1970

A Dual-Mode Routing Algorithm for an Autonomous Roving Vehicle

Kirk, Donald E.; Lim, Larry Y.

A Dual-Mode Routing Algorithm for an Autonomous Roving Vehicle

DONALD E. KIRK, Member, IEEE
Naval Postgraduate School
Monterey, Calif.

LARRY Y. LIM
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, Calif.

Abstract

A dual-mode algorithm for routing an unmanned autonomous roving vehicle designed to explore the uncertain terrain of other planets is presented. The algorithm consists of a global mode, which uses dynamic programming and terrain information available from photo reconnaissance data to determine a nominal optimal path, and a local mode, which routes the vehicle around obstacles whose presence, location, and extent are not known in advance. Gaussian probability density functions are used to simulate terrain for examples that illustrate the performance of the algorithm.

Manuscript received October 9, 1969; revised November 20, 1969.

This work was performed at the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, and was supported by NASA Contract NAS 7-100.

Introduction

Plans for future space missions in the Mars exploration series include an unmanned roving vehicle designed to explore the Martian surface and transmit television pictures and other scientific data to earth. Because of the long transit times required for transmission of television information from Mars to earth, it is not feasible for an earth-based operator to control the vehicle from observations obtained by monitoring a television picture of the terrain; therefore, the vehicle is to be controlled by a computer having a limited weight and limited memory capacity, that is, either on board or in an orbiting satellite. This paper presents a routing algorithm—consisting of a *local* and a *global* mode—that has been proposed for providing heading commands to the roving vehicle.

The local mode of the algorithm utilizes only information available from the vehicle's sensors to determine a path from a specified starting point to a designated terminal point. This mode provides a means of circumnavigating hazards of arbitrary size and shape. The primary limitation of the local mode is its inability to utilize terrain information which will be available from a previous Mars orbiter mission. As a result, using the local mode alone will determine a path from point to point, but this path may be far from desirable in terms of distance traveled, energy expended, or some other measure of system performance. To alleviate this difficulty, a global mode is employed to determine a nominal optimal path from gross terrain information available from a previous Mars orbiter mission. In the dual-mode strategy the vehicle follows the precomputed nominal optimal path as long as no hazards are detected by on-board sensors; detection of an obstacle initiates a transfer to the local mode which finds a path around the obstacle to another point on the nominal optimal path where the global mode resumes control.

The Local Mode

The description of the local mode of the path-finding algorithm requires the following definitions.

1) *Acceptable Point:* An acceptable point is defined as a point (X, Y) in a two-dimensional Euclidean space (E^2) such that $|F(X, Y)| \leq E$, where $F(X, Y)$ is a continuous function that describes the elevation of the terrain and E is the elevation limit.

2) *Path:* A path is defined as a route that connects two points (X_i, Y_i) and (X_j, Y_j) such that the elevation of every point of the route satisfies $|F(X, Y)| \leq E$.

3) *Obstacle:* An obstacle is defined as a finite bounded region in the domain of the elevation function F where the magnitude of F is greater than E, i.e., $|F(X, Y)| > E$.

4) *Target Distance Sequence:* Let (X_o, Y_o) and (X_n, Y_n) be the coordinates of the initial point P_o and target point P_n, respectively. The target distance sequence generated

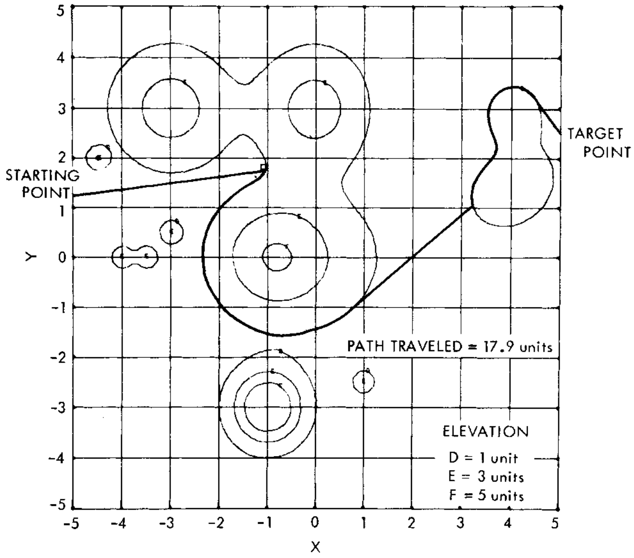


Fig. 1. Path from local mode only.

by moving toward the target point is $\{d_1, d_2, \cdots, d_n\} = \{d_k\}$ where the kth member of the sequence is defined as

$$d_k = [(X_n - X_{k-1})^2 + (Y_n - Y_{k-1})^2]^{\frac{1}{2}}. \qquad (1)$$

An obvious property that follows from 4) is that, if there exists a path from the starting point to the target point, then there exists a subsequence $\{d_{kj}\}$ of $\{d_k\}$ that is monotonically decreasing, and the last term of the sequence is $d_n \leq \varepsilon$, where ε is an arbitrary constant. The local mode of the path-finding algorithm is based on two propositions whose proof can be found in [1].

Proposition 1: If there exists a path from the point P_o to the point P_n, and there are no obstacles between these two points, then the path can be found.

Proposition 2: If a path exists from the point P_o to the point P_n, and there exists at least one obstacle between the two points, then following either the right or left contour of the obstacle and always heading toward the target point P_n will result in determining a path P_oP_n.

Local Mode Algorithm

Step 0: Compute $d_m = [(X_n - X_o)^2 + (Y_n - Y_o)^2]^{\frac{1}{2}}$.

Step 1: From P_o go directly to P_n if there are no obstacles between the starting point and the target point. Every point generated in this case is an acceptable point, and every d_i generated is such that $d_j < d_i$ for $j > i$, where

$$d_i = [(X_n - X_{i-1})^2 + (Y_n - Y_{i-1})^2]^{\frac{1}{2}} \qquad (2)$$

with

$$X_i = X_{i-1} + R\cos\theta_i, \quad Y_i = Y_{i-1} + R\sin\theta_i, \quad i = 1, 2, \cdots, N$$

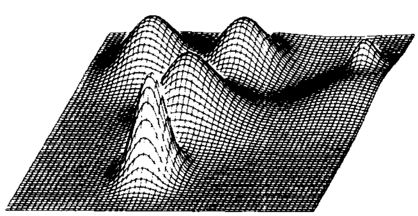
$$\theta_i = \tan^{-1}\left(\frac{Y_n - Y_{i-1}}{X_n - X_{i-1}}\right)$$

where R is the scanning range. However, if there is an obstacle encountered at point P_k with coordinates (X_k, Y_k), replace d_m by d_k, and go to Step 2.

Step 2: At point P_k, scan alternately right then left at appropriate increments in the angle θ until an acceptable point is found. Let P_{k+1} be the acceptable point. If $d_{k+1} < d_k$, replace d_m by d_{k+1}, P_k by P_{k+1}, and return to Step 1. If $d_{k+1} \geq d_k$, go to Step 3.

Step 3: Determine whether the point P_{k+1} was the result of scanning right or scanning left. If P_{k+1} was the result of the right scanning process, stay on the right contour of the obstacle. Similarly, if P_{k+1} was determined from the left scanning process, stay on the left contour of the obstacle. In either case, the right or left contour is followed until an acceptable point P_q (X_q, Y_q) that satisfies the relationship $d_m > d_q$ is found. Once this relationship has been satisfied, replace d_m by d_q, P_{q-1} by P_q, and return to Step 1. The process is repeated until P_n is reached.

Fig. 2. Gaussian terrain.



Local Mode Example

The local mode algorithm was coded in FORTRAN IV as implemented in the IBM 7090/94 IBJOB system. An example of a route generated by the local mode program is shown in Fig. 1. The terrain function used for the simulation is the summation of several Gaussian functions—this provides the elevation contours shown in Fig. 2.

Because the local mode uses only local terrain information, the determined path may be far from desirable in terms of distance traveled, energy expended, or some other measure of system performance. Furthermore, gross terrain information will normally be available from orbital reconnaissance missions before the landing of a planetary roving vehicle. Therefore, the global mode was introduced to overcome the above limitations and to take advantage of the orbital reconnaissance information.

The Global Mode

Let (X_i, Y_i), $i=1, 2, \cdots, N$, be the coordinates of a set of N points (or nodes) in two-dimensional Euclidean space (E^2). Let the cost of moving from the point P_k with coordinates (X_k, Y_k) to the point P_j with coordinates (X_j, Y_j) along the straight line joining these two points be denoted by t_{kj}. Assume that reconnaissance data from previous Mars orbiter mission has been used to select this grid of N points, and that the values of t_{kj} corresponding to these grid points have been calculated and stored in a matrix $\boldsymbol{T}$ whose elements have the properties

$$t_{kj}\begin{cases} >0 & \text{if } k \neq j \\ =0 & \text{if } k = j \end{cases} \qquad k,j = 1, \cdots, N. \tag{3}$$

The problem is to find a sequence of nodes to be traversed in moving from an initial point P_k to any other specified grid point P_j so that the minimum cost is incurred; such a path will be called an "optimal path" from P_k to P_j. From the principle of optimality [2] the functional recurrence equation

$$c_{kj} = \min_{\substack{i \\ i\neq k}} \{t_{ki} + c_{ij}\}, \qquad k,j = 1, 2, \cdots, N \tag{4}$$

is obtained. c_{kj}—the kjth entry of the cost matrix $\boldsymbol{C}$—is, by definition, the minimum cost of going from point P_k to point P_j via any number of intermediate points. Clearly, the number of intermediate points cannot exceed $(N-2)$ since this implies that at least one loop exists, and eliminating such a loop reduces the cost. The initial condition for (4) is

$$c_{kk} = 0, \qquad k = 1, 2, \cdots, N.$$

To solve (4) directly is a difficult task because the quantities to be determined (the c's) appear on both sides of the equation; however, one way to obtain a solution is to use Picard's method of successive approximations; that is,

$$c_{kj}^{(l+1)} = \min_{\substack{i \\ i\neq k}} \{t_{ki} + c_{ij}^{(l)}\} \qquad k,j = 1, 2, \cdots, N \tag{5}$$

with the initial values

$$c_{kj}^{(0)} = t_{kj} \qquad k,j = 1, 2, \cdots, N. \tag{6}$$

These successive approximations have the following physical interpretation:

1) $c_{kj}^{(0)}$ is the minimum cost to go from node P_k to node P_j directly, i.e., via no intermediate nodes.

2) The minimum cost to go from node P_k to node P_j via *at most* one intermediate node is

$$c_{kj}^{(1)} = \min_{\substack{i \\ i\neq k}} \{t_{ki} + c_{ij}^{(0)}\} = \min_{\substack{i \\ i\neq k}} \{t_{ki} + t_{ij}\}.$$

3) The minimum cost to go from P_k to P_j via at most $(l+1)$ intermediate nodes is

$$c_{kj}^{(l+1)} = \min_{\substack{i \\ i\neq k}} \{t_{ki} + c_{ij}^{(l)}\}.$$

To solve for $c_{kj}^{(l+1)}$, t_{ki}, $i\neq k$, $i=1, 2, \cdots, N$ (the kth row of the matrix $\boldsymbol{T}$) and $c_{ij}^{(l)}$, $i\neq j$, $i=1, 2, \cdots, N$ (the jth column of the $\boldsymbol{C}^{(l)}$ matrix) are required. The solution is found simply by comparing the values of $t_{ki}+c_{ij}^{(l)}$ for $i=1, 2, \cdots, N$, $i\neq k$, to find the minimizing value of i and the corresponding minimum cost.

The solution is carried out by first setting $\boldsymbol{C}^{(0)}=\boldsymbol{T}$, that is, $c_{kj}^{(0)}=t_{kj}$, $k, j=1, 2, \cdots, N$. Using $\boldsymbol{C}^{(0)}$, $\boldsymbol{C}^{(1)}$ can be generated from (4).

The iterative process continues until $\boldsymbol{C}^{(l+1)}=\boldsymbol{C}^{(l)}$; this must occur in at most $(N-2)$ iterations because the optimal path can contain at most $(N-2)$ intermediate nodes. In addition to determining the minimum costs to move from point to point, the algorithm also generates information which is sufficient to determine the sequence of nodes on an optimal path [3].

To use the algorithm, a grid of points must be selected, and the costs of traveling directly between any two points must be determined. Factors which may influence the number and placement of grid points are:

1) the resolution of reconnaissance data which is available,
2) the observed topographical features of the terrain,
3) the range of vehicle sensors, and
4) the amount of computer storage available.

Computer storage availability should not be critical since it is envisioned that the computations will be performed on earth, and relevant information concerning optimal routes relayed to the vehicle through a command link.

Several possible selections for the elements of the cost function matrix $\boldsymbol{T}$ were considered:

1) two-dimensional Euclidean distance,
2) three-dimensional Euclidean distance,
3) elapsed time,
4) energy expended, and
5) various statistical estimates of the cost functions 1) through 4).

The solution of (4) is carried out in the same manner regardless of the particular choice of $\boldsymbol{T}$ (assuming, of course, that the properties of $\boldsymbol{T}$ given in (3) are satisfied).

As an example, reconsider the terrain configuration

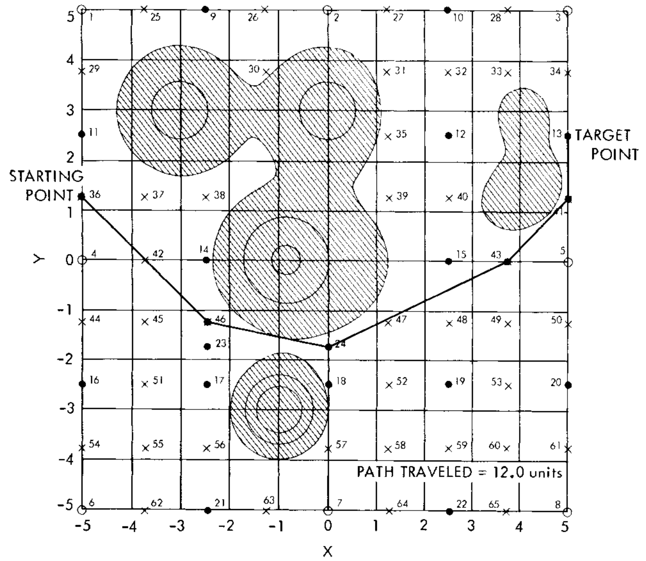


Fig. 3. Optimal path.

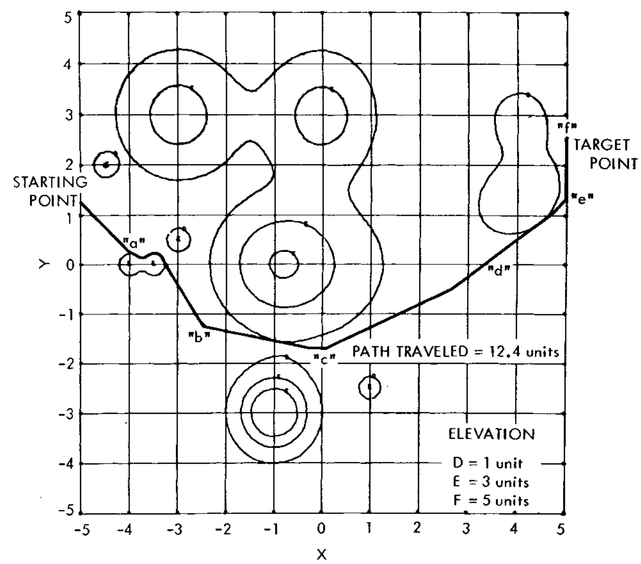


Fig. 4. Actual path.

TABLE I

Characteristic Features of the Local and Global Modes

	Local Mode	Global Mode
Merit	Requires no a priori terrain information to negotiate obstacles	Determines an optimal path from available terrain information
Limitation	May determine a devious and costly route when confronted by large obstacles	No capability of avoiding obstacles whose presence is not indicated in the terrain information

shown in Fig. 3. If it is desired to travel from point 36 (−5.0, 1.25) to point 13 (5.0, 2.5) via the path having minimum length in the X-Y plane, it is found that the nodes to be traversed are 36-46-24-43-41-13. The total distance traveled along this optimal path is 12.0 units, compared with the path length of 17.9 found previously using the local mode. The optimal path is shown in Fig. 3.

Dual-Mode Operation

The motivation for complementary use of these two algorithms is made apparent by Table I in which the characteristics of the two modes are summarized.

The global mode uses available terrain data to determine the sequence of nodes which defines an optimal path between two specified grid points. This optimal path is called the "nominal path" because obstacles which are not detected from the terrain data may require the roving vehicle to deviate from the optimal path. The vehicle follows the nominal path as long as no obstacles are detected by on-board sensors; when an obstacle is detected, an intermediate target point, which lies on the nominal path beyond the obstacle, is selected and the local mode is used to circumnavigate the obstacle. Upon arriving at the intermediate target point, the vehicle resumes its journey along the nominal path. By combining the local mode's capability of negotiating obstacles with the global mode's capability of determining optimal paths, an efficient and feasible routing procedure is obtained.

To illustrate the dual-mode operation, consider the terrain configuration shown in Fig. 4. Notice that the topography is identical with that of Fig. 3 except that an obstacle appears on the nominal path. This obstacle would not be accounted for in the global mode, because its presence would not be known. The vehicle proceeds along the nominal path until reaching point a where a "long wall" is encountered. An intermediate target point b is selected, and the local mode routes the vehicle around the wall. At point b the vehicle resumes its journey along the nominal path.

Notice that before reaching point d the vehicle departs from the nominal path and moves directly to point e. This is in response to an additional feature of the dual-mode algorithm in which the grid points on the nominal optimal path are regarded as secondary target points. Whenever a secondary target is within sensor range, and there is an unobstructed straight-line path from the current location to this secondary target, the vehicle follows this path. As shown in Fig. 4, this feature reduces the effect of grid coarseness.

Conclusion

A dual-mode routing strategy which utilizes available terrain information to determine a nominal optimal path, and negotiates obstacles using only on-board sensor information has been proposed and illustrated. Further investigation is required to ascertain the feasibility of using statistical measure of cost, perhaps in conjunction with pattern recognition techniques, to cause avoidance of terrain areas with a high incidence of obstacles. It also may be beneficial to consider making the global mode adaptive by using cost information derived from actual Martian explorations to revise the entries in the cost matrix $\boldsymbol{T}$.

References

[1] L. Y. Lim, "A pathfinding algorithm for a myopic robot," Jet Propulsion Lab., Pasadena, Calif., Tech. Rept. 32-1288, 1968.

[2] R. E. Bellman and R. Kalaba, *Dynamic Programming and Modern Control Theory*. New York: Academic Press, 1965, p. 50.

[3] D. E. Kirk, "A pathfinding algorithm for an autonomous roving vehicle," Jet Propulsion Lab., Pasadena, Calif., Tech. Rept. 32-1369, 1969.

Donald E. Kirk (S'59–M'62) was born in Baltimore, Md., on April 4, 1937. He received the B.S. degree in electrical engineering from Worcester Polytechnic Institute, Worcester, Mass., in 1959, the M.S. degree from the Naval Postgraduate School, Monterey, Calif., in 1961, and the Ph.D. degree from the University of Illinois, Urbana, in 1965.

He was an Instructor in Electrical Engineering at the Naval Postgraduate School from 1959 to 1962. While at the University of Illinois, from 1962 to 1965, he was an Instructor in Electrical Engineering, Collins Radio Instructor Fellow, and RCA Fellow in Electrical Engineering. He is currently an Associate Professor of Electrical Engineering at the Naval Postgraduate School. His research activity is in the area of optimal control theory, and he is the author of the forthcoming text *Optimal Control Theory: An Introduction*, to be published in 1970.

Dr. Kirk is a member of the American Society for Engineering Education, Sigma Xi, Tau Beta Pi, and Eta Kappa Nu.

Larry Y. Lim was born in Canton, China, on September 30, 1936. He received the B.S. degree in electrical engineering from the University of California, Berkeley, in 1960. He attended the University of Washington, Seattle, from 1961 to 1964.

He was with the Boeing Company, Seattle, Wash., from 1960 to 1964. He has been with the Jet Propulsion Laboratory, Pasadena, Calif., since 1964, where he is now a Senior Engineer in the Flight Computers and Sequencers Section of the Guidance and Control Division.

Mr. Lim is a member of the Association for Computing Machinery.