

複数台マニピュレータの相互衝突回避に関する研究

田 口 幹*・河原崎 徳 之*

A Study of Mutual Collision Avoidance for Multiple Manipulators

Kan TAGUCHI and Noriyuki KAWARAZAKI

Abstract

This paper provides a new method of planning collision-free trajectories for multiple manipulators. If multiple manipulators move in a common workspace at the same time, collisions may occur. We can consider the manipulators to be time-varying obstacles. If one manipulator approach to another more closely, the input torque is suppressed by the distance sensitivity which give a criterion of collisions. The advantage of our method is that not any one manipulator has priority to the other manipulators. Each manipulator can avoid collision mutually corresponding to their situations. The effectiveness of proposed method is clarified by several experimental results.

Key Words: Multiple Manipulators, Path Planning, Collision-Free Trajectory, Lagrange's Formulation, Computed Torque Method, Distance Sensitivity

1. はじめに

マニピュレータの制御方法については、これまでに数多く研究されている。本研究では、1台のマニピュレータの単独制御ではなく、複数台のマニピュレータが動作し、それらが互いに衝突の可能性をもつ場合の問題を取り扱う。同一作業空間上で2台以上のマニピュレータを動作させる場合、お互いの動作領域が重なった場合に衝突の危険がある。つまり、あるマニピュレータから他のマニピュレータが動的障害物と見なせる。現在までのマニピュレータの障害物回避に関する研究は、作業空間上に静止した既知の障害物を対象としたものが主流である。しかし、ロボットによる高度で複雑な作業が要求されてくるにつれて、このようなマニピュレータ同士の衝突回避を含む軌道生成問題の解決は、マニピュレータの自律化に必要不可欠となってくる。人間が教示するとおりにしか行動できないマニピュレータの場合、人間にとって常識的なことまでマニピュレータに教示することになり、人間に多大な

負担を強いることになる。従って、ロボットを扱う人間の負担を軽くするためには、マニピュレータが基本的なことを自律的に判断して行動することが望ましい。複数マニピュレータの衝突回避問題は、近年盛んに研究されつつあるが、設定条件の違いにより異なった問題となる。多くの場合、経路計画とその経路を時間的にどのように配分するかという軌道計画は、分けて考えられている。また、いずれかのマニピュレータに優先度が付けられ、回避するマニピュレータが決められている。関節経路が与えられた2台のマニピュレータに動的計画法を用い非衝突最短時間軌道を得る方法¹⁾やニューラルネットワークを適用しエネルギー最小となる非衝突経路を得る方法²⁾等がある。また、コンフィギュレーション(マニピュレータの関節角度)空間に時間を考慮した回避方法も発表^{3,4)}されている。しかし、従来の多くの研究では、マニピュレータの動力学的な要素まで考慮されていない。そのため、回避軌道が生成されても実際には動作不可能な場合がある。マニピュレータにこのような回避アルゴリズムを用いる場合、動力学的要素を含みかつ計算が複雑でないほうがより実用的である。

そこで、本研究では動力学を考慮した複数台マニ

1994年9月16日受理

* 機械システム工学科

ピュレータの相互衝突回避問題⁵⁾を扱う。マニピュレータの動学的制御には、よく知られた計算トルク法⁶⁾を用いる。そして、2台のマニピュレータがある閾値以上に接近した場合、衝突の危険があると判断し、マニピュレータの入力トルクを修正する。その際、入力トルクはマニピュレータ間の最短距離の2乗に反比例させることで非衝突を保証する。また、特別にどちらかに優先度を付けることはせず距離感度というものを新たに定義し、衝突に大きな影響をもつ方を積極的に回避させる。これにより、状況に応じてマニピュレータ同士が相互に衝突を回避するようになる。最後に2台の水平型マニピュレータのシミュレーションにより、本手法の有効性を示す。

2. マニピュレータの動学的制御

2.1 マニピュレータモデル

ここで扱うのは、Fig. 1 に示すような2台の平面型2自由度マニピュレータ A, B である。この2台のマニピュレータに初期形状と目標形状を与え、互いに衝突せず目標に到る軌道を生成することを目的とする。Fig. 1 のパラメータを基にした平面型マニピュレータ A のラグランジュの運動方程式より、マニピュレータ A の2関節のトルク τ_1, τ_2 は

$$\tau_1 = M_{11}\ddot{\theta}_1 + M_{12}\ddot{\theta}_2 + h_{122}\dot{\theta}_2^2 + 2h_{112}\dot{\theta}_1\dot{\theta}_2 \quad (1)$$

$$\tau_2 = M_{21}\ddot{\theta}_1 + M_{22}\ddot{\theta}_2 + h_{211}\dot{\theta}_1^2 \quad (2)$$

となる。
ただし、

$$M_{11} = m_1 l_{g1}^2 + \tilde{I}_1 + m_2(l_1^2 + l_{g2}^2 + 2l_1 l_{g2} C_2) + \tilde{I}_2$$

$$M_{12} = M_{21} = m_2(l_{g2}^2 + l_1 l_{g2} C_2) + \tilde{I}_2$$

$$M_{22} = m_2 l_{g2}^2 + \tilde{I}_2$$

$$h_{122} = h_{112} = -h_{211} = -m_2 l_1 l_{g2} S_2$$

$$S_2 = \sin \theta_2, C_2 = \cos \theta_2$$

I_i : リンク i の重心回りの慣性モーメント

m_i : リンク i の質量

l_i : リンク i の長さ

l_{gi} : リンク i の付け根から重心までの長さ

である。

マニピュレータ B も同型であるので、同じ運動方程式となる。また、平面型マニピュレータであるため重力項は影響しないものとした。

2.2 計算トルク法

マニピュレータ実機に適用させるためには、軌道生成に際して動力学を考慮することが不可欠である。そこで、マニピュレータ制御でよく知られている計算トルク法を用いる。Fig. 2 に計算トルク法のブロック線図を示す。計算トルク法は、マニピュレータの各関節角度を状態変数とした制御方法であり、非線形フィードバック補償を設けることによって、制御対象とするシステムを線形システムとして取り扱うことができる。また、サーボ補償器を設けることによって、システムのモデル化誤差や外乱を補償することができる。目標とする各関節角度、角速度、角加速度を Fig. 3 のように与え、実際の関節角度、角速度をフィードバックすることで以下に示すサーボ補償器となる。そして、入力トルク τ は、(4) 式より求められる。

$$\ddot{\theta}_d^* = \ddot{\theta}_d + k_v(\dot{\theta}_d - \dot{\theta}) + k_p(\theta_d - \theta) \quad (3)$$

k_v, k_p : フィードバックゲイン

$$\tau = \hat{R}^{-1}(\theta, \dot{\theta}, \ddot{\theta}^*) \quad (4)$$

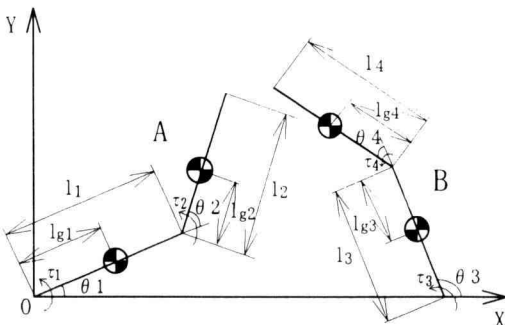
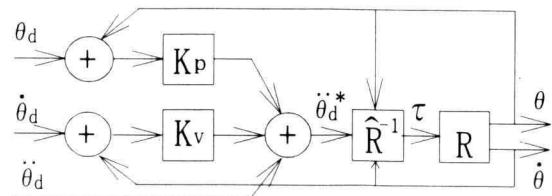


Fig. 1. Model of horizontal type manipulators



$\theta_d, \dot{\theta}_d, \ddot{\theta}_d$: specified value
 τ : input torque

Fig. 2. Block diagram of computed torque method

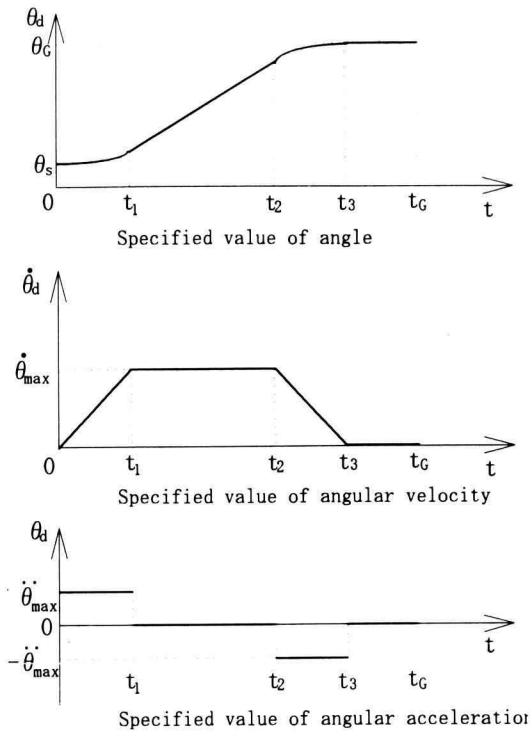


Fig. 3. Specified value pattern

また, Fig. 3 に示すようにマニピュレータ A, B の各関節とも, 目標角度に到達する時間は同じと仮定した。

3. マニピュレータの相互衝突回避

3.1 衝突回避の方針

本研究では, マニピュレータの入力トルクが本来の値では衝突を起こす可能性がある場合, 衝突を回避するため入力トルクの値を修正し, マニピュレータに回避動作を行わせる。その際, マニピュレータ間の最短距離がトルクに反映され, 衝突は生じない。これは一種のポテンシャル法と考えられる。また, 新たに距離感度というパラメータを定義し, これを回避トルクの重みづけに役立てる。

まず, 2 台のマニピュレータのモデルに必要な次の条件を以下のように設定した。

設定条件

- 1) 水平面内において, 作業空間を共有する複数台の水平型 2 自由度マニピュレータを動作させる。マ

- ニピュレータは, 先端リンクのみ衝突の可能性のあるものとする。
- 2) 各マニピュレータの形状パラメータ, 配置位置, 初期位置, 目標位置は既知とする。
- 3) 各マニピュレータにあらかじめ優先度をつけない。全てのマニピュレータが衝突回避動作を行うことができ, 状況に応じて各マニピュレータが衝突回避動作をする。
- 4) 動力学の要素を考慮し, 軌道制御には計算トルク法を用いる。

3.2 最短距離パラメータ

本研究の第一の目的は, マニピュレータ同士を衝突させないことである。そこで, 互いに衝突する可能性がある場合には, Fig. 4 に示すように 2 台のマニピュレータ間の最短距離 l を手先リンクの入力トルクの値に反映させる。

$$\begin{cases} \tau_2^* = \tau_2 - \frac{k_a}{l^2} \\ \tau_4^* = \tau_4 - \frac{k_b}{l^2} \end{cases} \quad (5)$$

k_a, k_b : 重み係数

前述した計算トルク法により求められた入力トルクから, 上式に示すようにマニピュレータ間の最短距離 l の 2 乗に反比例した値を回避トルクとして引く。そして, この新たなトルク τ_2^*, τ_4^* をマニピュレータの入力トルクとすることでマニピュレータの軌道が変わる。(5) 式で 2 台の先端トルク τ_2, τ_4 だけ修正しているのは, 設定条件 1) の先端リンク同士のみ衝突の可能性があるのである。マニピュレータ間の最短距離が衝

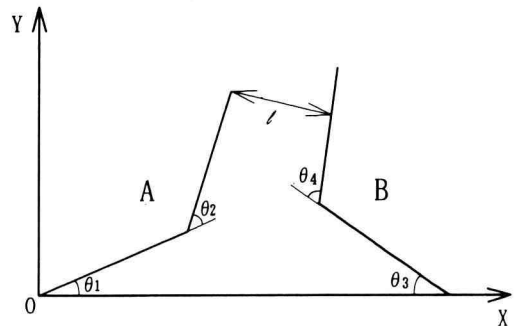


Fig. 4. Shortest distance between manipulators

突可能性の閾値より小さくなった場合には衝突の可能性があると判断し、この反トルク（回避トルク）の作用によって衝突回避の動作が行われる。最短距離 l が小さくなればなるほど、反トルクが大きくなり衝突することはない。

この反トルクは最短距離 l に影響を受けるので、この l を最短距離パラメータと呼ぶ。

3.3 距離感度の導入

重み係数 k_a, k_b を適当に選ぶことによって、マニピュレータ同士が互いに回避動作を行いそれぞれの目標に到達する。しかし、重み係数 k_a, k_b の値の選び方によっては、一方のマニピュレータがもう一方のマニピュレータに先を妨げられ目標に到達できない場合や、マニピュレータ同士が衝突する場合があります。そこで、2台のマニピュレータがそれぞれ微小時間動作した場合に、各マニピュレータが最短距離 l に与える影響を考え、これを重み係数 k_a, k_b の値に反映させる。これを距離感度と呼び、マニピュレータ間の最短距離の変化に影響を与える値として定義する。Fig. 5 に示すように、 A_{i-1} を時刻 t_{i-1} でのマニピュレータ A の第2リンクとし、 A_i を時刻 t_i でのマニピュレータ A の第2リンクとする。マニピュレータ B についても同様に B_{i-1}, B_i を定義する。 $\overline{A_i B_i}$ は A_i と B_i の最短距離を表す線分であり、 $\overline{A_i B_{i-1}}, \overline{A_{i-1} B_i}$ も同様とする。マニピュレータ A が止まっているとした場合にはマニピュレータ B が、または、マニピュレータ B が止まっているとした場合にはマニピュレータ A が、微小時間 dt の期間動くことによるマニピュレータ間の最短距離 l への影響は各々次の式となる。

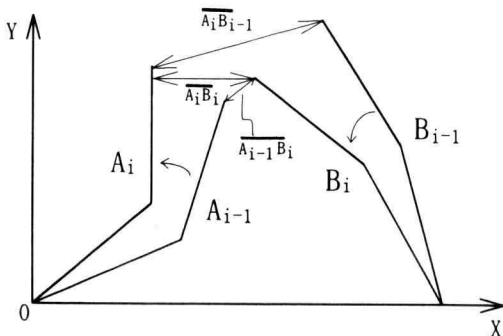


Fig. 5. Distance sensitivity

$$dl_b = \frac{\overline{A_i B_i} - \overline{A_i B_{i-1}}}{dt} \quad (6): \text{ B が動くことによる最短距離 } l \text{ への影響}$$

$$dl_a = \frac{\overline{A_i B_i} - \overline{A_{i-1} B_i}}{dt} \quad (7): \text{ A が動くことによる最短距離 } l \text{ への影響}$$

これらは、マニピュレータ間の最短距離に対する単位時間当たりの双方の感度というべきものとなり、これを距離感度と呼ぶこととする。距離感度の値によってマニピュレータ間の最短距離がどのように変化するのが判る。

この距離感度の比を前に述べた回避のためのトルクの重み係数 k_a, k_b の比とする。

$$k_a : k_b = dl_a : dl_b \quad (8)$$

ただし、 k_a, k_b の絶対値の小さい方の値が1となるように正規化する。つまり、最短距離への影響が大きい方のトルクを大きく変化させることになる。

重み係数の修正

リンク間距離の最短距離 l が衝突可能性の閾値よりも小さくなった場合に衝突の可能性があるとした。し

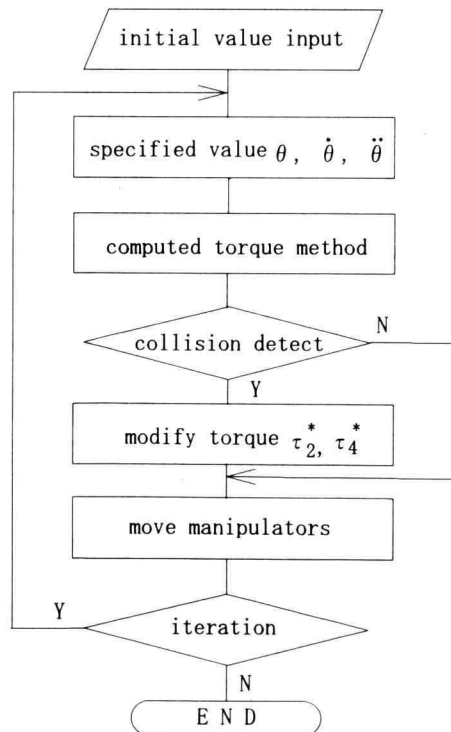


Fig. 6. Flow chart our method

かし、最短距離が短い状態であっても、マニピュレータ同士が近づきつつある場合と遠ざかりつつある場合とがある。

遠ざかりつつある場合には、衝突する可能性がないのが明らかである。よって、トルクの修正は必要ないので、回避のためのトルクの重み係数は、

$$k_a=0, k_b=0$$

とする。これによって、マニピュレータには反トルクが加わらずラグランジュの運動方程式より得られた入力トルクが与えられる。

また、近づきつつある場合には、重み係数の絶対値が大きいマニピュレータに回避のためのトルク、重み係数の絶対値が小さいマニピュレータに回避のためのトルクとは反対向きのトルク(推進トルク)を加える。このことによって、早期に回避動作を終了する。定性的には、重み係数の絶対値が大きいマニピュレータの

動きが抑えられている間に、もう一方のマニピュレータが通り過ぎることになる。

4. シミュレーション実験と検討

4.1 計算機シミュレーション

これまで述べた手法の有効性を確認するため Fig. 1 のモデルに対して計算機シミュレーションを行った。まず、解析の手順の概略を Fig. 6 に示す。なお、計算を行うに当たり運動方程式を解く部分は、4次の Runge-kutta 法を用いた。プログラム言語には、C言語 (Microsoft C/C++ Version 7.0A, Microsoft Visual C++ Verison 1.0) を用いた。

また、各々のマニピュレータが衝突しないと考えた場合の目標とする角度、角速度、角加速度のパターンは Fig. 3 のように設定し、以下のシミュレーションで用いた。角速度については、加速期間と減速期間がそ

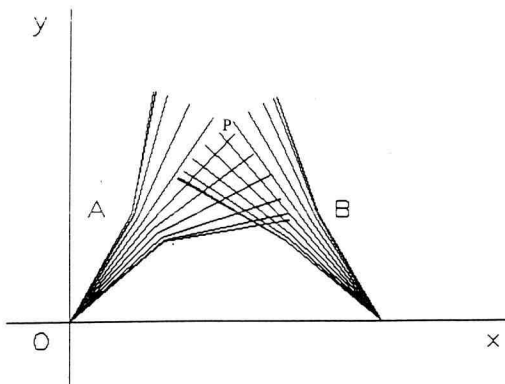


Fig. 7(a). Stick diagram of manipulators (case 1)

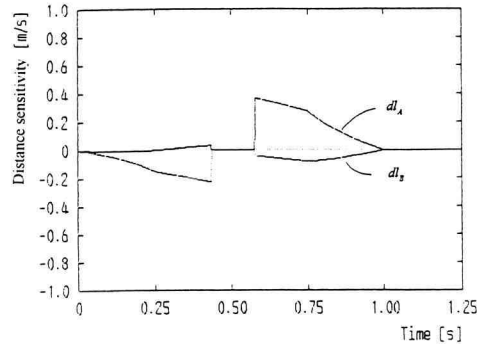


Fig. 7(b). Distance sensitivity (case 1)

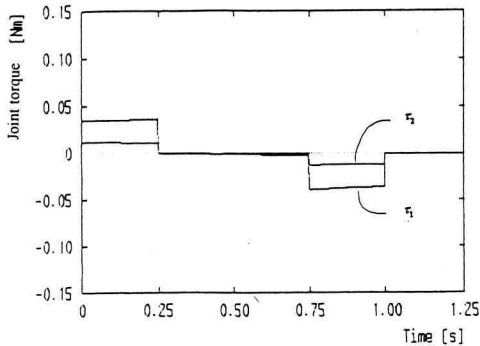


Fig. 7(c). Joint torque of manipulator A (case 1)

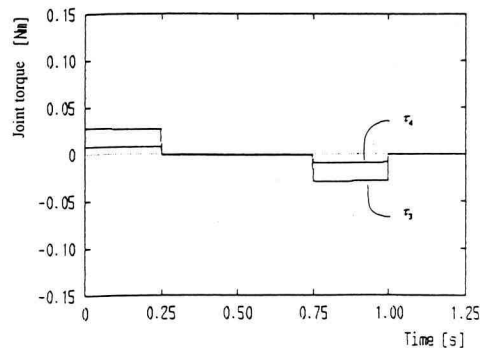


Fig. 7(d). Joint torque of manipulator B (case 1)

それぞれ1/5ずつとなる台形パターンとし、最後に1/5の到達点保持期間を持つものとした。

ただし、 θ_s は開始角度、 θ_c は終了角度、 $\dot{\theta}_{max}$ は最大角速度、 $\ddot{\theta}_{max}$ は最大角加速度の絶対値を示す。また、具体的な数値は、

時間

$$t_1=0.25[s] \quad t_2=0.75[s] \quad t_3=1.00[s] \quad t_c=1.25[s]$$

初期位置

$$\theta_{1s}=40^\circ \quad \theta_{2s}=-30^\circ \quad \theta_{3s}=120^\circ \quad \theta_{4s}=-10^\circ$$

目標位置

$$\theta_{1c}=60^\circ \quad \theta_{2c}=20^\circ \quad \theta_{3c}=140^\circ \quad \theta_{4c}=10^\circ$$

である。

計算の条件として次の値はすべてのシミュレーションに対して共通である。

- 1) サンプリングタイム1 [ms]

- 2) マニピュレータの動作時間1.00 [s]
- 3) マニピュレータの目標角度到達後の到達点保持時間0.25 [s]
- 4) 衝突可能性の閾値0.02 [m]
- 5) 関節トルクの絶対値の上限0.10 [Nm]

シミュレーションモデルのパラメータ

本研究で行ったシミュレーションでは、マニピュレータのパラメータを Table 1 のように設定した。

Table 1. parameter of simulation model

Length of each links : ℓ_i	0.2 [m]
Center of gravity of each links : ℓ_{gi}	0.1 [m]
Mass of each links : m1, m3	0.2 [kg]
m2, m4	0.1 [kg]
Moment of inertia of each links : I1, I3	0.667×10^{-3} [kgm ²]
I2, I4	0.333×10^{-3} [kgm ²]

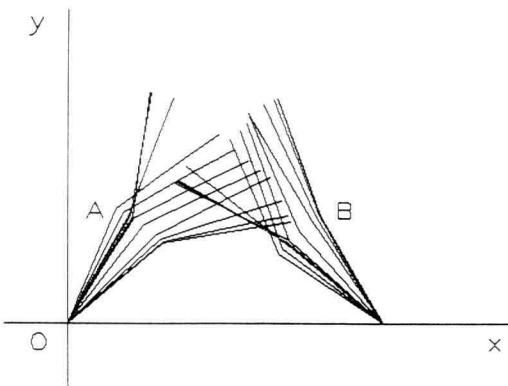


Fig. 8(a). Stick diagram of manipulators (case 2)

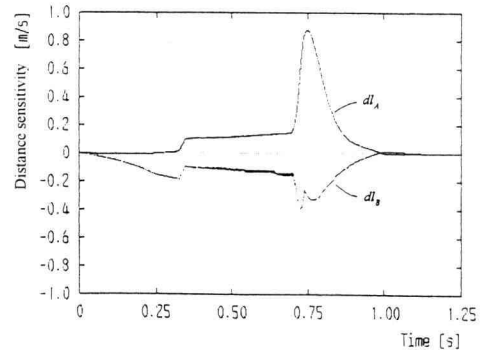


Fig. 8(b). Distance sensitivity (case 2)

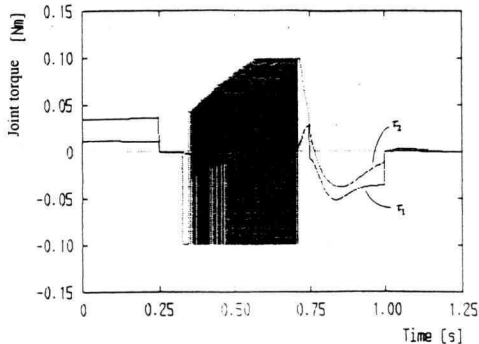


Fig. 8(c). Joint torque of manipulator A (case 2)

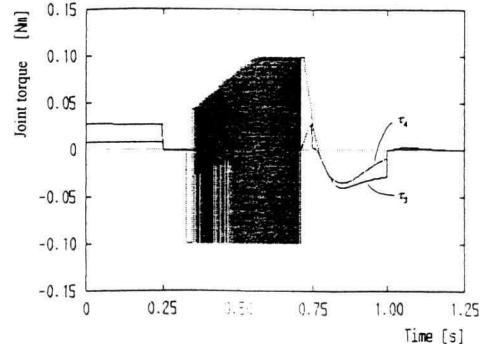


Fig. 8(d). Joint torque of manipulator B (case 2)

ただし、これらのパラメータは、すべてのマニピュレータに共通に用いた。

また、Fig. 2 に示した計算トルク法のサーボ補償器のフィードバックゲインは、

$$k_v=20, k_p=300$$

とした。

上記の設定条件より、次の3つの場合について計算機シミュレーションを行った。

- case 1) 回避を行わない場合。
- case 2) 距離感度を重み係数に反映して回避した場合。
- case 3) 重み係数を修正して回避した場合。

シミュレーション実験

- case 1) 回避を行わない場合 (Fig. 7(a)~(d))。回避を行っていないので、中央付近P点でマニピュ

レータ同士が衝突した。衝突中はマニピュレータ間の距離が0のままなので距離感度も0となった。距離感度は、負の場合にマニピュレータ間距離が近づくことを、正の時には遠ざかることを示している。

距離感度を重み係数に反映して回避した場合 (Fig. 8(a)~(d))。

次に距離感度の比によって重み付けした反トルクを与えた場合である。

マニピュレータ B が A に近づくと、反トルクが生じ動きが少し停滞して衝突回避した。また、マニピュレータ A も B に近づいている間は、反トルクのために動きが停滞し衝突回避動作の終了が遅くなったようである。衝突回避動作の終了時刻は、約 0.7 秒のときであった。マニピュレータ A, B ともに手先リンクの関節のトルクが飽和し非常に短い周期でトルクの正逆を繰り返したようである。

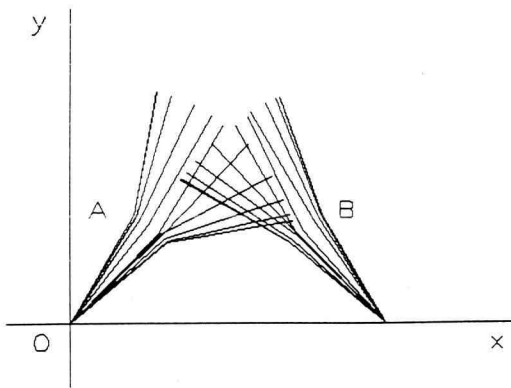


Fig. 9(a). Stick diagram of manipulators (case 3)

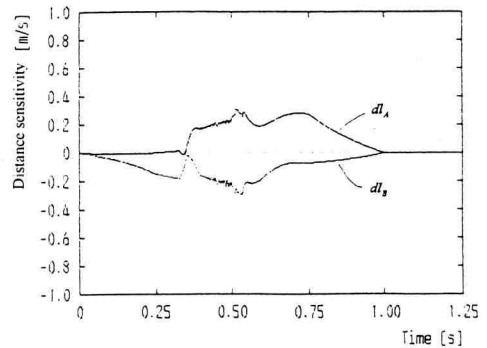


Fig. 9(b). Distance sensitivity (case 3)

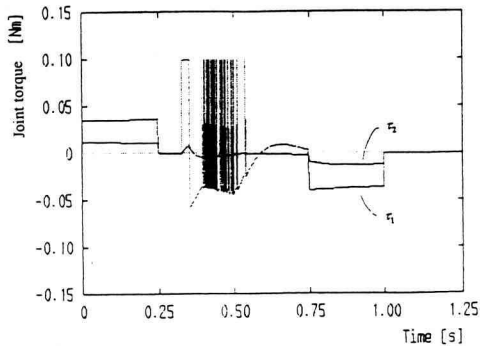


Fig. 9(c). Joint torque of manipulator A (case 3)

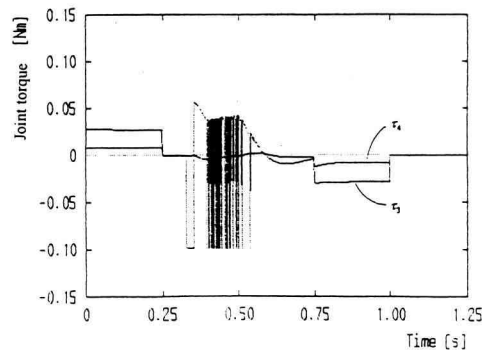


Fig. 9(d). Joint torque of manipulator B (case 3)

case 3) 重み係数を修正して回避した場合 (Fig. 9(a) ~ (d)).

マニピュレータが、近づくか遠ざかるかによって修正したトルクを与えた場合である。

マニピュレータ B が A に近づくとき、反トルクが生じ動きが少し停滞し、マニピュレータ A は推進トルクによって回避動作が早く終了した。衝突回避動作の終了時刻は、約 0.55 秒から 0.60 秒の間となった。今回、回避シミュレーションを行ったものの中では、距離感度の絶対値が最も小さかった。これは、マニピュレータ同士が、急激に近づいたり遠ざかることがなかったことを示している。マニピュレータ A の手先リンクの関節トルクが、回避を行うときにマニピュレータを推進する方向のトルクとして生成されていた。

4.2 シミュレーションの検討

回避を行わない場合 Fig. 7(a) の P 点で衝突が生じているが、本手法の回避処理の結果 case 2), 3) ともに衝突は生じなかった。また、case 2) と 3) の衝突回避方法では、回避時にかかる関節トルクは短い周期にトルクの飽和レベルで正逆反転するという結果になった。この原因は、衝突可能性の閾値を境に回避のためのトルクが二値的に加わったり、加わらなかったりするためであろう。これは衝突可能性の閾値に幅を持たせ、回避のためのトルクを距離によって補間することで回避が可能であると考えられる。

case 2) と 3) を比較すると、回避動作期間の短さや関節角度誤差の小ささから見て、case 3) の方が優れていると考えられる。

case 3) の衝突回避方法の欠点は、回避時にかかる関節トルクが短い周期にトルクの飽和レベルで正逆反転することである。

5. おわりに

マニピュレータの相互衝突回避に動力学を考慮し、マニピュレータの最短距離に応じた回避トルクを順次変更することで衝突回避軌道を生成する手法を提案した。結果として次のような結論を得た。

- 1) マニピュレータの相互衝突回避に動力学を考慮した方法を提案した。
- 2) 距離感度を設定しトルクの修正に反映させた。
- 3) 本手法により、優先度、経路を予め設定せず、回避軌道が生成できた。
- 4) 修正した距離感度により、回避動作を早く終了することができた。
- 5) 2台のマニピュレータのシミュレーションにより本手法の有効性を確認した。

今後は、以下のことが課題として残されている。

- 1) 衝突可能性の閾値に幅を持たせるなどして関節トルクをなめらかに変化させる必要がある。
- 2) より実用に近づけるために 3 次元空間で 3 自由度以上のマニピュレータが衝突回避を行えるよう本手法を拡張する必要がある。

本研究は、本学大学院学生榎本卓也君 (現、(株)日立製作所) の協力によってなされた。記して感謝を表す。

参考文献

- 1) 毛利, 圓島, 山本: 2台のマニピュレータの衝突回避軌道計画, 日本機械学会論文集, 57-538C, pp. 1985-1990, (1991).
- 2) J. Lee and Z. Bien: Collision-free trajectory control for multiple robots based on neural optimization network, Robotica, vol. 8, pp. 185-194, (1990).
- 3) 田口・河原崎, 2台のマニピュレータに関する衝突回避問題, 第9回日本ロボット学会学術講演会予稿集, pp. 507-508, (1991).
- 4) 田口・河原崎, 2台のマニピュレータの衝突回避に関する研究, 神奈川工科大学研究報告 B 理工学編, 第18号, pp. 103-113 (1994).
- 5) 田口・河原崎・榎本, 複数マニピュレータの相互衝突回避, 第11回日本ロボット学会学術講演会予稿集, pp. 1135-1136, (1993).
- 6) 吉川, ロボット制御基礎論, コロナ社, (1988).