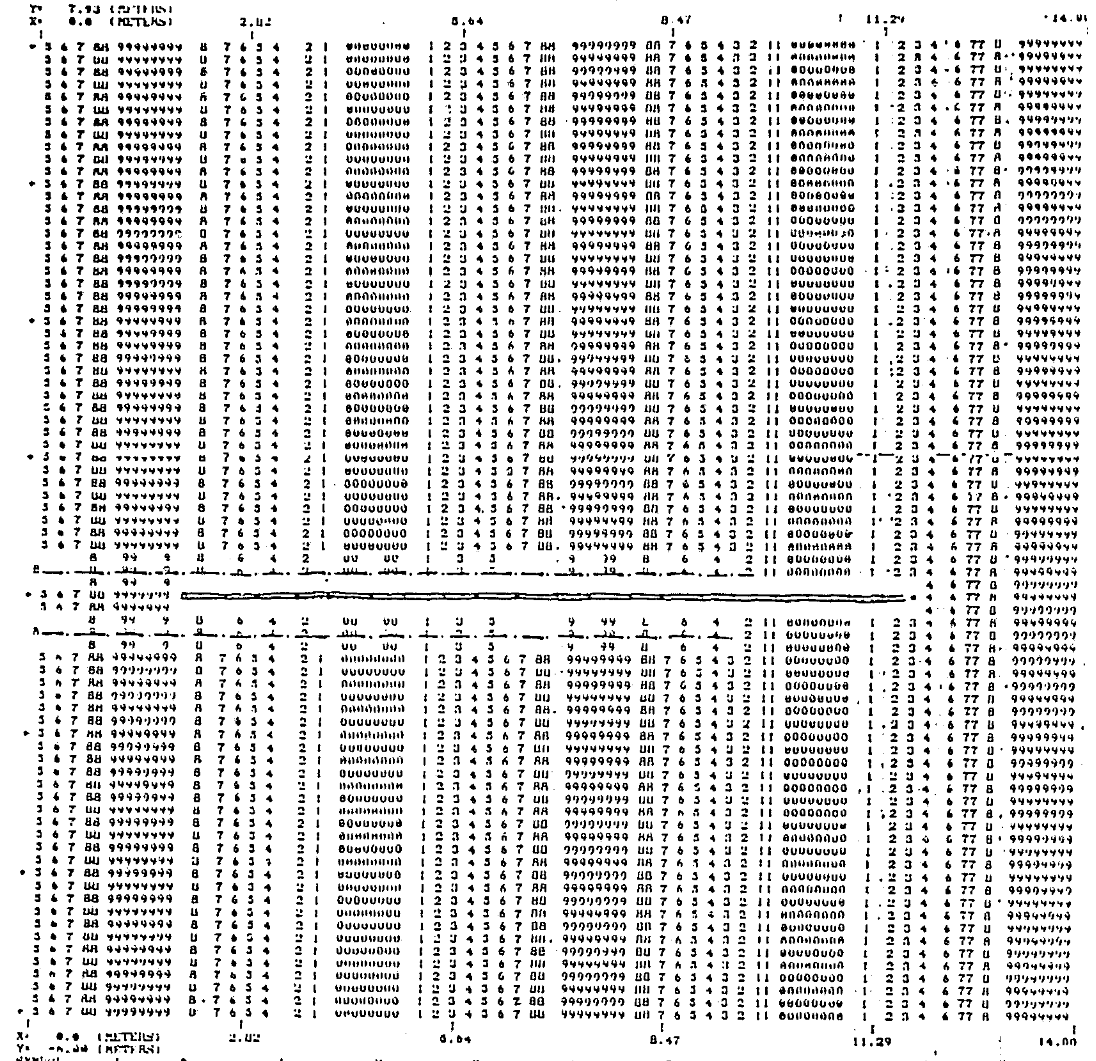


Figure 25. Missing Returns Generated by the Rolling Terrain

VEHICLE PATH MAP
(TERRAIN OVERLAY)

Y= 7.93 (METERS)
X= 0.0 (METERS)



X= 0.0 (METERS) 2.82 5.64 8.47 11.29 14.00
Y= -6.00 (METERS)

SYMBOL	0	1	2	3	4	5	6	7	8	9
RANGE OF ALTITUDES (METERS)	-0.25 TO -0.23	-0.20 TO -0.17	-0.15 TO -0.12	-0.09 TO -0.07	-0.04 TO -0.01	0.01 TO 0.04	0.07 TO 0.09	0.12 TO 0.10	0.17 TO 0.20	0.23 TO 0.25

R .... INITIAL VEHICLE POSITION
S .... MID-RANGE SENSOR LOCATION
. ... VEHICLE PATH
* .... TARGET LOCATION

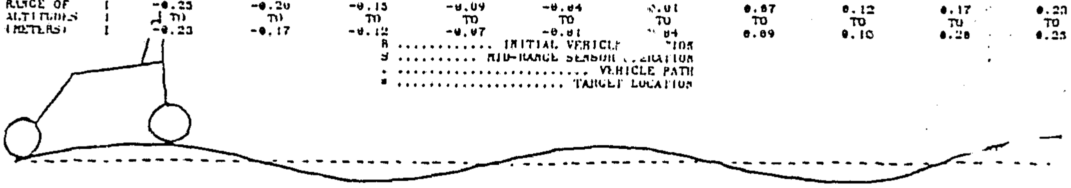


Figure 26. 0.25m Amplitude, 6.0m Period Sine Based Terrain

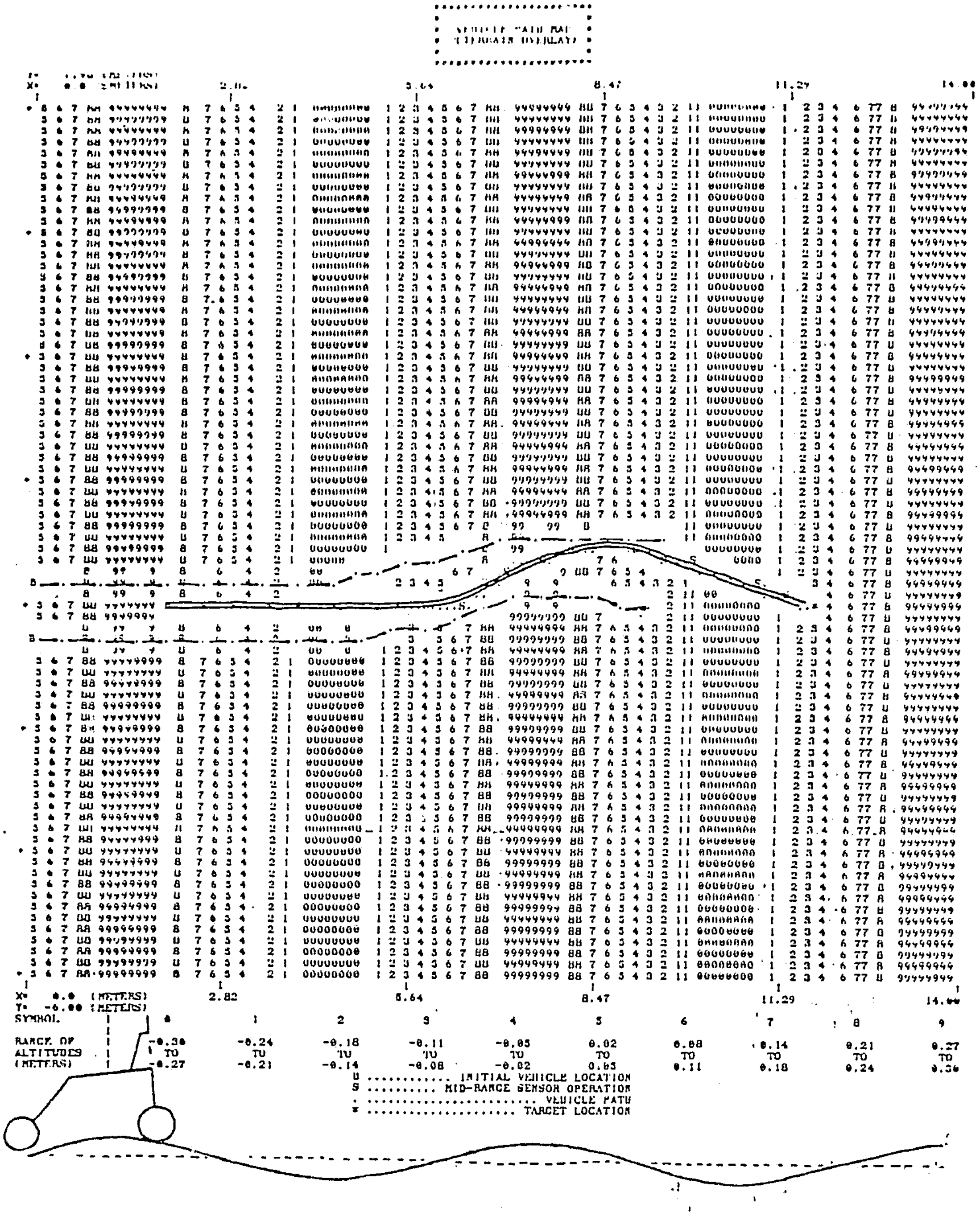


Figure 27. 0.30m Amplitude, 6.0m Period Sine Based Terrain

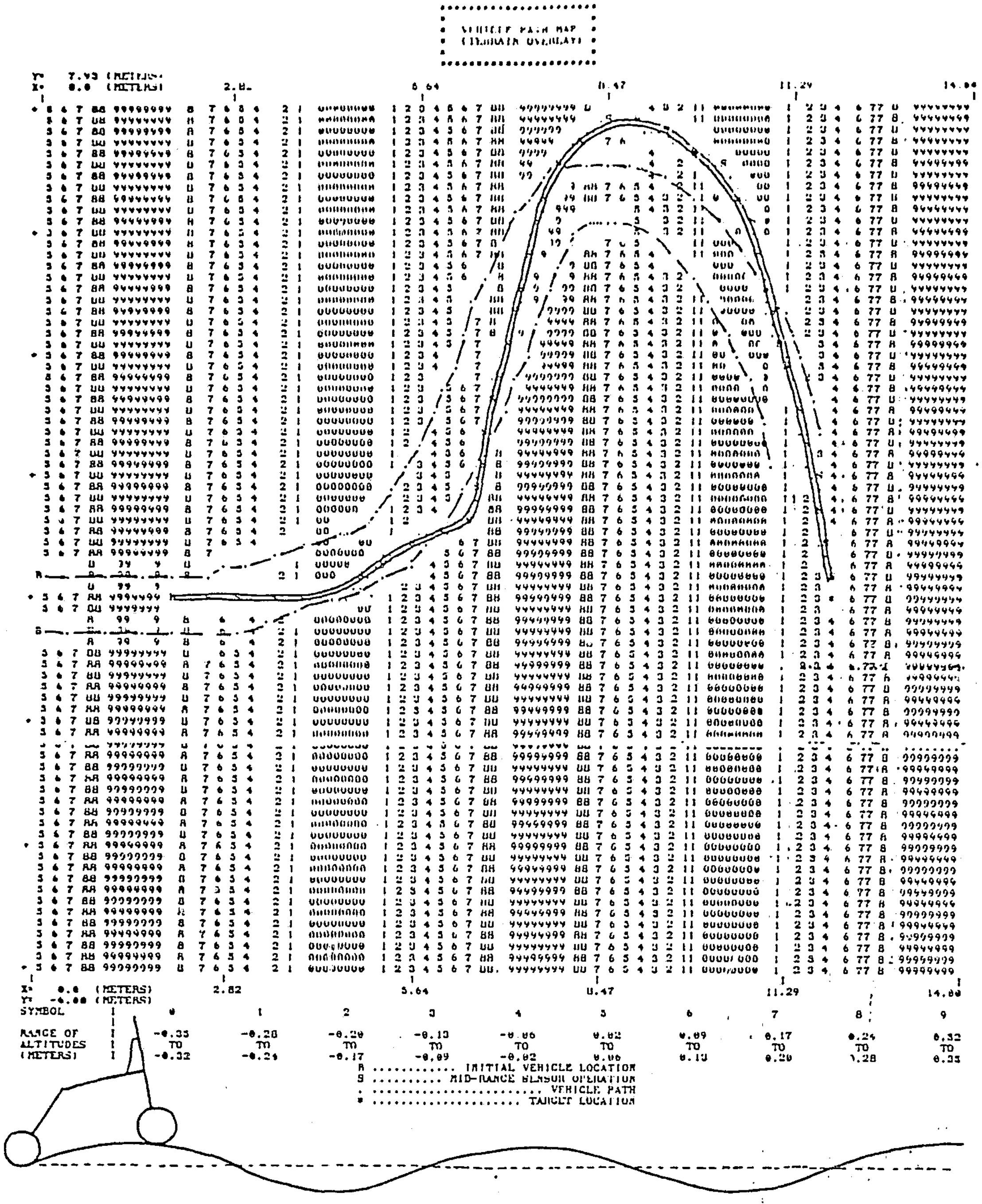


Figure 28. 0.35m Amplitude, 6.0m Period Sine Based Terrain

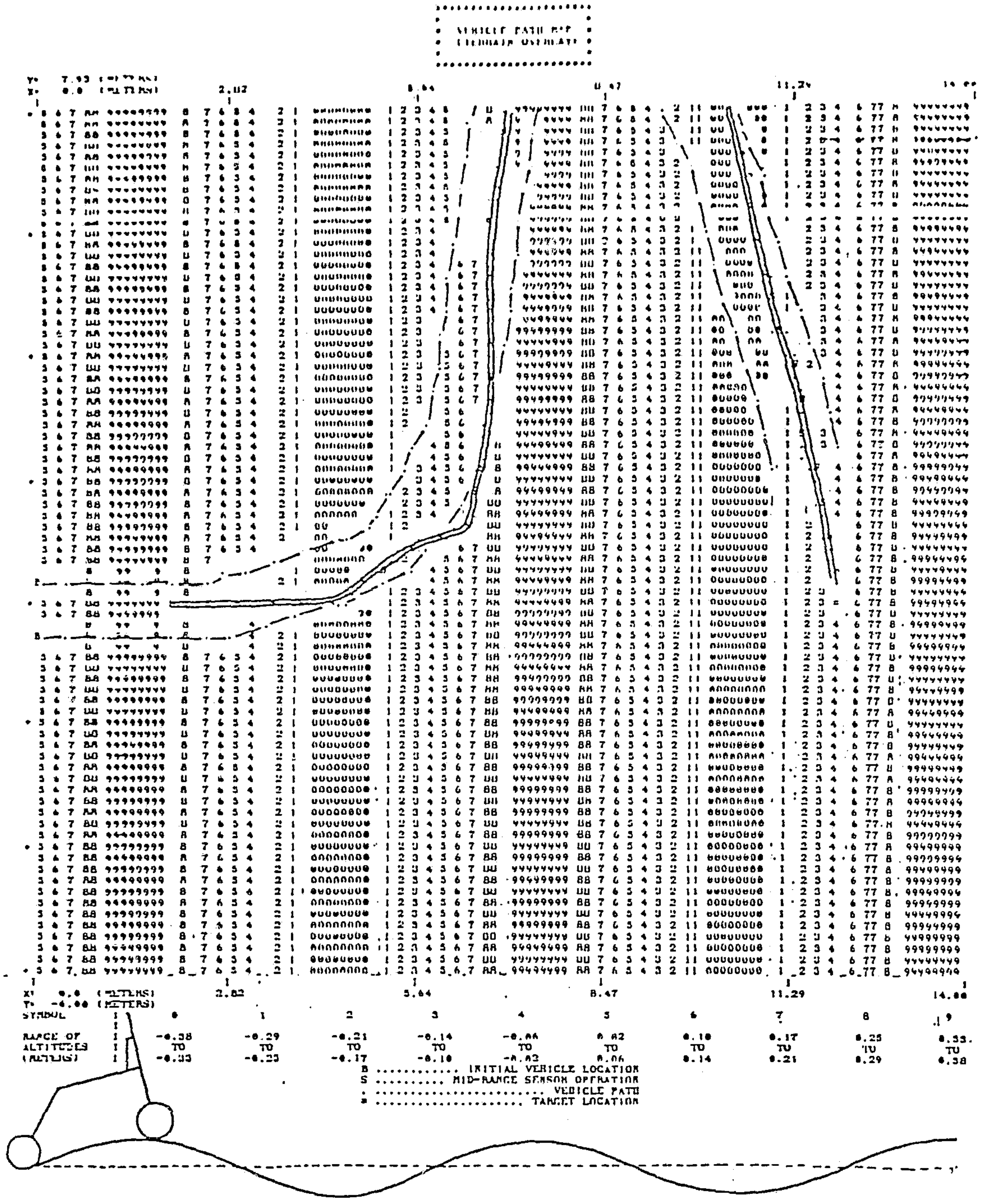


Figure 29. 0.38m Amplitude, 6.0m Period Sine Based Terrain

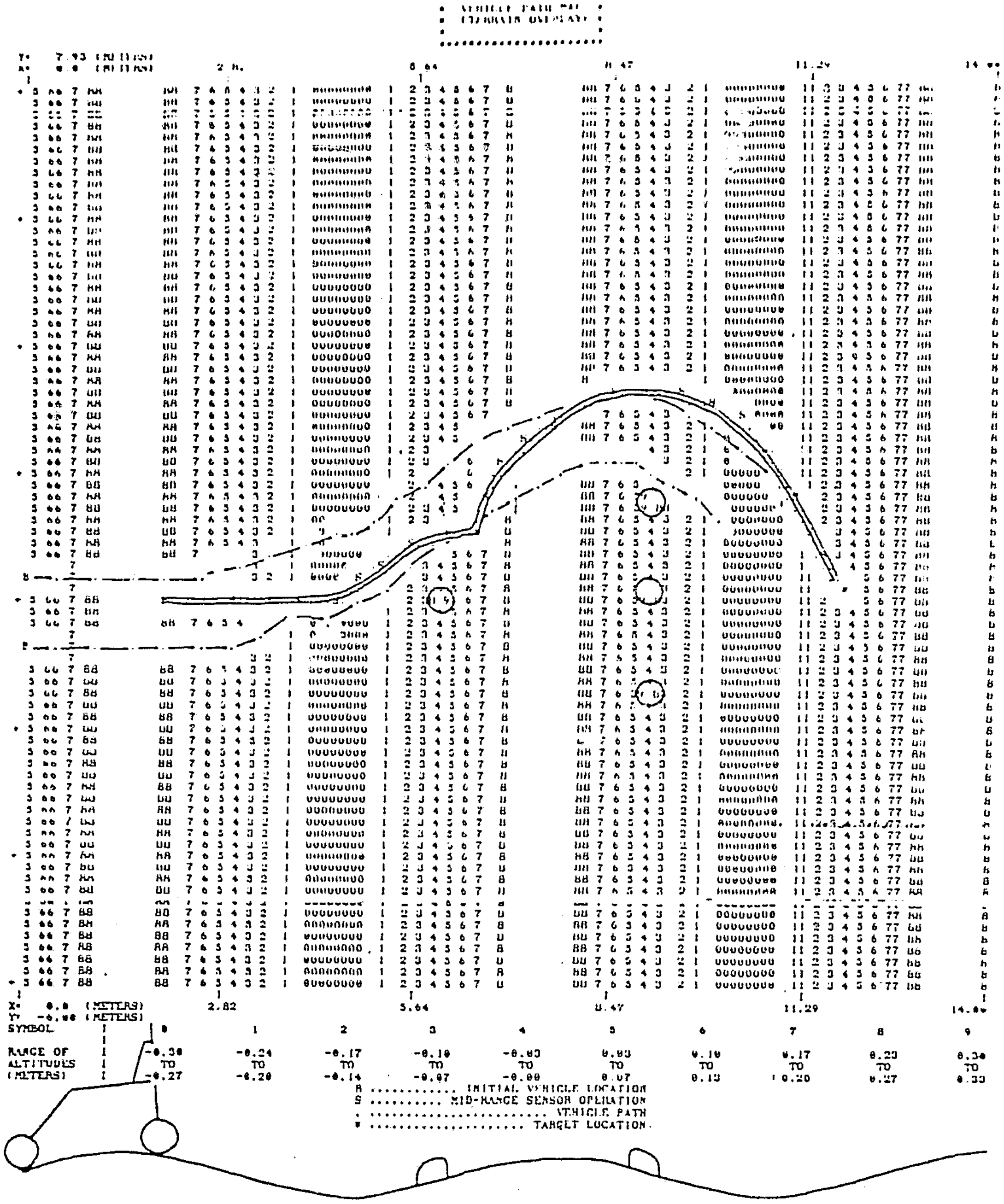


Figure 30. 0.30m Amplitude, 6.0m Period Sine Based Terrain with 0.30m boulders

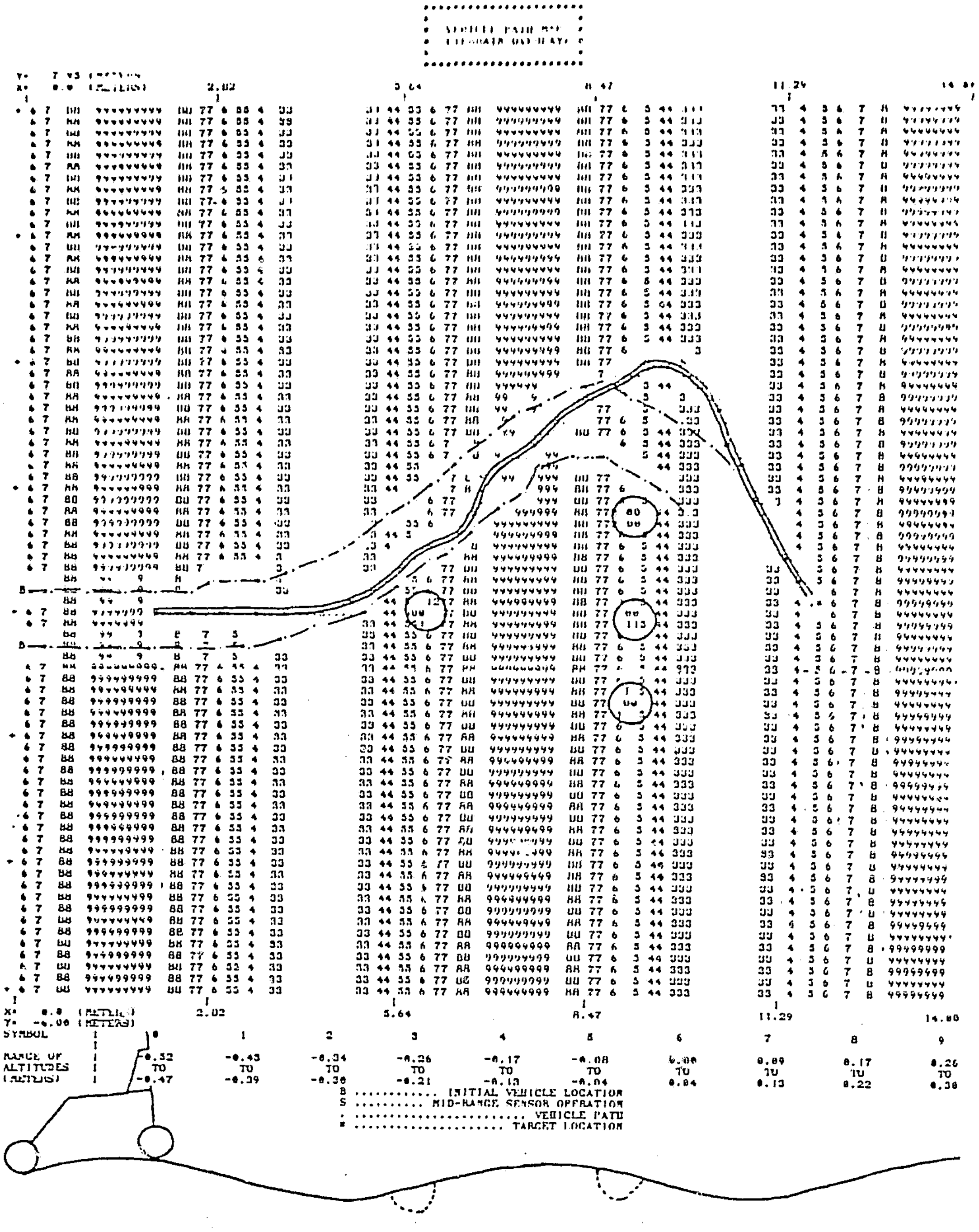


Figure 31. 0.30m Amplitude, 6.0m Period Sine Based Terrain with 0.30m deep craters

In Figure 30, small boulders were used, while in Figure 31 small craters that were of the same size as the boulders were used. In each case, the rover easily avoided the obstacles and proceeded to the target.

Admittedly, a terrain with this magnitude of variation is not expected to be encountered. If one examines photographs of Mars, although it does have a rolling landscape, the variation is much slower. However, it was reassuring to note that the vehicle performed as well as it did during these test. Since a varying terrain such as that used in these simulations is not expected, the sensor system and modeler are satisfactory. If the vehicle were going to encounter terrain of this nature, several things could be changed to permit the vehicle to tackle the job. More lasers and sensors might be added to increase the range of sight of the rover (more sensors would probably do the trick). Another possibility would be to oscillate the sensor angles. On scan 1, the vehicle might look at ground 0.8 to 3.0 meters from the rover, scan 2 would cover the terrain from 2.5 to 6.0 meters; scans 3 and 4 would be the same as scan 1 and 2. Both of these possibilities are realistic solutions.

5.5 Obstacle Avoidance

In the last few sinusoidal terrain simulations, some small boulders or craters were added to the landscape. While these small obstacles exist, in reality a more general mix of both large and small terrain features would be found. With this in mind, the next task would be to set the rover moving through boulder-crater fields with varying obstacle sizes.

Each boulder-crater field simulation was run several times. During the second run, "noise" was added to both the pitch and roll of the vehicle. Examination of the martian surface reveals not only the reletively larger obstacles that the modeler would detect, but a surface littered with smaller rocks and holes that would probably go unnoticed. The "noise" added to these runs corrupts the attitude of the vehicle and is intended to disturb the vehicle in the same manner as the actual rubble. This noise had a maximum deviation of plus or minus ten degrees with a standard deviation of from 3.5 to 4.0 degrees. Figure 32 shows how this would change various terrain surfaces.

All of the simulations had successful terminations - the target was safely reached. There were of course minor variations to the initial path chosen when the attitude noise was added, but essentially the same course was followed for each of the noisy runs. Since approximately the same path was chosen by the vehicle if the noise was present, in Figure 36B, the noise was changed for the worse; for this run (and the next) the noise had a maximum deviation of plus or minus fifteen degrees with a standard deviation of 4.2 degrees. This type of rubble would really shake up the planetary rover, so it was really no surprise that the vehicle chose an entirely new path to follow. This run was repeated for a third time with the

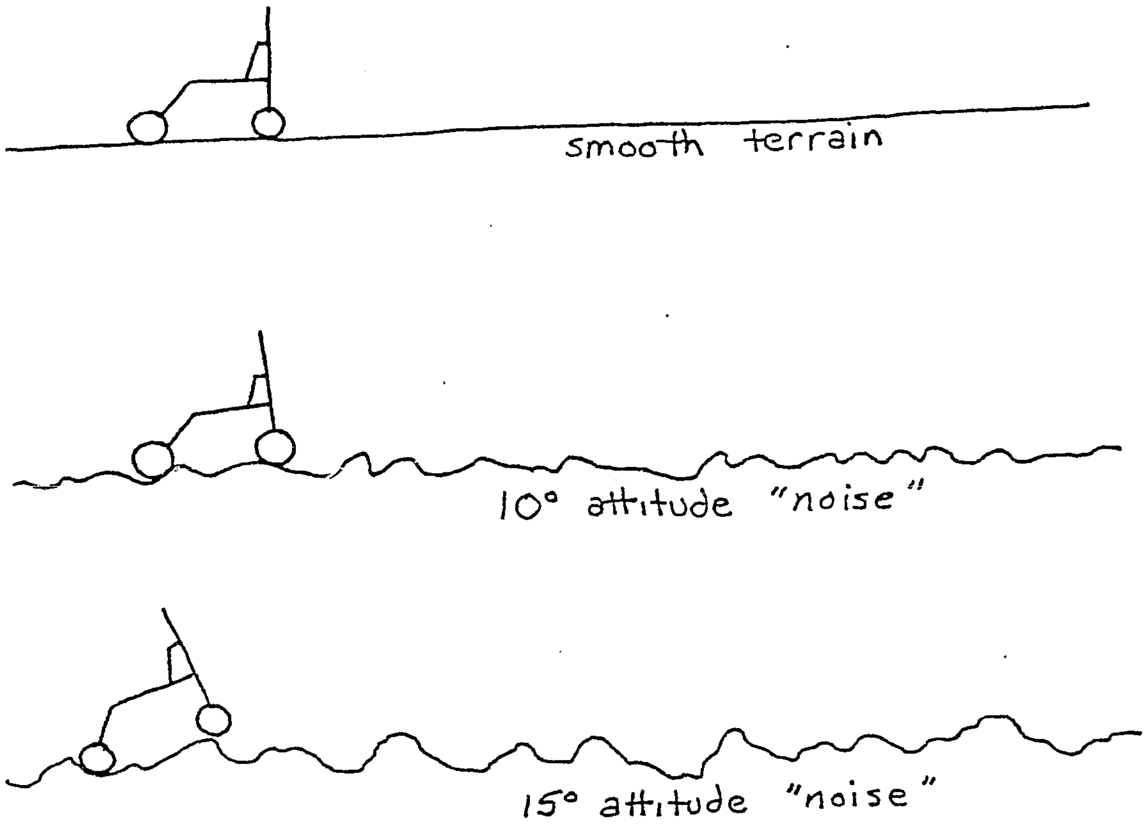


Figure 32. Effect of noise on the terrain

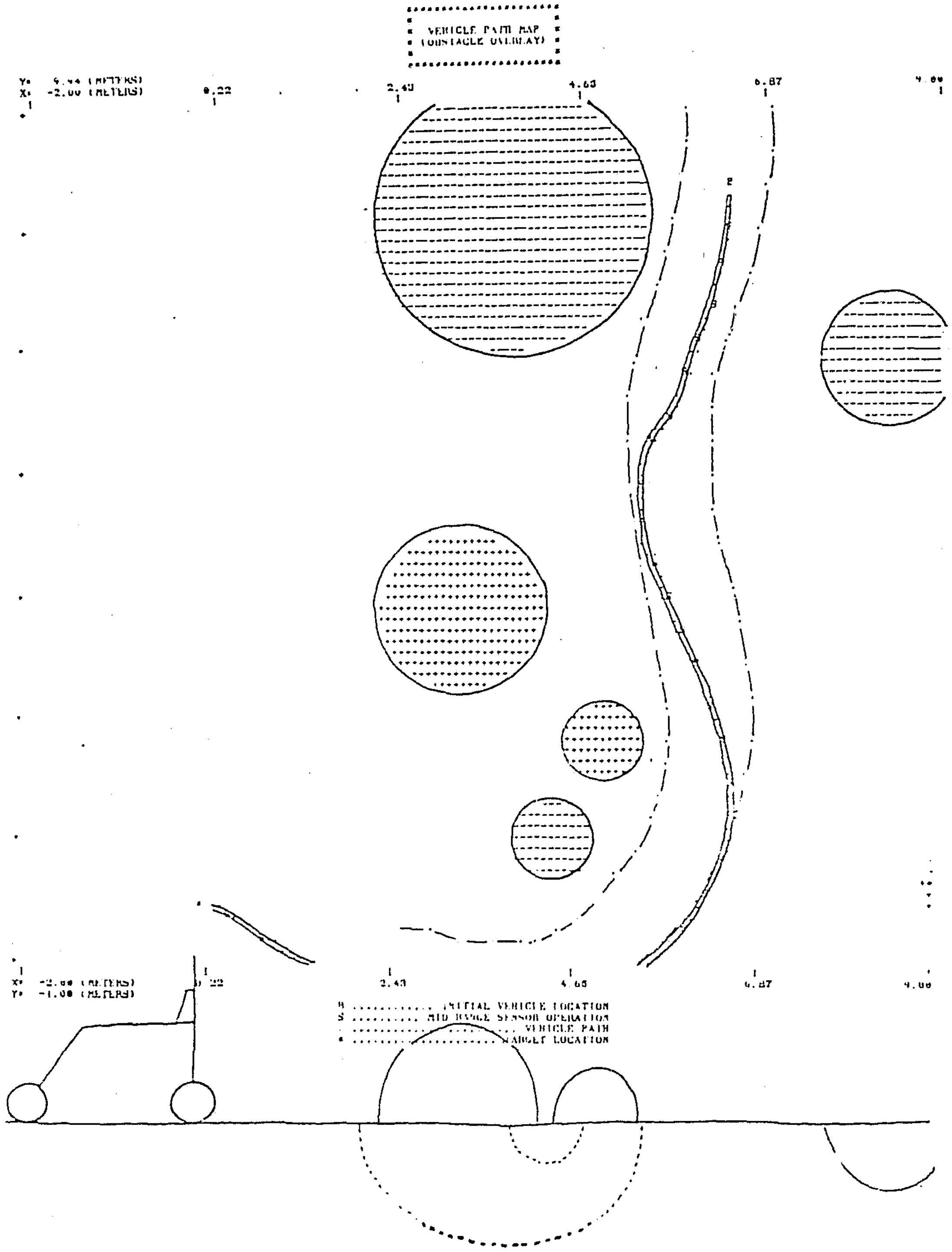


Figure 33A. Boulder-Crater Field, no noise

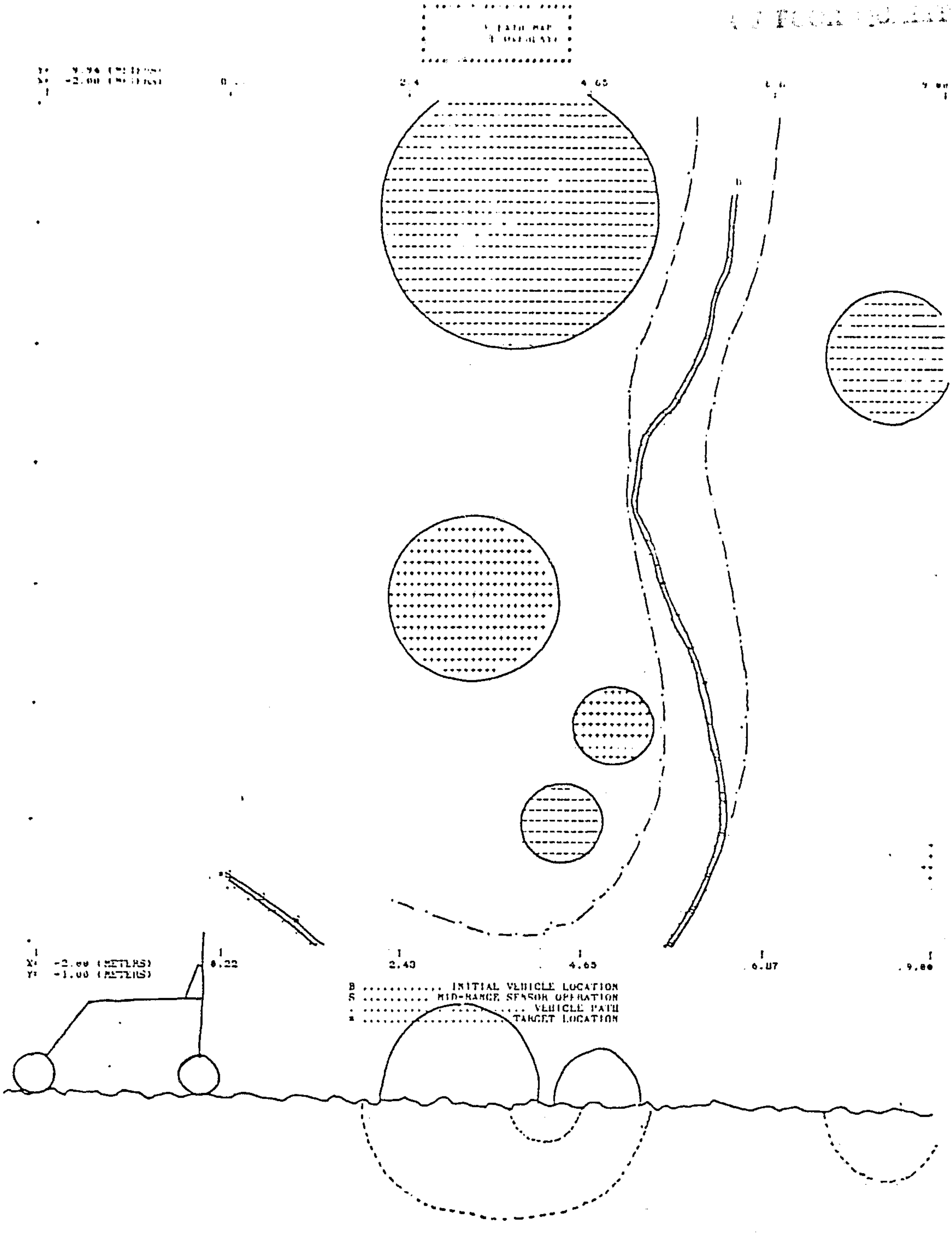

PATH MAP
OVERLAY
Y= 9.94 (METERS)
X= -2.00 (METERS)
0
2.4
4.65
9.88
B
X= -2.00 (METERS)
Y= -1.00 (METERS)
0.22
2.43
4.65
6.87
9.88
B .... INITIAL VEHICLE LOCATION
S .... MID-RANGE SENSOR OPERATION
. ... VEHICLE PATH
* .... TARGET LOCATION

Figure 33B. Boulder-Crater Field, 10° Attitude Noise

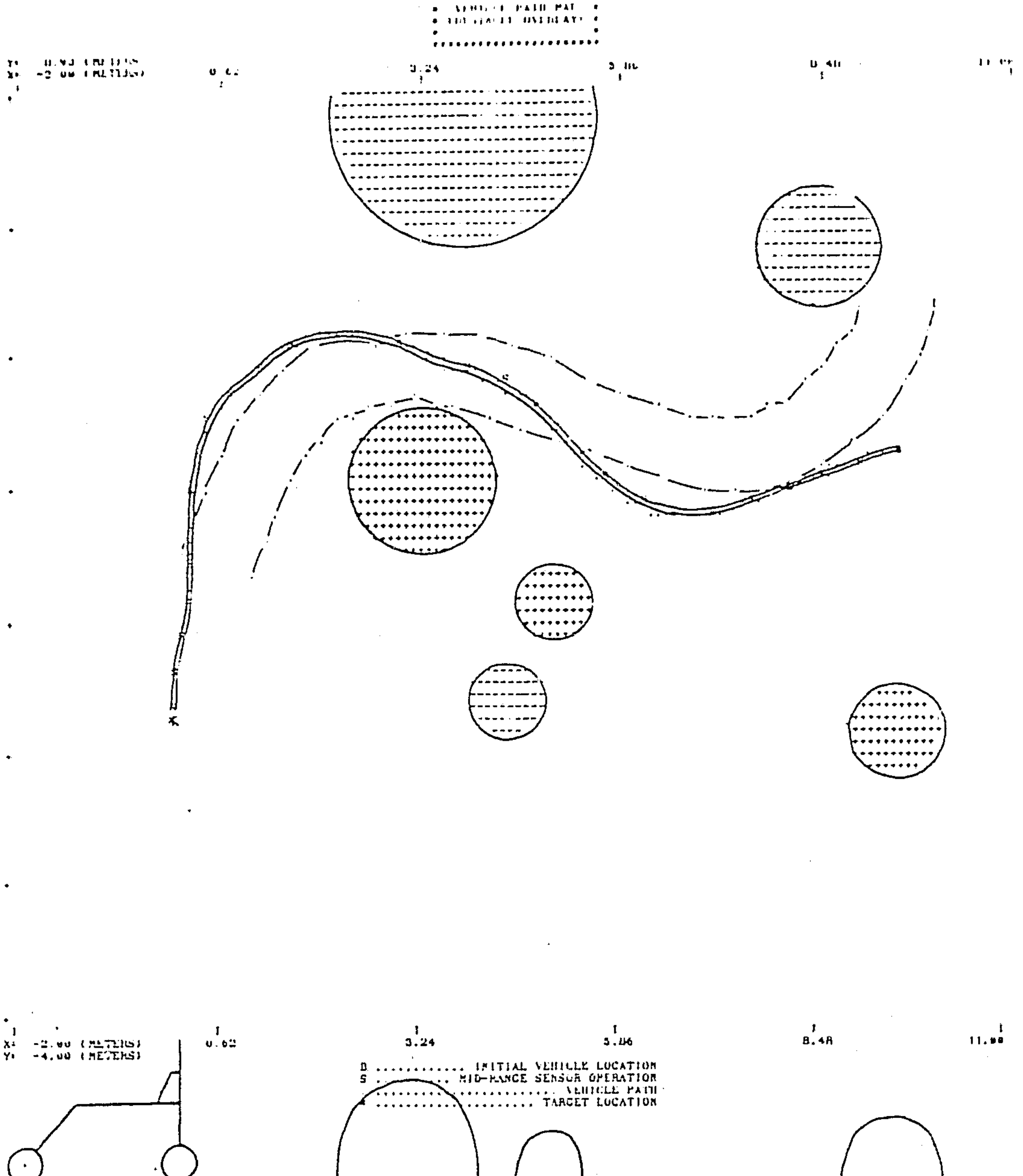


Figure 34A. Boulder-Crater Field, no noise

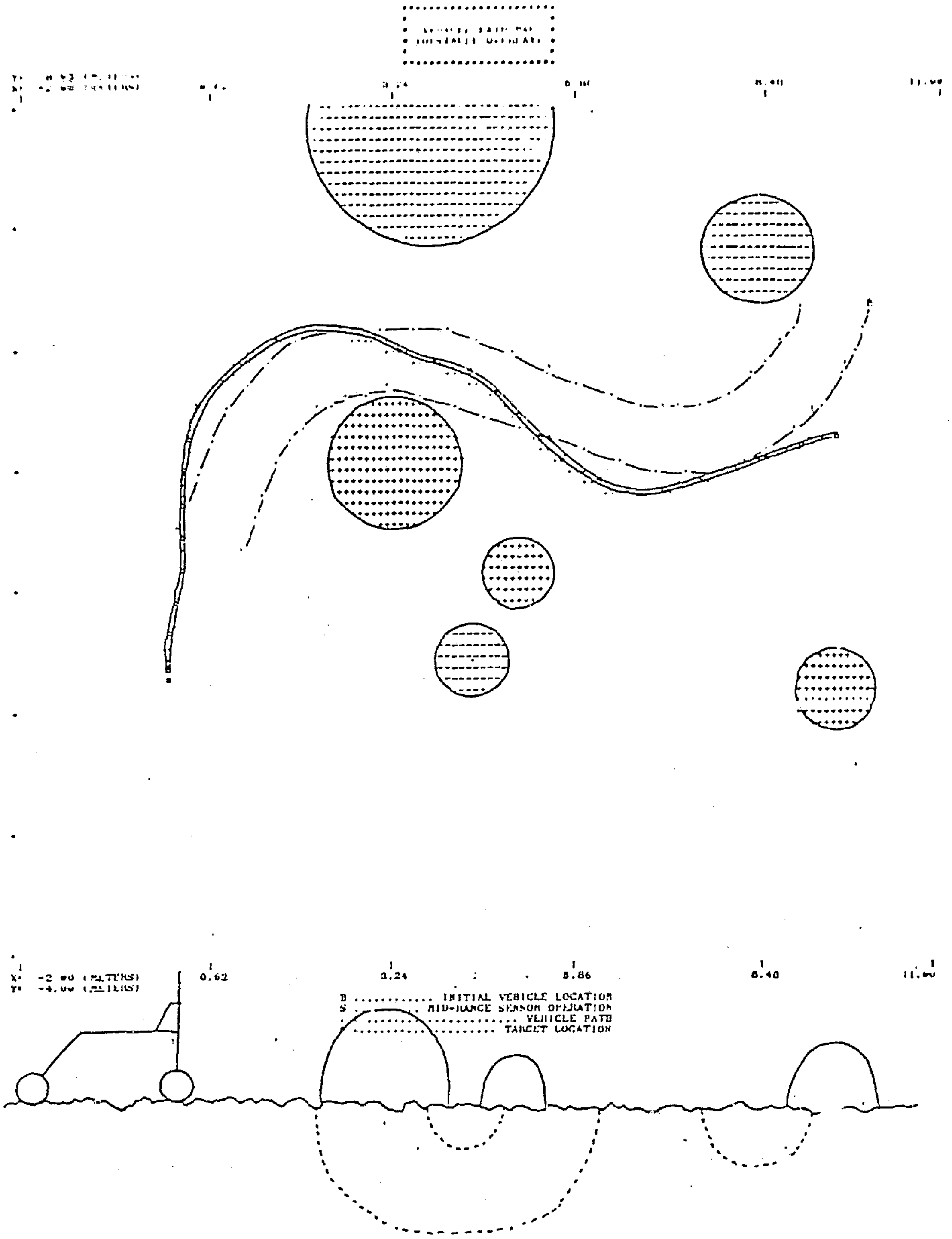

3.24
8.40
B
X= -2.00 (METERS)
Y= -4.00 (METERS)
0.62
3.24
5.86
8.40
11.00
B .... INITIAL VEHICLE LOCATION
S .... MID-RANGE SENSOR OPERATION
. ... VEHICLE PATH
. ... TARGET LOCATION

Figure 34B. Boulder-Crater Field, 10° Attitude Noise

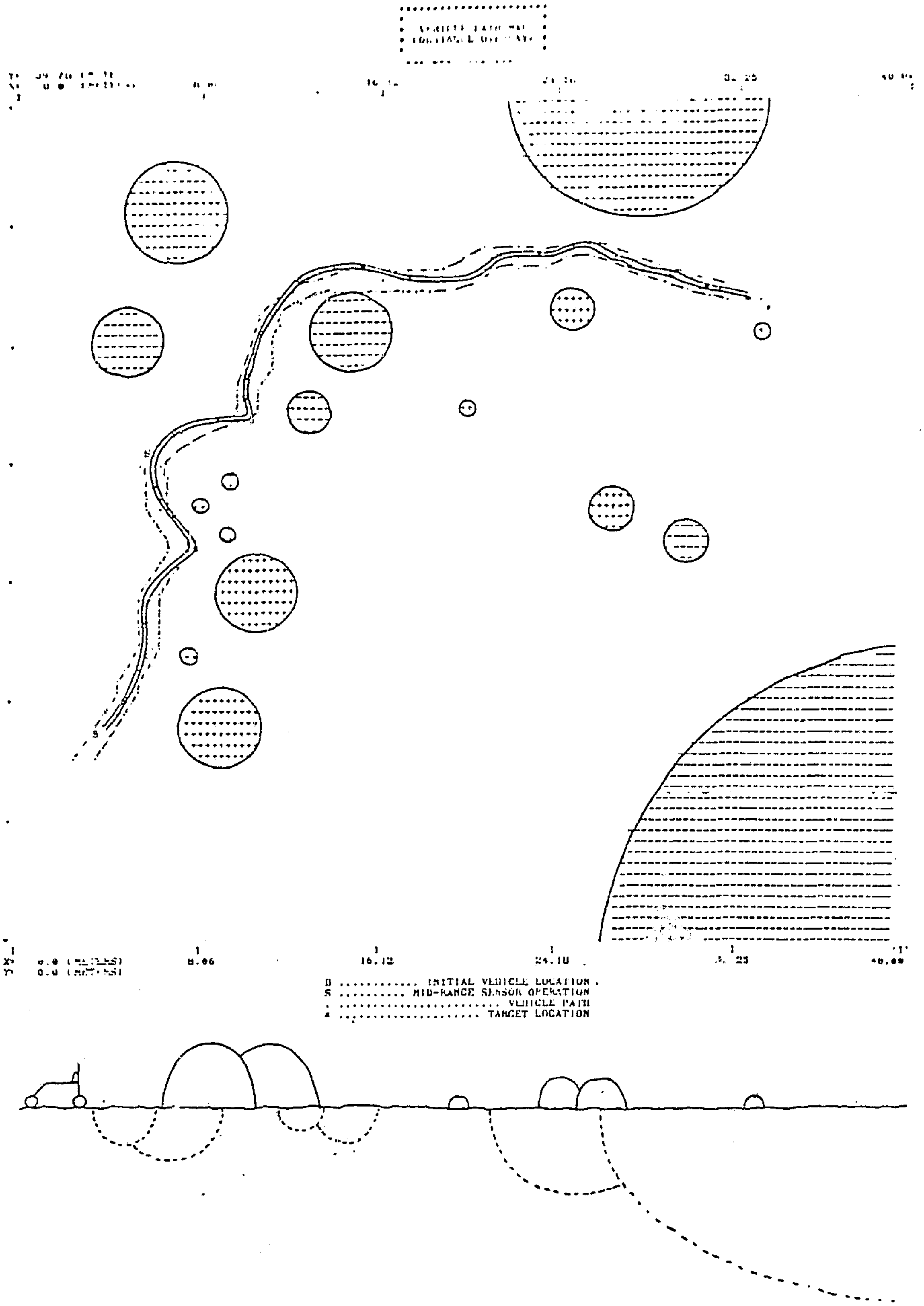

X= 0.0 (METERS)
Y= 0.0 (METERS)
8.06
16.12
24.18
40.00
B .... INITIAL VEHICLE LOCATION
S .... MID-RANGE SENSOR OPERATION
. ... VEHICLE PATH
* .... TARGET LOCATION
S

Figure 35. Boulder-Crater Field, 10° Attitude Noise (due to the scale of this map, no difference could be noted between the normal and the noisy run. The simulation map of the un-noisy run was not printed.)

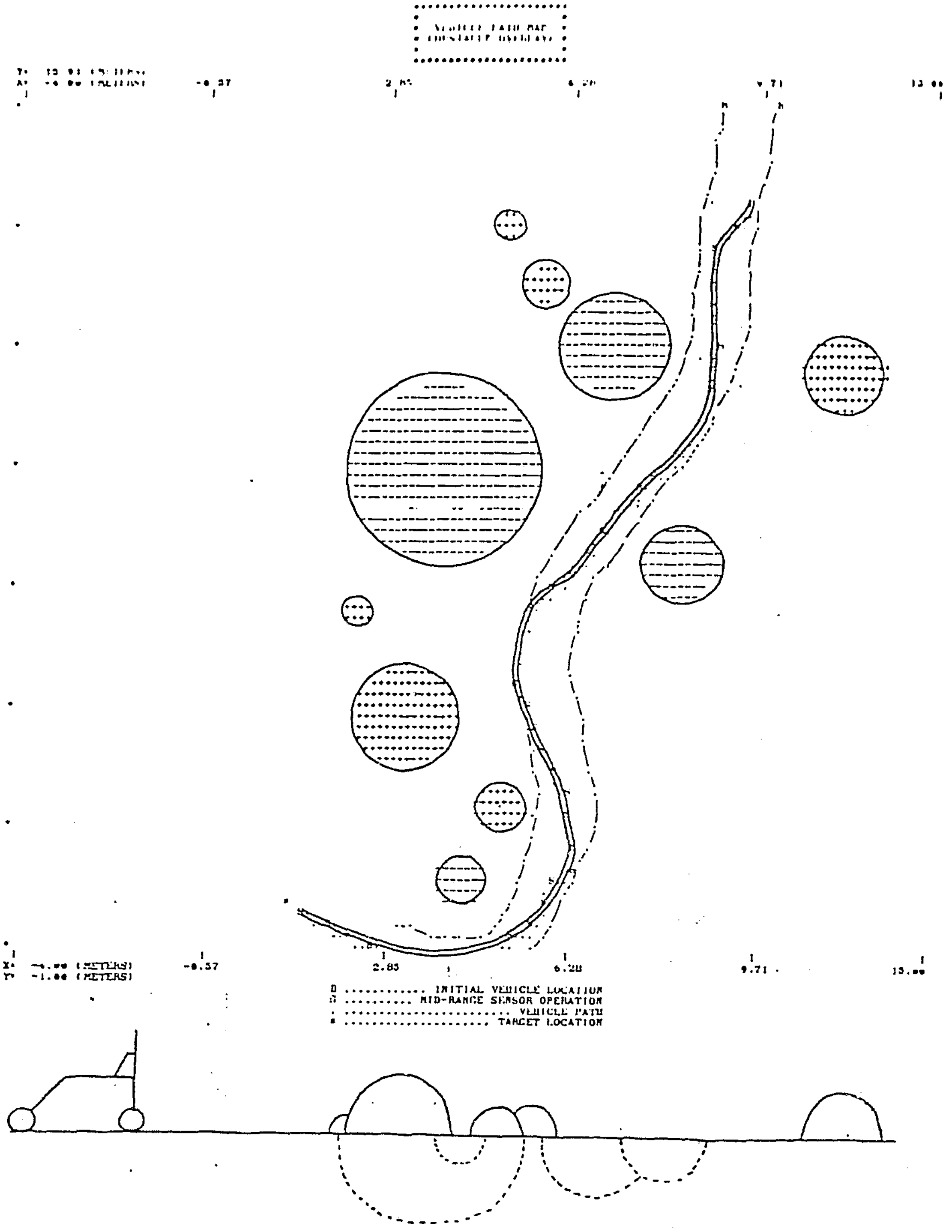

-0.57
2.85
6.28
9.71
X= -4.00 (METERS)
Y= -1.00 (METERS)
-0.57
2.85
6.28
9.71
13.00
D .... INITIAL VEHICLE LOCATION
S .... MID-RANGE SENSOR OPERATION
. ... VEHICLE PATH
* .... TARGET LOCATION

Figure 36A. Boulder-Crater Field, no noise

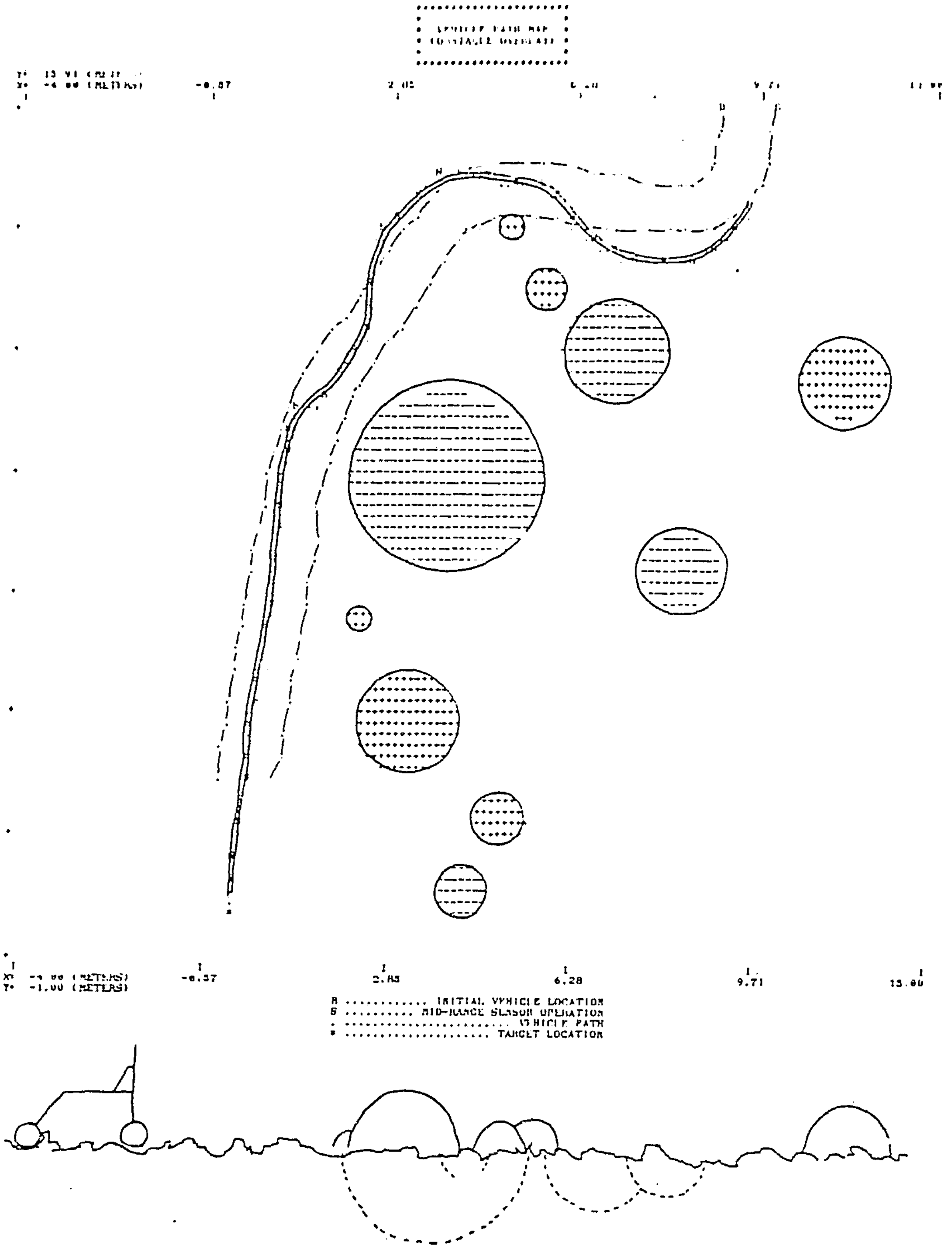


Figure 36B. Boulder-Crater Field, 15° Attitude Noise

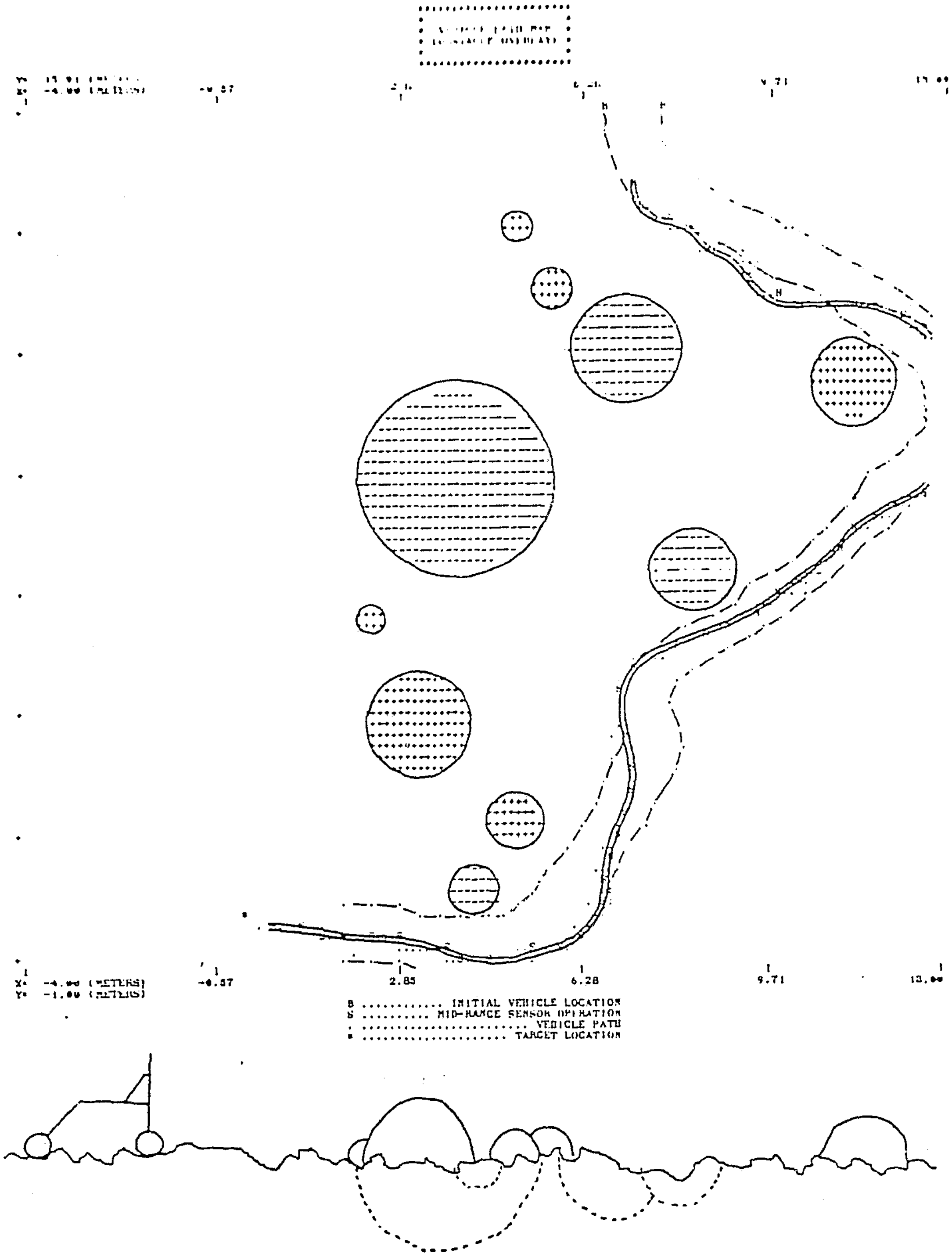


Figure 36C. Boulder-Crater Field, 15° Attitude Noise

vehicle starting from a slightly different location. With this starting angle, the vehicle could not safely pass between the first two obstacles. This was observed in the simulation, and although it initially began to move away from the target, the rover none-the-less proceeded on as short a path as possible toward its destination.

5.6 Realistic Terrains

The final simulations are fairly representative of the terrain that is likely to be found on Mars. Photographs of the martian surface were used in the creation of these landscapes[5]. The output of each run is preceeded by a "three-dimensional" projection of the terrain used. This was done solely tc provide a better grasp of what was displayed on the vehicle path maps.

Figures 37A and 37B show a large gaussian hill along with scattered boulders and craters that were placed in the vehicles path. The destination was on the top of another hill. Ten degree attitude noise was added to both the pitch and the roll. The vehicle started toward the hill, but then changed course to avoid a crosspath hazard formed by the hill and a crater (the case not shown in the crosspath tests). The larger hill was slightly too steep for the vehicle to climb directly toward the target, but the hill was negotiated by reducing its pitch (veering toward the left). A close up of the last eighteen meters of this simulation are shown in Figures 38 (A and B).

The last run (Figures 39A and 39B) again used the ten

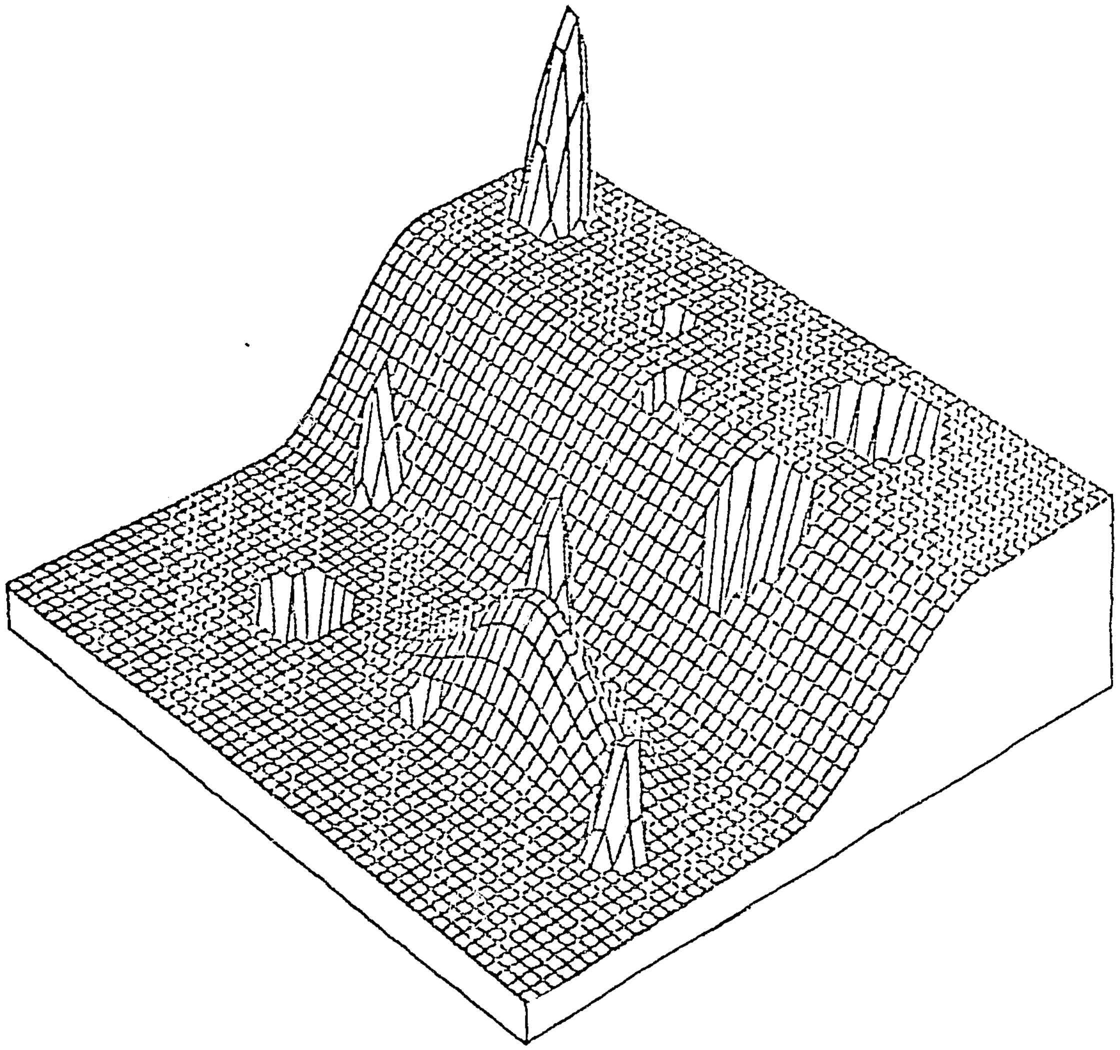


Figure 37A. Three-dimensional Projection of Figure 37B

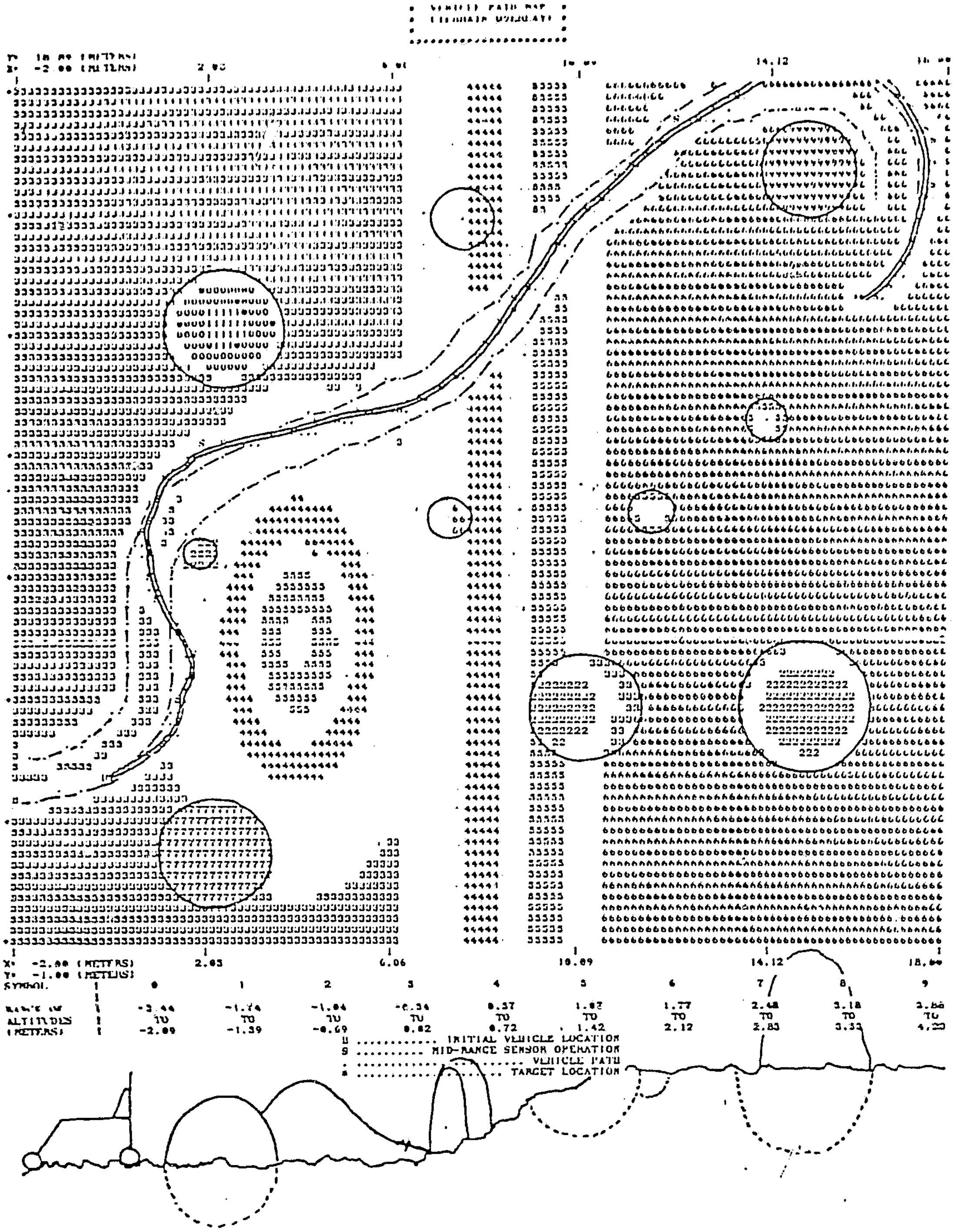


Figure 37B. Boulder-Crater Field, Hills, and 10° Attitude Noise

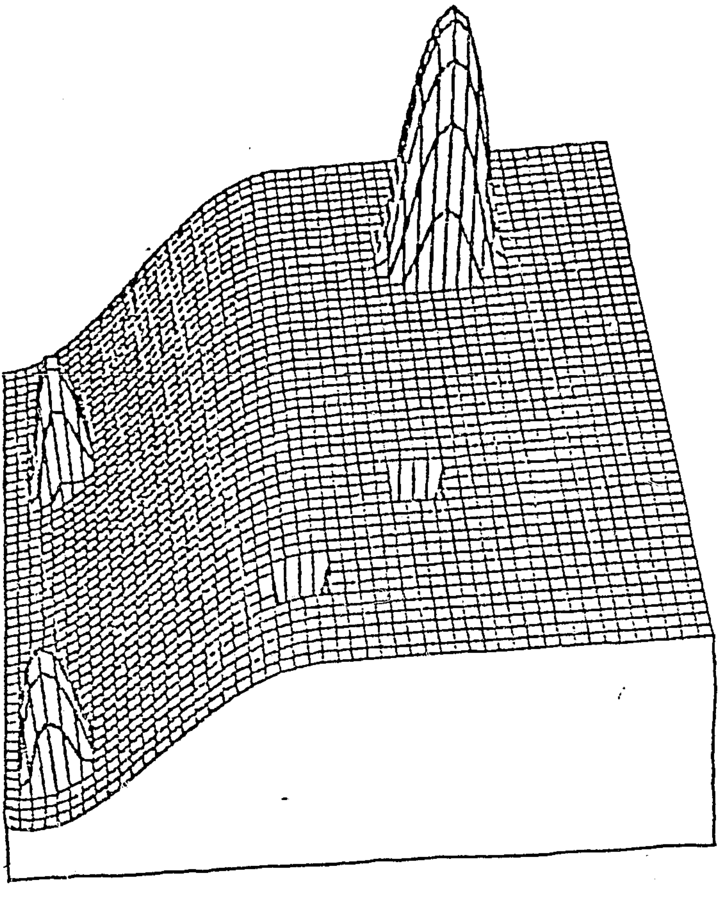


Figure 38A. Three-dimensional Projection of Figure 38B

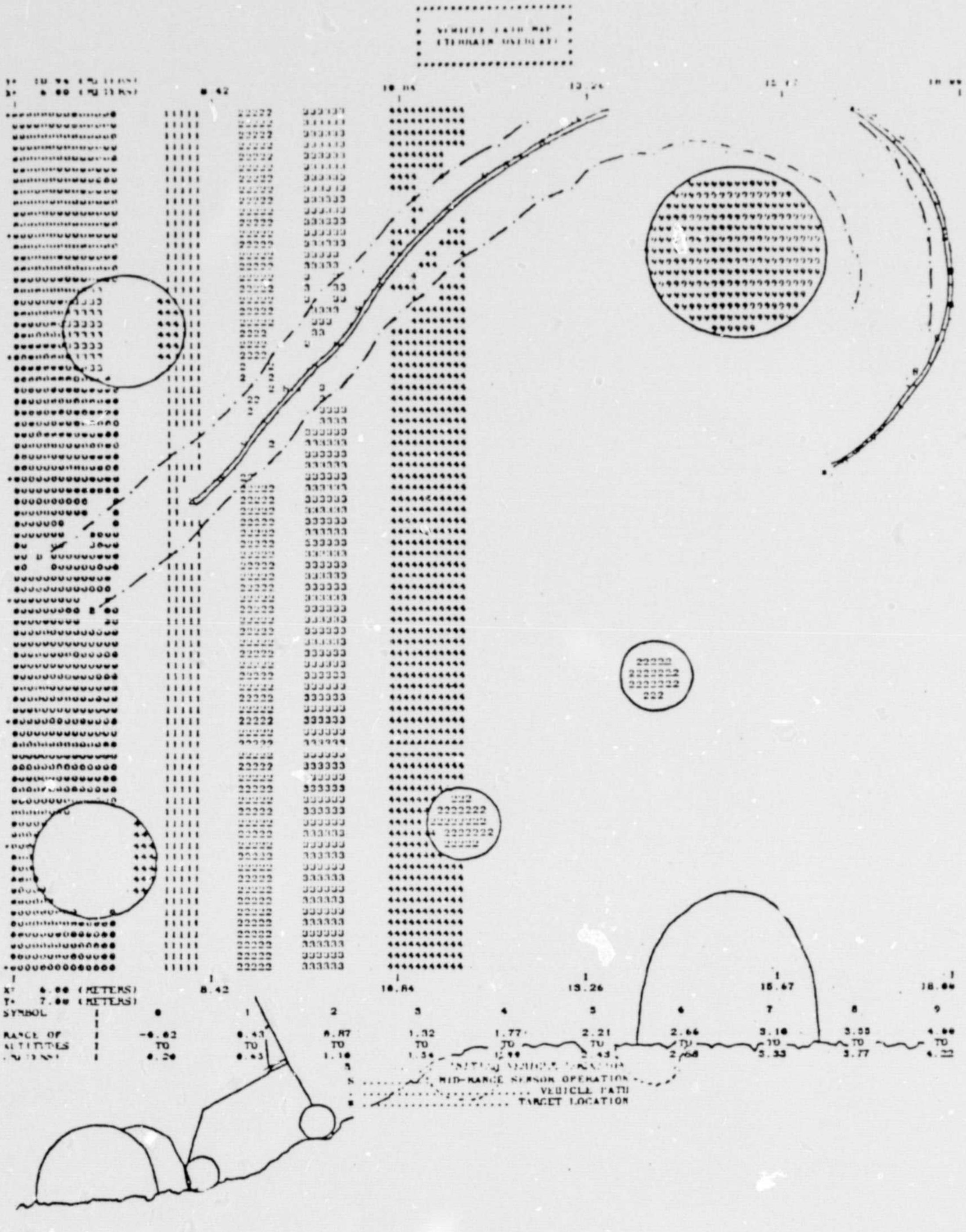


Figure 38B. Boulder-Crater Field, Hills, 10° Attitude Noise (last 18 meters of simulation shown in Figure 37B)

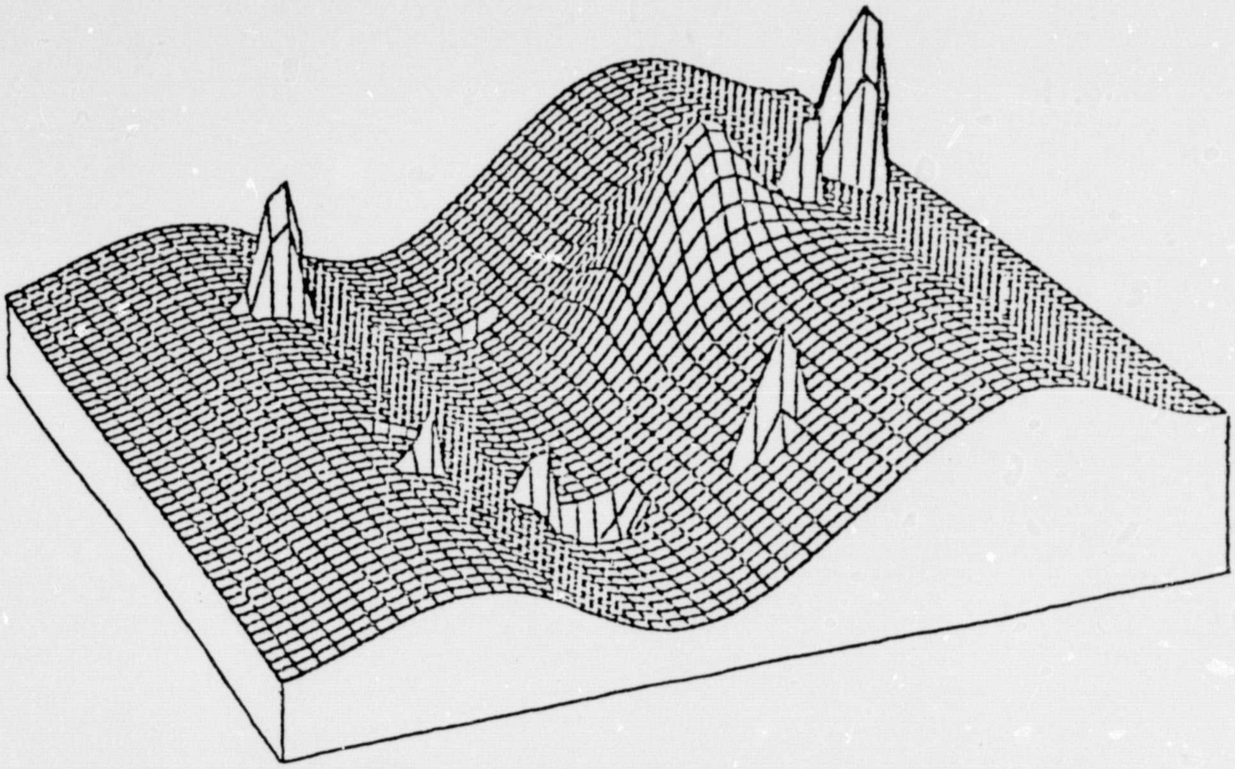
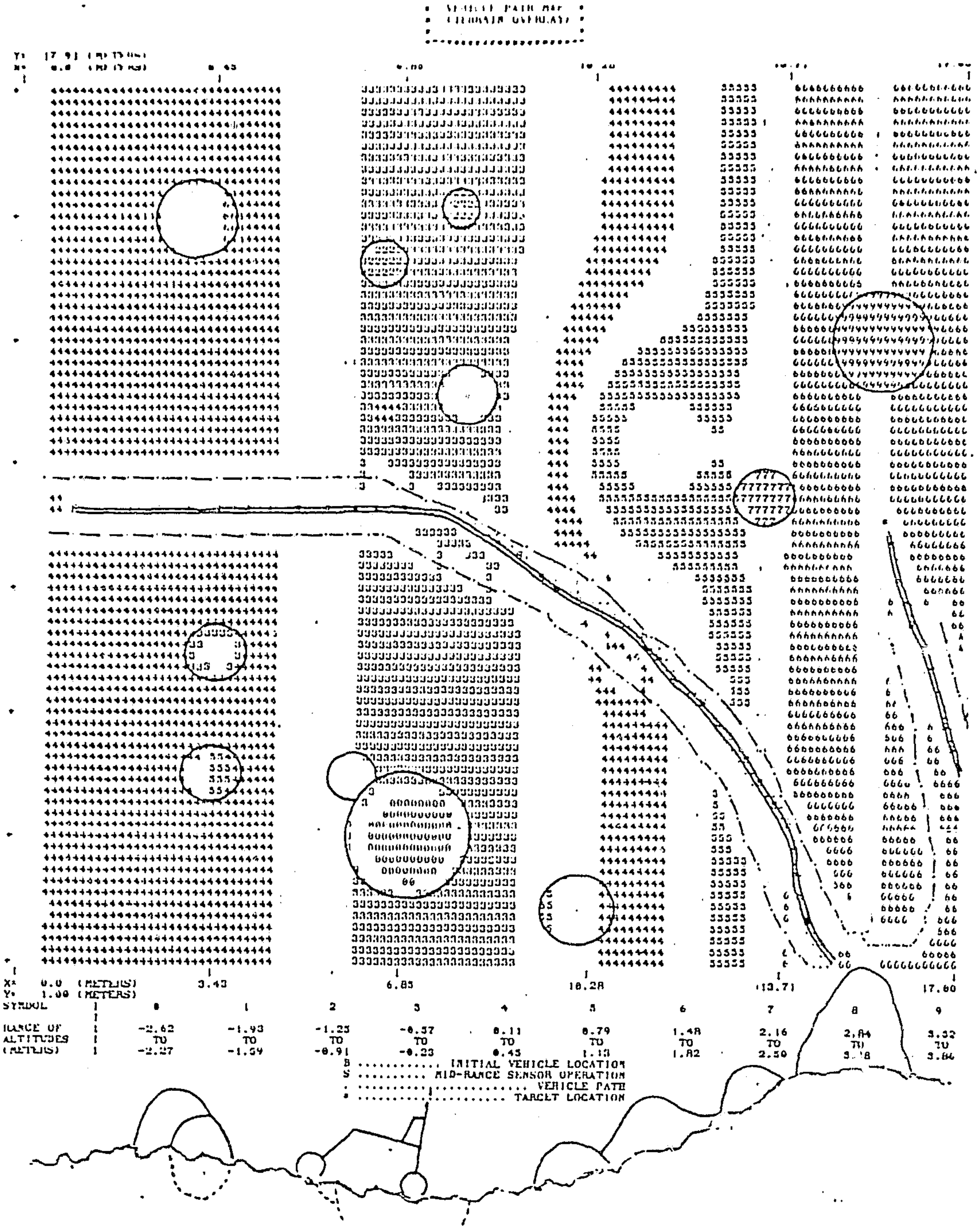


Figure 39A. Three-dimensional Projection of Figure 39B



VEHICLE PATH MAP
(TERRAIN OVERLAY)
Y= 17.91 (METERS)
X= 0.0 (METERS)
3.43
6.85
10.28
13.71
17.60
X= 0.0 (METERS)
Y= 1.00 (METERS)
B .... INITIAL VEHICLE LOCATION
S .... MID-RANGE SENSOR OPERATION
. ... VEHICLE PATH
* .... TARGET LOCATION

SYMBOL	0	1	2	3	4	5	6	7	8	9
RANGE OF ALTITUDES (METERS)	-2.62 TO -2.27	-1.93 TO -1.59	-1.25 TO -0.91	-0.57 TO -0.23	0.11 TO 0.45	0.79 TO 1.13	1.48 TO 1.82	2.16 TO 2.50	2.84 TO 3.18	3.52 TO 3.86

Figure 39B. Boulder-Crater Field, Hills, and 10° Attitude Noise

degree attitude noise, but on a sine wave based terrain with an amplitude of 0.40 meters over a ten meter period. A 1.5 meter hill was added to the top of a crest of the sine wave; boulders and craters completed the view. A large bump on the side of the hill did not prevent the vehicle from arriving at its destination.

5.7 Observations

Each successive simulation posed a more difficult task to the modeler's interpretive abilities. Simple tests examined the reconstruction and classification of various terrain obstacles: small boulders, craters, and slopes that were near the hazard definition for that feature. The crosspath algorithm worked quite well, concluding the first phase of simulations.

With few or no obstacles in its pathway, how the rover would handle a sinusoidal terrain with a period of only three times the vehicle length was of interest. The particular laser/sensor geometry was found to limit the reconstruction of the terrain accurately when the vehicle had a rapidly changing pitch. In each of the boulder-crater field tests the rover reached its target - even when the vehicle was almost bouncing due to the "rubble" beneath its tires. Although some of these tests have been rather extreme, they give an example of how the modeler reacts under the most adverse of conditions.

PART 6

CONCLUSIONS

A terrain modeler employing vastly different techniques than the methods used by previous modeling schemes has been proposed in this paper. Each of the earlier modeling attempts encountered problems with its slope estimations when terrain steps were present. This was due to the quantization errors that are inherent with a laser/sensor data collection system of the type used on the Mars Rover. These errors were greatly reduced and almost eliminated when a thinning algorithm removed data points that were too closely spaced (in range) before any further processing was performed.

The heart of the modeler was a dual filtering scheme that essentially separated the problem of identifying step and slope hazards. The returns from the scanning elevation mast were known height and range points. This raw data was lightly filtered for the terrain height estimates. These were examined for sudden steps that the rover might not be able to climb. Some raw data was more heavily filtered forming a series of slower varying points which were used for the slope estimates.

The results of initial computer simulations indicated excellent predictions by the modeler of the actual terrain. Boulders and craters were distinguishable to within a few centimeters; slopes (both up and downhill) were determined to within one or two degrees. The fact that negative obstacles were more readily apparent is significant in itself; several earlier modelers had problems detecting them. A crosspath obstacle detection algorithm that had been implemented also was found to work extemely well. Crosspath hazards were easily avoided.

A second round of simulations involved moving the rover over a variety of surfaces: sinusoidal terrain with a period of a few times the vehicle length, smooth terrain with boulder and crater obstacles, rubble strewn boulder-crater fields, and finally realistic martian landscapes. The modeler performed well under all of these circumstances, flagging hazards to be avoided by the path selection algorithm.

The new terrain modeler detected every type of hazard as well as, or better than any previous modeler. Minor refinements might benefit the modeler, but improvements should probably be concentrated in other areas. As was pointed out a number of times in this paper, the modeler would have performed better if there had been more lasers and sensors, increasing the range of sight for the vehicle. There are however, limits to be considered. Adding more lasers/sensors increases not only the difficulty in controlling them, but increases the complexity and the time needed for the software to interpret the data. Exactly how closely spaced the laser shots have to be should be reviewed. It may be that information of the terrain further from the vehicle could be gathered using the same number of lasers by varying the geometry of the mast, without significantly affecting the excellent near field results.

REFERENCES

1. Krajewski, M.T., "Development and Evaluation of a Short Range Path Selection System for an Autonomous Planetary Rover", Master of Engineering Report, Rensselaer Polytechnic Institute, Troy, N.Y., May 1976

2. Maroon, G., "Development of a Multiple Laser-Detector Hazard Detection System for an Autonomous Martian Rover", Master of Engineering Report, Rensselaer Polytechnic Institute, Troy, N.Y., May 1977

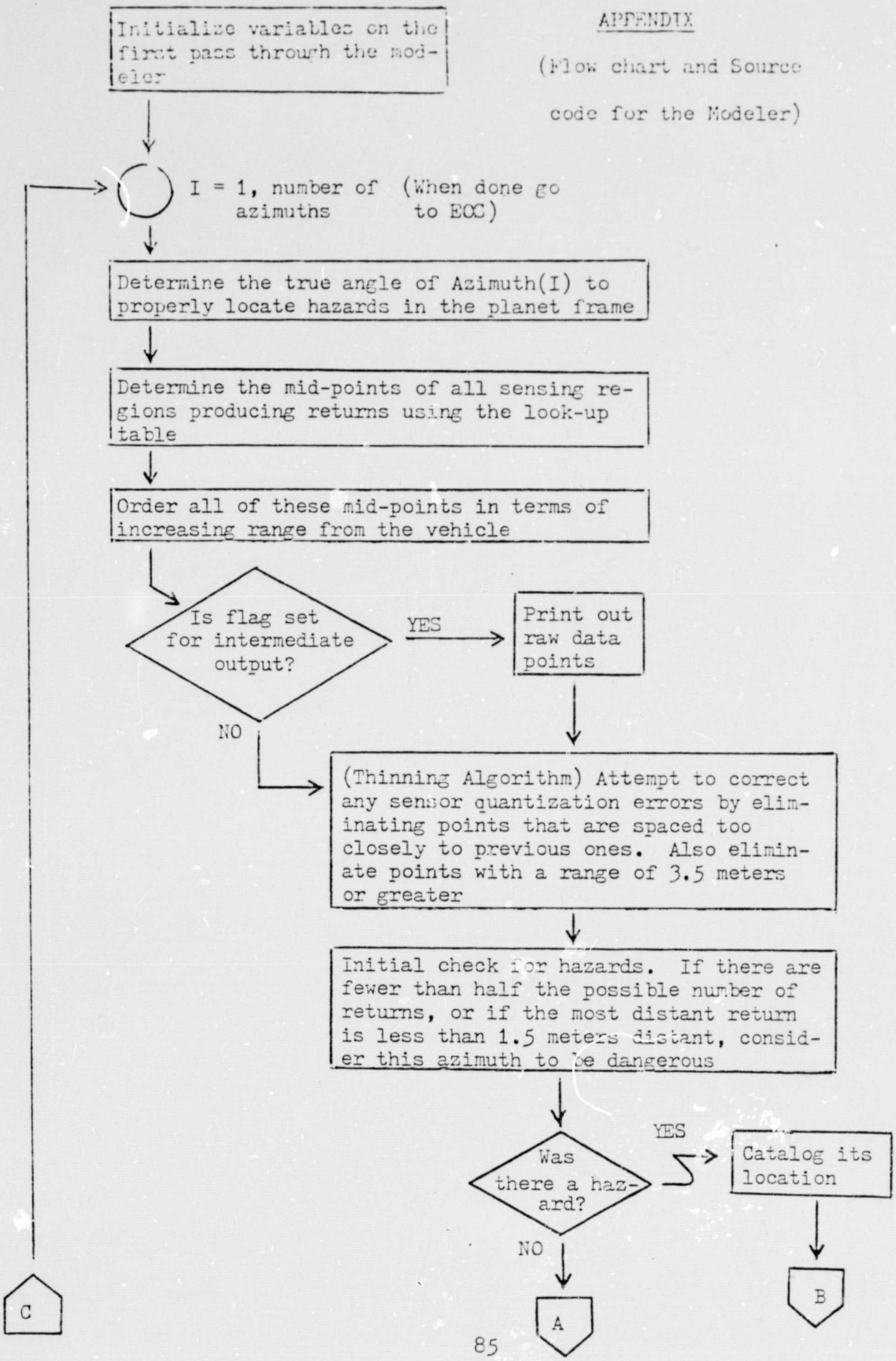
3. Troiani,N. and Yerazunis, S.W., "Procedures for the Interpretation and Use of Elevation Scanning Laser/Multi-Sensor Data for Short Range Hazard Detection and Avoidance for an Autonomous Planetary Rover", RPI Technical Report MP-57, Rensselaer Polytechnic Institute, Troy, N.Y., July 1978

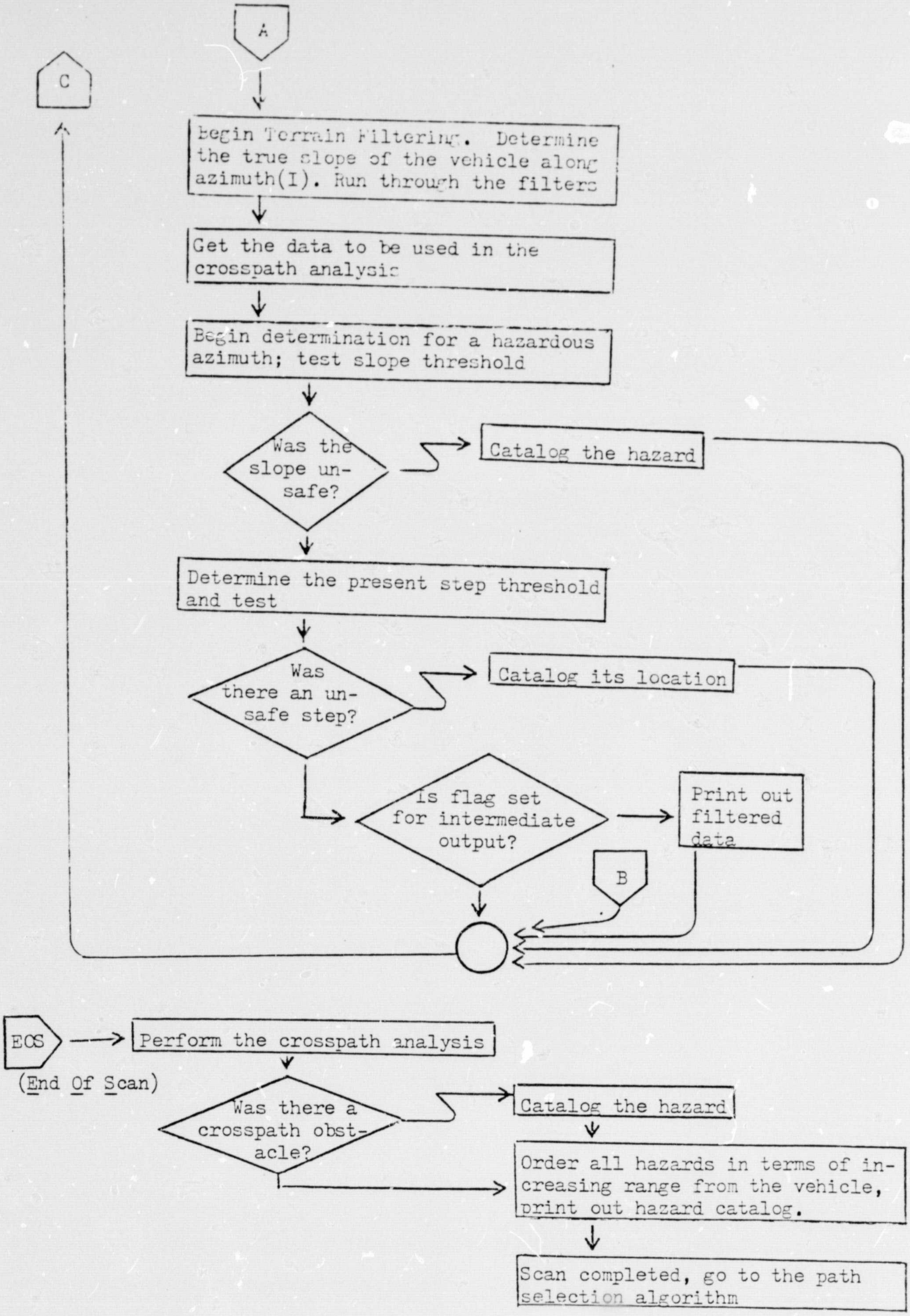
4. Hunter, E.L., "An Advanced Terrain Modeler for an Autonomous Planetary Rover", Master of Engineering Report, Rensselaer Polytechnic Institute, Troy, N.Y., 1979

5. The Viking Lander Imaging Team, "The Martian Landscape", U.S. Government Printing Office, Washinton D.C., 1978

APPENDIX

(Flow chart and Source code for the Modeler)


Initialize variables on the first pass through the modeler
I = 1, number of azimuths (When done go to EOC)
Determine the true angle of Azimuth(I) to properly locate hazards in the planet frame
Determine the mid-points of all sensing regions producing returns using the look-up table
Order all of these mid-points in terms of increasing range from the vehicle
Is flag set for intermediate output?
YES
Print out raw data points
NO
(Thinning Algorithm) Attempt to correct any sensor quantization errors by eliminating points that are spaced too closely to previous ones. Also eliminate points with a range of 3.5 meters or greater
Initial check for hazards. If there are fewer than half the possible number of returns, or if the most distant return is less than 1.5 meters distant, consider this azimuth to be dangerous
Was there a hazard?
YES
Catalog its location
NO
C
A
B


A
C
Begin Terrain Filtering. Determine the true slope of the vehicle along azimuth(I). Run through the filters
Get the data to be used in the crosspath analysis
Begin determination for a hazardous azimuth; test slope threshold
Was the slope unsafe?
Catalog the hazard
Determine the present step threshold and test
Was there an unsafe step?
Catalog its location
Is flag set for intermediate output?
Print out filtered data
B
EOS
(End Of Scan)
Perform the crosspath analysis
Was there a crosspath obstacle?
Catalog the hazard
Order all hazards in terms of increasing range from the vehicle, print out hazard catalog.
Scan completed, go to the path selection algorithm

```
      SUBROUTINE MODEL1
C***********************************************************************
C
C   SUBROUTINE MODEL1
C
C   PURPOSE:
C          THIS SUBROUTINE ACCEPTS SENSOR DATA FROM SENSR1 AND
C          PROCESSES IT TO FIND AND CATALOG HAZARDS.
C
C   METHOD:
C         THE METHOD USED IS A TERRAIN ESTIMATION TECHNIQUE WHERE RAW
C         SENSOR DATA IS PASSED THROUGH A DUAL FIRST ORDER FILTERING
C         SCHEME.  THE FIRST FILTER IS A HIGH-PASS FILTER WHICH
C         ESTIMATES THE TERRAIN HEIGHT.  THE SECOND FILTER IS A LOW-PASS
C         FILTER WHICH USES THE HEIGHT ESTIMATES TO PRODUCE A SLOPE ESTIMATE.
C         AFTER CONVERSION TO THE PLANET FRAME OF REFERENCE, THE RESULTING
C         VALUES ARE COMPARED TO THE APPROPRIATE THRESHOLD VALUES.
C         HAZARDS ARE THEN CATALOGED.
C         IN ADDITION TO THE IN-PATH ANALYSIS DESCRIBED ABOVE, A CROSS
C         PATH EXAMINATION IS ALSO PERFORMED AT VARIOUS DISTANCES FROM
C         THE VEHICLE. THIS CROSS-PATH ANALYSIS FREQUENTLY FINDS HAZ-
C         ARDS THAT ARE NOT NOTICED BY THE IN-PATH PROCESSING. THESE
C         HAZARDS ARE ALSO CATALOGED. THIS CATALOG IS MADE
C         AVAILABLE FOR USE BY THE PATH SELECTION ALGORITHM.
C
C    INPUT:
C          A     - HEIGHT FILTER COEFFICIENT
C          B     - SLOPE FILTER COEFFICIENT
C          STTH  - LEVEL GROUND STEP THRESHOLD (METERS)
C          SLTH  - SLOPE THRESHOLD (DEGREES)
C          RH    - ASSUMED HAZARD RADIUS (METERS)
C
C    OUTPUT:
C           XHAZ(N)  - X COORDINATE OF HAZARD N (PLANET FRAME)
C           YHAZ(N)  - Y COORDINATE OF HAZARD N (PLANET FRAME)
C           RHAX(N)  - ESTIMATED RANGE OF HAZARD N (VEHICLE FRAME)
C           IHAZ     - NUMBER OF HAZARDS CATALOGED THIS SCAN
C           RH       - ASSUMED HAZARD RADIUS (METERS)
C
C
C***********************************************************************
      COMMON /SENX/
     1       HITLAS,HITSEN,NUMLAS,NUMSEN,NUMAZ,INTDAT,NMDTPR,LASAGL,
     2       SENGLE,SCON,DATA,DIAG,POS
      COMMON /TIEUP/
     1       THETNU,ALPHA,SLPIN,SLPCRS,TALLOW
      COMMON /CHOOSE/
     1       NMMOD,NMSEN,NMPSA,NMFAIL,INTVDB,INTSEN,INTMOD,INTPSA,
     2       INTFAL,INTCYR
      COMMON /VEHCLE/
     1       VELNOW,HEADNG,XMRV,YMRV,ZMRV,TARLOX,TARLOY,TARLOZ
      COMMON /SENSR/
     1       ASMUTH,LANGLE,RTN
      COMMON /XMOD/
     1       XHAZ,YHAZ,RHAZ,IHAZ,RH
      COMMON /RMOD/
     1       RANGE

      REAL SX(100),XHAZ(100),YHAZ(100),RHAZ(100),TZ(50),TR(50)
      REAL LASAGL(50),SENGLE(50),POS(50,51,2)
      REAL ASMUTH(50),LANGLE(50),RANGE(50)
      REAL CRSZ(7,50),SUMZ(7),CRSAVG(7),DIV(7)
      INTEGER DIAG(50,50),LINE(130),IAZI(100),RTN(50)
      INTEGER*2 DATA(50,50)
      DATA INIT/0/,LINE/130*'_'/
      DRCON=3.14159/180.0
      RDCON=180.0/3.14159
```

```
C*************************************************************************
C
C       VARIABLES USED IN MODEL1
C
C*************************************************************************
C       A         HEIGHT FILTER COEFFICIENT
C       ALPHA     TRUE HEADING OF THE VEHICLE
C       B         SLOPE FILTER COEFFICIENT
C       DELTA     TRUE HEADING ALONG AN AZIMUTH
C       IHAZ      NUMBER OF HAZARDS CATALOGED THIS SCAN
C       RHAZ(N)   ESTIMATED RANGE OF HAZARD N (VEHICLE FRAME)
C       SLTH      SLOPE THRESHOLD (DEGREES)
C       STHRES    CURRENT STEP THRESHOLD (METERS)
C       STTH      LEVEL GROUND STEP THRESHOLD (METERS)
C       SX(N)     ESTIMATED SLOPE ALONG AZIMUTH N (PLANET FRAME)
C       TSID      ESTIMATED SLOPE ALONG AZIMUTH (VEHICLE FRAME)
C       TR(N)     RANGE OF LASER/SENSOR INTERSECTION (RAW AND FILTERED DATA)
C       TZ(N)     HEIGHT OF LASR/SENSOR INTERSECTION (RAW AND FILTERED DATA)
C       XHAZ(N) X CORRDINATE OF HAZARD N (PLANET FRAME)
C       YHAZ(N) Y CORRDINATE OF HAZARD N (PLANET FRAME)
C
C*************************************************************************
C
C       INITIALIZE THE SUBROUTINE ON THE FIRST PASS, THEN RETURN.
C
C*************************************************************************
        IF(INIT.NE.0) GOTO 100
        READ(5,10) A,B,STTH,SLTH,RH
10      FORMAT(5F10.2)
        WRITE(6,20) A,B,STTH,SLTH,RH
20      FORMAT(////51X,'**  MODEL1 PARAMETERS  **'///41X,'HEIGHT FILTER ',
     1 'COEFFICIENT = ',F10.2//41X,'SLOPE FILTER COEFFICIENT = ',F10.2//
     2 41X,'LEVEL GROUND STEP THRESHOLD = ',F10.2,' METERS'//41X,
     3 'SLOPE THRESHOLD = ',F10.2,' DEGREES'//41X,'ASSUMED HAZARD ',
     4 'RADIUS = ',F10.2,' METERS')
        SLTH=SLTH*DRCON
        DO 30 I=1,100
        SX(I)=0.0
        XHAZ(I)=0.0
        YHAZ(I)=0.0
        RHAZ(I)=0.0
        IAZI(I)=0
30      CONTINUE
        DO 40 I=1,50
        RTN(I)=1
        RANGE(I)=1000.0
40      CONTINUE
        INIT=1
        RETURN
100     CONTINUE
C*************************************************************************
C
C        THE SUBROUTINE HAS BEEN INITIALIZED. IF INTERMEDIATE OUTPUT FROM
C        MODEL1 IS DESIRED, PRINT OUT THE HEADING.
C
C*************************************************************************
        IF(INTMOD.EQ.0) GO TO 120
        WRITE(6,110)
110     FORMAT(//51X,'**  MODEL1 OUTPUT  **'//)
120     CONTINUE
C*************************************************************************
C
C        RUN THROUGH THIS LOOP ONCE PER AZIMUTH TO PERFORM THE IN-PATH TERRAIN
C        PROCESSING.
C
C*************************************************************************
        IHAZ=0
        DO 900 IAZ=1,NUMAZ
        RTN(IAZ)=1
        RANGE(IAZ)=1000.0
        ISAVH=IHAZ
```

```
C***********************************************************************
C
C       FIRST, DETERMINE TRUE AZIMUTH ANGLE TO PROPERLY LOCATE HAZARDS
C       IN THE PLANET FRAME.
C
C***********************************************************************
        TEMP=HEADNG+ASMUTH(IAZ)+ALPHA
        IF(TEMP.LT.0.0) TEMP=6.28318+TEMP
        IF(TEMP.GT.6.28318) TEMP=TEMP-6.28318
C***********************************************************************
C
C       DETERMINE MIDPOINT OF ALL SENSE REGIONS PRODUCING RETURN SIGNALS
C       AND STORE THEM.
C
C***********************************************************************
        ICOUNT=0
        DO 200 IL=1,NUMLAS
        IHOLD=DATA(IAZ,IL)
        IF(IHOLD.LE.0) GO TO 200
        ICOUNT=ICOUNT+1
        IPLS=IHOLD+1
        TZ(ICOUNT)=(POS(IL,IPLS,1)+POS(IL,IHOLD,1))/2.0
        TR(ICOUNT)=(POS(IL,IPLS,2)+POS(IL,IHOLD,2))/2.0
        SX(ICOUNT)=1.0
200     CONTINUE
C***********************************************************************
C
C       ORDER THESE POINTS IN TERMS OF INCREASING RANGE.
C
C***********************************************************************
        IX=ICOUNT-1
        IF(IX.LE.0) GO TO 230
        DO 220 IJ=1,IX
        IR=IJ
210     IPLS=IR+1
        IF(TR(IPLS).GE.TR(IR)) GO TO 220
        XHOLD=TR(IPLS)
        TR(IPLS)=TR(IR)
        TR(IR)=XHOLD
        XHOLD=TZ(IPLS)
        TZ(IPLS)=TZ(IR)
        TZ(IR)=XHOLD
        IR=IR-1
        IF(IR.LE.0) GO TO 220
        GO TO 210
220     CONTINUE
230     CONTINUE
C***********************************************************************
C
C       PRINT OUT RAW DATA SAMPLE POINTS IF DESIRED.
C
C***********************************************************************
        IF(INTMOD.EQ.0) GO TO 390
        WRITE(6,300) LINE
300     FORMAT('0',130A1)
        AZH=ASMUTH(IAZ)*RDCON
        WRITE(6,310) AZH
310     FORMAT('0AZIMUTH = ',F7.2,' DEGREES'/'0SAMPLED TERRAIN POINTS:')
        IF(ICOUNT.EQ.0) WRITE(6,320)
320     FORMAT('0NO SAMPLE POINTS')
        IF(ICOUNT.EQ.0) GO TO 390
        IS=ICOUNT
        IF(ICOUNT.GT.20) IS=20
        WRITE(6,330) (TR(I),I=1,IS)
330     FORMAT('0RANGE: ',2X,20(1X,F5.2))
        WRITE(6,340) (TZ(I),I=1,IS)
340     FORMAT(' HEIGHT:',1X,20(1X,F5.2))
        IF(ICOUNT.LE.20) GO TO 390
        IS=ICOUNT
        IF(ICOUNT.GT.40) IS=40
        WRITE(6,350) (TR(I),I=21,IS)
350     FORMAT('0RANGE:',2X,20(1X,F5.2))
        WRITE(6,360) (TZ(I),I=21,IS)
360     FORMAT(' HEIGHT:',1X,20(1X,F5.2))
        IF(ICOUNT.LE.40) GO TO 390
        IS=ICOUNT
        WRITE(6,370) (TR(I),I=41,IS)
370     FORMAT('0RANGE:',2X,20(1X,F5.2))
        WRITE(6,380) (TZ(I),I=41,IS)
380     FORMAT(' HEIGHT:',1X,20(1X,F5.2))
390     CONTINUE
```

```
C***********************************************************************
C
C       ATTEMPT TO CORRECT ANY SENSOR QUANTIZATION ERRORS BY ELIMINATING
C       SAMPLE POINTS SPACED TOO CLOSELY TO PREVIOUS ONES OR POINTS
C       BEYOND 3.5 METERS RANGE. (THINNING ALGORITHM)
C
C***********************************************************************
        IF(ICOUNT.LT.3) GO TO 460
        SUM=0.0
        FTIM=TR(ICOUNT)-TR(1)
        DO 400 IJ=2,ICOUNT
        IF(TR(IJ).GT.3.5) GO TO 410
        IM=IJ-1
        ADIFF=ABS(TR(IJ)-TR(IM))
        IF(ADIFF.GT.FTIM) SUM=SUM+FTIM
        IF(ADIFF.LE.FTIM) SUM=SUM+ADIFF
        AVG=SUM/(IJ-1)
        FTIM=AVG*5.0
400     CONTINUE
410     AAVG=AVG/2.0
        DO 440 IJ=2,ICOUNT
        IF(TR(IJ).GT.3.5) SX(IJ)=0.0
        IF(TR(IJ).GT.3.5) GO TO 440
        IM=IJ-1
420     IF(SX(IM).NE.0.0) GO TO 430
        IM=IM-1
        GO TO 420
430     DIFF=TR(IJ)-TR(IM)
        IF(DIFF.GE.AAVG) GO TO 440
        IF(ABS(TZ(IM)).GE.ABS(TZ(IJ))) SX(IJ)=0.0
        IF(ABS(TZ(IM)).LT.ABS(TZ(IJ))) SX(IM)=0.0
440     CONTINUE
        IP=0
        DO 450 IJ=1,ICOUNT
        IF(SX(IJ).EQ.0.0) GO TO 450
        IP=IP+1
        TZ(IP)=TZ(IJ)
        TR(IP)=TR(IJ)
450     CONTINUE
        ICOUNT=IP
460     CONTINUE
C***********************************************************************
C
C       IF THERE ARE LESS THAN HALF THE POSSIBLE NUMBER OF RETURNS
C       OR IF THE MOST DISTANT RETURN IS LESS THAN 1.5 METERS AWAY
C       THIS AZIMUTH IS CONSIDERED TO BE HAZARDOUS.
C
C***********************************************************************
        IX=NUMLAS/2
        IF(ICOUNT.LE.IX) GO TO 500
        IF(TR(ICOUNT).GE.1.5) GO TO 550
C***********************************************************************
C
C        THERE ARE TOO FEW RETURNS TO MAKE THIS A SAFE PATH. CATALOG AN IN-
C        SUFFICIENT DATA HAZARD.
C
C***********************************************************************
500     IHAZ=IHAZ+1
        IF(IHAZ.GT.100) GO TO 1000
        RTN(IAZ)=0
        IAZ1(IHAZ)=IAZ
        RNG=1.0
        IF(ICOUNT.LE.IX) GO TO 510
        RNG=TR(ICOUNT)
510     RHAZ(IHAZ)=RNG
        RANGE(IAZ)=RNG
```

```
C***********************************************************************
C
C       RANGE IS THE DISTANCE TO THE HAZARD. PU THIS INTO ITS X,Y COORDINATES
C       (PLANET FRAME).
C
C***********************************************************************
        IF(TEMP.GT.1.5708) GO TO 520
        ANG=TEMP
        YHAZ(IHAZ)=YMRV+RNG*COS(ANG)
        XHAZ(IHAZ)=XMRV+RNG*SIN(ANG)
        GO TO 550
520     IF(TEMP.GT.3.14159) GO TO 530
        ANG=3.14159-TEMP
        YHAZ(IHAZ)=YMRV-RNG*COS(ANG)
        XHAZ(IHAZ)=XMRV+RNG*SIN(ANG)
        GO TO 550
530     IF(TEMP.GT.4.71239) GO TO 540
        ANG=4.71239-TEMP
        YHAZ(IHAZ)=YMRV-RNG*SIN(ANG)
        XHAZ(IHAZ)=XMRV-RNG*COS(ANG)
        GO TO 550
540     ANG=6.28318-TEMP
        YHAZ(IHAZ)=YMRV+RNG*COS(ANG)
        XHAZ(IHAZ)=XMRV-RNG*SIN(ANG)
550     CONTINUE
C***********************************************************************
C
C       BEGIN TERRAIN FILTERING.
C       FIRST, DETERMINE THE TRUE SLOPE OF THE VEHICLE ALONG THIS AZIMUTH.
C
C***********************************************************************
        DELTA=ASMUTH(IAZ)+ALPHA
        TSID=ATAN(TAN(SLPIN)*COS(DELTA)-TAN(SLPCRS)*SIN(DELTA))
        SX(1)=TSID
C***********************************************************************
C
C       NOW RUN THROUGH THE FILTERS.
C
C***********************************************************************
        IF(ICOUNT.LE.1) GO TO 810
        ISUP=1
        TSK=0.0
        DO 600 IJ=1,7
        CRSZ(IJ,IAZ)=1000.0
        DIV(IJ)=0.0
        SUMZ(IJ)=0.0
600     CONTINUE
        DO 800 IJ=2,ICOUNT
        IM=IJ-1
        TZ(IJ)=((1.0-A)*TZ(IJ))+(A*TZ(IM))
C***********************************************************************
C
C        GET THE DATA TO BE USED IN THE CROSS-PATH ANALYSIS.
C
C***********************************************************************
        IF(TR(IM).LE.(1.25)) CRSZ(1,IAZ)=TZ(IM)
        IF((TR(IJ).GT.(1.25)).AND.(TR(IJ).LE.(1.5))) CRSZ(2,IAZ)=TZ(IJ)
        IF((TR(IJ).GT.(1.5)).AND.(TR(IJ).LE.(1.75))) CRSZ(3,IAZ)=TZ(IJ)

        IF((TR(IJ).GT.(1.75)).AND.(TR(IJ).LE.(2.0))) CRSZ(4,IAZ)=TZ(IJ)
        IF((TR(IJ).GT.(2.0)).AND.(TR(IJ).LE.(2.25))) CRSZ(5,IAZ)=TZ(IJ)
        IF((TR(IJ).GT.(2.25)).AND.(TR(IJ).LE.(2.5))) CRSZ(6,IAZ)=TZ(IJ)
        IF((TR(IJ).GT.(2.5)).AND.(TR(IJ).LE.(2.75))) CRSZ(7,IAZ)=TZ(IJ)
        TSK=((1.0-B)*ATAN((TZ(IJ)-TZ(IM))/(TR(IJ)-TR(IM))))+(B*TSK)
        ESTSLP=TSID+TSK
        SX(IJ)=ESTSLP
        IF(ISUP.NE.1) GO TO 800
```

```
C***********************************************************************
C
C        BEGIN DETERMINATION FOR HAZARDOUS AZIMUTH.  TEST SLOPE THRESHOLD.
C
C***********************************************************************
         IF(ABS(ESTSLP).GT.SLTH)  IMM=IJ
         IF(ABS(ESTSLP).GT.SLTH)  GO TO 700
C***********************************************************************
C
C    DETERMINE PRESENT STEP THRESHOLD AND TEST.
C
C***********************************************************************
         STHRES=STTH
         AES=ABS(ESTSLP)
         IF(AES.GT.(SLTH*.67).AND.AES.LE.(SLTH*5./6.))STHRES=STTH/2.
         IF(AES.GT.(SLTH*5./6.))STHRES=STTH/4.
         DO 610  ICHK=1,IM
         IMM=IJ-ICHK
         DIFF=TR(IJ)-TR(IMM)
         IF(DIFF.LT.STHRES)  GO TO 610
         IF(ICHK.EQ.1)  GO TO 620
         IF(DIFF.EQ.STHRES)  GO TO 620
         IMM=IMM+1
         GO TO 620
610      CONTINUE
620      ESTDH=TZ(IJ)-TZ(IMM)
         ESTDH=ABS(ESTDH)
         IF(ESTDH.LE.STHRES)  GO TO 800
C***********************************************************************
C
C        CATALOG SLOPE OR STEP HAZARD.
C
C***********************************************************************
700      CONTINUE
         ISUP=0
         RNG=TR(IMM)
         IHAZ=IHAZ+1
         IF(IHAZ.GT.100)  GO TO 1000
         IAZI(IHAZ)=IAZ
         RTN(IAZ)=0
         RHAZ(IHAZ)=RNG
         RANGE(IAZ)=RNG
C***********************************************************************
C
C         FIND THE X,Y COORDINATE OF EACH HAZARD.
C
C***********************************************************************
         IF(TEMP.GT.1.5708)  GO TO 720
         ANG=TEMP
         YHAZ(IHAZ)=YMRV+RNG*COS(ANG)
         XHAZ(IHAZ)=XMRV+RNG*SIN(ANG)
         GO TO 800
720      IF(TEMP.GT.3.14159)  GO TO 730
         ANG=3.14159-TEMP
         YHAZ(IHAZ)=YMRV-RNG*COS(ANG)
         XHAZ(IHAZ)=XMRV+RNG*SIN(ANG)
         GO TO 800
730      IF(TEMP.GT.4.71239)  GO TO 740
         ANG=4.71239-TEMP
         YHAZ(IHAZ)=YMRV-RNG*SIN(ANG)
         XHAZ(IHAZ)=XMRV-RNG*COS(ANG)
         GO TO 800
740      ANG=6.28318-TEMP
         YHAZ(IHAZ)=YMRV+RNG*COS(ANG)
         XHAZ(IHAZ)=XMRV-RNG*SIN(ANG)
800      CONTINUE
```

```
C************************************************************************
C
C       PRINT OUT FILTERED DATA IF DESIRED.
C
C************************************************************************
810     CONTINUE
        IF(INTMOD.EQ.0) GO TO 900
        IF(ICOUNT.EQ.0) GO TO 900
        DO 815 I=1,ICOUNT
        SX(I)=SX(I)*RDCON
815     CONTINUE
        ISAVH=IHAZ-ISAVH
        WRITE(6,820)
820     FORMAT('0TERRAIN ESTIMATES:')
        IS=ICOUNT
        IF(ICOUNT.GT.20) IS=20
        WRITE(6,825) (TR(I),I=1,IS)
825     FORMAT('0RANGE:',2X,20(1X,F5.2))
        WRITE(6,830) (TZ(I),I=1,IS)
830     FORMAT(' HEIGHT:',1X,20(1X,F5.2))
        WRITE(6,835) (SX(I),I=1,IS)
835     FORMAT(' SLOPE:',2X,20(1X,F5.1))
        IF(ICOUNT.LE.20) GO TO 870
        IS=ICOUNT
        IF(ICOUNT.GT.40) IS=40
        WRITE(6,840) (TR(I),I=21,IS)
840     FORMAT('0RANGE:',2X,20(1X,F5.2))
        WRITE(6,845) (TZ(I),I=21,IS)
845     FORMAT(' HEIGHT:',1X,20(1X,F5.2))
        WRITE(6,850) (SX(I),I=21,IS)
850     FORMAT(' SLOPE:',2X,20(1X,F5.1))
        IF(ICOUNT.LE.40) GO TO 870
        IS=ICOUNT
        WRITE(6,855) (TR(I),I=41,IS)
855     FORMAT('0RANGE:',2X,20(1X,F5.2))
        WRITE(6,860) (TZ(I),I=41,IS)
860     FORMAT(' HEIGHT:',1X,20(1X,F5.2))
        WRITE(6,865) (SX(I),I=41,IS)
865     FORMAT(' SLOPE:',2X,20(1X,F5.1))
870     IF(ISAVH.EQ.0) GO TO 900

        IF(ISAVH.EQ.1) WRITE(6,875)
875     FORMAT('0  1 HAZARD CATALOGED ON THIS AZIMUTH.')
        IF(ISAVH.EQ.1) GO TO 900
        WRITE(6,880) ISAVH
880     FORMAT('0',I3,' HAZARDS CATALOGED ON THIS AZIMUTH.')
900     CONTINUE
C************************************************************************
C
C        AT THIS POINT EACH AZIMUTH HAS BEEN EXAMINED AND THE IN-PATH ANALYSIS
C        IS COMPLETE. NOW PERFORM THE CROSS-PATH ANALYSYS.
C
C************************************************************************
        WRITE(6,300) LINE
        CRSPTH=SLPCRS
        STHRES=STTH/.88
        SLOPTH=SLTH
        IF(ABS(CRSPTH).GE.SLOPTH)GOTO 910
        DO 970 IJ=1,7
        DO 970 IAZ=1,NUMAZ
        IF(IAZ.EQ.1)GOTO 970
        IF(CRSZ(IJ,IAZ).LE.100.)GOTO 905
        GOTO 970
905     IF(CRSZ(IJ,IAZ-1).GT.100.)GOTO 970
        CHK=(CRSZ(IJ,IAZ)-CRSZ(IJ,IAZ-1))
        IF(CRSPTH.GE.SLOPTH*.67)GOTO 950
        IF(CRSPTH.LE.(-.67*SLOPTH))GOTO 960
        IF(CRSPTH.GE.(0.0).AND.CHK.LE.-STHRES)GOTO 911
        IF(CRSPTH.LT.(0.0).AND.CHK.GE.STHRES)GOTO 910
        IF(CHK.LE.STHRES)GOTO 970
910     IAZT=IAZ
        GOTO 912
911     IAZT=IAZ-1
912     RNG=1.00+FLOAT(IJ)*.25
```

```
C************************************************************************
C
C       IF THERE IS AN INPATH OBSTACLE CLOSER THAN THE CROSSPATH HAZARD,
C       IGNORE THE CROSSPATH RESULTS; OTHERWISE CATALOG A CROSSPATH HAZARD.
C
C************************************************************************
        IF(IHAZ.EQ.0)GOTO 925
        DO 920 K=1,IHAZ
        IF((IAZI(K).EQ.IAZT).AND.(RHAZ(K).LE.RNG))GOTO 970
920     CONTINUE
925     AZH=ASMUTH(IAZT)*RDCON
        WRITE(6,930)AZH
930     FORMAT(' CROSSPATH HAZARD CATALOGED ON AZIMUTH ',F7.2)
        IHAZ=IHAZ+1
        IAZI(IHAZ)=IAZT
        RTN(IAZT)=0
        RHAZ(IHAZ)=RNG
        RANGE(IAZT)=RNG
        IF(TEMP.GT.1.5708) GOTO 935
        ANG=TEMP
        YHAZ(IHAZ)=YMRV+RNG*COS(ANG)
        XHAZ(IHAZ)=XMRV+RNG*SIN(ANG)
        GOTO 970
935     IF(TEMP.GT.3.14159) GOTO 940

        ANG=3.14159-TEMP
        YHAZ(IHAZ)=YMRV-RNG*COS(ANG)
        XHAZ(IHAZ)=XMRV+RNG*SIN(ANG)
        GOTO 970
940     IF(TEMP.GT.4.71239) GOTO 945
        ANG=4.71239-TEMP
        YHAZ(IHAZ)=YMRV-RNG*SIN(ANG)
        XHAZ(IHAZ)=XMRV-RNG*COS(ANG)
        GOTO 970
945     ANG=6.28318-TEMP
        YHAZ(IHAZ)=YMRV+RNG*COS(ANG)
        XHAZ(IHAZ)=XMRV-RNG*SIN(ANG)
        GOTO 970
950     IF(CRSPTH.GE.SLOPTH*.83)GOTO 955
        IF(CHK.LT.(-.5*STHRES))GOTO 911
        GOTO 970
955     IF(CHK.LT.(-.25*STHRES))GOTO 911
        GOTO 970
960     IF(CRSPTH.LE.(-.83*SLOPTH))GOTO 965
        IF(CHK.GT.(.5*STHRES))GOTO 910
        GOTO 970
965     IF(CHK.GT.(.25*STHRES))GOTO 910
970     CONTINUE
        GO TO 1020
C************************************************************************
C
C       HAZARD LIMIT EXCEEDED.
C
C************************************************************************
1000    WRITE(6,1010)
1010    FORMAT(///' OVER 100 HAZARDS CATALOGED - PROCESSING TERMINATED.')
```

```
C***********************************************************************
C
C       ORDER ALL HAZARDS IN ORDER OF INCREASING RANGE.
C
C***********************************************************************
1020    CONTINUE
        IX=IHAZ-1
        IF(IX.LE.0) GO TO 1050
        DO 1040 IJ=1,IX
        IR=IJ
1030    IPLS=IR+1
        IF(RHAZ(IPLS).GE.RHAZ(IR)) GO TO 1040
        XHOLD=RHAZ(IPLS)
        RHAZ(IPLS)=RHAZ(IR)
        RHAZ(IR)=XHOLD
        XHOLD=XHAZ(IPLS)
        XHAZ(IPLS)=XHAZ(IR)
        XHAZ(IR)=XHOLD
        XHOLD=YHAZ(IPLS)
        YHAZ(IPLS)=YHAZ(IR)
        YHAZ(IR)=XHOLD
        IHOLD=IAZI(IPLS)
        IAZI(IPLS)=IAZI(IR)
        IAZI(IR)=IHOLD
        IR=IR-1
        IF(IR.LE.0) GO TO 1040
        GO TO 1030
1040    CONTINUE
1050    CONTINUE
C***********************************************************************
C
C       PRINT OUT HAZARD CATALOG IF DESIRED.
C
C***********************************************************************
        WRITE(6,1100) LINE
1100    FORMAT('0',130A1)
        IF(IHAZ.EQ.0) WRITE(6,1110)
1110    FORMAT('0',35X,'NO HAZARDS CATALOGED THIS SCAN')
        IF(IHAZ.EQ.0) GO TO 1150
        WRITE(6,1120)
1120    FORMAT('0',41X,'HAZARD CATALOG'//24X,'AZIMUTH',10X,'RANGE',11X,
     1 'X',14X,'Y'/25X,'(DEG)',12X,'(M)'/)
        DO 1140 I=1,IHAZ
        IA1=IAZI(I)
        AOUT=ASMUTH(IA1)*RDCON
        WRITE(6,1130) AOUT,RHAZ(I),XHAZ(I),YHAZ(I)
1130    FORMAT(21X,F10.2,5X,F10.2,5X,F10.2,5X,F10.2)
1140    CONTINUE
1150    WRITE(6,1160)
1160    FORMAT('0')
1200    CONTINUE
        RETURN
        END
```