学術論文

平面を移動する複数の移動障害物とその速度を考慮した 移動ロボットのプランニングとナビゲーション

坪内 孝司^{*1} 浪花 智英^{*2} 有本 卓^{*2}

Planning and navigation by a mobile robot in the presence of multiple moving obstacles and their velocities

Takashi Tsubouchi^{*1}, Tomohide Naniwa^{*2} and Suguru Arimoto^{*2}

This paper presents a navigation scheme for a mobile robot which works even in an environment with multiple moving obstacles. An "iterated forecast and planning" approach is proposed by the authors. It is assumed that each obstacle moves at a constant velocity in the approach. The most feasible path for a robot is planned in (x,y,t) space. The planning and motion according to the plan are iterated frequently to cope with changes of motion of the moving obstacles. The behavior of proposed navigation algorithm is also presented by means of computer simulations.

Key Words: Mobile Robot, Navigation, Moving Obstacle Avoidance, Path Planning, Forecast of Obstacle Motion

1. はじめに

著者らは複数の移動ロボット、あるいは移動ロボットと人間などが同じ作業平面の上で移動している際の問題、特に、移動の開始点と終点（目標点）を移動ロボットに与えたとき、移動している障害物を避けながら移動の目標点に到達するための技術に興味がある。この種の問題についてはこれまでも多くの研究が行われている。複数の移動ロボット間に相互の通信チャンネルがあれば、衝突を回避する問題は複数ロボットの協調行動をいかに行うかに帰着される[1][2][3]。一方、もし相互の通信チャンネルがなければ、相手障害物の軌跡に関する知識やロボット自身に搭載された外界センサの情報を用い、衝突を避けうる経路の生成問題に帰着される。Kantら[4]は、移動障害物の各時刻における軌跡が既知の場合に、2次元の移動平面に時間を加えた時空空間において、移動障害物が占有する空間を避けるような経路計画を行った。Fujimura[5]は移動障害物の軌跡は既知であるが通過時刻が未知の場合に、移動障害物に衝突しない経路計画を取り扱った。またAraiら[6]は、仮想インピーダンスを用いて障害物を回避しながら移動ロボットを制御する手法を開発した。仮想インピーダンスを用いると、ポテンシャル場の局所停留による問題があるが、Araiらはこれを制御メカニズムを階層化して対処している。Noborioら[7]は、目標点に確実に到達できることを保証できる障害物回避アルゴリズムを提案した。また、武野ら[8]は、衝突してくる移動障害物

の危険度評価と回避行動の計画の方法についての提案を行っており、久保田ら[9]は移動障害物の回避行動の生成にファジイ規則を用いた。石川らは超音波測距器を多数リング上に配置した移動ロボットを用い、実際に多数の人が歩いている環境で非常に多数のファジイ規則を利用しながら実際のナビゲーションを行った[10]。Zhu[11]は、移動障害物の各時刻における加速度の発現が隠れマルコフモデルに従うとの仮定のもとに、加速度の観測値の時系列からその平均と分散を求めた。それらから、将来の時刻における移動障害物の位置を確率密度分布として予測した。さらに、ロボットの移動方向を8方向に量子化した上で、その確率密度分布を考慮した評価関数をもっとも高い値を示す方向に移動するような計画を局所的に行ってロボットの移動速度ベクトルを決定している。

著者らは、移動ロボットに搭載されているセンサによって複数の移動障害物の存在を知り得たとき、それらすべての影響を考慮に入れながらできるだけ移動の目標点に向かって安全にロボットを誘導する手法を開発したい（Fig.1）。その際、他の移動障害物の集団的な行動様式についての仮定はおかず、移動ロボット自身が独立して目標に向かう手法の開発が目標である。その意味でここでは[12]にあるような集団としての最適な行動形成を考慮に入れてはいない。本論文は、このような移動ロボットの自律誘導方式に関する一提案と、計算機によるシミュレーション結果の報告である。

2. 複数移動障害物存在下での自律誘導手法

Fig.1に示したような環境の中で人間が目標に向かって動くとするれば、視野のなかに入った相手（移動障害物）のその瞬間の位置と移動方向を観察して将来の動きを予測し、その予測

原稿受付 1993年9月10日

^{*1}筑波大学電子情報工学系

^{*2}東京大学工学部

^{*1}Univ. of Tsukuba, Inst. of Info. Sci. and Elec.

^{*2}Univ. of Tokyo, Faculty of Engineering

How does a robot navigate where many moving obstacles exist?

Fig. 1 Flow of obstacles and path planning.

と自分がめざすべき目標の方向とを考慮しながら近い将来にできるだけ衝突が起きないような経路を考え、それにしたがって移動しているとみることができる。もちろん予測に反して相手の移動方向が変われば、衝突が起きないような経路を再び考え直さなければならない。このように、「観察と障害物の将来の動きの予測・経路計画（プランニング）・計画（プラン）」に基づく移動の実行を繰り返しながら、目標点に向かって移動していると考えられる。この「将来の動きの予測」とそれに基づく「経路計画」は、実際に移動してみる前にどのような経路がよいかを頭の中で試行錯誤し、その結果、その時点でもっとも適すると思われる経路を解としてとみることができる。そこで、移動ロボットの場合も、

- 1°: それぞれの移動障害物の将来の動きを予測し、
- 2°: 将来において衝突が起きず、かつできるだけ目標地点に到達できるような経路をその予測に基づいて計画し、さらに
- 3°: 1°と2°を短い時間間隔で繰り返す、

という手法を用いることが効果的であると考えられる。1°においては、仮に移動障害物の将来の動きに関する知識（行動軌跡、運動拘束など）があれば、それを障害物の動きの予測に用いることもできる。また、予測に反して移動障害物が突然予期しない動きをしても、1°及び2°の頻繁な繰り返しによって対処できる。障害物の将来の動きの「予測」と「予測に基づく経路計画」は、あたかも近い将来の自らの行動の姿をシミュレーションによって求めているととらえることができる。以下本論文では、上に述べた「予測」と「予測に基づく経路計画」とを合わせて「予測シミュレーション」と呼ぶことにする。

このような自律誘導手法を構築していくための第1歩として、本論文では、移動障害物との間の通信手段を持たず、また移動障害物の軌道そのもの、あるいは移動障害物がとり得る運動拘束に関する知識もない場合を取り扱った。そこで1°では、「移動障害物は観測された速度を保つ」として将来の予測位置を求め、2°では1°の予測のもとで目標地点にできるだけ早く到達する経路を求めている。経路計画問題は移動平面2次元+時間1次元の3次元時空（以下では単に「時空」という）において取り扱い、[15]で提案されている発見的経路計画手法が応用できることを示す。

Fig. 2 A module structure for a mobile robot.

2.1 予測シミュレーションを含む自律誘導システムの構成

移動ロボット上に Fig. 2 に示すような構成を持つシステムを搭載することにより、予測シミュレーションを含む移動ロボットの自律誘導システムを構成しようと考えている。ロボットにひとたび移動の開始点と目標点が与えられると、地図 (map) を用いて path planner が大域的な経路計画を行う。このような大域的な経路計画には、例えば [14] のような手法を用いればよい。生成された経路が自律誘導のためのナビゲーションモジュール (navigation module) に与えられると、外界センサからの情報を用いながら実際の移動を開始する。ナビゲーションモジュールの内部に予測シミュレータ (forecast simulator) があり、ここで予測シミュレーションを短い時間間隔で繰り返し、そのたびごとの局所的な計画経路を走行モジュール (locomotion module) に送る。目標点に到達するまでこれが繰り返される。

なお、このようなナビゲーションモジュールの構築にあたっては、著者らの次のような考え方が背景にある。一般に、どのような状況のもとでもロボットの行動のための解が求まるアルゴリズムを考案することは困難である。むしろ、ひとつの想定した環境のもとである程度ロバストにはたらく戦略を開発し、それらを異なる想定環境ごとに用意して適宜使い分けるほうがよいであろう。本論文で2.2節以下で述べる予測シミュレータは、そのようなロバストな戦略の実現として位置づけている。さらに、ナビゲーションモジュールの内部にメインコントローラをおき、そのような戦略を実現するアルゴリズムを複数統括できるような枠組をおいた。本研究ではいまのところ予測シミュレータのほかに緊急監視 (emergency watchdog) をおいている。これについては2.4節で簡単に触れる。著者らの今後の研究の進展にしたいナビゲーションモジュールのなかにおくアルゴリズムを変更・追加して行く計画である。

さて、著者らは計算機上にナビゲーションモジュールを実現し、計算機上でシミュレーションを行うことによって上に述べた考え方によるナビゲーション手法の有効性の確認を試みた。実現にあたり、現在のところ以下のような仮定をおいている。

- (1) 移動ロボットは全方向移動が可能であり、自己の位置を正確に知ることができる。ロボットの移動可能な速さの最大値（最高速さ $|V_{max}|$ ）に制限がある。
- (2) 与えられた目標点にできるだけ速く到達しようとするが、

Fig. 3 A schematic view of the forecast simulation.

Fig. 4 An avoiding line.

- 必ずしも最速パスを要求しない。
- (3) ロボットは全周方向に測距可能な理想的距離センサを持ち、相手の移動障害物のその瞬間の位置を知ることができる。ただし測距範囲は有限とする。
 - (4) 移動障害物の移動軌跡に関する知識は持たない。そのかわりに距離センサからの情報から移動障害物の移動速度ベクトルが推定できる。
 - (5) ロボット及び移動障害物はすべて円筒形とみなすことができる。したがって、以下ではロボットを点とみなし、移動障害物はロボットの大きさ相当分に 1 より大きな係数をかけた値だけ半径を拡張した円筒とみなす。
 - (6) ロボットは他の移動障害物と通信を行う手段を持たない。また、移動障害物はセンサを持たず、自己の行動予定を移動途中で変えない。

なお上述の仮定 (3) 及び (4) についてであるが、例えば西澤ら [13] による全方位視覚センサの研究や情報処理速度の日進月歩の向上などを考慮すれば、近い将来における技術的な実現可能性は十分にあると考えている。

2.2 予測シミュレータ

予測シミュレータでは、外界センサからの情報を用いながら短い時間間隔で予測シミュレーションを繰り返し、次の予測シミュレーションを行うまでの間の局所的な経路が逐次ロボットの走行系に向けて出力される。移動障害物の移動軌跡に関する知識が事前に与えられていれば、予測シミュレーションはこの知識に基づいて行えばよい。しかし、ここではこの事前知識が与えられないという仮定をおいているので、予測シミュレーションを行う瞬間のセンサ情報から推定される各移動障害物の速度ベクトルにしたがって、各障害物が等速直線運動をすると仮定する。このような仮定のもとでは、移動障害物は時空中において斜円柱 C とみなされ (Fig. 3, 4), また、移動の目標点はその作業平面上の位置に立っている時間軸に平行な直線 L_G とみなされる。したがって、繰り返される予測シミュレーションの各回は、「移動の開始点 S と目標点 L_G を結び、すべての斜円柱 C と交差せず、しかもロボットの最高速 $|V_{\max}|$ を越えないで、できるだけ速く L_G に到達する時空中の妥当な経路 P を求める」というプランニング問題に帰着される。

この問題に対して大略次のような考え方によるアルゴリズムを用いた。すなわち、ロボットの移動開始点と目標点を直線で

Fig. 5 A reachable cone.

結び、それが障害物と交差しなければその直線を解とする。もし交差すればその障害物を避ける経路を作り、それが他の障害物と交差しないかを調べ、障害物と交差しない経路が見つかるまで再帰的に経路生成を試みる、という手法である。これは登尾らの提案したプランニング手法 [15] にヒントを得ているが、本論文では時空を分割してこれをオクトツリーで表現する手法は用いず、時空中の幾何学的な関係を利用して解析的な計算で済むようなアルゴリズムとした。さらに、ある障害物 X を回避して目標に到達する経路が別の障害物 Y に衝突するとわかったとき、 X, Y をこの順で避けるよりも先に Y を避ければ X には衝突せず、より早く目標点に到達できるというような場合がある。以下に述べるアルゴリズムは、このような場合も考慮されている。次にこのアルゴリズムの概要を説明する。

2.3 予測シミュレーションのアルゴリズム

まず、Fig. 4 のような回避直線を定義する。すなわち、時空中の一点 p を含み斜円柱に接する平面を考え、この平面と斜円柱の接線を半径方向に斜円柱の外側にある距離だけ平行移動した直線を回避直線と定義する。 i 番目の回避直線を A_i と書く。一方、時空中のある一点 p にロボットがいるとき、ロボットの到達可能範囲は p を頂点とする上に開いた円錐 N となる。これを「到達可能円錐」と呼ぶことにする。頂角はロボットの最高速 $|V_{\max}|$ で決まり、この円錐と回避直線とがもし共通部分を持つとすれば、それは、障害物の速さが移動ロボットの速さより遅ければ半直線となり、速ければ線分になる (Fig. 5)。この線分または半直線はロボットが回避直線上に到達することが可能な点の集合である。いま仮に、回避直線 A_i の並びが順

Fig. 6 Path planning avoiding oblique cylinder.

序づけされている集合 O_j が与えられたとする。移動の開始点における到達可能円錐とその順序づけされた 1 番目の回避直線との共通部分のうち、もっとも時刻の早い位置に次の到達可能円錐をおく。これと 2 番目の回避直線との共通部分を求め、その共通部分のうちもっとも早い時刻に到達可能円錐をおく、という操作を繰り返すことにより、最終的にその順序の最後にある回避直線上においた到達可能円錐と移動の目標点にある半直線 L_G との共通部分が求められる。 L_G との共通部分のうちもっとも早い時刻 T_j が、集合 O_j で指定されている順序で回避直線を通過するときの目標点への到達時刻である。このように、時空中の具体的な経路を求めなくても T_j を求めることができる [16]。

さて、 $\{O_j, T_j\}$ を組にして ω_j と表すことにする。また、 O_j で指定される順序にしたがって回避直線を経由する経路を P_j と記す。 P_j は時刻 T_j に移動の目標点に到達する。 O_j で指定された順に回避直線を経由する時空中の折れ線になる。さらに、 Ω を用いて、集合： $\{\omega_j\}$ ($j = 1, \dots, m$) (m は ω_i の数) を表すことにする。予測シミュレーションのアルゴリズムは次のようにまとめることができる (Fig. 6)。

Algorithm [予測シミュレーション]

- (1) いま $i = 1, j = 1, \Omega = \phi$ とする。移動の開始点 S と目標点 L_G をロボットの最高速さ $|V_{\max}|$ で結ぶ経路 $P_j (= P_1)$ を作る。このとき、 $\omega_1 = \{\phi, T_1\}$ (T_1 は最高速さ $|V_{\max}|$ で目標点にまっすぐ向かったときに要する時間) となり、 P_1 を作ったことにより $\Omega = \{\omega_1\}$ となる (Fig. 6(a))。
- (2) もし経路 P_j がまったく移動障害物を表す斜円柱に交差しなければ、この経路を解とする。
- (3) そうではなく、 P_j が斜円柱 C' と交差しているときには、以下の処理を行う。
 - (a) いま、回避直線の系列 O_j のなかで、経路 P_j が回避直線 A_i とその次の順にある回避直線 A_{i+1} の間で C' と交差しているものとする。各回避直線 A_k ($k = 1 \dots l$) 上にあり、かつ P_j 上にある各点 $p_{(k)}$ から、Fig. 4 のように斜円柱 C' の両側に回避直線 $A_{i(k)}$ 及び $A_{i+1(k)}$ を決める (Fig. 6(b)(c))。すなわち、 $P_{(k)}$ を含む平面の C' との接線をさらに半径方向に移動したものをそ

れぞれ $A_{i(k)}, A_{i+1(k)}$ とする。同様にして、移動の開始点 S から C' の両側に回避直線 $A_{i(0)}$ 及び $A_{i+1(0)}$ を決める。

- (b) いま、ひとつの要素からなる系列 $(A_{i(0)}), (A_{i+1(0)})$ をそれぞれ O_{j+1}, O_{j+2} と表す。また、すべての k ($k = 1 \dots l$) に対し、 $O_{j+(2k+1)}$ を系列 $(A_1, A_2, \dots, A_k, A_{i(k)})$ とおき、また $O_{j+(2k+2)}$ を系列 $(A_1, A_2, \dots, A_k, A_{i+1(k)})$ とおく。 ω_j を Ω から取り除く。
- (c) すべての k ($k = 0, \dots, l$) について、 $O_{j+(2k+1)}$ で指定される順に回避直線を経由する場合の、ロボットの最高速さ $|V_{\max}|$ の制限下での移動の目標点への到着時刻 $T_{j+(2k+1)}$ を評価する。同様に $O_{j+(2k+2)}$ で指定される回避直線を経由する場合の、目標点への到着時刻 $T_{j+(2k+2)}$ を評価する。こうして、 $\omega_{j+(2k+1)}$ と $\omega_{j+(2k+2)}$ をすべての k ($k = 0 \dots l$) について求め、すでに求められている Ω に付け加える。ただし、 $\omega_{j+(2k+1)}$ や $\omega_{j+(2k+2)}$ が物理的に実現不可能な経路である場合は除く。すなわち、このような実現不可能な経路は Ω から取り除かれる。
- (d) もし Ω が空集合になれば、このアルゴリズムを終了し、「解なし」とする。
- (e) そうでなければ、集合 Ω のすべての要素 ω_j を、 ω_j のなかの要素である T_j の昇順に並べ替え、そのなかでもっとも先頭にある $\omega_{j'}$ (集合 Ω のなかでの最速経路) を取り出す。 i, j を 1 ずつ増やし、 $A_{i(k)}$ と $A_{i+1(k)}$ の添字を適切につけなおす。
- (4) $\omega_{j'}$ に基づく経路 $P_{j'}$ が他の移動障害物 (斜円柱) と交差しないかどうかを調べる。
- (5) もし $P_{j'}$ がいかなる斜円柱とも交差しなければ、 $P_{j'}$ を解とし、交差する場合には $P_{j'}$ を P_j とみなしてステップ 3 にもどる。□

以上が本論文における予測シミュレーションのアルゴリズムの概要である。このようにして解が得られたら、短い時間間隔 dt の後に次の予測シミュレーションを行うまでの間、この解にし

たがう時空中の経路にそってロボット走行させる．また，上で述べたアルゴリズムにおいて解が求まらなかった場合には，解が求まった一番最近の予測シミュレーションの結果に基づく経路にそってロボットを走行させるようにしている．

2.4 緊急監視

予測シミュレーションによって時空中の経路 P_j' が求まれば，その経路にしたがった走行により，移動障害物の速度が変化しなければロボットは必ず目標に到達できる．しかし，実際には移動障害物の速度が大きな変化をする場合もあり，上で述べたアルゴリズムでは解が求まらない状況も起こり得る．

例えば， n 回目の予測シミュレーションと $n+1$ 回目の予測シミュレーションの間の時間 dt の間に，いままでロボットと近接して並走していた移動障害物 C が移動方向を急激に変化させ，ロボットに接近し過ぎるかもしれない．すると $n+1$ 回目の予測シミュレーションでは C から逃れることは不可能であるために解なしとなる． C から逃れることが不可能な条件はロボットと障害物の移動する速度に関係するが，やはり時空間における解析により求めることができる [16][17]．このような場合は，いったん移動の目標点に向かうことをあきらめ，ロボットに可能な最高速さで C から逃れるような行動を生成することにした．そこでナビゲーションモジュールに緊急監視をおき，いま述べたような状況が生じた場合に対処できるようにしている．ただし，いくら最高速さで C から逃れるような行動をとったとしても，ロボットを標的にロボットよりも高速で障害物が移動してくれば，これをロボットが回避することは不可能である．また，ロボットのまわりを複数の移動障害物が意図的に取り囲み，移動目標の方向とは異なる方向に移動しても 2.3 節で述べたアルゴリズムでは解は得られない．しかし，このような場合について対処することは現在のところ行っていない．

3. 実験結果

計算機シミュレーションにより，上で述べた移動障害物が複数存在する環境での移動ロボットのナビゲーション手法の振舞いを観察した．アルゴリズムはすべて Lisp 系の言語である i-scheme によって，サンマイクロシステムズ社製の Sparc Station 2 上で実現した．ここで示したシミュレーションにおいては移動環境を壁や柱などの固定障害物が存在しない二次元自由空間とし，あらかじめ与えられた地図による静的大域的な経路計画問題は省いている．しかし，本論文で提案した手法は，壁などの固定した障害物を排除することが本質的なわけではなく，障害物が円筒であるという仮定をはずすような拡張を行えばよい．

Fig. 7 にこの実験で与えた環境を示す．図において二次元平面のメッシュ間隔は 100 [cm] である．移動障害物の軌跡は 2.5 秒おきの位置をあらかじめ与え，これを実際にはスプラインで補間して滑らかな動きになるようにしている．移動障害物の速さはおよそ 10~20 [cm/s] 程度，またほぼ 2~3 秒の周期で振動するような動きが与えられている．また移動ロボットの移動の開始点 (Start) と目標点 (Goal) を図のように与えた．ロボットの外界センサの測距範囲は 300 [cm] とした．したがってこの範囲に入る障害物が予測シミュレーションにおいて考慮

Fig. 7 Experimental environment.

Fig. 8 Trajectories at $t = 19.4$ [s] and $|V_{\max}| = 25$ [cm/s].

Fig. 9 Trajectories at $t = 30.2$ [s] and $|V_{\max}| = 25$ [cm/s].

の対象となる．実験では，Fig. 7 に与えた環境を用いて，ロボットの最高速さ $|V_{\max}|$ を変化させた場合の振舞い (Fig. 8~13) と予測シミュレーションの繰り返し時間間隔を変化させた場合の振舞いを観察した (Fig. 14~16) ．

Fig. 8~10 は，移動ロボットの最高速さ $|V_{\max}|$ を 25 [cm/s] としたときのものであり，Fig. 11, 12 は $|V_{\max}|$ を 50 [cm/s]，Fig. 13 は $|V_{\max}|$ を 75 [cm/s] としたときのロボットの軌跡を障害物の軌跡と共に表示したものである．ただし予測シミュレーションの繰り返し時間間隔 dt を 0.2 秒と固定した．これらの例でみられるように，移動ロボットはそれぞれ移動障害物の集団と集団の間を通り抜けて目標点に到達している．通り抜ける集団と集団のすき間が各図によって異なっており，移動ロボットの最高速さ $|V_{\max}|$ が大きいほど移動障害物を追い越してから移動の目標に近づいていくような振舞いをしていること

Fig. 10 Trajectories at $t = 55.6$ [s] and $|V_{\max}| = 25$ [cm/s]. Fig. 11 Trajectories at $t = 9.8$ [s] and $|V_{\max}| = 50$ [cm/s]. Fig. 12 Trajectories at $t = 19.0$ [s] and $|V_{\max}| = 50$ [cm/s]. Fig. 13 Trajectories at $t = 12.6$ [s] and $|V_{\max}| = 75$ [cm/s].

は興味深い。これは、予測シミュレーションの各時点で、常に外界センサレンジの範囲内に入る障害物を回避しつつ移動の目標点に到達するまでの経路計画を常に行っているための帰結であると思われる。

Fig. 14, 15は、ロボットの最高速さ $|V_{\max}|$ を 25 [cm/s] に

Fig. 14 Trajectories at $t = 20.6$ [s], $|V_{\max}| = 25$ [cm/s] and $dt = 4$ [s]. Fig. 15 Trajectories at $t = 56.0$ [s], $|V_{\max}| = 25$ [cm/s] and $dt = 4.0$ [s].

固定し、予測シミュレーションの時間間隔 dt を 4.0 秒と長くしたときの振舞いである。また、Fig. 16 には、ロボットの最高速さ $|V_{\max}|$ を 25 [cm/s] に固定し、予測シミュレーションの時間間隔 dt を 0.2, 1.2, 2.0, 4.0 秒としたときのロボットの軌跡のみをトレースしたものである。図をみると明らかにように、移動障害物の動きの変化の周期とほぼ等しい時間以上に dt をとると、ロボットの軌跡は滑らかさを失い、大きく蛇行するようになる。いいかえれば、想定される移動障害物の運動の変化の周期よりも短く dt を選んでおけばよいことがわかる。なお、Fig. 16 中で特に記した区間 A では、ロボットの直前に障害物があり、これがロボットの前で左右にあたかも意地悪をするかのように振る舞ったため、常にロボットが障害物のない方向に動こうとして軌跡が蛇行している (Fig. 8)。

ところで、本論文で用いた予測シミュレーションのアルゴリズムでは、移動障害物の速度ベクトルまで考慮した。しかし、毎回の予測シミュレーションにおいて、障害物が観測された位置に留まっているとして（障害物の速度がゼロであるとして）経路計画を行っても同様な結果が出るかもしれないと考えられる。そこで、障害物の速度がゼロであるとして、上で述べた予測シミュレーションのアルゴリズムを援用して比較実験を行ってみた。移動障害物の軌跡はこれまでの実験と同様にし、ロボットの移動開始点と目標点を障害物の移動方向を横切るように配置した。Fig. 17, 18 に移動障害物の速度ベクトルを考慮した実験結果を示し、Fig. 19~21 に移動障害物の速度ベクトル

Fig. 16 Coarse trace of robot trajectories.

 Fig. 17 Trajectories at $t = 15.8[s]$ and $|V_{\max}| = 25[cm/s]$. (Navigation by the the original **Algorithm** and the emergency watchdog.)

が常にゼロであり、観測された位置に留まっていると仮定した実験結果を示す。図から明らかなように、移動障害物の速度ベクトルを考慮しないと移動ロボットが障害物にひっばらつてしまうような現象が起こる。これは障害物の前を通る経路が適する解として求まり、次の予測シミュレーションでまた障害物の前を通る経路が求まる、ということが連続して起こるためと考えられる。このことから、移動障害物の速度ベクトルも考慮するとよいことが示された。

最後に、本論文で述べたナビゲーションモジュールの構成の限界を示す例を Fig. 22～24 に掲げる。移動の目標点は図の左方にある。この例では移動ロボットの最高速さ $|V_{\max}|$ を 25 [cm/s] とし、移動障害物の速さを 100 [cm/s] 程度とした。図からわかるように速い移動障害物に移動ロボットが囲まれてし

 Fig. 18 Trajectories at $t = 26.8[s]$ and $|V_{\max}| = 25[cm/s]$. (Navigation by the the original **Algorithm** and the emergency watchdog.)

 Fig. 19 Trajectories at $t = 11.4[s]$ and $|V_{\max}| = 25[cm/s]$. (Velocity of each obstacle is ignored in the forecast simulation.)

 Fig. 20 Trajectories at $t = 17.2[s]$ and $|V_{\max}| = 25[cm/s]$. (Velocity of each obstacle is ignored in the forecast simulation.)

 Fig. 21 Trajectories at $t = 31.6[s]$ and $|V_{\max}| = 25[cm/s]$. (Velocity of each obstacle is ignored in the forecast simulation.)

もう例である。Fig.22において移動ロボットは障害物に囲まれ
てしまい、ゴールと反対の方向に少しずつ流されてしまってい
ることがわかる。さらに、Fig.23では、上から3番目の障害物
に接近しすぎたために緊急監視が働いて、この障害物から全
力で逃げる振舞いをしている。しかし、他の障害物から逃れ
ることができず、上から4番目の障害物に結局衝突している
(Fig.24)。

以上の計算機シミュレーションでは、実現の記述を Lisp 系
のインタープリタベースの言語で行っているために、ここで示
した例ではロボットの移動の開始から終了までの計算に最高約
25分を要している。1回の予測シミュレーションにおいて、経
路と移動障害物との交差判定は最高 30回 程度行われ、そのと
きに展開された経路の候補数は最高 40 程度であった。また1
回の予測シミュレーションに、最高 50秒 程度を要しており、1
回の交差判定に 3秒程度、1回の経路の展開に 0.5秒 程度を要
している計算になる。ここで示したアルゴリズムの実時間性の
評価は、アルゴリズムをコンパイラベースの言語で記述した上
で行いたいと考えている。

Fig. 22 Trajectories at $t = 3.5$ [s] and $|\mathbf{V}_{\max}| = 25$ [cm/s]. (A robot is surrounded by faster obstacles.)

Fig. 23 Trajectories at $t = 5.4$ [s] and $|\mathbf{V}_{\max}| = 25$ [cm/s]. (A robot is surrounded by faster obstacles. A watchdog was invoked.)

Fig. 24 A robot finally collided with an obstacle.

4. お わ り に

複数の移動障害物が存在する環境下での移動ロボットのナビ
ゲーション手法について述べた。本手法では、外界センサから
移動障害物の位置と速度ベクトルが得られると仮定し、ある瞬
間では移動障害物が外界センサ情報に基づく等速直線運動を行
うとして、障害物の将来の動きを予測しながら経路計画を行っ
ている。さらにこの経路計画を短い時間間隔で繰り返すこと
により、移動障害物の軌跡に関する事前の知識がなくてもよい
ような手法になっている。また、本稿で述べた移動ロボットの
ナビゲーション手法の有効性を計算機シミュレーションによっ
て示した。また移動障害物の速度ベクトルを考慮しない場合の比
較、及び提案した手法ではうまくいかない例も示した。

本論文で提案した手法は、移動障害物の速さが移動ロボット
と同程度か遅い場合にはかなりロバストに働くことが計算機シ
ミュレーション上で確認されている。このようにロバストに働
くアルゴリズムを、想定される典型的な状況ごとに対応して組
み立てておき、必要に応じて使い分けることで、状況の変化に
おいて破綻のきたしにくい移動ロボットのためのナビゲーション
システムが構築できるのではないかと考えている。本稿で述
べた手法はその一部をなすものである。

今後の課題としては、移動ロボットの運動に関するより現実
的な制約の導入、障害物の先見的知識に基づいた運動の予測、
ロボットがとった行動の学習による経路計画の負荷の軽減など
があげられる。

謝 辞

本研究におけるアルゴリズムの計算機上への実現実験には、
平岡和幸氏（東京大学大学院工学系研究科）の労によるところ
が大きい。ここに感謝の意を表す。

参 考 文 献

- [1] F. R. Noreils: "Coordinated execution of trajectories by multiple mobile robots," Proc. of IEEE/RSJ International Workshop on Intelligent Robots and Systems '91 (IROS'91), pp. 1061-1067, Nov., 1991.
- [2] 浅間一, 尾崎功一, 板倉広明, 松元明弘, 石田慶樹, 遠藤勲: "通信およびルールに基づく複数の移動ロボットの衝突回避", 日本ロボット学会 第1回ロボットシンポジウム予稿集, pp. 73-78, 1991.
- [3] S. Yuta and S. Premvuti: "Consideration on cooperation of multiple autonomous mobile robots - introduction to modest cooperation -," Proc. of 5th Int. Conf. on Advanced Robotics, 1991.
- [4] K. Kant and S. Zucker: "Planning collision-free trajectories in time-varying environments: a two level approach," Proc. of Int. Conf. on Robotics and Automation 1988, pp. 1644-1649, 1988.
- [5] K. Fujimura: "Route planning for mobile robots amidst moving obstacles," Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems '92 (IROS'92), pp. 433-438, July, 1992.
- [6] T. Arai and J. Ota: "Motion planning of multiple mobile robots," Proc. of IEEE/RSJ International Workshop on Intelligent Robots and Systems '92 (IROS'92), pp. 1761-1768, Sept., 1992.
- [7] H. Noborio and J. Hashime: "A Feasible collision-free and deadlock-free path-planning algorithm in a certain workspace where multiple robots move flexibly," Proc. of IEEE/RSJ International Workshop on Intelligent Robots and Systems '91 (IROS'91), pp. 1074-1079, Nov., 1991.

- [8] 武野純一, 柿倉正義: “移動ロボットの移動障害物に対する衝突回避問題について,” 日本ロボット学会誌, vol. 4, no. 5, pp. 33-37, 1986.
- [9] 久保田 孝, 橋本秀紀, 原島文雄: “移動障害物に対する衝突回避行動形成の一手法”, 日本ロボット学会 第 8 回学術講演会予稿集, pp. 181-182, 1990.
- [10] 石川繁樹, 浅香俊一: “移動障害物を含むような動的変化を伴う走行環境における自律誘導ロボットの走行方式,” 日本ロボット学会誌, vol. 11, no. 6, pp. 856-867, 1993.
- [11] Q. Zhu: “Hidden Markov model for dynamic obstacle avoidance of mobile robot navigation,” IEEE Trans. on Robotics and Automation, vol. 7, no. 3, pp. 390-397, June 1991.
- [12] 倉林大輔, 太田 順, 新井民夫: “動的な群形成による移動ロボット群の動作計画,” 日本ロボット学会第 11 回学術講演会予稿集, pp. 411-414, 1993.
- [13] 西澤慶満, 八木康史, 谷内田正彦: “全方位視覚センサ COIPS を搭載した移動ロボットのための環境マップの生成と移動自由空間の推定,” 日本ロボット学会誌, vol. 11, no. 6, pp. 868-874, 1993.
- [14] Y. H. Liu and S. Arimoto: “A feasible algorithm for planning local shortest path of mobile robots based on reachability graph,” Proc. of IEEE/RSJ Int. Workshop on Intelligent Robots and Systems (IROS'90), pp. 749-756, July, 1990.
- [15] 登尾啓史, 浪花智英, 有本 卓: “クワッドツリーを利用した移動ロボットの高速経路生成アルゴリズム,” 日本ロボット学会誌, vol. 7, no. 5, pp. 403-412, 1989.
- [16] T. Tsubouchi, K. Hiraoka, T. Naniwa and S. Arimoto: “A mobile robot navigation scheme for an environment with multiple moving obstacles,” Proc. of IEEE / RSJ Int. Conference on Intelligent Robots and Systems (IROS'92), pp. 1792-1798, July, 1992.
- [17] 平岡和幸: “複数移動障害物環境下における移動ロボットのナビゲーション手法”, 東京大学工学部計数工学科卒業論文, 1992 年 3 月.
- [18] 平岡和幸, 坪内孝司, 浪花智英, 有本 卓: “複数移動障害物環境下における移動ロボットのナビゲーション—第 1 報 移動障害物がない場合のナビゲーション—”, 第 6 回 知能移動ロボットシンポジウム予稿集, pp. 69-74, 1992.

坪内孝司 (Takashi Tsubouchi)

1959 年 8 月 7 日生. 1983 年筑波大学第 1 学群自然科学類卒業, 1988 年筑波大学大学院博士課程工学研究科修了. 日本学術振興会特別研究員, 宇都宮大学工学部情報工学科助手, 東京大学工学部計数工学科助手を経て, 1994 年 4 月より筑波大学電子情報工学系講師となり現在に至る. 工学博士. ロボット工学, 特に移動ロボット全般に興味を持つ. 人工知能学会, IEEE の会員. (日本ロボット学会正会員)

浪花智英 (Tomohide Naniwa)

1965 年 11 月 6 日生. 1990 年大阪大学大学院基礎工学研究科前期課程修了. 同年東京大学大学院工学系研究科博士課程入学. 同年中退と同時に東京大学工学部計数工学科助手となり現在に至る. ロボットマニピュレータに対する学習・適応制御の研究, 移動ロボットの経路計画の研究などに従事. IEEE の会員. (日本ロボット学会正会員)

有本 卓 (Suguru Arimoto)

1936 年 8 月 3 日生. 1959 年京都大学理学部卒業. 沖電気工業 K.K., 東京大学工学部助手, 講師を経て 1968 年大阪大学基礎工学部助教授, 1973 年同教授, 1988 年東京大学工学部教授となり現在に至る. 工学博士. ロボティクス, 制御理論, 信号処理, 情報理論などの研究に従事. 日本機械学会, 情報処理学会, 計測自動制御学会などの会員. IEEE の fellow 会員. (日本ロボット学会正会員)