

移動ロボットの経路探索と空間認識に関する研究

A Path Find and Workspace Recognition for Mobile Robot

○戸田 肇*, 渡部慶二*

○Hajime Toda*, Keiji Watanabe*

*山形大学工学部

*Faculty of Engineering, Yamagata University

キーワード： 移動ロボット (mobile robot), レイヤ構成 (layerd system), 経路探索 (path find), 空間認識 (workspace recognition), リージョン (region), 可到達性 (possibility of reaching)

連絡先： 〒992 米沢市城南山形大学工学部電子情報工学科 渡部研究室

戸田肇, Tel.: (0238)26-3326, Fax.: (0238)26-3326, E-mail: hajime@yz.yamagata-u.ac.jp

1. はじめに

閉ざされた環境以外でロボットに自立した行動を行わせることは困難な問題である。その困難さは、フレーム問題を完全に解くことが出来ないことに起因している^{1),2)}。

フレーム問題とは、限定された情報処理能力しか持たないシステムが、その能力をはるかに上回る複雑性を持つ情報をどのように扱うかという問題である³⁾。

フレーム問題に関する現実世界の例をひもといてみよう。現実世界では、想像もつかないような出来事が起こることは良くある。自動車の運転を例にとると、熟練した職業ドライバーが事故を引き起こす可能性は低いとはいえ全くゼロではない。事故が起こるのは熟練した職業ドライバーでも予測できないことが起こるからである。これは、特定のタスクに熟練した人間ですらフレーム問題を解決できないことを示している。逆に言うと、そのような予測不可能性が現実世界を規定しているとも言えよう。

このような現実世界の多様性により、あり得る全ての状況に対処しようとするロボットの行動規則をインプリメントしようとする、膨大なプログラミング作業が必須となる。また、プログラム作成者が想定していない状況が発生した場合、このような方法論によっ

て構築されたロボットは行動不可能になる。この事実は学習アルゴリズムを搭載したロボットが、特定の環境について学習を深化させると、環境変化に対する適応力が減少する⁴⁾ことにも関連する。

本研究ではそれぞれが独立した単純な機能層を組み合わせることによって、相当な自由度を持つロボット行動が得られることをシミュレーションによって明らかにする。シミュレーションにあたっては、後述する小型移動ロボット想定し、以下の3つの状況を扱う。

- ロボットが作業空間についての情報を持たない場合の行動
- ロボットが作業空間についての情報を持たない場合の記憶形成
- ロボットが記憶として作業空間についての情報を持つ場合の行動

2. 想定するロボット-Khepera

本研究ではFig.1に示す小型の移動ロボット Khepera を対象とする。これはシミュレーションを行う上でロボットのサイズ、走行性能、センサ構成を規定する必要からである。ただし、考察するロボットアーキテクチャは一般性を失わないように留意する。

本節ではこの Khepera の外観を示すと共に、Khepera の持つ機能を紹介する。

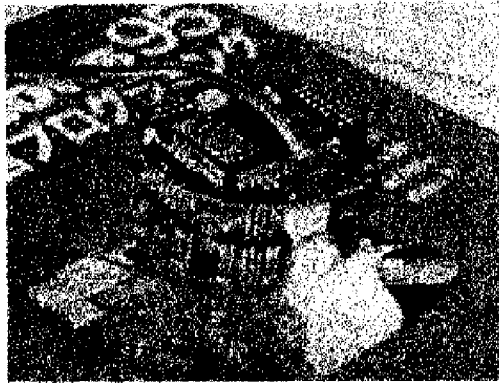


Fig. 1 移動ロボット Khepera の外観

Khepera は2つの車輪によって移動することが出来る。2つの車輪にはパルスエンコーダが装備されておりロボットの速度と移動距離を検出可能である。また、本体の周囲に8個の赤外線センサを持つ (Fig.2)。それぞれの赤外線センサは障害物との距離に応じて0~1023の値を出力する。

Khepera は本体にCPUとメモリを持っており、行動のためのプログラミングが可能である。また、シリアルインターフェイスを装備しているため、ホストコンピュータと有線通信を行うことが可能である。従って Khepera 自身の容量を超過する複雑なプログラムをホストコンピュータに置き Khepera をコントロールすることも可能である。

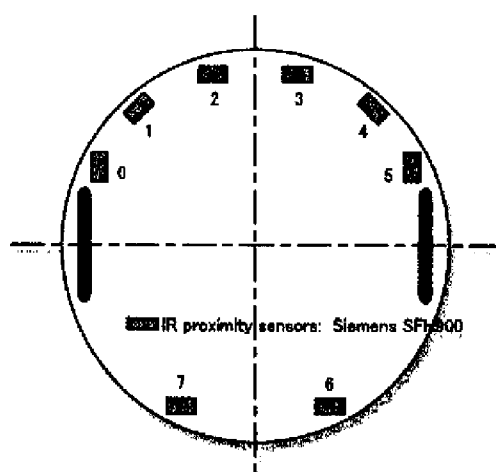


Fig. 2 Khepera における車輪とセンサの配置

3. レイヤ構成ロボット

本研究ではロボット行動を生成するために、単機能で完結したレイヤを重ねた構成をとる Fig.3.

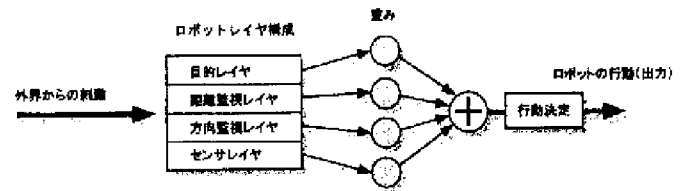


Fig. 3 ロボットアーキテクチャ

レイヤ構成ロボットでは、各レイヤの出力が統合されてロボットの行動が決定される。

ここで、各レイヤは自らが考えるロボットにとって望ましい行動-センサレイヤは障害物から出来るだけ離れるような出力、距離監視レイヤは出来るだけゴールに向かい続けるような出力-を行う。各レイヤの出力はベクトル \vec{v}_i で表現される。ここで、 $|\vec{v}_i|$ はレイヤの主張がどれほど強固なものかを表し、 $\angle \vec{v}_i$ はレイヤが考えるロボットにとって最も望ましい進行方向を表現する。

ロボットの行動決定は、各レイヤの出力を重み付けた後、それらを合成することで行われる。したがって、行動決定とは単にベクトルの合成であるといえる。

簡単な例として、センサレイヤと方向監視レイヤの2つのレイヤのみを考えた場合の行動決定機構の動作の様子を Fig.4に示す。

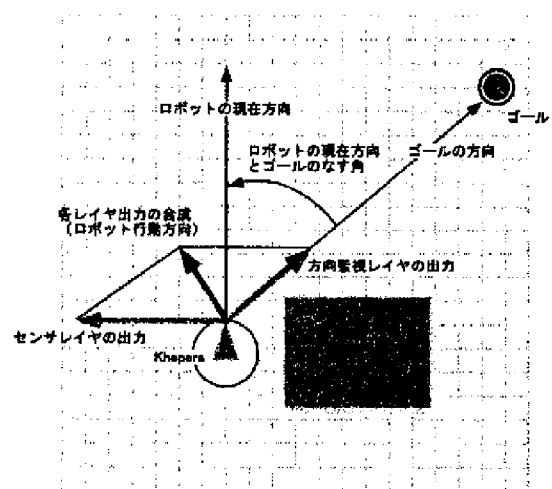


Fig. 4 意志決定機構の動作模式図

センサレイヤ・方向監視レイヤの働きは粗く言えば次のようにまとめられる。

- センサレイヤ：障害物情報を検出し、出来るだけ遠ざかるよう指示を発行する

- 方向監視レイヤ：ロボットの方向ベクトルが作る角度とロボットの現在位置と目的地を結ぶベクトルがなす角との差を小さくするような指示を発行する

Fig.4において、センサレイヤは障害物から遠ざかる様な方向を指すベクトルを出力している。また、方向監視レイヤは目標位置方向を指すベクトルを出力している。これら2つのベクトルを重み付けした結果が同図の灰色太線矢印である。Fig.4にはこのベクトルの合成によってロボット行動方向が決定される様子が示されている。

4. レイヤの詳細

本節では各レイヤに共通する要件について述べ、その後代表的なレイヤについて機能説明を行う。

4.1 各レイヤに共通する要件

レイヤの出力はベクトルであり、式1の仕様を満たすことが必要である。

$$\begin{cases} 0 \leq |\vec{v}_o| \leq 1023 \\ 0 \leq \angle \vec{v}_o < 360 \end{cases} \quad (1)$$

また、ロボット行動プログラムはオブジェクト指向で設計される。そのためにレイヤに科せられる要件は、レイヤオブジェクトが必ずレイヤ基底クラスから派生されなければならないというものである。派生により、レイヤはロボット行動に対して2つのエントリポイントを持つ。

エントリポイントは、メソッド `Layer::Output` と `Layer::Work` である。前者は戻り値としてベクトルを出力するメソッドである。後者はロボットや他のレイヤが公開している情報及び手続きを自由に利用し、レイヤ独自の処理を行うために設けられている。

オブジェクト指向による実装の特徴は、オブジェクト間のインターフェイスが規定されれば、新たな独自オブジェクトを実装しやすい⁵⁾こと、既存の機能を継承することによって、より専門化した機能を実現するレイヤを構成出来ること、ランタイムに動的にレイヤを組み込むことが出来ること等が挙げられる。

4.2 センサレイヤ

センサレイヤはまずロボットの赤外線センサから受けた情報を処理する。処理の結果、センサレイヤ

は最も近い障害物がどの方向にどれだけの距離で存在するかを得る。その後、障害物との距離をセンサ強度 Vol に換算しなおし、大きさが Vol 、向きが障害物方向と反対になるベクトルを出力する。

したがって、重みの調整によってセンサレイヤの出力を優先すると、ロボットは障害物から遠ざかる傾向を示す。

4.3 方向監視レイヤ

方向監視レイヤはロボットの現在位置と目標位置との関係から出力を決定する (Fig.5)。

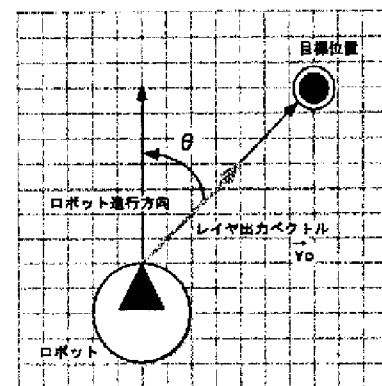


Fig. 5 方向監視レイヤの出力決定

ロボットの進行方向を表わす単位ベクトルを導入し、これを \vec{v}_R とする。また、ロボットの現在位置と目標地点を結ぶベクトルを \vec{v}_g とする。ロボットの行動目標が目標地点に到達することである場合、 \vec{v}_g を目標進行方向と考えることが出来る。よって方向監視レイヤの出力ベクトル v_o とした時、 $v_o // v_g$ である。 $|v_o|$ については、現在進行方向と目標進行方向の食い違い $\theta = |\angle \vec{v}_g - \angle \vec{v}_R|$ をレイヤ出力要件を満たすように規格化し使用する。

4.4 ランドマイズレイヤ

ランドマイズレイヤは乱数によってランダムベクトルを出力する。本研究ではランドマイズレイヤの出力をロボットのランダムウォークに利用する。

4.5 リージョン生成レイヤ

リージョンという語が本研究においてどのように規定されるかに付いては次節で詳しく解説する。

ここでは、リージョン生成レイヤはロボットが1ステップ移動する度に新しいリージョンを生成するというのみを述べるにとどめる。

また、リージョン生成レイヤは Work メソッドのみが機能の提供に参与しており、Output メソッドについては、零ベクトルを出力するような実装になっている。したがって、シミュレーションのパラメータを示す各表 (Tbl.1~4) ではリージョン生成レイヤの重み値を省略した。

5. セーフ領域・リージョン

本節では、本研究で提案するロボットの行動決定において重要な位置を占めるリージョンの考え方とその生成アルゴリズムについて述べる。

本研究においてリージョンとは、ロボットが安全に行動できる領域と規定する。ロボットの作業空間は平面座標に赤外線センサによって観測した障害物反応の強度をマップしたものである。作業空間を視覚化すると Fig.6 に示す様な三次元プロットが得られる。この作業空間中でセンサ強度が低い部分を被う矩形をリージョンと呼称する。

今、ロボットの作業空間が Fig.6 に示すような、センサ強度マップとして与えられた場合を考える。Fig.7 は Fig.6 を単純化したものである。

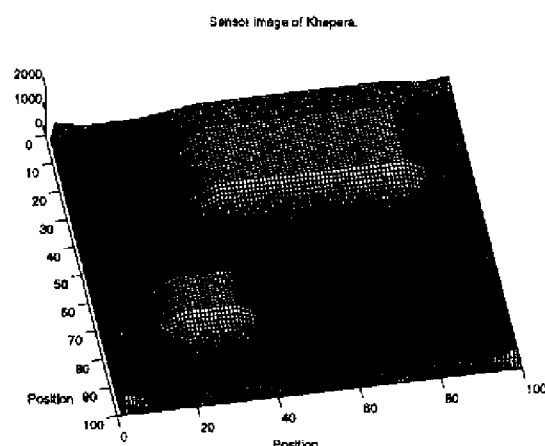


Fig. 6 作業空間例

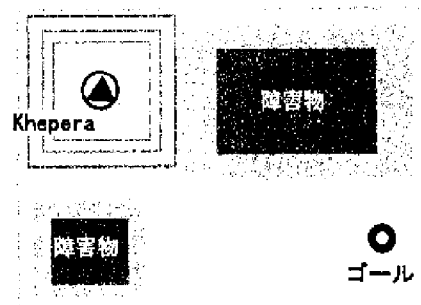


Fig. 7 作業空間の簡略図

リージョンの生成規則を Fig.8 のように定める。

while リージョン内部に $th < mug$ を満たす点が存在しない
(ただし tk は閾値, mug はその座標におけるセンサ強度)

Step1: 自分自身を囲む正方リージョンを広げて行く

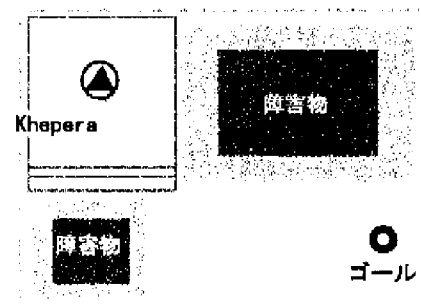


endwhile;

while まだ4辺のうちいずれかを外側に移動しても $th < mug$ を満たす点が現れない

更に矩形領域を拡大できる

Step2: 移動可能な辺を矩形の外側に向かって移動する



endwhile;

Fig. 8 リージョン生成規則

ロボットが1ステップ行動する毎に Fig.8 に示したアルゴリズムによってリージョンを生成すると、ロボットがリージョンの外周部分に近づくにつれて、新たなリージョンが発見されることになる。

例えば、Fig.7 において、第一番目のリージョンの内部をロボットが徘徊することを考える。リージョンの中心部付近では各ステップ毎に同一のリージョンが生成される。しかし、ロボットがリージョンの右下隅に接近すると新たなリージョンが生成される。

また、Fig.8 に示したアルゴリズムによって生成されたリージョン内部には障害物が存在しないので、ロボットは安全に行動できる。

次節では、このリージョンの考え方を使用した経路探索方法について考察する。

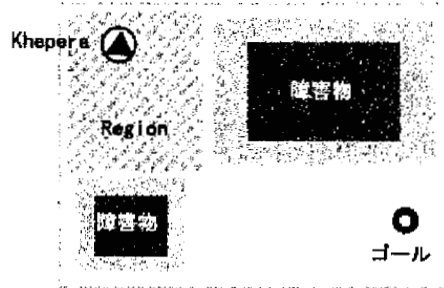
6. リージョンに基づいたサブゴール設定

本節では、サブゴールの概念を紹介し、その設定方法について解説する。

サブゴールとは、目標位置(ゴール)に代わる、ロボットの当面の目標位置を意味する。

サブゴール設定アルゴリズムとロボットの行動規則を Fig.9 に示す。

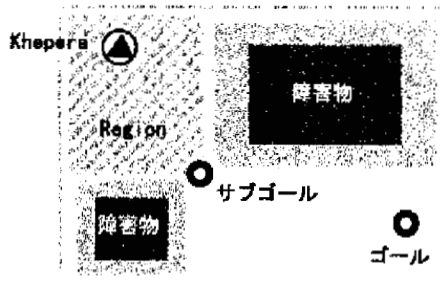
```
while ゴールに未到達 do
  Step1: 閾値をthとしてリージョンを生成する。
         (ただしthはセンサのノイズを考慮した小さい値)
```



```
  Step3: 前回求めたRegionが有れば、今回求めたRegionが
         前回求めたものと似ている場合thを増加させて
         Step2を再試行
```

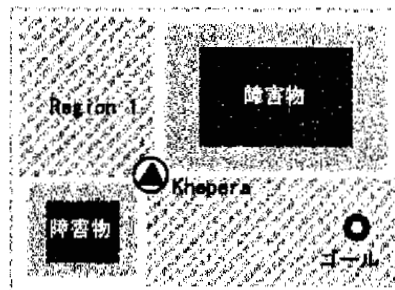
```
  if Regionの内部に目標地点が入っている
    Step4: 何もしない
```

```
  else
    Step5: 四角形の頂点及び4辺の中点のうち、目標
           地点に最も近いものをサブゴールとして設定する
  endif;
```



```
  Step6: サブゴールが設定されている時はサブゴールに
         サブゴールが設定されていない場合はゴールに向かう
```

```
Next Step1~2 : 説明の便宜上次回のStep1~2を
               ここに示した
```



```
endwhile;
```

Fig. 9 サブゴール設定アルゴリズム

Step1 及び Step2 では Fig.8 に示した、アルゴ

リズムにしたがいリージョンを生成する。作業空間が動的に変化するが、ロボットの位置検出に大幅な誤差が出ない限り、リージョン内に障害物は存在しないことになる。

Step3 終了後、リージョン内部にゴールが含まれるかどうかを求め、その後の手続きを変更している。リージョン内部にゴールが含まれる場合は、サブゴールは設定せず (Step4)、ロボットは単にゴールに向かって進行する。これに対して、リージョン内部にゴールが含まれない場合、サブゴールを設定し、ロボットはサブゴールを当面の目標位置として進行する (Step5)。サブゴール設定の基準は Fig.9 に示した通りである。

これにより、ロボットの目標位置への進行はサブゴール設定とサブゴールへの到達の繰り返しの帰着される。このとき、ロボットの経路はリージョンを連ねたものの内側になる。そこで、この手法をリージョン連結法と呼称する。

リージョン連結法では、繰り返しの最中に Step1、Step2 で、前回と全く同じリージョンを作成してしまう可能性がある。これは、先の Step1、Step2 であまりに保守的なリージョンを決定した場合に起こりうる。Step3 はこのような問題を解決する。Step3 でロボットはより危険な経路を選択する。つまり、障害物に近づくリスクと引き替えに、新たなリージョンを獲得しようと試みるのである。具体的にはリージョン拡大の停止条件にあたる閾値 th を上げ、より広範囲なリージョンを獲得する。これによって、新たな行動範囲が獲得できる。

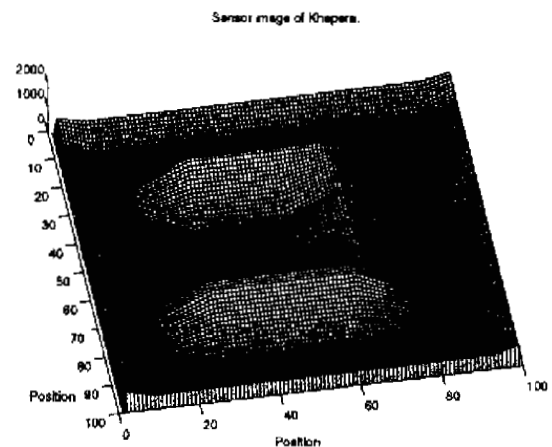


Fig. 10 C 形状の作業空間

思考実験を繰り返してみると、このアルゴリズムは Fig.6 の様な単純な作業空間については上手く機能

する。ただし、このままでは Fig.10のような作業空間では行き詰まってしまうことが考えられる。

この問題を解決するためには、上記のアルゴリズムにリージョン外周探索等の機構を加えると良いだろう。つまり、サブゴールに到達した後、複数回のリージョン作成を試行したにも関わらず、有効なパスを発見できない場合、リージョン内部を移動し、新たなリージョン生成を試みるのである。

これによってC形状マップ等にも対応できると考える (Fig.11)。

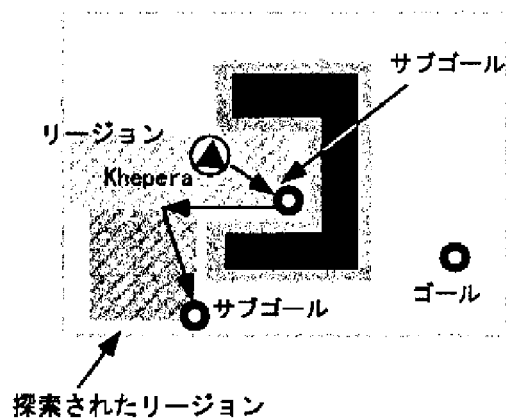


Fig. 11 パス発見の手法の一例

7. リージョン連結法における可到達性

本節では前節で述べたリージョン連結法に関して、ロボットが目標位置に到達可能であるための条件を考察する。

明らかに、以下のことが言える。

ある作業空間 S において、ロボットの初期位置・目標位置をそれぞれ P_s, P_g と定めたとき、リージョン生成アルゴリズムによって生成されるリージョン r_1, r_2, \dots, r_n が、 $P_s \in r_1, P_g \in r_n$ を満たし、ロボットが通過可能な幅を持ち、連結している場合、ロボットは目標位置に到達可能である。

8. シミュレーション

シミュレーションで使用するロボットの作業空間を Fig.12に示す。

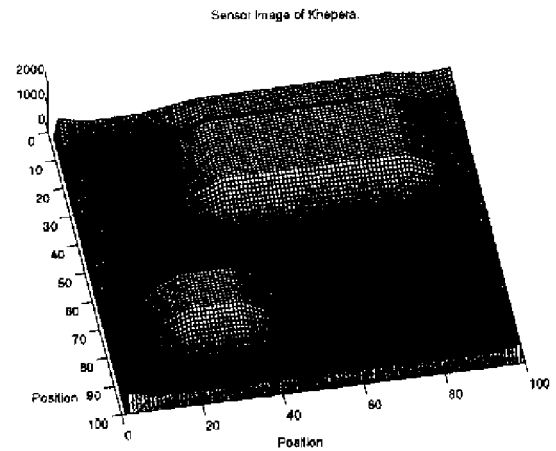


Fig. 12 シミュレーションで使用する作業空間

8.1 ロボットが地図情報持たない場合の行動

行動目標を初期位置 (10, 10) から目標位置 (80, 80) に到達することに設定しロボットが地図情報を持たず、初期位置と目標位置のみが与えられたときの行動をシミュレーションによって明らかにする。

この時、上記目的を達成するために設定した各レイヤの重みを Table.1に示す。なお、今回はリージョン生成レイヤが稼働していない。

Table 1 各レイヤの重み

レイヤ名	センサ	ランドマイズ
重み	0.20	0.45
レイヤ名	方向監視	距離監視
重み	0.60	0.10
レイヤ名	目的	/
重み	0.80	/

シミュレーション結果を Fig.13に示す。

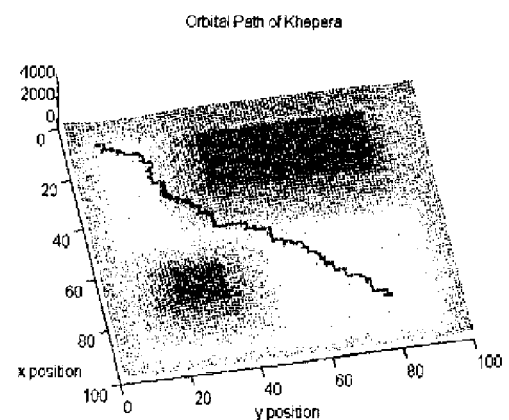


Fig. 13 シミュレーション結果

また、Table.2に示す通りレイヤの重み変更し、その他の各シミュレーションパラメータは同一な場合のシミュレーション結果を Fig.14に示す。

Table 2 各レイヤの重み

レイヤ名	センサ	ランドマイズ
重み	0.68	0.45
レイヤ名	方向監視	距離監視
重み	0.60	0.10
レイヤ名	目的	/
重み	0.80	/

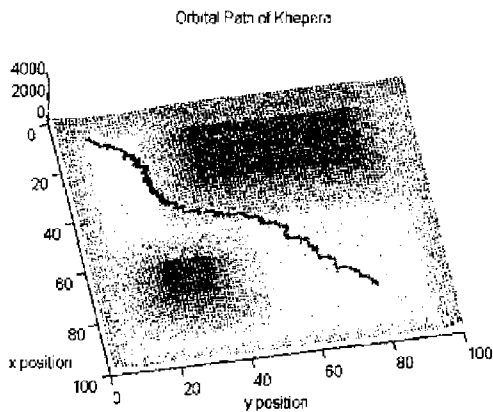


Fig. 14 シミュレーション結果

更に Tbl.1に示した重み値に対して、センサレイヤの重みのみを 0.3, 0.6, 0.9 と変化させてシミュレーションを行った結果を Fig.15にまとめて示す。

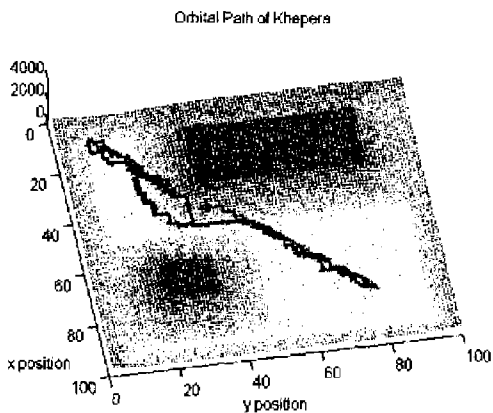


Fig. 15 シミュレーション結果

Fig.15において、障害物を最も大きく迂回している経路が、センサレイヤの重み 0.9, 以下順に 0.6, 0.3

の場合ロボットが取った経路である。

この結果から、センサレイヤの重みを変更しただけでも、ロボットの取る経路は様々に変化することが理解される。

8.2 ロボットが地図情報持たない場合の記憶形成

ロボットが自らが置かれた作業空間に関して、ア prioriな知識が全くない場合、その方法はどうか、作業空間中を動き回ることによって、情報を獲得する必要がある。

情報収集の方法には、障害物に沿って移動することにより、障害物のトポロジカルな性質を把握する方法⁶⁾や、作業空間の幾何学マップを作成して行くもの⁷⁾等、高度な方法が提案されている。

本研究においては、ロボットが作業空間の情報を持たない場合、作業空間中を障害物を避けながらランダムウォークする。ランダムウォークはランドマイズレイヤの重みを調整することで容易に実現可能である。その際、センサから得た各地点の障害物情報を記録し、次の試行に利用する。

本節では記憶の形成を目的としたランダムウォークのシミュレーションを行う。この場合ロボットは初期位置のみを与えられる。ロボットの行動目的はランダムウォークを行い、作業空間の情報を収集しながら、マーキングされた場所を発見することにある。

上記目的を達成するために設定した各レイヤの重みを Table.3に示す。なお、今回はセンサレイヤとランドマイズレイヤ以外は稼働していない。

Table 3 各レイヤの重み

レイヤ名	センサ	ランドマイズ
重み	0.20	0.75
レイヤ名	方向監視	距離監視
重み	/	/
レイヤ名	目的	/
重み	0.80	/

ランダムウォークのシミュレーションによって得たロボットの記憶を Fig.16に示す。

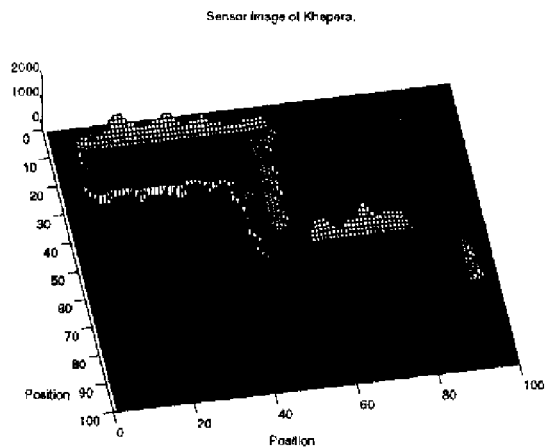


Fig. 16 シミュレーション結果

Fig.16より、大部分の作業空間をセンサ強度のマップとして記憶していることが理解される。ただし、Fig.16に示す情報を得るのに要したステップ数はおよそ1700であった。したがって、ランダムウォークは、実験系に適用するにはあまりに非効率な方法であると思われる。また、ランダムウォークでは、作業空間のあらゆる地点を探索することは保証できない。これらの問題点を改善することは今後の課題である。

8.3 ロボットが記憶として地図情報を持つ場合の行動

本節では、ロボットがランダムウォークによって作成した地図情報を持つ場合の行動をシミュレーションによって明らかにする。

シミュレーションにあたってはセンサレイヤが観測するセンサ強度に $0 \leq N_z \leq 127$ のホワイトノイズを印加した。ノイズの印加によって、ロボットが観測する作業空間は Fig.17のようになる。

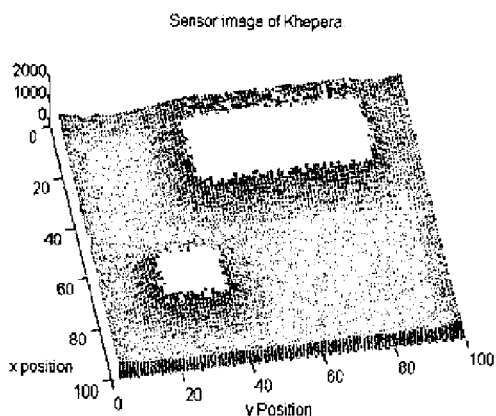


Fig. 17 センサ観測値にノイズを印加した場合、ロボットが観測する作業空間の例

このシミュレーションは、まずノイズのある環境で前節同様のランダムウォークを行い、ロボットのメモリに記憶を構成する。

次に、記憶として得た作業空間内で、バーチャルなロボットを行動させることを想定し、リージョン生成レイヤを含むあらゆるレイヤを稼働させた仮想ロボットを走行させた。この時、リージョン生成レイヤが生成したリージョンをもとに、複数のサブゴールが設定される。

次に、作業空間の記憶と仮想的な行動で得たサブゴール列を利用して、ロボットに実際の作業空間を行動させた。

記憶を構成する段階では前節 Table.3に示した重み値を使用し、その後のシミュレーションでは、一貫して Tbl.4に示す重み値を使用した。

Table 4 各レイヤの重み

レイヤ名	センサ	ランドマイズ
重み	0.20	0.45
レイヤ名	方向監視	距離監視
重み	0.60	0.10
レイヤ名	目的	/
重み	0.80	/

シミュレーションの最終結果を Fig.18に示す。

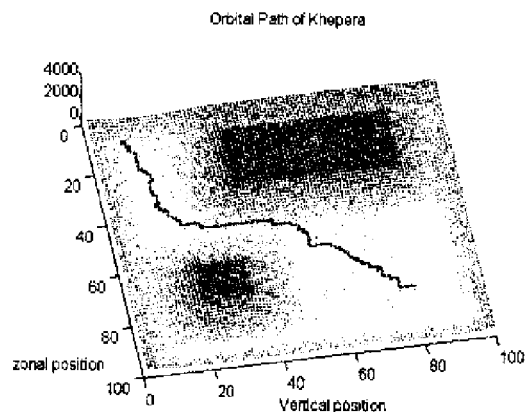


Fig. 18 シミュレーション結果

Fig.18によると、リージョン内の安全な箇所を通過点として、障害物に近寄ることなく、目的位置に到達していることが理解される。

9. 終わりに

本研究ではレイヤ構成ロボットを提案し、シミュレーションによって重みの違いでロボットの経路がどのように変化するかを示した。また、ランダムウォークによって作業空間をロボットのメモリ内に構築することを試みた。加えて、リージョンの概念及び、リージョンに基づくサブゴールの設定方法を提案し、記憶された作業空間を利用して、行動をシミュレートした後、実際の行動を行う手法を提案した。

その結果、記憶を利用した行動では、ロボットが最も上手く障害物を回避し目標に到達する結果を得た。

今後の課題として、現在重み値の決定はヒューリスティックな手法によっているので、遺伝的アルゴリズム等による最適重み値の探索を検討したい。また、スタティックな値となっている重み値を、動的に変化させるアルゴリズムの研究、探索アルゴリズムの強化、及びロボット行動のロバスト性に関する定性的な研究を今後の課題としたい。

最後に、研究の過程で山田 功氏をはじめとする研究室の諸兄には貴重な御意見、多大な御協力をいただいた。この場をお借りし心より感謝したい。

参考文献

- 1) 松原 仁, 橋田浩一: 情報の部分性とフレーム問題の解決不可能性, 人工知能学会誌, 4-6, 659/703 (1989)
- 2) 松原 仁: フレーム問題をどうとらえるか, 日本認知科学会編「認知科学の発展」, 2, 155/187 (1990)
- 3) 松原 仁: 人工知能におけるロボットの役割, 日本ロボット学会誌, 14-4, 478/481 (1996)
- 4) 星野 力: ロボット行動の進化とその頑健性, 計測自動制御学会論文集, 33-6, 533/540 (1997)
- 5) Steve Maguire, 関本健太郎 (訳): Writing Solid Code, 73/100, アスキー出版局 (1995)
- 6) 山本元司, 牛見宣博, 毛利 彰: 自己位置情報を用いない移動ロボットのセンサベースト・ナビゲーション, 第15回計測自動制御学会九州支部学術講演会予稿集, 167/170 (1996)
- 7) 大石隆文, 吉田一雄, 近藤駿介: 自己生成自己組織化ニューラルネットワークを用いた自立移動ロボットの環境認識とナビゲーション, 計測自動制御学会論文集, 33-3, 203/208 (1997)
- 8) R. A. Brooks: A Robust Layered Control System for a Mobile Robot, IEEE J, RA-2-1, 14/23 (1986)
- 9) R. A. Brooks: Intelligence without representation, AI Journal, 47, 139/159 (1986)
- 10) Rudy Racker, 日暮雅道 (訳), 山田和子 (訳): Artificial Life LAB, アスキー出版局, (1996)
- 11) 谷内田正彦: ロボットビジョン, 昭晃堂, 41/51 (1990)
- 12) 杉原厚吉: ボロノイ図を通してみると, 日本評論社, 数学セミナー7月号, 74/78 (1996)
- 13) Herbert Edelman, 今井 浩 (訳), 今井 桂子 (訳): 組み合わせ幾何学のアルゴリズム, 共立出版株式会社, (1995)
- 14) P. ベルジェ, Y. ボモウ, Ch. ビダル, 相澤洋二 (訳): カオスの中の秩序, 産業図書, 58/76 (1992)
- 15) 矢野健太郎監修 宮本敏雄訳: 数学ハンドブック, pp.1132-1133, 森北出版株式会社, (1985)
- 16) 合原一幸 (編): 複雑系が開く世界, 別冊日経サイエンス, 158/163 (1997)