

# 未知空間におけるパスプランニングアルゴリズムを設計するための十分条件とそれにもとづくアルゴリズムの評価†

——ゴールからのユークリッド距離にもとづく経路計画——

登 尾 啓 史††

ロボットを知能化するための課題のひとつに、障害物を回避しながらゴールまで自律的に到達する機能を移動ロボットに持たせることがある。これは洗練されたパスプランニングアルゴリズムを設計することで実現できる。本稿では、未知空間、すなわち障害物の位置や形状が全く判らない作業空間において機能するパスプランニングアルゴリズムを設計するための十分条件を提案する。移動ロボットは基本的にゴール方向へ直進するが、もしそれが障害物によって妨げられたら、その周囲を時計回りか反時計回りに辿る。そして、ゴール方向に向って直進できるうへ、ゴールへ単調に接近できるところから障害物を離れ、再びゴール方向へ直進する。これが移動ロボットを確実にゴールへ到達させる十分条件である。障害物に妨げられない限り移動ロボットはゴール方向に向かって直進するので、障害物から離れるところが単調にゴールへ接近するなら、移動ロボットは衝突できる障害物やその周囲の領域も単調に減少し、最終的にそのようなものは作業空間には存在しなくなり移動ロボットはゴールへ到達する。この十分条件にもとづいたアルゴリズムを利用すると、障害物と接触しているかどうかというローカルな情報のみを頼りに移動ロボットは障害物を回避し確実にゴールへ到達できる。

## 1. はじめに

ロボティクスや計算幾何学等の分野において、移動ロボットのパスプランニング、すなわち、作業空間に存在する障害物を回避しながらロボットを自律的にゴールへ到達させる経路計画が勢力的に研究されており、これまでに種々のパスプランニングアルゴリズムが提案されてきた。

一般に、ロボットの作業空間には種々の形状をした障害物が多数配置されているが、それらの位置や形状があらかじめ判っているなら、それらを組み合わせ的に探索することでスタートとゴールを結ぶパスは確実に選択できる。したがって、そのようなアルゴリズムは多数提案され、主として作業空間の形状の複雑さとパスを選択するときの計算時間のトレードオフについて議論されてきた<sup>1)~4)</sup>。

このようなアルゴリズムを利用するためには、すべての障害物の位置や形状についてのデータをコンピュータに入力しなければならない。これは、工場内のような組織立てられた構造をした空間が対象のときでさえオペレータに多大な負担をかける。もちろん、アウトドアのような組織立てられていない構造をした空

間を対象のときには、そのような障害物に関する情報をあらかじめコンピュータへ入力できない。このことから、障害物に関する情報をコンピュータに入力しなくてもスタートとゴールを結ぶパスが確実に選択できるアルゴリズム、すなわち、障害物の位置や形状が全く判らない作業空間においても機能するパスプランニングアルゴリズムが求められてきた。しかし、これまで、そのようなアルゴリズムの設計は困難だと考えられ、Lumelsky によりそのようなものが2種類提示されるまで全く研究されなかった<sup>5),6)</sup>。

本論文では、Lumelsky のアルゴリズムが未知の作業空間において任意のスタートとゴールを結ぶパス(障害物回避経路)の選択に何故成功したのかを検討し、そのようなアルゴリズムを設計するための十分条件を見出した。この十分条件を利用すると、そのようなパスを確実に選択するアルゴリズムが Lumelsky のもの以外にも自由に設計できることがわかる。また、生成されるパスの長さの観点から、それらのアルゴリズムのいくつかは Lumelsky のものよりも優れていることを示す。

本研究では、ロボットがゴールへ到達するまでいくつかの障害物と衝突したりそれらから離れたりすることに注目する。そして、ロボットがゴールへ確実に到達するための十分条件が、それらの障害物から離れる点とゴールのユークリッド距離が単調に減少することであることを示す。次に、この十分条件を満足するようにパスプランニングアルゴリズムを作成したうへ

† A Sufficient Condition to Design a Deadlock-Free and Collision-Free Path-Planning Algorithm in an Uncertain Workspace and Its Algorithm's Evaluation—Path-Planning Based on the Euclidean Distance toward the Goal—by HIROSHI NOBORIO (Department of Precision Engineering, Faculty of Engineering, Osaka Electro-Communication University).

†† 大阪電気通信大学工学部精密工学科

で、それらが生成するパスの長さや Lumelsky のアルゴリズムが生成するパスの長さを障害物の形状や配置に関して理論的かつ実験的に評価する。

## 2. ロボットが確実にゴールへ到達するパスプランニングアルゴリズムを設計するための十分条件

われわれは、Lumelsky が提案した2つのアルゴリズムを考察したうえで、未知空間で機能するパスプランニングアルゴリズムを設計するのに有効な十分条件を見出した。ここでは、まず、この研究が対象としている移動ロボットや作業空間の障害物について述べ、次に、そのような設定においてパスプランニングアルゴリズムを柔軟に設計するための指針となる十分条件について説明する。

〈作業空間の定義〉 障害物は自由形状でよいが、その周囲長は有限でなければならない。また、障害物はお互いに接触せず、その個数は有限とする。□

〈ロボットの定義〉 移動ロボットは、複数の障害物がなすどのような隙間をも通過できるような点で表現される。移動ロボットは、現在位置やゴール位置を知ることにより、ゴールの方向やそれまでの距離をいつも計算できる。また、移動ロボットは、触覚センサにより障害物との接触を知りそれを迎れる。□

これらのことから、ロボットは3つの動作、すなわち、ゴールへ直進する動作、障害物の周辺を辿る動作、そして停止動作が実行できる。

ロボットは、まず、スタート  $S=L_0$  からゴール方向へと動き始める。そして、障害物に妨げられなければロボットは常にゴール方向へ直進する。そうでなければ、ロボットは衝突した点（以後衝突点  $H_i$  と記す）から障害物を辿り始め、ある点（以後離脱点  $L_i$  と記す）からそれを離れる。このため、作業空間では、衝突点  $H_i$  や離脱点  $L_i$  ( $i=1, \dots, n$ ) が次々と現れる。障害物に妨げられなければ、ロボットはゴール方向に向かって直進するので、離脱点  $L_i$  が単調にゴールへ接近するならば、そこからロボットが衝突できる障害物またはその周囲の領域は単調に減少する。このことから、そのような障害物またはその周囲の領域は最終的には作業空間に存在しなくなり、ロボットは確実にゴールへ到達する。この詳細を定理3で示す。また、ロボットがゴール方向に向かって直進するのを妨げる障害物の周囲では、離

脱点  $L_i$  が必ず見出せることを定理1で示す。さらに、そのような離脱点  $L_i$  が見出せなければ、スタートとゴールを結ぶパスは作業空間に存在しないことを定理2で示す。

【定理1】 ロボットのゴール方向への直進を妨げる障害物の周囲において、次のような性質をもつ離脱点  $L_i$  が必ず見つけられる。

- (1) 点  $L_i$  からゴール方向に向かって直進できる。
- (2) 点  $L_i$  は単調にゴールへ接近する。□

（証明） まず、点  $L_{i-1}$ （最近点  $P^*$ ）から障害物を離れたあとロボットはゴール方向に向かって直進し、障害物と再び点  $H_i$ （最近点  $P^*$ ）で衝突するとする。このとき、衝突点  $H_i$  が離脱点  $L_{i-1}$  よりもゴールに近いことは明らかである。

次に、ロボットは衝突点  $H_i$  から障害物の周囲を辿り始める。このとき、衝突点  $H_i$ （最近点  $P^*$ ）よりもゴールに近く、かつその方向に向かって直進できるような点  $A$  が障害物の周囲において必ず見つけられる。なぜなら、線分  $H_iG$  は障害物周囲と何回か交差するが、それらのうち少なくとも最もゴールに近い点  $A'$  はそのような点  $A$  の候補となるからである。実際には、ロボットは障害物の周囲を連続的に辿るので、点  $A'$  に到達するまでの別の点  $A''$  を離脱点  $L_i$ （最近点  $P^*$ ）として障害物を離れる（図1(a)）。その結果、ロボットは衝突した障害物の周囲で条件(1)や(2)を満足する離脱点  $L_i$  が選択できる。

また、ロボットが障害物を辿るとき、点  $H_i$ （最近点  $P^*$ ）よりもゴールに近いがその方向に直進できないような点  $B$  を見つけても、点  $B$ （最近点  $P^*$ ）よりもゴールに近いうえ、ゴールに向かって直進できるような点  $B'$  が障害物の周囲で必ず見つけられる。なぜな

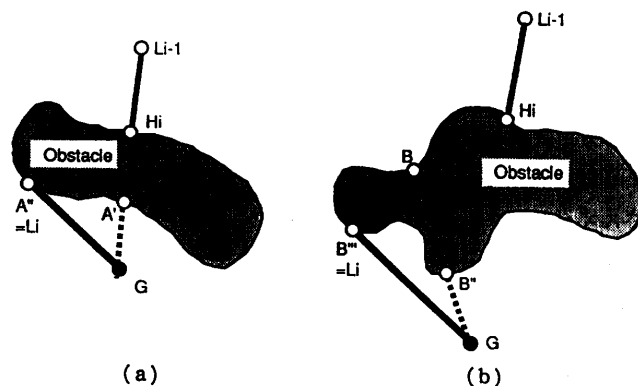


図1 障害物周囲における離脱点  $L$  の選択  
Fig. 1 Selection of a leave point  $L$  around an obstacle.

ら、線分BGと障害物の周囲は何回か交差するが、それらのうち少なくとも最もゴールに近い点 $B''$ はそのような点 $B'$ の候補となるからである。実際には、ロボットは障害物の周囲を連続的に辿るので、点 $B''$ に到達するまでの別の点 $B'''$ を離脱点 $L_i$ (最近点 $P^*$ )として障害物から離れる(図1(b))。その結果、ロボットは衝突した障害物の周囲で条件(1)や(2)を満足する離脱点 $L_i$ が選択できる。

最後に、ロボットはスタート $S=L_0$ (最近点 $P^*$ )からゴール方向に向かって直進するので、初めて障害物と衝突した点 $H_i$ は明らかに点 $L_0$ よりもゴールに近い。また、上記と同じ議論から、衝突した障害物の周囲で条件(1)や(2)を満足する離脱点 $L_i$ が選択できる。したがって、数学的帰納法により、条件(1)や(2)を満足する離脱点 $L_i$ は必ず障害物の周囲で選択される。■

**[定理2]** ロボットがゴール方向に向かって直進するのを妨げる障害物の周囲において離脱点 $L_i$ が見つからなければ、スタートからゴールまでロボットが移動できるパスは作業空間に存在しない。□

**(証明)** 定理1によると、点 $H_i$ で衝突した障害物 $O$ の周囲で離脱点 $L_i$ は必ず見つけれられる。それにもかかわらず、離脱点 $L_i$ に到達できないことは、それらの点 $H_i$ と $L_i$ が障害物 $O$ によって分断されていることを意味する。スタート $S$ から衝突点 $H_i$ までロボットは既に到達しているが、離脱点 $L_i$ からゴール $G$ までロボットが移動できるとしてもスタート $S$ とゴール $G$ は障害物 $O$ によって分断される(図2)。したがって、スタート $S$ とゴール $G$ を結ぶパスは作業空間には存在しないことが理解できる。■

ある障害物によってスタートとゴールが分断されれば、それらを結ぶパスは作業空間には存在しない。われわれのアルゴリズムでは、ロボットが障害物から離れられるかどうかをチェックすることでこのようなケースを判断している。

**[定理3]** スタート $S(=L_0)$ から動き出してゴールに到達するまで、ロボットは点 $H_i$ ( $i=1, \dots, n$ )で複数の障害物と衝突する。このとき、それらから離れる点 $L_i$ ( $i=1, \dots, n$ )が単調にゴールへ接近するなら、ロボットは確実にゴールへ到達する。□

**(証明)** 障害物に妨げられないとき、ロボットは常にゴールへ直進するので、障害物から離れる点 $L_i$ よ

りもゴールから遠い障害物の周囲とロボットは衝突できない。それゆえ、もし離脱点 $L_i$ が単調にゴールへ接近するなら、ロボットが衝突可能な障害物周囲の領域も単調に減少する(図3)。そして、最終的にはそのような領域が存在しなくなり、ロボットはゴールに到達する。■

### 3. パスプランニングアルゴリズム

ここでは、前章の十分条件にもとづいて設計されたアルゴリズムがロボットをどのように移動させるかについて説明する。

未知の作業空間においてスタート $S$ とゴール $G$ を任意に選択したとき、それらが障害物によって分断されていなければ、パスが確実に選択できるアルゴリズムについて説明する。スタートとゴールが障害物によって分断されているときには、どのようなアルゴリズムを以てしてもそれらを結ぶパスは選択できない(図2)。このアルゴリズムは、そのようなケースを自動的に認識し終了する。

ロボットは基本的にはゴール方向に向かって直進す

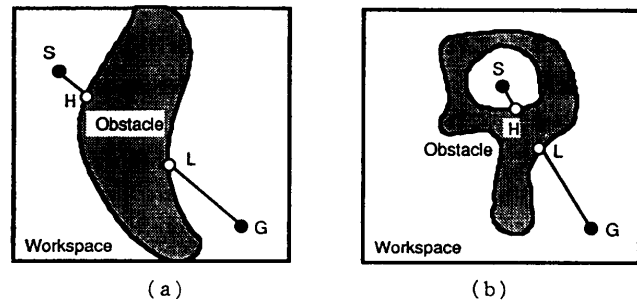


図2 障害物によるスタートとゴールの分断  
Fig. 2 Partition of start and goal points by an obstacle.

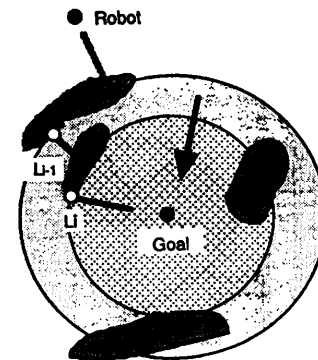


図3 ロボットが衝突できる障害物領域の単調減少  
Fig. 3 Monotonous decrease of obstacle boundary where the robot can hit.

るが、それが障害物によって妨げられたならその周囲を任意の方向に辿り、適当なところから再びゴール方向に向かって直進する。

このアルゴリズムは3つのステップから構成される。スタート $S$ を $L_0$ 、そして $i=1$ とする(図4)。

(ステップ1) 次のうちの1つが成り立つまで、点 $L_{i-1}$ からゴール方向に向かってロボットを直進させる。

(a) ゴール $G$ に到達すればアルゴリズムを終了させる。

(b) 障害物に妨げられれば、衝突点 $H_i$ を定義したうえでステップ2に進む。

(ステップ2) 任意の方向(右回りまたは左回り)に障害物の周囲を辿る。もし、ゴール $G$ に到達すればアルゴリズムを終了させる。そうでなければ、ゴール方向に向かって直進できるうえ、ゴールからのユークリッド距離が距離 $d^*$ よりも小さくなる点を離脱点 $L_i$ とし、ロボットは障害物から離れる。そして、ステップ1に戻る。ここで、ユークリッド距離 $d^*$ を次のように定義する。

(クラス1) これまでに生成されたパスにおけるすべての点とゴールの距離の最小値 $d_1$

(クラス2) これまでに生成されたパスにおけるすべての衝突点 $H_1, \dots, H_i$ とゴールの距離の最小値 $d_2$

(クラス3) これまでに生成されたパスにおけるすべての離脱点 $L_0, \dots, L_{i-1}$ とゴールの距離の最小値 $d_3$

● S=L0

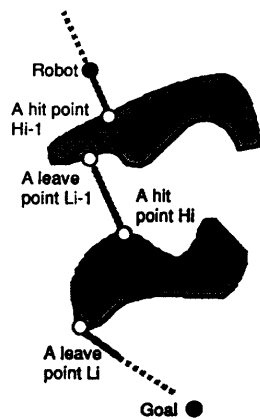


図4 アルゴリズムによる衝突点や離脱点の選択  
Fig. 4 Selection of hit and leave points in a workspace with several obstacles by my algorithm.

(ステップ3) ロボットが点 $L_i$ を見つけられずに点 $H_i$ に戻ってきたら、スタートとゴールは障害物によって分断されている。したがって、それらを結ぶパスは選択できないのでアルゴリズムを終了する。

ユークリッド距離 $d^*$ が単調に減少することにより離脱点 $L_i$ は単調にゴールへ接近し、ロボットは最終的にゴールへ到達する。この距離 $d^*$ をどのように決定するかによって離脱点 $L_i$ が制御できパスの形状を変えることができる。すなわち、(クラス1)から(クラス3)へと推移するにつれて、ユークリッド距離 $d^*$ の減少ペースは緩やかになり、ロボットは障害物から離れやすくなる。これにより、アルゴリズムが生成するパスの長さや形状が変化する(図5)。

なお、ユークリッド距離に関する不等式 $d_1 < d_2 < d_3$ が成立することから、(クラス1)や(クラス2)のアルゴリズムは(クラス3)のアルゴリズムに含まれ、(クラス1)のアルゴリズムは(クラス2)のアルゴリズムに含まれることに注意せよ。

#### 4. われわれと Lumelsky のアルゴリズムが生成するパスの長さの比較

ここでは、生成されるパスの長さの観点からわれわれと Lumelsky のアルゴリズムを比較し、作業空間における障害物の形状や配置に関してそれらの得失を議論する。

われわれのアルゴリズムは前章で説明したとおりであるが、Lumelsky のアルゴリズム Bug1 と Bug2 は以下のようなものである(図6)。

**Bug1:** ロボットは基本的にゴールへ直進する。もし、ロボットの直進が障害物 $O$ によって妨げられたなら、それを一周することによりゴールに最も近い点を

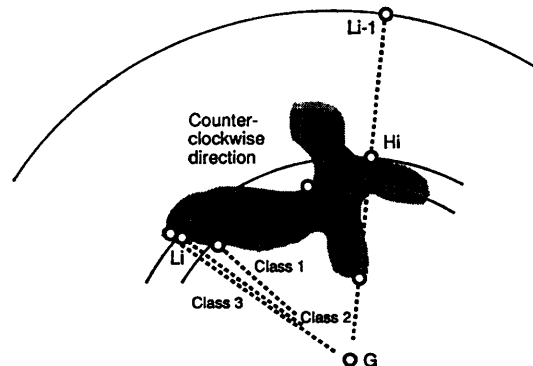


図5 3つのクラスのアルゴリズムが生成するパス  
Fig. 5 Paths made by my algorithms with three classes.

障害物 $O$ の周囲から選択する。そして、そこから再びゴールへ直進する。

**Bug 2:** スタートとゴールを結ぶ直線に沿ってロボットは直進する。このとき、ロボットの直進が障害物 $O$ によって妨げられたなら、衝突した点  $H_i$  から任意の方向（右回りまたは左回り）にそれを辿る。そして、その直線上にあり、ゴール方向へ直進でき、かつ点  $H_i$  よりもゴールに近い点  $L_i$  から再びゴールへ直進する。

まず、ロボットが初めて障害物と衝突する点  $H_1$  は最近点  $P^*$  である。また、その障害物の周囲には点  $H_1$  (最近点  $P^*$ ) を含む無数の点が存在するが、アルゴリズム Bug 1 はそれらのうち最もゴールに近い点を離脱点  $L_1$  として選択する。したがってこの点も最近点  $P^*$  となる。

このようなことがアルゴリズム Bug 1 では帰納的に起こる。すなわち、点  $L_{i-1}$  (最近点  $P^*$ ) から障害物を離れたのちロボットはゴール方向に向かって直進することから、障害物と次に衝突する点  $H_i$  は最近点  $P^*$  となる。また、衝突した障害物の周囲においてゴールに最も近い点が離脱点  $L_i$  として選択されるが、その周囲には点  $H_i$  (最近点  $P^*$ ) が必ず存在するので、離脱点  $L_i$  はいつも最近点  $P^*$  となる。したがって、アルゴリズム Bug 1 では、離脱点  $L_i$  として最近点  $P^*$  が選ばれる。このことから、アルゴリズム Bug 1 はクラス 1 のアルゴリズムの特殊なものであることがわかる。

ここで、障害物 $O$ の周囲で最もゴールに近い点が離脱点  $L_i$  として選ばれることから、ロボットは再び障害物 $O$ とは衝突しない。この特性自体は非常に好ましいものである。しかし、ロボットはそのような最近点を選択するため障害物を必ず一周しなければならない

い。このことから、アルゴリズム Bug 1 はどうしても長いパスを生成するという好ましくない性質をもつ。

一方、衝突点  $H_i$  が単調にゴールへ接近することから、アルゴリズム Bug 2 はクラス 2 のアルゴリズムの特殊なものであることがわかる。アルゴリズム Bug 2 では、ロボットは障害物を一周せずに離脱できるので、アルゴリズム Bug 1 に比べてより短いパスが選択できる。

アルゴリズム Bug 1 は、障害物の周囲で最もゴールに近い点を離脱点として選択する。アルゴリズム Bug 2 は、スタートとゴールを通る直線上で離脱点を選択する。これらのことから、それらのアルゴリズムでは、作業空間において離脱点の候補やその個数が容易に評価できる。これに対して、われわれのアルゴリズムは、障害物周囲のある領域において離脱点を柔軟に選択するので、その候補や個数を正確には評価できない。この結果、柔軟な離脱点の選択はパスの形状をスムーズにしその長さを短くする一方でその評価を困難にしている。

Lumelsky のアルゴリズムでは、ロボットの離脱点が作業空間で固定されるので、最悪のケースでも合理的なユークリッド距離でパスの長さを評価することに成功している。しかし、われわれのアルゴリズムでは、作業空間で離脱点を固定できないので、最悪のケースでは有限だが非常に大きな値でしかパスの長さを評価できない。これは以下の理由による。われわれのアルゴリズムでは、スムーズな形状のパスを生成する目的で、障害物を辿らせながらではなくゴールに直進させながら、ロボットをそれに接近させている。このことから、ゴールに直進できる自由領域が非常に狭いとき、ロボットはなかなかゴールに接近できず、その結果としてパスは非常に長くなる (図 7)。もちろん、障害物の狭間には微少でも自由空間が存在するので、ロボットのゴールへの到達は保証される (パスの長さは有限に抑えられる)。これに対して、Lumelsky のアルゴリズムでは、ロボットは障害物を辿るときもゴールに接近するので、そのようなときでもパスの長さは合理的なものに保たれる。

このような奇妙な形状の作業空間を作ることは困難なことからもわかるように、そのような最悪のケースについての評価は、われわれが日常取り扱う作業空間でのパスの長さを評価することにとって意味はなく、平均的なケースでパスの長さを評価する必要がある。

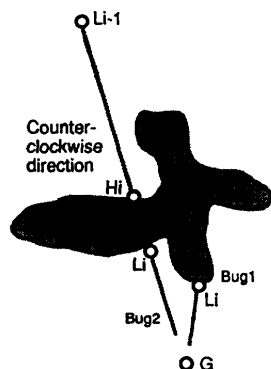


図 6 Lumelsky のアルゴリズムが生成するパス  
Fig. 6 Paths made by Lumelsky's algorithms.

しかし、平均的なケースでパスの長さを正確に評価することはさらに困難である。そこでわれわれは、必ずしも厳密ではないが、そのような平均的な意味でのパスの長さの評価につながる議論を以下に述べる。

任意の形状の障害物が唯一つ存在する作業空間では、離脱点の柔軟な選択によりロボットはその障害物をスムーズに回避できる。例えば、ロボットがある形状の障害物を反時計回りに辿るオーソドックスな設定を考える。このとき、Lumelsky のアルゴリズム Bug 2 では、ロボットは障害物周囲を点 A' または B'' まで辿らなければならないが、われわれのアルゴリズムでは、点 A'' から点 A' または点 B'' から点 B'' の間のある点までしか辿らなくてよく (図 1), その結果としてパスの形状はよりスムーズになる。ただし、スタートかゴールが凹形状の障害物によって完全に囲まれる (外側からは全く見えない) ときのみパスの長さの優位性は逆転する (図 8 (a), (b)). もちろん、ロボットが外側からその障害物と衝突したときにはそのような逆転は起こらない (図 8 (c)). そして、スタートとゴールが凹形状の障害物によって完全に囲まれるケースは、ロボットがスタートするときやゴールするときにはしかおこらないが、ロボットがその障害物に外側から衝突するケースは頻繁におこる。したがって、前者のケースは例外的なものにとらえることができる。

一般に、作業空間には多数の障害物が存在し、ロボットはそれらを順々に回避する。作業空間においてそれらの配置を平均的に考えると、この問題は障害物を 1 つずつ回避する問題の集合として眺められる。このとき、上述のように、われわれのアルゴリズムは Lumelsky のアルゴリズムよりも短い (スムーズな) パ

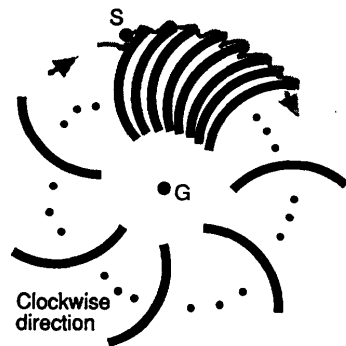


図 7 われわれのアルゴリズムが非常に長いパスを生成する作業空間

Fig. 7 A bad workspace in which my algorithm generates a very long path.

スを平均的に生成する。このような観点から、われわれのアルゴリズムが生成するパスの長さは Lumelsky のアルゴリズムが生成するパスの長さよりも平均的に短くなるのがわかる。

## 5. 実験結果

この章では、各々のクラスのアルゴリズムがどのようなパスを生成するのかをその長さの形状の観点から検討する。また、それらと Lumelsky のアルゴリズムが生成するパスの長さや形状を実験的に比較する。ここでは、自由形状の障害物を容易に表現するため、それらが存在する作業空間を二次元のデジタル画像 [0-511, 0-511] で表現する。そして、ロボットがゴー

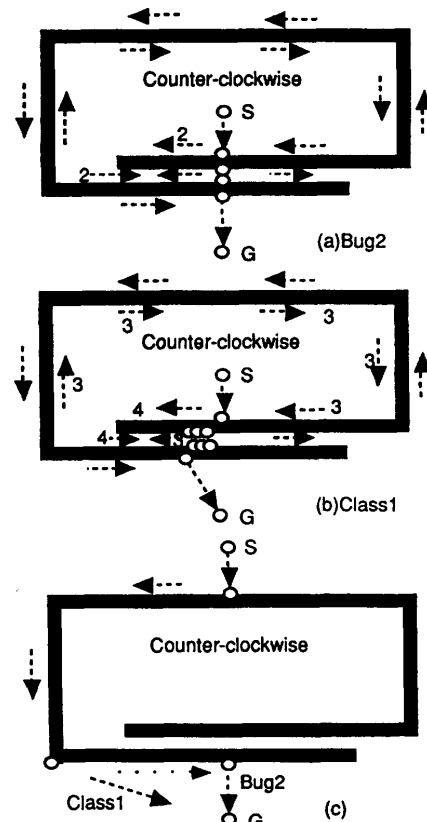


図 8 スタートが障害物に囲まれてパスの長さが逆転するケース (a), (b) と囲まれずにパスの長さが逆転しないケース (c)

Fig. 8 Cases in which my and Lumelsky's path length becomes converse because the start is surrounded by a concave obstacle (a), (b), and a case in which my and Lumelsky's path length does not become converse because the start is not surrounded by the obstacle (c).

ルに直進するときにはそのユークリッド距離, また障害物を迎るときにはデジタル画像における距離を考え, その合計でパスの長さを定義する. デジタル画像では, 上下左右に隣接する画素の距離を1, 右上, 右下, 左上, 左下に隣接する画素の距離を  $\sqrt{2}$  と定義する.

### 5.1 各々のクラスのアルゴリズムが生成するパスの長さや形状の評価

本研究では, スタートとゴールを確実に結ぶパスを生成するアルゴリズムを3つのクラスに分類した. ここでは, それらのアルゴリズムによって生成されるパスの長さや形状を実験的に評価する.

まず, 障害物が散らかった(スタートそしてゴールが障害物によって完全には取り囲まれていない)作業空間において, 3つのクラスのアルゴリズムが生成するパスの長さや形状を評価する(図9, 10, 表1, 2). また, 複雑な形状の障害物が入り組んだ(スタートそしてゴールが障害物によって完全に取り囲まれている)作業空間において, 3つのクラスのアルゴリズムが生成するパスの長さや形状を評価する(図11, 表3).

これらの結果から, 障害物が散らかった作業空間では, 障害物から早く離れてもロボットはそれを回避したことになるので, クラス1からクラス3へと移行す

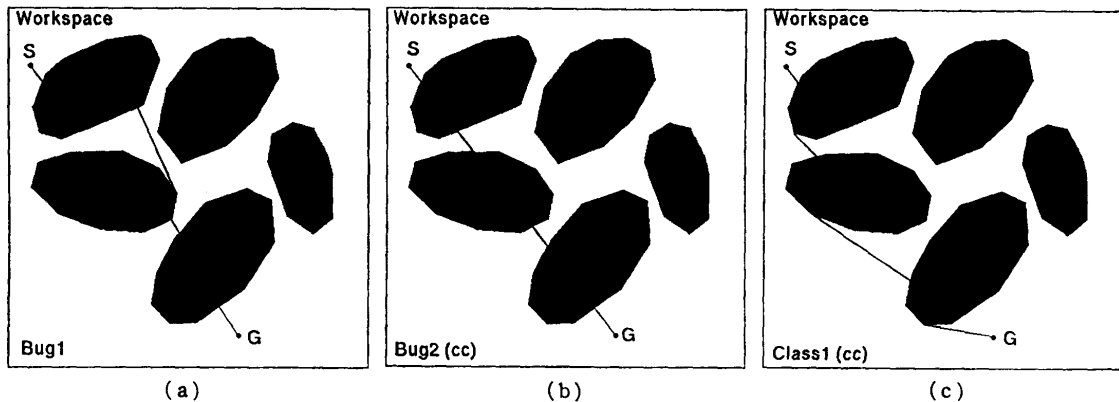


図9 簡単な形状の障害物が散らかった作業空間 W1 においてわれわれと Lumelsky のアルゴリズムが生成したパスの比較, (a) Bug1, (b) Bug2 (counter-clockwise), (c) クラス1 (counter-clockwise)

Fig. 9 Comparison of three paths made by my and Lumelsky's algorithms in a cluttered workspace W1 with simple shape obstacles, (a) Bug1, (b) Bug2 (counter-clockwise), (c) Class1 (counter-clockwise).

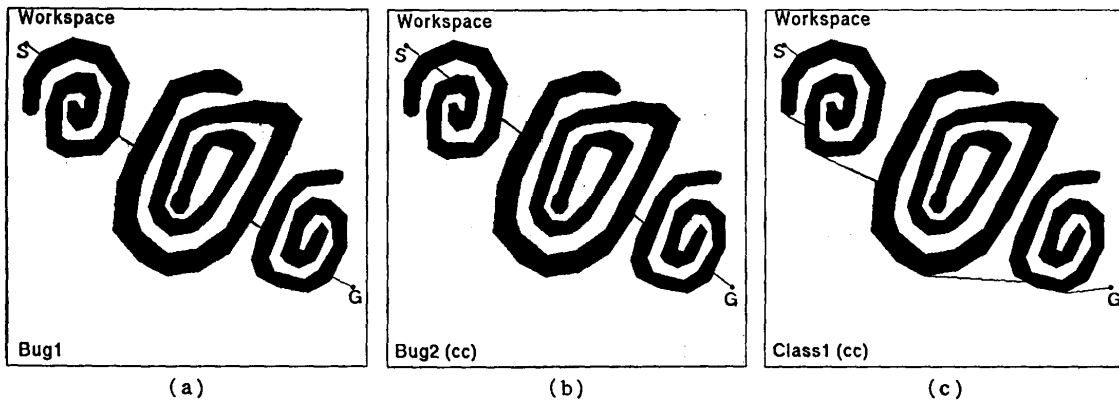


図10 複雑な形状の障害物が散らかった作業空間 W2 においてわれわれと Lumelsky のアルゴリズムが生成したパスの比較, (a) Bug1, (b) Bug2 (counter-clockwise), (c) クラス1 (counter-clockwise)

Fig. 10 Comparison of three paths made by my and Lumelsky's algorithms in a cluttered workspace W2 with complex shape obstacles, (a) Bug1, (b) Bug2 (counter-clockwise), (c) Class1 (counter-clockwise).

るにつれてパスの形状はスムーズになりその長さも短くなる。したがって、そのような作業空間にはクラス3のアルゴリズムが適していることがわかる。これに対して、複雑な形状の障害物が入り組んだ作業空間では、障害物から早く離れてもそれを回避したことにはならないので、障害物を確実に回避したうえでそれから離れるクラス1のアルゴリズムが良いことがわか

る。現実の作業空間はあまり込み入ったものではないことから、クラス3のアルゴリズムが有用であると思われる。

なお、ロボットが障害物を右回りに回避するか左回りに回避するかでパスの長さは大きく異なる。そこで、すべての実験においてパスの長さを両方向に評価している。

表1 作業空間 W1 においてわれわれと Lumelsky のアルゴリズムが生成したパスの長さの比較

Table 1 Length comparison of paths made by my and Lumelsky's algorithms in the workspace W1.

S(30, 80)-G(330, 470)		
	Clockwise	Counter-clockwise
Bug 1		2478.14
Bug 2	1164.15	856.94
Class 1	924.45	637.68
Class 2	924.45	639.23
Class 3	922.71	639.23

表2 作業空間 W2 においてわれわれと Lumelsky のアルゴリズムが生成したパスの長さの比較

Table 2 Length comparison of paths made by my and Lumelsky's algorithms in the workspace W2.

S(25, 50)-G(495, 400)		
	Clockwise	Counter-clockwise
Bug 1		6434.53
Bug 2	1724.76	1269.16
Class 1	786.49	760.05
Class 2	788.17	760.22
Class 3	788.17	760.22

### 5.2 Lumelsky のアルゴリズムが生成するパスとの比較

われわれの分類では、Lumelsky のアルゴリズム Bug1 と Bug2 は、各々クラス1とクラス2のアルゴリズムに属する。これらのアルゴリズムによりパスを生成し前節のパスと比較する (図 9, 10, 11, 表 1, 2, 3)。これらの結果から、障害物がばらばらに存在する作業空間ではクラス3のアルゴリズムが優れているが、複雑な形状の障害物とスタートやゴールが込み入った作業空間ではクラス1のアルゴリズムやアルゴリズム Bug2 が優れていることがわかる。また、アル

表3 作業空間 W3 においてわれわれと Lumelsky のアルゴリズムが生成したパスの長さの比較

Table 3 Length comparison of paths made by my and Lumelsky's algorithms in the workspace W3.

S(100, 130)-G(425, 335)		
	Clockwise	Counter-clockwise
Bug 1		7084.74
Bug 2	2882.00	2150.92
Class 1	3251.56	1685.35
Class 2	3215.24	1685.54
Class 3	23427.80	4569.31

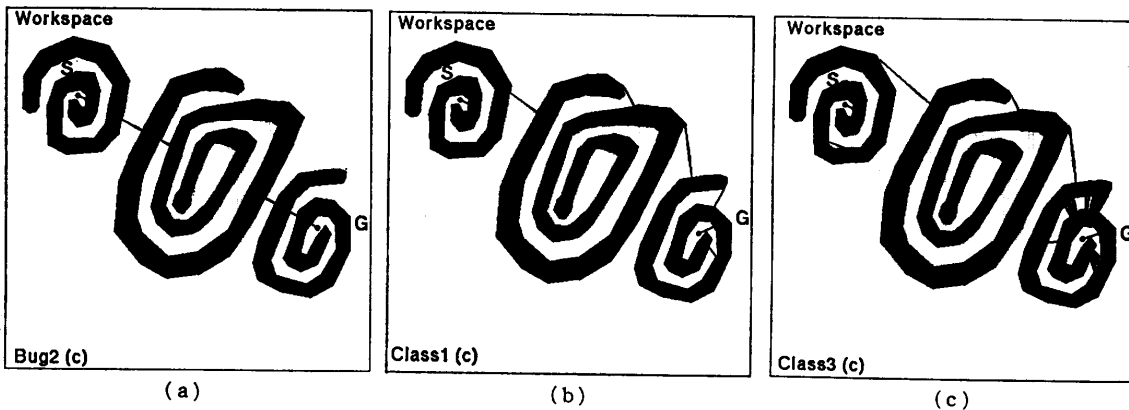


図11 スタートそしてゴールが複雑な形状の障害物に囲まれた作業空間 W3 においてわれわれと Lumelsky のアルゴリズムが生成したパスの比較, (a) Bug2 (clockwise), (b) クラス1 (clockwise), (c) クラス3 (clockwise)

Fig. 11 Comparison of three paths made by my and Lumelsky's algorithms in a complicated workspace W3 with complex shape obstacles, (a) Bug2 (clockwise), (b) Class1 (clockwise), (c) Class3 (clockwise).



ゴリズム Bug 1 は全般的に良くない性質を示すことがわかる。この理由は、障害物の周囲においてゴールから最も近い点を離脱点として選択するため、どうしてもそれを一周しなければならずパスが長くなるからである。

## 6. む す び

ここでは、未知の作業空間において機能する障害物回避経路計画アルゴリズムを設計するために役に立つ十分条件を見出した。この十分条件とは、順次衝突する障害物から離れる点をゴールへ単調に接近させると、ロボットは確実にゴールへ到達するというものである。この十分条件にもとづくと、Lumelsky のアルゴリズムはもちろん、それ以外にも多数のパスプランニングアルゴリズムが自由に生成できる。そして、そのようなもののなかには、Lumelsky のアルゴリズムが生成するパスよりもスムーズで短いパスを生成するアルゴリズムもいくつか存在する。

**謝辞** ワークステーションを利用させていただいた(株)テイジ総合研究所に感謝いたします。また、アルゴリズムのインプリメントにご協力いただいた大阪大学大学院の和住誠一郎氏(現東芝)に深謝いたします。なお、本研究の一部は、文部省科学研究費補助金(奨励研究(A))課題番号 03780039 より予算の援助を得ている。

## 参 考 文 献

- 1) Lozano-Perez, T. and Wesley, M. A.: An Algorithm for Planning Collision-Free Paths among Polyhedral Obstacles, *Comm. ACM*, Vol. 22, pp. 560-570 (1979).
- 2) Brooks, R. A.: Solving the Find-Path Problem by Good Representation of Free Space, *IEEE*

*Trans. on Systems, Man and Cybernetics*, Vol. 13, pp. 190-197 (1983).

- 3) Khatib, O.: Real-Time Obstacle Avoidance for Manipulators and Mobile Robots, *The International Journal of Robotics Research*, Vol. 5, pp. 90-98 (1986).
- 4) Kambhampati, S. and Davis, L. S.: Multiresolution Path Planning for Mobile Robots, *IEEE Journal of Robotics and Automation*, Vol. 2, pp. 135-145 (1986).
- 5) Lumelsky, V. L.: Algorithmic and Complexity Issues of Robot Motion in an Uncertain Environment, *J. of Complexity*, Vol. 3, pp. 146-182 (1987).
- 6) Lumelsky, V. L. and Stepanov, A. A.: Path-Planning Strategies for a Point Mobile Automaton Moving amidst Unknown Obstacles of Arbitrary Shape, *Algorithmica*, Vol. 2, pp. 403-430 (1987).

(平成 3 年 6 月 26 日受付)

(平成 4 年 6 月 12 日採録)



登尾 啓史 (正会員)

昭和 33 年生。昭和 57 年静岡大学工学部情報工学科卒業。昭和 59 年静岡大学大学院工学研究科情報工学専攻修士課程修了。昭和 62 年大阪大学大学院基礎工学研究科物理系専攻博士課程修了。同年大阪大学基礎工学部機械工学科助手。昭和 63 年大阪電気通信大学工学部精密工学科講師を経て、平成 2 年より助教授。ロボットのパスプランニング、ロボットビジョン等の研究を経て、現在ロボット作業をロボストに行うためのアルゴリズムの研究に興味を持っている。工学博士。日本ロボット学会、計測自動制御学会、電子情報通信学会、IEEE 各会員。