

## 自律移動ロボットの実時間行動獲得に適した環境表現方法

### Environment Representation Method Suitable for Real-time Action Acquisition of Autonomous Mobile Robot

道木 加絵<sup>†</sup>, 鳥井 昭宏<sup>†</sup>, 植田 明照<sup>†</sup>  
Kae DOKI, Akihiro TORII, Akiteru UEDA

**Abstract** We have focused on and proposed an action acquisition method of an autonomous mobile robot. This method is based on the real-time search, and it adopts the anytime sensing, which treats the trade-off between the search time and the quality of the search result. To treat this trade-off, the environmental information utilised in the action search changes over the search time in this method. Considering the search process, the environmental information used in the action search influences the search result. Therefore, it is important to consider how to represent the environmental information. According to this reason, we propose an environment representation method suitable for an autonomous mobile robot to acquire an action, and we verify its usefulness.

#### 1. はじめに

近年、自律移動ロボットの知的行動獲得に関する研究が盛んに行われ、様々な方法論が提案されてきた。従来手法の多くは、ロボットの行動をルールとして記述するルールベース手法<sup>1)</sup>、ロボットが取った行動に対する環境からの報酬に基づき環境に適した行動を学習する強化学習手法<sup>2)</sup>、そして両者を組み合わせた手法<sup>3)4)</sup>に大別することができる。これらの手法に加え、近年の計算機能力の著しい向上により、実時間探索による自律移動ロボットの行動獲得手法が提案されるようになった<sup>5)6)</sup>。我々も実時間探索に着目し、これに基づいた行動獲得手法を提案してきた<sup>7)</sup>。

実時間探索による自律移動ロボットの行動獲得手法では、現在の状況に適した行動を実時間で探索することにより獲得する。ここでの実時間とは、ロボットを取り巻く状況から決められるある限られた時間を意味する。このように、実時間探索では探索時間が限られ、さらにロボットを取り巻く状況によって変化するため、探索時間が変化しても効率的な探索手法が必要である。一般に、探索はセンシングで獲得した環境に関する情報を用いて行われる。これは、センシングで獲得した環境情報が探索結果に影響を与えることを意味する。本論文でのセンシングとは、機械的なセンシングよりも、むしろ獲得したセンサ値から探索で用いる環境情報を構築する信号処理を意味する。これより、効率的な探索を行うためにはそれに適した環境情報の構築方法が必要となる。これまで、特定のタスクに適した行動をある一定時間内に探索

するのに適した環境情報表現方法と行動探索手法は数々提案されてきた。しかしながら、本研究で想定する実時間探索では探索時間が状況に応じて変化するため、探索時間と探索で用いる環境情報の質とのトレードオフを考慮しなければならない。例えば、探索時間が短い場合、探索に必要な計算量を少なくすることが重要であり、これには質を落した環境情報を用いることで解の質を落しながらも計算量を減らし、短時間で解を獲得することが重要である。反対に、探索時間が十分与えられる場合は、十分な計算量を割くことができるため、高い質の解を獲得するためのより詳細な環境情報が行動探索に必要とされる。従来手法では探索時間が一定であるため、このようなトレードオフは考慮されておらず、以上のことから本研究の枠組みに適用することは困難であると考え。そこで、我々は探索時間と環境情報の質とのトレードオフを扱う Anytime Sensing<sup>8)</sup>に着目し、探索時間に応じて変化した環境情報の質と探索で得られる解の質とのトレードオフを扱った自律移動ロボットの実時間行動探索手法を提案してきた<sup>9)</sup>。この方法では、探索時間の経過と共に探索で用いる環境情報（以後、仮想環境と呼ぶ）の質を上げながら探索を行うことで、仮想環境の質（以後、仮想環境情報量と呼ぶ）と探索で得られる解の質とのトレードオフを扱った。これにより、様々な状況での探索効率が改善されることを確認した。

仮想環境情報量を低くすることは、探索に必要な計算量を減らすのと同時に、環境に関する何らかの情報を失った仮想環境を構築することを意味する。先に述べたように、仮想環境の質は探索結果に影響を与えるため、仮

<sup>†</sup> 愛知工業大学 工学部 電気工学科 (豊田市)

想環境の構築方法, すなわち仮想環境の表現方法が問題となる. 従来手法では探索で用いる環境情報の質と探索で得られる解の質とのトレードオフを扱う方法論に焦点を当ててきたため, 仮想環境の構築方法, すなわち探索に用いる環境情報の表現方法については検討を行ってこなかった. そこで本研究では, 環境情報の表現方法と探索効率の関係について言及し, 自律移動ロボットの行動獲得に適した環境情報の表現方法の提案を行う.

## 2. Anytime sensing

本論文では, 探索時間と仮想環境情報量のトレードオフを考慮するため Anytime Sensing を適用する. この手法は, 探索後に得られる解の質の期待値が最大となるようセンシングの時間割当を行う手法である. この手法では, センシングは物理的なセンサ処理からセンサより得た情報を探索に用いる環境情報 (仮想環境) への変換までの一連の処理を意味する. センサの処理時間や環境情報の変換方法に費やす時間を変化させることで, 結果として仮想環境の質を変化させている.

自律移動ロボットの行動獲得を考えると, 予期せぬ状況に直面した場合は途中で探索を打ち切り何らかの対応を行う必要が考えられる. このような場合, 事前に決められた探索時間が確保されないこととなる. この時, 場合, 探索に多くの計算を必要とする仮想環境を用いて探索を行っている時, 打ち切り時に十分な質の解が得られていない (最悪の場合では解が得られていない) ことが考えられる. このことから, Anytime Sensing をそのまま適用することは困難である. そこで本研究では, 探索時間と仮想環境情報量のトレードオフをはかるため, 探索時間の経過と共に仮想環境情報量を増加させながら探索を行う. すなわち, 質の低い仮想環境を用いて探索を開始し, その後, 探索時間の経過と共に徐々に質の高い仮想環境を用いて探索を行う. これにより, 探索途中に状況変化による探索打ち切りが生じた場合にも, 探索時間に応じた質の解を得ることが可能となる. また, 探索開始時に少ない計算量を用いて探索を開始することで, 短時間で初期解が得られ, 結果としてより実時間性の高いシステムが実現できる. さらに, 時間と共に仮想環境情報量を上げながら探索を進めることで, 解空間上で荒い探索から詳細な探索を行うこととなり, 結果として探索効率の向上が期待できる.

## 3. 環境情報の表現方法

本論文では, 具体例として複数移動障害物回避問題を扱い, 自律移動ロボットの行動獲得に適した環境情報の表現方法について検討する.

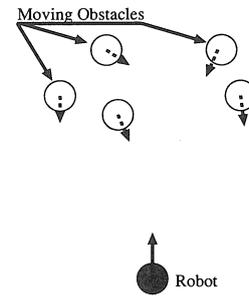


図 1 : ロボットを取り巻く環境

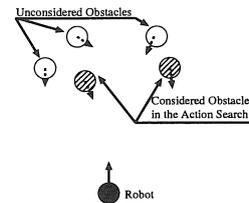


図 2 : 構築した仮想環境 (環境情報の制限)

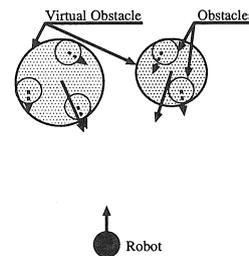


図 3 : 構築した仮想環境 (分解能の変化)

図 1 に示す環境でロボットが状況に適した行動を探索する場合を例に環境情報の表現方法について述べる. 機械的なセンシングにより, ロボットから見た全ての移動障害物の状況を表すパラメータが獲得できるものとし, その得られたセンサデータから探索に用いる環境情報を構築することを考える. その際, 環境情報の質を変化させる方法として次の 2 つが考えられる. まず第 1 に, 図 2 に示すように, 探索時に利用する環境情報を制限する方法である. この図では, 行動探索時に考慮する移動障害物の数を図のように制限することで, 環境情報の質を下げ, 結果として行動探索に必要な計算量を減らすという方法である. 第 2 の方法として, 図 3 のように環境情報の分解能を変化させる方法が考えられる. この図では, 複数の移動障害物を 1 つの障害物とみなすことで全ての障害物を考慮に入れながら, 見かけ上の移動障害物数を減らすというものである.

上記の 2 つの環境情報の表現方法には, それぞれ以下のような長所および短所がある. まず, 探索に利用する環境情報を制限する手法では, 仮想環境を構築するため

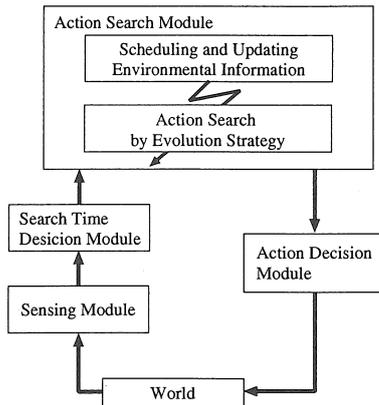


図4：複数移動障害物回避動作計画システム

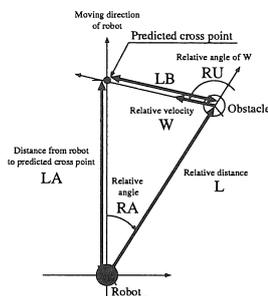


図5：センサから得られる移動障害物の状況

の計算は必要としないが、探索効率を劣化させないように情報の取捨選択を行わなければならない。一方、環境情報の分解能を変化させる手法は、仮想環境の構築に計算を必要とするが、獲得した全てのセンサデータを何らかの形で利用することとなる。以上を考慮に入れ、探索効率改善の観点から考えると、次に挙げる理由から環境情報の表現方法として分解能を変化させる方がより望ましい。図2のように環境情報を制限すると、行動探索に重要な情報を完全に落としてしまい、かえって探索性能を劣化させる可能性がある。環境情報の取捨選択を行うメタ推論システム等を導入する方法も考えられるが、一般に推論システムは多くの計算量を必要とする。これは限られた計算量の効率的な利用を目指した本研究の目的に反する。これに対し、環境情報の分解能を変化させる手法では獲得した環境に関する全ての情報が何らかの形で考慮されるため、このような事態に陥りにくいと考えられる。

#### 4. 移動障害物回避動作計画システム

提案手法を用いて構築した複数移動障害物回避動作計画システムについて述べる。システムを構築する際、以下のような前提条件を与えた。

1. ロボットの動作平面上には移動障害物のみが存在する。

2. ロボットは速度および操舵を同時に制御する。
3. ロボットはセンサ範囲内にある障害物の運動が検知可能である。
4. 移動障害物は等速直線運動を行う。
5. センサ誤差は考慮しない

##### 4.1 システム構成

図4に本論文で構築した複数移動障害物回避動作計画システムを示す。本システムは主に4つのモジュールから構成されている。動作開始後、ロボットは予め決められた時間、与えられた速度でゴール方向へまっすぐ向かい、それ以降の行動は以下の流れにより決定する。なお、ロボットが1つの行動を出力した後、次の行動を決定するまでに行う一連の処理の単位をステップとして表現する。

1. センサにより、センサ範囲内にある移動障害物の状況およびゴール状態を認識する。(Sensing Module)  
移動障害物の状況は、ロボットと移動障害物間の相対角度 ( $RA$ )、相対距離 ( $L$ )、相対速度 ( $W$ )、相対速度ベクトル方位角 ( $RU$ ) により表される (図5)。また、ゴール状態はゴール距離 ( $RTL$ ) および方位角 ( $RTA$ ) により表される。
2. センサから得られた情報および現ステップでの探索時間に基づき、次ステップの探索時間を決定する。(Search Time Decision Module)
3. 現ステップの探索時間の間、ロボットが次に取るべき行動を仮想環境を基に探索する。仮想環境は、探索時間の経過と共に更新される。(Action Search Module)
4. 探索により得られた行動をロボットへの速度および操舵指令値へと変換し、行動を出力する。(Action Decision Module)

##### 4.2 環境情報の表現方法とその更新

続いて、本システムにおける環境情報の表現方法について述べる。自律移動ロボットの行動獲得に適した環境情報の表現方法についての検討を行うため、環境情報を制限する方法および分解能を変化させる方法の2通りを実装した。以下、順に述べる。

###### 4.2.1 分解能の変化

図6のような状況で障害物回避行動を探索するものと仮定し、この状況で分解能を変化させる仮想環境の構築方法について述べる。本論文では次のような方法で仮想環境を構築する。まず初めに図6 (a) に示すように全ての障害物を囲む領域を求める。この領域を障害物とみなし (以後、仮想障害物と呼ぶ)、この障害物を回避する

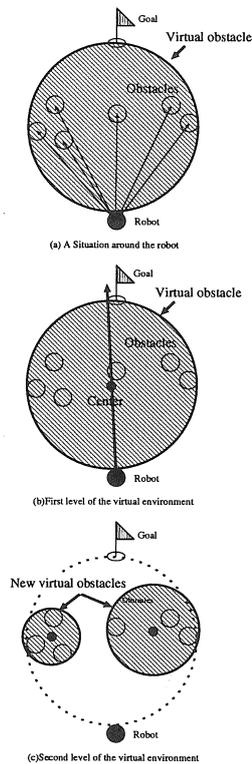


図 6 : 分解能を変化させる場合の仮想環境の表現方法

ような行動の探索を開始する。その後、ある程度探索が進んだら、次に述べる方法でこの領域を分割し、仮想障害物の数を増やすことで仮想環境の分解能を変化させる。まず、ロボットと仮想障害物の中心を結ぶベクトルにより仮想障害物を二分割し、各領域に含まれる移動障害物の状況から新たな仮想障害物を計算する。このように新たに求めた仮想環境を用いて行動探索を続行する。仮想環境の更新と行動探索は時間が終了するまで繰り返し行う。

#### 4.2.2 環境情報の制限

提案手法との比較を行うため、環境情報を制限する場合の環境表現方法を以下のように構築した。本システムでは、図 3 に示したように移動障害物回避行動を探索する際に考慮する移動障害物の数を探索時間の経過と共に増加させることで、探索時間の経過と共に仮想環境の質を変化させる。行動探索時に考慮される移動障害物の順番は、ロボットから移動障害物の相対距離が近い順とする。これは、ロボットから近い距離の移動障害物ほどロボットが取る行動に影響を与える可能性が高い、と考えるからである。

### 4.3 探索手法

#### 4.3.1 探索アルゴリズム

問題設定で述べたように、ロボットは速度および操舵

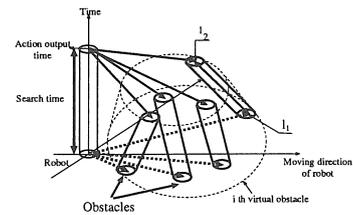


図 7 : 障害物回避に対する評価値

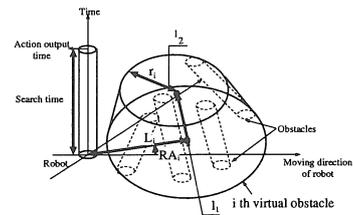


図 8 : 障害物回避に対する評価値 (仮想障害物を使った場合)

操作量により制御される。そこで、本論文では各操作量の組合せを行動とし、進化的探索手法である Evolution Strategy (ES)<sup>10)</sup> を用いて適切な組合せを探索する。今回は各操作量を直接遺伝子としてコーディングし、遺伝操作として突然変異のみを用いる。

#### 4.3.2 評価関数

移動障害物回避問題では、ロボットはゴールに向かうという目的と同時に移動障害物を回避するという目的も持つ。そこで、本論文では、(1) 式で表されるように、その行動を取った時のゴールに対する評価値  $G$  と障害物回避に対する評価値  $V$  を個別に求め、重み  $\alpha$  により合成したものを行動に対する評価値とする。この評価値が小さいほど状況に適した行動とみなす。重み  $\alpha$  はロボットから最も近い障害物の距離に比例する。これにより、障害物がロボットから遠い場合はゴール到達重視の評価を、移動障害物が近い場合は障害物回避重視の評価を行う。

$$Evaluation = \alpha \cdot G + (1 - \alpha) \cdot V \quad (1)$$

$\alpha$  重み ( $0 \leq \alpha \leq 0.5$ )  
 $G$  ゴールに対する評価値  
 $V$  障害物回避に対する評価値

(2) 式にゴールに対する評価式を示す。ゴールに対する評価は、取った行動によりロボットがゴールに近づく程、また進行方向がゴール方向と一致するほど値が小さくなる。

$$G = \frac{1}{1 + \exp\left(\frac{\delta RTL - RTL_{th}}{c_1}\right)} \cdot \frac{1}{1 + \exp\left(\frac{RTA_{th} - RTA}{c_2}\right)} \quad (2)$$

$t_2$  次ステップの探索時間  
 $\Delta RTL$  行動前後のゴール距離の変化分  
 $RTL_{th}, c_1, RTA_{th}, c_2$  設計パラメータ

(3) 式に障害物回避に対する評価式を示す。

障害物回避に対する行動評価を考える場合、ロボットに対する障害物の相対距離が小さく、またロボットの進行方向に対する障害物の相対角度が小さくなるほど障害物はロボットに対し危険であると考えられる。しかし、ロボットおよび障害物は探索中も動作しているため、図7のように障害物の状況は時間と共に変化する。そこで、(4)式で定義されたロボットに対する*i*番目の障害物の危険度  $W_i$  を経路積分し、危険度の時間平均を求める((5)式)。この危険度の全障害物の平均値を障害物回避に対する評価値とする。

$$V = \frac{\sum_{i=1}^n V_i}{n} \quad (3)$$

$$W_i = \begin{cases} \frac{1}{1 + \exp\left(\frac{RA_i - |RA_i|}{c_3}\right)} \cdot \frac{1}{1 + \exp\left(\frac{L_{th} - L_i}{c_4}\right)} & (L_i > r_i) \\ c_5 & \text{otherwise} \end{cases} \quad (4)$$

$$V_i = \frac{1}{l_2} \int_{l_1}^{l_2} W_i dl \quad (5)$$

ただし、  
 $W_i(t)$   $i$  番目の障害物の危険度  
 $RA_i$   $i$  番目の障害物の相対角度  
 $L_i$   $i$  番目の障害物の相対距離  
 $r_i$   $i$  番目の障害物の半径  
 $V_i$   $i$  番目の障害物の平均危険度  
 $l_1, l_2$  行動前後の障害物の中心位置  
 $n$  全障害物数  
 $RA_{th}, L_{th}, c_3, c_4, c_5$  設計パラメータ

本手法では、仮想環境の更新のタイミングは設計者により事前に与えるものとする。本論文では、ESのある決められた世代数毎に、前節で述べた方法により仮想障害物の数を増加させる。分割する仮想障害物は、実際の障害物を最も多く含むものとする。

### 5. 実験

提案手法の有効性を検証するため、走行シミュレーションを用いた実験を行った。以下に実験結果および考察を示す。

#### 5.1 移動障害物回避

まず初めに、本論文で構築した複数移動障害物回避動作計画システムを用いた自律移動ロボットの走行シミュレーションを行った。シミュレーション条件は次の通りである。

1. ロボットのゴールは事前に与えられる。
2. ロボットの初期位置、初期速度および初期探索時間は予め与える。行動開始直後の探索時間は1[sec]とする。
3. 移動障害物の台数は5台とする。

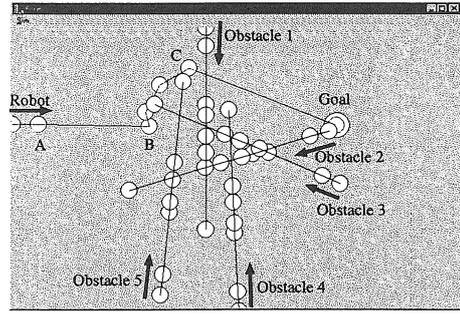


図9：走行シミュレーション結果

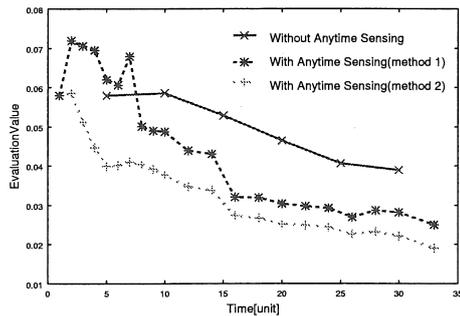


図10：探索中の評価値推移 (case 1)

4. 移動障害物は常に等速直線運動を行い、その初期位置および移動速度はある限られた範囲でランダムに与えられる。
5. 障害物との衝突判定を行うためのロボットと障害物の予測交差点の距離の閾値は、両者の半径合計の2倍とする。
6. 環境情報の更新は、ESで10世代毎に行う。

図9に走行シミュレーション結果を示す。図中の○はロボットが行動出力した位置および同時刻の移動障害物の位置を示す。ロボットはある地点で行動を出力した後、次の行動出力地点まで移動する間に行動探索を行い、次の行動を決定する。すなわち、隣接する2点間を移動する時間が探索時間であり、状況に応じて変化していることが結果より分かる。この図は、ロボットの進行方向に対し前方および左右方向から障害物が接近する場合の結果である。ロボットは行動開始直後、A地点での回避行動を探索するが状況がまだ安全であると判断し、結果として障害物を回避せずゴール方向へ近づく行動を取る。その後、B地点で状況が危険であると判断し、大きく左へ旋回することで障害物を回避する。そしてC地点で障害物との衝突はないものと判断し、まっすぐゴールへ到達している。

以上の結果より、ロボットは状況に応じた行動探索時間で、探索時間の経過と共に環境情報を更新しながら行動探索を行うことで状況に適した行動が獲得できることを確認した。

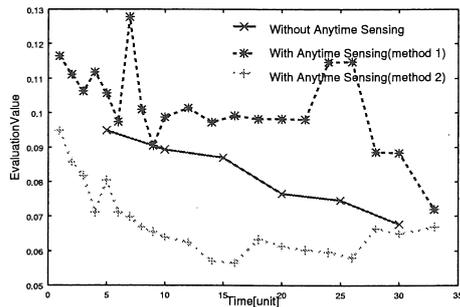


図 1.1 : 探索中の評価値推移 (case 2)

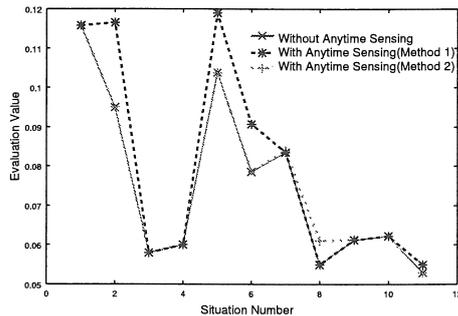


図 1.2 : 行動探索時の評価値 (初期解)

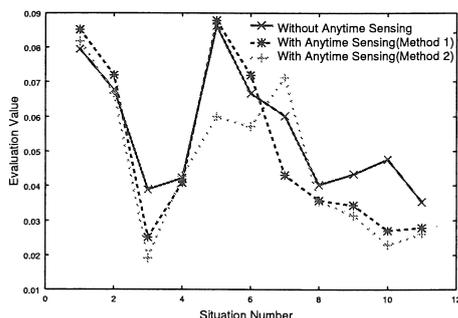


図 1.3 : 行動探索時の評価値 (最終解)

## 5.2 仮想環境とその更新による探索速度の向上

続いて、仮想環境が行動探索過程に与えた影響について検討する。図 1.0 および図 1.1 に、ロボットがある状況で行動探索を行った時の評価値推移を示す。各図は、それぞれ行動探索を 100 回試行した時の評価値平均を示す。実線は Anytime Sensing を適用せず探索開始時より全ての障害物を個別に認識した場合、また濃い破線は環境情報を制限した場合、薄い破線は環境情報の分解能を変化させた場合の評価値推移を表す。各線上の点は、ES の各世代において個体の評価終了、すなわちロボットの行動候補が決定した時間を示す。横軸は時間、縦軸は評価値を表す。同じ状況で同じ行動を取った場合でも、仮想環境の適用・不適用により評価方法が異なるため評

価値が異なるため、仮想環境を用いて獲得された評価値は、仮想環境を用いていない場合に得られる評価値へと変換した。

図 1.0 および図 1.1 より、初期行動候補を獲得するまでの時間が短縮され、探索回数が増えたことが分かる。これは、質の低い仮想環境を用いることで評価に必要な計算量を減らし、1 世代当たりに必要な探索時間が短縮した結果である。また、図 1.0 では仮想環境を用いた場合探索の方が、Anytime Sensing を適用せず、仮想環境を用いない場合に比べて探索効率が改善されている。一方、図 1.1 では環境情報を制限した時の探索効率が、仮想環境を用いない場合に比べて悪化していることが分かる。これより、環境情報を制限する手法では行動探索に必要な情報を落としてしまい、結果として図 1.1 のように探索効率の悪化を招くことが確認された。一方、環境情報の分解能を変化させる手法では、獲得したセンシングデータ全てを用いて仮想環境を構築しているため、探索効率の悪化を回避することが出来ると考えられる。

続いて、図 1.2 および図 1.3 に、ロボットが様々な状況で行動探索を行った時の初期解と最終解を示す。各図とも、横軸はロボットと移動障害物の状況を表す番号を、縦軸は評価値を表す。図 1.2 より、Anytime Sensing を用いなかった場合と環境情報の分解能を変化させる手法は同程度の初期解が得られる一方、環境情報を制限する手法では前述の 2 つの手法より劣った初期解が得られることが分かる。また、図 1.3 より、最終的に得られる解が平均で最もよい手法は環境情報の分解能を変化させる手法であることが分かる。

以上の結果より、提案手法の有効性を確認した。

## 6. 結論

本研究では、自律移動ロボットが実時間探索により行動を獲得する際の環境情報の表現方法と探索効率について言及し、分解能を変化させる環境情報の表現方法の適用を提案した。本論文では具体例として複数移動障害物回避問題を扱い、提案手法の探索効率が最も良いことを確認した。

本論文で扱った複数移動障害物回避は、移動障害物の行動を等速直線運動に限定することで問題を単純化し、動作計画システムの構築を行った。これは、提案手法の有効性を明確に示すためであった。しかし、本研究で目指すところは動的未知環境における実時間探索であり、移動障害物の行動における不確実性を考慮したシステムを構築しなければならない。そこで、今後の課題として、提案手法を用いて動的未知環境における不確実性を考慮したシステムの構築を行い、実機実験を行う予定である。

さらに、本論文ではセンサ情報として、超音波センサ等で獲得出来る情報を扱った。今後の課題として、獲得できる情報量の多さから近年注目を集めている視覚を持つ自律移動ロボットにおける Anytime Sensing について検討を行う予定である。

## 参考文献

- [1] 渡辺他：“あいまい行動型制御”，日本機械学会論文集 (C 編)，Vol.64, No.620, pp.162-170, 1998
- [2] 浅田他：“視覚に基づく強化学習によるロボットの行動獲得”，Vol.13, No.1, pp/68-74, 1995
- [3] 青木他：“多段階学習による障害物回避のための行動獲得”，日本ロボット学会誌，Vol.15, No.4, pp.648-656, 1997
- [4] 中村他：“Instance-Based Classifier Generator による自律移動ロボットの行動獲得”，日本ロボット学会誌，Vol.17, No.3, pp.381-389, 1999
- [5] 坪内他：“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”，日本ロボット学会誌，Vol.12, No.7, pp.1029-1038, 1994
- [6] 井上他：“複数移動障害物の行動予測に基づく自律移動ロボットのオンライン障害物行動計画”，日本ロボット学会誌，Vol.15, No.2, pp.249-260, 1997
- [7] 藤澤他：“自律移動ロボットにおける実時間行動探索”，日本ロボット学会誌，Vol.17, No.4, pp.503-512(1999)
- [8] S.Zilberstein et. al:”Anytime Sensing, Planning and Action: A Practical Model for Robot Control”, Proc. of the 13th International Joint Conference on Artificial Intelligence, Vol.2, pp.1402-1407,1993
- [9] K.Doki, S.Hayakawa, T.Suzuki, T.Aoki, S.Okuma :” Intelligent Control for Autonomous Mobile Robot based on Virtual Environmental Information”, J. of Robotics and Mechatronics, Vol.13, No.2, pp.114-124, 2001
- [10] R.Salomon:”Increasing Adaptivity through Evolution Strategies”, Proc. of the 4th International Conference on Simulation of Adaptive Behavior(SAB96), pp.411-420, 1996
- [11] 鈴木：“情報量規準による統計解析入門”，講談社サイエンティフィク

(受理 平成15年 3月19日)