JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 22, No. 4, July–August 1999

Experiments on Space Robot Arm Path Planning Using the Sensors Database

Toshiaki Iwata,* Kenzo Kodama,* Fumio Numajiri,* and Hiroshi Murakami*
Electrotechnical Laboratory, Ibaraki 305-8568, Japan

Two sensor-based path-planning methods are proposed to control the orientation of the main body and the arm position of a space robot simultaneously using only arm motion. In this study, the database of relationships between the current sensor states, the robot motion, and the change of sensor states is required. A modified breadth-first search, or A* search, is then used to determine the arm path. It is possible to operate the robot and gather data simultaneously as well as to modify the database easily. The validity of the methods used is confirmed in experiments using a drop shaft, which provides 10 s of microgravity during free fall.

I. Introduction

THE orientation of a free-flying space robot is disturbed by the arm movement. This property is undesirable because it affects the communication links, vision systems, and the precision of the end effector manipulation. However, the space robot can simultaneously control the orientation of its main body and arm position using only arm motion because of the presence of the nonholonomic constraints.[1–7] Several methods have been proposed for this kind of control. Nakamura and Mukherjee presented a space robot control method using the Lyapunov function.[1] Yamada demonstrated a method using optimal control.[2] Some researchers found that cyclic arm motion is effective.[3–5] The use of the enhanced disturbance map (EDM)[6,7] is also effective to plan the path of the robot. However, these methods assume a precise robot model and the angular momentum conservation law. Some robot model parameters, such as the inertia tensor, are difficult to measure precisely, and atmospheric drag and/or gravity-gradient force violates the angular momentum conservation law. We propose two sensor-based path-planning methods that are not based on these assumptions. The requirement for these methods is reproducibility. Our methods require a database of the relationships among the current sensor states, the robot motion, and the change of sensor states during the motion. It is possible to operate the robot and gather data simultaneously as well as to modify the database easily. These path-planning methods could be used for other robots if the sensor output changes during robot movement.

There are several methods for achieving a three-dimensional microgravity field to perform space robot experiments on Earth: 1) using parabolic flight of an aircraft,[8] 2) using buoyancy in a water pool,[9] 3) using suspension[10,11] or special mechanisms such as a parallel link,[12] and 4) using a drop shaft.[13] In case 1, the microgravity environment is of low quality (10^{-2} g, $g = 9.8$ m/s^2), the experiment time is short (about 20 s) and the standard frame (fixed with respect to the aircraft) is not inertial. However, the experimenter can accompany the equipment so that interactive experimentation is possible. In case 2, there is viscous resistance and it is difficult to achieve a balance between gravity and buoyancy, but the experiment time can be long, which makes it suitable for demonstrating the efficiency of a method or a strategy of task sequence. Moreover, there is no limit on space or motion. In case 3, control of the wire tension or mechanism is complicated, and motion of the robot (in particular, free rotation) is limited. However, the experiment time is not restricted. In case 4, time and space are restricted, but the quality of the microgravity environment is good (10^{-4} g), observation in the inertial frame is feasible, and it is suitable for dynamic experiments.

We used a drop shaft to demonstrate the validity of path-planning methods based on the database of sensor states and robot motion in microgravity because the experiment requires a short time, and we can observe the robot motion in an inertial frame and the comparison with the simulation is easy. This experiment with a space robot using a drop shaft was the first trial of its kind in the world.

II. Path-Planning Methods

In this section, we propose two sensor-based path-planning methods, A and B. Method A is fundamental, whereas Method B is an optimized variation of Method A. We consider the space robot situation shown in Fig. 1. The x, y, and z axes are fixed on the robot body (L_0). The robot has one arm with a shoulder joint (q_1) and an elbow joint (q_2). Joint q_1 rotates $0 \le q_1 \le 360$ deg and joint q_2 moves $-90 \le q_2 \le +90$ deg. This is the simplest arm configuration that can cause three-dimensional orientation variation. The robot is equipped with a charge-coupled device (CCD) camera that is installed parallel to the x axis of the body. The CCD camera detects the light source, similarly to a star tracker. The light source is located in front of the robot and its position on the CCD image is indicated as (p_1, p_2). This expresses part of the main body orientation; the robot estimates its orientation based on this position. However, the light source position gives only the distance of the light source from the robot. The robot cannot estimate the rotation around the direction vector from the CCD camera to the light source. If the robot moves its arm, the body also moves, so that the image (p_1, p_2) changes. As a result, this space robot uses two input values (shoulder and elbow joint angle motions) and controls four degrees of freedom [two joint angles q_1, q_2 and light source position (p_1, p_2) on the image]. This is possible because the system considered is nonholonomic.

We use the discrete joint angles and motions in these methods. The joint angles are discrete every 30 deg; i.e., the shoulder joint is at $q_1 = 15, 45, 75, \ldots, 345$ deg and the elbow joint at $q_2 = -75$, -45, -15, 15, 45, and 75 deg. The joint motions of q_1 and q_2, which are indicated as ac_1 and ac_2, respectively, are limited to ± 30 deg/s for 1 s. There are eight-motion patterns expressed as (ac_1, ac_2) = (-30, -30), (0, -30), (30, -30), (-30, 0), (30, 0), (-30, 30), (0, 30), and (30, 30) deg/s. The light source position (p_1, p_2) is indicated by an ordinary pixel position, on a grid of 512×512, where the top-left position is (1, 1) and the bottom-right one is (512, 512).

We compile the database for path planning with computer simulation. The database includes the discrete joint angles (q_1, q_2), joint motions (ac_1, ac_2) and the change of the light source position on the CCD image (dp_1, dp_2) during the motion. Figure 2 shows a part of the database. In some joint angles, the joint cannot move in eight directions because the joint angle is beyond the permissible range. The total number of data entries was 472.

The process of searching for the path is shown in Fig. 3. We consider the discrete joint angles to be a node of a graph and the light source position on the CCD image (p_1, p_2) is treated as variables

Received 2 June 1997; presented as Paper 97-3556 at the AIAA Guidance, Navigation, and Control Conference, New Orleans, LA, 11–13 August 1997; revision received 12 September 1998; accepted for publication 14 December 1998.

*Senior Researcher, Frontier Technology Division, 1-1-4 Umezono, Tsukuba.

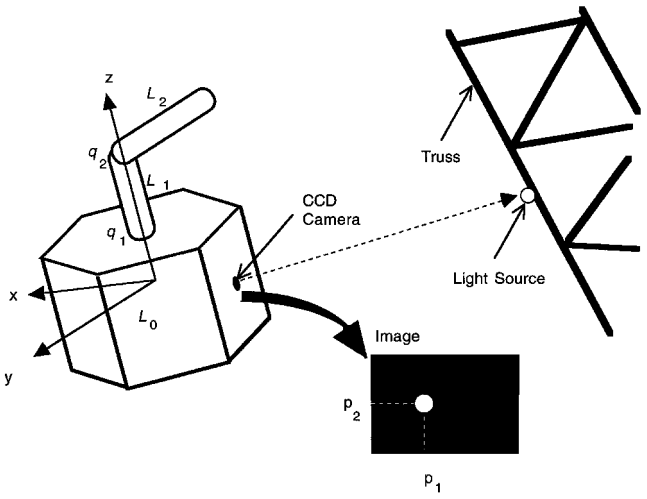


Fig. 1 Situation of space robot.

q_1	q_2	ac_1	ac_2	dp_1	dp_2
15.0	-75.0	0.0	30.0	-25	-3
15.0	-75.0	30.0	0.0	82	55
15.0	-75.0	30.0	30.0	32	41
15.0	-45.0	0.0	-30.0	24	3
15.0	-45.0	0.0	30.0	-29	-2
15.0	-45.0	30.0	-30.0	118	48
15.0	-45.0	30.0	0.0	65	31
15.0	-45		30.0		
15.0					
15.0					
15.0					

Fig. 2 Part of database.

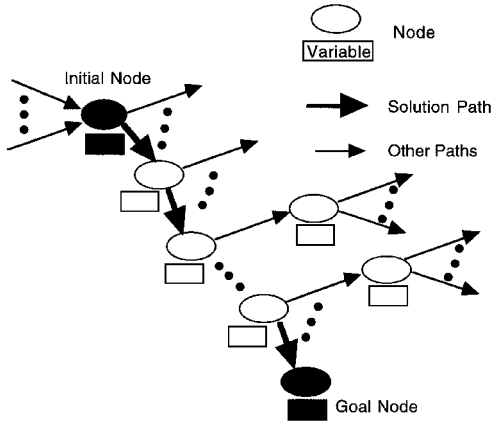


Fig. 3 Process of path planning.

of the node. The joint motion (ac_1, ac_2) and the change of the light source position (dp_1, dp_2) are included in the node transition. The initial light source position is given or measured; however, the light source positions of all other nodes are calculated using the data (dp_1, dp_2) when the path is searched. Part of the actual expressions of the nodes are shown in Fig. 4. Using the database, the robot searches for the path from the initial node to the goal node. We must distinguish the nodes of the same joint angles if the light source position is different. However, if there is a node that has the same variable as the node that has already expanded, we will not expand the path to that node to avoid wasting time (called pruning). Our aim is to find the path from the initial node to the goal node, and to achieve the aim, we use Methods A and B.

The robot model used in the computer simulation is shown in Fig. 5. This model, which has the same configuration as that in Fig. 1, was based on the robot used in the drop shaft experiment, as shown in the photograph in Fig. 5.

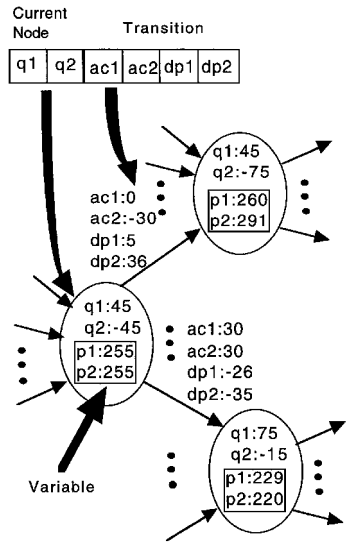


Fig. 4 Actual node expression.

Method A— Fundamental Method

Method A is a fundamental method and based on breadth-first search.[14,15] In this method, the initial node is expanded first, then all of the nodes generated by the initial node are expanded next, and then their successors, and so on. We need to change the expression of the states and the state transitions from a graph (Fig. 3) to a tree (Fig. 6). Breadth-first search will always find the shallowest (i.e., shortest) goal state. Because the system should deal with infinite states (i.e., node includes variables), we modify the conventional breadth-first search that is used for finite states. This means that pruning is a rare case because the variable of the generated node is seldom coincident with those of the nodes expanded so far, and numerous nodes must be searched for and evaluated, which consumes time and memory. We assume an example in which we seek a seven-motion path, each node has eight transitions and pruning does not occur. Then, the total number of searched nodes nt will be $8^6 = 262{,}144 < nt \leq 8^7 = 2{,}097{,}152$.

The goal node should be given as discrete joint angles and the permissible range of the light source position. If the range is narrow, the search might fail within the limit of time or node number.

We will present the path-planning process from the initial node (q_{1i}, q_{2i}, p_{1i}, p_{2i}) to the goal node (q_{1g}, q_{2g}, p_{1g}, and p_{2g} within the permissible range of error of ε pixels) according to modified breadth-first search as follows (see Fig. 6) :

Step 1. Find the initial node of (q_{1i}, q_{2i}) and give the light source position as (p_{1i}, p_{2i}).

Step 2. Expand the possible paths from the initial node using the data of the database (q_{1i}, q_{2i}, ac_{1i}, ac_{2i}, dp_{1i}, dp_{2i}); find the next nodes. The joint angles (q_{1n}, q_{2n}) and the light source position (p_{1n}, p_{2n}) of the generated node are calculated as $q_{jn} = q_{ji} + ac_{ji}$, $p_{jn} = p_{ji} + dp_{ji}$ ($j = 1, 2$) (see Fig. 4).

Step 3. Expand the paths from one of the generated nodes in Step 2 to find the consecutive nodes using the data of the database (q_{1n}, q_{2n}, ac_{1n}, ac_{2n}, dp_{1n}, dp_{2n}) and calculate the light source position similarly to Step 2 while checking pruning.

Step 4. Repeat Step 3 for all other nodes of this generation.

Step 5. Expand the next-generation nodes as in Steps 3 and 4 and check whether the node is the goal node (q_{1g}, q_{2g}, p_{1g}, and p_{2g} within the permissible range of error ε) or not. If the goal node is found, stop.

Step 6. Repeat Step 5.

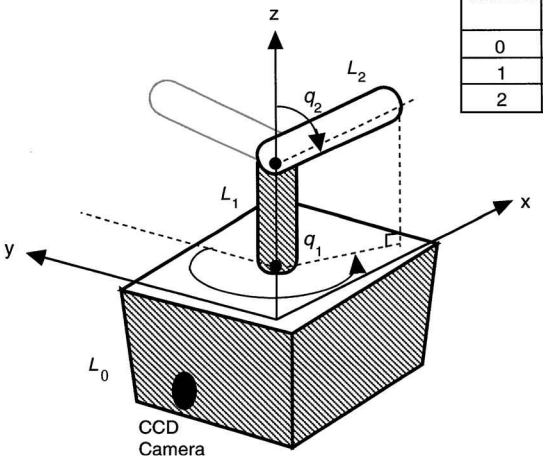
Modified breadth-first search yields the minimum time for robot motion to achieve the goal.

To summarize, when using Method A, 1) the search is time consuming and if the permissible error is small, the solution would not be found within the limited time or memory, and 2) the shortest path to the goal is obtained.

Method B—Using A* Search

To overcome the drawback of Method A (long search time), the concept of heuristics and cost was used in Method B. This type

Link No.	Dimensions (m)			Mass (kg)
	x	y	z	
0	0.19	0.19	0.14	3.25
1	0.04	0.04	0.10	0.45
2	0.04	0.04	0.10	0.47



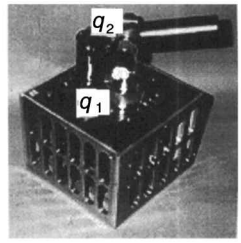


Fig. 5 Robot model.

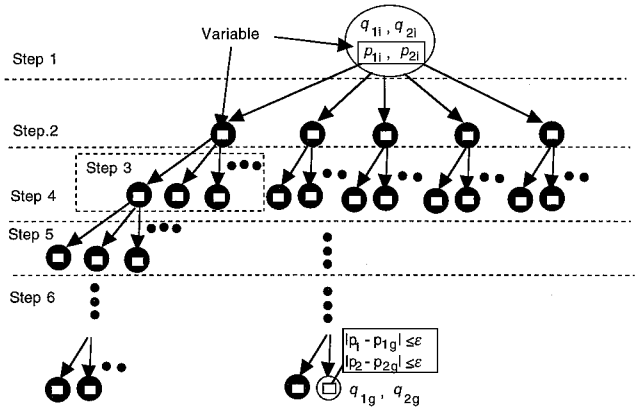


Fig. 6 Search process of Method A.

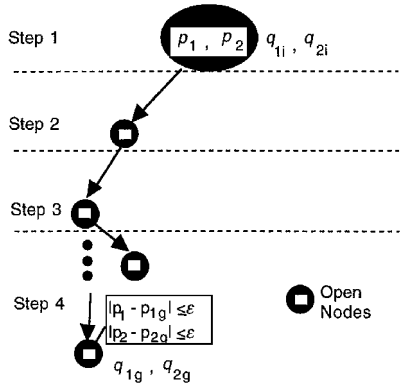


Fig. 7 Search process of Method B.

of search method is called A* search in the artificial intelligence field,[14,15] and can significantly shorten the search time if the path can be easily found. A* search is one of the optimal search patterns and combines the dynamic-programming principle. In this method, only one of the expansible paths expands to the next node in each step. To determine the expanding path, we use the heuristics and cost. In our case,

$$c = \sum_{\text{path = initial}}^{\text{current}+1} (|ac_1| + |ac_2|) \tag{1}$$

is determined as the cost, and

$$h = 1.0\left|q_1^{\text{goal}} - q_1^{\text{current}+1}\right| + 1.0\left|q_2^{\text{goal}} - q_2^{\text{current}+1}\right| + 0.05\left(\left|255 - p_1^{\text{current}+1}\right|^2 + \left|255 - p_2^{\text{current}+1}\right|^2\right) \tag{2}$$

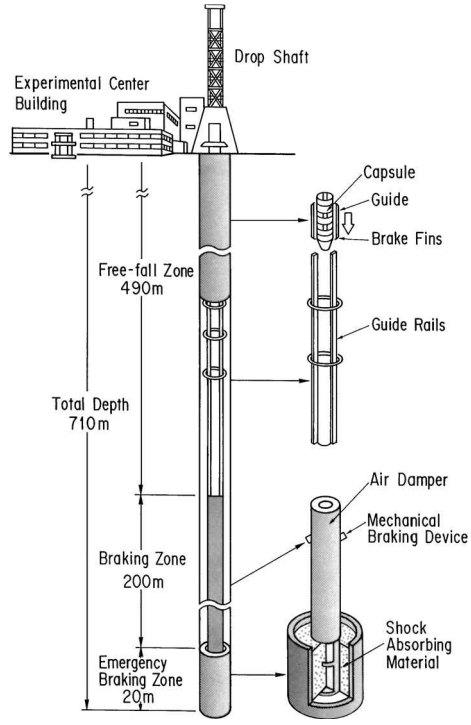


Fig. 8 Drop shaft.

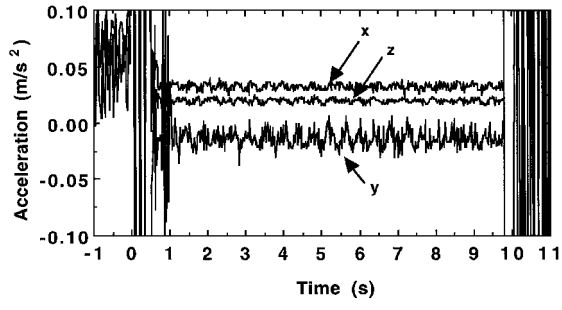


Fig. 9 Microgravity environment.

is determined as a heuristic function, where "current" indicates the open node that is being investigated ("open node" indicates the node that has been investigated and includes extensible paths), "current + 1" indicates the candidate node of the expanding path, and (255, 255) is the central pixel position of the image, which is given as the initial light source position and the goal light position. Then,

$$f = c + h \tag{3}$$

is the estimated cost. The next node selected is that at which the estimated cost reaches a minimum. In this case, the expanding path is determined by the cost of total joint motion to be as small as possible, the joint angles to be as close to the goal as possible and the light source position to be as motionless as possible. Therefore, the found path is not always the shortest as in breadth-first search. Since the estimated cost is subjective, what exactly is the optimal cost is a debatable point.

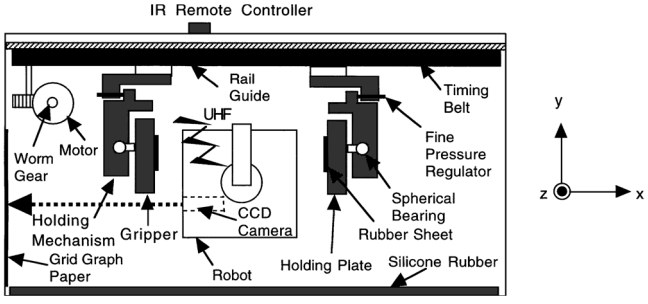


Fig. 10 Experiment rack.

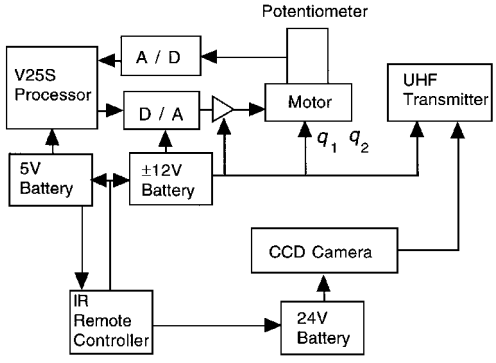


Fig. 11 Block diagram of robot for drop shaft experiments.

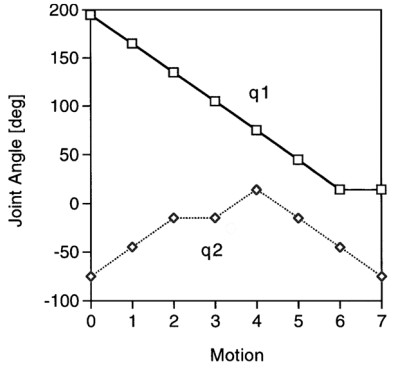


Fig. 12 Results of path planning with Method A.

The path-planning process from initial node $(q_{1i}, q_{2i}, p_{1i}, p_{2i})$ to goal node $(q_{1g}, q_{2g}, p_{1g}$, and p_{2g} within the permissible range of error of ε pixels) according to A* search can be expressed as follows (see Fig. 7) :

Step 1. Find the initial node of (q_{1i}, q_{2i}) and give the light source position as (p_{1i}, p_{2i}) and add this node to the open nodes set.

Step 2. Expand the path of the smallest estimated cost from the initial node to the corresponding node while checking pruning using the database and add it to the open nodes set. The calculation method for the joint angles and the light source position of the generated node is the same as that in Step 2 of Method A.

Step 3. Expand the path of the smallest estimated cost from the only node in the open nodes set; find the reached node while checking pruning. Add the node to the open nodes set. If all of the paths of the open node are searched, the node is deleted from the open nodes set. If the reached node is the goal node, the search is successfully concluded and stopped.

Step 4. Repeat Step 3.

To summarize, when using Method B, 1) the search time is short, and 2) the estimated cost is not given a priori.

These path-planning methods could be used for any other type of robot control if the sensor states change during robot motion. First, we will quantize the sensor states, and compile the database among the sensor states, motion, and change of the sensor states. Then, using either Method A or B, we will find the path from the initial state to the goal state.

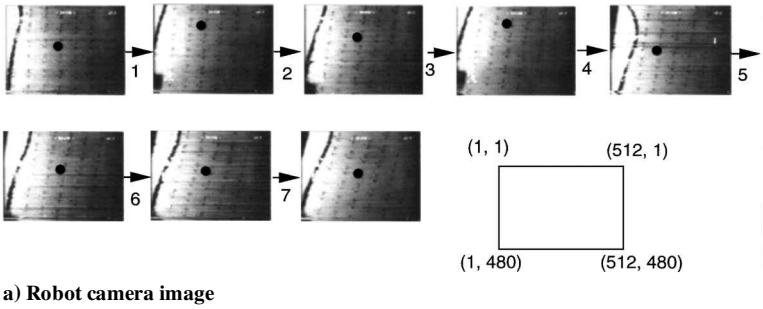
III. Experiments

Drop Shaft

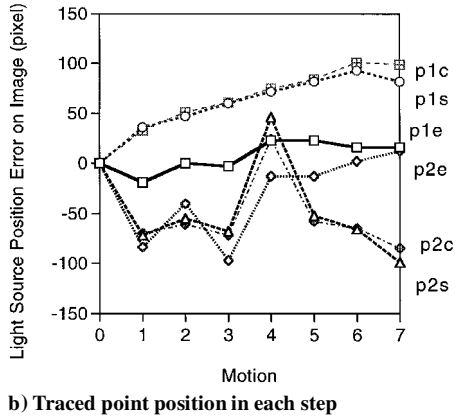
The drop shaft used in our experiment, formerly a coal mine, is 710-m deep, and the free-fall height is 490 m, which provides a 10-s period of microgravity, as shown in Fig. 8. This facility belongs to the Japan Microgravity Center.[16]

The typical microgravity condition during the experiment is shown in Fig. 9. In the figure, the x, y, and z directions are the same as those in Fig. 10 (i.e., the y direction indicates the direction of gravity). These directions are in agreement with the robot frame in the initial state. The condition data were not adjusted to absolute values, so that only the magnitude of change is meaningful and the offset is meaningless.

Figure 10 shows the rack used in our experiment, which is assembled in the inner capsule. The full size of the rack is 0.870 (width) × 0.870 (length) × 0.886 (height) m; however, users divide the space into quarters for different experiments. We used a half-rack space of 0.870 (width) × 0.870 (length) × 0.443 (height) m. The rack is equipped with CCD cameras to observe the robot motion in the inertial frame, a holding mechanism to hold the robot until the free-fall condition, an infrared remote controller to drive the robot, a power supply for the holding mechanism, a command controller to levitate the robot in microgravity, a video deck that receives UHF signals from the CCD on the robot body and silicone rubber (α gel) to absorb the collision shock. The levitation method is one of the most difficult techniques in drop shaft experiments. The grippers are opened for 0.5 s after achieving the free-fall condition, then closed



a) Robot camera image



b) Traced point position in each step

Fig. 13 Light source position error on image (Method A).

for 0.5 s, then opened again to realize stable levitation. In the original setup, we used a light source to estimate the orientation. However, if the robot loses the light source outside the visual field, we cannot evaluate anything. Therefore, we use a sheet of grid graph paper instead of the light source.

Figure 11 shows the block diagram of the robot (the specifications and photograph are shown in Fig. 3). The robot was controlled by a V25S CPU; the power was switched on via an infrared remote controller. The robot was equipped with 8-bit D/A and A/D converters to control the motors and measure the joint angles. A CCD camera was installed on the robot and the image was transmitted to the video deck on the UHF band. The viewing angles were 18 deg for the horizontal direction and 12 deg for the vertical direction. The number of pixels was 512×480.

Experiment for Method A

Although the amount of data required for the database is considerably small (=472), it is time-consuming to compile the data using the drop shaft. Therefore, the path-planning database was compiled by computer simulation. We searched for the path from the initial node ($q_1 = 195$, $q_2 = -75$, and $p_1 = p_2 = 255$) to the goal node ($q_1 = 15$, $q_2 = -75$, and $p_1 = p_2 = 255$ within a pixel error range of 100) using Method A . This indicates that only the shoulder joint rotates 180 deg, without a change in the camera direction and without movement of the elbow joint. Eventually, a seven-motion path was obtained after a 93,452-node search (5,179 s of CPU time using Sun Sparc 10; Fig. 12).

The change of the light source position on the image ($p_1 - c_x$, $p_2 - c_y$) (in experiment, $c_x = 277$ and $c_y = 223$, in simulation, $c_x = c_y = 255$) between the initial and the goal positions is shown in Fig. 13. In the figure, ($p1e$, $p2e$), ($p1s$, $p2s$), and ($p1c$, $p2c$) indicate the traced point positions on the grid graph paper (instead of the original light source position) of the experiment, the experimental simulation, and calculation from the database, respectively. The difference between the points of ($p1s$, $p2s$) and ($p1c$, $p2c$) is because of the following: 1) the database was obtained when the light source position was 0.9 m away from the front of the robot, but the experimental simulation involved a distance of 0.25 m; 2) ($p1c$, $p2c$) assumes that the light source position is centered in the image, but in the simulation, this assumption is violated (see Discussion). The difference between the points of ($p1e$, $p2e$) and ($p1s$, $p2s$) is because of the incompleteness of the model, the joint motion inaccuracy, and the initial condition of levitation (initial momentum and angular momentum). However, the global tendencies of the simulation are in agreement with the results of the experiment.

Experiment for Method B

The searched path is the same as that in Method A, that is, the path from the initial node ($q_1 = 195$, $q_2 = -75$, and $p_1 = p_2 = 255$) to the goal node ($q_1 = 15$, $q_2 = -75$, and $p_1 = p_2 = 255$ within a pixel error range of 100). Form 3 of the evaluation function is used. The search time is significantly shorter than that in Method A. An 11-motion path was found after a 282-node search (1 s of CPU time; Fig. 14). This result fits our intuitive estimation of rotating the main body as little as possible; first, minimize the arm inertia by straightening the arm ($q_2 \sim 0$), then rotating the arm (moves q_1), and then bending joint q_2. The change of the light source position ($p_1 - c_x$, $p_2 - c_y$) ($c_x = 228$ and $c_y = 226$ in the experiment, $c_x = c_y = 255$ in the simulation) on the image between the initial and the goal positions is shown in Fig. 15.

IV. Discussion

Change of Light Source Position

In compiling the database, the assumption that the change of the image (dp_1, dp_2) is the same in the whole image plane is debatable. If rotation takes place about the viewing direction or the relative position between the light and the camera changes during the motion, this assumption is not valid. In our case, the validity depends on the robot motion and the position of the center of the robot mass. Figure 16 shows the examples of the change of the image in the whole image. In the figure, the arrows indicate the motions of the light source positions (i.e., the initial light position and the final light position) during the motion. The initial light source positions on the CCD image are ($255 + 55i$, $255 + 55j$) (i, $j = -4, \ldots, 4$). The initial positions of joint angles are the same ($q_1 = 15$ and $q_2 = -75$) in Figs. 16a and 16b. However, the motions are different. In Fig. 16a, $ac_1 = 30$ and $ac_2 = 0$. The image change remains similar in the entire plane because the rotation of the main body is approximately around the z axis. In Fig. 16b, $ac_1 = 30$ and $ac_2 = 30$, however, the

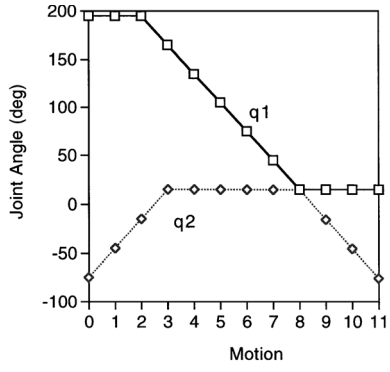


Fig. 14 Results of path planning with Method B.

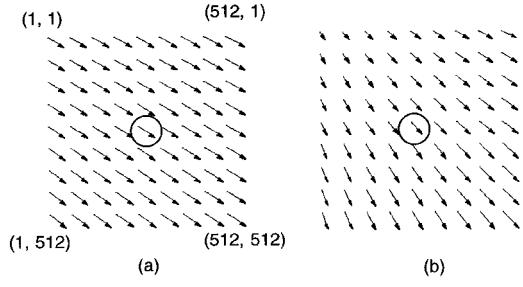
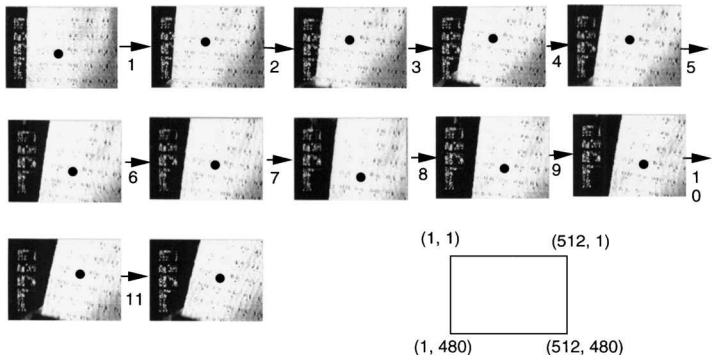
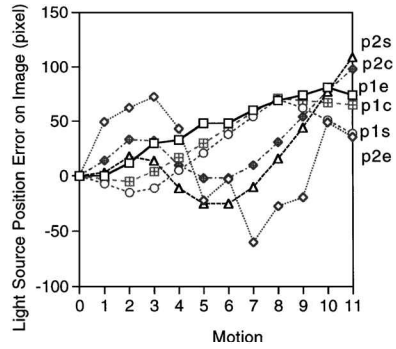


Fig. 16 Image distribution.

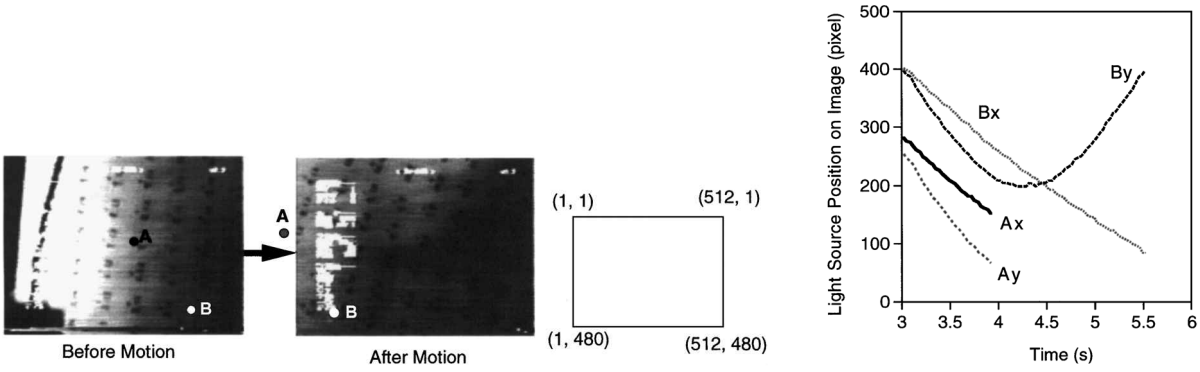


a) Robot camera image



b) Traced point position in each step

Fig. 15 Light source position error on image (Method B).



a) Robot camera image **b) Light source position during motion**

Fig. 17 Results of simplest path.

image change is distributed widely because the main body rotates around the x axis (camera installed direction). In both cases, the database was represented by the change in the motion at the center of the image (circled in the figure) as (dp_1, dp_2). However, to reduce the database size, we compiled the database assuming that the change of the image is the same in the whole image plane.

Generality

We discuss the generality of the proposed methods (i.e., we consider whether or not these methods can solve the path-planning problem between any nodes). We can find a path between any two joint-angle states; however, the light source position has many states. We introduced the permissible range of error in Sec. II. If the permissible range of error is small, it is difficult to find the path within a limited time period (in particular, by Method A). Therefore, if the search time is limited, we should make the robot report the paths that satisfy the final joint angles and their error for the light source position and select the best path.

Reference Experiment

The simplest path of joint motion from the initial state ($q_1 = 195$, $q_2 = -75$) to the final state ($q_1 = 15$, $q_2 = -75$) without path planning is one in which joint q_1 rotates uniformly, whereas joint q_2 does not move at all. We experimentally investigated the change of the grid graph paper image. The results are shown in Fig. 17. The robot started its motion 3 s after the microgravity state was achieved, and ended its motion at 5.5 s. Figure 16a shows the raw images of the initial (3 s) and final state (5.5 s). Figure 16b shows plots of the image motion from 3 s to 5.5 s. Because point A, which was located at the center [(258, 282) in pixel value] of the image at the initial state, disappeared during the motion, point B, which was observed throughout the motion, is also shown in the figure. If we assume that points A and B move parallel to each other in the image, the final position of point A on the image is $(-36, 261)$ in pixel value. We estimated 11 deg to be the difference between the initial and final directions, using the CCD images of the grid graph paper.

V. Conclusions

We proposed two methods (A and B) to plan a path to control the orientation and arm position of a space robot simultaneously, using only arm motion. The database of relationships among the current sensor states, the robot motion, and the change of states is required. Method A is based on breadth-first search. The system has infinite states, so that the search time becomes lengthy. To overcome this drawback of Method A, Method B, which uses A* search, was proposed. The search time becomes very short; however, the selection of the estimated cost is crucial for the search. We performed experiments with Methods A and B using a drop shaft that provides 10 s of microgravity during free fall. The results show that these path-planning methods are effective in controlling both orientation and the arm position.

Acknowledgment

This work is supported by the New Energy and Industrial Technology Development Organization through the Japan Space Utilization Promotion Center in the program of the Ministry of International Trade and Industry.

References

[1] Nakamura, Y., and Mukherjee, R., "Nonholonomic Path Planning of Space Robots Via a Bidirectional Approach," *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 4, 1991, pp. 500–514.

[2] Yamada, K., "Attitude Control of Space Robot by Arm Motion," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 5, 1994, pp. 1050–1054.

[3] Longman, R. W., "The Kinematics and Workspace of a Satellite-Mounted Robot," *Journal of the Astronautical Sciences*, Vol. 38, No. 4, 1990, pp. 423–440.

[4] Nakamura, Y., and Suzuki, T., "Planning Spiral Motions of Nonholonomic Free-Flying Space Robots," *Journal of Spacecraft and Rockets*, Vol. 34, No. 1, 1997, pp. 137–143.

[5] Yamada, K., and Yoshikawa, S., "Feedback Control of Space Robot Attitude by Cyclic Arm Motion," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 4, 1997, pp. 715–720.

[6] Okubo, H., Nagano, N., and Tsumura, T., "Path Planning for Space Manipulators Using Enhanced Disturbance Map," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA, Washington, DC, 1995, pp. 1510–1517 (AIAA Paper 98-3338).

[7] Torres, M. A., and Dubowsky, S., "Minimizing Spacecraft Attitude Disturbances in Space Manipulator System," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 4, 1992, pp. 1010–1017.

[8] Norsworthy, R. S., "Grasping Objects Autonomously in Simulated KC-135 Zero-G," *Proceedings of the AIAA Conference on Intelligent Robots for Factory, Field, Service, and Space*, AIAA, Washington, DC, 1994, pp. 757–762.

[9] Akin, D. L., and Spofford, J. R., "Redundancy Control of a Free-Flying Telerobot," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 3, 1990, pp. 515–523.

[10] Sato, Y., Iida, Y., Kanda, S., Maruyama, T., Uchiyama, T., and Fujii, H., "Micro-G Emulation System Using Constant-Tension Suspension for a Space Manipulator," *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Inst. of Electrical and Electronics Engineers, New York, 1991, pp. 1893–1990.

[11] Campbell, P. D., Swaim, P. L., and Thompson, C. J., "Charlotte Robot Technology for Space and Terrestrial Applications," 25th International Conference on Environmental Systems, Paper 951520, San Diego, CA, 1995.

[12] Shimoji, H., Inoue, M., Inaba, N., and Wakabayashi, Y., "Evaluation of Space Robot Behavior Using Berthing Dynamics Simulator," *Journal of Robot Society of Japan* (in Japanese), Vol. 13, No. 1, 1995, pp. 127–133.

[13] Iwata, T., Kodama, K., Numajiri, F., and Murakami, H., "Experiment on Robotic Motion Using Drop Shaft," *Strengthen Cooperation in the 21st Century* (6th ISCOPS), Vol. 91, Advances in the Astronautical Sciences, Univelt, Long Beach, CA, 1996, pp. 939–947.

[14] Winston, P. H., *Artificial Intelligence*, 2nd ed., Addison–Wesley, Reading, MA, 1984, Chap. 4, pp. 87–132.

[15] Russell, S., and Norvig, P., *Artificial Intelligence: A Modern Approach*, Prentice–Hall, Englewood Cliffs, NJ, 1995, Chaps. 3, 4.

[16] Kudo, I., "Microgravity Research in Japanese Industry," *Advances in Space Research*, Vol. 13, No. 7, 1993, pp. 13–21.