大域的探索とこう配法によるマニピュレータの 障害物回避準最短時間軌道計画*

山 本 元 司^{*1}, 毛 利 彰^{*1}

Collision Free Quasi-Minimum-Time Trajectory Planning for Manipulators Using Global Search and Gradient Method

Motoji YAMAMOTO and Akira MOHRI

A method for time optimal collision free trajectory planning problem is proposed. In this paper, the dynamics of manipulator and obstacles in workspace are considered. This optimal trajectory planning problem is very difficult and complicated one because of non-linearity of manipulator dynamics and existence of obstacles. Generally, it results in a multimodal optimization problem. The proposed method is basically an iteratively improving one based on a gradient method. Firstly, two global methods (a genetic algorithm and an exact cell decomposition method) are used to search multiple initial feasible spatial paths for the gradient method. Next, the gradient method searches time optimal solution locally with the multiple initial feasible solutions. Simulation results show that this approach is effective to the time optimal collision free trajectory planning problem.

Key Words: Manipulator, Trajectory Planning, Collision Avoidance, Minimum-Time, Gradient Method, Global Search

1. ま え が き

障害物が存在する作業空間においてマニピュレータに効率よく作業を行わせたい場合、幾何学的な問題としての障害物との衝突回避と、動力学を考慮した上での時間軌道の最適化を同時に考えなければならない。本論文ではこのような動力学を考慮したマニピュレータの障害物回避準最短時間軌道計画問題を検討する。

今までにロボットマニピュレータの障害物回避問題に対しては多くの研究がなされ、さまざまな方法が提案されている⁽¹⁾⁽²⁾。しかしそのほとんどは幾何学的な拘束条件のみを考慮したものであり、動力学を考慮した障害物回避軌道計画は、その問題の複雑さからあまり検討されなかった。完全な形で動力学を考慮した軌道計画の研究としてDubowskyら⁽³⁾、Shinら⁽⁴⁾、尾崎ら⁽⁵⁾による研究がある。これらは空間的な移動経路があらかじめ与えられる場合の最適な時間軌道を求める方法を示している。障害物回避問題との関連で考えるなら、種々提案されている幾何学的な経路生成法で生成された経路にこれらの方法を適用して時間軌道を得ることになる。しかし、このとき得られる時間軌道

の最適性は当然空間経路に依存するため、真の最適軌道を生成するためには空間経路自体も同時に最適化されなければならない。

この空間経路の最適化も含めた軌道生成に関する研究として、最適制御理論(最大原理)を応用する方法^{(6)~(9)}や、離散化した空間を探索することにより(準)最適軌道を求める方法^{(10)~(12)}などが提案されている。しかしこれらはマニピュレータが作業空間内を自由に移動できることを仮定しているため、本論文で検討する障害物が存在する場合の軌道の最適化問題にはこれらの方法はそのままでは適用できない。

一方Shillerら⁽¹³⁾は本論文で検討する動力学と障害物の存在を考慮した軌道計画問題を取り扱っている。彼らはまず作業空間を格子状に離散化し、分枝限定法により大域的に探索を行い、次にスプライン関数の制御点に対して局所最適化を行うことで最短時間軌道生成を試みている。しかし彼らの方法では、探索回数を減らすために最適軌道が経験的に仮定されており、もし離散化された空間全域にわたって大域的な探索を実行すると膨大な計算時間が必要であることが容易に予想される。

本論文で検討するのは、作業空間内に障害物が存在する場合の最適軌道計画問題であり、かなり解くのが

* 原稿受付 1995年3月2日。

*¹ 正員、九州大学工学部 (〒812 福岡市東区箱崎6-10-1)。

困難な非線形最適化問題と考えられる。そこで著者は必ずしも厳密な最適軌道が得られなくても、ほぼ最適解に近い良い解がいくつか得られれば満足することにする。この考え方により著者は複数の比較的良い初期解から探索するこう配法を基本とする方法を提案する。この方法において複数の比較的良い初期軌道を得る方法として遺伝的アルゴリズムあるいは完全セル分割法を用いることを提案する。また例題において提案する方法の有効性を示すとともに、複数の初期解を得るための大域的探索能力としてのこの二つの方法(遺伝的アルゴリズムと完全セル分割法)を比較、検討する。

2. 軌道計画問題と経路パラメータによるダイナミクス表現

一般に n 自由度マニピュレータのダイナミクスは次のように表せる。

$$H(\theta)\ddot{\theta} + h(\theta, \dot{\theta}) + g(\theta) + D\dot{\theta} = u \quad \cdots \cdots (1)$$

ここで、 $\theta \in R^n$ は関節変数ベクトル、 $H(\theta) \in R^{n \times n}$ は慣性マトリクス、 $h(\theta, \dot{\theta}) \in R^n$ は遠心・コリオリ力ベクトル、 $g(\theta) \in R^n$ は重力ベクトル、 $D \in R^{n \times n}$ は粘性摩擦係数の対角マトリクス、 $u \in R^n$ は駆動力ベクトルを表す。駆動力と関節速度は次のように制限されるとする。

$$u_{imin} \leq u_i(t) \leq u_{imax}, (i=1, \dots, n) \quad \cdots \cdots (2)$$

$$\dot{\theta}_{imin} \leq \dot{\theta}_i(t) \leq \dot{\theta}_{imax}, (i=1, \dots, n) \quad \cdots \cdots (3)$$

ここで、 t は時間を表し、 u_{imax} , u_{imin} は駆動力の、 $\dot{\theta}_{imax}$, $\dot{\theta}_{imin}$ は関節速度の上限、下限をそれぞれ表す。また、 $u_{imin} \cdot u_{imax} < 0$, $\dot{\theta}_{imin} \cdot \dot{\theta}_{imax} < 0$ とする。さらに、障害物との衝突回避のため制約として

$$X_M(t) \cap X_O = \phi \quad (0 \leq t \leq t_e) \quad \cdots \cdots (4)$$

が必要となる。ただし $X_M(t)$ は時刻 t におけるマニピュレータの占有空間、 X_O は障害物の占有空間を表す。軌道の両端での境界条件として次のように位置と速度が与えられるとする。

$$\theta(0) = \theta_0, \theta(t_e) = \theta_e \quad \cdots \cdots (5)$$

$$\dot{\theta}(0) = \dot{\theta}_0, \dot{\theta}(t_e) = \dot{\theta}_e \quad \cdots \cdots (6)$$

ここで t_e は終端での時刻を表す。本論文で扱う(準)最短時間軌道計画問題は式(1)のダイナミクスに対して式(2)～(4)の制約条件と式(5), (6)の境界条件を満足する入力 $u(t)$ を求める問題である。

マニピュレータの軌道計画問題において、関節の空間経路が与えられる場合については、経路の長さを表すパラメータを用いて効率的に最適軌道を計画する方法⁽⁵⁾が提案されている。この方法の基本的な考え方は以下になる。まず、正規化されたパラメータ $s(0$

$\leq s \leq s_e)$ で表された関節経路を $\theta(s)$ と書き、 s を経路パラメータと呼ぶ。時間 t と経路パラメータ s を結びつける量として $\lambda(s)$ を次のように定義する。

$$dt = \frac{1}{\lambda(s)} ds \quad \cdots \cdots (7)$$

この関係を用いてダイナミクス式(1)を関節経路 $\theta(s)$ を用いて書き換えると、

$$\begin{aligned} \lambda(s)^2 \left[H(\theta(s)) \frac{d^2 \theta(s)}{ds^2} + h\left(\theta(s), \frac{d\theta(s)}{ds}\right) \right] \\ + \lambda(s) \left[\frac{d\lambda(s)}{ds} H(\theta(s)) + D \right] \frac{d\theta(s)}{ds} \\ + g(\theta(s)) = u \quad \cdots \cdots (8) \end{aligned}$$

となる⁽¹⁴⁾。上式は関節経路 $\theta(s)$ が与えられれば、 $\lambda(s)$, $d\lambda(s)/ds$ を決めることによりダイナミクスが計算できることを示している。

3. B スプラインの制御点を用いた経路表現

ここでは、B スプラインの制御点を用いてマニピュレータの空間経路を表現する方法について述べる。

関節空間上で B スプラインの制御点 $\hat{\theta}_k = [\hat{\theta}_k^1, \dots, \hat{\theta}_k^n]$ ($k=0, \dots, m$) が与えられたとき、経路パラメータ $0 \leq s \leq s_e$ を m 個の等間隔の区間 [$s_0 (=0), \dots, s_m (=s_e)$] に分割し、 $s_k \leq s \leq s_{k+1}$ における空間経路 $\theta(s)$ を位数 4 の B スプライン関数を用いて表すと次のようになる。

$$s(l) = s_k + l\Delta s, 0 \leq l \leq 1 \quad \cdots \cdots (9)$$

$$\theta(s) = \theta(l) = \sum_{j=-1}^2 \hat{\theta}_{k+j} N_{k+j,4}(l) \quad \cdots \cdots (10)$$

$$\frac{d\theta(s)}{ds} = \frac{1}{\Delta s} \sum_{j=-1}^2 \hat{\theta}_{k+j} \frac{dN_{k+j,4}(l)}{dl} \quad \cdots \cdots (11)$$

$$\frac{d^2 \theta(s)}{ds^2} = \frac{1}{(\Delta s)^2} \sum_{j=-1}^2 \hat{\theta}_{k+j} \frac{d^2 N_{k+j,4}(l)}{dl^2} \quad \cdots \cdots (12)$$

ここで、

$$\left. \begin{aligned} \Delta s &= s_{k+1} - s_k \\ N_{k-1,4}(l) &= (-l^3 + 3l^2 - 3l + 1)/6 \\ N_{k,4}(l) &= (3l^3 - 6l^2 + 4)/6 \\ N_{k+1,4}(l) &= (-3l^3 + 3l^2 + 3l + 1)/6 \\ N_{k+2,4}(l) &= l^3/6 \end{aligned} \right\} \quad \cdots \cdots (13)$$

ここでは関節駆動力の連続性を考えて、B スプラインの位数を 4 とすることで、関節経路 $\theta(s)$ の加速度までを連続にしている。

以上により B スプラインの制御点の組によって、空間経路が得られることがわかる。マニピュレータ動作の空間経路が与えられている場合には、第 2 章に示した考え方によりスカラ関数 $\lambda(s)$ を最適化することにより効率的に最適軌道を求めるアルゴリズム⁽⁵⁾が提案されている。ここではそのアルゴリズムを

MTTPPP (Minimum Time Trajectory Planning with Preplanned Path) のアルゴリズムと呼ぶことにする。この方法により、いったん空間経路が B スプラインの制御点により与えられたなら、MTTPPP のアルゴリズムによりその最短動作時間が一意的に決定されるため、境界条件と制約条件が同じであれば最短動作時間は次のように制御点のみの関数となる。

$$T_{\min} = f(\hat{\theta}) \quad \dots\dots\dots (14)$$

4. 遺伝的アルゴリズムによる 大域的な初期解の探索

4.1 遺伝的アルゴリズム 遺伝的アルゴリズム⁽¹⁵⁾ は生物の遺伝子が環境に適応して変化(進化)していくことをモデル化したものであり、多峰性を有する最適化問題などへの適用が研究されている。このアルゴリズムでは個体の集団の適応的な変化により最適化を行うため、複数の比較的良い解が得られることが特徴である。ここではこのアルゴリズムをあとで示されるこう配法による局所探索で必要な比較的良い複数の初期実行可能解を得るための一つの方法として用いることにする。この場合の初期実行可能解とは、障害物を回避する空間経路を意味するものとする。ただし、本論文ではこのアルゴリズムを厳密な最適軌道を求めるために用いているわけではない。

4.2 個体の表現方法と適応度の計算 遺伝的アルゴリズムにおいて探索空間における変数は、遺伝子の列として表現されなければならない。これはコード化と呼ばれているが、このコード化が適切でないと遺伝的アルゴリズムにおいて遺伝オペレータによる最適化がうまくなされないことになる。したがって、どのようにマニピュレータの空間経路をコード化するかと

Fig. 1 Representation of phenotype and genotype for spatial paths $\theta(s)$ by B-spline control points

いうことは重要な問題である。ここでは以下のように経路を表現することにする。

まず経路の始端と終端の関節変位 θ_s, θ_G は、あらかじめ指定されているものとする。そして、その間に m 個の B スプラインの制御点 $\hat{\theta}_1, \dots, \hat{\theta}_m$ がある個体 j に対して与えられているとき、それらをその個体の遺伝子とし実数型で定義する。次に遺伝子列 $j(\theta_s, \hat{\theta}_1, \dots, \hat{\theta}_m, \theta_G)$ を、位数 4 の B スプラインを用いて補間し $\theta(s)$ を求める。こうすることである一つの経路を一つの個体に対応づけることができる。このコード化により交叉、突然変異等の遺伝的オペレータの物理的意味が明確になるという利点があるが、同時に局所的探索能力が劣るという問題が生じることになる。しかし、この欠点はあとで示されるこう配法による局所探索で補われる。図 1 に空間経路の表現型と遺伝型による空間経路の表現方法を示す。

第 3 章で述べたように、マニピュレータの空間経路が個体の遺伝子として B スプラインの制御点により表現されたなら、次のようにその個体に対する適応度 F は空間経路に対する最短動作時間 $T_{\min}$ と関連づけるのが適当である。

$$F = -T_{\min} \quad \dots\dots\dots (15)$$

4.3 個体の初期集団の求め方 遺伝的アルゴリズムにおいて初期集団の選び方は最終的に得られる集団(解集合)に影響を及ぼすため重要である。初期解集団は障害物と干渉する衝突経路も含めて多様な空間経路で構成されていることが望ましい。しかし、同時にそれらの解の良さもある程度は考慮する必要がある。そこでここでは図 2 に示すようにして M_p 個の初期個体(初期経路)を作る。まず関節空間において初期点と終点を直線で結ぶ。次に仮の B スプライン制御点 $(\hat{\theta}_1, \dots, \hat{\theta}_m)$ をその直線上に等間隔に割り当てる。その仮の各制御点に対して、平均 $\hat{\theta}_i$ 、分散 σ_i^2 による n 次独立な正規分布を考え、その分布に従う M_p 個の制御点の

Fig. 2 Initial group of individuals generated by linear approximation and normal distribution

組を生成する。この正規分布は次式で表される。

$$h(\hat{\theta}_i^1, \dots, \hat{\theta}_i^n) = \frac{1}{\sqrt{2\pi}\sigma_i^1} \exp\left[-\frac{(\hat{\theta}_i^1 - \hat{\theta}_i^1)^2}{2\sigma_i^1}\right] \times \dots \\ \dots \times \frac{1}{\sqrt{2\pi}\sigma_i^n} \exp\left[-\frac{(\hat{\theta}_i^n - \hat{\theta}_i^n)^2}{2\sigma_i^n}\right] \\ (i=1, \dots, m) \dots \dots \dots (16)$$

M_p 個の制御点の組により M_p 個の個体(初期経路)が式(9)～(12)により生成できる。これらは、障害物と干渉する経路や曲がりくねった現実的でない経路も含んでいるかもしれない。しかし遺伝的アルゴリズムの性質を考えるとそれらの経路も初期集団には必要であり、上述の初期個体群の生成法はその意味から適切であると考えられる。

4.4 遺伝的オペレータ 遺伝的オペレータは遺伝的アルゴリズムの本質的部分であり、種々のオペレータが提案されている。アルゴリズムで利用されるオペレータは、適用する問題との関連や意味付けを十分に考えて選択されなければならない。ここでは基本的には単純遺伝的アルゴリズム(Simple Genetic Algorithm)を用いる。これは遺伝的オペレータとして、突然変異(Mutation)と交叉(Crossover)を用いている。突然変異のオペレータは、より多様な個体(経路)を得るために必要な基本的なオペレータであると考えられる。このオペレータを用いることで、探索空間内においてより広い、またより細かい解(経路)の探索が可能となる。ここでは、突然変異オペレータは具体的には図3(a)に示すように各制御点を乱数により選び、その点に対して式(16)に従って新たな制御点を生成することにする。

一方、交叉のオペレータは遺伝的アルゴリズムの特

徴ともなるより本質的な操作である。このオペレータにより、前世代の個体の良い部分同士が混ざり合っ、次の世代の個体が生成されることが可能となる。ここで検討する問題においては、空間経路をBスプラインの制御点列により表現している。したがって一個体の制御点列を素直に一本の遺伝子とみなせば、遺伝子の間隔は二つの制御点 $\hat{\theta}_i$, $\hat{\theta}_{i+1}$ の間とみなすことができる。実際、このようにして交叉オペレータを作用させれば図3(b)に示すように二つの経路の良い部分をつなぎ合わせて新しい、より良い空間経路を生成することが可能となる。ここでは確率的に選ばれた二つの個体(k, l)の制御点の間隔 $\hat{\theta}_i$ と $\hat{\theta}_{i+1}$ の間と、 $\hat{\theta}_i$ と $\hat{\theta}_{i+1}$ の間で交叉を行うことにする。

4.5 次世代への個体の選別 突然変異や交叉の遺伝的オペレータで新たに生成された空間経路はMTTPPPのアルゴリズムを適用することで、それらの経路に対応する最短動作時間 $T_{\min}$ が計算される。同時に障害物との干渉がチェックされ、もしそれが衝突経路であれば $T_{\min}$ として無限大の値が設定される。個体の適応度 F は、式(7)により最短動作時間 $T_{\min}$ と関連づけられる。ここでは次世代へ残す個体の数を一定の M_p 個とする。この M_p 個の個体を選別するため、まず $M_p - m_p$ 個を適応度の高い順から選ぶ。次に比較的少数の m_p 個は衝突経路も含めて、“悪い”経路からランダムに選ぶことにする。この m_p 個の個体は個体群の多様性を維持していくために必要となるものであり、個体群からなる集団が唯一の解(ある一つの局所最小解)に収束することを防ぐ働きをする。

5. 完全セル分割法による大域的な初期解の探索

こう配法の初期解としての複数の空間経路を得るもう一つの方法として、ここでは完全セル分割法⁽¹⁾を用いる。この方法によれば配置空間内の自由空間の幾何学的構造に対応した複数の空間経路を得ることができる。障害物回避のための幾何学的な空間経路生成法は他にも多くあるが、例えばポテンシャルに基づく多くの方法は一つの解しか得ることができないため、本論文で提案するこう配法に基づく方法でその複数の初期解を得る方法としては利用できない。

完全セル分割法で得られる空間経路は、配置空間内の折れ線として表現される。このままではなめらかな時間軌道が得られないため、Bスプラインの制御点をこの折れ線上に配置して、なめらかな空間経路を再生成する。元の折れ線とBスプライン曲線との差を少なくするために、折れ線の接点上に制御点をおくことが

Fig. 3 Mutation operator using a normal distribution and crossover operator using control points of B-spline

(a) Cell decomposition by convex polygons (cells) (b) A sequence by convex polygons and a polygonal line

(c) Generation of connectivity graph (d) B spline curves are generated by control points which are placed on polygonal lines

Fig. 4 Searching of global initial feasible paths by exact cell decomposition method

望ましいが、制御点が多い場合や接点が局所的に多く存在する場合は接点以外の折れ線上にもほぼ等間隔になるように制御点を配置することにする。もし再生成された B スプラインによる空間経路が障害物と干渉する経路になるならば、干渉点近傍の折れ線上の制御点の数を増やして干渉しないようにする。

以下に完全セル分割法を用いて、複数の大域的な解を求める手順の概略を示す。

(1) 問題の配置空間を求め、障害物との干渉領域を多角形で近似する。

(2) (1) で求めた多角形の頂点から、自身の多角形頂点を含まない最も近い多角形の頂点を結ぶ。図 4 (a) に示すように、この操作を自由空間が凸多角形に分割されるまで繰り返す。この凸多角形をセルと呼ぶ。

(3) 初期位置が含まれるセルから目標位置が含まれるセルまでのセルのシーケンスをグラフ探索により求める。得られたシーケンスにおいて各セルの中点を通るように折れ線経路を決定する [図 4 (b), (c) 参照]。

(4) 図 4 (d) のように折れ線経路上に B スプラインの制御点を配置し、空間経路を再生成し MTTPPP により $T_{\min}$ を求める。

6. こう配法による局所探索

本論文では 4 章、5 章で示した大域的な方法により最終的な厳密な最適解を探索するという方法とはって

Table 1 Link parameters of a two DOF manipulator

i	$s_i(\text{m})$	$l_i(\text{m})$	$m_i(\text{Kg})$	$I_i(\text{Kg}\cdot\text{m}^2)$
1	0.1	0.2	1.0	0.00333
2	0.1	0.2	1.0	0.00333

いない。これは何らかの方法で評価の微分情報が得られるなら、離散化して探索するよりはこう配法を適用する方が効率的で、なめらかな軌道が得られると考えられるからである。しかし、前述のように多峰性を有する問題に対してこう配法をそのまま適用すると、局所最小解に陥りやすく、初期解によって最終的な解が大きく異なるという問題がある。そこで 4 章、5 章いづれかの方法により得られた、大域的にある程度良い複数の初期経路をもとに局所的に最適化することにより、一つだけの局所最小解に陥ることを防ぐことにする。本論文では、空間経路を B スプラインの制御点の組により表現している。したがってこう配法における探索変数は、制御点とすることが適当であると考えられる。このこう配法による軌道改善の基本的な更新式は

$$\hat{\theta}_{\text{step}i+1} = \hat{\theta}_{\text{step}i} - k \left(\frac{\partial F}{\partial \hat{\theta}} \right)_i \dots\dots\dots (17)$$

と表せる。ここで k は適当な正の小さな値である。この配ベクトル $\partial F / \partial \hat{\theta}$ は MTTPPP のアルゴリズムを用いて数値的に微分して計算する。式 (17) により改善された空間経路は、収束の最終的な段階での経路に対してのみ障害物との干渉チェックを行う。もしその経路が干渉する経路ならば収束の一つ前の step の空間経路に対して干渉チェックを行う。これを繰り返して、干渉しない経路が見つければそれを最終的な軌道とする。初期経路は実行可能経路なので、最悪初期経路まで戻れば必ず干渉しない軌道は得られる。一般に干渉チェックは、マニピュレータリンクと障害物の表現において適切な近似を行わなければ、かなりの計算量を要することになる。ここでは上記のような方法により、干渉チェックの回数を減らしている。

7. 二次元と三次元の例題

本例題では、2 リンク二自由度と 3 リンク三自由度のマニピュレータを考える。二次元の例題において障害物は中心 (0.3, 0.3), (-0.3, -0.3)m, 半径 0.2 m の二つの円に設定した。また、マニピュレータを線分近似することにより干渉チェックを行っている。ダイナミクス計算のためのリンクパラメータを表 1 に示す。表中で l_i は各リンクの長さ、 s_i は重心までの距離、 m_i

は各リンクの質量, I_i は各リンクの重心回りの慣性モーメントを表す。駆動力の制限条件と関節軌道の始端と終端における境界条件は,

$$-10.0 \leq u_1(t) \leq 10.0, -5.0 \leq u_2(t) \leq 5.0 \quad \cdots (18)$$

$$\theta_s(0) = (0, 0), \dot{\theta}_s(0) = 0$$

$$\theta_{G1}(s_e) = \left(\frac{2}{3}\pi, \frac{\pi}{2} \right), \dot{\theta}_{G1}(s_e) = 0$$

$$\theta_{G2}(s_e) = \left(-\frac{5}{6}\pi, -\frac{\pi}{2} \right), \dot{\theta}_{G2}(s_e) = 0$$

である。ここで, u_1, u_2 N・m は各関節の駆動力, $\theta_{G1}(s_e), \theta_{G2}(s_e)$ は終端における関節変位について二とおり考えることを示す。この例題の配置空間を図5に示す。ここでは配置空間の計算においては θ_1 を離散化し, θ_2 を連続表現することで衝突領域の写像の計算を行っている。図5において黒い部分が衝突領域に対応している。図6は完全セル分割法により構造化された配置空間と, 三とりの折れ線経路を示している。またそれらの経路上の点はなめらかな経路を再生成するために用いられる B スプラインの制御点である。遺伝的アルゴリズムにおいては, 各世代の個体数を 400, その中で悪い個体を残す個体数を 10, σ^2 を 0.2 rad とし, B スプラインの制御点数 10 とした。これらのパラメータは何回かの試行の後に各世代の集団としての適

Fig. 5 Configuration space of the example with two degrees of freedom

Fig. 6 The free space of configuration space is decomposed by convex polygons (cells)

応度の変化速度の大きい値を採用した。

図7にシミュレーション結果を示す。太線はマニピュレータの手先軌道を, 数字は最短動作時間を示す。遺伝的アルゴリズムにより得られる(A)の軌道は, 各世代の最良個体の適応度の値がほぼ変化しなくなった150世代での, 最も適応度が優れた解を示している。遺伝的アルゴリズムにより求められた, 複数の実行可能解を初期軌道としこう配法を適用したあとで得られた最も評価の良い軌道は(B)のようになる。シミュレーションを複数回試みたが, いずれも(B)に近い手先軌道であり, 最短時間としておおむね0.31 sを得た。したがって, 厳密な最適解は(B)に近い手先軌道であることが予想される。

一方, 完全セル分割法からは(C), (E), (G)の三とりの軌道が得られる。それぞれの最短動作時間は遺伝的アルゴリズムにより得られた最良個体のそれに比べ劣っている。これは完全セル分割法においては, 評価を考慮していないことによるものである。完全セル分割法にはこのような欠点があるが, こう配法により経

Fig. 7 Simulation results of a 2-DOF manipulator (Trajectories of manipulator's tip and minimum travelling time for the trajectories)

路が改善されると(D),(F),(H)のようなかなり良い解が得られる。特に(H)は、遺伝的アルゴリズムとこう配法を組み合わせることによって得られた(B)の軌道と非常に良く似ている。このことは逆に、遺伝的アルゴリズムとこう配法の組合せによる方法により得られた解は、大域的にかなり良い軌道であることも示している。図8は(A),(B),(G),(H)に対応する最適駆動トルクを示す。それぞれこう配法により駆動トルクをなめらかにしていることがわかる。この二次元の例題ではこう配法の複数の初期解を求めるために用いられた遺伝的アルゴリズムと完全セル分割法的能力はほとんど同じであった。

次に、より現実的な三次元の例題を取り扱う。三次

元以上の問題では配置空間を利用する完全セル分割法は探索のための計算量が増えてその適用は非常に困難になる。遺伝的アルゴリズムにおいても探索空間が広がることには変わらないが、こう配法の初期解のための大域的な複数の実行可能解の探索というためだけに計算量の点で有利であると期待できる。図9に三次元の例題に対して遺伝的アルゴリズムとこう配法を組み合わせる方法で100世代まで計算して得られたいくつかの解のうち評価の良いものを示す。図9(a)はこの例題のマニピュレータの構造を示している。また、図9(b)に障害物、マニピュレータと得られた手先経路を、その経路に対する最短動作時間の値とともに示している。ここでは干渉チェックのための幾何プリミ

Fig. 8 Time optimum driving torques by the MTTPPP algorithm

Fig. 9 Simulation results of a 3 DOF manipulator

Table 2 Link parameters of a three DOF manipulator

i	$s_i(\text{m})$	$l_i(\text{m})$	$m_i(\text{Kg})$	$I_i(\text{Kg}\cdot\text{m}^2)$
1	0.1	0.2	1.0	0.00333
2	0.1	0.2	1.0	0.00333
3	0.1	0.2	1.0	0.00333

ティプはマニピュレータを線分、障害物を角柱としている。遺伝的アルゴリズムのための各種パラメータは各世代の個体数を150としてその他は二次元の例題とほぼ同じ値を採用した。ただし、世代の若い段階では実行可能解は非常に少なかったため、実行可能解以外をすべて悪い解として次の世代に残して計算を進めた。この例題での動力学計算のためのリンクパラメータを表2に示す。駆動力の制限条件と関節軌道の始端と終端における境界条件は以下のとおりである。

$$\left. \begin{aligned} -15.0 \leq u_1 \leq 15.0, -10.0 \leq u_2 \leq 10.0 \\ -5.0 \leq u_3 \leq 5.0 \end{aligned} \right\} \dots\dots (20)$$

$$\left. \begin{aligned} \theta_s(0) &= \left(0, \frac{\pi}{4}, \frac{\pi}{6}\right), \dot{\theta}_s(0) = 0 \\ \theta_c(s_e) &= \left(\frac{3}{7}\pi, \frac{3}{4}\pi, -\frac{2}{7}\pi\right), \dot{\theta}_c(s_e) = 0 \end{aligned} \right\} \dots (21)$$

本例題では最終姿勢を一つに限定しているが、それでも多様な経路が求められていることが確認できる。なお、この例題(二次元、三次元)での計算時間は約20 MIPSのWS(Cによるプログラム)で、二次元の完全セル分割法とこう配法の組合せですべての解を得るのに約8分、遺伝的アルゴリズムとこう配法の組合せで最終的な解を得るのに約40分を要した。さらに三次元の例題(遺伝的アルゴリズムとこう配法の組合せ)では約70分の計算時間を要した。

8. あとがき

作業空間内に、障害物が複雑に存在するような空間での最短時間軌道を計画する手法を提案した。その手法は基本的にはこう配法であり、こう配法で用いる複数の比較的良好初期解を得る方法として、遺伝的アルゴリズムと完全セル分割法の二つの大域的な探索法を用いた。本論文で計算された二次元の例題からは、二つの方法による最終的な最良の解には大きな差は見られなかった。しかし、計算時間の点からは完全セル分割法の方が有利であることがわかった。一方、三次元

の例題においてこう配法のための初期解を得る方法としての完全セル分割法では、一つの実行可能解を得るためだけでも配置空間の作成に多大な計算時間が必要であり、また探索域も拡大するためにその適用は困難になる。しかし、実行可能ないくつかの大域的な初期解を得るためだけであれば、遺伝的アルゴリズムは有効であり、こう配法との組合せにより三次元の問題にも適用できることが例題により示された。より多次元の例題に対してもこの方法は適用可能と思われるが、この場合は遺伝的アルゴリズムの個体集団を、計算時間の許す限りある程度大きくしておかないと、探索の中で実行可能解(衝突しない経路)が一つも得られない可能性がある。この問題は確率的な手法である遺伝的アルゴリズムを適用する上での問題点として残されている。

文 献

- (1) Latombe, J. C., *Robot Motion Planning*, (1991), 200-245, Kluwer Academic Publishers.
- (2) Hwang, Y. K. and Ahuja, N., *ACM Computing Surveys*, **24**-3 (1992), 219.
- (3) Dubowsky, S. and Shiller, Z., *Proc. of RoManSy, 5th CISM-IF ToMM Symp.*, (1984), 133-143.
- (4) Shin, K. G. and McKay, N. D., *IEEE Trans.*, **AC 30** 6 (1985), 531-541.
- (5) 尾崎弘明・山本元司・毛利 彰, 計測自動制御学会論文集, **23**-3 (1987), 288-293.
- (6) Kahn, M. E. and Roth, B., *Trans. of the ASME DSCM*, **93**-3 (1971), 164-172.
- (7) Geering, H. P., Guzzella, L., Hepner, S. A. R. and Onder, C. H., *IEEE Trans.*, **AC 31** 6 (1986), 512-518.
- (8) Yamamoto, M. and Mohri, A., *Robotica*, **7**-1 (1989), 43-47.
- (9) 山本元司・井上卓見・毛利 彰, 機論, **55** 516, C(1989), 2077-2082.
- (10) Rajan, V. T., *Proc. of IEEE Int. Conf. on Robotics and Automation*, (1985), 759-764.
- (11) Sahar, G. and Hollerbach, J. M., *Proc. of IEEE Int. Conf. on Robotics and Automation*, (1985), 751-758.
- (12) 山本元司・尾崎弘明・毛利 彰, 計測自動制御学会論文集, **24**-9 (1988), 942-947.
- (13) Shiller, Z. R. and Dubowsky, S., *IEEE Trans. on Robotics and Automation*, **7** 6 (1991), 785-797.
- (14) 山本元司・圓島信也・毛利 彰, 機論, **56** 527, C(1990), 1881-1886.
- (15) Goldberg, D. E., *Genetic Algorithm in Search, Optimization, and Machine Learning*, (1989), 59-89, Addison Wesley.
- (16) Rogers, D. F. and Adams, J. A., コンピュータグラフィックス, (1979), 161-172, 日刊工業新聞社.