

仮想アームを用いた冗長マニピュレータの逆運動学解法と まきつき制御への応用*

辻 敏 夫*¹, 中山 聖 也*²
荒 木 淳*², 伊 藤 宏 司*³

Instantaneous Inverse Kinematic Solution for Redundant Manipulators Based on Virtual Arms and Its Application to Winding Control

Toshio TSUJI, Seiya NAKAYAMA,
Atsushi ARAKI and Koji ITO

The present paper proposes an instantaneous inverse kinematic solution for redundant manipulators based on virtual arms. The virtual arm has the same kinematic structure as the manipulator except that its end-point is located on the joint or link of the manipulator. When the appropriate virtual arms are used, the configuration of the manipulator can be represented by a set of end-points of the virtual arms. First of all, this paper formalizes the kinematics of virtual arms and derives instantaneous inverse kinematics. Then, the method is applied to winding control for hyper-redundant manipulators. The winding control presented here is divided into two steps: 1) planning desired positions for virtual end-points, 2) integrating them into the joint trajectory of the manipulator. The desired positions of each virtual arm can be computed in a parallel and distributed way and it is not necessary to consider joint space of the manipulator. Finally computer simulations show that the winding control for a hyper-redundant manipulator can be performed in 3D-space.

Key Words: Computer Control, Mechanics, Robotics, Redundant Manipulator, Inverse Kinematics, Winding Control, Virtual Arm

1. まえがき

多関節マニピュレータの制御では、手先の運動だけでなく、マニピュレータ全体の姿勢を考慮しなければならない。例えば、障害物回避を含む経路生成問題では、手先だけでなくアーム全体が障害物と干渉しないような動作計画を行う必要がある。そのため、一般にはマニピュレータの関節空間で動作計画を行う方法がとられている⁽¹⁾⁻⁽³⁾。この空間ではマニピュレータの姿勢が一つの点で表現されるため、アームと環境との関係を簡単に表現でき、多関節マニピュレータを移動ロボットとして取り扱うことが可能になる。しかしながら、この方法では作業空間で表現された環境(例えば、障害物など)を直接取り扱うことはできず、また、カメラなどのセンサ情報も関節空間に変換する必要を生じる。ところが、関節自由度が大きくなるにつれて、この変換は非常に複雑になるため、本稿で取り扱うような冗長マニピュレータへの適用は一般に困難である。

そこで、本稿では仮想アームという概念を導入し、マニピュレータの姿勢を作業空間で表現する方法を採用する⁽⁴⁾⁻⁽⁶⁾。仮想アームとは、対象とするマニピュレータ(以下、仮想アームに対して実アームと呼ぶ)の各関節やリンク上に手先を持ち、リンク長や関節角度などのパラメータは実アームと同じであるような仮想的なマニピュレータである。この仮想アームを複数、定義することにより、実アームの姿勢を各仮想アームの手先の集合として表現することができ、作業環境とアーム全体との干渉を作業空間上で考えることが可能になる。

われわれは、先にこの仮想アームをエネルギー最小化の原理で動作するニューラルネットを用いて学習的に表現することにより、冗長マニピュレータの軌道を分散的に生成できることを示した⁽⁶⁾。本稿では、仮想アームの運動学をモデル化し、その逆運動学解法を明らかにするとともに、本手法を超多自由度マニピュレータのまきつき動作制御に応用することを試みる。

以下、2章では仮想アームの運動学を定式化し、3章で逆運動学解を導出する。4章では、仮想アームを用いた超多自由度マニピュレータのまきつき制御法を

* 原稿受付 平成4年4月20日。

^{*1} 正員, 広島大学工学部(〒724 広島市鏡山1-4-1)。

^{*2} 広島大学工学部。

^{*3} 正員, 豊橋技術科学大学(〒441 豊橋市天伯町雲雀ヶ丘1-1)。

提案し、計算機シミュレーションにより、三次元空間でのまきつき制御が実現できることを示す。

2. 仮想アームの定義と運動学

対象とするマニピュレータ (実アーム) は、関節自由度が m の冗長マニピュレータとし、作業空間は実アームのベースを原点とする直交座標系とする。この実アームに対して、実アームの関節部分やリンク上に手先を持つような仮想アームを定義する。図1は、 m 関節マニピュレータに対して、仮想アームを設定した例である。各仮想アームのベースの位置、関節角度、リンク長などのパラメータは、実アームに対応している。ここでは、一般に $n-1$ 本の仮想アームを定義することにし、 n 番目の仮想アームを実アームに対応させる。適切な仮想アームを用意することにより、実アームの姿勢を作業空間上の仮想アームの手先位置の集合として表現することが可能となる。

いま、作業空間の自由度を l とし、アーム i の手先変位を表すベクトルを $dX^i=(dX^i_1, dX^i_2, \dots, dX^i_l)^T$ 、実アームの関節角変位を表すベクトルを $d\theta=(d\theta_1, d\theta_2, \dots, d\theta_m)^T$ とすると、アーム i ($i=1, 2, \dots, n$) の手先変位と実アームの関節角変位との関係は、

$$dX^i = J^i(\theta) d\theta \quad (1)$$

と表現することができる。ここで、 $J^i(\theta) \in R^{l \times m}$ は i 番目の仮想手先に関するヤコビ行列である (以下、 J^i と略記)。例えば、仮想アーム i が実アームの $r+1$ 番目の関節上に手先を持つ場合、ヤコビ行列 J^i は実アームのヤコビ行列 J^m を用いて

$$J^i = J^m \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} \quad (2)$$

$$\begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} \in R^{m \times m}, I_r: r \text{ 次の単位行列}$$

のように表すことができる。

式 (1) を全アームについてまとめると、

$$dX_v = J d\theta \quad (3)$$

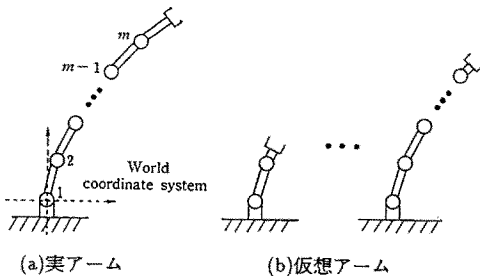


図1 m 関節平面マニピュレータに対する仮想アームの例

を得る。ただし、

$$dX_v = \begin{bmatrix} dX^1 \\ dX^2 \\ \vdots \\ dX^n \end{bmatrix}, J = \begin{bmatrix} J^1 \\ J^2 \\ \vdots \\ J^n \end{bmatrix}$$

で、 $dX_v \in R^L$ は各手先の変位をまとめたベクトル、 $J \in R^{L \times m}$ は、各アームのヤコビ行列を一つにまとめた行列である。記号 $L=ln$ は仮想アームの手先の自由度の総和を表す。仮想アームの定義から、アーム i はアーム 1 からアーム $i-1$ を含むことになるので、 J の構造は規則性をもち、簡単に計算することができる。

一方、アーム i の手先の力・モーメントを表すベクトルを $F^i=(F^i_1, F^i_2, \dots, F^i_l)^T$ 、アーム i の関節トルクを表すベクトルを $\tau^i=(\tau^i_1, \tau^i_2, \dots, \tau^i_m)^T$ とすると、

$$\tau^i = J^{iT} F^i \quad (4)$$

の関係が成り立つ。上式を全アームについてまとめると、

$$\tau = \sum_{i=1}^n \tau^i = J^T F_v \quad (5)$$

となる。ただし、

$$F_v = \begin{bmatrix} F^1 \\ F^2 \\ \vdots \\ F^n \end{bmatrix}$$

で、 $F_v \in R^L$ は各手先の力・モーメントをまとめたベクトルである。

以上の関係を図2に示す。ここで、 $K_j \in R^{m \times m}$ 、 $K_v \in R^{L \times L}$ はそれぞれ、実アームの関節剛性行列、仮想手先の剛性行列を表す。図2から、仮想アームを用いて実アームを制御するには、二つの方法があることがわかる。第一は各仮想手先の目標変位 dX_v^* を実現可能な実アームの関節目標変位 $d\theta^*$ に変換し、関節フィードバック制御を行う方法で、第二は dX_v^* から手先フ

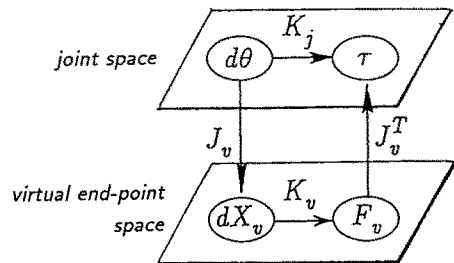


図2 関節空間と仮想手先空間

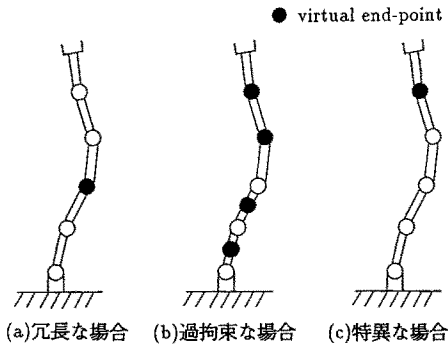


図3 仮想手先の3つの例

ードバック制御を用いて F_v を求め、実アームの関節駆動トルク τ に変換する方法である。ここではこのうち、関節フィードバック制御を取り上げることにし、仮想手先目標変位から実アームの関節変位を求める問題（仮想アームの逆運動学問題）を考察する。

3. 仮想アームの逆運動学

各仮想手先の目標変位から実アームの関節変位を求める問題は、式(3)の連立方程式の解法に帰着する。このとき、実アームの関節自由度と仮想アームの設定条件に応じて、以下の三つの場合を考慮せねばならない⁽⁴⁾。

図3にこの三種類の例を示す。実アームは、5関節平面アーム ($m=5$) で手先の自由度は姿勢を除いて $l=2$ であるから、このアームは冗長自由度を持つ。図3(a)は第3関節上に仮想手先を設定した場合 ($n=2$) である。このとき、目標仮想手先変位は仮想手先と実アームの手先を合わせて4自由度であるから、このアームにはまだ冗長性が残っている ($L < m$)。すなわち、式(3)の行列 J はアームが特異姿勢にない限り、行に関してフルランク ($\text{rank} J = L$) となり、式(3)の連立方程式は不定となる。図3(b)は4ヶ所に仮想手先を設定した場合 ($n=5$) である。目標仮想手先変位は10自由度となり、アームは過拘束状態となっている ($L > m$) ことがわかる。このとき、行列 J (式(3)) は列に関してフルランク ($\text{rank} J = m$) となり、式(3)の解は存在しない。一方、図3(c)は第5関節上に仮想手先を設定した場合 ($n=2$) である。目標仮想手先変位の自由度は図3(a)の場合と同様に4自由度となり、一見、式(3)の解が存在するように見える。しかしながら、この場合は仮想アームの手先と実アームの手先との間に関節が一つしか存在しないため、仮想アームと実アームの手先目標変位を同時に実現すること

はできない。このとき、行列 J は行、列のいずれに関してもフルランクでなくなる。このように式(3)の解の存在条件は、行列 J のランクに応じて、①冗長な場合、②過拘束な場合、③冗長な部分と過拘束な部分を合わせもつ特異な場合の三つに分類できることがわかる。

これら三種類の場合を取り扱うため、行列 J をあらかじめ最大階数分解⁽⁷⁾し、

$$J = J_a J_b \quad (6)$$

とおく。ただし、 $J_a \in R^{L \times p}$ 、 $J_b \in R^{p \times m}$ で、 $\text{rank} J = \text{rank} J_a = \text{rank} J_b = p$ である。上式を式(3)に代入すると、

$$dX_v = J_a J_b d\theta \quad (7)$$

となる。 J_a は行列 J のうち過拘束な部分を、 J_b は行列 J のうち冗長な部分を表現している。

まず、式(7)を $J_b d\theta$ に関して解くことを考えよう。

$$dX_b = J_b d\theta \quad (8)$$

とおくと、式(7)は、

$$dX_v = J_a dX_b \quad (9)$$

と書ける。行列 J_a は列に関してフルランクであるから、一般に式(9)を満たす解 dX_b は存在せず、近似解を求めることになる。そこで、評価関数、

$$Q(dX_b) = (dX_v^* - J_a dX_b)^T W (dX_v^* - J_a dX_b) \quad (10)$$

を導入し、この評価関数を最小にする dX_b を求めることにする。ここで、 $W \in R^{L \times L}$ は正定な重み行列で、

$$W = \text{diag.} [w_1^1, \dots, w_1^l, w_2^1, \dots, w_2^l, \dots, w_n^1, \dots, w_n^l] \quad (11)$$

とする。 $\text{diag.}[\]$ は対角行列を表す。 $w_j^i > 0$ は、アーム i の目標手先変位の j 番目の要素に対する重み係数で、この値を調節することにより、各仮想手先に対する優先度を指定することができる。この評価関数 $Q(dX_b)$ を最小にする解 dX_b は、

$$dX_b = (J_a^T W J_a)^{-1} J_a^T W dX_v^* \quad (12)$$

で与えられる。

このようにしてベクトル dX_b が求まると、次に式(8)を解き、実アームの関節変位 $d\theta$ を求めればよい。式(8)に含まれる行列 J_b は行に関してフルランクであるから、式(8)の解は必ず存在する。 J_b の擬似逆行列を用いて、関節変位 $d\theta$ を求めると、

$$d\theta = J_b^+ dX_b + (I_m - J_b^+ J_b) r \quad (13)$$

となる。ただし、 $J_b^+ = J_b^T (J_b J_b^T)^{-1} \in R^{m \times p}$ は J_b の擬似逆行列、 I_m は m 次の単位行列、 $r \in R^m$ は任意のベク

トルで、このベクトル r を用いてアームの冗長自由度を利用することも可能である⁽⁸⁾。ここでは $r=0$ とし、最小ノルム解

$$d\theta = J_0^+ dX_0 \quad (14)$$

を用いることにする。

以上のように、目標仮想手先変位 dX_0^* が与えられると式 (12),(14) を用いて実アームの関節変位を求めることができる。また、実際に生じる各アームの手先変位 dX_i も、式 (7),(12),(14) から、

$$dX_i = J_a(J_a^T W J_a)^{-1} J_a^T W dX_0^* \quad (15)$$

として計算することができる。

この方法は、図3に示した三種類の場合のいずれにも適用することができる。例えば、図3(a)のように冗長な場合は、 $J_a = I_L$ (L 次の単位行列)、 $J_b = J$ であるから、 $dX_b = dX_0^*$ となり、関節変位 $d\theta$ は式 (14) より、

$$d\theta = J^+ dX_0^* \quad (16)$$

で与えられる。一方、過拘束な場合(図3(b))では、 $J_a = J$ 、 $J_b = I_m$ であるから、 $dX_b = d\theta$ となり、関節変位 $d\theta$ は式 (12) より、

$$d\theta = (J^T W J)^{-1} J^T W dX_0^* \quad (17)$$

となる。もちろん、冗長な部分と過拘束な部分を合わせ持つ場合(図3(c))には、式 (12),(14)をそのまま用いればよい。

以上の方法を用いることにより、各仮想手先の動作計画を作業空間で考えることができ、また、重み行列 W を用いて特定の仮想アームの運動を優先することが可能になる。なお、吉川⁽⁸⁾ はサブタスクの優先順位を考慮した冗長アームの逆運動学解法を示している。この方法は、複数の優先順位付けされたサブタスク(目標軌道)が与えられたとき、零空間ベクトルを利用して上位のサブタスクを満足する範囲内で下位のサブタスクをできるだけ実現しようとするものである。本論文で示した各仮想手先の目標変位は、この優先順位付けされたサブタスクと考えることもできるが、冗長自由度が上位のサブタスクですべて利用されてしまった場合(零空間ベクトルが存在しない)や同じ優先度を持つ複数のサブタスクが存在する場合にも適用可能である。次章では、この仮想アームの逆運動学解法を利用した超多自由度マニピュレータのまきつき制御法を提案する。

4. まきつき制御のための軌道生成法

生物の運動には、象の鼻や蛇の体幹など形状を能動的

に変形し、対象物に応じて巧みにまきつき動作を実現している例が観察される。広瀬⁽⁹⁾はこのような機能体を索状能動体と呼び、その運動制御メカニズムを明らかにするとともに、数々の実機を開発している。例えば、ACMⅦは、全8節24関節自由度を有する象の鼻型マニピュレータである⁽¹⁰⁾。和田ら⁽¹¹⁾は、広瀬らによって提案された側抑制形の制御方式⁽¹²⁾を用いて、まきつき動作を行う方法を示している。この方法ではアームの触覚情報を利用し、各関節がそれぞれ協調しながらまきつき動作を行うことができる。しかしながら、各関節の回転軸が平行であるために、すべてのリンクが同一平面内でしか動くことができず、結局、二次元平面におけるまきつき動作のみ適用可能である。そこで本稿では、仮想アームを用いることにより、三次元空間におけるまきつき動作が実現できることを明らかにする。

4・1 まきつき動作 いま、図4のような状況を考えよう。マニピュレータは三次元空間($l=3$ とする)において凸型の対象物と第 k リンクで接触している。作業目的は、第 k リンクから実手先まで順に対象物に接触し、まきつくことである。ここで、対象とするマニピュレータは、リンク長が対象物の大きさに比べて十分短く、まきつき動作が可能であると仮定する。もし、対象物の形状や位置などの情報が与えられていれば、対象物の表面に各仮想手先の目標位置を定めることによって容易にまきつき動作を実現することができる。しかし、ここでは対象物に関する情報は全く与えられていないとし、触覚情報のみに基づいてまきつき動作を行うものとする。

ここで、触覚情報として接触力ベクトル $c^i \in R^3$ が利用できることと仮定する。接触力ベクトル c^i は第 i リンクが対象物と接触しているとき、どの方向から対象物

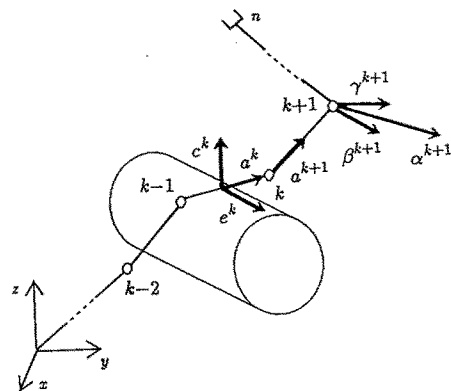


図4 まきつき動作

と接触しているかを表す作業座標系で表現されたベクトルである(図4参照)。第*i*リンクが対象物と接触していないときは、 $c^i = 0$ となる。また、簡単のため、仮想アームは各リンクの先端部分に手先を持つものをリンク数(*n*)だけ定義する。

4・2 目標仮想手先変位の計画 以下に、まきつき動作を実現するための目標仮想手先変位 dX^* の計画方針を示す。

- (1) 対象物と接触している第*k*リンクよりベース側のリンクが動くと、接触しているリンクが離れる可能性がある。そこで、 $i = 1, \dots, k$ の仮想手先の目標変位は $dX^{i*} = 0$ とする。そして、重み行列 W (式(11))の対応する要素を大きく設定することによって仮想手先の優先度を高く設定する。
- (2) まきつきためには次に第 $k+1$ リンクが対象物と接触せねばならない。そこで、 $i = k+1$ の仮想手先には対象物の方向に目標変位を与える。
- (3) 第 $k+2$ リンク以後の仮想手先の運動はまきつき動作には直接影響を与えない。ここでは、 $i = k+2, \dots, n$ の仮想手先の目標変位は $dX^{i*} = 0$ とし、重み行列 W の値により優先度を調節することにする。

このようにして計画された目標仮想手先変位 dX^* を用いて、式(12),(14)によりマニピュレータを動作させる。そして第 $k+1$ リンクが対象物に接触すれば $k = k+1$ とし、同様の手続きを $k = n$ となる(まきつき動作の終了)まで繰り返す。

上記の計画方針を満たすため、各仮想手先の目標変位 $dX^{i*} \in R^3$ ($i = 1, \dots, n$) を、

$$dX^{i*} = \eta \alpha^i (1 - M^i) (M^{i-1} - M^i) \quad (18)$$

のように定める。ここで、 η は目標変位の大きさを与える正の定数で、 $\alpha^i \in R^3$ はまきつき方向を与えるベクトル(作業座標系)である。 M^i は第*i*リンクと対象物との接触状態を表し、 $M^i = 1$ のとき接触、 $M^i = 0$ のとき非接触を意味する。この接触状態 M^i は接触力ベクトル c^i から簡単に求めることができる。式(17)において、すべての dX^{i*} ($i = 1, \dots, n$) が0(平衡状態)となるのは、

$$\textcircled{1} \text{ まきつき状態: } M^0 = M^1 = \dots = M^{k-1} = 0$$

$$M^k = M^{k+1} = \dots = M^n = 1$$

$$\textcircled{2} \text{ 非接触状態: } M^0 = M^1 = \dots = M^n = 0$$

に限られる。したがって、方向ベクトル α^i が適切に与えられれば、一度、マニピュレータが対象物に接触すると実手先まで自然にまきつき、平衡状態に達して動作が終了することになる。

第*i*番目の仮想手先に与える方向ベクトル α^i は次式のように決定する。

$$\alpha^i = \beta^i + \gamma^i \quad (19)$$

$$\beta^i = s^i e^{i-1}, \quad \gamma^i = a^i \times e^{i-1} \quad (20)$$

$$s^i = \begin{cases} 1 & \text{if } (a^i, e^{i-1}) < 0 \\ 0 & \text{if } (a^i, e^{i-1}) = 0 \\ -1 & \text{if } (a^i, e^{i-1}) > 0 \end{cases} \quad (21)$$

$$e^{i-1} = a^{i-1} \times c^{i-1} \quad (22)$$

ここで、 \times は外積、 $(*, *)$ は内積、 $a^i \in R^3$ は第*i*リンクの方向ベクトル(単位ベクトル)を表す。式(19)の方向ベクトル α^i は、第*i*-1リンクが対象物と接触状態にあるとき($c^{i-1} \neq 0$)のみ値をもち、それ以外は0となる。

第*k*リンクが対象物と接触している場合について考えよう。このとき、第*k*+1番目の仮想手先に与える方向ベクトル α^{k+1} (式(19))は次のように生成される。

まず、 $c^k \neq 0$ であるから、式(22)のベクトル $e^k \in R^3$ は a^k と c^k が張る平面 P に直交する方向を与える(図4)。したがって、式(20)のベクトル $\beta^{k+1} \in R^3$ は第*k*+1番目の仮想手先(第*k*+1関節)が平面 P に向かう方向となる。一方、 $\gamma^{k+1} \in R^3$ はその平面 P 上で第*k*+1リンクを対象物の方向へ回転させるためのベクトルである。すなわち、第*k*リンクと第*k*+1リンクができるだけ同一平面 P 上で対象物と接触するように方向ベクトル α^{k+1} を決定していることになる。

以上、本稿で提案するまきつき動作のための軌道生成法について述べた。次節では、本手法の有効性を示すため、シミュレーション実験を行う。

4・3 シミュレーション実験 図5にシミュレーションに用いた超多自由度マニピュレータを示す。このマニピュレータは17リンク49関節自由度($m = 49$)を有し、各リンク長は0.06mとした。なお、第2・3・4関節、第5・6・7関節、 \dots 、第47・48・49関節は、それぞれ球対偶を想定している。仮想アームは各リンクの先端部分に手先を持つものを17本定義している($n = 17$)。

まず、ある初期姿勢のマニピュレータに対して種々の方向から対象物(直径0.14mの円柱)を接触させ、まきつき動作を行なわせた。結果の一例を図6に示す。図6(a)(b)(c)(d)は、同じ初期姿勢に対して同じ対象物をそれぞれ上側、下側、手前側、向こう側から接触させた結果である。ただし、式(11)の重み行列 $W \in R^{51 \times 51}$ は

$$w_j^i = \begin{cases} 200.0 & (i = 1, 2, \dots, k; j = 1, 2, 3) \\ 20.0 & (i = k+1; j = 1, 2, 3) \\ 0.01 & (i = k+2, \dots, 17; j = 1, 2, 3) \end{cases} \quad (23)$$

とし、式(18)の η は $\eta=2.0$ とした。ここで、 k は対象物に連続して接触しているリンクのうち最も実手先側のリンク番号を表す。図6からいずれも、接触後、うまく対象物にまきついており、三次元空間におけるまきつき動作が実現できることがわかる。

図7は接触点より手先側の仮想アームの優先度を変化させた場合の結果である(4.2節参照)。図7(a)は重み行列 W (式(11))を

$$w_j^i = \begin{cases} 200.0 & (i = 1, 2, \dots, k; j = 1, 2, 3) \\ 20.0 & (i = k+1; j = 1, 2, 3) \\ 1.0 & (i = k+2, \dots, 17; j = 1, 2, 3) \end{cases} \quad (24)$$

のように設定し、図7(b)は式(23)を用いた。第 $k+2$ 以後の仮想手先の運動はまきつき動作に影響しないため、いずれも、対象物である円柱に接触した後、うまくまきつき動作を行っている。ただし、図7(a)ではまきつき動作を完了していない手先側の仮想手先の重みが図7(b)より大きいため、その位置にとどまろうとする力が働き、しなるようにまきつき動作を行っている。このように、仮想手先の優先度を指定することにより、さまざまなマニピュレータの姿勢を実現することが可能となる。

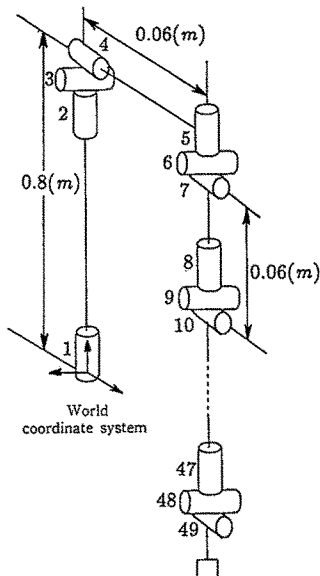


図5 超多自由度マニピュレータのリンクモデル

以上のように、本稿で提案するまきつき制御法は、対象物の形状が凸型で、各リンクが連続して対象物に接触できる場合は、安定に動作することが明らかになった。なお、本手法では式(3)に基づいて制御法を導出しているため、目標手先変位 dX^* を微小変位に限定する必要がある。手先変位や関節角変位が大きくなると計算誤差を生じてしまうからである。今後は、関節角度と手先位置の関係を表す非線形方程式から、直接、逆運動学解を導出する必要がある(6)。

5. まとめ

本論文では、冗長マニピュレータを複数の仮想アームの集合体として捉え、仮想手先の数、位置に伴う運動学的な条件を明らかにするとともに、冗長、過拘束いずれにも適用可能な仮想アームの逆運動学解を導出

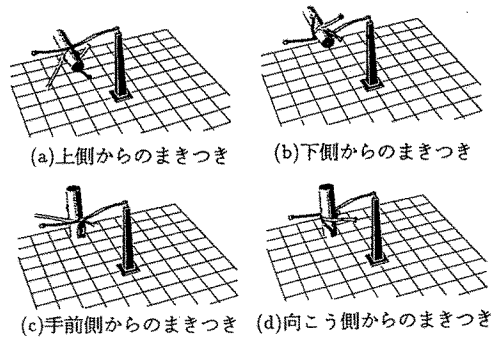


図6 まきつき動作のシミュレーション結果

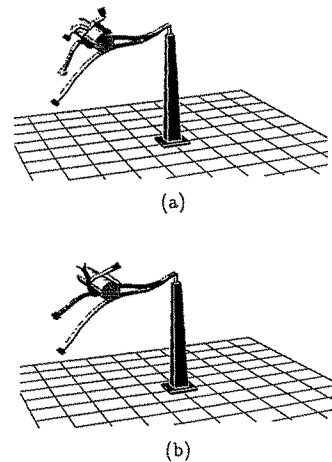


図7 重み行列による軌道の変化

した。そして、軌道生成問題の一つとして、まきつき動作を取り上げ、仮想アームを用いたまきつき制御法を提案した。その結果、

- (1) 仮想アームを用いることにより、冗長アームの姿勢を作業空間上で表現することが可能となり、作業環境を動作計画時に考慮することが容易である、
- (2) アームの代表点を仮想アームとして捉えることにより、任意の数、位置の代表点を実手先と同じ形で統一的に定式化できる、
- (3) 仮想アームを用いることにより、対象物の位置や形状に関する情報が全く与えられていない場合でも、触覚情報のみに基づいて三次元空間におけるまきつき動作を実現できる、
などの点が明らかになった。

今後は、本まきつき制御法をより一般的なリンク長や対象物形状に適用できるように拡張するとともに、仮想手先フィードバックを用いた分散的な制御則の開発、仮想アーム間の通信機能の導入などを検討し、仮想アームの特徴を生かした冗長マニピュレータの制御法を開発したいと考えている。なお、本研究の一部は文部省科学研究費(01750399, 03234105)の研究助成によるもので、ここに改めて謝意を表します。

[参考文献]

- (1) Lozano-Pérez, T., IEEE Trans. Sys. Man Cybernet., 11-10(1981), 681.
- (2) Lozano-Pérez, T., IEEE Trans. Comput., 32-2(1983), 108.
- (3) Lozano-Pérez, T., IEEE J. Robotics Automa., 3-3(1987), 224.
- (4) 辻・中山・伊藤, 第1回自律分散シンポジウム資料, (1990-4), 13.
- (5) 辻・中山・伊藤, 第2回自律分散シンポジウム資料, (1991-1), 89.
- (6) 中山・辻・伊藤, ロボティクス・メカトロニクス91'講演会講演論文集(A), No.910-31(1991-6), 177.
- (7) 児玉・須田, システム制御のためのマトリクス理論, (1986), 64, 計測自動制御学会.
- (8) 吉川恒夫, 日本ロボット学会誌, 2-6(1984), 587.
- (9) 広瀬茂男, 生物機械工学やわらかいロボットの基本原理と応用, (1987), 工業調査会.
- (10) 広瀬茂男, ロボット技術百科, (1982), 99, 日刊工業新聞社.
- (11) 和田・久場, 計測自動制御学会論文集, 20-10(1984-10), 973.
- (12) 広瀬・梅谷, 計測自動制御学会論文集, 12-5(1976-5), 543.