

衝突の危険性を評価関数とする移動ロボットの 経路とセンシング点の計画

永谷 圭司*1 油田 信一*2

Path and Sensing Point Planning for Mobile Robot Navigation to Minimize the Risk of Collision

Keiji Nagatani*1 and Shin'ichi Yuta*2

A new path planning method for mobile robots is proposed. When a mobile robot navigates, an error in estimated position by dead-reckoning of the robot increases with the traveling distance. Generally, environment recognizing functions of a robot are limited, and hence the positioning error gives a risk of collision with objects in the environment. To reduce this error, the robot should detect the natural or artificial landmark in the environment by its own sensors, and adjust its estimated position. It is necessary for a mobile robot navigation to adjust its position in suitable intervals. However, there are not so many landmarks to adjust robot's position in real environment. Hence planning the path with good landmarks is necessary for mobile robot navigation to a goal. On the other hand, it is nonsense to take huge cost to detect such landmarks. In this case, trade-off between sensing costs and good landmarks is important.

In this paper, the cost function for the mobile robot's path which minimizes the risk of collision related with sensing cost, and the algorithm to find the optimal path and sensing points are proposed. The authors implemented the proposed algorithm on real robot system, and made several experiments to verify this algorithm.

Key Words: Mobile Robot Navigation, Path Planning, Risk of Collision, Land-marks, Positioning Error

1. はじめに

移動ロボットの自律移動にとって、目的地まで安全に自律走行することは最も重要な基本機能である。この自律走行中、最大の問題となるのが環境中の物体との衝突である。通常移動ロボットは外界センサを用いてロボットの周囲の環境を監視し、衝突の危険を回避する。しかし、移動ロボットの初期位置と環境のモデルが既知である場合、予め目的地までの安全な経路を計画することによっても、衝突の危険を小さくすることができる。本稿では、衝突の危険性を評価関数とした移動ロボットの経路計画法の提案を行う。

環境モデルと初期位置が与えられた場合の移動ロボットの経路計画は、これまでに様々な方法が提案されてきた。例えば、ポテンシャル法を用いた方法 [1]、凸自由空間を用いた方法 [2]、距離変換を用いた経路計画法 [3] などがある。これらの研究の中には、ロボットが最も安全に走行できる経路を求めることを定式化したものもあるが、ここで論じられる安全な経路とは、そのほとんどが単に環境中の障害物からできるだけ離れた経路であると定義されている。

一方、移動ロボットが現在位置を常に把握しながら環境中を安全に走行するためには、現在位置を正確に認識することが重要である。通常この位置認識は外界センサを用いて外界情報を獲得し、ロボットの保持する環境モデルと比較をすることにより実現される。したがってこのロボットの位置認識を正確に行うことのできる経路が最も安全な経路である。本稿では、この立場に立って、移動ロボットの安全なナビゲーション行動の計画という観点から位置認識を行う場所と移動ロボットの走行経路を同時に計画するという問題を定式化し、一つのアルゴリズムを与える [4] [5]。またこのアルゴリズムを用いた経路とセンシング点の計画例及び実環境での実験例を示し、このアルゴリズムが現実的に有意義であることを示す。

2. 自律移動における衝突の危険性

移動ロボットにとって安全な経路を計画するために、ロボットが自律走行中に環境中の物体と衝突する原因について考察する。

まず、予め移動ロボットは環境のモデルを有するものとする。また、ロボットは内界センサを用いて自己位置を推定するドレコニング機能を有するものとする。このとき、移動ロボットが目的地まで走行するために、まず目的地までの経路を計画し(経路計画)、次に計画した経路を走行する(走行段階)とい

原稿受付 1995年2月13日

*1 筑波大学大学院

*2 筑波大学 電子情報工学系

手順をとる。ここで、経路計画は環境モデルという仮想空間内で行われ、走行段階は現実の物理的な空間内で行われる。移動ロボットの衝突は、この走行段階において発生する。したがって、この走行段階において以下の問題に対処する必要がある。

(1) 予想外の障害物

ロボットが保持する環境モデル中に記載されていない障害物が走行経路上に存在する場合、ロボットがその経路を走行することは非常に危険である。

(2) 環境モデルの致命的な欠陥

ロボットが保持する環境モデルに致命的な欠陥があった場合、そのモデルを用いて計画された経路をロボットが走行することは非常に危険である。例えば、計画した経路が実環境中に存在しなければ、当然ロボットは、その経路を走行することはできない。

(3) ロボットの推定位置の誤差

ロボットがデッドレコニングを用いて自己の位置を推定しつつ走行する際、ロボットの実際の位置と推定位置との間には誤差が生じる。走行距離が短い場合、この誤差を無視することはできるが、走行距離が長くなるにしたがって、この誤差は累積される。この誤差のため、デッドレコニングのみを用いて走行する場合、移動ロボットは計画した経路から次第に逸脱していく。このため、ロボットは環境中の物体に衝突する危険が生じる。

(4) 環境のモデリング誤差

環境モデルを作成する際に、実環境を実測しコンピュータ上のモデルに変換する必要がある。通常この作業は人手で行われるため、この変換の際にも誤差が生じる。本稿では、この誤差をモデリング誤差と呼ぶことにする。この誤差のために、ロボットが環境中の物体に衝突する危険も存在する。

ロボットが計画した経路を安全に走行するためには、上記の問題を解決しなければならない。

問題(1)・(2)に関しては、経路の計画段階では、ロボットがどこで障害物と衝突するかは予想できない。したがって走行段階において、できるだけ強力な外界センサを用いて周囲の環境を監視し衝突を回避する以外に、この問題の解決策はない。

問題(3)に関しては、環境中の障害物に衝突する状況は、経路の計画段階である程度予想ができる。なぜならば、ロボットが走行する距離に応じて、ロボットの推定位置の誤差は累積されるという性質を持っているからである。この累積誤差が大きいほど、つまりロボットの走行距離が長いほど、ロボットが環境中の物体に衝突する危険性は増大する。この累積誤差は、環境中のランドマークに注目し、次に示す手法を用いて減少させることができる[6][7]。

- (1) ロボットの持つ環境モデルとデッドレコニングを用いて、ロボットに対する環境中のランドマークの相対位置を推定
- (2) 外界センサを用いてロボットに対するランドマークの相対位置を実測
- (3) ランドマークの推定位置と実測位置の偏差を用いてロボットの推定位置を修正

この動作によってロボットの推定位置の累積誤差を減少させることができ、環境中の物体に衝突する危険を軽減させることができる。この外界センサを用いたロボットの推定位置の修正動作を、位置センシングと呼び、この位置センシングを行う場所をセンシング点と呼ぶ。実環境において移動ロボットを走行させる場合、適当な間隔で位置センシングを行うことは必要不可欠である。しかし、一般に位置センシングの行うことのできる場所は、実環境中にそれほど多くは存在しない。したがって、経路の計画段階において、適当な間隔でセンシング点が存在する経路を計画する必要がある。

問題点(4)に関しても、ロボットが環境中の物体に衝突する危険な場所は部分的に予想可能である。これは、モデリング誤差による衝突の危険性は、走行経路の幅など、経路の状況に依存するためである。この危険性を回避するために、モデリング誤差による衝突の危険性の低い経路を計画する必要がある。

経路の計画段階で衝突の予想が可能なものは、問題(3)と問題(4)である。本稿では、これらの問題に着目し、これらの問題が原因である衝突の危険を最小とする経路を計画する。

仮に、ロボットの周囲の状況を全て把握できる高性能な外界センサとその処理を高速に行うコントローラが、ロボットに搭載されていたとする。このときロボットは、走行しながら自己の現状をリアルタイムに把握することができ、どのような経路であっても安全に走行できる。しかし、そのようなセンサを実現することは非常に難しい。そこで現実的な対応として、制約された技術の下では、「如何に安全な経路を計画するか」ということが重要である。

3. 安全性に基づく経路の評価

移動ロボットが目的地まで走行する際に、環境中の物体と衝突するおそれが最も小さい経路が、ロボットにとって最も安全な経路である。したがってロボットの経路の計画法として、目的地まで到達する各経路について、予測される衝突のリスクを数量化し、それを最小とするものを選ぶことにする。

移動ロボットが走行するにしたがい、デッドレコニングで推定した位置と実際の位置との誤差は増大する。一方、ロボットが保持している環境モデルのモデリング誤差は常に存在する。問題を簡単にするため、ロボットが環境中の物体と衝突する危険性は、上記2つの誤差のみによって生じるものと仮定する。本稿では、この2つの誤差によって生じる衝突の危険が生じることを、衝突の危険性と呼ぶことにする。

3.1 衝突の危険性

衝突の危険性は、ロボットの走行経路と走行距離に依存する。そこでまず、ロボットの各々の位置での衝突の危険性を $u(x)$ とおく。ここで x は、ロボットの走行開始位置からの走行距離である。

さて、ロボットが推定した自己位置の誤差がどれほど大きくても、また、環境のモデリング誤差がいかに大きくても、ロボットが動かない限り衝突は起こらない。つまり、環境中の物体との衝突の危険はロボットが移動することによって生じる。すなわち、計画した経路全体の衝突の危険性 J は、各位置での衝突

の危険性 $u(x)$ の積分値として (1) 式で表すことができる．

$$J = \int_L u(x) dx \quad (1)$$

ここで L は計画したロボットの走行経路である．この経路全体の衝突の危険性 J を最小とする経路が最も安全な経路である．

次にロボットの位置を表す点 x における環境中の物体との衝突の危険性 $u(x)$ を、(2) 式のように分けて考える．

$$u(x) = u_1(x) + u_2(x) \quad (2)$$

ここで $u_1(x)$ は、デッドレコニングで推定したロボットの位置の誤差によって生じる衝突の危険性である．また $u_2(x)$ は、環境モデルのモデリング誤差によって生じる衝突の危険性である．

$u_1(x)$ は (3) 式のように 2 つのパラメータの積で表す．

$$u_1(x) = \eta(x) \cdot E(x) \quad (3)$$

ここで $\eta(x)$ は、 x で表されるロボットの位置から衝突する可能性のある環境中の物体までの距離などロボットの位置に依存するパラメータであり、一般にはこの距離の逆数で表現する． $E(x)$ はその点におけるロボットの推定位置の誤差の大きさを示すパラメータである．これにより、推定位置の誤差が同じであれば、広い道路の方が衝突の危険性は低くなり、また同じ道路であれば、その道路の中心を走行する経路が衝突の危険性の低い経路となる．

$E(x)$ は次の性質を持つ．

- $E(x)$ は走行してきた経路に依存
 x 点における $E(x)$ は、ロボットの走行経路の走行開始地点から現在走行している位置までの走行履歴によって決定される．
- $E(x)$ はセンシング点以外では単調増加
 センシング動作を行わない限り、ロボットの推定位置の誤差は減少することなく累積される．
- センシング点において、 $E(x)$ は一定値まで減少
 ロボットが位置センシングを行うことにより、累積した推定位置の誤差は減少するので、 $E(x)$ は一定値まで減少する．一般には、位置センシングによって更新されたロボットの推定位置にも多少の誤差が含まれるので、 $E(x)$ は 0 にはならない．

$u_2(x)$ は次のような性質を持つ．

- $u_2(x)$ は経路の状況に依存
 $u_2(x)$ は、モデリング誤差によって生じる衝突の危険性なので、環境中のロボットの位置のみに依存する．例えば、ロボットの走行経路が環境中の物体が近ければ、その地点の $u_2(x)$ は大きくなる．

従来の経路計画では、経路のコストを考える場合、走行してきた履歴は考慮されていない．これは、本節で提案した衝突の危険性の観点から考えた場合、 $u_1(x) = 0$ であり、 $u_2(x)$ のみ考慮されることと等価である．しかし、移動ロボットが走行する際、モデリング誤差に比べロボットの推定位置の誤差が遥かに

重要な問題となる．そこで、移動ロボットを目的地まで安全に走行させるためには、本節で提案した衝突の危険性を考慮した評価法を用いて、経路とセンシング点の計画を行う必要がある．

3.2 経路の評価

ロボットの推定位置と実際の位置との間に生じる誤差を減少させるために、ロボットは位置センシングを行う必要がある．しかし、一般にはこの位置センシングにもコストが発生する．例えば、外界センサを用いた外界情報の獲得や処理には時間がかかり、スムーズな走行を妨げる．本稿では、位置センシングによって生じるコストを、センシングコストと呼び、 S で表現する．このセンシングコスト S は、位置センシングを行うたびに増加する．このコストを経路の計画時に考慮することにより、経路上のセンシング点の数を抑えることができ、必要以上に位置センシングを行わないスムーズな走行を実現することができる．

以上より、最適な経路とは、必要以上に頻繁なセンシング動作を行うことなく、なおかつ衝突の危険性を最小とする経路である．このことを考慮し、経路の評価値として、次に示す関数 P を定義し、この関数を最小とする経路を最適な経路とする．

$$P(path) = k_1 J + k_2 S \quad (4)$$

ここで k_1 と k_2 は、衝突の危険性とセンシングコストの重み係数である．例えば、 k_1/k_2 が大きくなれば、センシングコストに対して衝突の危険性が、より重要視される．このため計画する経路中の位置センシングの回数は増加する．これらの重み係数は、ロボットの性能や環境に大きく依存し、一般には経験的に決定される．

筆者らの経験上、典型的な経路の評価値の変化は Fig.1 のようになる．この図からもわかるように、モデリング誤差による衝突の危険性 $u_2(x)$ は走行経路上に常に存在しており、環境中の物体に近い区間においてその値は大きくなる．ロボットの推定位置の誤差 $E(x)$ はセンシング点以外では単調に増加し、センシング点において、一定値まで減少している．推定位置の誤差による衝突の危険性 $u_1(x)$ は、 $E(x)$ とロボットの位置に依存するパラメータ $\eta(x)$ との積なので、位置センシングを行わなければ増加傾向にある．また、センシング点では、センシングコストが発生している． J は $u(x)$ の積分値であり、 S はセンシングコストの合計値なので、経路の評価値 $P(path)$ は、Fig.1 中の色のついていない面積全体の合計値である．

4. 道路ネットワーク環境における経路とセンシング点の計画アルゴリズム

本章では、前章で提案した経路の評価法を用いて、道路ネットワーク環境における経路とセンシング点の計画アルゴリズムを提案する．

4.1 問題の記述

対象とする環境として、屋内の廊下、室内において配置された机や本棚などによって形成された通路、または屋外の道路などを想定する．このような環境を道路ネットワークと呼ぶ．この環境中には、位置センシングを行うことのできるセンシング

点の候補が複数存在するものとする。センシング点の候補とは、ロボットが外界センサを用いて環境中のランドマークを観測することができる場所である。この環境例を Fig.2 に示す。

マークを観測して計算処理を行うことによりその誤差を減少させることができる。

このような環境において、前章で提案した評価法を用いて、どの区間を通り、どのセンシング点で位置センシングを行えば目的地までの安全な経路となるかを考察することが本論文の問題であり主題である。以下この問題に対するセンシング点と走行経路の最適選択アルゴリズムを検討する。

4.2 グラフによる環境の表現

センシング点と走行経路の最適選択を行うために、センシング点の候補地点を含む道路ネットワークを交通流等のシミュレーションなどに使用されるグラフで表現し、グラフ探索によってこの最適選択を行うことにする。まず、この道路ネットワークを以下の手順でグラフに表現する。ただし、ロボットは道路の中央を通るものとする。ロボットから見て左右の境界までの距離が等しい経路が、その道路における最も安全な経路と考えられるからである。

- 交差点，T字路，センシング点の候補，走行開始位置，目的地をノードで表現する
- 各ノード間に走行可能な経路があれば，これらのノードを双方向アークで接続する
- 走行経路のアークに，ノード間の距離や経路の状況などの属性を加える
- センシング点の候補のノードには，位置センシングを行うことを示す閉アークを加える
- 位置センシングのアークにセンシングコストの属性を加える

以上の手順を用いて，Fig.2の道路網ネットワークをグラフで表現したものを Fig.3に示す。位置センシングを示すアークは，Fig.3中のノード6などに示される閉アークである。

ここで作成したグラフを，経路計画用グラフと呼ぶ。この経路計画用グラフを用いることによって，経路とセンシング点の計画は，グラフ上での最適パス探索の問題とみなすことができる。この問題に前章で提案した経路の評価法を適用し，グラフ上での最適パスを発見することによって，移動ロボットの走行経路とセンシング点の計画が行われる。

4.3 走行経路アークにおける衝突リスク

上記の走行経路に相当するアークをグラフ上で探索する際、各アークにおける環境中の物体と衝突するリスクを、

$$J_i = J_{1i} + J_{2i} \quad (5)$$

$$J_{1i} = \int_{arc-i} u_1(x) dx \quad (6)$$

$$J_{2i} = \int_{arc-i} u_2(x) dx \quad (7)$$

Fig. 2 道路ネットワークの例

移動ロボットが走行開始位置から目的地まで達するための最も安全な走行経路を求めるのが本研究の目的である。ここでロボットはデッドレコニング機能を有し、それによって推定された自分の位置を頼りに道路の各区間が走行できるものとする。また、デッドレコニングによるロボットの推定位置は、走行距離に応じて誤差が累積されるが、センシング点においてランド

と表す。ここで i は注目したアークを示す添字である。 $u_2(x)$ は前章で示した通りモデリング誤差に起因する衝突の危険性なので、(7) 式より J_{2i} はアーク i に固有なパラメータと考えることができる。一方 $u_1(x)$ は、ロボットの推定位置の誤差に起因する衝突の危険性である。よって $u_1(x)$ は、移動ロボットがそのアーク i に進入した時点でのロボットの推定位置の誤差に依存

性は,

$$J_{1i} = \alpha_i \cdot E_{i_in} + \beta_i \quad (11)$$

でモデル化される．ここで， α_i, β_i はアーク i に固有のパラメータである．

4.4 位置センシングアークにおけるセンシングコスト

4.2節で示した位置センシングに相当するアークでは，ロボットは実際には位置センシングを行い，デッドレコニングによる推定位置の累積誤差を削減させる．この動作により，推定位置の誤差の評価値 $E(x)$ はある一定の値まで減少する．この値は位置センシングによる位置測定の誤差によって決まる．一方外界センサを用いた外界情報の獲得や処理に時間がかかるなど，位置センシング動作にもコスト s が発生する．そこで位置センシングに相当するアークのコストを k_2s と表現する．一方，位置センシングに要する時間が無視できる場合，または目的地までの到達時間を考慮しなくても良い場合，センシングコスト $s = 0$ となる．このとき，位置センシングの回数を減らす必要は無く，全てのセンシング点で位置センシングを行うこととなる．

5. グラフサーチアルゴリズムと最適パスの決定

3章で提案した経路の評価法をこのグラフサーチに適用する場合，各アークのコスト arc_i は (12) 式で表現される．

$$arc_i = \begin{cases} k_1(J_{1i} + J_{2i}) & : \text{経路} \\ k_2s & : \text{位置センシング} \end{cases} \quad (12)$$

ここで s は一回の位置センシングにかかるコストである．

このコスト関数を用いてグラフサーチを行うことにより，経路とセンシング点の計画を行うことができる．

このグラフサーチで問題となるのは， J_{1i} の値がそれまでのパスに依存するため，アークのコストが一意に決まらないという点である．具体的には，位置センシングを行うことにより推定位置の誤差 $E(x)$ が減少するので，同じアークにおいてもその前に位置センシングに相当するアークを通過したか否かで $E(x)$ の値が異なるため，(11)(12) 式よりそのアークのコストが変わってくる．一方，ダイクストラのアルゴリズム [8] や A* アルゴリズム [9] など従来のグラフサーチアルゴリズムは，各アークのコストは一定であることが条件となっている．したがって，これらの従来のグラフサーチアルゴリズムは，本研究で提案した評価法には適用できない．以上より，このグラフサーチ問題を解くためには，経路計画用グラフの性質を考慮しつつ，新しいグラフサーチアルゴリズムを考案する必要がある．

5.1 経路計画用グラフの性質

上記に示した位置センシング用アークの性質によれば，同じ場所で 2 回以上の位置センシングを行うことは無意味である．また，最適経路であれば，同じ区間を同じ方向へ 2 度走行することはあり得ない．そこでグラフサーチの際には，同じアークは 2 度通過しないという条件を付加することができる．

この条件下では，グラフ上のスタート地点からゴール地点までのパスの候補数は必ず有限となる．したがって，この条件下でフルグラフサーチを行うことにすれば，評価関数を最小とする最適なパスを発見することができる [10] ．

Fig. 3 経路計画用グラフの例

する．したがって (6) 式より， J_{1i} は走行開始位置からこのアーク i に達するまでの走行経路に依存する．

さて，ロボットの推定位置の誤差分散は，走行とともに増大しその増加率は経路の性質によると考えられる．したがって，このアーク i に進入した時点での推定位置の誤差を E_{i_in} ，このときのロボットの位置を x_{i_in} とすると， x の地点における推定位置の誤差 $E(x)$ ，及びこのアーク i を出る地点での推定位置の誤差 E_{i_out} は，次式で表される．

$$E(x) = E_{i_in} + \int_{x_{i_in}}^x \varepsilon(x') dx' \quad (8)$$

$$E_{i_out} = E_{i_in} + \int_{arc_i}^x \varepsilon(x') dx' \quad (9)$$

ただし， $\varepsilon(x')$ はそのアークが示す経路にユニークな推定位置の誤差分散の増加率である．以上より推定位置の誤差による衝突の危険性 J_{1i} は，ロボットから環境中の物体までの距離などロボットの位置に依存する重み関数 $\eta(x)$ を用いて，

$$\begin{aligned} J_{1i} &= \int_{arc_i} u_1(x) dx \\ &= \int_{arc_i} \eta(x) \cdot E(x) dx \\ &= \int_{arc_i} \eta(x) \cdot \left(E_{i_in} + \int_{x_{i_in}}^x \varepsilon(x') dx' \right) dx \\ &= \int_{arc_i} \eta(x) \cdot E_{i_in} dx \\ &\quad + \int_{arc_i} \eta(x) \cdot \left(\int_{x_{i_in}}^x \varepsilon(x') dx' \right) dx \end{aligned} \quad (10)$$

と表される． $\eta(x)$ 及び $\varepsilon(x')$ は，アークに固有の関数なので，ロボットの推定位置の誤差に起因するアーク i での衝突の危険

しかしグラフが複雑な場合、フルグラフサーチでは組み合わせ爆発を起こしてしまう。そこでこのグラフにおけるコストの性質を利用して、グラフサーチを行う時に評価すべきパスを削減する。

探索パス削減の手法

スタートノードからある共通のノードに至る2つのパスがあると仮定し、これらをそれぞれ $path1$, $path2$ とおく (Fig.4 参照)。このとき次の条件を満たしていれば、これ以降のグラフサーチにおいて $path1$ を無視することができる。

$$E_{path1} > E_{path2} \quad \text{かつ} \quad P_{path1} > P_{path2}$$

ここで ' E ' は到達した共通のノードにおける推定位置の誤差の評価値である。また ' P ' は、スタートノードからその共通のノードに至るまでの経路の評価値である。

Fig. 4 共通ノードに至る2つのパス

この手法をグラフサーチに適用することにより、グラフがある程度複雑でも探索経路の組合せ数は相当削減できる。よってフルグラフサーチに比べて、遥かに高速にグラフサーチを行うことが可能となる。

5.2 グラフサーチアルゴリズムの提案

次に、フルグラフサーチに探索パス削減の手法を適用したグラフサーチの具体的なアルゴリズムについて述べる。経路計画用グラフはポインタ接続で表現するものとする。各ノードはスタートノードから直列にポインタ接続される。各ノードに接続するアーク列はポインタ接続で表現され、各アークは接続先のノードのポインタを保持する。このアークの構造体がノードと接続先のノードとの間の経路情報を保持する。また各ノードは、グラフサーチの際に必要なそのノードに至るまでの推定位置の誤差 E 及び経路の評価値 P を保持する。これらの値は経路に依存するため複数存在するので、経路情報と共に RECORD という名の構造体で表現し、ノードにポインタ接続する。これを図示すると、Fig.5のようになる。

このグラフ構造を用いて、提案したグラフサーチアルゴリズムを PASCAL 風に表現したものを Fig.6 に示す。

このアルゴリズムを用いてグラフサーチを行い、ゴールノードに登録された経路のうち、経路の評価値 P が最小となるときの経路が最適パスである。

Fig. 5 経路計画用グラフの表現

```

begin
  node:=start_node ;
  repeat
    arc:=node に接続する先頭アーク ;
    repeat
      next_node:=arcによって接続する node ;
      arcを通った時の next_node における E(x),P(x) を計算 ;
    if next_nodeに登録した RECORD 中の
      (node.E, node.P) において
       $\exists [ (node.E \leq E(x)) \text{ and } (node.P \leq P(x)) ]$ 
      begin
        arc:=next_arc ;
      end
    else begin
      node:=next_node ;
      nodeに (E(x), P(x), nodeまでの経路) を新たに
      RECORD に登録 ;
      arc:=NULL ;
    end
  until arc=NULL ;
  until node=NULL ;
end

```

Fig. 6 グラフサーチアルゴリズム

5.3 探索パス削減アルゴリズムの有効性の検証

次に前節で提案した探索パス削減手法を適用したグラフサーチアルゴリズムの有効性を、いくつかの簡単なグラフの例を用いて検証する。

まず、Fig.7に示す簡単なグラフについて考える。フルグラフサーチを行った場合、アークのコストの計算回数は2172回となった。この数字は各ノードを巡った回数に相当する。探索パス削減の手法を用いることによって、これを61回にまで削減することができた。また、Fig.8に示すグラフでは、フルグラフサーチを用いると計算回数が50280回であった。この回数を、探索パス削減の手法を用いることにより85回まで削減することができた。また、Fig.3に示すグラフ上でフルグラフサーチを行った場合、組合せ爆発をおこしてしまったため、計算回数は計数することができなかった。これに対して、このグラフサーチに探索パス削減の手法を適用すると、1102回の計算で終了した。ただしこれらの計算は、全てSunのワークステーションSparc Station 2上で行った。また、グラフサーチを行うために必要な関数や定数はTable 1を使用した。ここで、 E_{in} は、アークに侵入した地点での推定位置の誤差の評価値である。また A は、位置センシング後の推定位置の誤差の評価値の大きさを表す定数である。なお、 $\eta(x)$ については、問題を簡単に示すため、対象とする道路ネットワークの道幅一定で、かつ経路を道路の中心とおいたので、 $\eta(x) = 1$ とした。これらの結果から、探索パス削減の手法は、グラフの組合せ爆発を抑制していることがわかる。これらの計算結果をTable 2に示す。

Fig.7 簡単なグラフの例—その1

Table 1 関数・定数の定義

$\eta(x)$	=	1
$E(x)$	=	$2x + E_{in}$
$u_2(x)$	=	10
A	=	30
k_1	=	1
k_2	=	1
s	=	100

Table 2 フルグラフサーチと探索パス削減法を用いたサーチとの比較

Example of Graph	Node Num	Arc Num	Full Graph Search	Path Reduction Algorithm
Figure7	8	18	2172	61
Figure8	10	24	50280	85
Figure3	22	58	計算不可	1102

5.4 経路とセンシング点の計画例

次に、探索パス削減の手法を適用したグラフサーチアルゴリズムを用いて、経路とセンシング点の計画を行った例を示す。

例として使用した経路計画用グラフには、Fig.3で示した経路計画用グラフを用いる。また、グラフサーチを行うために必要な関数や定数はTable 1を用いる。

提案したグラフサーチアルゴリズムをFig.3のグラフに適用した結果、Fig.9の経路とセンシング点が計画された。また、計

Fig.8 簡単なグラフの例—その2

画されたパスの $u(x)$ の変遷とこの経路全体のコストを Fig.10 に示す。この例より、提案したグラフサーチアルゴリズムが衝突の危険性を押えるために、適当な間隔で位置修正を行う経路を計画していることがわかる。

Fig. 9 経路とセンシング点の計画結果

Fig. 10 計画した経路の評価値

6. 実環境での経路計画と走行実験

提案した経路とセンシング点の計画アルゴリズムの妥当性を示すため、実環境における移動ロボットの経路計画及びナビゲーションの問題に本手法を適用し実験を行った。

6.1 環境—屋内環境—

対象とした実験環境は、我々の実験室である筑波大学第三学群 E-105 号室とした。この部屋の床面は段差がほとんどなく、車輪型移動ロボットの走行が可能である。また、この環境中には、超音波レンジセンサで検知可能な平らな壁が複数存在する。また、超音波レンジセンサでは検知が比較的困難な障害物、例えば足の細い机や椅子なども複数存在する。

6.2 ロボット—自律型移動ロボット山彦—

本実験では、現在我々の研究室で所有する自律型移動ロボット「山彦」を使用した [11]「山彦」の写真を Fig.11 に示す。山彦は、ほぼ 40cm 立方の小型のロボットで、DC モータで駆動される車輪を用いて走行する。

走行システムには、独立二駆動輪操舵システムを採用している [12]。また、デッドレコニング機能は、モータに付属するエンコーダを用いたオドメトリで実現している。

位置センシングを行うための外界センサとしては、RADIAN センサ (Rotary Acoustic DIrection ANgle sensor) [13] を使用した。このセンサは、一つの超音波送信器と複数の受信器、それにステッピングモータで駆動されるセンサの台から構成される。超音波の送信器・受信器は、分解能が 0.9 度のステッピングモータ上に固定され、センシングする向きを変えることができる。このセンサでは、超音波の伝播時間を計測することにより、ロボットから平らな壁までの距離を測定することができる。また複数の受信器に入ってくる反射波の伝播時間差を用いることにより、センサに対する平らな壁の相対角度を測定することができる。このセンサを用いることにより、次節で述べる手法を用いて位置センシングを行うことができる。

Fig. 11 研究用自律型移動ロボット「山彦」(写真)

6.3 RADIAN センサによる位置センシング法

RADIAN センサによって「山彦」からセンサ前面に存在する壁までの距離と、その壁に対する「山彦」の相対角度を得る

ことができる．このセンサを用いて，環境中の平らな壁をランドマークとして，次に示す手順で位置センシングを行うことにする．

- (1) 計画されたセンシング点で停止
- (2) 山彦が保持する環境モデルとオドメトリで推定した山彦の位置から，平らな壁までの距離と相対角度を推定
- (3) RADIAN センサを用いて平らな壁までの距離と相対角度を実測
- (4) 推定情報と RADIAN センサより得られた情報を比較し，推定したロボットの位置及び向きに誤差を計算
- (5) 推定したロボットの位置及び向きを更新し，計画された経路を走行

6.4 経路とセンシング点の計画

仮定した環境とロボットを用いて経路計画を行った．

まず，仮定した屋内環境を道路ネットワークとみなし，Fig.12 に示すグラフを構築する．センシング点となるノードは，平らな壁の近傍に適当に設定し，経路から見た壁の方向と壁までの距離を属性として与える．ここで作成した環境グラフは，同一環境であれば何回でも使用可能なので，この作業はワークステーション上においてオフラインで行った．

Fig. 12 環境モデルと環境グラフ

グラフサーチを行うための評価関数やパラメータは Table 3 に示すように定めた．

次にロボット上で経路とセンシング点の計画を行うために，このグラフをロボットに転送し，ロボットの初期位置と目的地を与えた．この初期位置と目的地をノードとし，それらが接続

Table 3 関数・定数の定義(実環境)

$\eta(x)$	=	1
$E(x)$	=	$10x + E_{in}$
$u_2(x)$	=	10
A	=	100
k_1	=	1
k_2	=	1
s	=	300000

可能なノードとアークで結ぶことによって，Fig.13に表される経路計画用グラフが作成される．

Fig. 13 経路計画用グラフの作成結果

次に提案した経路とセンシング点の計画のアルゴリズムを用いてロボット上でグラフサーチを行った．この結果の例を Fig.14 に示す．

この結果からわかるように，まず計画したセンシング点は，計画した経路上の全てのセンシング点の候補ではない．これは本アルゴリズムにセンシングコストを導入しているからである．頻りに位置センシングをする必要は無いという観点から，このセンシング点の計画例は妥当であるといえる．また，スタート位置から最初の交差点に行く経路は最適化されておらず，少し遠回りをした経路となっている．これは，環境グラフのノードが適当な位置に配置されていなかったためである．これを解決するには，経路計画用グラフを作成する際に，環境グラフに新たなノードとアークを付け加える必要がある．しかし，グラフのノードが増えれば，探索パス削減アルゴリズムが働いているとはいえ，グラフサーチの時間が必要以上にかかってしまう．今回の経路とセンシング点の計画アルゴリズムを実装するに際

7. まとめと今後の課題

本稿では、ロボットが走行中に環境中の物体に衝突する危険性についてその原因を考察し、ロボットが走行する際の安全性を考慮した経路の評価法を提案した。また、提案した経路の評価法を用いた経路計画アルゴリズムを提案した。しかし、提案した評価法を用いたグラフサーチは、そのグラフにおけるアークのコストが一定とならないために、従来の方法が適用できない。そこで、評価関数の性質を考慮した新しいグラフサーチ法を提案した。次に実ロボットシステムに、提案したアルゴリズムと走行プログラムを搭載する方法を示した。最後に、いくつかの経路計画とその経路の走行実験を行い、ロボットが安全に走行することを確認し、本アルゴリズムの有効性を検証した。

一般に、位置センシングを行った時に修正できる推定位置は、センサの性能やセンシング方法に依存する。例えば、RADIANセンサを使用して、平らな壁を用いた位置センシングを行うことによって、壁に垂直な向きのロボットの推定位置の誤差は減少する。しかし壁に平行な向きのロボットの推定位置の誤差は減少しない。このように推定位置の誤差による衝突の危険性は、一変数で扱えるものでなく、むしろロボットの向きの関数であると考えられる。しかし本稿では、経路計画の問題を簡単にするために、位置センシングをすることによって修正できたロボットの推定位置の誤差の方向に拘らず、衝突の危険性は減少するものと仮定した。つまり衝突の危険性は一変数であると仮定してアルゴリズムを構築した。

今後の課題としては、衝突の危険性を一変数で扱わず、位置センシングを行う時のロボットの向きまで考慮した経路とセンシング点の計画アルゴリズムの構築というものがある。また、種々のセンサに対応した経路計画アルゴリズムについても考えてゆくつもりである。

参 考 文 献

- [1] 池上 孝則・大園 成夫 「距離値ポテンシャルに基づく経路探索」日本ロボット学会第9回学術講演会予稿集(1) pp267-270 1991年
- [2] Maki K.Habib and H.Asama: "Efficient Method to Generate Collision Free Paths for Autonomous Mobile Robot Based on New Free Structuring Approach", in Proc. of IEEE/RSJ Int. Workshop on Intelligent Robots and Systems, pp563-567, 1991.
- [3] A.Zelinsky: "Navigation with Safety", in Proc. of IEEE/RSJ Int. Workshop on Intelligent Robots and Systems, pp331-338, 1989.
- [4] 永谷 圭司・油田 信一 「センサ利用を考慮した移動ロボットの経路計画」第2回ロボットシンポジウム予稿集pp73-78 1992年
- [5] K.Nagatani and S.Yuta: "Path and Sensing Point Planning for Mobile Robot Navigation to Minimize the Risk of Collision", in Proc. of IEEE/RSJ Int. Conference on Intelligent Robots and Systems, pp2198-2203, 1993.
- [6] 小森谷 清・大山 英明・谷 和男 「移動ロボットのためのランドマーク観測計画」日本ロボット学会誌 第11巻 第4号 pp533-540 1993年
- [7] S.Maeyama, A.Ohya and S.Yuta: "Positioning by Tree Detection Sensor and Dead Reckoning for Outdoor Navigation of a Mobile Robot" in IEEE Int. Conference on MFI'94, pp653-660 1994.
- [8] E.W.Dijkstra, "A Note on Two Problems in Connection with graph", in Numerische Mathematik, 1, pp269-271, 1959.
- [9] 白井 義明・辻井潤一 「岩波講座 情報科学 2 2 人工知能」岩波書店 pp38-48 1982年

Fig. 14 経路とセンシング点の計画結果

し、特にこの点には注意を払わなかった。

本稿に示した図例は、作成した経路計画用グラフを用いて机上のワークステーション上で計画したものである。同様の処理はロボット上でも行われるが、ロボットにディスプレイが無いため、ここではワークステーション上で計画した図を使用した。次節で述べる走行実験の際には、目的地の指定から経路とセンシング点の計画まで、全てロボット上で行われた。

6.5 走 行 実 験

計画した経路とセンシング点による移動ロボットの走行実験を行った。ロボットに搭載されているコンピュータ(68000, 10MHZ)を用いて、経路とセンシング点の計画には、およそ3秒を要した。また、計画した経路の長さが約12メートルに対して、ゴールに到達するまでの時間は、およそ80秒であった。この中には、位置センシングを行う時間が含まれており、1回の位置センシングに約5秒かかる。

位置センシングを行う度にロボットは停止するので、ロボットの走行自体は決してスムーズとは言えない。しかし位置センシングを行うことにより、環境中の物体に衝突する危険性は相当押えられ、安全に目的地に到達することができた。

我々は、本稿で提案した新しい経路とセンシング点の計画アルゴリズムと走行プログラムを用いて、この例の他にも数々の計画と走行実験を行った。提案したアルゴリズムで計画した経路を走行する場合、どの実験においてもロボットは目的地まで安全に到達し、この手法の実環境への適用時の有効性が検証された。

- [10] A.V.エイホ・J.E. ホップクロフト・J.D. ウルマン 「データ構造とアルゴリズム」培風館 1987年
- [11] S.Yuta, S.Suzuki, S.Iida : "Implementation of a small size experimental self-contained autonomous robot", in Proc. of Second International Symposium On Experimental Robotics, 1991.
- [12] S.Iida and S.Yuta : "Vehicle Command System and Trajectory Control for Autonomous Mobile Robots", in Proc. of IEEE Int. Workshop on Intelligent Robots and Systems, pp212-217,1991.
- [13] Y.Nagashima and S.Yuta : "Ultrasonic sensing for mobile robot to recognize an environment", in Proc. of IEEE/RSJ Int. Workshop on Intelligent Robots and Systems, pp805-812,1992.



永谷圭司 (Keiji Nagatani)

1968年9月21日生。1993年筑波大学第三学群情報学類を卒業。現在筑波大学大学院博士課程在学中。移動ロボットの経路計画，移動マニピュレータの制御などの研究に従事。

(日本ロボット学会学生会員)



油田信一 (Shin'ichi Yuta)

1948年3月23日生。1975年慶応義塾大学大学院(電気工学)修了。東京農工大電子工学科を経て現在筑波大学電子情報工学系教授。知能移動ロボット，ロボットセンサ，ロボット用コントローラ等の研究に従事。自動制御や信号処理の理論と応用にも興味を持つ。工学博士。IEEE，計測自動制御学会等の会員。

(日本ロボット学会正会員)

List of Figures

1	ある経路における衝突の危険性 $u(x)$ の変遷と経路の評価値	4
2	道路ネットワークの例	4
3	経路計画用グラフの例	5
4	共通ノードに至る2つのパス	6
5	経路計画用グラフの表現	6
6	グラフサーチアルゴリズム	6
7	簡単なグラフの例—その1	7
8	簡単なグラフの例—その2	7
9	経路とセンシング点の計画結果	8
10	計画した経路の評価値	8
11	研究用自律型移動ロボット「山彦」(写真)	8
12	環境モデルと環境グラフ	9
13	経路計画用グラフの作成結果	9
14	経路とセンシング点の計画結果	10