

PF法を用いた車輪型移動ロボットの行動決定

Decision Making for a Mobile Robot Using Potential Function Method

○及川一美*, 大久保重範*

○Kazumi Oikawa*, Shigenori Okubo*

*山形大学工学部

*Faculty of Engineering, Yamagata University,

キーワード: ポテンシャル関数法 (Potential Function Method), 重ね合わせ (Superposition),
行動規範型ロボット (Behavior-based Robotics), 屋内ナビゲーション (Indoor Navigation),
サブサンプション・アーキテクチャ (Subsumption Architecture)

連絡先: 〒992-8510 米沢市城南四丁目三番地十六号 山形大学 工学部 機械システム工学科
及川一美, Tel.: (0238)26-3246, Fax.: (0238)26-3246, E-mail: okazu@yz.yamagata-u.ac.jp

1. はじめに

Subsumption Architecture¹⁾ (以下, SAと呼ぶ) に代表される行動規範型手法において, 互いに独立した複数のモジュールから出力される行動を全体の行動としてどのように採択するかという問題に対して, ポテンシャル関数 (以下, PFと呼ぶ) を用いて重ね合わせる手法について述べる.

我々はSAが計画を必要とするタスクに不向きであるという問題を解決するためにオブジェクト指向設計法を提案し²⁾, ナビゲーションタスクに適用した³⁾. オブジェクト指向設計法ではSAを干渉型と非干渉型の二つに分けて考え, プログラミングが容易な非干渉型を採用した場合に起こる上記の問題に対して, 状態に応じて階層構造を切替えることで解決を図った. しかし, 局所的には階層構造の問題は残されたままで, 非効率な行動が依然として現れた.

ところで, 複数のルールを統合する手法として,

津崎ら⁴⁾のポテンシャルメンバーシップ関数 (以下, PMFと呼ぶ) を利用する手法がある. RoboCupのサッカーロボットに適用し動的且つ複雑な環境においてロバストな結果を示した. この手法は複数のルールをPMFで表現し, それらをファジィ演算により統合した結果から行動を決定する.

本論文ではPMF法のような重ね合わせの手法を採用することで, SAの階層構造に起因する問題の解決を図る. ここではPMF法を改良したPF法について述べる.

2. 問題設定

本研究では, 移動ロボットの目的地到達問題を扱う. この問題を解くためにロボットが保持するマップの内部表現としてグラフ表現³⁾を採用する. 本稿ではローカルランドマークが存在する通路状環境を, グラフマップで表現する手法を用いる. こ

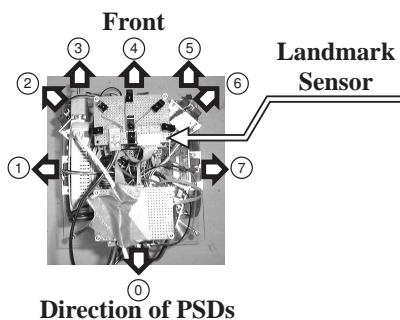


Fig. 1 Mobile robot



Fig. 2 Corridor environment

ここで、ローカルランドマークとはランドマークの近傍でのみ観測可能なランドマークと定義し、以下、特に断わらない限りランドマークと表記する。

2.1 移動ロボット

本論文では実験で使用する実機を想定してシミュレーションを行った。ここで想定するロボットは本研究室で製作した移動ロボットである。このロボットは独立駆動車輪型移動ロボットで、測距センサとしてPSDを8個、電子コンパスを1個、そしてランドマーク検出センサを搭載している。ロボットはデッドレコニングは用いず正確な自己位置はわからないものとする。上部からの概観と測距センサの方向をFig. 1に示す。

2.2 作業環境

作業環境は文献^{3, 2)}と同様に通路状環境とし、Fig. 2に示されるように交差点やT字路、端点にランドマークが配置される。ランドマークは環境中の任意の特徴物が望ましいが、ロボットに搭載

するセンサやマイコンの性能、及び技術的な問題でロボットが検出しやすい特別な人工物とする。ここでは、赤外線を用いてID信号を発信するランドマークを想定しており、ロボットはその赤外線が届く範囲でランドマークのIDを識別することが可能であるとする。

3. SAの問題点

非干渉型SAはFig. 3に示すような階層型構造による行動決定手法であり、行動を出力する各モジュールは他のモジュールからの干渉はない。そのため、モジュールから出力された行動が互いに相反するときでも、上層のモジュールが優先して採用されるので、複雑な処理を必要とせず実現が容易である。しかし、この単純な調整法では、モジュールのレベルで見たととき最適な行動であっても、全体的には不適切な行動となる場合が多い。例えば、Fig. 4に示されるような状況で、障害物回避を担当するモジュールの選択できる行動は、AもしくはB方向に回避することである。一方、通路沿い走行を担当するモジュールはA方向を選択する。しかし、障害物回避モジュールは他のモジュールの意図は全く知らないため、AかBか独自の判断で決めることになる。運悪くB方向を選択したとしよう。その場合、通路沿い走行モジュールと衝突することになるが、障害物回避モジュールが上層であればB方向が優先され採用される。以上のように、上層を優先する階層構造は不適切な行動を生成しやすいという問題を抱えていた。それは下層の意図を無視し上層の独断で決めてしまうところに問題がある。

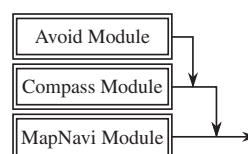


Fig. 3 Layered Control

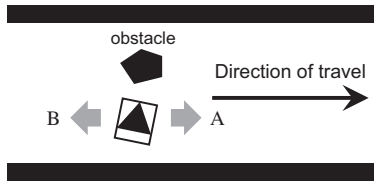


Fig. 4 Example of problem of SA

以下、この階層構造の問題を解決し、かつできるだけ単純な調整方法を有するような行動決定方式を考察することにする。

4. ポテンシャル関数法

PMF法では、ファジィ理論におけるメンバーシップ関数を用いて横軸が行動の種類、縦軸が優先度を意味する関数を用いる。ところで、ある瞬間において、能動的な行動と抑制的な行動が同時に存在することは珍しくない。それらと同時に、受動的、つまり能動的でも抑制的でもない行動も存在する。この3種類の行動を一つのメンバーシップ関数で表現する事は困難である。何故なら、関数と演算がペアになって3種類の行動の内、2種類の行動を表現するからである。例えば、AND演算の場合、1の値を与えられた行動は受動的を意味し、0の値を与えられた行動は抑制的を意味する。一方、OR演算の場合、1の値を与えられた行動は能動的を意味し、0の値を与えられた行動は受動的を意味する。このように、関数だけでは一意に決まらず、演算を伴って表現しなければならない上に、同時に3種類の行動は表現できないのである。

したがって以上の理由により、メンバーシップ関数およびファジィ演算は本手法では採用せず、3種類の行動を一つの関数で表し、上記の要求を満たした演算を考える。

4.1 ポテンシャル関数及び演算

PFとは行動に対する優先度を与えるものであり、優先度は $(-\infty, \infty)$ で与えられる。これより、PFは

行動の集合を $X = \{x_1, x_2, \dots, x_n\}$ とすると、次のように定義される。

$$F: X \mapsto (-\infty, \infty); \quad F(x_i) \in (-\infty, \infty), x_i \in X$$

優先度は行動が優先的に選択される度合いのことであり、 $F(x_i) > 0$ は行動 x_i が能動的行動であることを意味する。同様に $F(x_i) < 0$ は抑制的行動、 $F(x_i) = 0$ は受動的行動を意味する。PF同士の演算は、全てのモジュールの意思を反映させるために多数決を採用し、基本的には代数和を用いる。但し、抑制的行動はロボット自身を護るための重要な意思決定であり優先されるべきである。したがって、異符号同士の演算は負の値を演算結果とする。本演算の演算子を \oplus と表すと、演算は以下のようなになる。

$$x \oplus y = y \oplus x = \begin{cases} x + y & (x \cdot y \geq 0) \\ x & (x < 0 \leq y) \end{cases} \quad (1)$$

ところで、優先度の範囲が $(-\infty, \infty)$ であると、モジュール間に不平等が生じるので、モジュールが出力するPFを、ある適当な定数 M を用いて次のように定義する。

$$f: X \mapsto [-M, M]; \quad f(x_i) \in [-M, M], x_i \in X$$

以下、特に説明がなければ優先度の範囲でPFを大文字と小文字で区別して表記する。以上よりPFの演算は新たなPFを生成し、次のように定義する。

$$F(x_i) = f_a(x_i) \oplus f_b(x_i), \quad x_i \in X$$

ただし、右辺のPFは f でも F でも定義可能であるが、生成される関数は F のみである。

4.2 統合及び行動決定

m 個のモジュールそれぞれから出力されるPFを f_1, f_2, \dots, f_m とすると、統合演算は次のように定義される。

$$F^*(x_i) = f_1(x_i) \oplus f_2(x_i) \oplus \dots \oplus f_m(x_i), \quad x_i \in X$$

この統合演算を x_1 から x_n まで行った結果,

$$F^*(x_{\max}) = \max(F^*(x_1), F^*(x_2), \dots, F^*(x_n))$$

となる行動 x_{\max} をロボットの行動として採用する。但し、複数の行動が得られた場合は適当な規則で1つに決定する。

4.3 移動ロボットへの適用

独立二輪駆動型移動ロボットが行動するために必要なパラメータは前進速度 v 及び回転速度 ω である。この2つのパラメータを独立に決定するために文献⁵⁾を参考にして開空間の方向 θ と距離 ϕ の情報を用いることにする。方向に関してはPFを用いて求め、距離に関しては変換関数(以下、TF)を用意することで求める。PFの定義域はロボットの正面からの相対角度 $\Theta = \{\theta_1, \theta_2, \dots, \theta_n\}$ とする。TFの定義域はPFと同様に Θ であるが、値域は各方向に対するロボットから障害物までの距離とする。したがって、TFを次のように定義する。

$$G: \Theta \mapsto [0, \infty); \quad G(\theta_i) \in [0, \infty), \theta_i \in \Theta$$

モジュールから出力されるTFも値域は変わらないが、便宜上PFと同様に大文字と小文字を区別して表記する。TFの演算を次のように定義する。

$$G(\theta_i) = \min(g_a(\theta_i), g_b(\theta_i)), \quad \theta_i \in \Theta$$

よって、TFの統合演算は次のように定義される。但し、 m 個のモジュールから出力されるTFを g_1, g_2, \dots, g_m とし、以下の演算を θ_1 から θ_n まで行う。

$$G^*(\theta_i) = \min(g_1(\theta_i), g_2(\theta_i), \dots, g_m(\theta_i)), \quad \theta_i \in \Theta$$

ここで、PFの統合演算によって得られた角度が θ_{\max} であるとき、距離 ϕ を次のように求める。

$$\phi = \begin{cases} G^*(\theta_{\max}) & (F^*(\theta_{\max}) \geq 0) \\ 0 & (F^*(\theta_{\max}) < 0) \end{cases}$$

以上の操作から前進速度 v 及び回転速度 ω は次のよ

うに求める。

$$v = K_v \phi$$

$$\omega = K_\omega \theta_{\max}$$

ここで、 K_v 、 K_ω は比例ゲイン定数である。

ところで、 θ_{\max} が一意に決まらない場合、今回はロボットの正面に最も近い角度を選んだ。それでも決まらない場合は乱数で求めた。

4.4 ナビゲーションタスクにおけるPF

Fig. 6にシミュレーションの一場面と、統合後のPF及びTFを示す。黒の三角印が付いた四角がロボットとその姿勢を意味しており、三角印の向きがロボットの正面である。黒の塊は壁もしくは障害物を意味し、それ以外は通路を意味する。通路状にある数字はランドマークのID及び位置を表している。その周囲のグレーの領域がランドマークを観測できる範囲を表す。Fig. 5に各モジュールのPF及びTFを示す。以下、各モジュールについて、紙面の関係上PFについては概要のみ述べる。

Avoid Module

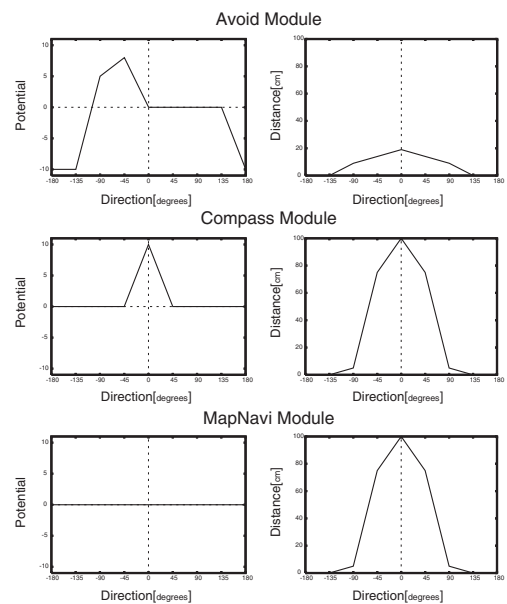


Fig. 5 A example of PF and TF of each module

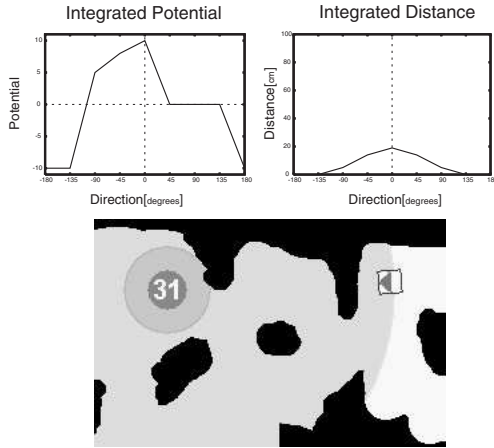


Fig. 6 Integrated PF and TF

距離センサからの情報により，障害物の方向及び開空間を求め，障害物回避が可能なPFを生成するモジュールである．状態遷移図をFig. 7に示す．

PFの生成では，距離値 x [cm]からポテンシャル p に変換するのに次式を用いている．

$$p = \begin{cases} -10 & (x < 5) \\ \frac{(x-20) \times 10}{15} & (5 \leq x < 20) \\ 0 & (20 \leq x < 25) \\ \frac{(x-25) \times 10}{15} & (40 < x) \end{cases} \quad (2)$$

TFは旋回中に障害物との衝突を避けるために，正面方向のTFが最大になるように，距離センサからの値を加工している．加工する式は以下の通りである．

$$\tilde{d}_\theta = \begin{cases} d_\theta W(\theta) & (d_\theta > d_{\theta_f}) \\ d_{\theta_f} W(\theta) & (d_\theta \leq d_{\theta_f}) \end{cases} \quad (3)$$

但し， θ は各方向の角度， θ_f は正面方向の角度， d_θ は θ 方向の距離値， \tilde{d}_θ は θ 方向の加工した距離値， d_{θ_f} は正面の距離値である．関数 $W(\theta)$ はTable 1に示すような重み関数である．

この処理はRotate State以外で行われ，Rotate

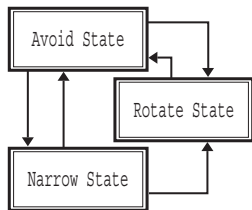


Fig. 7 State Transition Diagram of Avoid Module

Table 1 Weighting function

$\theta[^\circ]$	-180	-135	-90	-45
$W(\theta)$	0.0	0.25	0.50	0.75
$\theta[^\circ]$	0	45	90	135
$W(\theta)$	1.0	0.75	0.50	0.25

Stateでは超信地旋回を行うため，TFは全ての方向を0にする．

Compass Module

通路を沿う行動を出力するには距離センサの数が極端に少ないために，地磁気を利用して目標の方位に誘導することで，沿い走行を実現するためのモジュールである．このモジュールでは状態遷移は行わない．PFは単純に目標の方位に最大値を与えている．TFは正面を最大にし常に固定である．

MapNavi Module

指示された経路にしたがって目的地に誘導するためのモジュールである．状態遷移図をFig. 8示す．図中の矢印の根元に書かれた単語は遷移時に発生させるイベント名である．PFはランドマークの観測可能領域外では全方向を0，領域内ではラン

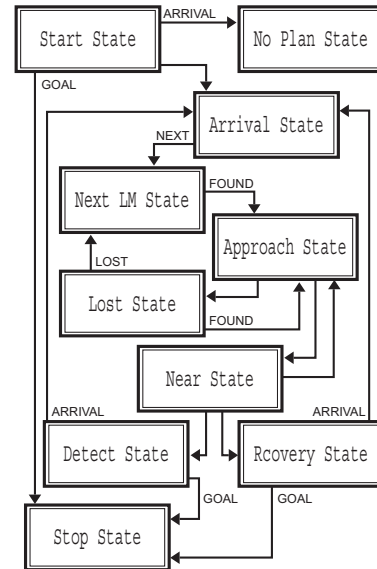


Fig. 8 State Transition Diagram of MapNavi Module

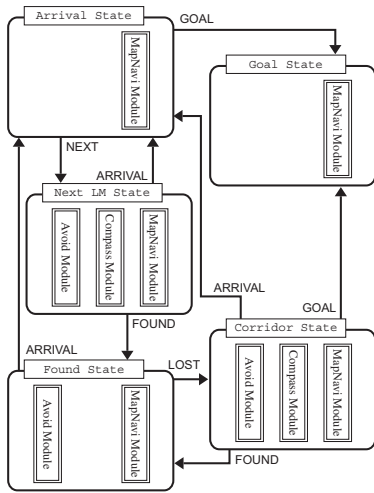


Fig. 9 Event Driven State Transition Diagram with PF Method

ドマークがある方向を最大値にするように生成する。TFはCompass Moduleと同じ生成方法である。例では、ランドマークの観測可能領域内に入っていないため、PFは全ての方向に対して0を出力している。

4.5 イベント駆動状態切替手法

タスクを円滑に遂行するためには、状況に合わせて適宜必要なモジュールを選択する必要がある。この仕組みにはオブジェクト指向設計SA²⁾で用いたイベント駆動による状態遷移を用いる手法を採用する。同様にモジュール内のPF及びTF生成アルゴリズムも状態遷移により切替える。今回のナビゲーションタスクにはFig. 9のように設計した。図中の矢印の根元の単語は遷移のきっかけとなるイベント名である。

5. シミュレーション

5.1 SAとPF法の比較

PF法の有効性を確認するためにSAとの比較実験を行った。実験は計算機上で行う。本実験では2つの手法の効率を比較する。その指標としてスタートからゴールまでのステップ数を用いる。ス

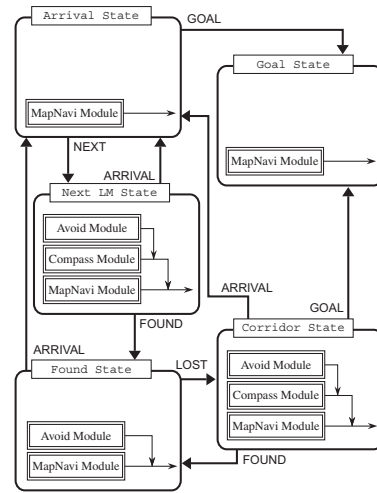


Fig. 10 Event Driven State Transition Diagram with SA

テップとはサンプリング周期のことを意味し、今回は50msecとした。ただし、ステップ数はロボットの移動速度の違いによる影響を受けるので、一概にステップ数だけで効率が良いと判断できないことに注意が必要であるが、適切な速度を選択しているかという判断も含めて、ここではステップ数による比較を採用した。したがって、ここではステップ数が少ないほど効率が良いと判断することにする。

5.2 SAの設定

本実験で用いるSAは文献²⁾の手法によって設計する。比較の精度を上げるために、モジュールはPF法で設計したものを階層構造に適用できるように改良して用いる。今回、設計した状態遷移図をFig. 10に示す。

5.3 Task 1

1つ目のタスクに対する作業環境をFig. 11に示す。図にはランドマークとそのID、及びスタート位置と向きが示されている。経路はTable 2に示すような経路を与えた。

PF法とSAをそれぞれ20回試行した時のステッ

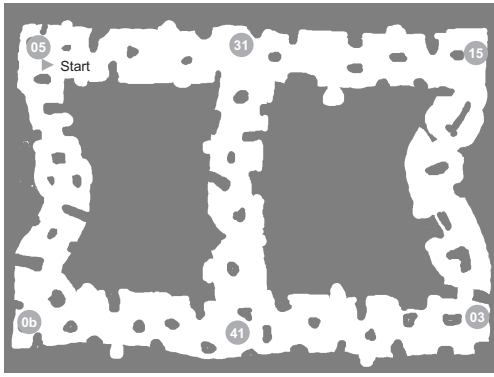


Fig. 11 Corridor Environment of Task 1

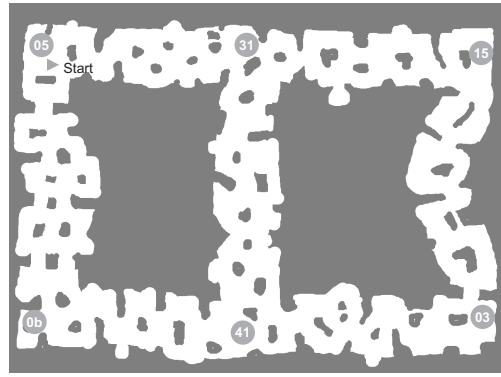


Fig. 12 Corridor Environment of Task 2

Table 2 Path of Task 1

order	1	2	3	4
ID	0b	03	15	05

ブ数の平均と標準偏差，最大及び最小ステップ数を**Table 3**に示す。

Table 3 Result of Task 1

	Ave.	SD	min	max
PF	2035.8	133.0	1814	2409
SA	2072.7	446.3	1539	3378

Task 1においては比較的容易な環境であるので，PF法とSAに有意な差は出なかった．標準偏差に少し差が見られるのは，Avoid Moduleが速度を抑える作用をしているのであるが，SAの場合はAvoid Moduleが起動しないときはその抑制がかからず速度が出やすい，つまり，障害物が少ない場所では速度が出やすいので，巧く障害物を避けていけば非常に早い結果が得られる．それに対して，障害物を巧くよけれないときは階層構造の問題により，無駄な動きを繰り返すのである．そのため，SAは標準偏差が若干大きくなっている．このことから言えるのは，PF法においてもより適切な速度設定が可能であるということであり，速度について再検討することにより，より少ないステップ数で目的地に到達できることを示唆している。

5.4 Task 2

Task 1の環境に障害物を多く配置した環境 (**Fig. 12**)で行う．手書き地図はTask 1と同じものを使用した．与えた経路も同じである．

Task 1と同様に20回の試行を行ったときの，ステップ数の平均値及び標準偏差，最大及び最小ステップ数を**Table 4**に示す。

Table 4 Result of Task 2

	Ave.	SD	min	max
PF	2482.1	164.7	2207	2838
SA	3660.8	959.4	2102	5400

Task 2においてはウェルチのt-検定により有意な差が出た．すなわち，PF法の方が少ないステップ数で目的地に到達できることが示された．**Fig. 13**，**Fig. 14**に最大ステップ数の結果を示す．これらは4ステップ毎のロボットの位置と姿勢を示している．図中の数字は経路を意味しており，分かりや

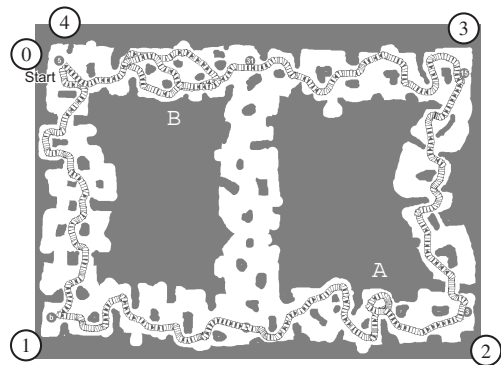


Fig. 13 Horrible result of task 2 with PF method

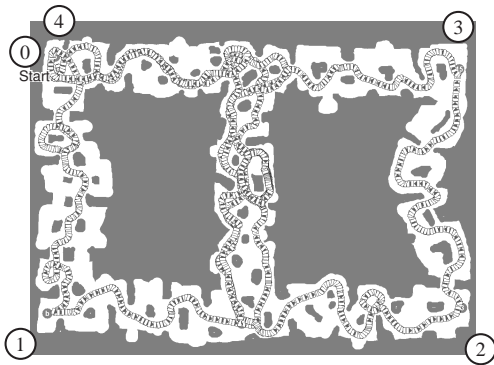


Fig. 14 Horrible result of task 2 with SA

すくするために加えた。ただし、0はスタート地点を意味する。Fig. 13はPF法の結果である。図中A、Bの付近で無駄な軌跡を生成しているのが分かる。これは、一つの理由として距離センサの角度分解能の低さ（周囲に対して8方向しか測定していない）があげられ、角度分解能の高いセンサを使用することで回避することが可能であると考える。Fig. 14はSAの結果である。ID41のランドマークで適切な通路に入れず本来ならばID03に向うはずが、ID31に向っているのが分かる。ちなみにPF法ではこのようなことは一度も起きなかった。しかし、本手法は部分観測情報のみで行動決定する手法であるから、PF法においても障害物の配置によっては同じようなことが起こり得る。また、目的地へ到達可能な解が存在し、且つ目的地に到達できないような障害物の配置も可能である。それは情報量の少ないグラフマップと部分観測のみの手法の大きな課題である。

6. おわりに

SAの階層構造が原因で起きる無駄な行動を軽減することについて、PMF法の考えをベースにしてPF法を提案し、その問題の解決を図った。通路状態環境における目的地到達問題に適用し、同様に階層構造の問題の解決を図ったイベント駆動による階層構造切替手法と比較して、シミュレーション

により有効性を示した。

今後は実験を行い実世界での実現可能性を検討する。また、シミュレーション結果より明らかになった、速度決定に関する問題点の解決を図っていく。

参考文献

- 1) Rodney A. Brooks: "A Robust Layered Control System For A Mobile Robot," *IEEE Journal of Robotics and Automation*, vol. RA-2, no. 1, pp. 14–23, 1986.
- 2) 及川一美, 土谷武士, 大久保重範: "サブサンプリング・アーキテクチャのオブジェクト指向設計", *日本ロボット学会誌*, vol. 23, no. 6, pp. 697–705, 2005.
- 3) Kazumi Oikawa, Hidenori Takauji, Takenori Emaru, Shigenori Okubo and Takeshi Tsuchiya: "Navigation Using Local Landmarks in a Corridor Environment," *Journal of Robotics and Mechatronics*, vol. 17, no. 3, pp. 262–268, 2005.
- 4) 津崎亮一, 吉田和夫: "ファジィポテンシャル法に基づく全方位視覚を用いた自律移動ロボットの行動制御手法", *日本ロボット学会誌*, vol. 21, no. 6, pp. 656–662, 2003.
- 5) Takanori Emaru, Kazuo Tanaka and Takeshi Tsuchiya: "Speed control of a sonar-based mobile robot with considering the self-localization," In *IEEE International Conference on Mechatronics & Automation*, pp. 125–130, 2005.