

局所センサ情報を用いた未知環境における複数の移動ロボットの運動計画法

劉 雲輝* 有本 卓*

Motion Planning Based on Local Sensor Information For
Two Mobile Robots Amidst Unknown Environments

Yun-Hui LIU Suguru ARIMOTO

When multiple mobile robots are working in the same environment, it is necessary to plan such motions for the robots that they never collide each other. This paper proposes a useful algorithm for planning a collision-free coordination for two mobile robots in an unknown environment on the basis of local information from their sensors. This algorithm conquers the problem of coordination planning by a two-levels strategy. In the case that no collision between two robots is detected, the robots are independently guided by the motion planning algorithm proposed by V. Lumelsky. However, in the case that a potential collision is expected, their motions are coordinated to avoid the collision. To coordinate the robots, a two-dimensional task space (T-space) and a point automaton are introduced. By controlling the motion of the automaton in the T-space, any collision between the robots can be successfully avoided. Convergence of the proposed algorithm is proved, and its usefulness is shown by a simulation.

Key Words: Coordination Planning, Multiple Robots, Unknown Environment, Local Information and Task Space.

1. はじめに

ロボットの自律化と知能化をはかるためには、ロボットそのものが作業環境にある障害物や他の移動物体と干渉しないような経路及び運動を自律的に計画することが要求される。このようなロボットの経路計画と運動計画の研究が 10 年前から始まって以来、様々な手法が開発されている。作業環境に関する情報が与えられたとき、1 台のロボットの経路を計画するための可視グラフや接線グラフなどによる計画法や人工ポテンシャル法などの多数のアルゴリズムが提案されている^{1)~3)}。障害物に関する情報が分からぬ未知環境で局所的なセンサ情報の下で 1 台の移動ロボットの運動を生成するためには、有

効と思われるアルゴリズムが Lumelsky によって開発されている⁴⁾。

2 次元の環境において、移動ロボットの初期位置と目標位置を結ぶ直線 (M 直線と呼ばれている) は閉じた障害物の境界と有限個の交点で交差する。そして、ロボットは 2 次元空間の障害物の境界を追跡するとき、時計回りと反時計回りの 2 つの追跡方向しか選択できない。これらのことから、環境の障害物に関する情報がないとき、ロボットが M 直線に沿って移動し、または障害物を検出したとき予め決めた追跡方向で障害物の境界をたどって行くと、結局は目標位置に到達できる。Lumelsky はさらにこの方法を 2 リンク平面マニピュレータと 3 リンク直交型マニピュレータの運動計画に拡張した⁵⁾。またこれを改良したアルゴリズムが最近、登尾らによって提案されている⁶⁾。

原稿受付 1991 年 6 月 17 日

* 東京大学工学部計数工学科

しかし、これらのアルゴリズムは1台のロボットの場合にしか利用できない。ロボットの能力の限界から、1台のロボットが複雑な作業に対処しきれないので、複数のロボットの協調で作業を進める機会が工場などで多くなってきている。複数ロボットによる協調作業の効率化と知能化をはかるために、ロボットは互いに干渉しない運動を自律的に計画しなければならない。既知の作業環境に限れば、複数ロボットの経路及び運動を計画するアルゴリズムが幾つか提案されている^{7)~10)}。しかし、未知環境においては、このような研究が見当たらない。

そこで、本論文は、未知環境でセンサからの局所的な情報を用いて2台の移動ロボットがお互いに干渉しない運動を計画するアルゴリズムを提案する。このアルゴリズムは、複数のロボットのタスク空間（Task Space, T空間）を定義し¹¹⁾、その下でロボットの運動を協調している。T空間はそれぞれの経路を移動する複数のロボットの位置で定義される状態空間であり、複数のロボットの移動はT空間における点オートマトンの移動に対応している。このことから、複数のロボットの協調運動はT空間における点オートマトン（システムオートマトンと呼ぶ）の運動計画によって生成できる。

提案するアルゴリズムは、2つのロボットの協調運動の計画をそれぞれのロボットの独立な運動計画と衝突回避との2つの問題に分割している。2つのロボットが互いに衝突する危険がないとき、Lumelskyのアルゴリズムを用いてそれぞれの運動を独立に計画する。Lumelskyのアルゴリズムにしたがうロボットが常に障害物の境界、またはM直線にある。2つのロボット間の衝突危険が検出されたとき、その衝突を回避するために障害物の境界およびロボットのM直線の下で2次元のT空間を定義する。そして、その2次元のT空間でロボットの運動に対応するシステムオートマトンの運動を制御することによって2つの移動ロボットの運動を協調させる。ロボットが実空間の障害物の境界に関する情報を持っていないので、T空間の障害物（T障害物）を予め計算できない。このことから、システムオートマトンの運動計画がまた2次元未知環境の問題となる。T障害物は閉じた領域となるとは限らないので、Lumelskyのアルゴリズムはこのような運動計画に拡張できない。しかし、一般に実空間の障害物が単連結領域で、その単連結性を利用

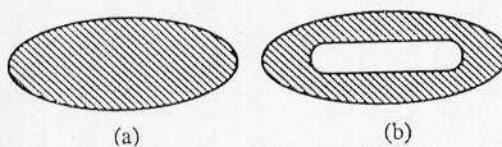


Fig. 1 Simple connected region (a) and non-simple connected region (b).

してT空間でシステムオートマトンがT障害物の境界を追跡していくと2つのロボットの衝突が回避できる。ロボット間の衝突が回避できれば、提案するアルゴリズムがそれを回避するような運動を必ず生成できる。アルゴリズムの収束性をさらに証明し、その有効性をシミュレーション実験で確かめる。

2. ロボットと環境のモデル

作業環境に2つの移動ロボットR1, R2が存在し、それらに対してそれぞれの初期位置と目標位置が与えられている。移動ロボットが簡単な形を持ち、円盤で近似できるとする。そして、それらのロボットが次の仮定を満たしている。

(1) ロボットが自己位置を正確に認識でき、平面上の曲線を精度良く追従できる。

(2) ロボットは作業環境に関する知識を持ってはいないが、それに搭載されているセンサで障害物と他のロボットを認識できる。そのうえ、ロボットのセンサの感知範囲はその境界を一周する他方のロボットの通る領域より大きい。

作業環境は2次元の開領域または閉領域であり、その中に有限個の閉じた障害物領域が存在する。障害物は単連結領域であり(Fig. 1)，そのうえ与えられた移動ロボットが障害物と干渉することなくその境界を一周することができる。このことから、ロボットの姿勢空間においても障害物が単連結となる。一般に障害物は単連結性を満足するので、その仮定はアルゴリズムの応用を大きく制限することとはならないと思われる。

この論文は、このように定義された2つのロボットが如何にセンサ情報の下でお互いに衝突しなくかつ環境にある障害物とも干渉しないで初期位置から目標位置に移動する運動を計画するかを問題にしている。

3. タスク空間とタスク障害物

N台のロボットに対して独立にそれぞれN個の経路を計画したとする。ロボットR_iの経路をP_iで記述し、P_iにおけるロボットR_iの位置を表わす位置座標l_iを次のように定義する(Fig. 2)。l_iの大きさを経路P_iにおける原点Q_iからロボットの現在位置Q_iまでの距離Q_iと定義する。現在位置Q_iが原点O_iより目標G_iに近ければ、l_iは正であり、遠いときは負となる。



Fig. 2 Coordinate defined on the path of robot R_i

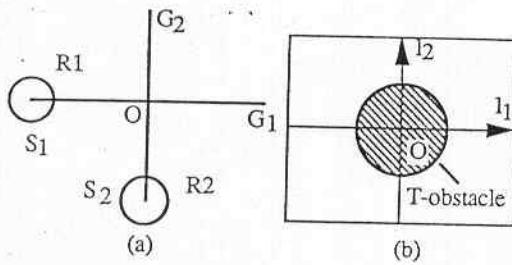


Fig. 3 A case of two robots (a) and the corresponding task space (b)

定義 1 N 個のロボットの状態をそれらの経路におけるロボットの位置ベクトル (l_1, l_2, \dots, l_n) で表現できる。このようなすべての位置ベクトルが定義する N 次元空間をタスク空間 (Task Space, T 空間) と呼ぶ。

文献¹²⁾ではタスク空間を task-completion diagram と呼んでいる。あるロボットの経路に循環路があるとき、その 1 つの位置を無限個の座標で表現できる。このため、 N 個のロボットの位置状態がタスク空間で無限の点に対応するが、タスク空間の点はロボットの唯一の位置状態に対応する。タスク空間の障害物：タスク障害物 (Task Obstacle, T 障害物) は次のように定義する。

定義 2 タスク空間の点 W に対応する位置にある N 台のロボットで、少なくとも 2 台のロボットが互いに干渉していれば、点 W をタスク障害物点という。タスク障害物点の集合がタスク障害物となる。

タスク障害物の定義から、それは作業環境の実障害物からタスク空間への写像ではない。タスク障害物がロボットの経路とその大きさによって決められる。T 空間と実空間の障害物を区別するために、以下では実空間の障害物を実障害物、T 空間の障害物を T 障害物と呼ぶこととする。

例えば、2 つのロボット R_1, R_2 が Fig. 3(a) の直線経路を移動するとき、それらの経路で定義する T 空間が Fig. 3(b) のようになる。 l_1, l_2 がそれらの経路で定義されている座標で、それらの原点が 2 つの経路の交点 O に指定されている。Fig. 3(a) の直線経路を移動する 2 つのロボットの干渉条件は

$$l_1^2 + l_2^2 < r_1^2 + r_2^2$$

である。ここで r_1 と r_2 はそれぞれロボット R_1 と R_2 の半径を表わす。従って、T 障害物は原点 O を中心とする円形領域となる (Fig. 3(b))。

4. アルゴリズムの概要

他のロボットの運動を考えずにロボットの進むべき方向を前進方向と呼ぶ。ロボットが前進方向で移動すると他のロボットと衝突するとき、そのロボットが危険、衝

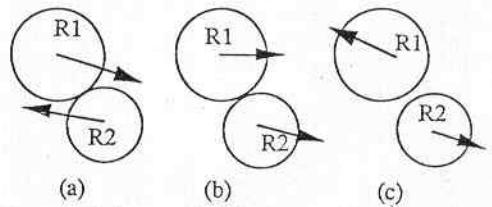


Fig. 4 Definitions of dangerous state (a) and semi-safe state and complete-safe state (c).

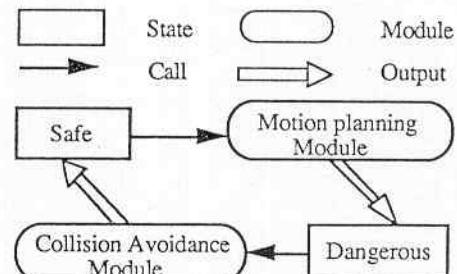


Fig. 5 The coordination planning algorithm.

突しないとき安全と言う。2 つのロボットを持つシステムは、どちらのロボットも危険なとき危険状態 (Fig. 4 (a)), 少なくとも 1 つのロボットが安全なときには安全状態にあると定義する。安全状態をさらに完全安全状態と半安全状態に分ける。半安全状態では 1 つのロボットが安全でもう 1 つのロボットが危険である (Fig. 4 (b))。それに対して、完全安全状態では、2 つのロボットも安全である (Fig. 4 (c))。システムが危険状態にあるとき、どちらのロボットも前進方向に進めないので、それらの運動を協調しなければならない。システムが完全安全であれば、ロボットがお互いに干渉することなく自由に前進できる。そして、システムが半安全であれば、安全なロボットが優先に移動すると、システムの状態が完全安全状態か危険状態かに遷移する。このことから、以下では完全安全状態と危険状態だけを議論する。

2 つの移動ロボットが同じ未知環境で移動するとき、それらに搭載するセンサの情報からそれらの状態を判定できる。提案する協調運動計画アルゴリズムは、2 つのロボットの状態に応じて運動計画モジュールと衝突回避モジュールを用意している (Fig. 5)。システムが完全安全なとき運動計画モジュールはロボットの運動を独立に計画する。それに対して、2 つのロボットが危険状態にあるときには、衝突回避モジュールは T 空間の下でそれらの衝突を回避するようにセントラルコンピュータを用いて互いの運動を協調する。ここでは、ロボットとセントラルコンピュータの通信が常に可能であると仮定する。モジュール間の呼び出しが Fig. 5 に示されている。

5. 運動計画モジュール

未知環境で局所情報を使って1つのロボットの運動を計画するためのアルゴリズムが多数提案されているが、ここで有効と思われるLumelskyのアルゴリズムを利用する⁴⁾。ロボットの初期位置と目標位置を結ぶ直線をM直線といい、M直線が有限個の点で閉じた障害物の境界と交差する。ロボットはその初期位置からM直線に沿って移動すると、M直線と障害物との交点で障害物にぶつかる。そして、ロボットが指定した方向で障害物境界を辿って行くと必ず他の交点でM直線に戻る。これらの戦略がつぎのアルゴリズムにまとめられる。

アルゴリズム1

ステップ1 点Qをロボットの初期位置とする。

ステップ2 ロボットが点QからM直線に沿って移動する。ロボットが目標位置に着いたとき、アルゴリズムが終了する。また、点Aで障害物を検出したとき、ステップ3へ進む。

ステップ3 点Aからロボットが時計回りに障害物の境界を追跡する。点Aより目標位置に近い点BでM直線に戻ったとき、点Qを点Bで更新しステップ2を行なう。

Fig. 6 がこのように計画された経路を示している。アルゴリズム1の収束性がLumelskyによって証明されている。ここで、2つのロボットに対して障害物の追跡方向を同じにすると、それらの出会いの機会を少なくすることが期待できることに注意されたい。

6. 衝突回避モジュール

6.1 衝突回避アルゴリズムの概要

2つの移動ロボットに対応するT空間は2次元であり、

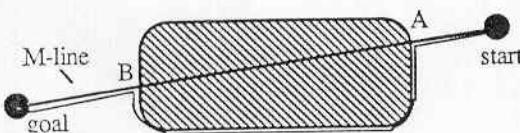


Fig. 6 Motion planning in an unknown environment.

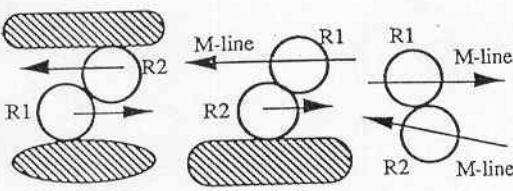


Fig. 7 Possible dangerous patterns when two robots move according to algorithm 1.

それらの運動がT空間でシステムオートマトンの運動に対応する。このことから、提案する衝突回避アルゴリズムは、2次元のT空間でシステムオートマトンの運動を制御することによってロボット間の衝突を回避している。

アルゴリズム1に従って移動する2つのロボットの危険状態は3通りのパターンで検出され得る(Fig. 7)。パターン1では2つのロボットが実障害物の境界にあり、パターン2では1つのロボットがM直線上で移動し、もう1つのロボットが実障害物の境界を辿っている。パターン3では2つのロボットがそれらのM直線上にある。接触パターンにより衝突回避の戦略が違い、それらのパターンに応じてそれぞれアルゴリズム2, 3, 4の3つが用意されている。パターン間の遷移から、アルゴリズム3からアルゴリズム2を、そしてアルゴリズム4からアルゴリズム3を呼び出すことがある。

6.2 実障害物境界上の衝突回避

T空間の定義

時刻 t_1 に2つのロボットが実障害物の境界で危険状態で接触したとする(Fig. 8(a))。それらの衝突を回避するために、実障害物の境界をロボットの経路としてT空間を定義する。それらの経路における位置座標の原点を時刻 t_1 のロボットの位置とし、その正方向を時計回りとする。このことから、システムオートマトンの初期位置がT空間の原点となる。

明らかに、システムオートマトンの目標点はロボット間の衝突を回避した状態に対応しなければならない。実障害物の境界の複雑さによって任意の完全安全状態を衝突回避の判断基準(目標状態)としたとき、Deadlockが起こる可能性がある。このため、次の条件を満足するような完全安全状態を目標状態とする。この安全状態では2つのロボットがお互いに接触し、そのうえ1つのロボットが時刻 t_1 の位置にあり、もう1つのロボットが時刻 t_1 の位置より進んだ(時計回り)位置にある。例えば、Fig. 8(b) は Fig. 8(a) の状態にある2つのロボットに対する1つの目標状態を示している。T空間においては、このような目標状態に対応する点は座標軸の

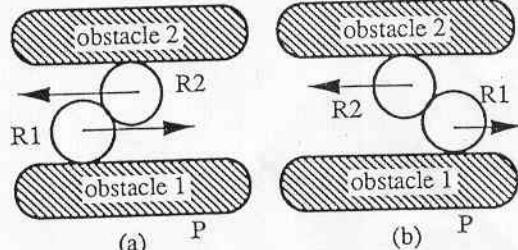


Fig. 8 A dangerous state of two robots on obstacle boundaries (a), and one of the goal states (b).

正半軸にある。このことから、T空間の正半軸で完全安全状態に対応する点をシステムオートマトンの目標点に選ぶ。T空間でシステムオートマトンを原点からその目標点に移動させることでロボット間の衝突回避できる。

例えば、Fig. 10 は Fig. 8 の実障害物の境界で定義したT空間を示している。原点Oはシステムオートマトンの初期位置で、Fig. 8(a) の状態に対応している。そして、点Gはシステムオートマトンの目標位置で Fig. 8(b) に対応している。システムオートマトンが点Oから点Gに移動すれば、ロボット間の衝突が回避される。

衝突の回避

ロボットが実障害物に関する情報を持っていないので、実障害物の境界で定義されたT空間でのT障害物を予め計算できない。システムオートマトンの周辺のT障害物が、その周辺の点に対応する実障害物の境界上の点にある2つのロボット間の干渉チェックによって計算できる。この干渉チェックはシステムオートマトンにとってセンサの役割を果たしている。このことから、システムオートマトンの運動計画は従来の未知環境における運動計画問題に帰着する。しかし、T障害物が閉じた領域になるとは限らないので、開いたT障害物に対して5章の経路計画アルゴリズム（アルゴリズム1）を利用できない。

後で証明されるように、実障害物の単連結性を利用して、システムオートマトンがT障害物の境界を追跡していくと目標点に移動できる。T障害物の境界追跡に対応して、実空間で2つのロボットが互いに接触しながら実障害物の境界をたどる。2つのロボットが同じ実障害物にある時、つぎのような角状態になることがある。

定義3 2つのロボットが同じ障害物の境界にある点A, Bで接觸しているとする (Fig. 9)。1つのロボットは境界線分ABを辿るとき、どの位置においても動かない。もう1つのロボットと交差すれば、2つのロボットが角状態にあるといふ。

角状態にある2つのロボットは引き続き同じ方向で実障害物の境界を追跡していくと、その衝突が回避できなくなる。このため、角状態から脱出するように2つのロボットが接觸しながら逆方向で実障害物の境界を追跡する。これに対応して、システムオートマトンはT障害物の境界を逆たどる。

アルゴリズム2

ステップ1 システムオートマトンは次のことが起こるまで反時計回りにT障害物の境界を追跡する。

(1) システムオートマトンが目標点に到達できたら、アルゴリズムは終了する。

(2) センサの局所的な情報から2つのロボットが角状態になったら、ステップ2へ進む。

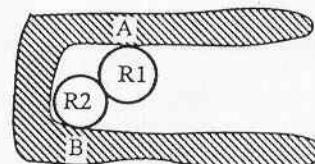


Fig. 9 Definition of a corner state for the robots.

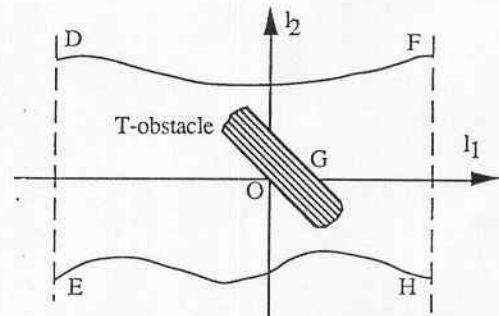


Fig. 10 An example of T-space defined on the boundaries of two obstacles and collision-free curves in the T-space.

(3) システムオートマトンがその出発点に戻ったら、ロボット間の衝突は回避できないとしてアルゴリズムは終了する。

ステップ2 システムオートマトンは目標点に到達するまで時計回りにT障害物の境界を追跡する。それが目標点に着いたら、アルゴリズムが終了する。システムオートマトンが出発点に戻ったら、ロボット間の衝突は回避できないとしてアルゴリズムは終わる。

アルゴリズム2の収束性

パターン1の危険状態に対して2つのケースが考えられる。ケース1では2つのロボットが違う実障害物の境界にあり、ケース2では2つのロボットが同じ実障害物の境界にある。

補題1 2つのロボットが違う実障害物の境界で危険状態になったとき、その衝突が回避できれば、システムオートマトンが任意の方向でT障害物の境界を辿っていくと必ず目標点に到達する。

証明：ロボットR1, R2 がそれぞれ実障害物1, 2の境界で危険状態となったとする (Fig. 8)。まず、閉じた境界曲線で定義されたT空間において、半平面 l_1 (または l_2) と半平面 $-l_1$ (または $-l_2$) の自由領域とT障害物領域の模様が同じとなる。そして、単連結な実障害物に対して、どちらか (実障害物1とする) の境界で次の条件を満足する点Pが存在する。点PにあるロボットR1が実障害物2の境界を辿るロボットR2と干渉し

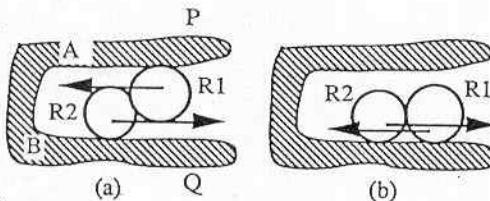


Fig. 11 Possible (a) and impossible (b) dangerous states for two robots on the same boundary.

ない。これに対応して、T空間では直線DE, FHがT障害物と干渉しない(Fig. 10)。また、実障害物のモデルから実障害物2の境界でロボットR2を動かしながらロボットR1が点Pから実障害物1を一周できるので、T空間の上半平面と下半平面にT障害物と干渉しない曲線DF, EHが必ず存在する。これらのことから、システムオートマトンが初期位置(原点)で接触しているT障害物は閉じている。

システムオートマトンは、閉じたT障害物の境界を辿っていけば、ロボット間の衝突が回避できれば必ず目標点に到達する。ロボット間の衝突は回避できないとき、それは出発点に戻ってくる。□

補題2 2つのロボットが同じ実障害物の境界で危険状態となったとき、その衝突が回避できれば、ある方向が存在してシステムオートマトンがその方向でT障害物を追跡して行くと目標点に到達する。

証明: アルゴリズム1で2つのロボットが同じ方向で実障害物を追跡しているので、Fig. 11(b)のような危険状態が生じない。Fig. 11(a)の危険状態に対しては、実障害物の境界では次のような点P, Qが存在する。点PにあるロボットR1が曲線BQを追跡するロボットR2と衝突しない。そして、点QにあるロボットR2が曲線APを追跡するロボットR1と交差しない。ここで、境界曲線ABで2つのロボットが角状態になる。これに対応して、T空間においては直線線分DE, DFが障害物と交差しない(Fig. 12)。

まず、原点Oから線分DEへの経路OHが存在するとき、曲線OHDFを原点に向けて縮小したものはシステムオートマトンがT障害物の境界を辿る経路と一致する。明らかに点Gがシステムオートマトンの目標点の条件を満足する。

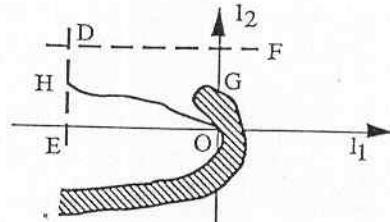


Fig. 12 Collision-free curves in the T-space defined on boundary of the same obstacle.

件を満足する。

次に、原点Oから線分DE, DFへの経路は存在しないときを考える。このとき、実空間で2つのロボットは実障害物の境界の一部しか追跡できない。これに対応して、システムオートマトンがT空間で移動できる領域が有限となる。ロボット間の衝突が回避できれば、その有限の領域でシステムオートマトンが目標点に到達できる。そうでなければ、それは出発点に戻ってくる。□

補題1と補題2から、次の定理が明らかに成立する。

定理1 2つのロボットがパターン1の危険状態で検出されたとき、それらの衝突が回避できればアルゴリズム2はそれを回避する運動を計画できる。

6.3 M直線上と実障害物境界上の衝突回避

T空間の定義

時刻t1にM直線上で移動しているロボットR1と実障害物の境界を辿っているロボットR2が危険状態になったとする。その衝突を回避するために、それらのM直線と実障害物の境界をロボットの経路としてT空間を定義する。時刻t1の状態がT空間の原点に対応するようにロボットの位置座標を定義する。このことから、システムオートマトンがT空間の原点となる。その目標点がパターン1と同じように選択される。

衝突の回避

M直線上のロボットR1がロボットR2の境界を沿って実障害物と干渉することなくM直線上の他の点に移動できれば、2つのロボットが安全となる(Fig. 13)。このような回避動作を回転回避動作と呼ぶ。パターン2の危険状態に対して、回転回避動作とT障害物の境界跡を組み合わせてロボットの運動を協調する。

システムオートマトンがT障害物の境界を辿っているとき、ある時刻t2にロボットR1が実障害物と接触することがある。この時、危険状態がパターン2からパターン1に変化し、アルゴリズム2を呼び出してそれらの衝突を回避する。アルゴリズム1がDeadlockに落ちないように、アルゴリズム2で衝突を回避したときロボットR1はM直線上に乗らなければならない。このため、

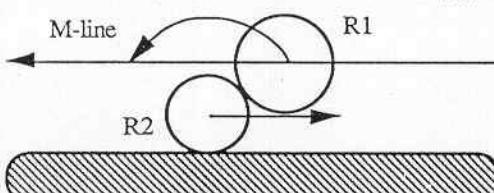


Fig. 13 The rotational avoiding action.

アルゴリズム 2 を行なうためのシステムオートマトンの目標点を次のように設定する。この目標点に対応する安全状態では 2 つのロボットが接触し、そのうえ M 直線にあるロボット R1 が時刻 t2 の位置にある。但し、時刻 t2 に 2 つのロボットが角状態にあれば、このような目標点の設定が難しいので、システムオートマトンが逆方向で T 障害物の境界を追跡していく。

アルゴリズム 3

ステップ 1 T 障害物の追跡方向を反時計回りと指定する。

ステップ 2 システムオートマトンは次のどれかが起こるまで指定した方向で T 障害物の境界をたどる。

(1) システムオートマトンがその目標点に到達できたら、ステップ 4 へ進む。

(2) ロボット 1 が回転回避動作を行なえば、その動作を行ないステップ 4 へ進む。

(3) ロボット 1 が実障害物に接触したとき、2 つのロボットが角状態にあれば、ステップ 3 へ進む。そうでなければ、アルゴリズム 2 を呼び出す。ステップ 4 へ進む。

ステップ 3 T 障害物の追跡方向を時計回りと指定して、ステップ 2 へ戻る。

ステップ 4 アルゴリズムは終了する。

アルゴリズム 3 の収束性

補題 3 パターン 2 の危険状態では、ある方向が存在してシステムオートマトンがその方向で T 障害物を迫っていくと、2 つのロボットは角状態にならない。

証明：2 つのロボットが角状態になったとき、それらが同じ実障害物の境界にある。システムオートマトンがどの方向で T 障害物を迫っていってもロボットが角状態になると、ロボットが追跡している実障害物は単連結ではない。これは実障害物のモデルの仮定に反する。

定理 1 と補題 3 から、明らかに

定理 2 2 つのロボットがパターン 2 の危険状態となつたとき、それらの衝突が回避できればアルゴリズム 3 がそれを回避するような運動を生成できる。

6.4 M 直線上の衝突回避

T 空間の定義

時刻 t1 に M 直線で危険状態になった 2 つのロボットの衝突を回避するために、それらの M 直線をそれぞれの経路として T 空間を定義する。時刻 t1 に 2 つのロボットの状態が T 空間の原点に対応するようにロボットの位置座標を定義する。このことから、システムオートマトンの初期位置が T 空間の原点となる。このパターンでは、2 つのロボットが安全となれば、それらの衝突が回避されたことになる。システムオートマトンの目標点が安全

状態に対応するように選択される。

衝突の回避

パターン 2 と同じように、T 障害物の境界追跡と回転回避動作を制御することで 2 つのロボットの運動を協調させる。

アルゴリズム 4

ステップ 1 システムオートマトンは次のどれかが起るまで T 障害物の境界を反時計回りに追跡する。

(1) オートマトンがその目標点に到達したとき、ステップ 2 へ進む。

(2) どちらかのロボットが回転回避動作を行えば、それを行いステップ 2 へ進む。

(3) どちらかのロボットが実障害物に接触したとき、アルゴリズム 3 を呼びだす。

ステップ 2 アルゴリズムは終了する。

アルゴリズム 4 の収束性はアルゴリズム 3 の収束性から明らかであり、その詳細を省略する。

7. 協調運動計画アルゴリズムの収束性

前章では、各アルゴリズムの収束性を論じたが、大局的にロボットがそれらの目標位置に到達できるかどうかは議論されていない。この章は提案した協調運動計画アルゴリズムの収束性を説明する。

運動計画アルゴリズム（アルゴリズム 1）の収束性からそれで独立に生成されるロボット R_i の経路が有限の長さである。その経路に沿って時刻 t にロボット R_i ($i = 1, 2$) の位置から目標位置までの距離を $D_i(t)$ で表す。 $D_1(t)$ と $D_2(t)$ がゼロに収束すれば、ロボットがそれらの目標位置に到達する。

定理 3 アルゴリズム 1 を行なうとき、 $D_1(t)$ と $D_2(t)$ は単調に減少する。

証明： $D_1(t)$ と $D_2(t)$ の定義から定理の成立は明らかである。□

定理 4 時刻 t_1 に 2 つのロボットがパターン 1 で検出されたとする。アルゴリズム 2 にしたがって時刻 t_2 にその衝突を回避したとすれば、 $D_1(t_2) \leq D_1(t_1)$ と $D_2(t_2) \leq D_2(t_1)$ 。なお、等号は同時に成り立たない。

証明：パターン 1 でシステムオートマトンの目標点の選び方から、定理の成立は明らかである。□

定理 5 2 つのロボットが時刻 t_1 にパターン 2 で検出されたとする。ある時刻 t_2 が存在して $D_1(t_2) < D_1(t_1)$ と $D_2(t_2) < D_2(t_1)$ 。

証明：一般性を失わずに時刻 t_1 にロボット R1 が M 直線線分 CD に、ロボット R2 が実障害物の境界にあるとする。ここで、点 C, D はロボット R1 の M 直線と実障害物の交点である。そして、アルゴリズム 3 で時刻

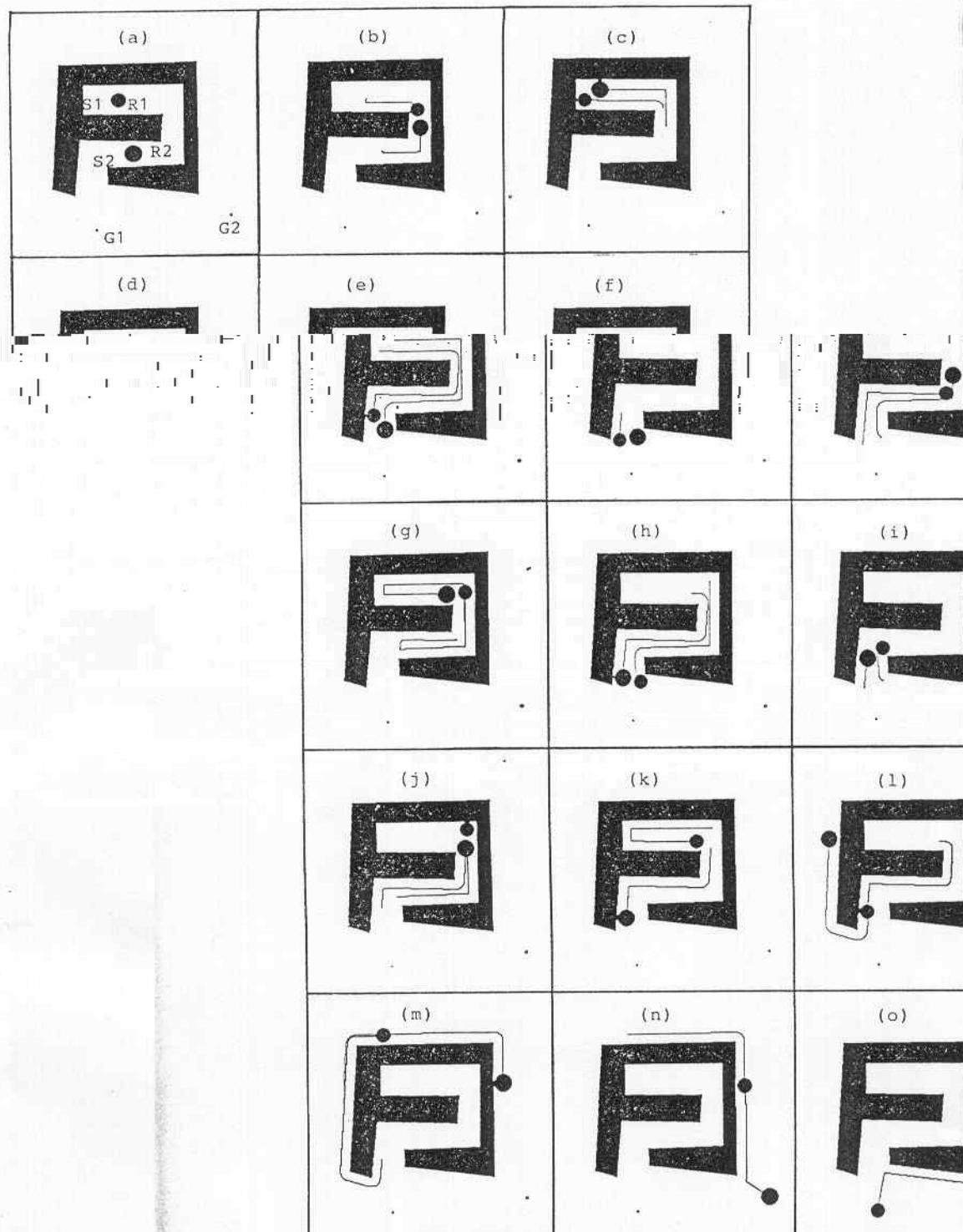


Fig. 14 Motion of two mobile robots in an unknown environment

在して任意の $t > T$ に対して $D_1(t) = 0$ と $D_2(t) = 0$.

8. シミュレーション

提案したアルゴリズムの有効性を示すために、局所的な情報を用いて2つの移動ロボットの運動計画のシミュレーションを行なった。シミュレーションに利用した環境とロボットが Fig. 14 に示されている。Fig. 14(a) はロボット R_i ($i=1, 2$) の初期位置 S_i と目標位置 G_i を、Fig. 14(b) は2つのロボットの危険状態を示している。Fig. 14(b) から Fig. 14(f) までは衝突回避アルゴリズムに従うロボットの運動を表している。Fig. 14(c) で2つのロボットが角状態にあり、Fig. 14(f) では2つのロボットが衝突を回避した状態にある。図にある線分が1つ前のシーンからこれまでロボットが移動した経路を表示している。Fig. 14(b) から Fig. 14(f) までのロボットの運動に対応して、システムオートマトンが辿った経路は Fig. 15 に示されている。Fig. 15 の b, c などの点は Fig. 14(b), Fig. 14(c) などの図に示されているロボットの状態に対応する。Fig. 15 から分かるように、システムオートマトンがまず反時計回りに T障害物の境界を辿り、角状態に対応する点 c から T障害物を逆走する。Fig. 14(g) はもう1つの危険状態、Fig. 14(j) はその衝突を回避した状態を示している。

シミュレーションの結果から分かるように、提案したアルゴリズムは未知環境で作業を行なう2台の移動ロボットの運動計画に有効である。

9. 終わりに

本論文は、複数のロボットのタスク空間(Task Space)を定義し、その下で未知環境で作業を行なう2台のロボットの協調運動を計画するアルゴリズムを提案した。複数ロボットのタスク空間を導入することにより、ロボットの協調作業の計画はタスク空間における点システムオートマトンの運動計画によって実現できる。このことから、従来の1台のロボットのための経路計画アルゴリズムを複数ロボットの協調作業の生成に利用することが可能となる。今回提案した未知環境における2つのロボットの協調運動計画アルゴリズムはその1つの例である。提案したアルゴリズムは、2つロボットが衝突する危険が検出されたとき、定義した2次元タスク空間でシステムオートマトンがT障害物の境界を辿ることでロボット間の運動を協調しそれらの衝突を回避している。作業環境に存在する障害物は単連結であれば、アルゴリズムの収束性が理論的に証明されている。しかし、非単連結な障害物領域への拡張は解決されていない。なお、未知環境においては局所的なセンサ情報しか利用できないため、

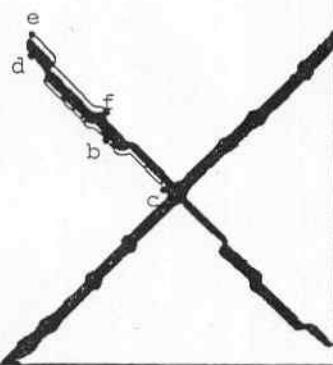


Fig. 15 The system automaton's motion corresponding to the robots' motion from Fig. 14(b) to Fig. 14(f).

t_3 にこの衝突を回避したとする。時刻 t_1 から時刻 t_3 までロボット R_2 が辿った実障害物の境界線分を曲線 PQ で表す。ロボット R_2 が曲線 PQ にある限り、時刻 t_3 から M 直線を目指に向けて移動するロボット R_1 の運動を阻止しない。ながら、このようなことがあつたとすると、アルゴリズム 3 は時刻 t_3 より早くロボット間の衝突を回避したはずである。従って、次の時刻 t_4 にバターン 2 の危険状態が検出されたとき、 $D_2(t_4) < D_2(t_1)$ 。

またアルゴリズム 1 で独立に計画されたロボット R_2 の経路では、線分 CD を辿るロボット R_1 の運動を阻止する部分が有限である。このことから、ある時刻 t_5 が存在してロボット R_1 がその時刻に点 D に到達する。明らかに $D_1(t_5) < D_1(t_1)$ 。時刻 t_5 からロボット R_2 が M 直線にのるまで、バターン 1 とバターン 2 の危険状態はしか生じないので、定理 3 と定理 4 及び前段の議論からある時刻 t_2 が存在して $D_1(t_2) < D_1(t_1)$ と $D_2(t_2) < D_2(t_1)$ 。□

定理 6 2つのロボットが時刻 t_1 に M 直線で危険状態となつたとする。ある時刻 t_2 が存在して、 $D_1(t_2) < D_1(t_1)$ と $D_2(t_2) < D_2(t_1)$ 。

証明：直線上で移動する2つのロボットの危険状態は高々一回だけ生じる。定理 3, 4, 5 から明らかに定理 6 が成り立つ。□

これらの定理から、定理 7 が簡単に導かれる。

定理 7 2つの移動ロボットが互いに衝突しなくかつ実障害物とも干渉しないで初期位置から目標位置に移動できれば、提案した協調運動計画アルゴリズムにしたがつて移動する2つのロボットに対して、有限の時間 T が存

計画された運動は最適とならない。これらの問題は今後の課題となる。

謝 詞

本論文の作成を手伝ってくださった東京大学工学部有本研究室の亀山博史、川口圭史氏、古賀弘樹氏及び他の皆様に感謝致します。

参 考 文 献

- 1) T. Lozano-Perez and M. A. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles." Commun. ACM, Vol.22, pp.560-570, 1979.
- 2) 劉 雲輝, 有本 卓, "一般的形状の障害物が存在する環境における移動ロボットの経路計画法", 日本ロボット学会誌, Vol.9, No.2, pp.56-63, 1991.
- 3) O. Khatib, "Real time obstacle avoidance for manipulators and mobile robots," the International Journal of Robotics Research, Vol.7, No.1, pp.90-98, 1986.
- 4) V. J. Lumelsky and A. A. Stepanov, "Dynamic path planning for mobile automaton with limited information on the environment", IEEE Trans. on Automatic Control, Vol.AC-31, 1986.
- 5) V. J. Lumelsky, "Continuous planning in unknown environment for a 3D cartesian robot arm", Proc. IEEE International Conference on Robotics and Automation, 1986.
- 6) H. Noborio, "A path-planning algorithm for generation of an intuitively reasonable path in an uncertain 2D workspace." Proc. the Japan-USA Symposium on Flexible Automation, pp.477-480, 1990.
- 7) K. Kant and S. Zucker, "Planning collision-free trajectories in time-varying environments: a two-level hierarchy." Proc. IEEE International Conference on Robotics and Automation, pp.1644-1649, 1988.
- 8) S. J. Buckley, "Fast motion planning for multiple robots." Proc. IEEE International Conference on Robots and Automation, pp.322-326, 1989.
- 9) Y. H. Liu, S. Koroda, T. Naniwa, H. Noborio, S. Arimoto, "A practical algorithm for planning a collision-free coordination of multiple mobile robots", Proc. IEEE International Conference on Robotics and Automation, pp.1427-1432, 1989.
- 10) 鈴木 寿, 有本 卓, "時変環境における点状の移動ロボットのペイント手続きを利用して再帰的起動計画法", 日本ロボット学会誌, Vol.5, No.5, pp.9-18, 1990.
- 11) Y. H. Liu and S. Arimoto, "Sensor-based coordination planning of two mobile robots in unknown environments." Proc. Japan-USA Symposium on Flexible Automation, pp.463-470, 1990.
- 12) P. A. O'donnell and T. Lozano-Perez, "Deadlock-free and collision-free coordination of two robot manipulators." Proc. IEEE International Conference on Robotics and Automation, pp.484-489, 1989.

劉 雲輝 (Yun-Hui LIU)



1985 年中国北京理工大学精密機械工学科卒業。1989 年大阪大学大学院基礎工学系研究科博士前期課程終了。同年東京大学大学院に入学。現在、同大学工学系研究科計数工学専攻博士後期課程に在学中。ロボットの運動計画、制御などの研究に従事。

IEEE の学生会員。(日本ロボット学会学生会員)

有本 卓 (Suguru ARIMOTO)



昭和 34 年京都大学理学部卒業。沖電気工業 KK, 東京大学工学部助手、講師を経て昭和 43 年大阪大学助教授、昭和 48 年同教授、基礎工学部に勤務。昭和 63 年より東京大学教授、工学部計数工学科に勤務。工学博士。ロボティクス、マシンインテリジェンスの研究に従事。日本機械学会、情報処理学会、計測自動制御学会等の会員。IEEE の Fellow 会員。

(日本ロボット学会正会員)