

# 混雑通路中のロボットの スムーズな通行のアルゴリズムの検討

## Investigation of the Algorithm for the Robot to Path Smoothly through the Crowded ways

野口 真也† 春木 竜一‡ 山内 俊明† 関 靖夫†

Shinya Noguchi Ryuichi Haruki Tosiaki Yamanouchi Yasuo Seki

### 1. まえがき

数年前の二足歩行ロボットの出現に端を発したロボット・ブームはいや増すばかりである。今後はさらに普及が進み、家庭やオフィスのみならず街中にロボットが見られる日もそう遠くないと思われる。そうすると、従来のようにロボットが単独もしくは数台であった場合には無かった問題が発生することが考えられる。その一つが、衝突回避の問題である。相手が建造物などの静止した物体であれば衝突回避はさほど難しくはないが、人間を含めた生物や自律的に移動するロボットが相手の場合にはそう単純ではない。

そこで本稿では、混雑した通路中をロボットがスムーズに通行できるようなアルゴリズムの開発を目的に、その第一歩としてより単純なケースを対象に計算機シミュレーションによる検討を行った。具体的には、直線双方向通路を想定し、各個体が同じ衝突回避アルゴリズムによって移動方向を変更するというものである。特に今回は単純化のために、各個体は通路内の全ての個体の各瞬間での位置と速度の情報を持っていると仮定した。さらに、衝突回避は至上命令であるが、それによって発生する目的地到達に要する時間の増大を最小限に抑制するため、当該個体の進行方向変動を最小限にするアルゴリズムを検討することにした。取り扱い問題及び衝突回避アルゴリズムの詳細、そして計算機シミュレーションの結果について述べる。

### 2. 問題の定式化

まず、ロボットや人などの自律的に移動する個体を次のように定義した。

#### (1) 各個体が有するパラメータ

各個体には、現在の位置と速度ベクトル(速さと方向)という2つのパラメータを持たせることにした。なお、全対象空間領域に出現する際(以後、初期)の速度ベクトルは、乱数を用いてランダムに設定する。これにより、通行速度の大きさと方向は個体によって異なるが、各個体のそれらは衝突回避が行われる時以外は不変である。なお、衝突回避時の速度ベクトルの変更も、方向のみに限定するものとする。

#### (2) 個体を得ることのできる情報

各個体は通路内の全ての個体の各瞬間での位置と速度の情報を持っている

#### (3) 個体の動作

各個体は自己が有する速度ベクトルに従い、一定間隔の瞬間のみに進行の可能性があり、その他は一定とする。なお、詳細は次の空間の定義の項にゆずるが、個体が通行する空間は画面の上下方向に通じた直線路であるため、初期の速度ベクトルにも画面の上下方向の成分が含まれるようにした。つまり、各個体の最終到達目標は通路の両端の出口いずれかであり、衝突回避のために速度ベクトルの方向が変化しても、その垂直方向成分の符号は終始変化しないとする。

次に個体が通行する空間の定義であるが、以下のように行った。

#### (1) 通路としての基本機能

上下に出入り口が存在し、左右は通り抜けることのできない壁で塞がれているものとする。ここで便宜上、画面の上の出入り口をゲート1、下の出入り口をゲート2とする(以後、他の章でも両出入り口をこれらの名称で呼ぶ)。ゲート1から当該通路に入った個体は、最終的にはゲート2から出なければならない。逆もまた然りである。また、通路上に存在する各個体は、いずれかのゲートへ向かって進行しなければならず、停止している個体は存在しないものとする。

#### (2) 個体の通行様式

各ゲートから当該通路に入ってくる個体の増減は、一定時間当たりの人数(パラメータとして与えられる)として調整することができる。なお、ゲートから同一瞬間に複数の個体が入ってくることはないものとする。各個体が、ゲートから侵入してくる際の水平方向の座標は、乱数でランダムに決定される。

### 3. 衝突回避アルゴリズム

#### 3.1 他の個体との衝突回避アルゴリズム

まず、どの個体と衝突しそうかを判定する必要があるが、当面効率を考えて次の3つの条件を満たす個体の中で当該個体から最も近い個体についてのみ判定を行う。

(1) 速度ベクトルの垂直方向成分の符号が、当該個体と異なる。

(2) 当該個体の速度ベクトルの方向からある視野半角内(閾値 $\phi_h$ )の方向に存在する。

(3) 当該個体からの距離が閾値 $r_h$ 以下である。

次に、衝突予測判定の方法であるが、図1に示すような当該個体(自己)を原点とする相対座標を考える。自己の速度ベクトルを $\mathbf{V}_i$ 、予測判定対象個体(以後、相手)の速度ベクトルを $\mathbf{V}_j$ 、現在の相対位置ベクトルを $\mathbf{P}$ とする。このまま両者の速度ベクトルが不変であれば、時刻 $\Delta t$ 後に相手の相対位置ベクトルは $\mathbf{P} + (\mathbf{V}_j - \mathbf{V}_i)\Delta t$ となる。従って、自己と相手の最短距離 $d_{\min}$ は、 $\mathbf{P}$ から $(\mathbf{V}_j - \mathbf{V}_i)$ 方向に

† 神奈川工科大学, Kanagawa Institute of Technology

‡ ㈱レオ, LEO Corporation

伸ばした直線に原点からおろした垂線の長さとなる。  $d_{min}$  が予め決められた距離 (閾値  $d_h$ ) 以内であれば、衝突の危険性があり、2つの場合に分けて対応する。

(A)  $d_{min}$  が閾値  $d_h$  より大の場合には、回避処理は行わない。

(B)  $d_{min}$  が閾値  $d_h$  以下である場合には、図2に示すように最短距離に達する以前に両者の距離が閾値  $d_h$  と等しくなってしまう。この場合は、両者の距離が閾値  $d_h$  と等しくなるまでの所要時間を求め、閾値  $t_h$  と比較して下回れば回避運動の必要性ありと判定する。

最後に回避の方法であるが、基本的に速度ベクトルの大きさは変えずに、方向のみを修正する。具体的には、修正後の両者の予測最短距離  $d_{min}$  が閾値  $d_h$  を超える条件を満たす速度ベクトルの方向を探索する。探索は、現在の速度ベクトルの方向を中心に、ある刻み幅の角度の定数倍で左右に振ることで、必要最小限の修正角度を見つけるというものである。

### 3.2 壁との衝突回避アルゴリズム

回避運動の方法自体は、個体との衝突回避の場合と同じであるので、ここでは衝突予測判定の方法についてのみ述べる。自己の速度ベクトルの水平方向成分の符号が正の時は右の壁、負の場合は左の壁との衝突までの所要時間を算出し、閾値  $t_h$  と比較して下回れば回避運動の必要性ありと判定する。

## 4. 計算機シミュレーション

表1に示すような閾値・パラメータを用いて、計算機シミュレーションを行ったところ、個体が近接して衝突の危険が予測されると、必要最小限の回避運動が行われることが確認できた。

しかし、通路の混雑度を示すパラメータ「個体の時間的間隔」を極端に小さく (混雑度を増す) すると、衝突の危険が予測されて回避運動を行おうとするのであるが、それでも衝突してしまう事例が発生してしまった。これは、衝突予測判定を行う対象を3つの条件をクリアした中で最も近い個体のみ限定しているためと思われる。

## 5. むすび

混雑した通路中をロボットがスムーズに通行できるようにアルゴリズムの開発を目的として、より単純なケースにおける衝突回避アルゴリズムを検討した。

具体的には、直線双方向通路内を同じ衝突回避アルゴリズムを持つ各個体が通行するというものである。今回は、単純で効率のよいアルゴリズムでどこまで衝突回避を順当に行うことができるかどうか重点を置いたため、ある個体が将来衝突するかどうかを判定する対象を1つの個体に限定した。当然、そのような個体は自己以外のすべての個体の中から、条件をクリアし、かつ最も距離の近いものを選ばれるのであるが、それでも通路がかなり混雑してくると衝突を回避しきれなくなることが予想される。実際に行った計算機シミュレーションにおいても、このことが確認された。より高度な衝突予測判定と回避運動を検討し、混雑の激しい通路にも対応させることが今後の課題である。

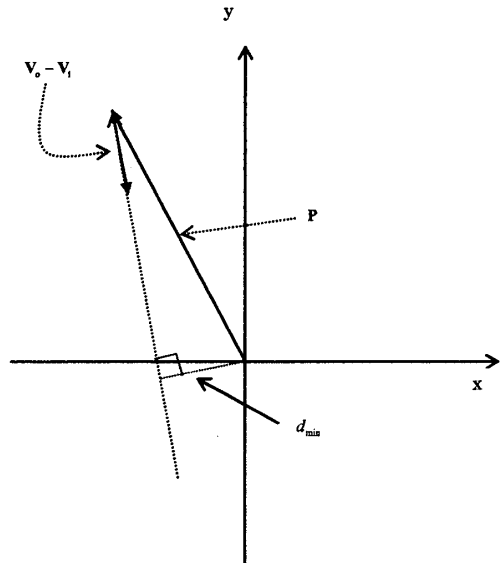


図1 当該個体を原点にした相対座標の例

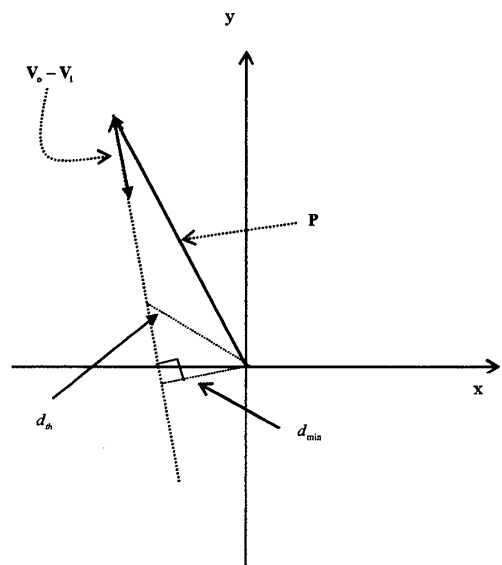


図2 両者の最短距離が閾値を下回る場合の位置関係の例

表1 使用した閾値・パラメータの一覧

閾値・パラメータの種類	値
通路の幅	400[ピクセル]
通路の長さ	650[ピクセル]
個体の半径	15[ピクセル]
個体の速度下限	4[ピクセル]
個体の時間的間隔	30[単位時間]
衝突判定対象個体選択時に用いられる視野半角の閾値	60[度]
衝突判定対象個体選択時に用いられる距離の閾値	100[ピクセル]
衝突回避の修正角度探索時に用いられる角度の刻み幅	5[度]