空間経路が指定されるマニピュレータ動作の軌道計画

尾崎 弘明*・山本 元司*・毛利 彰*

Planning of Joint Trajectories of Manipulators with Geometric Path Constraints

Hiroaki OZAKI*, Motoji YAMAMOTO* and Akira MOHRI*

This paper discusses how to plan manipulator joint trajectories in order to make a performance index as small as possible under geometric path constraints in consideration of physical constraints based on kinematics and dynamics of a manipulator system. Using a time scale factor $\kappa(t)$, the specified index and joint trajectories are described by functions of $\kappa(t)$ and its time derivative $\dot{\kappa}(t)$. Then the desirable $\kappa(t)$ is obtained by two proposed methods: the first one is the iteratively improving (II) method which is a feasible method and utilizes the local controllability of B spline; and the second one is a global optimization method using dynamic programming (DP). These methods are applied to an example of a collision-free path of the manipulator with four links and four degrees of freedom. The numerical results point out that II method is much more effective than DP method in view point of the computation time and used memories, if the strictly optimal solution is not required.

Key Words: manipulator, trajectory planning, time scale, optimal trajectory, collision-free

1. ま え が き

マニピュレータの空間経路が与えられる場合について、動力学などの物理的な拘束を考慮して、与えられた評価関数の値を小さくするように関節の時間軌道を計画する問題を考える。

この問題に関連して、Dubowsky ら¹⁾ はベース座標系で手先の空間経路が与えられる場合、また Shin ら²⁾ は関節座標系で空間経路が与えられる場合について、

それらの経路に沿う移動量 λ を用いてマニピュレータの運動を記述することを考えた。このとき、経路に沿う運動はパラメータ λ とその時間微分 $\dot{\lambda}$ が表わす λ - $\dot{\lambda}$ 位相面上の軌跡となるので、簡単なアルゴリズムで最短時間軌道を計画できる。しかし、これは評価関数の特殊性を利用するものであり、評価関数が動作時間でない一般の場合には、このような簡単なアルゴリズムは得られていない。そこで、Vukobratović ら³⁾ は λ を離散化し、動的計画法（以下 DP と記す）を用いて解くことを提案している。

これに対して筆者ら⁴⁾ は、空間経路を実現する一組の関節軌道と時間スケールファクタを用いて、動作時間を逐次短縮する方法や最短時間を直接求める方法を提案してきた。ここでは、この考えを発展させ、一般の評価関数の場合にも有効な二つの方法、すなわち評価を逐次改善する方法と DP を用いる方法を提案する。前者は実行可能解法であり、途中で得られた解がすべての制約条件を満足する実行可能解を与えるという長所をもつ。また後者では、物理的拘束条件が時間スケールファクタおよびその微分に対する拘束となるので、DP を用いる場合に考慮しなければならない探索域が制限される(注1)。このため Vukobratović ら³⁾ の方法に比べ最適化計算に要する演算は少なくなる。以下、次章では問題の定式化を行い、第3、4章においてそれぞれのアルゴリズムを説明する。また、第5章では4自由度4リンクのマニピュレータの障害物回避経路に対して両者のアルゴリズムを適用し、その結果を比較検討する。

* 九州大学工学部 福岡市東区箱崎 6-10-1

* Faculty of Engineering, Kyushu University, Fukuoka

(Received July 17, 1986)

(Revised November 26, 1986)

(注1) 最近、Shin ら⁵⁾ によって λ - $\dot{\lambda}$ 位相面の考えをもとに DP を用いて最適軌道を計画する方法が発表された。この方法は、ここでの時間スケールファクタを用いるアルゴリズムと同様な形になっているが、第4章に示すような探索域を制限する方法をとっていない。

2. 問題の定式化

2.1 マニピュレータ系の動力学

マニピュレータの運動方程式はつぎのように表わせる。

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

ここに、 θ は m 次元関節変数ベクトル、 $H(\theta)$ はマニピュレータ機構部および減速器を含めたアクチュエータ部の慣性モーメントを表わす $m \times m$ 慣性マトリクス、 $h(\theta, \dot{\theta})$ は遠心力、コリオリ力を表わす m 次元ベクトル、 $g(\theta)$ は m 次元重力ベクトル、 D はマニピュレータ機構部と減速器の摩擦やアクチュエータの制動特性を表わす $m \times m$ 対角マトリクス、 u は m 次元アクチュエータ駆動ベクトルである。なお、 m は関節の自由度を表わすものとする。

2.2 時間スケール変更による動力学表現

指定された空間経路を実現する一組の関節軌道 $\theta(t)$ (これを基準軌道と呼ぶ) に対して、空間経路を変えることなしに時間軸を伸縮すること (これを時間スケール変更と呼ぶ) を考える。すなわち、基準軌道 $\theta(t)$ の時刻 t における微小時間 dt 間の変動量を $d\theta$ とし、これを変えることなく時間 dt を $\kappa(t)$ 倍して得られる時間を $d\tilde{t}$ とする。

$$d\tilde{t} = \kappa(t) dt \quad (2)$$

ここで、 $\kappa(t)$ を時間スケールファクタと呼び、時間スケール変更によって得られた軌道を修正軌道と呼ぶことにする。このとき、修正軌道の時刻 $\tilde{t}$ 、位置 $\tilde{\theta}(\tilde{t})$ 、速度 $\dot{\tilde{\theta}}(\tilde{t})$ 、加速度 $\ddot{\tilde{\theta}}(\tilde{t})$ およびアクチュエータ駆動力ベクトル $\tilde{u}(\tilde{t})$ は基準軌道での対応する物理量と時間スケールファクタを用いてつぎのように表現される⁴⁾。

$$\tilde{t} = \int_0^t \kappa(\tau) d\tau \quad (3)$$

$$\tilde{\theta}(\tilde{t}) = \theta(t) \quad (4)$$

$$\dot{\tilde{\theta}}(\tilde{t}) = \frac{1}{\kappa(t)} \dot{\theta}(t) \quad (5)$$

$$\ddot{\tilde{\theta}}(\tilde{t}) = \frac{1}{\kappa^2(t)} \left\{ -\frac{\dot{\kappa}(t)}{\kappa(t)} \dot{\theta}(t) + \ddot{\theta}(t) \right\} \quad (6)$$

$$\begin{aligned} \tilde{u}(\tilde{t}) = & \frac{1}{\kappa^2(t)} u(t) - \frac{\dot{\kappa}(t)}{\kappa^3(t)} H(\theta) \dot{\theta} \\ & + \frac{\kappa^2(t)-1}{\kappa^2(t)} g(\theta) + \frac{\kappa(t)-1}{\kappa^2(t)} D\dot{\theta} \end{aligned} \quad (7)$$

2.3 軌道計画問題

関節速度やアクチュエータ駆動力が制限されるとして以下の軌道計画を考える。修正軌道の終端時刻を $\tilde{t}_e$ (対応する基準軌道の時刻は t_e) とし、評価関数はつぎのように与えられるものとする。

$$J = \int_0^{\tilde{t}_e} F(\tilde{\theta}(\tilde{t}), \dot{\tilde{\theta}}(\tilde{t}), \tilde{u}(\tilde{t})) d\tilde{t} \quad (8)$$

(8)式に(2)～(7)式の関係を代入するとつぎようになる。

$$J = \int_0^{t_e} L(\kappa(t), \dot{\kappa}(t)) dt \quad (9)$$

修正軌道に対して、関節速度 $\dot{\tilde{\theta}}(\tilde{t})$ やアクチュエータ駆動力 $\tilde{u}(\tilde{t})$ はつぎのように制約を受けるものとする。

$$\dot{\tilde{\theta}}_{i \min} \leq \dot{\tilde{\theta}}_i(\tilde{t}) \leq \dot{\tilde{\theta}}_{i \max} \quad (i=1, \dots, m) \quad (10)$$

$$u_{i \min} \leq \tilde{u}_i(\tilde{t}) \leq u_{i \max} \quad (i=1, \dots, m) \quad (11)$$

ここに、 $\dot{\tilde{\theta}}_{i \min}$, $\dot{\tilde{\theta}}_{i \max}$, $u_{i \min}$, $u_{i \max}$ はそれぞれ i 番目の関節速度およびアクチュエータ駆動力の上下限を表わすものとする。軌道の始端と終端の境界条件はつぎのように与えられるものとする。

$$\begin{aligned} \tilde{\theta}(0) &= \theta_0, \quad \dot{\tilde{\theta}}(0) = \dot{\theta}_0, \quad \ddot{\tilde{\theta}}(0) = \ddot{\theta}_0 \\ \tilde{\theta}(\tilde{t}_e) &= \theta_e, \quad \dot{\tilde{\theta}}(\tilde{t}_e) = \dot{\theta}_e, \quad \ddot{\tilde{\theta}}(\tilde{t}_e) = \ddot{\theta}_e \end{aligned} \quad (12)$$

ただし、第3および第4章のアルゴリズムでは端点で速度または加速度が指定されない場合も同様に扱える。

以上より軌道計画の問題は、制約条件(10)～(12)式を満たし、(9)式の評価関数を小さくするように $\kappa(t)$ および $\dot{\kappa}(t)$ を求める問題になる。

3. 逐次改善による解法

制約条件を満足する実行可能な解を逐次更新しながら、評価関数(9)式の値を小さくする方法を考える。提案する方法は実行可能解法であるから、まず制約条件を満たす初期実行可能軌道を求めておかなければならない。しかし、動作時間を十分長くとることにより、この軌道は容易に得られる。これを基準軌道とすれば、この軌道は $\kappa(t) = 1.0$ ($0 \leq t \leq t_e$) で表わされる。

いま、基準軌道の動作時間 t_e を等間隔 Δt をもつ n 個の区間に分け、その区分点の時刻を $t_0 (=0)$, t_1 , ..., t_i , ..., $t_n (=t_e)$ とする。時刻 t_i における $\kappa(t_i)$ の仮想目標値を $\hat{\kappa}_i$ とし、これを制御点としてBスプラインを用いて $\kappa(t)$ の軌道を表わすことにする。そこで、 $\kappa(t)$ の2階の導関数まで連続になるようにすれば、 $t_i \leq t \leq t_{i+1}$ における時刻 t と $\kappa(t)$ はパラメータ s ($0 \leq s \leq 1$) を用いてつぎのように得られる⁶⁾。

$$t(s) = t_i + s \Delta t \quad (13)$$

$$\begin{aligned} \kappa(s) = & \hat{\kappa}_{i-1} N_{i-1,4}(s) + \hat{\kappa}_i N_{i,4}(s) \\ & + \hat{\kappa}_{i+1} N_{i+1,4}(s) + \hat{\kappa}_{i+2} N_{i+2,4}(s) \end{aligned} \quad (14)$$

ただし、

$$N_{i-1,4}(s) = (-s^3 + 3s^2 - 3s + 1)/6$$

$$N_{i,4}(s) = (3s^3 - 6s^2 + 4)/6$$

$$N_{i+1,4}(s) = (-3s^3 + 3s^2 + 3s + 1)/6$$

$$N_{i+2,4}(s) = s^3/6 \quad (15)$$

また, $\hat{\kappa}(t)$ は (13), (14) 式よりつぎようになる。

$$\begin{aligned} \dot{\kappa}(t) = & \frac{1}{\Delta t} \left\{ \hat{\kappa}_{i-1} \frac{dN_{i-1,4}}{ds} + \hat{\kappa}_i \frac{dN_{i,4}}{ds} \right. \\ & \left. + \hat{\kappa}_{i+1} \frac{dN_{i+1,4}}{ds} + \hat{\kappa}_{i+2} \frac{dN_{i+2,4}}{ds} \right\} \quad (16) \end{aligned}$$

$\hat{\kappa}_i$ を変更したときの $\kappa(t)$ の変化は (13) ~ (15) 式より $t_{i-2} < t < t_{i+2}$ に限られ (B スプラインの局所制御性), この区間の評価は次式で計算される。

$$J_i = \int_{t_{i-2}}^{t_{i+2}} L(\kappa(t), \dot{\kappa}(t)) dt \quad (17)$$

以上を考慮すると, 軌道の逐次改善手順はつぎのようになる。

- (i) 基準軌道 $\theta(t)$ を求め, $\kappa(t) = 1.0$ ($0 \leq t \leq t_e$) とする。
- (ii) 分割数 n , 判定のしきい値 $\varepsilon (> 0)$, 定数 $c (> 1)$, $\delta\kappa_{\min}$ および $\delta\kappa_0$ を与える。
- (iii) (9) 式によって J を計算する。
- (iv) $i=1$, $J_{\text{old}}=J$, $\hat{\kappa}_j=1.0$ ($j=-1, \dots, n+1$) とおく。
- (v) $i=i+1$, $\delta\kappa=\delta\kappa_0$ とおく。
- (vi) $i < n-1$ なら (vii) へ, $i=n-1$ なら (9) 式の J を計算し, $J_{\text{old}} - J \geq \varepsilon$ なら $J_{\text{old}}=J$, $i=1$ として (v) へ, $J_{\text{old}} - J < \varepsilon$ なら計算終了。
- (vii) $\hat{\kappa}_i$ を $\hat{\kappa}_i \pm \delta\kappa$ とし, $t_{i-2} < t < t_{i+2}$ の区間で以下の計算を行う。(13) ~ (16) 式によって $\kappa(t)$, $\dot{\kappa}(t)$ を計算する。この結果を用いて (5), (7) 式を計算し, 制約条件 (10), (11) 式をチェックする。これらの条件を満たせば (viii) へ, 満たさなければ (ix) へ。

(viii) $\hat{\kappa}_i$, $\hat{\kappa}_i + \delta\kappa$, $\hat{\kappa}_i - \delta\kappa$ に応じて (17) 式の評価を計算し, それぞれ J_{i0} , J_{i+} , J_{i-} とする。これらを比較して J_{i+} が最小のとき $\hat{\kappa}_i = \hat{\kappa}_i + \delta\kappa$, J_{i-} が最小のとき $\hat{\kappa}_i = \hat{\kappa}_i - \delta\kappa$ として $\hat{\kappa}_i$ を更新し (vii) へ戻る。 J_{i0} が最小のとき (ix) へ。

(ix) $\delta\kappa = \delta\kappa/c$ とし $\delta\kappa > \delta\kappa_{\min}$ なら (vii) へ, $\delta\kappa \leq \delta\kappa_{\min}$ なら (v) へ。

以上のアルゴリズムは両端の速度, 加速度を基準軌道と等しくした場合である。これ以外の場合には, 条件に応じて $i = -1, 0, 1$ および $n-1, n, n+1$ における $\hat{\kappa}_i$ の値を変更すればよい。

4. 動的計画法を用いた解法

DP を用いて評価関数 (9) 式の値を最小にするように軌道を計画する方法を示す。前章と同様に t_e を n

個の区間に分割した場合を考える。いま $t_i \leq t \leq t_{i+1}$ において $\kappa(t)$ の軌道が決めると, その区間に対する評価関数の値は (9) 式よりつぎようになる。

$$c(\kappa(t_i), \kappa(t_{i+1})) = \int_{t_i}^{t_{i+1}} L(\kappa(t), \dot{\kappa}(t)) dt \quad (18)$$

この表現を用いると, Bellman の最適性の原理⁷⁾ より次式が成り立つ。

$$f_i(\kappa(0), \kappa(t_i)) = c(\kappa(0), \kappa(t_i)) \quad (19)$$

$$\begin{aligned} f_{i+1}(\kappa(0), \kappa(t_{i+1})) = & \min_{\kappa(t_i)} [f_i(\kappa(0), \kappa(t_i)) \\ & + c(\kappa(t_i), \kappa(t_{i+1}))] \quad (20) \end{aligned}$$

ここに, $f_i(\kappa(0), \kappa(t_i))$ は与えられた $\kappa(0)$ から $\kappa(t_i)$ に到る (18) 式の評価の最小値である。

つぎに, (10), (11) 式の制約条件によって $\kappa(t)$ の軌道に課せられる条件について考えてみる。まず, (11) 式に (7) 式を代入すると次式を得る。

$$\begin{aligned} & \{-u_i \max + [\mathbf{g}(\theta)]_i\} \kappa^3(t) + [D\dot{\theta}]_i \kappa^2(t) \\ & + [\mathbf{u} - \mathbf{g}(\theta) - D\dot{\theta}]_i \kappa(t) \leq [H(\theta)\dot{\theta}]_i \dot{\kappa}(t) \\ & \leq \{-u_i \min + [\mathbf{g}(\theta)]_i\} \kappa^3(t) + [D\dot{\theta}]_i \kappa^2(t) \\ & + [\mathbf{u} - \mathbf{g}(\theta) - D\dot{\theta}]_i \kappa(t) \quad (21) \end{aligned}$$

ここに, $[\mathbf{a}]_i$ はベクトル $\mathbf{a}$ の第 i 成分をとるものとする。簡単のために $Q_i \equiv [\mathbf{g}(\theta)]_i$, $R_i \equiv [D\dot{\theta}]_i$, $S_i \equiv [\mathbf{u} - \mathbf{g}(\theta) - D\dot{\theta}]_i$, $M_i \equiv [H(\theta)\dot{\theta}]_i$ とおくと, (21) 式は次式となる。

$$\begin{aligned} & \frac{Q_i \kappa^3(t) + R_i \kappa^2(t) + S_i \kappa(t) - \left[\begin{array}{l} u_i \max (M_i > 0) \\ u_i \min (M_i < 0) \end{array} \right] \kappa^3(t)}{M_i} \\ & \leq \dot{\kappa}(t) \\ & \leq \frac{Q_i \kappa^3(t) + R_i \kappa^2(t) + S_i \kappa(t) - \left[\begin{array}{l} u_i \min (M_i > 0) \\ u_i \max (M_i < 0) \end{array} \right] \kappa^3(t)}{M_i} \quad (22) \end{aligned}$$

ここに, $\left[\begin{array}{l} a (M_i > 0) \\ b (M_i < 0) \end{array} \right]$ は $M_i > 0$ のとき a を, $M_i < 0$ のとき b をとるものとする。(22) 式の $\dot{\kappa}(t)$ が存在するためには, 任意の $i, j (i \neq j)$ に対して次式が成立しなければならない。

$$\begin{aligned} & \left[\left(\frac{Q_i}{M_i} - \frac{Q_j}{M_j} \right) - \left[\frac{u_i \min (M_i > 0)}{u_i \max (M_i < 0)} \right] \right. \\ & \left. - \left[\frac{u_j \max (M_j > 0)}{u_j \min (M_j < 0)} \right] \right] \kappa^3(t) + \left[\frac{R_i}{M_i} - \frac{R_j}{M_j} \right] \kappa^2(t) \\ & + \left[\frac{S_i}{M_i} - \frac{S_j}{M_j} \right] \kappa(t) \geq 0 \quad (23) \end{aligned}$$

ここで, 左辺の $\kappa^3(t)$ の係数を A_{ij} , $\kappa^2(t)$ の係数を B_{ij} , $\kappa(t)$ の係数を C_{ij} とおき, $\kappa(t) > 0$ を考慮すると次式を得る。

$$A_{ij}\kappa^2(t) + B_{ij}\kappa(t) + C_{ij} \geq 0 \quad (24)$$

一方、(10)式に(5)式を代入すると次式となる。

$$\kappa(t)\dot{\theta}_{i\min} \leq \dot{\theta}_i(t) \leq \kappa(t)\dot{\theta}_{i\max} \quad (25)$$

したがって、(24)式の2次不等式または(25)式の1次不等式を満足しない $\kappa(t)$ は非許容域に属する。また、 $\dot{\kappa}(t)$ は(22)式より制約を受けるので、この非許容域も考え合せると、つぎのような許容範囲をもつ。

$$\dot{\kappa}_{\min}(t) \leq \dot{\kappa}(t) \leq \dot{\kappa}_{\max}(t) \quad (26)$$

これらの制約のもとに、(20)式によって最適な軌道を探索することになる。ただし、ここでは $\kappa(t_i)$ と $\kappa(t_{i+1})$ 間の $\kappa(t)$ の軌道を線形近似するものとする。すなわち、

$$\kappa(t) = \kappa(t_i) + \frac{\kappa(t_{i+1}) - \kappa(t_i)}{t_{i+1} - t_i} (t - t_i) \quad (27)$$

$$\dot{\kappa}(t) = \frac{\kappa(t_{i+1}) - \kappa(t_i)}{t_{i+1} - t_i} \quad (28)$$

以上より、DPを用いた解法の手順は以下のようになる。

- (i) 基準軌道 $\theta(t)$ を求める。
- (ii) t 方向のきざみ Δt と $\kappa(t)$ 方向のきざみ $\Delta \kappa$ を決め、 t - $\kappa(t)$ 面の格子を定める。初期値 $\kappa(t_0) = \kappa(0) = \kappa_0$ を与える。
- (iii) (26)式より κ_0 における $\dot{\kappa}_{\min}, \dot{\kappa}_{\max}$ を求める。 $\kappa(t_1)_{\min} = \kappa_0 + \dot{\kappa}_{\min} \Delta t$, $\kappa(t_1)_{\max} = \kappa_0 + \dot{\kappa}_{\max} \Delta t$ とし、 $\kappa(t_1)_{\min} \leq \kappa(t_1)_j \leq \kappa(t_1)_{\max}$ を満たす格子点 $\kappa(t_1)_j$ ($j = j_{\min}, \dots, j_{\max}$)を求める。 $\kappa(0)$ から $\kappa(t_1)_j$ に至る軌道の評価 $f(\kappa(0), \kappa(t_1)_j)$ を(19)式により計算し記憶する。 $i=1$ とおく。
- (iv) $j = j_{\min}, \dots, j_{\max}$ について以下の計算を行う。まず(26)式より $\kappa(t_i)_j$ における $\dot{\kappa}_{j\min}, \dot{\kappa}_{j\max}$ を求める。 $\kappa(t_{i+1})_{j\min} = \kappa(t_i)_j + \dot{\kappa}_{j\min} \Delta t$, $\kappa(t_{i+1})_{j\max} = \kappa(t_i)_j + \dot{\kappa}_{j\max} \Delta t$ とし、 $\kappa(t_{i+1})_{j\min} \leq \kappa(t_{i+1})_k \leq \kappa(t_{i+1})_{j\max}$ を満たす格子点 $\kappa(t_{i+1})_k$ ($k = k_{\min}, \dots, k_{\max}$)を求める。 $\kappa(t_i)_j$ から $\kappa(t_{i+1})_k$ に至る軌道の評価 $c_{jk} = c(\kappa(t_i)_j, \kappa(t_{i+1})_k)$ を(18)式を用いて計算し記憶する。
- (v) $k = k_{\min}, \dots, k_{\max}$ について、 κ_0 から $\kappa(t_{i+1})_k$ に至る最小の評価 $f_{i+1}(\kappa_0, \kappa(t_{i+1})_k)$ を(20)式により計算し、このときの軌道を記憶する。
- (vi) $\kappa(t_i)_k = \kappa(t_{i+1})_k$ ($k = k_{\min}, \dots, k_{\max}$)とする。
- (vii) $j = k, i = i+1$ とおき $t_i < t_n (= t_e)$ なら(iv)へ、 $t_i = t_n$ なら $\min_{\kappa(t_i)} \{f(\kappa_0, \kappa(t_i)_j)\}$ を求めて計算終了。このとき得られる軌道が $\kappa(0)$ を与えたときの最適軌道となる。

上記のアルゴリズムは終端自由の型であるが、終端の速度が指定される場合には $\kappa(t_n) = \kappa_n$ が指定される

ので、ステップ(iv)で $\kappa(t_n)_k \neq \kappa_n$ なる k に対して $c_{jk}(\kappa(t_{n-1})_j, \kappa(t_n)_k) = \infty$ とすればよい。さらに始端および終端で加速度も指定される場合には、 $\kappa(t_2)$ および $\kappa(t_{n-1})$ が指定されると考える。また両端自由の場合には、 κ_0 を変更して上記アルゴリズムを繰返して最適解を求めることになる。

5. 適用例

提案した二つの方法を、文献8)のアルゴリズムによって得られた4自由度4リンクのマニピュレータの障害物回避経路(Fig. 1 参照)に適用する。マニピュレータ各リンクは長さ20 cm, 断面積 $10 \times 10 \text{ cm}^2$, 質量1.0 kgの剛体とする。簡単のため、アクチュエータ部の特性と摩擦による影響は考えないことにする。このとき、(1)式において $H(\theta)$ はマニピュレータ機構部の慣性モーメント、 $D=0$, u は関節駆動トルクを表わすことになる。各関節の速度とトルクの制約条件はTable 1に示すように与えられるものとし、境界条件は $\dot{\theta}_0 = \dot{\theta}_e = 0$ とする。また評価関数はつぎの形とする。

$$J = \int_0^{t_e} \left(\lambda + \omega \sum_{i=1}^m |u_i| \right) d\tilde{t} \quad (29)$$

ここに、 λ と ω は重み係数である。初期実行可能解は文献9)を参考にして求めた。

Fig. 2は、重み係数 $\lambda=1.0, \omega=0$ の場合について、区間 $0 \leq t \leq t_e$ の分割数 $n=20$, 制約条件のチェ

Table 1 Joint velocity and torque constraints

	$\dot{\theta}_{i\max} = -\dot{\theta}_{i\min}$ (rad/s)	$u_{i\max} = -u_{i\min}$ (kgm ² /s ²)
Joint 1	1.74	32.0
Joint 2	2.62	18.0
Joint 3	3.49	8.2
Joint 4	4.36	2.0

Fig. 1 Collision-free spatial path of a manipulator

Fig. 2 Trajectories of $\kappa(t)$ obtained by proposed methods ($\lambda=1.0$, $\omega=0$)

Fig. 3 Trajectories of $\kappa(t)$ obtained by proposed methods ($\lambda=1.0$, $\omega=1.0$)

ック点数 200 として逐次改善による解法 [手法 (1)], DP を用いる解法で $t-\kappa(t)$ 面の格子数を 40×40 ($n=40$, $0 \leq \kappa(t) \leq 2.0$ の区間を 40 分割) [手法 (2-a)], 格子数を 100×150 ($n=100$, $0 \leq \kappa(t) \leq 1.5$ の区間を 150 分割) [手法 (2-b)] のそれぞれの方法で得られた $\kappa(t)$ の軌道を示す。また, この場合の評価は動作時間となることから, 数値積分の時間きざみを $0.0115 (=t_e/200)$ 秒として最短時間を求める方法⁴⁾ [手法 (3)] を適用した結果も付け加えている。なお, 図中の線影部は (24), (25) 式を満足しない非許容域を表わしている。この結果から, 格子点を増加した手法 (2-b) の解は最適解にきわめて近いこと, また手法 (1) によっても最短解をよく近似できることがわかる。ただし, $t=2.2 \sim 2.3$ 秒で手法 (2-a) の $\kappa(t)$ が手法 (3) のそれよりも小さいという矛盾が見られる。これは, 前者の制約条件のチェック点数が後者のそれより少ないため, 後者では許容できないのが前者の解となりうるからであろう。このことは, 同じ区間について格子点数を増加した手法 (2-b) の解が手法 (3) の解に近づくことから確かめられる。

Fig. 3 は, 重み係数を $\lambda=1.0$, $\omega=1.0$ として手法 (1), (2-a), (2-b) を適用した結果である。同図

Table 2 Comparison of proposed methods

Weighting factor	$\lambda=1.0, \omega=0$		$\lambda=1.0, \omega=1.0$	
	Performance J	CPU time (s)	Performance J	CPU time (s)
Reference trajectory	2.31	—	27.4	—
(1) Iteratively improving ($n=20$)	1.21	3.46	17.5	3.56
(2-a) DP (40×40 grids)	1.18	16.4	16.7	16.7
(2-b) DP (100×150 grids)	1.14	247	16.5	248
(3) Minimum time	1.13	0.268	—	—

Fig. 4 Reference trajectory and trajectory obtained by method (2-b) (Joint3; $\lambda=1.0$, $\omega=1.0$)

より, 手法 (1) は DP を用いる手法 (2-a), (2-b) に近い軌道を求めていることがわかる。Table 2 は, 以上の二つの評価関数に対して種々の手法を適用したときの評価値と演算時間をまとめたものである。また Fig. 4 は, $\lambda=1.0$, $\omega=1.0$ の場合に手法 (2-b) を適用して得られた結果の中から, 関節 3 の軌道に関するものを示している。

なお, 以上の適用例の計算には FACOM M380 S

を用いた。

6. あとがき

空間経路が指定される場合について、与えられた拘束条件のもとで、指定された評価関数の値を小さくするように関節軌道を計画する方法を検討した。

ここでは、評価関数が動作時間とは限らない一般的な場合について、時間スケールファクタ $\kappa(t)$ を用いて評価を改善する二つの方法を提案した。すなわち、一つはBスプラインを用いて評価を逐次改善する方法であり、もう一つは動的計画法を用いる方法である。前者はBスプラインの局所制御性を利用した実行可能解法であること、後者は、 $\kappa(t)$, $\dot{\kappa}(t)$ の制約を考慮して最適化計算に要する演算を少なくすることに特長がある。

これらの方法をマニピュレータの障害物回避経路に適用し、その有効性を比較した。その結果つぎのことが確認された。(1)逐次改善による解法では、演算時間が基準軌道の時間軸 t 方向の制約条件のチェック点数にほぼ比例する。これに対し動的計画法では、 t 方向と $\kappa(t)$ 方向の分割数をそれぞれ n , q とするとき、 t - $\kappa(t)$ 面の格子点数は $n \times q$ となり、必要な演算時間はほぼ $n \times q^2$ に比例する(ただし、 $\dot{\kappa}(t)$ の制約を考慮するとこれより少なくなる)。このため、後者によって解を求める場合は記憶容量や演算時間が多大になる。(2)逐次改善による解法によっても比較的良好な最

適解を近似できる。したがって、厳密な最適解を要求しなければ、逐次改善による解法は有効であると考えられる。

参考文献

- 1) S. Dubowsky and Z. Shiller: Optimum Dynamic Trajectories for Robotic Manipulators, Procs. of RoManSy, 5th CISM-IFTOMM Symp., 133/143 (1984)
- 2) K. G. Shin and N. D. McKay: Minimum-Time Control of Robotic Manipulators with Geometric Path Constraints, IEEE Trans. Automatic Control, **AC-30**-6, 531/541 (1985)
- 3) M. Vukobratović and M. Kirčanski: A Method for Optimal Synthesis of Manipulation Robot Trajectories, Trans of ASME, **G-104**-6, 188/193 (1982)
- 4) 尾崎, 秀田, 山本, 毛利: 空間経路が指定されるマニピュレータ動作の時間短縮, 第3回日本ロボット学会学術講演会予稿集, 377/380 (1985)
- 5) K. G. Shin and N. D. McKay: A Dynamic Programming Approach to Trajectory Planning of Robotic Manipulators, IEEE Trans. Automatic Control, **AC-31**-6, 491/500 (1986)
- 6) D. E. Rogers and J. A. Adams (山口 富士夫訳): コンピュータグラフィックス, 日刊工業新聞社, 161/172 (1980)
- 7) R. E. Bellman: Dynamic Programming, Princeton Univ. Press (1957)
- 8) 尾崎, 毛利, 高田: マニピュレータの占有空間を考慮した障害物回避動作の決定法, 計測自動制御学会論文集, **18**-9, 942/949 (1982)
- 9) C. S. Lin, P. R. Chang and J. Y. S. Luh: Formulation and Optimization of Cubic Polynomial Joint Trajectories for Industrial Robots, IEEE trans. Automatic Control, **AC-28**-12, 1066/1074 (1983)