

ニューラルネットワークを用いた移動ロボット車の走行制御
Traveling control of mobile robot vehicle using the neural network

○佐藤喜一郎,大友照彦,大槻恭士,石谷幹夫

○Kiichiro Sato, Teruhiko Ohtomo, Takashi Otsuki, Mikio Ishitani

山形大学工学部

Faculty of Engineering Ymagata University

キーワード：ニューラルネットワーク(the neural network)、移動ロボット車(mobile robot vehicle)、走行制御(traveling control)

連絡先 〒992-8510 山形県米沢市城南 4-3-16 山形大学工学部 共通講座 大友研究室

1. はじめに

移動ロボット車の走行制御や経路計画、障害物回避の研究が数多く行われている。しかし動特性を考慮した場合、移動ロボット車の数式モデル(運動方程式、状態方程式)の非線形性が強く、解くことが困難なモデルとなっている。そのため、非線形性が弱くなる低速度旋回や速度一定などの拘束条件を設けた研究が多い。

われわれは、運動方程式や状態方程式を解くということを経ずに、同定器モジュール内のニューラルネットワークを用いて移動ロボット車の動特性を学習させ、その同定器モジュールを用いてロボット車の軌道計画と走行制御を行う手法の有効性をシミュレーション上で示した。

本稿では、本手法の有効性を実機で検証するため、実走行より取得した動特性をニューラルネット学習し、走行軌道の生成を行ったので報告する。

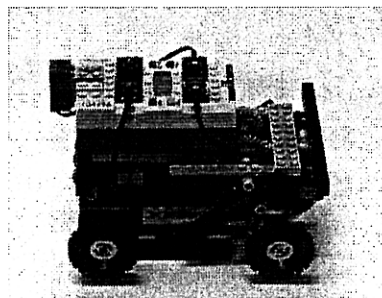


Fig.1 移動ロボット車

2. ロボット車

2.1 実機の概要

移動ロボット車は LEGO Mindstorms Robotic Invention System を使った Fig.1 のような前輪操舵後輪駆動車とする。全高 11.0cm、全幅 13.5cm、全長 17.0cm となっており前輪操舵部分に操舵量測定のために角度センサがあり、また光センサを搭載し明るさを測定できる。

運動指令としては前進、後進それぞれ 4 段階の駆動力、ブレーキ、左右 1 段階のステアリング方向を 0.1 秒毎に指令できる。さらに赤外線

を使って運動指令リストをPCから受信後、ストロボの閃光によって走行を開始し走行終了後に操舵量と運動指令をPCに送信できる。

2.2 数式モデル

このロボット車のモデルを Fig.2 に示す。基準となるのは絶対座標系($o-xy$)であり、ロボット車を中心を考えるときは相対的な動座標系($o_r-x_r y_r$)を用いる。移動ロボット車の運動の自由度は、 x 軸方向の運動、 y 軸方向の運動、重心回りの回転運動の3自由度であり、その操作量はステアリング指示 θ 、前輪駆動力 μ である。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のようになる。

$$m \frac{d^2 x}{dt^2} = F_x \quad (1)$$

$$m \frac{d^2 y}{dt^2} = F_y \quad (2)$$

$$I \frac{d^2 \phi}{dt^2} = M \quad (3)$$

m : ロボット車の全質量

F_x, F_y : x 、 y 軸方向の外力の総和

I : 重心点における鉛直回りの慣性モーメント

M : 重心点における鉛直回りのモーメントの総和

以上により、移動ロボット車のモデルは

$$\text{状態 } Z = (x, y, \dot{x}, \dot{y}, \phi, \omega)^T \quad (4)$$

$$\text{入力 } U = (\mu, \theta)^T \quad (5)$$

$$\text{状態方程式 } \frac{dZ}{dt} = F(Z, U) \quad (6)$$

(x, y) : 移動ロボット車の現在位置

μ : 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)

ϕ : 重心周りの回転角

θ : ステアリング指令(ロボット車のハンドルに相当するもの)

ω : ステアリング状態(ロボット車の角度センサより検出される、ステアリング状態)

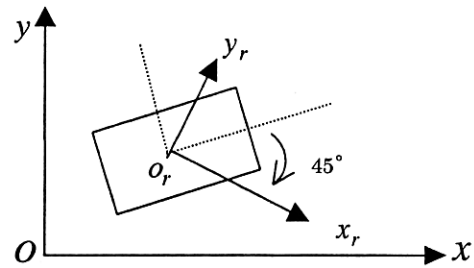


Fig.2 ロボット車モデル

と表わせる。ここで式(6)のベクトル値関数 F の非線形性が強いために、制御することが困難なモデルとなる。

3. 走行制御器

3.1 同定器モジュール (I.M.)

同定器モジュールはロボット車の動特性を同定するものであり、階層型の局所結合型ニューラルネットワークで構成されている。この同定器にロボット車の状態量と運動指令を入力すると1サンプリング後の状態量の増分が出力される。実験では状態量、運動指令、状態量の増分は求めることができ既知なのでニューラルネットの学習は教師あり学習となり、その学習法はバックプロパゲーション法を用いる。

3.2 繰返し構造型ネットワーク (CNN)

走行制御と走行経路は同定器モジュール (I.M.) を時間軸方向に多数個結合した Fig.3 に示すような繰返し構造型ネットワークを用いることになる。これは、(6)式の状態方程式のオイラー解法であり、式で表すと次式になる。ここで Δt はサンプリング時間でここでは $0.1[s]$ とする。

$$\begin{aligned} Z(k+1) &= Z(k) + F(Z(k), U(k))\Delta t \\ &= Z[k] + \Delta Z[k] \end{aligned} \quad (7)$$

この走行制御器では順方向の計算により移動ロボットの走行状態と走行軌道を推定でき、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 U を生成することができる。すなわち、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、軌道計画の問題と走行制御の問題を同時に解くことが

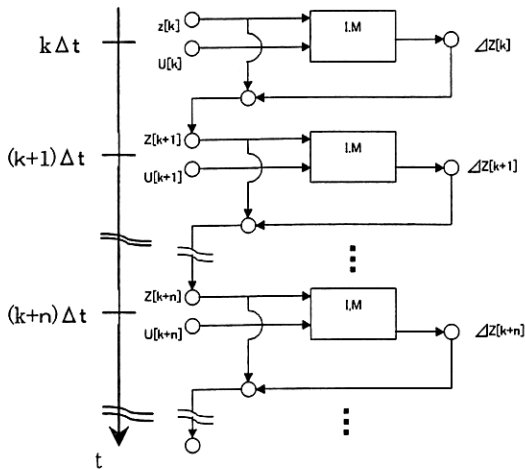


Fig.3 繰り返し構造型ネットワーク

できる。

4. 走行軌道の測定

4.1 測定方法²⁾

移動ロボット車の走行時の動特性を得るために、Fig.4 のような測定装置を組み立てて移動ロボット車の走行をビデオカメラで撮影する。撮影する走行は右旋回や直進といったものとそれらを組み合わせた右旋回後左旋回といったものを、1種類について5回ずつ走行させたものである。その後、撮影した走行画像から相関演算により 0.1[s]ごとの移動ロボット車の位置、姿勢角などの情報を読み取りニューラルネットの同定学習に必要なデータを作成する。

4.2 軌道のスムージング

実際に撮影した画像から走行の軌道を求めたのが Fig.5(a)となる。本来ならば滑らかな軌道が、解像度の問題により凹凸になっている。そして、この軌道の走行データをそのまま学習データとして用いた場合、同定能力が低下し本来の走行の動特性を同定することができなくなってしまう可能性がある。そこで、撮影した軌道を最小二乗法で近似し、滑らかな軌道(スムージング)にした後(同図(b))、それを学習データとして用いることとした。

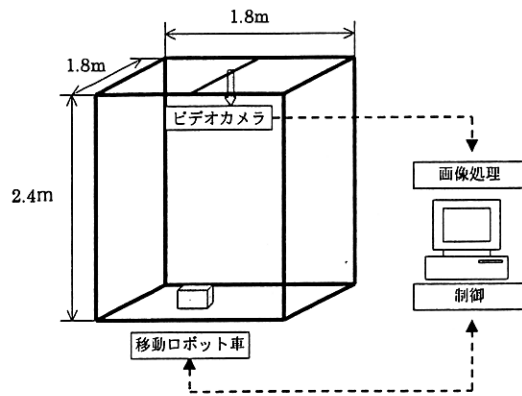
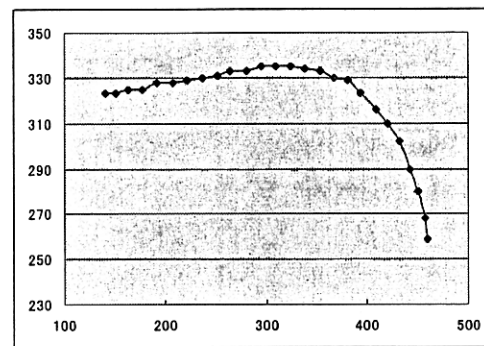
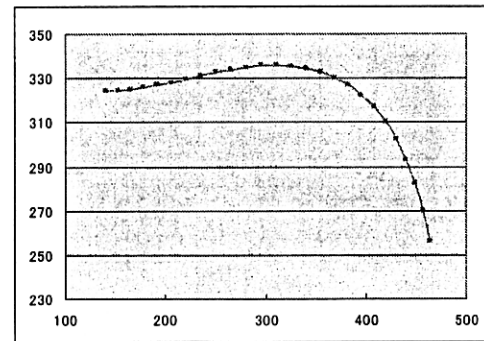


Fig.4 動特性測定装置



(a) 実際の軌道



(b) スムージング軌道

Fig.5 軌道のスムージング

5. 動特性の学習

5.1 3層型ネットを用いた場合

用いた3層型ニューラルネットの構造を Fig.6 に示す。出力層ユニットごとに中間層を配置した局所結合方式であり、入力層と中間層は全結合である。入力層の入力は状態量 $(\dot{x}_r, \dot{y}_r, \dot{\phi}, \omega_t)$ と運動指令 (μ, θ) の計6つ、出力層は1サンプリング後の状態量の増分 $(\Delta\dot{x}_r, \Delta\dot{y}_r, \Delta\dot{\phi}, \omega_{t+1})$ の4つである。中間層ユニット数と学習率は予備実験によって決定し、中間層ユニット数は、 ω_{t+1} に接続されるものは10、

それ以外は 1 出力ユニット当たり 30、学習率 0.1、学習回数 5 万回とした。

Fig.7 は縦軸に教師データと横軸に出力値をとった同定評価図であり、プロット点が $y = x$ の直線上に密集しているほど同定能力が高いことを示す。この 3 層型ネットの場合、密集度が低く同定能力が高いとは言えない。

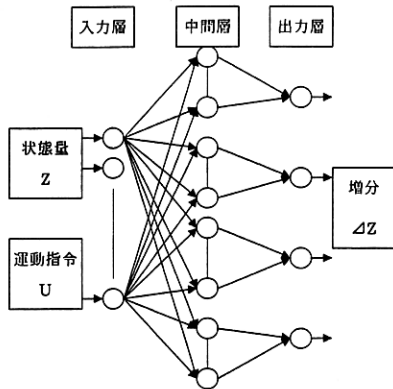


Fig.6 3層型ニューラルネットの構造

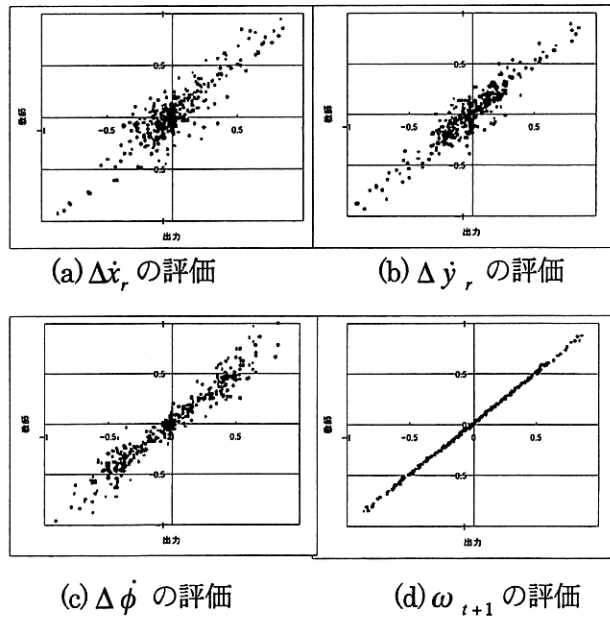


Fig.7 3層型ニューラルネットの同定評価図

5.2 4層型ネットを用いた場合

3層型ネットの同定能力が低かったため、ニューラルネットの構造を Fig.8 の 4 層構造とする。出力層のユニットごとに中間層 1、2 のユニットを配置した局所結合方式であり、入力層ユニットと中間層 1 のユニットは全結合である。また、状態量に加速度を加え、状態量

$(\dot{x}_r, \dot{y}_r, \dot{\phi}, \omega_t, \ddot{x}_r, \ddot{y}_r, \ddot{\phi})$ と運動指令 (μ, θ) の計 9 つを入力し、1 サンプル後の状態量の増分 $(\Delta\dot{x}_r, \Delta\dot{y}_r, \Delta\dot{\phi}, \Delta\omega_{t+1})$ の 7 つを出力する。

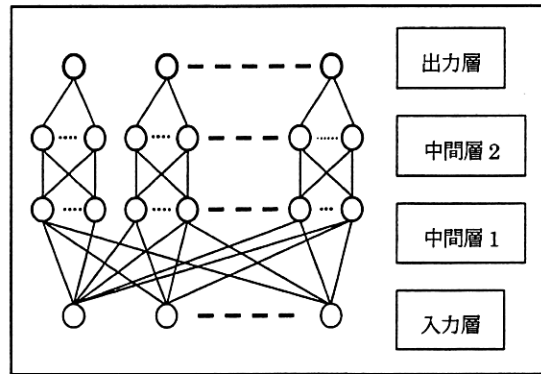


Fig.8 4層型ニューラルネットの構造

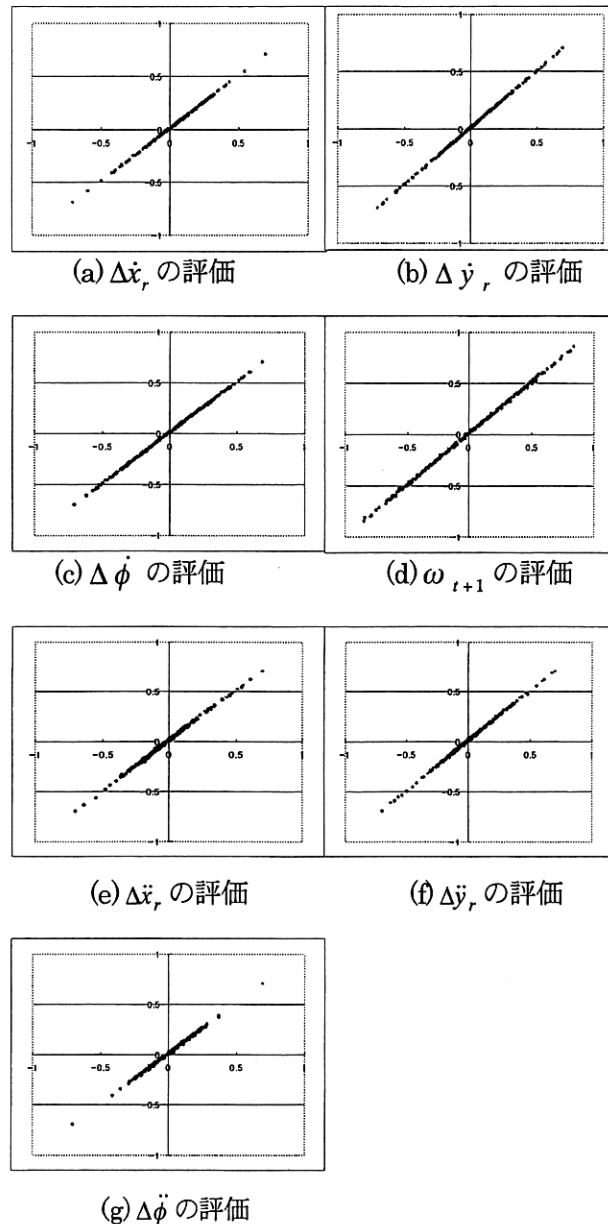


Fig.9 4層型ニューラルネットの同定評価図

中間層 1,2 のユニット数と学習率は予備実験によって決定し、出力ユニット 1 つ当たりの中間層 1,2 のユニットはそれぞれ 20、学習率と慣性率は 0.2、学習回数は Δx^r と Δy^r が 2 万回、他は 5 万回とした。Fig.9 は同定評価図であり、プロット点が $y = x$ の直線上にあり同定能力が高いことがわかる。

Fig.10,11 に、実走行における運動指令を同定器モジュールに入力して再現した軌道を、実走軌道と共に示す。

再現した軌道と実走行軌道がほぼ一致しておりこの図からも同定能力が高いことがわかる。今後、繰返し構造型ネットワークによる軌道生成にはこの 4 層型ネットネットワークを用いる。

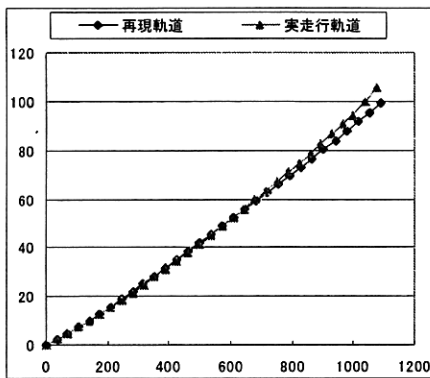


Fig.10 直進軌道

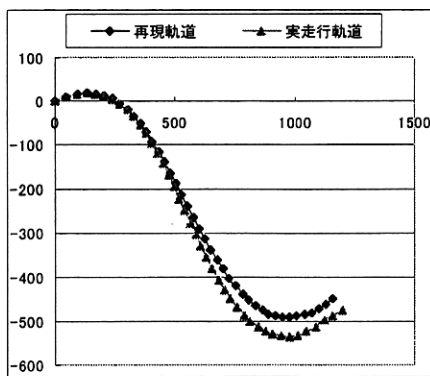


Fig.11 S字軌道

6. CNN による軌道生成

6.1 中継点を設定しない場合

学習した 4 層型ニューラルネットを用い、CNN を構成し走行軌道を生成する。ここでは

目標位置に到達するため、目標位置と CNN の順方向計算で得られた軌道の終点位置との差をエネルギー関数とし、これを逆伝播して運動指令（ステアリング方向： θ ）の修正を行う。また、運動指令は離散値のために修正後に離散化を行う。CNN での軌道生成を行う際は特に記述しない限り初期運動指令は 0 を与える。

実走行データの終点を目標位置に設定し、繰返し計算の結果得られた軌道を Fig.12,13 に示す。両図とも走行時間は 2.5 秒で、Fig.12 の場合は繰返し 7 回、Fig.13 の場合繰返し 4 回で、実走行と非常に近い軌道が生成された。

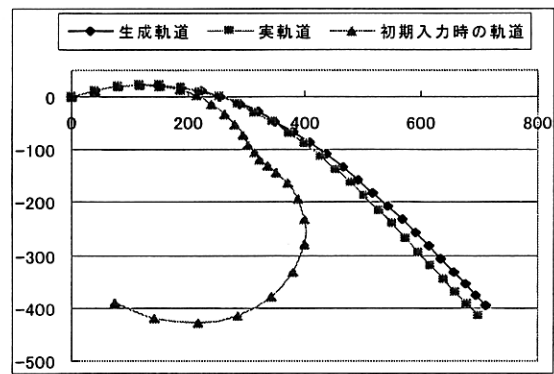


Fig.12 軌道生成の結果：右へ曲る軌道

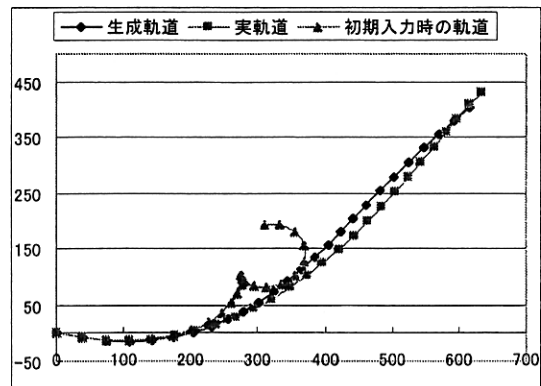


Fig.13 軌道生成の結果：左へ曲る軌道

6.2 中継点を経由する軌道

スタート地点から中継点を通り目標位置に到達するような軌道を作成する。この場合も目標位置と生成軌道の差を逆伝播していき、中継点の位置と生成軌道での中継点となる位置との差を今まで逆伝播された誤差信号に付加して運動指令の修正を行う。

(1) 生成軌道 1

走行時間を 3 秒とし、走行開始 1.5 秒後に中継点(409 mm,169 mm)を通り、目標位置(720 mm,615 mm)に到達する軌道。

この軌道生成の経緯を Fig.14 に示す。この軌道の場合繰返し 4 回で中継点の近くを通り、目標位置に到達する軌道を生成している。

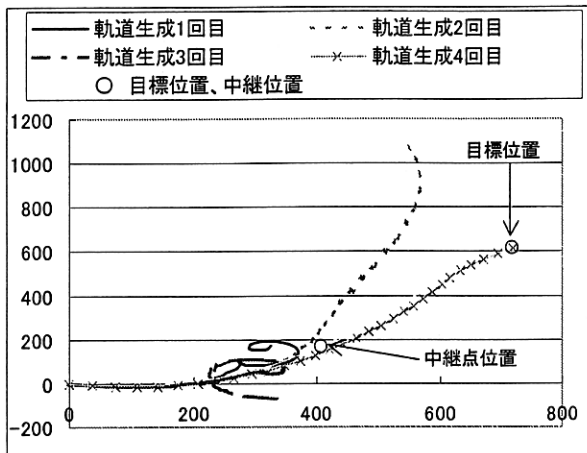


Fig.14 軌道生成の結果：軌道 1

(2) 生成軌道 2

走行時間を 4 秒とし、走行開始 2 秒後に中継点(587 mm,-251 mm)を通過し、目標位置(1158 mm,-564 mm)に到達するような軌道を生成する。

この軌道生成の経緯を Fig.15 に示す。この場合は繰返し 36 回で中継点を通り、目標位置に到達する軌道を生成している。

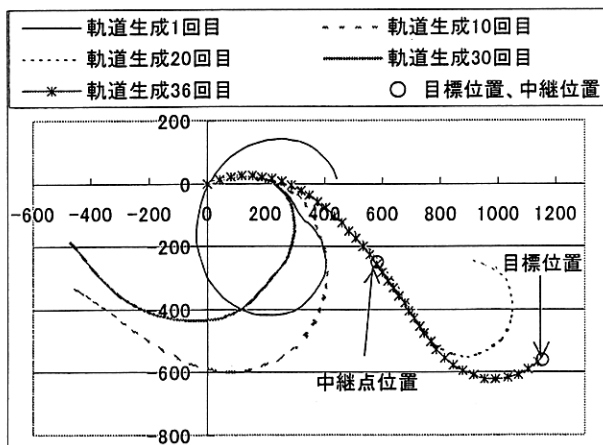


Fig.15 軌道生成の結果：軌道 2

(3) 生成軌道 3

走行時間を 5 秒とし、走行開始 2.5 秒後に中継点(586 mm,442 mm)を通り、目標位置(1389 mm,691 mm)に到達する軌道。

この軌道生成の経緯を Fig.16 に示す。この軌道の場合繰返し 126 回と他の軌道生成に比べて回数が多く、生成された軌道の中継点、目標位置との差も大きい。

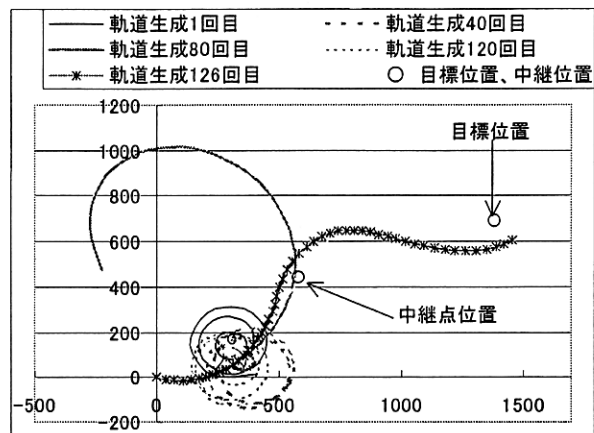


Fig.16 軌道生成の結果：軌道 3

6.3 初期運動指令を与えた軌道生成

6.2 では初期運動指令を 0 として軌道生成を行ったが、ここでは目標に近い軌道の運動指令を初期運動指令として軌道生成をする。走行時間、中継点位置、目標位置などは生成軌道 3 と同じだが、実軌道のうち目標の軌道と類似したものより初期運動指令を与えたときの軌道生成の経緯を Fig.17 に示す。この場合は繰返し 2 回で中継点を通り目標位置近くに到達する軌道を生成している。

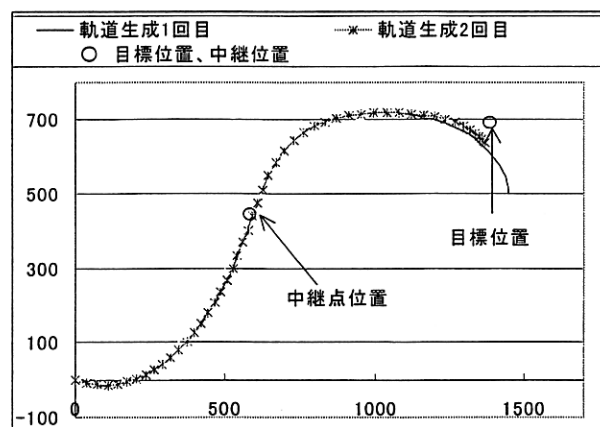


Fig.17 初期運動指令を与えた軌道生成の結果：軌道 3

6.4 考察

6.2 の生成軌道 3 と 6.3 の生成軌道を比較すると、曲がりの多い軌道の場合は初期運動指令値の入力を 0 とするよりも予め最適値に近い値を入力することが、目標への到達及び計算回数面で有効であることが分かる。これにより、われわれが以前提案したルックアップテーブル方式³⁾の有効性が実機走行においても確認できた。

また、生成した軌道はカーブ部分が膨らんでいるように見える。この原因としては、本実験において駆動力を一定としたことが考えられる。軌道生成時に中継点の通過位置と目標位置への到達時間が規定されているのに対し速度は一定であるため、軌道を膨らませ走行距離を長くすることで時間の調節を図っているのである。これに対しては、可変速で走行する走行データを学習した同定器を用い、軌道生成時に走行速度を変更することで時間の調節を行い、軌道のカーブ部分の膨らみを軽減することが可能と考えられる。また、今回の同定器を使用する場合は何らかの方法で最適な走行時間を決定して軌道生成を行うという方法が考えられる。

7. おわりに

本研究では、実機のロボット車を走行させ、その走行をビデオカメラで撮影して動特性を取得した。そしてニューラルネットを用いて動特性の学習を行った。その後、そのニューラルネットを使用した繰返し構造型ネットワークにより軌道生成を行った。この生成軌道は目標位置に到達するような軌道となっており、この方法で動特性を考慮した走行制御が可能なことを示している。

今後の課題としては速度変化のある動特性を学習した CNN での走行制御及び、障害物回避制御を検討することである。

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走行制御、電子情報通信学会論文誌 (D-II)、Vol. j79-D-II No. 1、91/100、(1996)
- 2) 大槻恭士、大友照彦：動特性測定のための移動ロボット車制御システム、平成 14 年度電気関係学会東北支部連合大会、36、(2002)
- 3) 大友照彦、大槻恭士：ニューラルネットを用いた移動ロボット車における軌道生成時間の短縮、システム制御情報学会論文誌、Vol.13 No.5、250/257、(2000)