2台のマニピュレータの衝突回避に関する研究

田　口　幹*・河原崎　徳　之*

Collision Avoidance Problem for Two Manipulators

Kan TAGUCHI and Noriyuki KAWARAZAKI

Abstract

In this paper a method for collision-free path planning of two manipulators in a common workspace is presented. Supposing one of two manipulators is given the priority for its motion, the other one plannes the collision-free path. We propose the searching space which describes time-depended collision regions first. Second, a collision-free path is calculated in this searching space using a heuristics (A* algorithm). Applying proposed method to the two manipulators with two links, effectiveness of the method is certified by a simulation.

1. はじめに

近年ロボット技術の発展はめざましく，工場などのような閉鎖的空間から開かれた環境にまでその作業範囲は広く，その用途も多岐に渡り研究されている。マニピュレータに関しては，産業用ロボットや極限作業用ロボットなどの実用化に向けた研究・開発が盛んに行われている。そして高度で複雑な作業を行うために，同一作業空間上に複数のマニピュレータを使用するようになり，複雑な制御が要求されるようになった。

複数のマニピュレータを同時に扱う際に問題となるのは，お互いの作業が重なってしまった場合，衝突する危険があるということである。人間が教示する通りにしか行動できないマニピュレータであった場合，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。

ロボットを使う人間の負担を軽くするためには，基本的なことはマニピュレータが自律的に判断し行動することが望ましい。そのような基本の一つにマニピュレータの障害物回避機能，すなわち，作業空間の障害物を回避しながらマニピュレータがタスクを達成する機能があり，これをもつことはロボットの知能化の第一歩だと見なされている。それゆえ，初期位置・姿勢から最終位置・姿勢まで互いに衝突しないようなマニピュレータの動作を自動的に生成するアルゴリズムの研究が必要となった。

そこで，様々な障害物回避のためのアルゴリズムが研究されてきた[1~4]。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回避するための経路生成などが主流である。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同一の作業空間上に複数のマニピュレータを有するようになる。そしてそれらが独自に作業を行う場合などに起こりやすいマニピュレータ同士の衝突回避問題については，今まであまり触れられていない。このマニピュレータ同士の衝突回避問題は，障害物となるマニピュレータの位置や姿勢が時間とともに変化することが問題となる。

本研究では，2台の2リンク2自由度マニピュレータを対象とし，それらが同一の作業空間上で可動領域に干渉を生じる場合，互いに接触しないための最適な経路を生成することを目的とする。

2台のマニピュレータを互いに干渉せずに目標位置に到達させるために，マニピュレータの関節角度座標で表現したコンフィギュレーション空間[5,6]を用い，その空間内を探索することによりマニピュレータの経路を決定する。

1993年9月27日受理

* 機械システム工学科

探索空間は，関節角度空間に時間的要素を加えると3次元空間での探索になってしまい，計算時間と記憶容量がかなり必要となってしまう。そこで，マニピュレータの姿勢変化による回避ではなく，時間による加減速回避を行わせる。これにより，探索空間は2次元となり計算時間と記憶容量を節約できる。この2次元空間内での探索によって得られた経路から，作業座標上において2台のマニピュレータは衝突を回避できる。

上記の考え方に基づき，マニピュレータの最適な経路を生成する衝突回避経路生成アルゴリズムの開発および計算機シミュレータによる検討を行ったので報告する。

2. 衝突回避経路生成

2.1 問題の設定

複数のマニピュレータを同時に扱う際，お互いの作業が重なって衝突する危険が生じる。その場合に衝突を回避させるための一方法を提案する。本研究では，同一平面上の作業座標系を有する2台の2リンク2自由度マニピュレータを対象として，図2.1のような環境を前提とした。向かって左側のマニピュレータをMアーム，右側のマニピュレータをSアームとする。Mアームの根元を原点とし，2台のマニピュレータは向かい合った格好で設置する。Mアームの根元のリンクをリンク1，先端リンクをリンク2とし，Sアームも同様に根元リンクをリンク3，先端リンクをリンク4とする。リンク1と3を線分とし，リンク2と4には厚みをもたせる。これは，衝突が生じるのを互いの先端リンクだけに限定したためである。また，各リンクは，X軸からの絶対角度で姿勢を示す。各マニピュレータを正面にまっすぐ伸ばした状態を0°とし，X軸より上にいけばプラスの角度，下にいけばマイナスの角度とする。この2台のマニピュレータの手先を与えられた始点Sから目標点Gまで移動させ，その間互いに衝突が生じないように経路を決定する。

2.2 衝突回避経路生成の方針

本衝突回避経路生成アルゴリズムは，以下の条件のもとで開発する。

(1) 平面内の2台の2リンク2自由度マニピュレータを対象とし，各々の先端リンク（リンク2,4）だけ衝突する可能性があるとする。

(2) 衝突回避手法として，関節角度座標によるコンフィギュレーション空間表現を用いる。

(3) Mアームは優先的に与えられた軌道を動き，回避行動はSアームだけに行わせる。

(4) 2台のマニピュレータの空間的経路は，固定とする。

(5) 衝突回避は，Sアームの加減速動作によって行う。

(6) 経路探索は，発見的手法（A*アルゴリズム）[7~10]を用いる。

さらに，以下の入力データは既知として与える。

① 各リンクの長さ，及びリンク2と4の厚み。

② マニピュレータの置かれた位置及び関節の可動範囲。

③ 各マニピュレータの手先の始点，目標点，および姿勢。

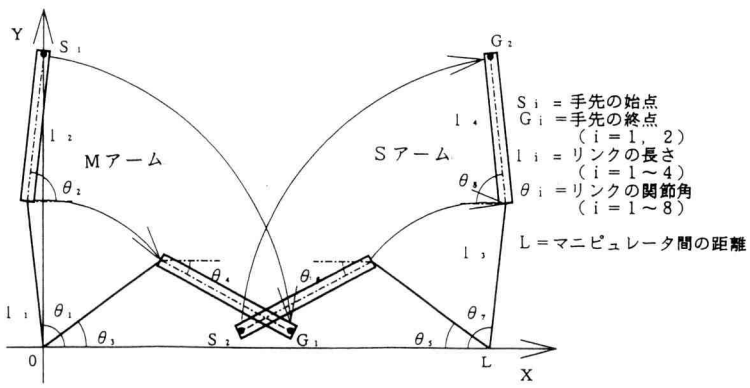


図2.1. 問題の設定

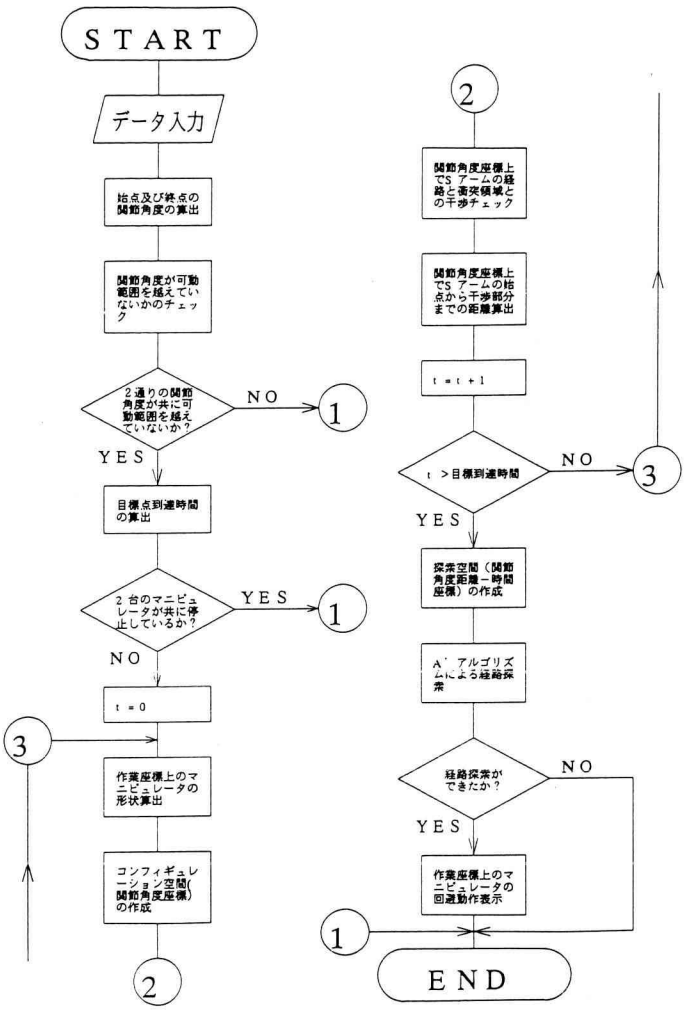


図 2.2.　全体の流れ図

これらの条件と入力データより，図 2.2 に示すような流れで衝突回避を行わせる。

各マニピュレータの手先位置と姿勢のデータ，各リンクの長さのデータを基に逆運動学方程式を解き，始点及び目標点における各関節角度を算出する。このとき，算出された角度が可動範囲を越えてないかを判定する。各マニピュレータの関節角度座標での始点と目標点の偏差より目標点到達時間を求める。

そして，各時間ごとのコンフィギュレーション空間（S アームの関節角度空間）において衝突する領域を求める。コンフィギュレーション空間上の S アームの軌跡と衝突領域との干渉部分を算出する。S アームの経路の始点から干渉部分までの距離と時間を軸とする 2 次元の探索空間を設定する。この探索空間内で，A* アルゴリズムを用い回避経路を生成する。求めた回避経路を基に，作業座標上のマニピュレータの手先を目標位置まで動作させる。

3.　経路探索アルゴリズム

3.1　衝突領域の表現

衝突回避経路計画は，関節角度座標系によるコンフィギュレーション空間上で計画を立てる。コンフィギュレーション空間では，マニピュレータの形状が点

で表現され，マニピュレータの取扱いが容易となる。つまり，コンフィギュレーション空間上で衝突領域と交差せず目標点に到る点経路が得られれば，マニピュレータの非衝突経路となる。しかし，コンフィギュレーション空間では，衝突領域を関数で表現することは難しいことなので，量子化した関節角度値ごとに衝突の有無を判断し，衝突領域を作成する。ここでは，Sアームの関節角度を座標系にとり M アームとの衝突領域を各時間ごとに作る。コンフィギュレーション空間は，横軸をリンク3の絶対角度，縦軸をリンク4の絶対角度とし，横軸をある刻み角度として衝突領域を求める。

また，障害物拡大法により，リンク4の厚みの分だけリンク2の厚みを増やすことでリンク4は線分で表現できる。図3.1(a)(b)にコンフィギュレーション空間の生成を示す。点 (X_{3i}, Y_{3i}) を中心とし，リンク4の長さを半径とする円と，リンク2とが干渉する部分を出し，そのときの θ_{2i} の最大値から最小値が衝突範囲になる。このようにして θ_{1i} のiを0～nまでくり返し，図3.1(b) のような衝突領域をコンフィギュレーション空間内に作成できる。このように，各時間についてそれぞれ衝突領域を算出する。

しかし，このようなコンフィギュレーション空間を

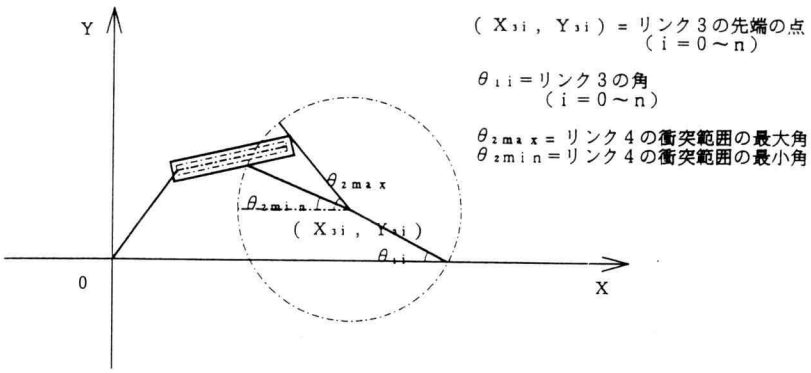


図3.1(a)．時間 t におけるリンク4の衝突範囲

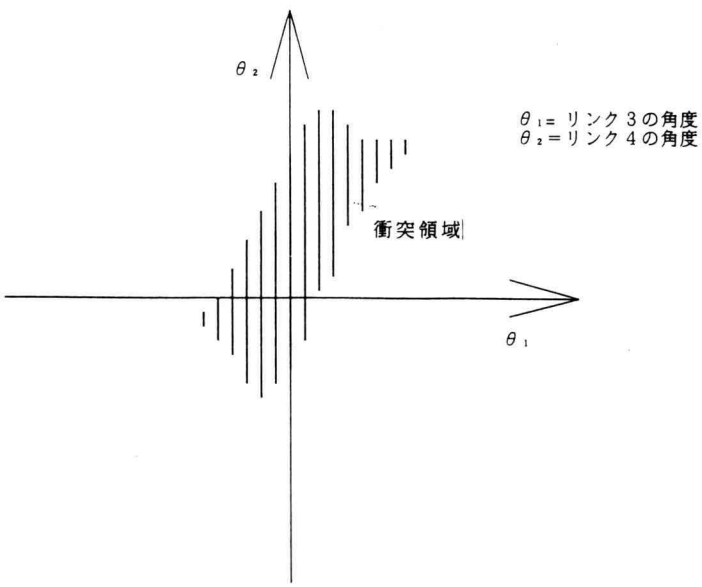


図3.1(b)．コンフィギュレーション空間

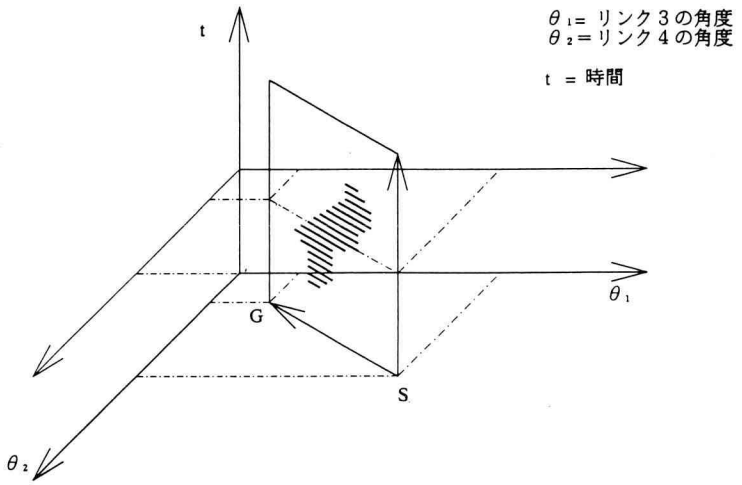


図 3.2.　時間的要素を加えた関節角度空間探索空間

そのまま探索空間にしてしまうと2次元の探索空間に時間軸も加わり3次元空間となり，探索アルゴリズムが複雑になって，探索時間，記憶容量共に増える。そこで，Sアームの経路が関節角度座標において始点と目標点を結んだ直線であると拘束し，加減速による衝突回避を考える。つまり，Sアームの空間的経路を固定とする拘束条件からこのような時間－距離空間での表現が可能となった。これにより，3次元空間での探索を2次元空間の探索にすることができる。

探索空間は，図3.2を図3.3のような2次元空間としたもので，座標系は横軸に時間，縦軸に関節角度座標での距離をとる。具体的には，この探索空間は次のように生成する。まず各時間ごとのコンフィギュレーション空間を用いて，ある時間での衝突領域を求める。

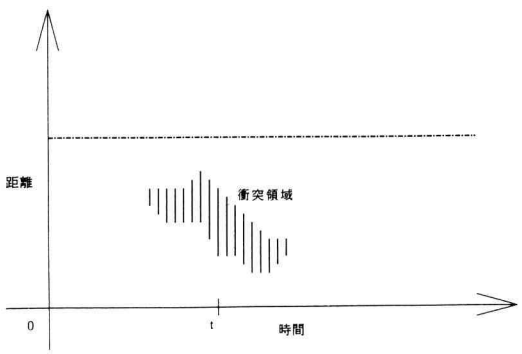


図 3.3.　探索空間

ある時間tにおいて量子化したコンフィギュレーション空間での衝突領域を図3.4のような多角形とし，Sアームの固定した動作経路と衝突領域とが干渉している部分に対して，始点からの距離を求める。これを各時間のコンフィギュレーション空間について行い，図3.3のような2次元の探索空間を作成する。

3.2　経路探索について

探索空間を設定した後は，この空間中で衝突領域を避け目標に到る点経路を生成しなければならない。これには，探索問題でよく行われる発見的手法である

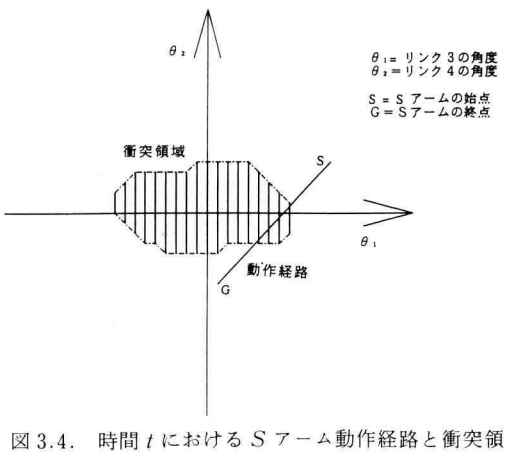


図 3.4.　時間 t における S アーム動作経路と衝突領域との干渉

A* アルゴリズムを用いる。A* アルゴリズムはA アルゴリズムの発展型である。

A アルゴリズム

探索空間上で任意の節点を n としたとき，スタート節点 S から節点 n までの最適な道のコストを $g(n)$ とし，節点 n からゴール節点 G までの最適な道のコストを $h(n)$ とする。この時，スタート節点 S から節点 n を通ってゴール節点 G までの最適な道のコスト $f(n)$ は次式のようになる。

$$f(n)=g(n)+h(n) \qquad (1)$$

すなわち，$f(n)$ は節点 n を通る最適な道のコストの評価値である。従って，節点として候補になっている点のうち $f(n)$ の最小値をもつ open 上の節点（候補として挙がってはいるがまだ展開していない節点）は，最小コスト経路上に存在する可能性のある点である。もし $f(n)$ が正確にわかっていれば，スタート節点 S から $f(n)$ が最小になる節点をたどることによって，ゴール節点 G までの探索に成功する。しかし実際には，$g(n)$ も $h(n)$ も正確にはわからない。そこで $g(n)$ と $h(n)$ を推定して $f(n)$ を求める。これら $g(n)$, $h(n)$, $f(n)$ の推定値を $g'(n)$, $h'(n)$, $f'(n)$ とすると $f'(n)$ は次式のようになる。

$$f'(n)=g'(n)+h'(n) \qquad (2)$$

この式を評価関数として用いる探索の手法をA アルゴリズムという。この手法は，探索効率が高く，最適なルートを見つけながら目標節点 G の探索に成功する。

探索問題は，問題の知識を利用することによって探索を著しく高めることができる。ここでいう問題の知識は，$g(n)$ と $h(n)$ で評価することになる。$h(n)$ は，発見的評価関数といい，探索空間の全体的な性質やゴール節点までの一般的な方向に関する，あるヒューリスティック情報を求める評価関数である。このような情報は，最も可能性のある節点を第一に展開させることによって，探索をゴール節点の方向に“導く”助けとして重要な評価関数となる。$g(n)$ は，距離関数といい，スタート節点 S から節点 n までの最適なルートを求める評価関数である。

A* アルゴリズム

A アルゴリズムでは必ず最適ルートが得られるという保証がない。発見的評価関数の推定値 $h'(n)$ が，実際の発見的評価関数 $h(n)$ より大きいと，$f(n)$ が過大に評価されて，最適ルート上の節点が展開されないで探索が終了してしまい，最適ルートが得られないのである。図 3.5 にこの例を示す。S を展開して子節点 A, B が得られ，その評価関数 f を計算すると

$$f'(A)=g'(A)+h'(A)=1+3=4 \qquad (3)$$
$$f'(B)=g'(B)+h'(B)=2+4=6 \qquad (4)$$

となる。

ここで $f'(A)<f'(B)$ より節点 A が選択される。そして G に到達して，解は S-A-G というルートを取ることになる。しかし，このルートのコストは最小でないのは明白である。S-B-G という最適なルートが見つからなかった理由は，$f'(B)$ が $f(B)$ より大きかったためである。つまり推定値を実際の値より大きくとってしまったためである。

従って，

$$h'(n)\leqq h(n) \qquad (5)$$

という関数を満足している発見的評価関数 $h(n)$ を用

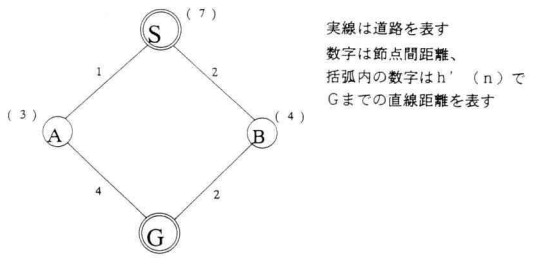


図 3.5.　最適ルートを求められない探索例

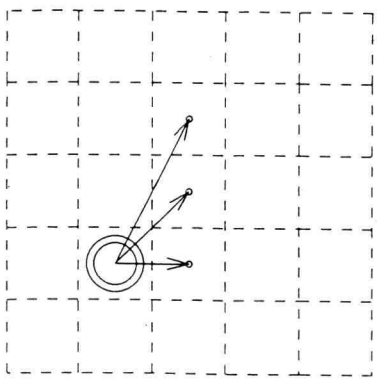


図 3.6.　展開可能な 3 方向について

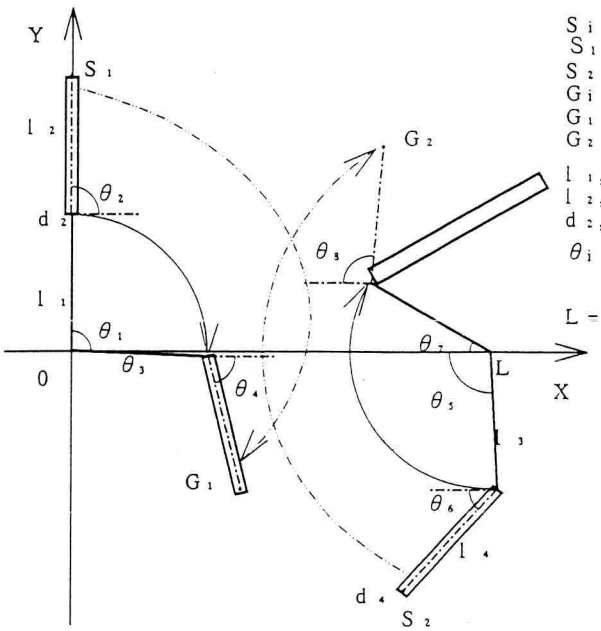
いた探索は，必ず最適解が得られる。このような関係を満たしている評価関数を用いた探索を，A* アルゴリズムによる探索という。

この(5)式の意味するところは発見的評価関数の推定値である $h'(n)$ を最小の値とするということを意味する。探索空間が有限で，スタート点からゴール点までのルートがあれば，最適ルートは必ず存在する。このことから，A アルゴリズムでは最適ルートが見つからない場合もあるが，A* アルゴリズムでは，必ず最適ルートが見つかることになる。

そこで，2次元の探索空間にこの A* アルゴリズムを適用し，点経路の最適ルートを見つける。

展開の拘束条件

探索空間は，横軸が時間を表しているので，展開できる点が限られることになる。つまり，時間は常に正方向にしか進めないので，探索空間の横軸は正方向(右方向）へ1マス分のみの展開となる。また，S アーム



S i = 手先の始点
S 1 = (0, 4 0 0)
S 2 = (4 9 0, - 3 6 0)
G i = 手先の終点
G 1 = (2 5 0, - 2 0 0)
G 2 = (4 5 0, 3 0 0)

l 1, l 3 = リンク1, 3の長さ= 2 0 0
l 2, l 4 = リンク2, 4の長さ= 2 0 5
d 2, d 4 = リンク2, 4の厚み= 1 8

θ i = リンクの関節角度
(i = 1 ～8)

L - マニピュレータ間の距離＝ 6 1 0

図 4.1.　作業座標系での環境

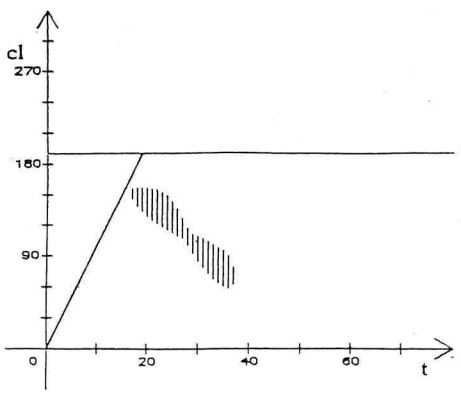


図 4.2(a).　時間を制限しないで距離を評価した場合

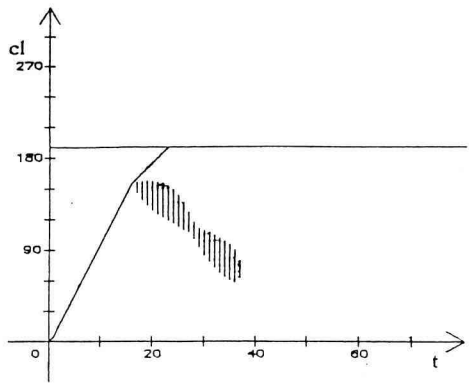


図 4.2(b).　時間を制限しないで速度を評価した場合

の空間的経路が固定であり，常に目標点に向かって動作し戻ることはないため，縦軸に関しても正方向のみの展開となる。よって，任意の節点から展開できる点はその節点から縦軸も横軸も正方向のみでさらに横軸に関しては，右へ1マス分の展開するものとする。縦軸と横軸に関するこのような展開の拘束条件に加えて，Sアームは3種類の動作だけするものとする。その3種類とは，“標準の速度”,“加速して2倍の速度”,そして“停止”である。これにより任意の点からは，3方向のみの展開となり，探索時間の短縮と情報量の節約となる。この展開によって，図3.6に示すように現在点を2重の円としたとき進める3方向は矢印方向となる。

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

本経路計画アルゴリズムの有効性を確認するため，計算機シミュレーションを行った。シミュレーションを行う際の作業座標系での環境を図4.1に示す。

発見的評価関数であるfをいくつか変えて探索を行う。まず，Sアームの目標点到達時間を指定した場合と指定しない場合。さらに，目標点までの距離で評価する場合と速度で評価する場合である。これらの組み合わせで4通りの評価関数が考えられる。具体的な評価関数は次のようである。

1)　Sアームの到達時間は指定しない場合

①　ゴール点までの距離で評価した場合

探索空間上での縦軸はSアームの目標点を表しているので，できるだけはやくゴール点に着くような評価にする。

$$g(n)=|ny-Sy| \tag{6}$$
$$h(n)=|Gy-ny| \tag{7}$$

②　速度を評価した場合

傾きによって評価値を変える。ここでは優先度は，傾き1の時1位，傾き2の時に2位，傾き0の時3位となるようにする。つまり標準の速度では到達不可能なら加速して2倍の速度で移動する。2倍の速度でも到達不可能なら停止し，障害物が通り過ぎるまで待つようにする。

$$g(n)=g(ni)+\mathrm{omomi} \tag{8}$$
$$h(n)=|Gy-ny|+|Gy-ny|*\mathrm{omomi} \tag{9}$$

ただし，

$$\mathrm{omomi}=\left\{\begin{array}{l}\text{傾き1の時0}\\ \text{傾き0の時1}\\ \text{傾き2の時2}\end{array}\right\}$$

とする。

2)　Sアームの到達時間を指定した場合

①　ゴール点までの距離で評価した場合

スタート節点Sから節点nまでの距離とその節点nからゴール節点Gまでの距離の和を求め，常に最短距離を選択するような評価にする。

$$g(n)=\sqrt{(ny-Sy)^2+(nx-Sx)^2} \tag{10}$$
$$h(n)=\sqrt{(Gy-ny)^2+(Gx-nx)^2} \tag{11}$$

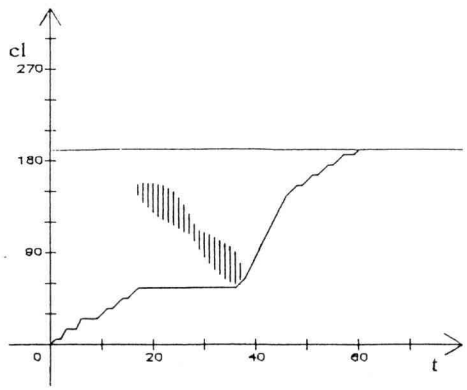


図4.2(c)．　時間を制限して距離を評価した場合

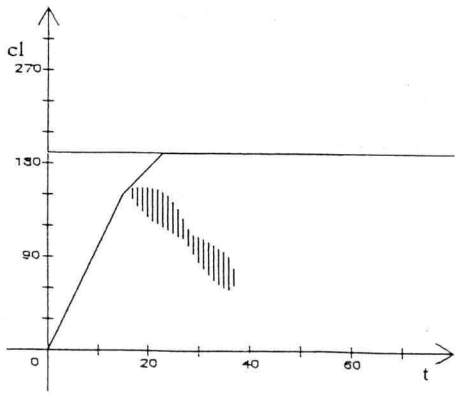


図4.2(d)．　時間を制限して速度を評価した場合

② 傾きに優先度をつけて評価した場合

評価関数 f 及び優先度の付け方は，1)-②と同様である。

4.2 実験結果及び考察

評価関数を4通り変えた場合の探索結果を図4.2(a)～(d)に示す。

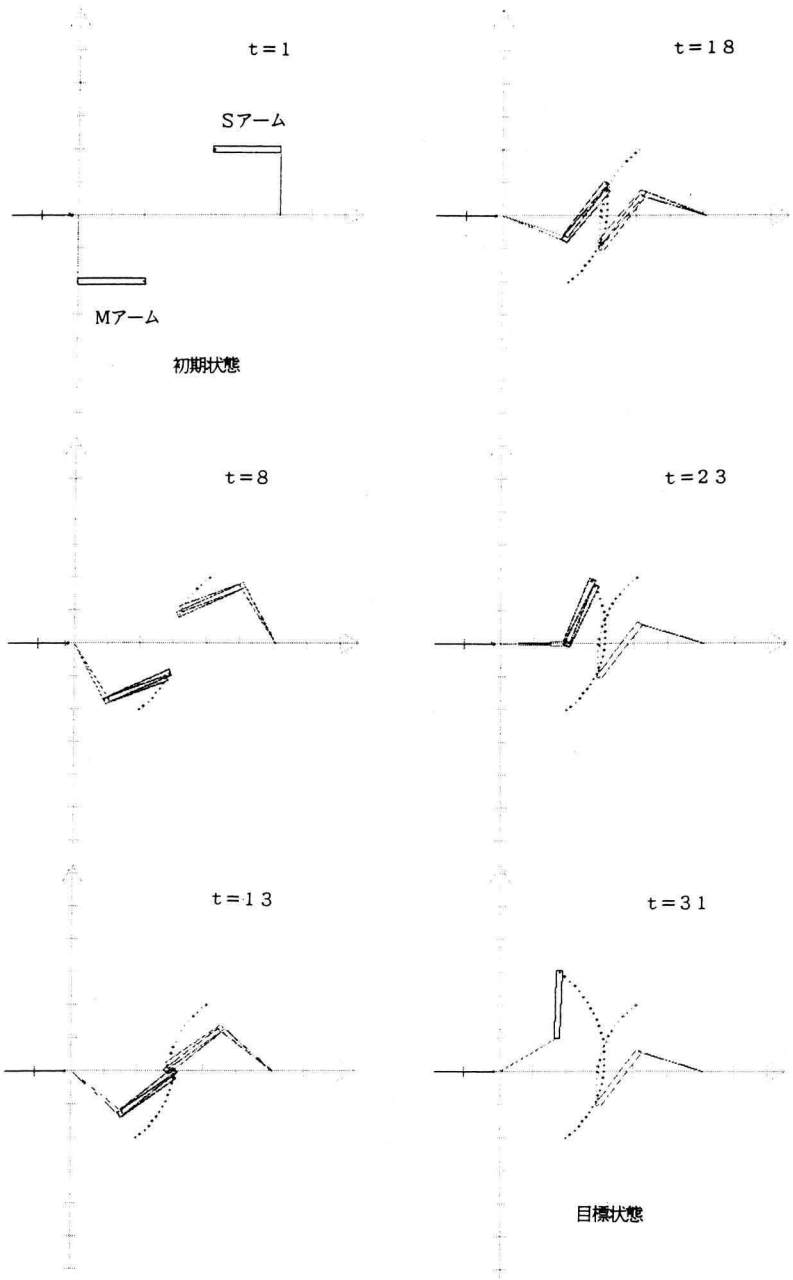


図4.3(a)．基本経路によるマニピュレータ動作

距離を評価値とした場合，常に最短距離を選択しながら探索が進められるので最終的に求められる経路は，スタート点からゴール点までの最短経路である。衝突領域が直線上にないときが最も良い評価となった。直線距離上に衝突領域が存在したら回避動作をするが，できるだけその直線経路から離れないようにスタート点からゴール点までの最短の経路が選択された。探索空間上で生成された経路は不規則な段差がで

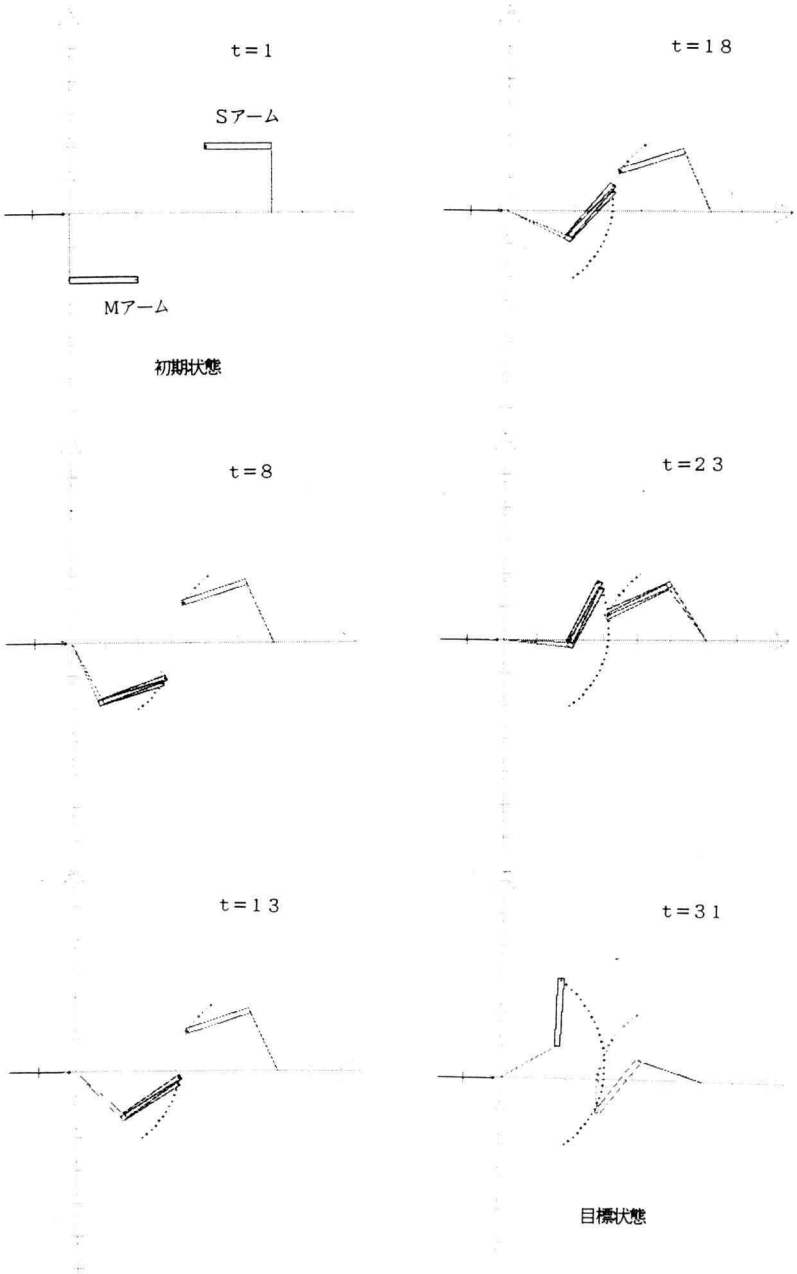


図 4.3(b)．回避経路によるマニピュレータ動作

きてしまい，加減速の連続となってしまう場合があった。スタート点からゴール点を結ぶ直線の傾きが小さいほど顕著である。これは，量子化したためであると思われる。探索空間の量子化を細かくして分割数を増やすことにより対処ができるであろうが実質上，探索時間や情報記憶量の増加の問題により，より細かく分割する事は，難しいと思われる。

速度を評価値とした場合，衝突領域がなければ生成された経路は，スタート点からゴール点を結ぶ直線上に完全に一致する。そして，一定速度で動くことになり，作業空間上では完全に速度変化量が0となる。経路生成中に衝突領域があっても出来る限り優先度の高い節点を選択することにより，越えられなかった衝突領域に対して速度の優先度を保ちつつ，最適な経路が得られた。なお，到達時間の有無による違いはそれほど顕著ではなかった。

また，基本経路と到達時間指定無しで距離を評価した場合の回避経路によるマニピュレータ動作の様子を図4.3(a)(b)に示す。基本経路において衝突が起こっている時間($t=13$)に回避経路ではSアームが停止し衝突を回避していることが分かる。

5. おわりに

2台のアームが同じ作業空間上で作業をする際，相互の衝突を避けるため1台のアームを優先し，もう1台のアームが回避動作をする経路生成アルゴリズムを提案した。そのために時間を考慮したコンフィギュレーション空間を探索空間とし，A*アルゴリズムを用いて最適経路を生成した。また，本手法の有効性を確認するため計算機シミュレーションを行った。そして，以下の結論を得ることが出来た。

(1) 探索空間を作成しA*アルゴリズムを使用することで効率的に非衝突経路を求めることができた。

(2) 経路を一定に拘束した事により探索空間を2次元座標系に設定することができた。

(3) A*アルゴリズムの評価関数を変えることで，異なる動作をさせることが可能であった。

(4) 距離を評価関数とした場合，直線距離が最適なルートとなったが作業空間上ではSアームが頻繁に加減速をした。

(5) 速度を評価関数とした場合，速度変化が少なくてすんだが作業空間上ではSアームとMアームが接近する場合があった。

また，以下のことが今後の課題として残されている。

(1) 今回は片方のマニュピレータ(Mアーム)に優先権をもたせていたが，優先権の判断を確立する必要がある。

(2) 探索空間上での経路は不規則な段差ができてしまう場合があり，これを取り除きスムージングを行う必要がある。

参考文献

1) 登尾啓史：ロボットのパスプラニングアルゴリズムの最近の動向，コンピュートロール，34，pp. 47-56，(1991).
2) 丁　全鋼，湯浅秀男，伊藤正美：関節空間における障害物回避パス探索の一方法，計測自動制御学会論文集，26-2，pp. 219-224 (1990).
3) 井上雄紀，吉村俊秀，北村新三：大局的経路目標とヒューリスティックなグラフ探索を用いたマニピュレータの障害物回避アルゴリズム，計測自動制御学会論文集，27-8，pp. 922-928，(1991).
4) 長谷川："自由空間分類表現法によるマニピュレータの衝突回避動作の計画"，計測自動制御学会論文集，22-6，pp. 616-622，(1986).
5) 長田　正：基礎ロボット工学—知能情報編—，(1986)，昭晃堂.
6) T. Lozano-Perez："Automatic Planning of Manipulator Transfer Movements"，IEEE Trans. on SMC，SMC-11-10，pp. 681-698，(1981).
7) 和多田作一郎：人工知能の理解を深める本，(1986)，実務教育出版.
8) 白井良明，辻井潤一：人工知能，(1984)，岩波書店
9) 合田周平，増田一比古：人工知能—問題解決のシステム論—，(1984)，コロナ社.
10) B. ラファエル，(溝口文雄，内田ユリ子，岩松聰共訳)：考えるコンピューター人工知能入門—，(1986)，近代科学社.