

# ニューラルネットを用いた移動ロボット車の 経路生成における計算時間の短縮

## Reduction of Calculating time for Path Planning of Moving Robot Vehicle Using Neural Networks

○柴崎 章, 大友照彦, 大槻恭士

○Akira Shibasaki, Teruhiko Ohtomo, Takashi Otsuki

山形大学

Yamagata University

キーワード : ニューラルネットワーク (Neural Network), 移動ロボット車 (Robot Vehicle), 経路生成 (Path Planning)  
同定器モジュール (Identification module), 繰り返し構造型ネットワーク (Cascaded network)  
ルックアップテーブル (Look Up Table)

連絡先 : 〒992 米沢市城南4-3-10 山形大学 工学部 共通講座 大友研究室  
柴崎 章, Tel.: (0238)26-3382, Fax.: (0238)26-3388, E-mail: shiba@banana.yz.yamagata-u.ac.jp

### 1. はじめに

現代生活において自動車は必要不可欠な存在であり、将来的には自律移動ロボット車の開発も望まれるところである。そのためには、可能な限り安全で効率の良い経路を生成する制御システムの確立が必要であり、これは人間社会において大変役立つものと思われる。

これまで障害物回避走行制御法として、ニューラルネットをシステム同定器に用い、それを縦続接続して力学系の流れを構成して軌道計画と走行制御を同時に行う方法について検討してきた。しかしこの方法では、複雑な環境に対して局所的最小に陥り安全な経路生成が出来なかったり、出来た場合でも経路生成に大幅に時間を要するという問題点があり、実時間制御が難しいという問題があった。

本研究では、これまでの制御法に中継点を自動的に設定する中継点自動設定方式と、経路生成のための繰り

返し構造型ネットワークに入力する運動指令の初期値をルックアップテーブルから与える方式を検討し、様々な環境における軌道の計画と経路生成時間の短縮について計算機シミュレーションを行い、その有用性を確かめたので報告する。

### 2. 制御対象

#### 移動ロボット車モデル

制御対象としてFig.1に示す前輪操舵後輪駆動の運動力学モデルを用いる。このとき、車両の重心点は地面上にあり、傾斜が無い良好な舗装路を走行するものとする。また、移動ロボット車におけるローリングとピッチングの影響は十分小さいものとし、前輪及び後輪の特性が左右二輪の総合特性を代表するものとする。

本実験で用いる移動ロボット車モデルは、

$$\text{状態 } Z = (x, y, \psi, \dot{x}, \dot{y}, \dot{\psi})^T \quad (1)$$

入力  $U = (\mu, \theta)^T$  (2)

状態方程式  $\frac{dZ}{dt} = F(Z, U)$  (3)

ロボット車の現在の座標は $(x, y)$ 、ロボット車の姿勢角は $\psi$ である。 $\mu$ は駆動係数(ロボット車のアクセル、ブレーキに相当するもの)であり、 $\theta$ はステアリング角(ロボット車のハンドルに相当するもの)である。また、移動ロボット車は式(3)のベクトル値関数 $F$ の非線形性が強いので制御するのが困難なモデルとなる。

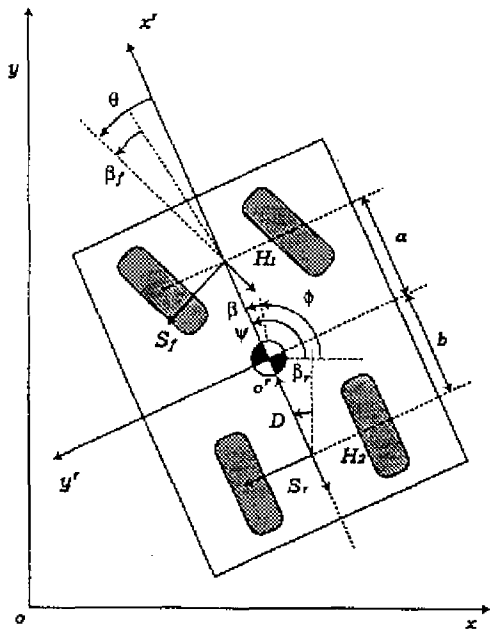


Fig. 1 移動ロボット車モデル

### 3. 制御システムの構成

#### 3.1 同定器モジュール

同定器モジュール(I.M.)の構成をFig.2に示す。同定器モジュールは移動ロボット車の動特性を同定するものであり、状態 $Z$ 、運動指令 $U$ を入力すると、1サンプリング時間( $\Delta t$ )後の状態の増分 $\Delta Z$ が出力される。その内部は、座標変換部とニューラルネット部から構成されている。同定器モジュールは内部に汎化能力を有するニューラルネットを持つために環境に対する適応力を持っている。ここで、時刻 $t$ は離散時刻 $k$ を用いて、

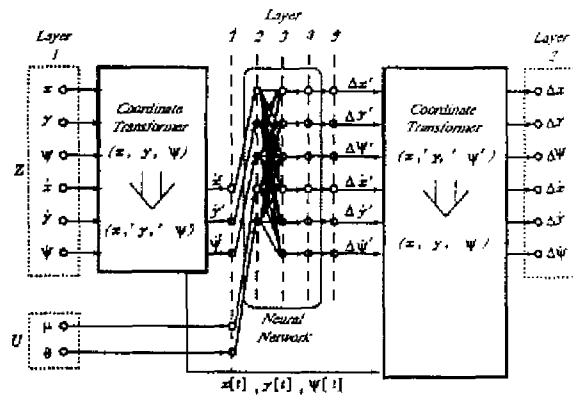


Fig. 2 同定器モジュール

$t = k\Delta t (k = 0, 1, 2, \dots)$ と表される。離散時刻 $k$ における第1層のユニットの入力を $u^1$ 、第2層のユニットを $u^2$ とすると、十分学習を行ったモジュールにおいては次式が成り立つと考えられる。

$$u^1[k] = \begin{bmatrix} Z[k] \\ U[k] \end{bmatrix} = \begin{bmatrix} X[k] \\ \dot{X}[k] \\ U[k] \end{bmatrix} \quad (4)$$

$$u^2[k] = \Delta Z[k] = \begin{bmatrix} \Delta X[k] \\ \Delta \dot{X}[k] \end{bmatrix} \quad (5)$$

但し、 $\Delta X$ 、 $\Delta \dot{X}$ は $X$ 、 $\dot{X}$ の増分である。

十分に学習したニューラルネットにおいては、式(4)、式(5)が成り立ち、動特性を学習したことになる。以後、学習により得た結合係数やしきい値を知識として利用する。

同定器モジュールに用いるニューラルネットワークは、計算機シミュレーションにより得られた9000個の学習パターンを用いて、移動ロボット車の動特性の学習を行った。学習法としてはバックプロパゲーション法を用い、学習回数を5000回とした。このとき、結合係数及び閾値は、その絶対値が0.01以下のランダムな値とし、神経回路モデルのサンプリング時間は $\delta t = 0.1[s]$ 、一つの出力層ユニットと隠れ層ユニットとの結合数を10とし、運動指令 $U$ のそれぞれの変数のとる値域を $-0.3 \leq \mu \leq 0.7, -20[deg] \leq \theta \leq 20[deg]$ とした。

### 3.2 繰り返し構造ネットワーク

本システムの離散時刻における力学系の流れは、Fig.3とFig.4の破線部を除いた部分で示されるような同定器モジュール(I.M.)を縦列接続したネットワーク構造として表現できる。これは式(3)の微分方程式のオイラー解法を示している。

$$Z[k+1] = Z[k] + F(Z[k], U[k])\Delta t = Z[k] + \Delta Z[k]$$

従って、繰り返し構造型ネットワークでは、順方向の計算により移動ロボット車の走行軌道を推定することができ、エネルギー関数により決定される誤差信号を逆伝搬することにより、運動指令 $U$ を生成することができる。すなわち、この制御システムでは制御対象の動特性を含めた軌道計画を立てることができ、軌道計画問題と動特性の問題を同時に解くことができる。また、ロボット車からフィードバックされる走行状態を現在の離散時刻 $k_0$ に対応したモジュールの入力ユニットに入力することにより推定状態が補正されるため、外乱が生じる場合にも対応できる。

### 4. 障害物回避(危険度関数)

離散時刻 $k$ における危険度関数を次式とFig.5のように定義し、第 $m$ 番目の障害物とロボット車の時刻 $t$ における位置関係モデルをFig.6のように仮定する。

$$P_{d,m}[k] = \begin{cases} \frac{r_{d,m}}{2} \{ \cos(\pi \frac{r_m[k]}{r_{d,m}}) + 1 \} & (r_m[k] \ll r_{d,m}) \\ 0 & (r_m[k] \geq r_{d,m}) \end{cases}$$

$$r_{d,m} = r_v + r_{o,m} + l_g$$

但し、 $r_m[k]$ は $k$ における障害物とロボット車間の距離、 $r_{d,m}[k]$ は危険度半径、 $r_v$ はロボット車の領域半径、 $r_{o,m}$ は障害物領域半径、 $l_g$ は緩衝長を表す。危険度関数は障害物の中心で最も危険度が高く、半径 $r_{d,m}$ において危険度が零になる関数である。

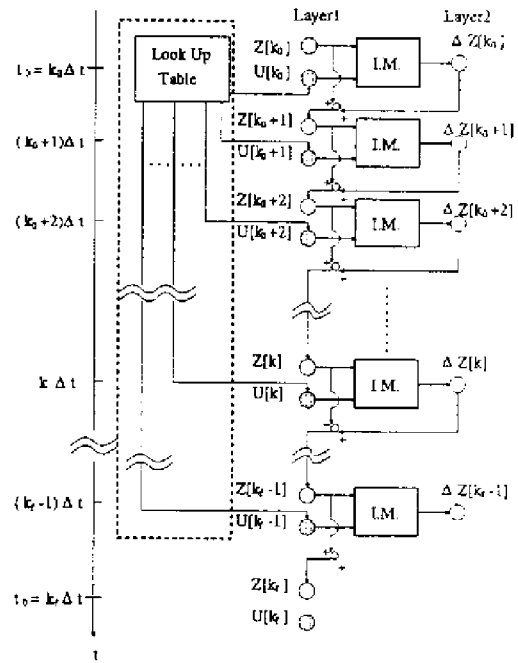


Fig. 3 繰り返し構造型ネットワーク(順方向)

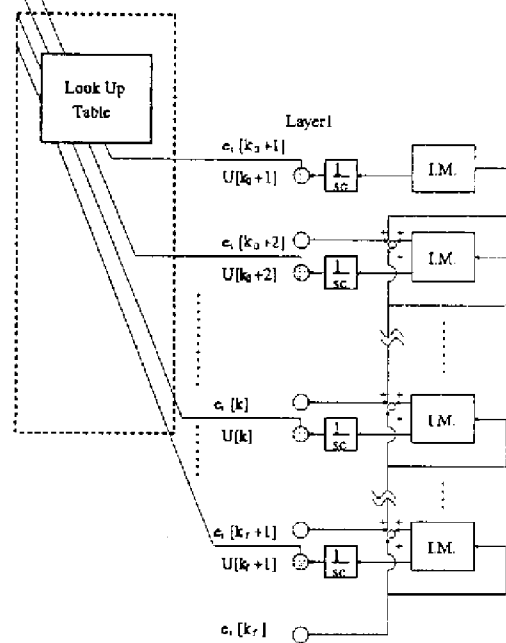


Fig. 4 繰り返し構造型ネットワーク(逆方向)

### 5. 走行制御アルゴリズム

目標時刻に目標状態を達成するのみならず、運動指令変化最小、障害物回避の観点から制御するためにエネルギー関数 $L$ を次式のように定義する。

$$L = E_t + E_u + E_o$$

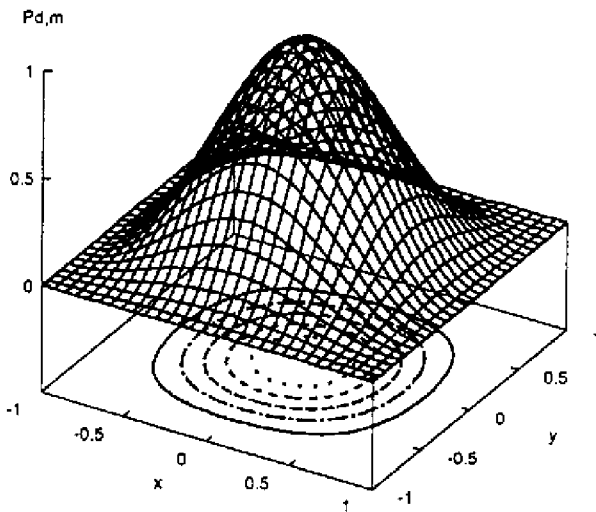


Fig. 5 危険度関数( $r_{d,m} = 1$ の場合)

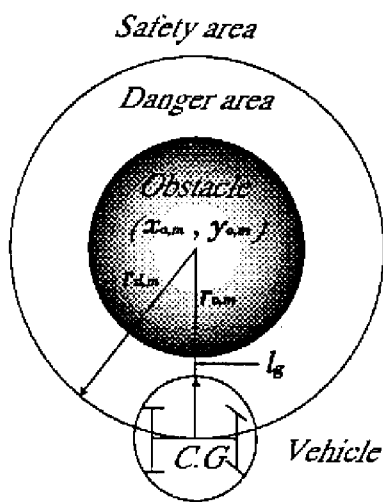


Fig. 6 移動ロボット車と障害物

ここで、 $E_t$ は目標状態に関する評価関数、 $E_u$ は運動指令変化最小に関する評価関数、 $E_o$ は障害物回避に関する評価関数を表す。エネルギー関数 $L$ が最小となるような運動指令 $U$ を誤差逆伝搬により求める。

## 6. 多角形状を持つ障害物モデルと中継点設定アルゴリズム

複雑な形状をした障害物が存在する動作空間内を逐次探索アルゴリズムにより移動する場合、Fig.7に示すようにいつも正しい道経路を探索できるとは限らない。従って、移動ロボット車が常に安全に走行するためには、動作空間内に多くのエネルギー極小軌道をつくら

ないことと簡単な空間マップの作成が必要になるものと考えられる。そこで経路探索を含めた障害物回避制御を行うため、中継点自動設定法式による回避制御システムを提案する。

まず、複雑な走行路を形成している多角形平面形状を持つ静的障害物を考慮し、そのモデルを定義する。多角形には凸多角形と凹多角形とがあるが、ここでは両者とも同等に扱う。前節の円領域で表される障害物は位置と半径から定義されるが、本多角形障害物は各頂点の位置から定義される。

中継点設定のアルゴリズムでは、中継点は各多角形障害物において内角が $180[\text{deg}]$ 以下の所に候補点を定め、障害物領域にあるものを外し、安全な走行経路上にあるものだけを残す。次に、初期位置、全ての候補点及び目標最終地を線分で結び、障害物に交差する線分を通過不可能とし、交差しない線分を通過可能とする。最後に各線分の距離を計算し、Fig.8のようにダイクストラ法で最短経路を求めた後、ロボット車が最短経路を構成する中継点を通るように繰り返し構造型ネットワークに命令する。ダイクストラ法とは、各節点への最短経路を始点の周辺から一つずつ確定し、徐々に範囲を広げていき、最終的には全ての節点への最短経路を求めるものである。このアルゴリズムで生成された最短経路は安全な走行経路上に生成されるので、この経路を構成する中継点を通して最終目標点に至る経路は常に安全であると期待できる。

## 7. ルックアップテーブル

従来の方法では運動指令の初期値を0とし運動指令の生成を行っていたが、経路生成が困難と思われる複雑な環境下では軌道の生成時間が長くなってしまふ。そこで、本実験では移動ロボット車の軌道生成を高速化するために、Fig.9に示すようなルックアップテーブルを用意する。ルックアップテーブルには、各時間毎

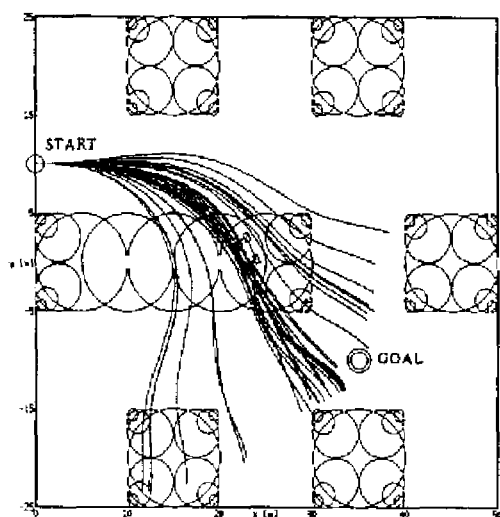


Fig. 7 従来法での経路生成

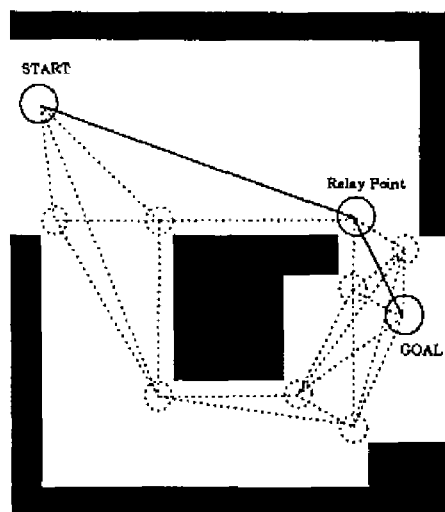


Fig. 8 中継点設定の様子

の運動指令が記憶されており、順方向計算の時、運動指令の初期値としてこれを参照する。また、繰り返し構造型ネットワークの順方向計算で最終的に生成された経路をロボット車が走行した時の参照値と近い値がテーブルに無い場合のみテーブルを更新することになっている。

ルックアップテーブルを用いることによって、経路生成における駆動係数 $\mu$ 及びステアリング角 $\theta$ を初期値の段階で最適な値に近づけることができ、その結果最適経路生成のための修正回数を減少させることによる高速化を目指している。例えば順方向計算の時、S地点から1サンプリング時間後の同定モジュールは、参照値がFig.9に示すように $Dis[k_0]$ と $Deg[k_0]$ の時、運動指令

$\mu[k_0] \sim \mu[k_0 + 9]$ 、 $\theta[k_0] \sim \theta[k_0 + 9]$ の値を読み込み、初期値として各同定器に入力する。それによってできた誤差信号を逆方向計算により運動指令を更新する。ルックアップテーブル導入後のネットワークの構造はFig.3とFig.4の破線部も含めた部分である。

本実験ではダイクストラ法で求めた直線経路上に同定器をFig.10のように等配分し、各同定器10個分の距離、ロボットの現在姿勢と10個目の同定器との相対角度を参照値とし、各同定器10個分の運動指令を検索する。更に、10個目の同定器に入力された初期値の運動指令によって出力された状態 $(x, y, \psi)$ を次の10個分の運動指令を検索する時の参照値として利用する。また、中継点を10個目の同定器の間に挟んだ場合、ロボット車の位置と中継点との延長線上で同定器10個分の距離の所に仮の目標点を作り運動指令は中継点までの同定器分だけ入力する。

ルックアップテーブル作成において、距離 $Dis$ はロボット車の初速度 $10[\text{km/h}] \sim 30[\text{km/h}]$ を $1[\text{km/h}]$ 毎にそれぞれの速度での同定器10個分の距離とした。また相対角度 $Deg$ は範囲を $|Deg| \leq 5[\text{deg}]$ と定め、 $|Deg| \leq 2[\text{deg}]$ の範囲では $0.1[\text{deg}]$ 毎に値をとり、それ以外の範囲では $1[\text{deg}]$ ずつ値をとった。以上の方法で計 $(41 + 6) \times 20 = 940$ パターンの参照値とそれに対応する運動指令のテーブルを作成した。

ここで参照値 $Dis$ を同定器10個分の距離としたのは、参照距離をロボット車が1秒間に移動する距離と定め、1秒間を神経モデルのサンプリング時間 $\Delta t = 0.1$ で割った数とした。(ロボット車の移動時間 $[s] \div$ 神経モデルのサンプリング時間 $(\Delta t = 0.1)[s] =$ 同定器数)

## 8. 結果および考察

Fig.13、Fig.16に示す環境において、中継点自動設定方式を用いて経路生成したとき(方法1)と $|Deg| \leq 2[\text{deg}]$ の範囲で $0.5[\text{deg}]$ 毎に値をとった計300パターンのテー

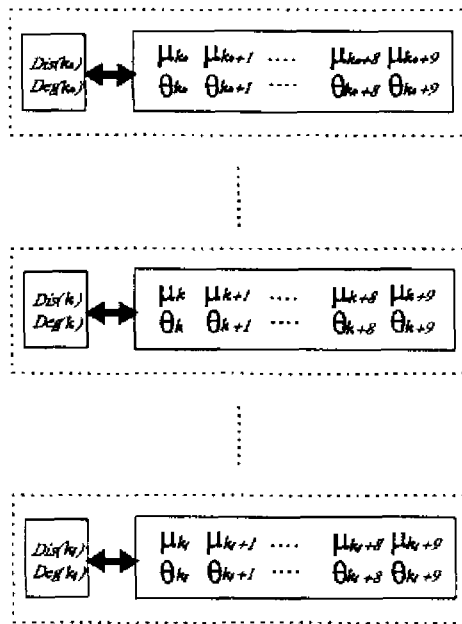


Fig. 9 ルックアップテーブルの構造

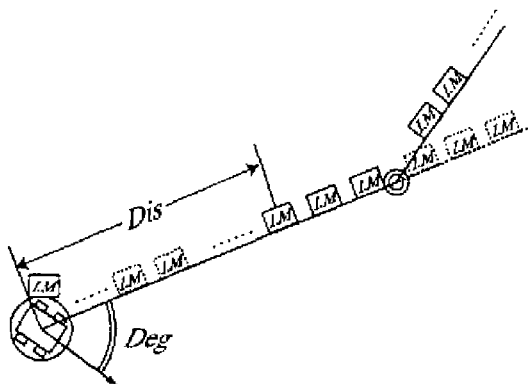


Fig. 10 テーブルの参照方法

ブルを併用したとき(方法2)、上記で説明した940パターンのルックアップテーブルを併用したとき(方法3)、の経路生成のシミュレーションを行った。この時、エネルギー関数 $L$ の値がある程度収束した時(ここでは、 $L$ の閾値を0.2と定め $L$ が0.2になったら安全な経路が生成された考える。)の思考回数とかかった思考時間、その時の $L$ の値をそれぞれ表1、表2に示す。また、それぞれの環境で方法1、方法2、方法3のシミュレーションにおいて、1回目に生成された経路とエネルギー関数 $L$ が閾値を下回った時に生成された経路を示したものをそれ

ぞれFig.13、Fig.??、Fig.15とFig.16、Fig.17、Fig.18に示す。また、それぞれの環境におけるエネルギー関数 $L$ の変化のグラフをそれぞれFig.11、Fig.12に示す。同様に、上記の2つの環境を含めた10種類の環境におけるシミュレーション結果から方法1、2、3による経路生成のための思考回数と思考時間の平均を表3に示す。

本シミュレーションから、中継点を自動設定することによっていずれの環境においても局所的最適経路に陥らず最適経路を生成することが確かめられ、ルックアップテーブルを用いることで計算時間が大幅に短縮されることが示された。Fig.13とFig.15、Fig.16とFig.18をそれぞれ比較しても解るように、ルックアップテーブルを用いることで一本目の経路生成で実際の走行経路とほぼ似たような経路が生成されている。また、Fig.11、Fig.12のグラフから解るようもルックアップテーブル併用方式の有用性が確かめられた。更に方法2と3を比較してみると、テーブルの参照値を荒くとした方法2では初回の思考で生成された経路が方法3のよりも多少のずれを生じているが、このことからはじめは精度の荒いテーブルでもより細かい参照値と対応する運動指令を更新し続けていくことでより最適な経路を初回の思考で生成できることが示された。

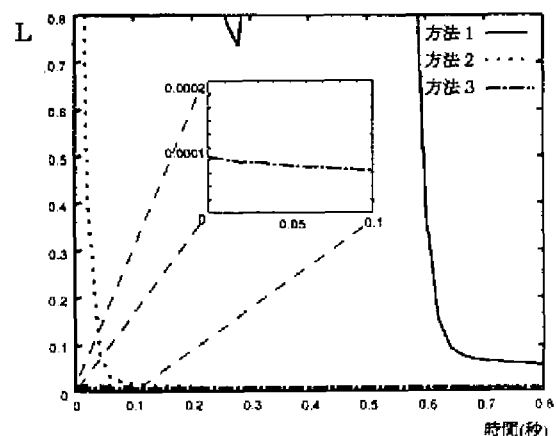


Fig. 11 エネルギー関数 $L$ の時間的变化(環境1)

Table 1 環境1

方法	回数(回)	時間(秒)	L
方法1	32	0.64	0.149
方法2	3	0.06	0.080
方法3	1	0.02	0.000085

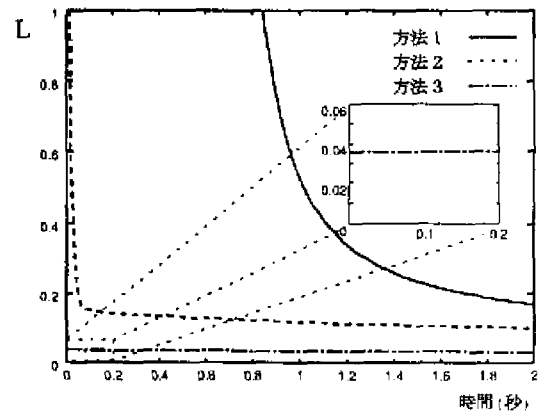


Fig. 12 エネルギー関数Lの時間的变化(環境2)

Table 2 環境2

方法	回数(回)	時間(秒)	L
方法1	86	1.72	0.198
方法2	4	0.08	0.169
方法3	1	0.02	0.035

Table 3 平均値(10環境)

方法	回数(回)	時間(秒)
方法1	80	1.60
方法2	4.3	0.086
方法3	1.7	0.034

## 9. まとめ及び今後の検討

ニューラルネットをシステム同定器として用いた移動ロボット車の経路生成システムに中継点自動設定方式とルックアップテーブルを併用した方法を検討した。その結果、従来法では経路生成が出来ないような複雑な環境においても安全な経路生成が可能となり、ルックアップテーブルを併用することで経路生成のための思考時間を大幅に短縮できることが確かめられた。今後は、中継点を通ろうとする運動指令とエネルギー関数の重みを思考回数に従って緩和することでより安全な経路を高速に生成させることの有用性や、ルックアップテーブルの運動指令をニューラルネットで構成する方法について検討する予定である。

## 10. 参考文献

- [1]小平実、大友照彦、田中 敦、岩月 正見、大内 隆夫：“ニューラルネットを用いた移動ロボット車の障害物回避制御” 信学論(D-II) vol.J79-D-II No.1 pp91-100 1996.
- [2]小平実、大友照彦、田中 敦、岩月 正見、大内 隆夫：“ニューラルネットを用いた移動ロボット車の遠心力を考慮した目標時間可変型走行制御アルゴリズム” 信学論(D-II) vol.J78-D-II No.9 pp1395-1403 1995.
- [3]小平実、岩月 正見、大内 隆夫：“ニューラルネットを用いた移動ロボット車の適応型速度制御” システム制御情報学会論文誌、5、4 pp131-138 1992.
- [4]申 東旭：“距離変換画像を用いた移動ロボットの高速度動作生成アルゴリズム” 信学論(D-II) vol.1 J76-D-II No.6 pp1194-1202 1993.

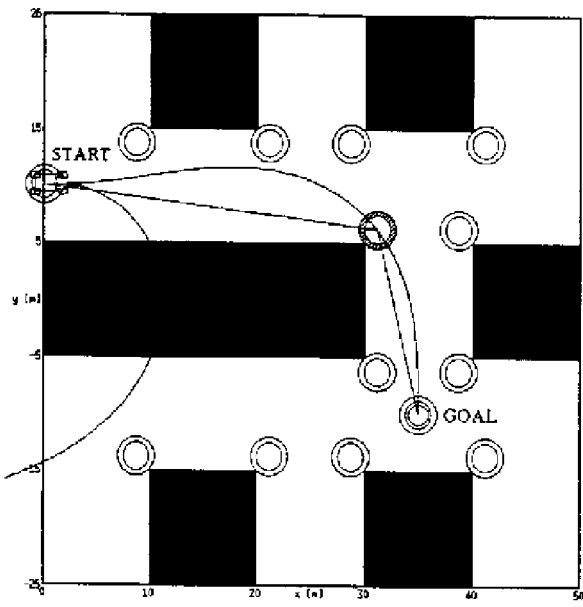


Fig. 13 方法1での経路生成(環境1)

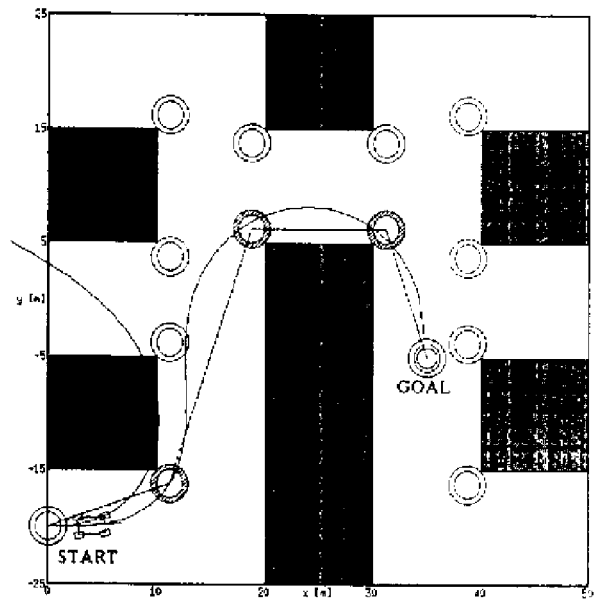


Fig. 16 方法1での経路生成(環境2)

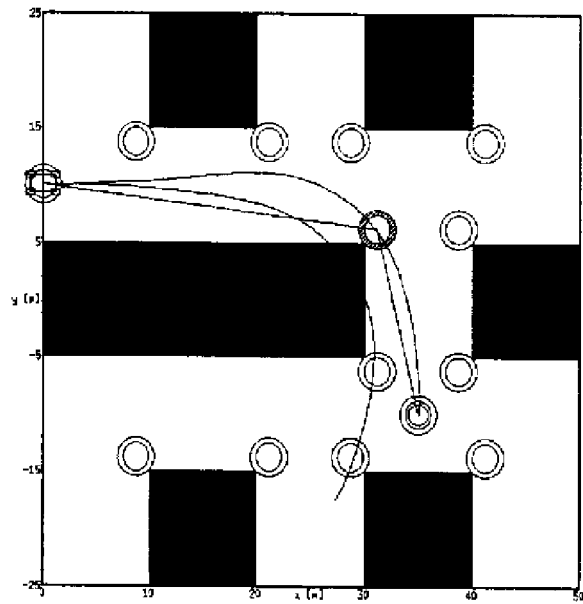


Fig. 14 方法2での経路生成(環境1)

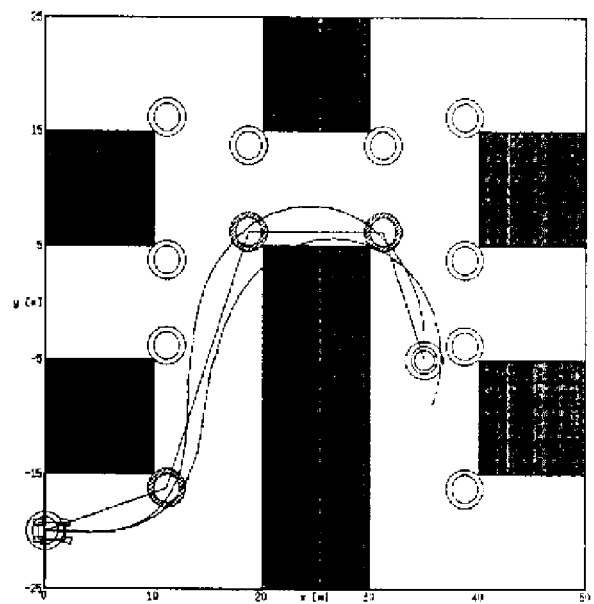


Fig. 17 方法2での経路生成(環境2)

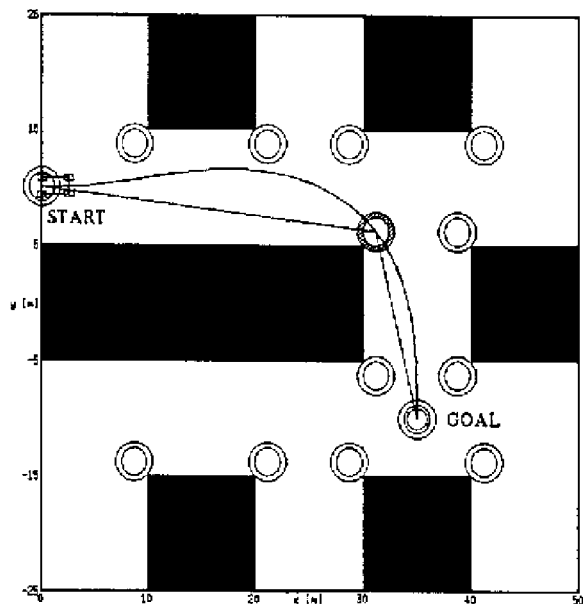


Fig. 15 方法3での経路生成(環境1)

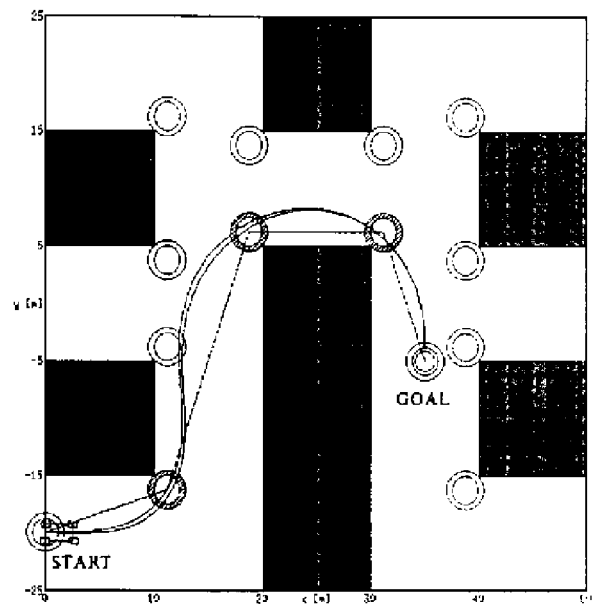


Fig. 18 方法3での経路生成(環境2)