ニューラルネットを用いた移動ロボット車の動的環境における走行制御法*

大友 照彦[†]・小平 実[‡]・大槻 恭士[†]・大内 隆夫[§]

Travel Control for Robot Vehicle Using Neural Networks in Dynamic Environment*

Teruhiko OHTOMO[†], Minoru KODAIRA[‡], Takashi OTSUKI[†] and Takao OHUCHI[§]

This paper proposes a more flexible and effective obstacle avoidance travel control system based on the switching of the thought in dynamic environment where many autonomous robot vehicles move. The proposed system plans various traveling paths by the Cascaded Neural Networks (CNN) which operates by a traffic rule or knowledge, so that robot vehicle can smoothly move the best suited route by evaluating the energy function value of those paths. To show the effectiveness of the control system, we carried out a computer simulation of the obstacle avoidance by using four vehicle models, and confirmed that each vehicle can avoid other vehicles without falling into the deadlock.

1. はじめに

近年, 様々な産業用途において自律移動ロボット車の需要が広がりつつあり, たとえば原子力発電所内における点検用ロボットや極限作業ロボットあるいはビルディング内外で働く知能ロボットなどの研究が盛んに行われている [1-5]. このような特殊な作業環境内における多数の移動ロボット車は, それぞれに与えられた作業を自律的に行ったり, 互いに協調して一つの作業を行うというタスクに従事することになる. 移動ロボット車単体が自律移動する場合には, 環境内の静的障害物のみを回避するアルゴリズムを用いればよいが, 多数の移動ロボット車がそれぞれ自律移動する場合においても衝突の危険性のない安全な走行制御システムを構築する必要がある. このためには, 各ロボット車は互いに相手を予測し衝突を回避する機能を持つ必要がある. したがって, ロボッ

ト車の衝突回避問題は, 衝突が発生する以前に各ロボット車がそれぞれ自己探索的に安全な走行経路を計画して走行制御する問題になる.

移動ロボット車の衝突回避問題は, これまで多数の研究がなされている. その中で強化学習等をベースにしたものとしては, Q-learning を用いたサッカーロボットの行動学習手法 [6], 経験強化型の学習システムによる基本行動獲得手法 [7], 複数の行動における漸進的な行動原理選択手法 [8], 下位レベルで障害物回避のための操作量を決定し上位レベルの学習オートマトンで適切な合成ルールを学習する手法 [9], 行動原理の異なるロボット群に漸進的学習を行うと群全体の作業効率を維持するための適応性が生じるという研究 [10], カーナビにおける経路選択を交通状況に適応させる進化的手法 [11], 赤外線センサーシステムを用いた階層的強化学習による適応的回避行動獲得手法 [12] などが提案されている. またルールをベースにした障害物回避や経路探索手法としては, 静電ポテンシャルと Sliding Mode を用いたもの [13], 移動障害物の動作を予測し状況によりポテンシャル法の最適パラメータを探索する手法 [14], 仮想的なインピーダンスを用いた動作計画手法 [15], ヒューリスティックな評価に基づいて衝突を予測し互いに速度ベクトルを修正しつつ回避する手法 [16], ファジィ評価に基づく相対関係の予測と次の行動をオペレータの操作履歴を基に意志決定し回避経路を得る手法 [17], 環境情報からのパター

* 原稿受付 2000 年 4 月 10 日

[†] 山形大学 工学部 Faculty of Engineering, Yamagata University; 4-3-16 Jonan, Yonezawa city Yamagata 992-8510, JAPAN

[‡] 米沢日本電気 (株) NEC Yonezawa, Ltd.; 2-6-80 Shimohanazawa, Yonezawa city, Yamagata 992, JAPAN

[§] 日本大学 工学部 Faculty of Engineering, Nihon University; 1 Nakagawara, Tamuracho, Kooriyama city, Fukushima 963, JAPAN

Key Words: robot vehicle, dynamic environment, cascaded neural network, traffic rules, path planning.

ン変換として経路計画を扱い作業環境の占有領域パターンを並列的に処理する手法[18]、掃引作業における協調掃引の動作計画手法[19]などの検討が行われている。

ところで、同じ動特性ならびに同じ回避アルゴリズムを持つ移動ロボット車が存在すると、制御問題の設定によっては制御不能のいわゆるデッドロック状態に陥り、その安全性に問題を生じることがある。たとえば、局所探索的アルゴリズムにより、自律移動を行うロボットでは、相手の動きを予測して自分の動きを変えるため、一度デッドロック状態に陥ると双方が同じ方向に回避軌道を決定し、最終的には衝突する危険性が生じる。したがって、多数の移動ロボット車が走行する動的な環境下においてもこのようなデッドロック状態が回避できる安全な走行制御システムを検討しておく必要がある。加藤ら[20]は交通ルールを適用して協調行動を実現する手法を提案し、井上ら[14]は目標に到着するまでの軌道予測と評価を行い行動パラメータを切り換える手法、山田[21]は熟考して行動するか即応行動するかを成功確率によって切り換える手法、藤沢ら[22]は2段階評価を行う行動監視機構が行動探索をうち切り探索途中の行動を出力することで予測誤差に対処する手法などを提案している。前者の交通ルールを用いた手法は規則が煩雑でしかも通行区分の拘束による走行の柔軟性に問題があり、後者の行動切り換え手法の多くは移動ロボット車の動特性を考慮しておらず、障害物を回避して目標点まで走行する軌道の滑らかさや最適性に関しても問題がある。

本論文では、各移動ロボット車が外界センサから他の移動障害物の位置と速度ベクトルを入手可能であり、また移動ロボット車の速さはいずれも同程度という仮定のもとに、多数の移動ロボット車が同一環境内を自律走行可能な柔軟で知的な制御法を提案する。本手法はニューラルネットワークを用いた移動ロボット車の障害物回避走行制御法[23]を拡張したもので、多様な知識や規則に基づいて軌道計画が可能のように、繰り返し構造型ニューラルネットワーク（以後、*CNN*と略す）を多重化する。走行制御器では、内蔵されたエネルギーオブザーバの評価に基づいて*CNN*が順次自動生成され、生成された各*CNN*は異なる知識あるいは規則に基づいてエネルギー関数が最小化されるように移動ロボット車の動特性を考慮した運動指令値がそれぞれ修正されるので、生成された*CNN*の数だけ多様な走行軌道が同時に計画される。この時走行制御システムは、それらのエネルギー関数の時間積分値からなる評価値に基づいて最適な一つの*CNN*を選択するので、知識に基づいた柔軟な走行制御が可能になるという考えに基づいている。最後に、本システムの有効性を四台の自律移動ロボット車の走行シミュレーション実験により明らかにする。

2. 移動ロボット車モデルと座標系変換

2.1 運動方程式

制御対象の移動ロボット車には、Fig.1に示すような一般的な前輪操舵後輪駆動の運動力学モデルを用いる。この時、車両の重心点は地面上にあり、傾斜のな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、前輪および後輪の特性が左右2輪の総合特性を代表するものとする。このような車両モデルをTable 1の記号を用いて直交座標系で導く。

ロボット車の運動の自由度は、 x および y 方向の運動(x, y)と重心点回りの回転運動(ψ)の3自由度とし、その操作量はステアリング角 θ および駆動力 D の二つである。Fig.1のような路面に固定された絶対座標系($o-xy$)を用いると、移動ロボット車の運動方程式は次式のように表される[23]。

$$m \frac{d^2 x}{dt^2} = -S_f \sin(\theta + \psi) - H_1 \cos(\theta + \psi) - S_r \sin \psi + (D - H_2) \cos \psi \quad (1)$$

$$m \frac{d^2 y}{dt^2} = S_f \cos(\theta + \psi) - H_1 \sin(\theta + \psi) + S_r \cos \psi + (D - H_2) \sin \psi \quad (2)$$

$$I \frac{d^2 \psi}{dt^2} = a(S_f \cos \theta - H_1 \sin \theta) - b S_r \quad (3)$$

Table 1 Symbols of the robot vehicle

記号	名称および定義
ψ	姿勢角
D	後輪駆動力 $D = \mu W_2$
μ	駆動係数 $\mu = D/W_2$, $0.3 \leq \mu \leq -0.7$
θ	前輪ステアリング角 $ \theta \leq 20$ [deg]
β	車体横滑り角 $D = \mu W_2$
β_f, β_r	前輪および後輪の横滑り角
S_f, S_r	前輪および後輪に作用するサイドフォース $S_f = C_f \beta_f W_1$, $S_r = C_r \beta_r W_2$
H_1, H_2	前輪および後輪に作用する転がり抵抗力 $H_1 = 0.02 W_1$, $H_2 = 0.02 W_2$
W_1, W_2	前輪および後輪の接地荷重 $W_1 = mga/(a+b)$, $W_2 = mgb/(a+b)$
C_f, C_r	前輪および後輪のコナリング係数 $C_f = C_{10}, C_r = C_{20}(1 - 0.32\mu(\mu + 0.5))$ $C_{10} = C_{20} = 5[1/\text{rad}]$
V	車速 $V = \sqrt{\dot{x}^2 + \dot{y}^2}$
ϕ	速度の方位角 $\phi = \arctan(\dot{y}/\dot{x})$
m	車両全体の質量 $110[\text{kgfs}^2/\text{m}]$
g	重力加速度 $9.8[\text{m/s}^2]$
I	重心点を通る鉛直軸回りの慣性モーメント $172[\text{kgfms}^2]$
a, b	前輪および後輪接地軸～重心距離 $1.25[\text{m}]$

Fig. 1 Model of a robot vehicle

また、前後輪の横滑り角 β_f , β_r は Fig. 1 から次のように求められる。

$$\beta_f = \theta + \arctan\left(\frac{\dot{x}\sin\psi - \dot{y}\cos\psi - a\dot{\psi}}{\dot{x}\cos\psi + \dot{y}\sin\psi}\right) \quad (4)$$

$$\beta_r = \arctan\left(\frac{\dot{x}\sin\psi - \dot{y}\cos\psi + b\dot{\psi}}{\dot{x}\cos\psi + \dot{y}\sin\psi}\right) \quad (5)$$

以上により、移動ロボット車モデルは、

$$\text{状態 } \mathbf{Z} = (x, y, \psi, \dot{x}, \dot{y}, \dot{\psi})^T \quad (6)$$

$$\text{入力 } \mathbf{U} = (\mu, \theta)^T \quad (7)$$

$$\text{状態方程式 } \frac{d\mathbf{Z}}{dt} = \mathbf{F}(\mathbf{Z}, \mathbf{U}) \quad (8)$$

と導出される。ここで、駆動係数 μ は自動車のアクセル量、ステアリング角 θ はハンドルの切れ角に対応する。

2.2 座標系変換

制御問題を考えるうえで基準となる座標系は、Fig. 1 に示すような路面に固定された絶対座標系 ($o-xy$) である。また、運動を検討しやすいように移動ロボット車の重心点を基準にした動座標系 ($o^r-x^ry^r$) も考える。

時刻 t における移動ロボット車の姿勢角を含めた絶対座標系での位置ベクトルを $\mathbf{X}[t] = (x[t], y[t], \psi[t])^T$ とすると、絶対座標系の任意の位置ベクトル $\mathbf{X} = (x, y, \psi)^T$ と動座標系の任意の位置ベクトル $\mathbf{X}^r = (x^r, y^r, \psi^r)^T$ の関係は次式で表される。

$$\mathbf{X} = \mathbf{X}[t] + \begin{bmatrix} \cos\psi[t] & -\sin\psi[t] & 0 \\ \sin\psi[t] & \cos\psi[t] & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{X}^r \quad (9)$$

絶対座標系と動座標系の座標変換は後述する同定器モジュール内の座標変換部で行われる。

3. ニューラルネットを用いた走行制御システム

3.1 制御システムの構成

宇野ら [25, 26] はニューラルネットを用いた運動制御法を検討し、人の手と繰り返し構造を持つニューラルネットを用いて制御したマニピュレータの運動との類似性を示した。この制御法を移動ロボット車に適用できるように拡張した走行制御システム [23] を Fig. 2 に示す。このシステムでは、目標時刻、目標状態、障害物の位置と速度およびそれらに付随する評価係数が走行制御器に入力されると、後述する制御器内部の繰り返し構造型ネットワークのエネルギー関数が最小化するように動作し、その結果運動指令 U が生成される。この時、制御器内部のニューラルネットは移動ロボット車の動特性を学習によって獲得しているため、与えられた制御問題を可能な限り満たす運動指令が得られることになる。

本制御器はニューラルネットによって学習された動特性を用いて運動指令および推定軌道を生成するが、付加的に走行状態をフィードバックすることにより推定軌道の補正を行うこともできる。

Fig. 2 Obstacle avoidance travel control system

3.2 同定器モジュールと繰り返し構造型ネットワーク

同定器モジュール ($I.M.$) の構成を Fig. 3 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 $\mathbf{Z}$ 、運動指令 $\mathbf{U}$ を入力すると、1 サンプル時間 (Δt) 後の状態の増分 $\Delta \mathbf{Z}$ 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がある。ここで、時刻 t は離散時刻 k を用いて $t = k\Delta t$ ($k = 0, 1, 2, \dots$) と表す。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層のユニットの入力を 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

Fig. 3 Structure of identification module

立すと考えられる。

$$\mathbf{u}^1[t] = \begin{bmatrix} \mathbf{Z}[k] \\ \mathbf{U}[k] \end{bmatrix} = \begin{bmatrix} \mathbf{X}[k] \\ \dot{\mathbf{X}}[k] \\ \mathbf{U}[k] \end{bmatrix} \quad (10)$$

$$\mathbf{u}^2[t] = \Delta \mathbf{Z}[k] = \begin{bmatrix} \Delta \mathbf{X}[k] \\ \Delta \dot{\mathbf{X}}[k] \end{bmatrix} \quad (11)$$

ただし、 $\Delta \mathbf{X}, \Delta \dot{\mathbf{X}}$ は $\mathbf{X}, \dot{\mathbf{X}}$ の増分である。

なお、同定器モジュール内部の各層 ($\hat{1} \sim \hat{5}$)、各ユニットの入出力や応答関数および各層間の結合係数、しきい値等については参考文献 [23] で詳述されているので省略するが、入力 ($\dot{x}^r, \dot{y}^r, \dot{\psi}^r, \mu, \theta$) と教師 ($\Delta x^r, \Delta y^r, \Delta \psi^r, \Delta \dot{x}^r, \Delta \dot{y}^r, \Delta \dot{\psi}^r$) の組からなる学習パターンを誤差逆伝播法 [24] を用いて学習することによって、移動ロボット車の動特性を学習することができる。

本同定器モジュールのパラメータは、移動ロボット車モデルの試走シミュレーションによって得られたサンプリング時間 $\Delta t = 0.1[\text{s}]$ における 9000 個の学習パターンを用いて、ニューラルネットを 5000 回学習したものをを用いている。

走行制御システムの離散時間における走行軌道計算の流れは、Fig. 4 に示すような同定器モジュールを縦続接続した繰り返し構造型ネットワーク (CNN) として表現できる。すなわち、同図は次式に示す (8) 式のオイラー解法を表している。

$$\begin{aligned} \mathbf{Z}[k+1] &= \mathbf{Z}[k] + \mathbf{F}(\mathbf{Z}[k], \mathbf{U}[k])\Delta t \\ &= \mathbf{Z}[k] + \Delta \mathbf{Z}[k] \end{aligned} \quad (12)$$

この CNN モデルでは、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、後述するネットワークのエネルギー関数により決定される誤差信号を逆伝播することにより運動指令 $\mathbf{U}$ 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、CNN を走行制御器として用いることにより制御対象の動特性を考慮した軌道計画を立てることができ、軌道計画問題と軌道制

Fig. 4 Cascaded network. I.M. represent the identification module shown in Fig.3

御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することができる。

3.3 障害物の経路推定と危険度関数

静的な障害物はその位置が変化しないが、動的な障害物はその位置および速度が時間的に変化する。そこで、このような動的障害物に対しても対応できるようにするために、動的障害物の移動経路を推定する必要がある。障害物の行動が既知の数学モデルで与えられる場合、現在の時刻を t_0 とすると、一般的な時刻 t における障害物の位置 $\mathbf{X}_o (= [x_o, y_o]^T)$ は次式のように表される。

$$\begin{aligned} \mathbf{X}_o(k) &= \mathbf{X}_o(t_0) + \frac{d\mathbf{X}_o(t_0)}{dt}(t-t_0) + \dots \\ &\quad + \frac{1}{n!} \frac{d^n \mathbf{X}_o(t_0)}{dt^n}(t-t_0)^n + \dots \end{aligned} \quad (13)$$

ただし、現実には障害物の行動は未知であるから、観測できる範囲のデータを用いて、その移動経路などを予測する必要がある。時刻 t を離散時刻 k を用いて $t = k\Delta t$ で表し、離散時刻 $k_0 (= 0, 1, \dots, k_f)$ の障害物の位置 $\mathbf{X}_o[k_0]$

および $\dot{X}_o[k_0]$ が観測できるものとする、観測時刻 k における障害物の位置は、(13) 式から得られる次の近似式で推定することができる。

$$X_o[k] = X_o[k_0] + \dot{X}_o[k_0](k - k_0)\Delta t \quad (14)$$

また、走行中に新たな $X[k_0]$, $\dot{X}[k_0]$ が得られれば、上式により推定軌道を更新する。

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. 5 のように仮定し、離散時刻 k における危険度関数 $P_{d,m}[k]$ を次式のように定義する。

Fig. 5 Vehicle and obstacle

$$P_{d,m}[k] = \begin{cases} \frac{r_{d,m}}{2} \left\{ \cos\left(\pi \frac{r_m[k]}{r_{d,m}}\right) + 1 \right\} & (r_m[k] < r_{d,m}) \\ 0 & (r_m[k] \geq r_{d,m}) \end{cases} \quad (15)$$

$$r_m[k] = \| \mathbf{X}_{o,m}[k] - [x[k], y[k]]^T \| \quad (16)$$

$$r_{d,m} = r_v + r_{o,m} + l_g \quad (17)$$

ただし、 $r_m[k]$ は時刻 k における障害物とロボット車間の距離、 $r_{d,m}$ は危険領域半径、 r_v はロボット車領域半径、 $r_{o,m}$ は障害物領域半径、 l_g は緩衝長を表す。 $P_{d,m}[k]$ 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となる関数である。

ここで、障害物回避に関する評価関数を E_o を定める。第 m 番目の障害物の離散時刻 k における評価関数 $E_{o,m}$ は (15) 式を用いると次のように定義することができる。

$$E_{o,m}[k] = \sum_{i=k_0}^k Q[k-i] P_{d,m}[i] \Delta t \quad (18)$$

$$Q[k] = q^k \quad (19)$$

ただし、 q ($0 < q < 1$) は危険度の減衰率であり、(19) 式は重み関数を表す。(18) 式は時刻 k 以前の危険度も考慮しているため、経路としてみたとき障害物回避能力が向上するものと期待できる。したがって、環境内に M 個の障害物が存在する場合、それらを回避するための評価

関数 E_o は次式で与えられる。

$$E_o = \sum_{m=1}^M \sum_{i=k_0}^k W_{o,m} E_{o,m}[k] \quad (20)$$

ここで、 $W_{o,m}$ は回避優先度係数であり、 $r_m[k]$ 等に基づいて実験的に決められる。

3.4 走行制御アルゴリズム

障害物を回避しながら目標時刻に目標状態を達成するのみならず、運動指令変化最小の観点から走行制御するために、CNN のエネルギー関数 L を次式のように定義する。

$$L = E_t + E_u + E_o \quad (21)$$

ここで、

$$E_t = \frac{1}{2} \sum_{k=k_0}^{k_f} \mathbf{e}_t[k]^T \mathbf{W}[k] \mathbf{e}_t[k] \quad (22)$$

$$E_u = \frac{1}{2} \sum_{k=k_0}^{k_f-1} \mathbf{e}_u[k]^T \boldsymbol{\lambda}_u \mathbf{e}_u[k] \Delta t \quad (23)$$

$$\mathbf{e}_t[k] = \mathbf{Z}^*[k] - \mathbf{Z}[k] \quad (24)$$

$$\mathbf{e}_u[k] = \mathbf{U}[k+1] - \mathbf{U}[k] \quad (25)$$

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 は危険度関数最小に関する評価関数である。 $\mathbf{Z}^*[k]$ は目標状態、 $\mathbf{W}[k] = \text{diag}[w_1[k], w_2[k], \dots, w_6[k]]$ 、 $\boldsymbol{\lambda}_u$ は評価係数を表す。なお、 k_0 は走行中に時間の経過と共にシフトする。

このとき、運動指令ユニット値 $\{U_i[k] \mid i=1, 2; k=k_0+1, \dots, k_f-1\}$ は次の修正式により最急降下方向に調節する。

$$\frac{dU_i[k]}{dt} = -\epsilon_U \frac{\partial L}{\partial U_i[k]} \quad (\epsilon_U: \text{修正係数}) \quad (26)$$

上式の修正を繰り返し行うことによって L に最小値を与える $U_i[k]$ が得られるが、それらは障害物を回避しながら目標時刻に目標状態を達成するのみならず、運動指令の変化も最小にするような滑らかな走行を計画するので、かつその軌道を忠実に走行制御する運動指令値になっている。

3.5 走行シミュレーション

ここでは、静的および動的な複数の障害物を考慮した走行制御シミュレーション [23] について述べる。走行制御のパラメータは下記のものを用い、神経回路の動作時間は $-5[s]$ から $7[s]$ まで、ロボット車の走行時間は $0[s]$ から $7[s]$ までとする。ただし、 $0[s]$ 以前の時間は走行前に CNN を用いて軌道を計画するための時間である。また、*印は無拘束を示す。

$$\mathbf{Z}^*[k] = (32.5 [m], 25 [m], 90 [\text{deg}], *, *, *)^T \quad k = 70$$

$$W[k] = \begin{cases} \text{diag}[0.01, 0.01, 1, 0, 0, 0] & k = 70 \\ 0 & \text{otherwise} \end{cases}$$

走行環境としては、Fig. 6(a) のような T 字路を左折する状況を想定し、走行路の外側の静的障害物は円群の集合体で表している。この走行路において、移動ロボット車の前を移動体が低速 10[km/h] で走行しており、対向車線をもう 1 台の移動体が中速 20[km/h] で走行してくる状況を設定する。この状況で移動ロボット車 (内塗り円で表す) が初速度 30[km/h] で 7[s] 後に S 地点から最終目標地点 G (太線小円) に到達するような指令を与え、障害物回避走行制御を行う。(a) 図には移動ロボット車を含めた三つの移動体を 0.5[s] ごとに細線円で示し、最接近した状況を太線円で示す。同図により、移動ロボット車は静的および動的障害物を巧みに避けながら走行路を左折し目標地点に到達している。Fig. 6(b) はこのときの車速 (*Speed*)、駆動係数 (*D.F.C.*)、ステアリング角 (*S.A.*) の時間変化を示している。いずれの走行特性も滑らかに変化しており、人間の運転と符合した特性を得ている。

(a) Traveling path

(b) Control characteristics

Fig. 6 An example of obstacle avoidance

4. 思考切り換え形走行制御システム

多数の移動ロボット車が同じ環境内を走行することを考える。この場合、同じ動特性ならびに同じ回避アルゴリズムを持つ移動ロボット車が多数存在すると、制御問題の設定によっては後述するような制御不能のいわゆるデッドロック状態に陥る可能性が考えられ、安全性に問題が生じる。そこで、多数の移動ロボットが存在する動的な環境でもデッドロック状態を回避できる制御システムとして、簡単な交通規則をそれぞれ埋め込んだ多数の *CNN* を用いた走行制御システムを提案する。

4.1 走行制御システムの構成

Fig. 7 は 1 台の移動ロボット車における思考切り換え形走行制御システムの構成を示す。本システムは、前述した *CNN* を基本 *CNN* および派生 *CNN* として内部に多数保有することが可能で、それらの *CNN* はそれぞれ各々の知識あるいは規則に基づいて軌道計画を行う走行制御器として動作させる。すなわち、各 *CNN* の前向き計算 (軌道計算) は全く同じであるが、誤差逆伝播による後ろ向き計算 (最適制御入力) の計算では各 *CNN* 独自のエネルギー関数を最小化する手法でその運動指令値が修正される。この時、各 *CNN* のエネルギー関数最小化を高次の知識や規則に基づいて実現できれば、走行制御問題をそれぞれ異なる観点から解いたことになる。たとえば、個々の *CNN* にそれぞれ異なるパラメータあるいは思考に必要な障害物回避ルールを設定すると、同一の制御問題設定に対して相異なる有効な軌道計画と運動指令の生成を並列に行うことができる。したがって、各移動ロボット車は種々の知識や規則に基づいて同じ制御問題をロボット車それぞれの立場で解き、生成された複数の軌道から最適な *CNN* を選択して走行することになる。

Fig. 7 Obstacle avoidance control system using multiple cascaded neural networks

4.2 エネルギーオブザーバ

Fig. 7 のエネルギーオブザーバは、各 *CNN* のエネルギー関数の値を観測し、後述するアルゴリズムによってトリガー信号 B_0 を出力する。 Δt 秒後の E_0 の増分値

ΔE_o が推定増分値 ΔE_o^* よりも大きい場合には、障害物が予想以上に近づいていることを表すので、このような状態が長く続けば衝突する危険性が非常に高くなると考えられる。そこで、エネルギーオブザーバは移動ロボット車が将来危険な状態に陥ることを未然に防止するために、以下のアルゴリズムを動作させてトリガー信号 B_0 をネットワークマネージャーに対して出力する。

$$\Delta E_o^* = \sum_{k=k_0}^{k_f} \left\{ \left(\frac{\partial E_o}{\partial x[k]} \right) \Delta x[k] + \left(\frac{\partial E_o}{\partial y[k]} \right) \Delta y[k] \right\} \quad (27)$$

$$e_o = \frac{\Delta E_o}{\Delta t} - \frac{\Delta E_o^*}{\Delta t} \quad (28)$$

$$B_0 = \begin{cases} 1 & (C_0 > C_0^* \text{ のとき}) \\ 0 & (C_0 \leq C_0^* \text{ のとき}) \end{cases} \quad (29)$$

ただし、

$$C_0 = \begin{cases} \int e_o dt & (B_0 = 0 \text{ のとき}) \\ 0 & (B_0 = 1 \text{ のとき}) \end{cases} \quad (30)$$

ここで、 Δx 、 Δy はそれぞれ移動ロボット車の位置 x 、 y の増分値であり、 C_o は貯蓄量、 C_o^* は最大許容量を表す。これらの各変数は各 CNN ごとに設定され、それぞれ独立して動作する。

4.3 ネットワークマネージャー

Fig. 7 に示すネットワークマネージャーは、構築、消滅、活性化、選択の四つの機能を管理する。

オブザーバから受け取ったトリガー信号 B_0 が 1 であれば、「構築」機能により基本 CNN とは異なる知識あるいは規則で後ろ向き計算を行う CNN を新たに派生させる。このとき、障害物に関する評価関数 E_o が 0 になるか C_o が非常に小さくなる場合には、 E_o の大きい方の CNN は不必要であると見なされ、「消滅」機能により消去される。また、すべての CNN の間で、ある派生 CNN のエネルギー関数が最も小さくなれば、「選択」機能によりその派生 CNN で計算された制御入力を選ばれ、走行することになる。

また、「構築」あるいは「選択」機能により運動指令や軌道計画あるいは障害物回避ルール等の各種パラメータの設定中は、一時 CNN が思考を中断するため、「活性化」機能により思考を始めるように各 CNN に対して命令する。このように、提案する走行制御システムでは、ネットワークマネージャーが各 CNN の走行制御問題を管理し、CNN がそれぞれ並列に思考を行うための支援を行う。

4.4 規則に基づく強制修正

一般に、同一環境内を多数の移動体（自動車、船舶等）がそれぞれ目的を持って移動するとき、安全で効率の良い移動を行うために交通規則や船舶運航規則などが定め

られ、スムーズな移動を可能にしている。そこで、本走行制御システムにおいても、効率よく衝突を回避するために、制御ルールによる強制修正の手法を提案する。

Table 2 には二つの障害物回避ルールを示す。基本 CNN (B-CNN) では障害物回避規則としてこれまでも用いてきた最急降下則による修正則を用い、経路上の点 $(x[k], y[k])$ での障害物回避に関する修正量を、 $-\partial E_o / \partial x[k]$ 、 $-\partial E_o / \partial y[k]$ で与える。一方、派生 CNN (D-CNN) では危険な障害物の右側を走行する修正則となっており、障害物回避に関する修正量は $-\partial E_o \sin(\psi[k]) / \partial x[k]$ 、 $-\partial E_o \cos(\psi[k]) / \partial y[k]$ で与えることになる。

Table 2 Obstacle avoidance rules

	Correcting value
B-CNN	$-\frac{\partial E_o[k]}{\partial x[k]}, -\frac{\partial E_o[k]}{\partial y[k]}$
D-CNN	$-\frac{\partial E_o[k] \sin(\psi[k])}{\partial x[k]}, -\frac{\partial E_o[k] \cos(\psi[k])}{\partial y[k]}$

5. 動的環境における走行制御シミュレーション

本システムの有効性を示すために、最急降下則で動作する基本 CNN と危険障害物右側走行則で動作する派生 CNN の二つの CNN からなる走行制御システムを構築し、有効性を検証する。二つの CNN 内の同定器パラメータには、3.2 で示した 9,000 個の試走パターンの学習から得られたものを用いている。

5.1 基本 CNN のみを用いた場合

Fig. 8 は、従来の単独 CNN で構成される制御器で走行制御を行った場合の回避経路と経路修正量を示す。走行条件は、4 台の移動ロボット車 V_1, V_2, V_3, V_4 がそれぞれスタート点 S_i から目標点 G_i ($i=1, \dots, 4$) まで、 V_1, V_2, V_4 は 7 秒、 V_3 は 8 秒後に到達するように設定した。Fig. 8 (a) は走行開始から 1 秒後の状態、同図 (b) は 2 秒後、同図 (c) は 2.5 秒後の状態を表している。図中の太い円は、各時刻における移動ロボット車の現在位置を表し、波線からなる円群は各移動ロボット車の障害物回避システムで推定した予定回避軌道を表している。また、矢印付きの実線は CNN が推定した回避軌道を表し、回避軌道から出ている矢印のない実線の長さは基本 CNN が修正した軌道修正量を、方向は軌道修正方向を表している。

4 台の移動ロボット車は、学習した動特性および用いた回避アルゴリズムが全く同じであり、移動ロボット車 V_2 および V_4 は出発点と目標点が互いに入れ替わった設定になっている。移動ロボット車 V_2, V_4 は、上方に移動ロボット車 V_1, V_3 が存在するために、はじめは両者とも下方に回避軌道を決定している。しかし、局所探索的アル

(a) after 1 [s]

(b) after 2 [s]

(c) after 2.5 [s]

Fig. 8 Estimated paths and correcting values of robot vehicles with basic CNN

ゴリズムによって自律移動を行うロボット車では相手の動きを予測して自分の動きを変えるため、Fig. 8 (b),(c)のように一度デッドロック状態に陥ると双方が同じ方向に回避軌道を決定し、最終的には衝突することになる。

5.2 思考切り換え形走行制御システム

Fig. 9は走行環境および走行位置の設定などの走行条件はFig. 8と全く同じにして、提案する思考切り換え形システムによる走行制御法を用いた場合の回避軌道と軌道修正量を、移動ロボット車 V_2, V_4 のみに着目して示したものである。出発点から1.5秒程度までは基本CNNにより実線で示される軌道が計画され、走行している。しかし、計画された軌道の危険度が増し最大許容量 C_0^* を越えることにより派生CNNが動作して、二台のロボット車 V_2 および V_4 はそれぞれ波線で示される二つの新たな回避軌道が生成される。実線で示された軌道は、

最急降下則に基づいて基本CNNを動作させた場合の回避軌道であり、波線で示された軌道は、危険障害物右側走行則に基づいて派生CNNを動作させた場合の回避軌道である。結果的に、ロボット車 V_2 は基本CNNによって計画された軌道をそのまま走行し続けるが、ロボット車 V_4 の方は派生CNNの計画した軌道の方が評価基準を満たす好ましい軌道であるために、現在の走行軌道に変えて派生CNNの計画した動的障害物であるロボット車 V_2 の右側を通過する経路を選択することになる。

Fig. 9 Planned avoidance paths of robot vehicles with double CNN

Fig. 10は、前述の二つの走行実験におけるスタートからゴールまでの各ロボットの走行軌跡を示している。図中、4台のロボット車の動きをそれぞれ0.5[s]ごとにハッチングを施した円で表している。Fig. 10(a)は基本CNNのみを用いて制御された場合であり、 V_2 と V_4 の移動ロボット車同士は障害物回避軌道がデッドロック状態に陥り、(25[m], -10[m])付近の黒丸で示す地点で避けきれずに衝突している。しかし、交通ルールを組み込んだ思考切り換えによる走行制御システムでは、思考中の二つの軌道から最も条件を満たす軌道が選択できるため、Fig. 10 (b)に示されるように、4台の移動ロボット車は共に衝突することなしに各目標値点まで到達し、しかも滑らかな走行軌道を走行していることが分かる。

一般に、各ロボット車の動特性や制御パラメータが全く同じということではなく、したがってデッドロック状態に陥ることは非常にまれなことである。しかし、ここで示したような非常に極端な設定をしても、危険障害物右側走行規則を付加するだけで通行不能の状態を回避することができ、さらに他の規則、たとえば危険がさらに迫ればスピードを一時緩める等の動作をするCNNを組み込むこと等によりさらに柔軟な走行制御が可能になるものと考えられる。ただし、移動障害物の速度によっては危険性を検知するための最大許容量 C_0^* が影響し、設定によっては回避軌道の生成動作が遅れて回避不能になることも考えられる。これには各CNNによる軌道生成のための計算時間や制御周期などとも密接な関連があり、実機による最適パラメータの設定や適用限界について今後検討する必要がある。

ところで、走行制御システムにおける各CNNの計算

(a) Using a basic CNN

(b) Using two CNNs with different avoidance rule

Fig. 10 Traveling paths of robot vehicles

は、並列処理が可能であり、思考の複雑さに対して思考時間が変わらないという特徴を有する。したがって、時々刻々変化する制御問題を解決しなければならない移動ロボット車の制御には特に有効になるものと考えられる。

6. おわりに

本論文では、多数の移動ロボット車が同一の環境内を自律的に走行する場合の制御問題に柔軟に対処するために、簡単な障害物回避ルールを導入した思考切り換え形走行制御システムを提案した。本システムは、システム内のエネルギーオブザーバが障害物の接近による危険性を検知し、種々の障害物回避ルールに基づく多様な走行軌道を計画する繰り返し構造形ネットワーク (CNN) を順次派生させ、生成された複数の計画軌道の中から評価基準に基づいて最良の軌道を自動的に選択する方式である。最急降下則と危険障害物右側走行則を用いて4台の移動ロボット車が同一環境内を自律移動する走行シミュレーションを行った結果、最急降下則に基づく基本CNNからなる走行制御器のみを用いるとデッドロック状態に陥り衝突するような条件でも、デッドロック状態に陥ることなしに安全で滑らかな軌道を計画し走行することが確かめられた。

今後、回避ルールの増加によるより複雑な制御問題の解決、思考時間の短縮および実機による適用限界の検討などを行う予定である。

参考文献

- [1] 竹原, 中山, 前田, 八田: 原子力点検ロボットに用いる半月型クローラ車の開発; 日本ロボット学会誌, Vol. 2, No. 3, pp. 62-67 (1984)
- [2] 沢崎, 石井, 内山: 月面移動探査機の自律走行方式と経路計画アルゴリズム; 日本機械学会第7回知能移動ロボットシンポジウム, No. 940, A211, pp. 136-141 (1994)
- [3] 塚越, 広瀬: 間歇クローラ歩容の提案とその生成原理; 日本ロボット学会誌, Vol. 17, No. 2, pp. 301-309 (1999)
- [4] 天野, 柿鞍, 芦川, 杉村, 岡本: 壁面仕上材剥離ロボットの開発—剥離作業の自動化に関する研究—; 日本機械学会第7回知能移動ロボットシンポジウム, No. 940, A201, pp. 85-88 (1994)
- [5] 倉林, 古賀, 新井, 太田: 掃引作業における局所的経路再計画による未知障害物回避; 日本ロボット学会誌, Vol. 17, No. 7, pp. 966-973 (1999)
- [6] 浅田, 野田, 俵積田, 細田: 視覚に基づく強化学習におけるロボットの行動獲得; 日本ロボット学会誌, Vol. 13, No. 1, pp. 68-74 (1995)
- [7] 中村, 黒山, 山田, 大倉, 上田: 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. 371-379 (1999)
- [8] 太田, 横川, 新井: 複数自律ロボット系における漸進的戦略形成に関する研究; 日本ロボット学会誌, Vol. 14, No. 3, pp. 379-385 (1996)
- [9] 青木, 岡, 鈴木, 大熊, 早川: 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)
- [10] 倉林, 太田, 新井, 野口: 不均質な行動原理を有する移動ロボット群の協調作業; 計測自動制御学会論文集, Vol. 35, No. 11, pp. 1377-1384 (1999)
- [11] 星野, 湊: 進化的手法による自動車の経路選択戦略の適応; 計測自動制御学会論文集, Vol. 34, No. 10, pp. 1495-1500 (1998)
- [12] 新井, 藤井, 浅間, 嘉悦, 遠藤: 階層的強化学習に基づく複数移動ロボット間の衝突回避; 日本機械学会ロボティクス・メカトロニクス講演会, 2A12-1 (1998)
- [13] 橋本, 国井, 原島, V. I. Draknov: 静電ポテンシャル法及びSliding Modeを用いた多自由度マニピレータの障害物回避制御; 日本ロボット学会誌, Vol. 11, No. 8, pp. 1220-1228 (1993)
- [14] 井上, 井上, 大川: 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回避行動計画; 日本ロボット学会誌, Vol. 15, No. 2, pp. 249-260 (1997)
- [15] 新井, 太田: 仮想的なインピーダンスを用いた複数移動ロボット系の動作計画; 日本ロボット学会誌, Vol. 11, No. 7, pp. 1039-1046 (1993)
- [16] 齊藤, 津村: 移動ロボット車間における衝突回避; システム制御情報学会論文誌, Vol. 3, No. 1, pp. 1-7 (1990)
- [17] 永田, 土谷: 移動ロボット車の障害物回避に関する知的制御; 日本ロボット学会誌, Vol. 11, No. 8, pp. 1203-1211 (1993)
- [18] 内田, 井出, 横山: 動的環境における障害物領域予測法: 電

- 気学会論文誌 C 分冊, Vol. 117-C, No. 2, pp. 198–204 (1997)
- [19] 倉林, 太田, 新井, 吉田: 掃引作業における移動ロボット群の動作計画; 日本ロボット学会誌, Vol. 16, No. 2, pp. 181–188 (1998)
- [20] 加藤, 西山, 武野: 交通ルール適用による複数の移動ロボットの協調行動; 日本ロボット学会誌, Vol. 12, No. 2, pp. 291–298 (1994)
- [21] 山田: 動的環境における成功確率を用いた熟考の制御; 人工知能学会誌, Vol. 11, No. 4, pp. 645–652 (1996)
- [22] 藤沢, 早川, 青木, 鈴木, 大熊: 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)
- [23] 小平, 大友, 田中, 大内: 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走行制御; 信学論 (D-II), Vol. J79-D-II, No. 1, pp. 91–100 (1996)
- [24] D. E. Rumelhart, J. L. McClelland and PDP Research Group: *Parallel Distributed Processing*, MIT Press, Vol. 1, pp. 318–362 (1986)
- [25] Y. Uno, M. Kawato and R. Suzuki: Formation and control of trajectory in human multijoint arm movement torque-change model; *Biological Cybernetics*, Vol. 61, pp. 89–101 (1989)
- [26] M. Kawato, Y. Maeda, Y. Uno and R. Suzuki: Trajectory formation of arm movement by cascaded neural network model based on minimum torque-change criterion; *Biological Cybernetics*, Vol. 62, pp. 275–288 (1990)