

## 〔研究〕

# センサ/動作データベースを用いた宇宙ロボットの腕・姿勢同時制御実験

## Simultaneous Control Experiment of Orientation and Arm Position of Space Robots Using Sensor-motion Database

岩田敏彰	村上寛	樹神謙三	沼尻文雄
T. IWATA	H. MURAKAMI	K. KODAMA	F. NUMAJIRI
鈴木俊幸	川田正國	戸田義継	町田和雄
T. SUZUKI	M. KAWADA	Y. TODA	K. MACHIDA

We propose four sensor-based path-planning methods to simultaneously control the orientation of the main body and the arm position of a space robot using only arm motion. In our study, the database of relationships among the current sensor states, the robot motion and the change of sensor states is required. We then use modified breadth-first search, A\* search, Genetic Algorithm or optimal control techniques to determine the arm path. It is possible to operate the robot and gather data simultaneously as well as to easily modify the database. We confirmed the validity of our methods in experiments using a drop shaft, which provides ten seconds of microgravity during free fall, and a water tank, which provides buoyancy.

### §1 緒言

宇宙ロボットは、無重力下で作業するため、運動が角運動量保存則に支配され、非ホロミックな拘束を受ける。このため、腕の最終形態とロボットの胴体の姿勢を腕の運動だけを用いて同時に制御できることが知られている<sup>1-5)</sup>。このような制御については数理的なモデルに基づいた方法が提案されており、山田は最適制御の考え方を用いて解決している<sup>1)</sup>。また、中村らは双方向にリアプノフの方法を用いて軌道を決定している<sup>2)</sup>。サイクリックな腕の運動が有効であるという報告もいくつかなされている<sup>3-5)</sup>。これらの方法は、厳密に角運動量保存則が成り立つ環境で、精密なモデル化が必要である。しかし、実際の宇宙環境では大気によるドラッグや重力傾斜などの外乱により、角運動量保存則は厳密には成り立たない。また、モデル化に必要なロボットのパラメータ

の中には実測が困難なもの（慣性テンソルの各軸の値など）も含まれている。そこで本稿では、角運動量保存則のような仮定やモデル化を必要としない方法を提案する。この方法は、腕の運動とそれによって引き起こされるロボット胴体に搭載された視覚センサ出力の変化をデータベース化し（以後、センサ/動作データベースと呼ぶ）、これをもとに腕の運動経路を決定するものである。この方法では、いろいろな初期状態から目標状態までの経路探索ができ、センサ/動作データベースそのものも実際に動作しながら順次訂正・追加することができる。このような方法は、宇宙ロボットに限らず、何らかの動作によってセンサの出力が変化するすべてのロボットに適用できる可能性がある。

無重力環境下での宇宙ロボット自身の運動や、対象物との相互作用を含めた運動を地上の施設を用いて三次元空間で観察・実験するには、(1) 航空機で

の弾道飛行による方法<sup>6)</sup>, (2) 水槽の中で水の浮力を用いる方法<sup>7-8)</sup>, (3) ロボットをつり下げたり<sup>9)</sup>, 平行リンクに取付けたりする方法<sup>10)</sup>, (4) 落下施設を用いる方法<sup>11-13)</sup>などが考えられる。宇宙ロボットの実験を地上で行う場合, これらの方法の長所・短所を把握し, それぞれの特性を活かした実験を行い, 複合的に評価することが望ましい。(1) は実験者が一緒に搭乗し, インタラクティブな実験を繰り返し行うことができるが, 無重力の質がパイロットの技量によるため,  $10^{-2}G$  ( $G$ は $9.8m/s^2$ ) と悪く, 一回の弾道飛行で得られる無重力時間も短い(約20秒)。また, 基準となる座標系(航空機の床)が飛行中に回転するため慣性系でなく補正が必要となるが, 慣性座標に依存しない対象物との相互関係(捕捉・衝突など)の実験に向く。(2) は空間的, 時間的制限は原理的には発生しないが, 水の粘性抵抗があり, 浮力と重力を釣り合わせ, 浮心と重心を各リンクについて一致させることが必要となる。力学は無重力と異なるため, 力学を含まない航法や作業の戦略を研究するのに向く。(3) は時間的な制限はないが, つり下げるワイヤの張力や平行リンクの制御が複雑であり, 回転運動の模擬は困難となる。またワイヤやリンク自身がロボットの運動を制限することがある。この方法は衛星の捕捉など, 対象物との相互関係を精密に実験するのに向く。(4) は無重力の質は良いが( $10^{-4}G$ ), 実験空間や無重力時間の制約がある。慣性座標系での観察が可能となるため, ロボット単体や対象物との相互作用のある系の慣性座標を基準系とする力学的な実験に向く。

筆者らは, 提案した方法を落下施設と水槽内で水の浮力を利用して実験を行った。落下実験の場合は実験時間が短いため, センサ/動作データベースを構築する過程は計算機によって行ったが, 水槽を用いた実験ではセンサ/動作データベースの構築から実際の動作まで実験した。これは落下実験では力学を反映した実験ができるため, シミュレーションによるセンサ/動作データベース構築が可能であるが, 水槽を用いた実験では力学をうまく計算機上で表現できないため, シミュレーションができないという事情を反映している。また, 水槽を用いた実験では浮力や粘性力による外乱が発生するため, 角運動量が保存されないような場合にも提案する方法が

有効であることを示すことができる。

本稿では, まず, 2章でモデルによらない腕と姿勢の経路計画法を4つ提案する。これらはいずれも同じセンサ/動作データベースを用いる。3章で落下施設を用いた実験方法と実験結果を説明する。4章で水槽による実験方法と実験結果について説明する。5章でそれぞれの実験に関する考察と提案した4つの方法の比較を行う。最後に6章で結論をまとめる。

## §2 経路計画法

### 2.1 センサ/動作データベース

Fig.1のような状況にある宇宙ロボットを考える。このロボットは2関節であり, 視覚センサとしてCCDカメラを備えている。関節角を $30^\circ$ 毎に離散化し, 肩関節 $q_1 = 15^\circ, 45^\circ, \dots, 195^\circ$ , 肘関節 $q_2 = -75^\circ, -45^\circ, \dots, 75^\circ$ をとりうる値とする。視覚センサ上の光源の位置( $p_{1s}, p_{2s}$ )は連続値(正確には $512 \times 512$ ピクセルで離散値であるが, 連続値にも適用できる)とする。関節の運動は肩, 肘について( $ac_1, ac_2$ )で表され, 角速度 $\pm 30^\circ/s$ または $0^\circ/s$ で1秒間動作するとする。関節を動作させるとロボット本体も動き, その結果視覚センサ上の光源位置も変化する。これらをまとめてデータベースを構築する。

Fig.2 に具体的なデータベースのデータ構造を示す。データベースは, 関節角の位置( $q_1, q_2$ ), 関節の動かし方( $ac_1, ac_2$ ), そのときの光源位置の変化( $dp_{1s}, dp_{2s}$ )で構成される。

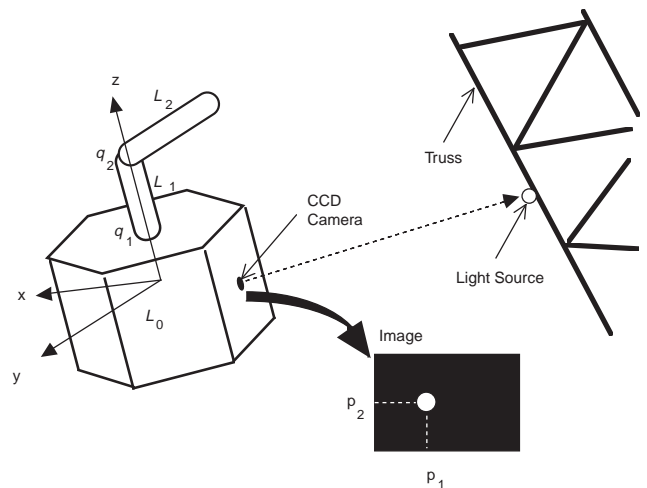


Fig.1 Situation of space robot.

以下に述べる経路計画では、2.2~2.4 節の方法では Fig.3 のように節点による状態遷移図を考え、ロボットの初期状態から目標状態までのグラフの経路を探索することになる。図中、太い矢印は求める経路の1つを表す。このグラフで、Fig. 2のデータベースとの関係は Fig.4 のように与えられる。節点は関節角 ( $q_1, q_2$ ) を代表し、視覚センサ上の光源の位置そのものは節点に付随したラベルとした。節点間の遷移は、関節の運動 ( $ac_1, ac_2$ ) と視覚センサ上の光源の位置の変化量 ( $dp_1, dp_2$ ) で与えられる。この方法では節点に付随したラベルを導入することにより、連続量（ここでは画像のピクセル位置）が扱えるようになる。目標値は節点（関節角）だけで与えることができず、ラ

$q_1$	$q_2$	$ac_1$	$ac_2$	$dp_1$	$dp_2$
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	34
15.0	-45.0	0.0	30.0		
15.0					
15.0					
15.0					
15.0					
15.0					
15.0					

Fig.2 Part of database.

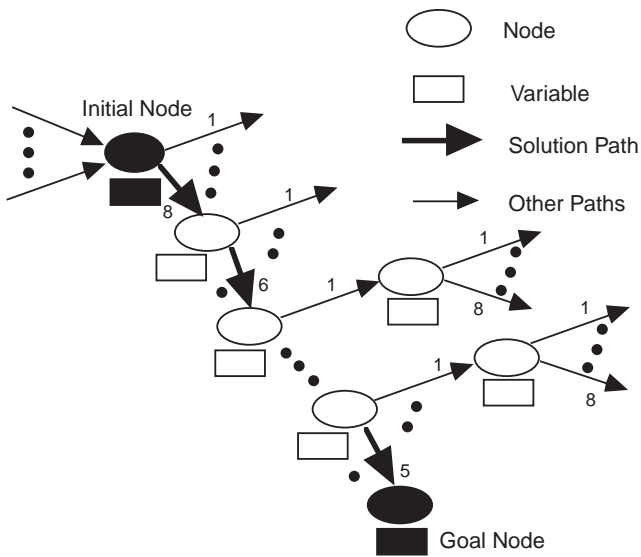


Fig.3 Process of path planning.

ベルの許容範囲を指定しなければならない。節点探索は、ラベルを考慮したものにならない。

これらの方法では、目標値は離散的な角度で目標節点としてしか与えられず、実際に用いるには大きな制約となる。このため、Fig.2に示されたデータベースを式で近似し、最適制御の手法を用いて経路を求めることにした。これが2.5節の方法である。

## 2.2 幅優先探索<sup>14)</sup>

ロボットの初期状態から順次動作を展開してFig.5のように探索木を作り、木の浅い節点を先に調べ

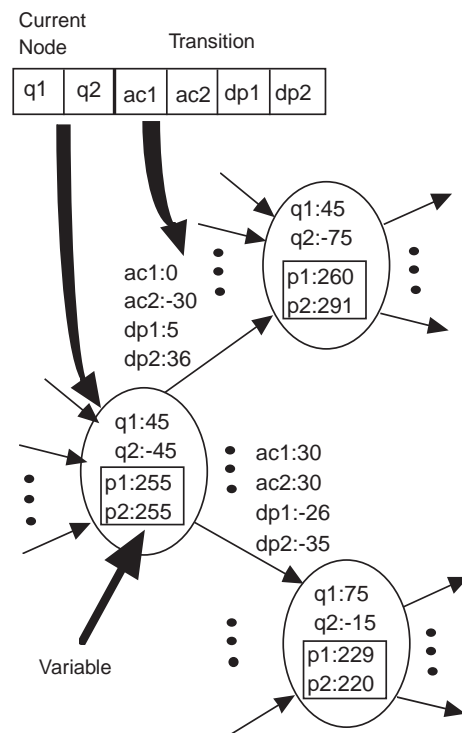


Fig.4 Actual node expression.

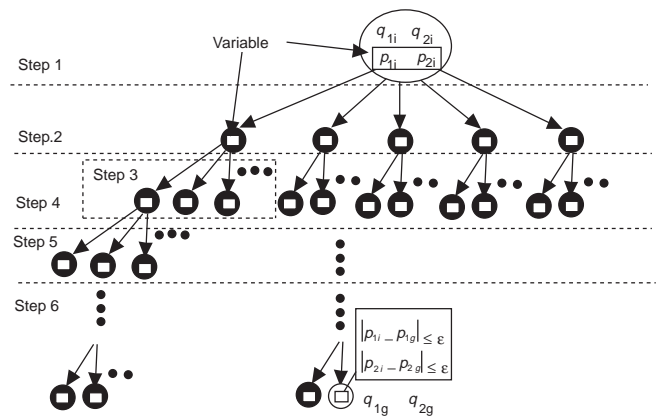


Fig.5 Search process of breadth-first search.

る。図中、Stepは探索順序を表す。この方法では可能な動作と節点をすべて調べることになり、効率的でないが、見つかった経路は最短動作であることが保証されている。これを幅優先探索という。従って、Fig.5 に示されているような順序で探索することになる。実際の探索では目標節点のラベルにある許容範囲を設けないと現実的な時間・計算資源の範囲内で解を得られないので、そのような範囲を設けた。具体的には以下のようなステップで探索を行う。

**ステップ1** 関節角 ( $q_{1i}, q_{2i}$ ) を表す節点  $i$  を見つけ、光源の位置 ( $p_1, p_2$ ) をその節点のラベルに入れる。

**ステップ2** ( $q_{1i}, q_{2i}$ ) から移動可能な関節角をセンサ/動作データベースデータ ( $q_{1i}, q_{2i}, ac_{1i}, ac_{2i}, dp_{1i}, dp_{2i}$ ) を使って全て求め、それぞれを新しい節点  $n$  とする。このとき、各節点について関節角 ( $q_{1n}, q_{2n}$ ) とラベルの値である光源の位置 ( $p_{1n}, p_{2n}$ ) は  $q_{jn} = q_{ji} + ac_{ji}, p_{jn} = p_{ji} + dp_{ji}$  ( $j = 1, 2$ ) で計算する。関節角、光源の位置がともに目標節点の条件を満たすものが見つければ終了する。見つからなければステップ3に進む。

**ステップ3** ステップ2で展開された節点の1つについて、ステップ2と同様に節点を展開する。このとき、すでに展開した節点と同じ関節角、ラベル値を持つものがないか調べ、同じものが見つければその節点は切り捨てる(枝刈りという)。目標節点の条件を満たすものが見つければ終了する。見つからなければステップ4に進む。

**ステップ4** ステップ2で展開された他の節点についてもステップ3の操作を繰り返す。目標節点の条件を満たすものが見つければ終了する。見つからなければステップ5に進む。

**ステップ5** ステップ3と4で展開された節点について、順次展開し、目標節点の条件を満たすものが見つければ終了する。見つからなければステップ6に進む。

**ステップ6** 前のステップで展開された節点について同様に繰り返す。目標節点の条件を満たすものが見つければ終了する。見つからなければステップ6を繰り返す。

### 2.3 A\*アルゴリズム<sup>15)</sup>

上に述べた幅優先探索では探索に時間がかかる上、コンピュータのメモリも膨大に必要となる。これらの問題点を解決するため、A\*アルゴリズムを用いた方法を試みた。A\*アルゴリズムというのは、人工知能の分野で効率的な探索を行うために用いられる方法で、コストとヒューリスティックという知識を用いて行うものである。これは動的計画法(ダイナミック・プログラミング)的なアプローチである。現在展開可能な節点(開節点という)のうちで、その節点に至るまでのコスト

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

と、その節点と目標とのずれ(ヒューリスティック)

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

を考え、これらの和  $f = c + h$  を評価関数とした。A\*アルゴリズムではこの評価関数のもっとも小さい経路だけを開く(Fig.6)。この場合も目標節点に許容範囲を設定した。具体的には以下のようなステップで探索を行う。

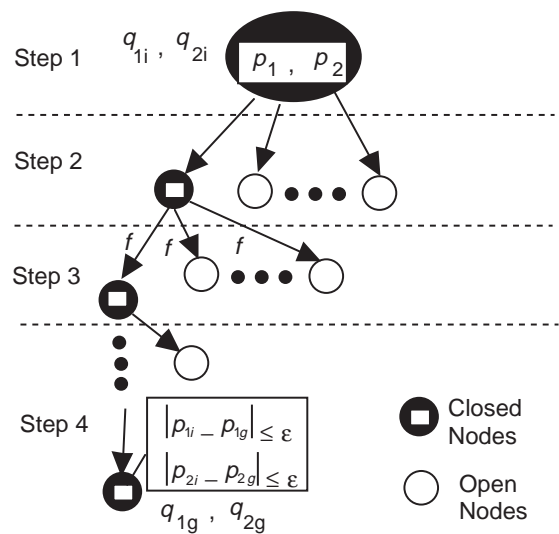


Fig.6 Search process of A\* algorithm.

**ステップ1** 関節角 ( $q_{1i}, q_{2i}$ ) を表す節点  $i$  を見つけ、光源の位置 ( $p_1, p_2$ ) をその節点のラベルに入れる。 $i$  を開節点の集合に加える。

**ステップ2** 開集合の節点から評価関数  $f$  が最小になるもの  $p$  を選び、 $p$  を閉集合に入れる。 $p$  が目標節点の条件を満たせば終了。

**ステップ3**  $p$  から展開可能な子節点全てについて評価関数  $f$  を計算する。子節点が開節点の集合にも閉節点の集合にも含まれていなければ開節点の集合に加える。子節点が開節点の集合に含まれていて、 $f$  の新値が小さければ  $f$  を更新する。子節点が開節点の集合に含まれていて、 $f$  の新値が小さければその節点を開節点の集合に移す。

**ステップ4** ステップ2以下を繰り返す。

## 2.4 遺伝的アルゴリズム (GA)<sup>16)</sup>

Fig. 3で、矢印に番号を付け、各節点毎に上から順に0, 1, 2, ... とすると、これを動作順に並べることによりロボットの運動が記述できる。そこで、これを遺伝子と考えることにする。Fig. 7 (a) にこの考え方を模式的に示す。また、Fig. 3 の太い矢印による経路を遺伝子で表すと(b) のようになる。遺伝子の長さの制限は原理的にはないが、実際の応用を考え、(1)

目標関節角になり、かつラベルが許容範囲になった場合はそれ以後の遺伝子は考えない、(2) 最大長さは30とする、という条件をつける。遺伝子の良さを表す適合度としては、動作数が少ないほどよい(遺伝子が短いほどよい)、遺伝子の終端では目標状態になっているのがよい、という2つの条件を含ませ、評価値  $fitness$  を次のように定義する。

$$fitness = \frac{1}{length} + g \quad (3)$$

ここで、 $length$  は遺伝子の長さ、 $g$  は最終状態の有無を表す係数で、 $g=0.3$  (最終状態を持つ場合)、 $g=0$  (最終状態を持たない場合) とした。

交叉については、2つの親から1つの子を作ることとした。ルーレット法で2つの遺伝子を選び出し、同じ状態(関節角とラベル)をもつ部分を選び出し、短い部分を選んでつなぎ合わせるといった方法をとった。また、同じ状態がない場合、親1の遺伝子をそのまま子の遺伝子にした(Fig. 8)。突然変異は終端が目標状態になっている遺伝子には起こりにくくした。また、突然変異はそれ以後のすべての状態に影響を与えるため、その遺伝子から以後すべてに起こるものとした(Fig. 9)。

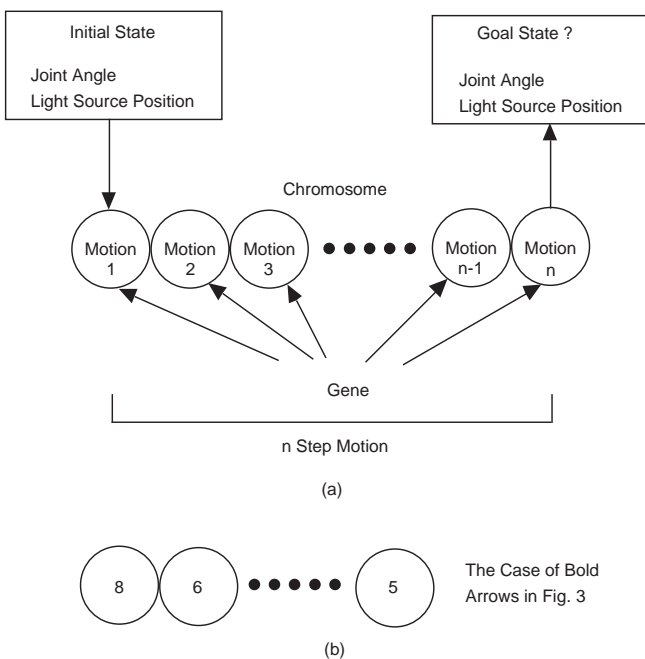


Fig. 7 Composition of chromosomes.

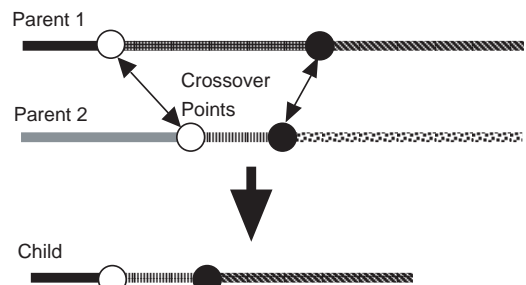


Fig. 8 Crossover.

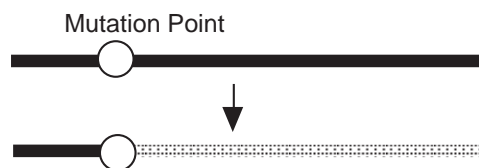


Fig. 9 Mutation.

遺伝子の個体数，許容される遺伝子の長さ（最大動作数），突然変異の確率などがパラメータとなる。また，目標節点のラベルに許容範囲を設けた。

### 2.5 最適制御の手法を用いる方法

データベースの近似式は，動作範囲の端にない一般の関節角では関節の動かし方が $(ac_1, ac_2) = (-30^\circ, -30^\circ), (0^\circ, -30^\circ), (30^\circ, -30^\circ), (-30^\circ, 0^\circ), (30^\circ, 0^\circ), (-30^\circ, 30^\circ), (0^\circ, 30^\circ), (30^\circ, 30^\circ)$  /s の 8 通りあることより，まず，各関節角において

$$f_i = a_{1i}u_1 + a_{2i}u_2 + a_{3i}u_1^2 + a_{4i}u_1u_2 + a_{5i}u_2^2 + a_{6i}u_1^2u_2 + a_{7i}u_1u_2^2 + a_{8i}u_1^2u_2^2 \quad (4)$$

とおき，16個の未知数  $a_{ij}(i=1,2, j=1,\dots,8)$  を求める。ここで，慣例に従い，入力  $u_1 = ac_1$ ， $u_2 = ac_2$ ，状態量は  $f_i = dp_i$  と表した。さらに求めた  $a_{ij}$  に対して，

$$a_{ij} = b_{1ij} + b_{2ij} \sin q_1 + b_{3ij} \sin q_2 + b_{4ij} \cos q_1 + b_{5ij} \cos q_2 + b_{6ij} \sin q_1 \sin q_2 + b_{7ij} \sin q_1 \cos q_2 + b_{8ij} \sin q_1 \cos q_2 + b_{9ij} \sin q_1 \cos q_2 \quad (5)$$

とおいて，最小二乗法で  $b_{kij}(k=1,\dots,9, i=1,2, j=1,\dots,8)$  を求めた。これらの式を用い，関節角の動作範囲を考慮して，次のような評価関数を最小にするように経路を決定した。

$$L = \int_0^{10} (u_1^2 + u_2^2) dt + f_p,$$

$$\text{ただし } f_p = \begin{cases} 0 & (x_2 \geq x_{2c}) \\ w \left( \frac{x_2 - x_{2c}}{x_{2m} - x_{2c}} \right)^2 & (x_2 < x_{2c}) \end{cases} \quad (6)$$

ここで  $w = 1$ ， $x_2 = q_2$ ， $x_{2c} = -75^\circ$ ， $x_{2m} = -90^\circ$  とした。ここで，角速度の 2 次式と慣性テンソルの積は運動エネルギーを表すので，この評価関数は関節モータに加えられるエネルギーに関係した量ということになる。

経路の計算には，Sequential Conjugate Gradient Restoration Algorithm (SCGRA)<sup>17)</sup>を誤差の大きい初期の計算に，Modified Quasi-linearization Algorithm (MQA)<sup>18)</sup>を誤差の小さい後半の計算で用いた。計算の過程を Fig.10 に示す。

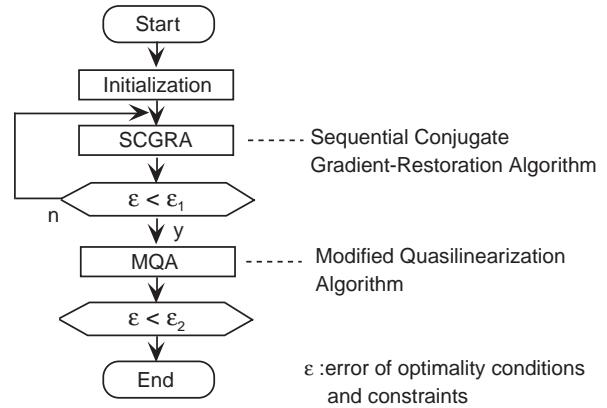


Fig.10 Solution process of optimal control.

## §3 落下実験

### 3.1 装置

#### (1) 地下無重力実験センター

落下施設は閉山した炭鉱を利用したもので，深さ 710m の立坑中（自由落下距離 490m）をカプセルが落下するものである。落下施設とカプセルを Fig.11 に示す。カプセルは二重構造になっており，外カプセルと内カプセルで構成されている。これらの間は真空（100Pa）で，外カプセルは内カプセルに接触しないようにスラストによって空気抵抗を補償しながら加速される。実験装置は内カプセル内の実験ラックに置かれる。落下施設の基本性能としては，微小重力レベルが  $10^{-5}G$  以下，微小重力時間が 10 秒，制動時の加速度が 10G 以下というものである<sup>19)</sup>。制御モニター室では実験ラックに搭載された CCD カメラの画像の受信，制御コマンドの送信，データ等の表示を行うことができる。

#### (2) 実験ラック

実験ラックの搭載可能な大きさは 0.87m（奥行き）× 0.87m（幅）× 0.886m（高さ）であるが，4 分割が可能である。今回の実験に用いた実験ラックの大きさは 0.87m（奥行き）× 0.87m（幅）× 0.443m（高さ）（2 分の 1 ラック）で，この中に電源を含めたすべての実験機器を収めなければならない。Fig.12 にラックを示す。(a) はその写真を，(b) は内部の概観を示す。図中にラックに対する落下方向も示す。微小重力終了後，カプセルの制動のためロボットが下面に衝突するので，その衝撃を緩和するため，シリコン樹脂

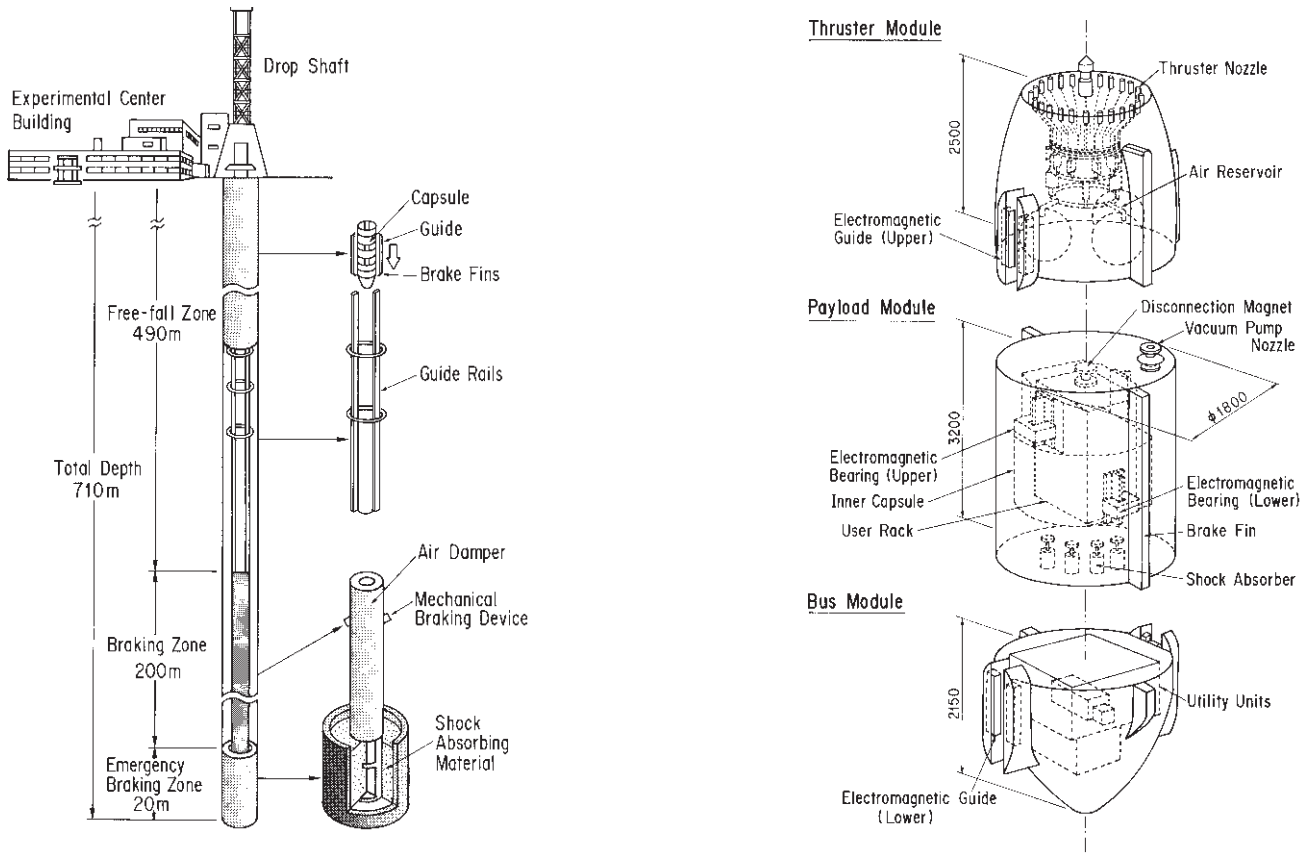
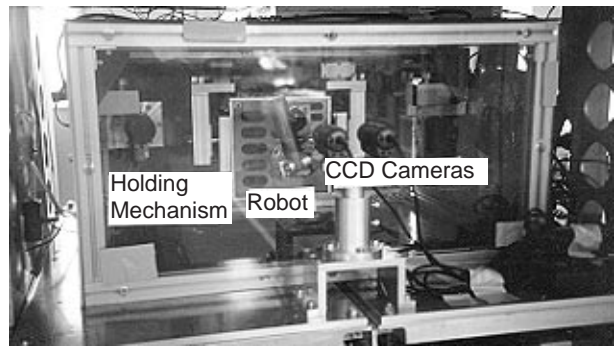
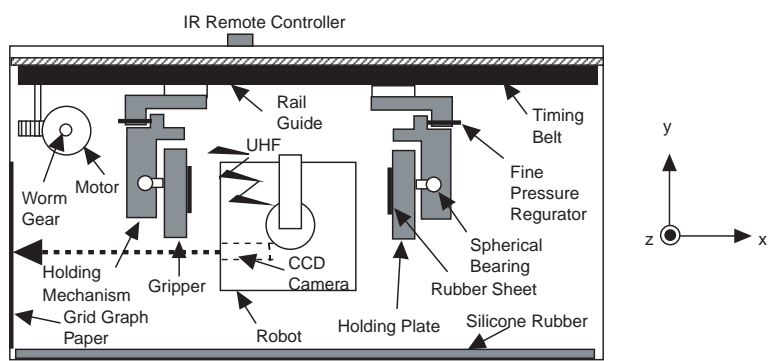


Fig.11 Drop shaft.



(a)



(b)

Fig.12 Experimental rack. (a) Photograph. (b) Schematic.

(商品名：アルファゲル)を用いた。ロボットはラックに取付けたCCDカメラにより慣性系での動きを観察され、画像は制御モニタ室に光伝送される。ロボットを無重力中で切り離す機構として平行2指の把持機構を用いた。切り離し法については次章で詳しく述べる。

### (3) 3リンクロボット

実験に用いたロボットを Fig.13 に示す。(a) は自由度配置・ロボットの座標系と仕様の概要(後述するシミュレーションに用いた値)，(b) はロボットのブロック線図である。図中，肩関節 ( $q_1$ ) はy 軸と肘関節 ( $q_2$ ) の回転面のなす角である。このロボットは全て電池で駆動され，赤外線によって電源がオン・オフされる。ロボットには V25S の CPU が搭載されており，電源のオンと同時に RAM 上の C 言語で開発されたプログラムが起動される。このプログラムはホストコンピュータ (PC9801ns/A) で開発され，ロボット側の RAM にダウンロードされる。ロボットに搭載されたセンサは各関節角を検出するポテンシオメータ 2 個と CCDカメラ (視野の全角度は水平方向 $18^\circ$ ，垂直方向 $12^\circ$ ， $512 \times 480$ ピクセル)である。ポテンシオメータの電圧値は AD 変換器で CPU に取り込まれ，実験終了後ホストコンピュータに伝送され，所望の動作をしたか確認がなされる。また，CCDカメラの画像は UHF 電波でラックに搭載されたビデオデッキに受信・録画されると同時に制御モニタ室にも光伝送される。搭載された CCDカメラで見える撮像方向は Fig.13 (a) のロボットのy方向が左に，z方向が下である。ロボットが姿勢変化を起こしたときに搭載された CCDカメラの画像で運動が評価できるようにラック壁面に1cm毎の格子に点を描いた方眼紙を

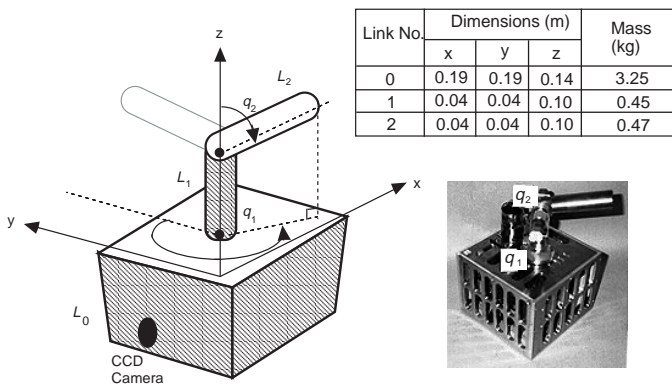


Fig.13 Robot model for drop shaft experiments.

張り付けた ( Fig.12 (b) ) 。

### 3.2 ロボットの切り離し実験

ロボットの無重力実験は，無重力中で運動量を与えずにロボットを浮遊させることが重要である。このための切り離し法はいくつか考えられるが，失敗するとロボットに外乱を与えることになり，(1) 実験中にロボットが装置の壁面や天井に衝突する，(2) ロボットの制御則を確認できない，(3) 実験の準備段階で初期条件がかわってしまう，などの問題が起こる。そこで，ロボットの切り離し法として以下の 6 種類の方法を試みた ( Fig.14 ) 。

#### 3.2.1 切り離し法

##### (1) 1本の糸でつるす方法

Fig.14 (1) に示す方法はもっとも単純であり，実施が容易であるが，糸のまわりの回転のため初期姿勢

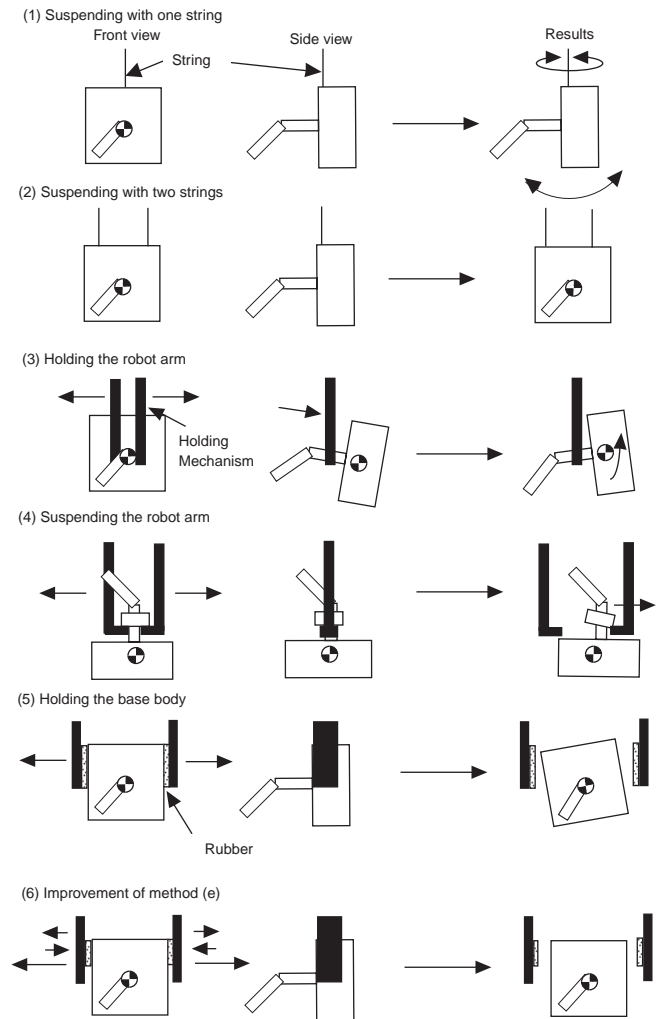


Fig.14 Robot detached methods.

が制御できない。また、落下直前まで行われる落下カプセルの保持動作による位置制御に伴う振動の影響を受け、振動する。この振動が初期の運動量を与える。

### (2) 2本の糸でつす方法

Fig.14(2)に示す方法は(1)について単純であり、実施が容易である。2本の糸で回転と振動は拘束することができ、初期運動は生じなかった。しかし、この方法では無重力になったあと、ロボットを自由に運動させるために外乱を与えることなく糸を切断しなければならない。現状では外乱を与えない切断方法が提案できないため、実現できていない。

### (3) ロボットの腕をはさむ方法

Fig.14(3)に示す方法はロボットの腕を切り離した後、ロボットの運動空間が大きく取れるということが期待される。しかし実際にこの方法を用いると、ロボットをセットするのが1Gの重力下であるため、ロボットの重心と把持位置のずれによる応力が把持部に生じ、無重力になった瞬間にこれが解放されて余分な運動量が発生する。

### (4) ロボットの腕をつす方法

Fig.14(4)に示す方法も切り離した後、ロボットの運動空間が大きく取れるということが期待される。しかし、この方法ではセットしたときに把持部と腕の間に凝着を生じ、把持機構の一方と切り離れずに図のような運動してしまう。

### (5) ロボットの胴体をはさむ方法

Fig.14(5)に示す方法は、ロボットの運動空間が狭くなるが、(3)の欠点であったロボットの重心と把持位置のずれを解消するものである。それでも把持部にわずかに残る重力による応力が微小な運動量を発生させ、実験の再現性の点で問題が残る。

### (6) (5)を改良した方法

Fig.14(6)に示す方法は(5)の方法で把持し、無重力になると一度わずかに開いた後、再度把持機構を閉じて応力のない状態で把持し直し、それから改めて切り離すものである。この方法ではほとんど運動

量は発生しなかった。以下に述べるロボットの落下施設での運動実験はこの方法によって切り離したものである。この方法による切り離しについて、次にさらに詳しく述べる。

### 3.2.2 方法(6)の評価

(6)の切り離し法についてロボットに搭載したCCDカメラで評価を行った。ロボットは次章以降で述べる実験の運動を行ったあと、落下終了まで数秒間静止した状態になる。この間のロボットの運動を調べることにより、ロボットが切り離しのときにもった運動量を推定することができる。ロボットの動作終了直後の計測開始時間を0秒とし、その後3.73秒間ロボットに搭載されたCCDカメラからラック壁面に張り付けた方眼紙を観察した。計測時間0秒と3.73秒のCCD画像をFig.15に示す。この2つの画像だけではほとんど変化は認められない。図中のA点とB点の画像上での時間変化をFig.16に示す。このデータは画角で示してあるが、3.73秒の間に0.5°程度の変化しか認められない。

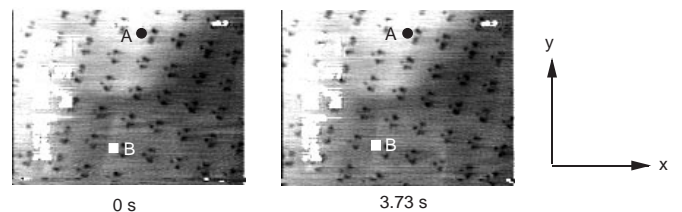


Fig.15 CCD images for evaluation of detached method (6).

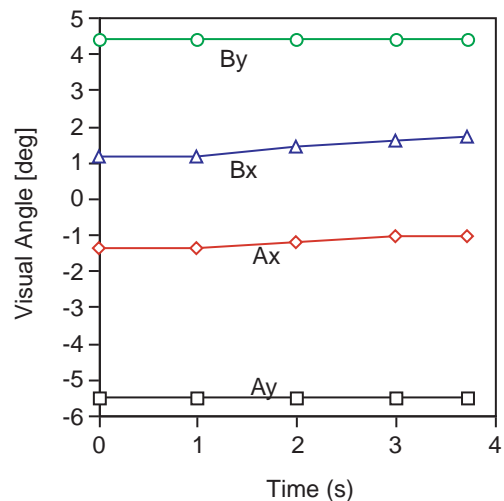


Fig.16 Evaluation of detached method (6).

### 3.3 各経路計画法による動作実験

ここでは2章で述べたデータベースとさまざまな方法を用いて具体的に経路探索し、その結果得られた経路と落下実験を行った結果を示す。比較のため、探索した経路の初期状態と目標状態は同じとし、初期状態は  $q_1 = 195^\circ, q_2 = -75^\circ, p_1 = p_2 = 0$  (実際は画像の中央付近であるが、便宜上このようにおいた)、目標状態は  $q_1 = 15^\circ, q_2 = -75^\circ, p_1 = p_2 = 0$  とした。これは視覚センサの方向を変えずに肩関節だけを  $180^\circ$  振る動作を表している。

#### (1) 幅優先探索

現実的な時間・計算資源で経路計画を終了するため、ラベルが100ピクセル以内であればよいことにした。この結果、93,452 節点を探索した後 (Sun Sparc10 で5,179秒 かかった)、Fig.17 のような7動作の経路が得られた。この時の光源の位置の変化をFig.18 に示す。図中、(a) の黒丸は仮定している光源の位置を示す。また、(p1e, p2e), (p1s, p2s), (p1c, p2c) はそれぞれ

光源と仮定したグラフ用紙上の点の実験結果、シミュレーション結果、データベースから得られる推定結果を示している。 $p_1$  については、実験結果の方が変化が少ないが、傾向はシミュレーションやデータ

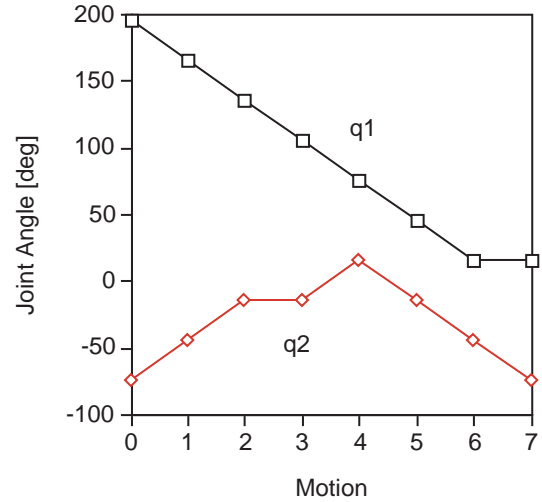


Fig.17 Results of path planning with breadth-first search and GA.

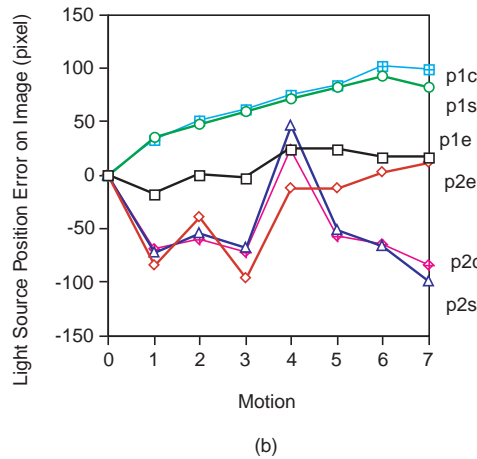
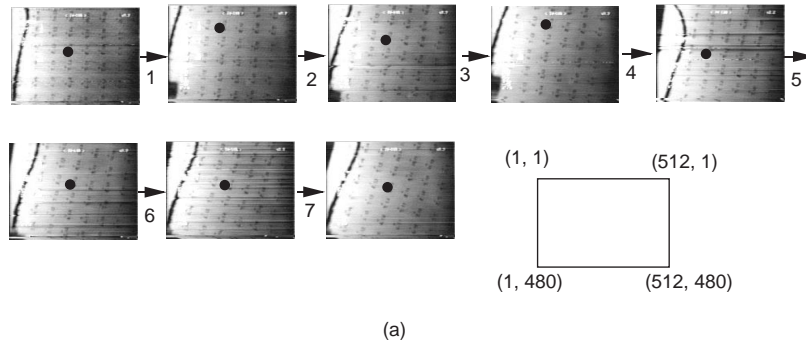


Fig.18 Light source position error on image (breadth-first search and GA). (a) CCD images on robot. (b) Graph.

ベースからの推定とほぼ一致している。また、 $p_2$ は、実験結果が5動作以後、シミュレーションや推定結果とずれているが、それ以前はほぼ一致している。これらの誤差の原因としては、初期の運動量や実験ロボットとシミュレーションに用いたモデルとの不整合が考えられる。

(2) A\*アルゴリズム

ここでも目標節点でラベルが100ピクセル以内であればよいことにした。この場合、最短経路は保証されないが、282節点の探索の後、11動作が見つかった(Fig.19)。この経路は人間の直感に近い経路である。まずはじめに腕を伸ばして腕全体のz軸回りの慣性モーメントを小さくし(3動作まで)、そのまま腕を旋回し(3動作~7動作)、最後に腕を再び曲げている(9動作~11動作)。この時の光源の位置の変化

をFig.20に示す。 $p_1, p_2$ とも傾向はシミュレーションやデータベースからの推定と一致している。

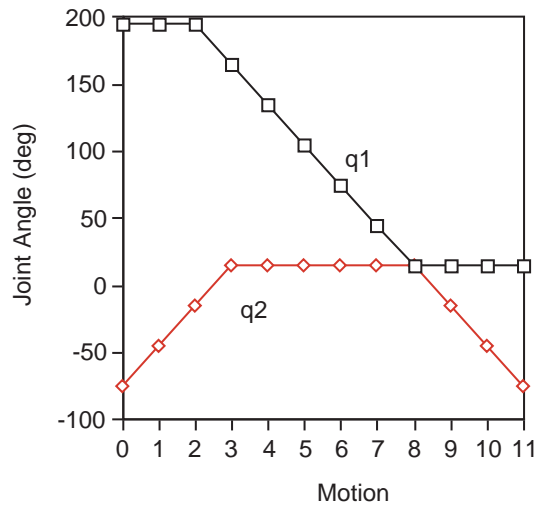


Fig.19 Results of path planning with A\* algorithm.

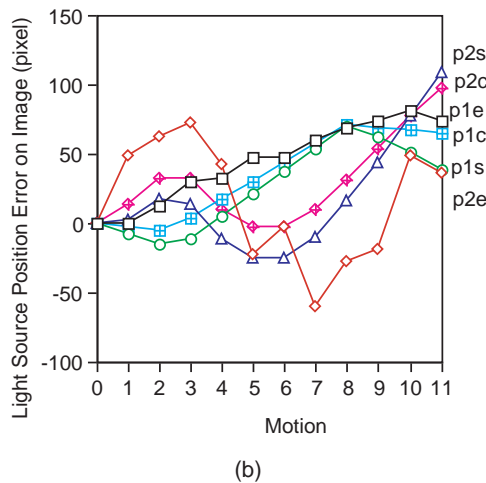
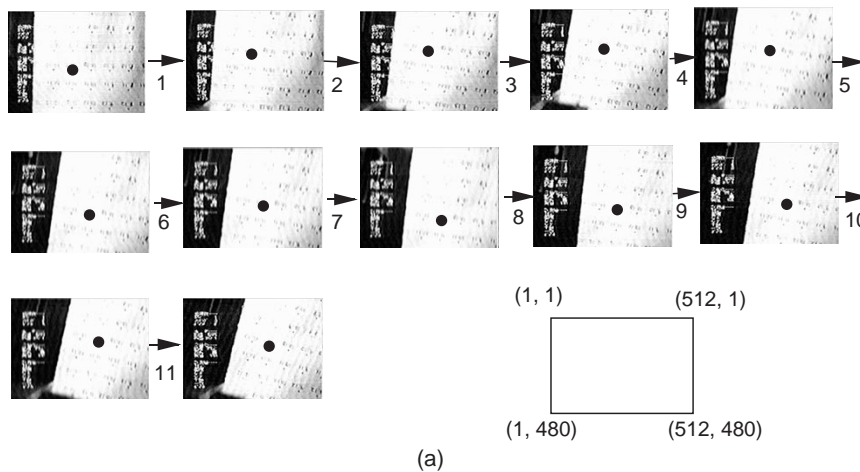


Fig.20 Light source position error on image (A\* algorithm). (a) CCD images on robot. (b) Graph.

### (3) GA

遺伝子の個体数 30, 許容される遺伝子の長さ (最大動作数) 30, 突然変異の確率 0.5, 世代交代数 2500 として計算機実験を行った。交配の経過 (進化のようす) を Fig.21 に示す。27 世代目に目標状態をもつ遺伝子が現れ, その後遺伝子の長さが徐々に短くなり, 最終的には 7 動作の動作が得られた。この経路は幅優先探索で得られたものと同じであったので, 実験結果は省略する。ここでも目標節点で, ラベルが 100ピクセル以内であればよいことにした。

### (4) 最適制御

2章で得られる経路は動作時間を 10秒としているため, 落下実験では少し縮める必要がある。そこで,

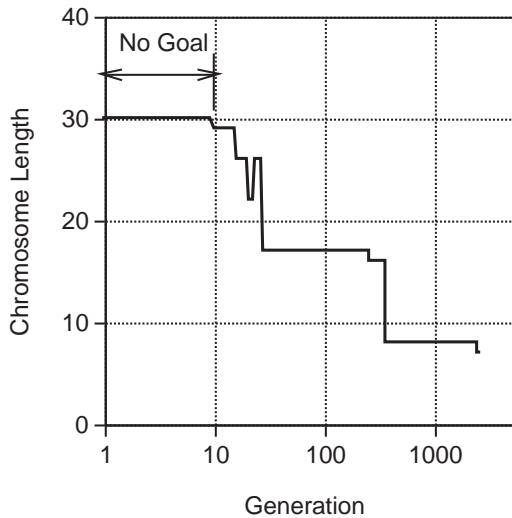


Fig.21 Evolution process of GA.

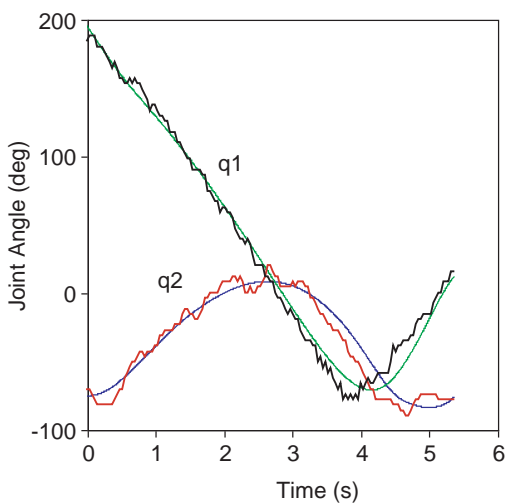


Fig.22 Joint motion of optimal control.

動作時間としては 5.5秒とした。また, ロボットは関節角速度では連続的に制御できないので, 得られた経路を 20分割し, それらを順次目標値として与えることにより動かした。実験中の実際の関節動作を Fig.22 に示す。図中, 点線は 2章の経路計画で得られたものである。また, 輝点の動いた結果を Fig.23 に示す。 $p_2$ の動きは計算値と比較的よく合っているが,  $p_1$ は全般に上にシフトしている。最終的には輝点を表す点が動作終了後も視野のほぼ中央にあることがわかる。

### (5) 対照実験

以上の経路計画法の効果を明確にするために,  $(q_1, q_2) = (195, -75)$  から  $(15, -75)$  への輝点の位置変化を考慮しないもっとも単純な経路をとった場合との比較を行う。この経路では  $q_2$  は変化させずに  $q_1$  だけ回転させてある。この場合の輝点に相当する方眼紙上の点の位置変化の実験結果を Fig.24 に示す。この実験では, 無重力になった後, 3 秒からロボットは動作を始め, 5.5 秒で動作を終了した。Fig.24 はこの時間で表示してある。図中, (a)は画像を, (b)は画

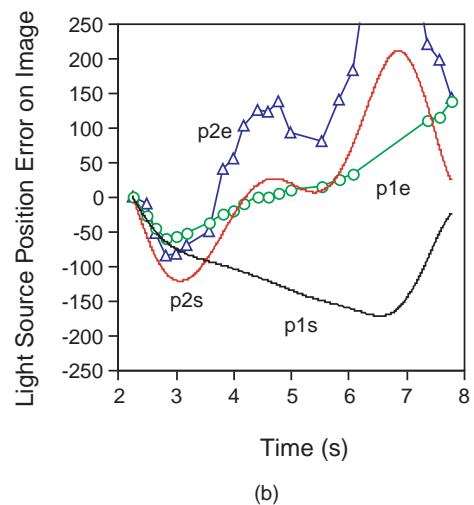
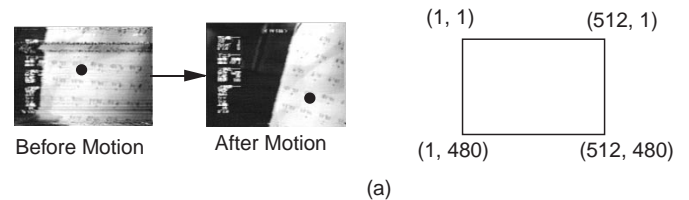


Fig.23 Light source position error on image (optimal control). (a) CCD images on robot. (b) Graph.

像を整理したグラフで示す。初め中央付近（ピクセル値で(258, 282)）にあった点（A）は途中で視野の外へでてしまったので、図には実験中ずっと追跡できた点（B）も合わせて示してある。点Aと点Bが画像中を平行に移動すると仮定すると、最終的な点Aの位置はピクセル値で(-36, 261)となる。腕の様な回転の反作用のために輝点は一様速度(-x方向)で移動している。また、ロボットの腕が曲がったままで運動するため、系全体の重心位置が変化することによるロボットのz軸回りの振れ（画像のy方向の往復）が観察される。

これらの比較をさらにわかりやすくするため、無重力実験での正面からみたロボットの初期状態と最終状態の写真を Fig.25 に示す。(a) - (c) は経路計画を行った場合でロボットの側面の向きが変化していないのに、(d) の最短経路をとった場合には側面が下を向いて傾いているのがわかる。

提案した経路計画法により、腕の形態と胴体姿勢の同時制御を無重力で実証することができた。

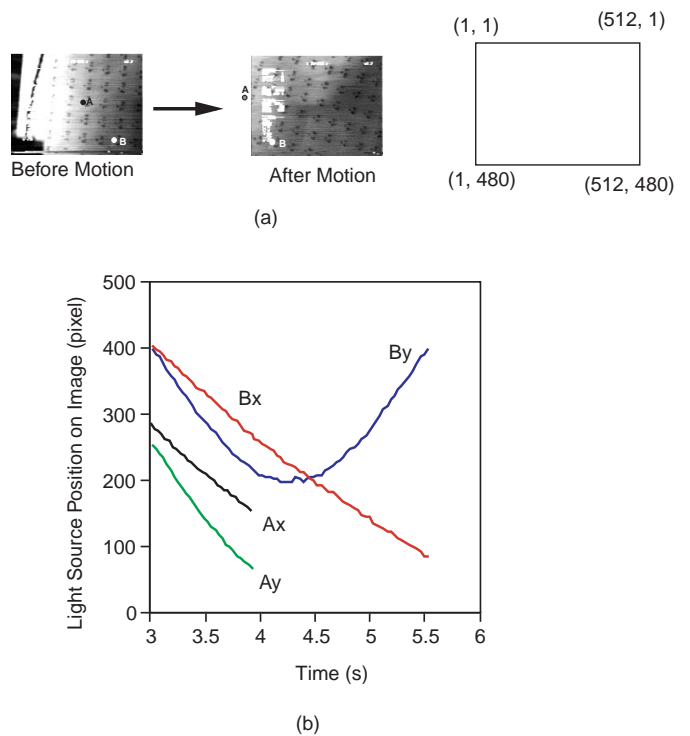


Fig.24 Results of simplest motion. (a) Robot camera images. (b) Light source position during motion.

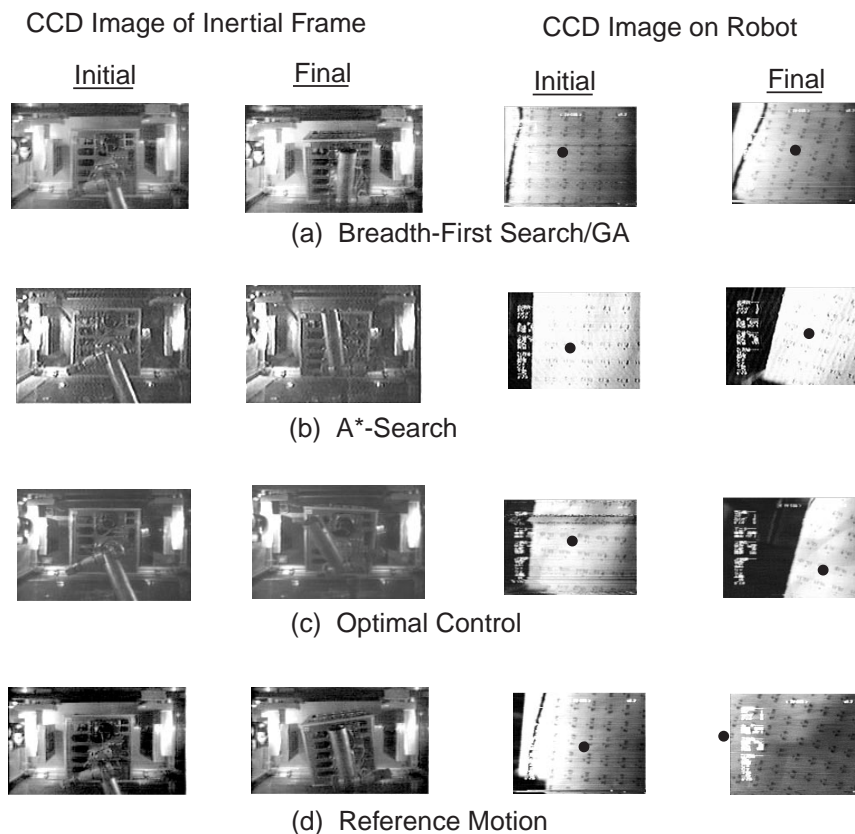


Fig.25 Summary of drop shaft experiments.

## §4 水槽を用いた実験

3章で述べた落下実験では実験時間の制約があり、経路計画のもとになるデータベースは計算機のシミュレーションによらざるを得なかった。そこで、提案した方法全体を実験で行うため、水の浮力を用いた実験を行い、データベースの獲得と、そのデータベースに基づいて幅優先探索による経路計画を行い、実験を行った。この実験は提案した方法が角運動量保存則を満たさない場合にも有効であることを示す目的もつ。

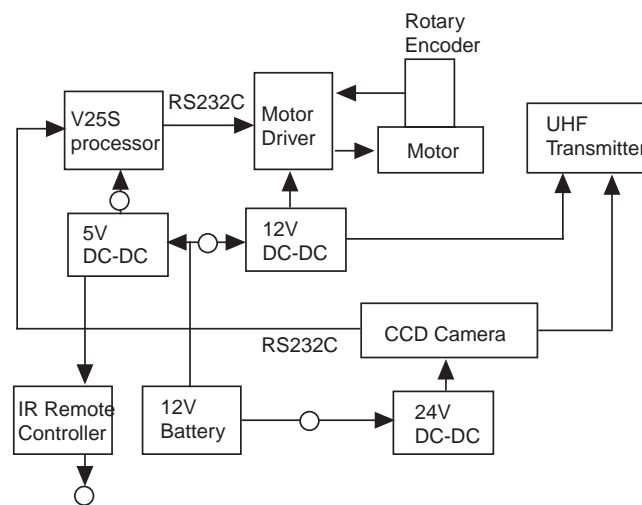
### 4.1 装置

#### (1) 水槽実験用ロボット

実験で用いたロボットを Fig.26 に示す。(a)は写真で、全体はアクリル製とし、もしも漏水が起こってもすぐに発見できるようにした。ロボットの自由度配置は落下実験で用いたロボットと同じである。また、リンク毎の浮力と重力、浮心と重心のバランスを取るためにロボット本体と腕先端に真鍮による重りでのバランスを取れる機構を設けた。ロボットのブロック線図を(b)に示す。落下実験ロボットとの違いは、モータの制御にエンコーダとモータドライバを用いたこと、CCD



(a)



(b)

Fig.26 Robot model for underwater experiments.  
(a) Photograph. (b) Block diagram.

カメラは画像処理能力をもち、その結果をRS232CでCPUに伝送するようにしたことである。また、CCDカメラの視野の全角度は光源を見失うことをなるべく防ぐため、広角レンズを取り付け、水平方向で71.5°とし、落下実験の約4倍とした。画像処理結果の伝送は、データベースをロボット自身が作成するために必要な機能である。

(2) 実験環境

ロボットで調整した浮力・重力、浮心・重心の調整は微妙であり、系は不安定平衡点にあるため、長時間の実験には向かない。試作したロボットは推進器を持たないので、自身では並進運動ができず、そのままでは実験を継続できなくなる可能性もある。そこで、ロボット底部に重りを付けた紐を取付け、ロボットは少し浮くように調整して水中に設置した。また、輝点としてはLEDを水槽外部に設置した。水槽は透明なアクリル製とし、LEDをロボットから直接見ることができるようにした。この様子を Fig.27 に示す。

4.2 実験結果

(1) データベースの構築

まずはじめに、データベースの構築を行った。実験中にロボットの姿勢が変化し、ロボットが輝点を見失うことがあるため、実験を7回に分けて行った。つまり、肩関節  $q_1 = 15^\circ, 45^\circ, 75^\circ, 105^\circ, 135^\circ, 165^\circ, 195^\circ$  毎に行い、水の粘性の効果を避けるために関節角速度を6.7°/sとした。ロボットは30°関節を動かすので、

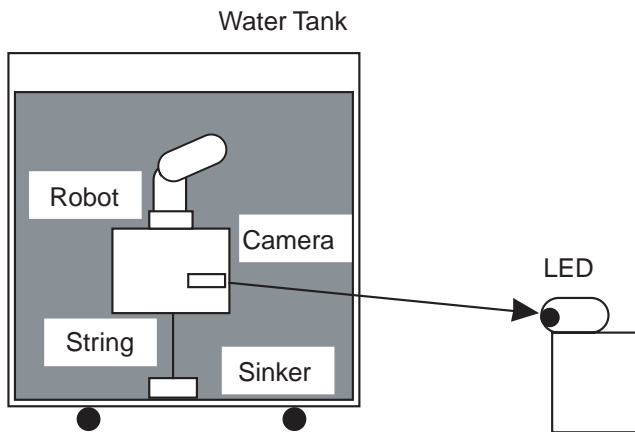


Fig.27 Setup for underwater experiments.

1動作約4.5秒かかることになる。データ獲得の順序は以下のとおりである。現在の関節位置を  $(q_{1c}, q_{2c})$  とすると、動作は  $(q_{1c}, q_{2c})$   $(q_{1c} - 30^\circ, q_{2c} - 30^\circ)$   $(q_{1c}, q_{2c})$   $(q_{1c} - 30^\circ, q_{2c})$   $(q_{1c}, q_{2c})$   $(q_{1c} - 30^\circ, q_{2c} + 30^\circ)$   $(q_{1c}, q_{2c})$   $(q_{1c}, q_{2c} - 30^\circ)$   $(q_{1c}, q_{2c})$   $(q_{1c}, q_{2c} + 30^\circ)$   $(q_{1c}, q_{2c})$   $(q_{1c} + 30^\circ, q_{2c} - 30^\circ)$   $(q_{1c}, q_{2c})$   $(q_{1c} + 30^\circ, q_{2c})$   $(q_{1c}, q_{2c})$   $(q_{1c} + 30^\circ, q_{2c} + 30^\circ)$   $(q_{1c}, q_{2c})$  の順に16動作行い、このところで画像の変化を計測し、データベースとした。動作が可動範囲を越える場合は省略した。例えば、 $q_1 = 45^\circ$  の時には、現在位置として6個選べる ( $q_2 = -75^\circ, -45^\circ, -15^\circ, 15^\circ, 45^\circ, 75^\circ$ )。結果として、 $q_1 = 45^\circ$  の時の学習時間は可動範囲を越えるものもあるので以下の見積よりは少し小さくなるが、およそ4.5(秒) × 16(動作) × 6(状態) = 432(秒) ~ 7(分)、センサ/動作データベース全体を得るための学習時間は、およそ7(分) × 7(実験) = 49(分)必要となった。

(2) 経路計画

$q_1 = 15^\circ, q_2 = -75^\circ$  から  $q_1 = 195^\circ, q_2 = -75^\circ$  まで光源の画像上での位置を12ピクセル以下で動かす経路計画を立てた。計画された経路を Fig.28 に示す。この経路は43,040個のノードを探索した後得られた。光源の位置の変化を Fig.29 に示す。

(3) 対照実験

水槽を用いた実験においても経路計画の効果を確認するために対照実験を行った。対照実験で用いた経路は、 $q_1$  を15°から195°まで連続して動かし、 $q_2$  は -75°

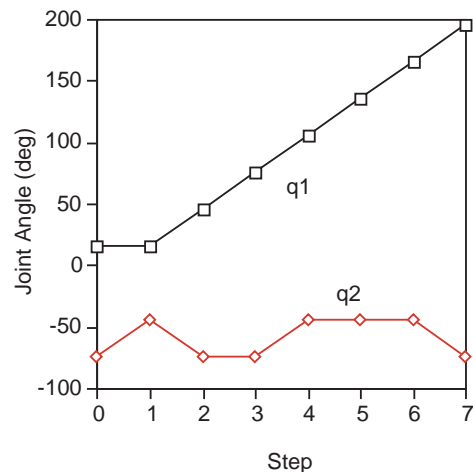


Fig.28 Results of path planning with breadth-first search.

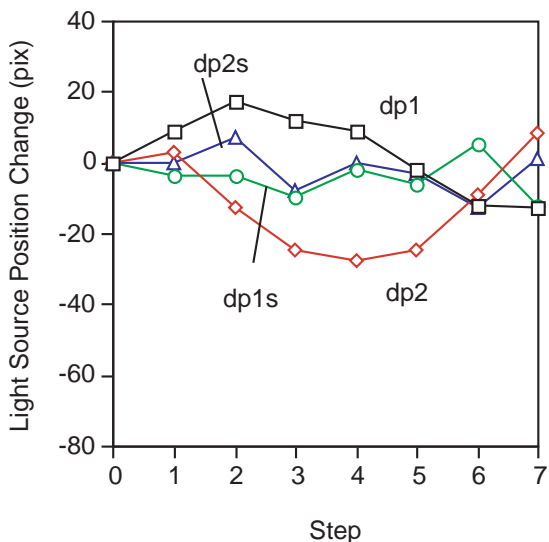


Fig.29 Light source position error on image (breadth-first search).

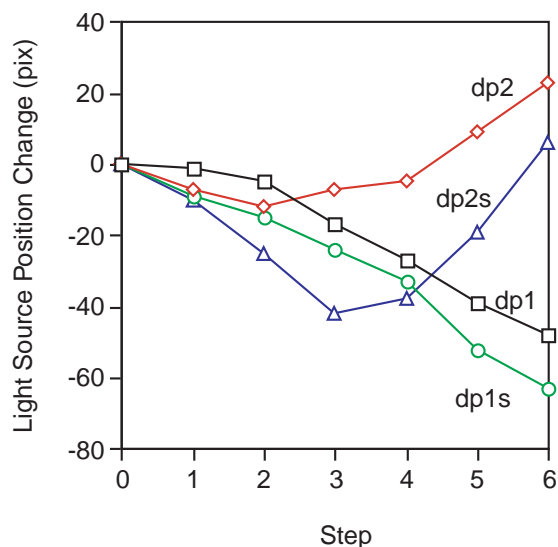


Fig.30 Results of simplest motion.

で変化させないというものである。結果を Fig.30 に示す。Fig. 29 と比較すると、経路計画の効果があったことがわかる。これらにより、提案した方法が角運動量保存則を満たさない場合でも有効であることが示された。

## § 5 考察

### 5.1 光源の位置の変化

本稿で述べた方法では、データベースを作成するとき、画像の変化( $dp_1, dp_2$ )が画像内全体で同じであると仮定している。しかし、もしも画像の見える方向軸回りに回転や、光源とカメラの間の相対距離が動作中に変わるとこの仮定は成立しない。本研究

の場合、ロボットの動作とロボットの質量中心の位置に依存することになる。Fig.31 は全画像面の変化の例である。矢印は動作前後の光源の位置の変化を表している。動作前に光源が画像上の  $(255+55i, 255+55j)$  ( $i, j = -4, \dots, 4$ ) にあるとし、同じ関節角 ( $q_1 = 15^\circ, q_2 = -75^\circ$ ) から (a) は  $ac_1 = 30^\circ, ac_2 = 0^\circ$ , (b) は  $ac_1 = ac_2 = 30^\circ$  動かした場合である。(a) は全面でほぼ様な動きであるのに対し、(b) は大きな分布を持っている。(a) ではロボットの回転が  $z$  軸回りに起こっているのに対して、(b) では  $x$  軸 (カメラの見える方向) 回りに起こっているためである。このことは誤差を発生するが、データベースを小さくするためには全面で変化が一樣であるという仮定を受け入れる必要がある。

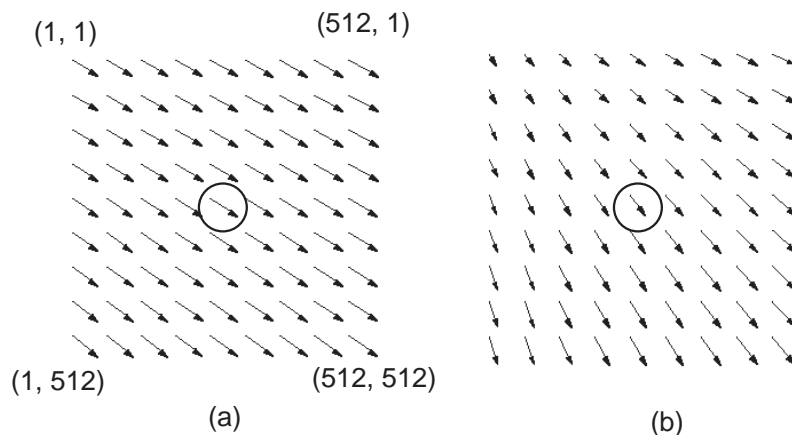


Fig.31 Image distribution.

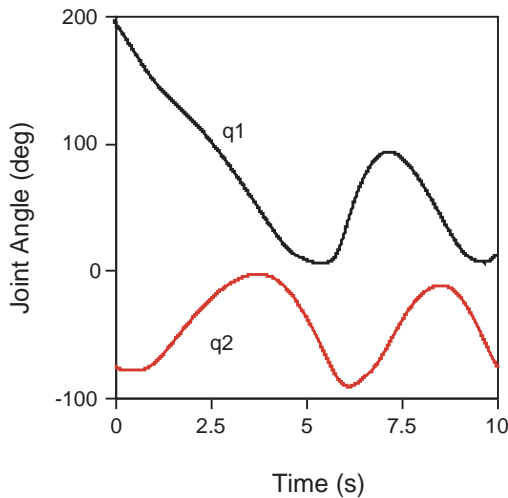
## 5.2 GAの優位性

本研究ではGAでも幅優先探索と同じ結果が得られたが、両者の比較を行うことにする。幅優先探索の場合、結果が出るまで93,452の節点を探索し、その間、経路の一時的な候補は存在しなかった。一方、GAでは最短経路は73,080個の遺伝子を調べたことになる。GAの場合、枝刈りをする必要がなく、メモリや時間が節約できるが、これだけでは幅優先探索に比べて優位性があると言えるほどでもない。さらに、経路やGAのパラメータの選び方によって探索される経路の数も異なる。しかし、一般的にGAは仮の経路の候補を早い段階から持つ。本研究の場合でも29動作という長いものであるが、10世代目ですでにこの候補を持っている。この性質は実際に応用する場合には重要な性質である。今回と同じパラメータを使うと、もしも最短である必要はなく、10動作以下の経路が許されるとすると、 $30 \times 356 = 10,680$ の経路を探索するだけで8動作の経路を得ている。

## 5.3 データベースの範囲内での最適制御

本稿で述べた最適制御の実験は $q_1$ がデータベースの範囲を越えており、誤差の原因になりうる。そこで、データベースの範囲で経路を求めてみた。このために、さらに次のペナルティ関数

$$f_{p1} = \begin{cases} 0 & (x_1 \geq x_{1c}) \\ w_1 \left( \frac{x_1 - x_{1c}}{x_{1m} - x_{1c}} \right)^2 & (x_1 < x_{1c}) \end{cases} \quad (7)$$



を使った。ここで、 $w_1 = 2$ ,  $x_1 = q_1$ ,  $x_{1c} = 15^\circ$ ,  $x_{1m} = 0^\circ$ である。評価関数を次のように書き換える。

$$L = \int_0^{10} (u_1^2 + u_2^2 + f_{p1} + f_{p2}) dt \quad (8)$$

この結果得られた経路と光源の位置を Fig.32 に示す。得られた経路はサイクリックな腕の動きのように見えるが、これまでサイクリックな腕の動きが腕の形状とロボットの姿勢を制御するのに有効であることがいくつか報告されている<sup>3-5)</sup>。ここで得られた結果は、腕の動作範囲を限定するときこのような宇宙ロボットのもつ性質を確認したことになった。

## 5.4 水槽実験の誤差の原因

ロボットの視野の角度を落下実験に比べて大きく取ったため、姿勢の変化に対して光源の位置変化が十分得られなかったことが考えられる。例えば、Fig. 29 や Fig. 30 では動作毎の変化としては10ピクセル程度かそれ以下であり、角度にすると $1.4^\circ$ 程度になる。関節動作時間が4.5秒なので、角速度にすると約 $0.3^\circ/s$ となり、水中では制御が困難である。

また、別の理由として、水の粘性が考えられる。データベースを獲得するとき、関節の動作は交互に行われたのに対し、経路計画の結果は連続して同じ方向への運動が含まれている。水の粘性により、同じ方向の運動が加速された可能性がある。

そのほかに、ロボットそのものにも問題が考えられる。関節部分の駆動部に遊びがあるため、予想さ

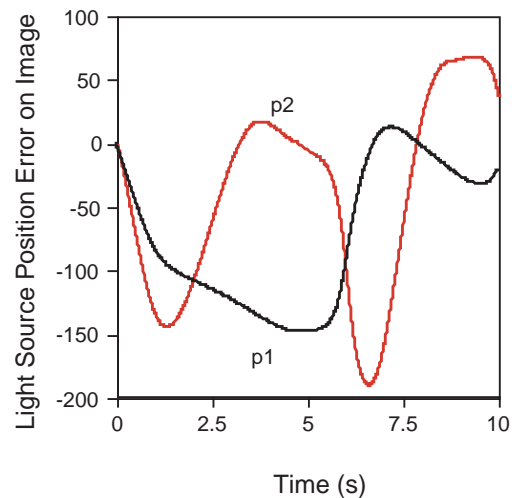


Fig.32 Results of optimal control using eqs. (7) and (8).

れる運動よりも関節の動きが小さくなっており、これも誤差を与える要因と考えられる。

### 5.5 提案した4つの方法の比較

提案した4つの方法は、いずれも同じセンサ/動作データベースを用いて経路計画を行うものであるが、それぞれに長所、短所を持っている。それらをTable 1にまとめる。現実的な問題に対応する場合、要求としては(1)経路計画が早く行えること、(2)連続値として関節角が得られること、が重要になると考えられるので、関節角の刻みを細かくしてGAを用いるか、最適制御の考え方で経路計画をするのがよいと考えられる。

## §6 結論

本研究で行ったことと得られた知見は以下のとおりである。

- (1) 宇宙ロボットが重力中で腕の経路により腕の最終的な形状とロボット本体の姿勢を同時に制御するため、ロボットが自分自身の数理的なモデルを持たなくても可能な方法として、腕の運動とロボットの画像センサの変化の関係をデータベース化する方法を提案し、幅優先探索、A\*アルゴリズム、GA、最適制御を用いて経路の決定を行った。

- (2) これらの方法を実験的に確認するため、落下施設を使った無重力実験と、水槽で水の浮力を用いた疑似無重力実験を行った。
- (3) 落下実験では、ロボットの切り離し方法が問題であり、落下後ロボットを一度切り離したあとすぐにもう一度保持し、さらに切り離すことで安定した切り離しが実現できるようになった。
- (4) 落下実験では実験時間が短いので、データベースの獲得は計算機を用いて行った。幅優先探索、A\*アルゴリズム、GA、最適制御の全てについて実験を行い、対照実験との比較により提案した方法の有効性を示した。
- (5) 水槽を用いた実験では浮力と重力、浮心と重心のバランスが問題となるが、ロボットに取り付けた重りによる調整と、重りを付けた紐によりバランスを実現し、実験を行った。
- (6) 水槽では実験時間を十分確保できるため、データベースの獲得も実験で行った。経路計画については幅優先探索についてだけ行った。対照実験との比較により、角運動量保存則が成り立たない場合でも経路計画の有効性を示すことができた。

なお、本研究の一部は、NEDO(新エネルギー・産業技術総合開発機構)からJSUP(宇宙環境利用推進センター)が委託を受けて実施したものである。

Table 1 Comparison among proposed method.

Search Method	Advantage	Disadvantage
Breadth-first Search	The path found is constructed of the shortest steps.	Search time is long and memory is wasted.
A* Search	Search time is short.	Estimated cost is subjective.
Genetic Algorithm	Temporary solution is found in early stage.	Optimalization is not guaranteed.
Optimal Control	Continuous values can be dealt with.	Penalty function is necessary if a solution is not realized.

## 参考文献

- 1) K. Yamada: JGCD, Vol. 17, No. 4 (1994) pp. 1050-1054.
- 2) Y. Nakamura and R. Mukherjee : IEEE Trans. RA, Vol. 7, No. 4 (1991) pp. 500-514.
- 3) R. W. Longman : J. Astronautical Sci., Vol. 38, No. 4 (1990) pp. 423-440.
- 4) Y. Nakamura and T. Suzuki : JSR, Vol. 34, No. 1 (1997) pp. 137-143.
- 5) K. Yamada and S. Yoshikawa : JGCD, Vol. 20, No. 4 (1997) pp. 715-720.
- 6) R. S. Norworthy : AIAA Conference in Intelligent Robots for Factory, Field, Service, and Space, Washington, D.C., 1994, pp. 757-762.
- 7) D. L. Akin and J. R. Spofford : JGCD, Vol. 13, No. 3 (1990) pp. 515-523.
- 8) T. Iwata, K. Machida, Y. Toda and M. Kawada : International Symposium on Space Technology and Science, Omiya, 1997, pp. 587-592.
- 9) 丸山, 飯田, 江尻, 佐藤, 神田, 内山, 内山, 藤井 : 日本ロボット学会誌, Vol. 17, No. 6 (1999) pp. 828-838.
- 10) 下地, 井上, 稲場, 若林 : 日本ロボット学会誌, Vol. 13, No. 1(1995) pp. 127-133.
- 11) 岩田, 樹神, 沼尻, 菊田, 村上 : 日本ロボット学会誌, Vol. 13, No. 8 (1995) pp. 146-149.
- 12) 岩田, 村上, 樹神, 沼尻, 鈴木 : 日本航空宇宙学会誌, Vol. 45, No. 527 (1997) pp. 705-712.
- 13) 中村, 渡辺, 荒木 : 日本ロボット学会誌, Vol. 17, No. 7 (1999) pp. 974-982.
- 14) P. H. Winton: Artificial Intelligence, 2nd ed. (Addison-Wesley, Reading, MA, 1984) pp. 87-132.
- 15) S. Russell and P. Norvig : Artificial Intelligence: A modern Approach (Prentice-Hall, Englewood Cliffs, NJ, 1995) pp. 87-132.
- 16) 平野 : インターフェース, CQ出版社, No. 2, Vol. 177 (1992) pp. 108.
- 17) A. K. Wu and A. Miele : Optimal Control Applications and Methods, Vol. 1, No. 1 (1980) pp. 69-88.
- 18) S. Gonzales and S. Rodrigues : J. Optimization Theory and Applications, Vol. 50, No. 1 (1986) pp. 109-128.