

# 移動ロボットの軌道獲得に関する研究

## Study on trajectory acquisition in moving robot

○鳥羽 慶, 大槻 恭士, 大友 照彦

○Kei Toba, Takashi Otsuki, Teruhiko Ohotomo

山形大学

Yamagata University

キーワード: 軌道生成(path generation), ニューラルネットワーク(neural network), 動特性学習(study of dynamics), 同定器モジュール(identification module)

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  
鳥羽 慶, Tel:(0238)26-3382, Fax:(0238)26-3388, E-mail:toba@banana.yz.yamagata-u.ac.jp

### 1. はじめに

ロボットが与えられた要求を満たす最適な運動を自発的に行うことは、各所で運用されている産業用ロボットの自動化や省力化を行う上で重要な要素であると考えられる。さらに、外界の情報を取り込み、外界の状態に対応した運動を行うことは、ロボットの柔軟な運用を行う上で必要な技術である。

ところで、ユーザーからの要求を満たす最適な運動を行うためには、条件を満たす軌道を決定し、その軌道を実現するための制御入力を決定しなくてはならない。しかしながら、軌道を決定しなければ制御入力を決定することができず、逆に、制御入力を決定しなければ軌道を決定することができない。このため、最適な運動を獲得するためには軌道と制御入力を同時に生成する必要がある。

運動生成のためには、運動を行う対象をモデル化

する必要があるが、対象の自由度が大きくなればなるほどモデルを数式で記述するのが困難になる。一般に市販されている運動解析ソフトウェアの場合、拘束条件付き運動方程式(混合微分代数方程式)により、内部でモデル化を行っているものの、実時間で解析することが不可能であり、多くの計算時間を必要とする。さらに、摩擦係数などの物理的パラメータを正確に推定し、モデル化することは不可能である。

そこで本研究では、ロボットのモデル化、つまり、動特性の獲得をニューラルネットワーク(以下、NN)によって行い、そのNNを縦続接続した繰り返し構造型ネットワークによって、各種拘束条件を満たす最適な軌道生成と制御入力を、同時に獲得することを目的とする。本稿では例として単純な2リンク構造のロボットを取り上げ、自身を移動する軌道生成を行い、計算機シミュレーションにより有用性を確かめる。

## 2. 制御対象のモデル化

制御の対象となる2リンクロボットを Fig.1 に示す。リンク1, リンク2の根元にはモータを配置し、リンクの質量は一様分布と仮定するため、リンクの重心は中央にあるものとする。

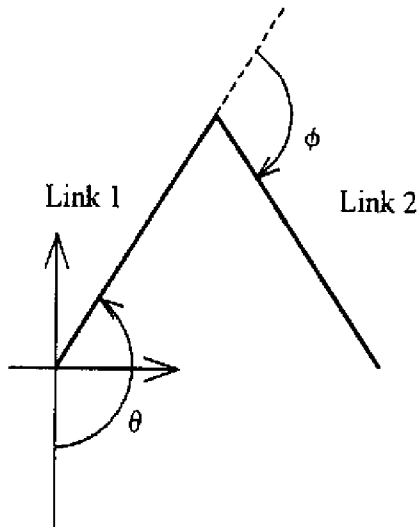


Fig.1 Control object

ロボットの角度ベクトル  $X$ , 制御入力ベクトル  $U$  を次のように表す。

$$X = [\theta, \phi]^T \quad (1)$$

$$U = [t_1, t_2]^T \quad (2)$$

一般に Fig.1 のような多リンク機構はラグランジュの方程式により、次のような一般式で表すことができる。

$$M\ddot{X} + H + G + D = U \quad (3)$$

$M(x)$ : 慣性行列,  $H(x, \dot{x})$ : 遠心力項

$G(x)$ : 重力項,  $D(x)$ : 摩擦

この方程式は一般に非線形性が強く、制御が困難である。ここで、 $U$  を決定し、式(3)を数値計算することで、2リンクロボットの順ダイナミクスを解くことができる。本研究においては、修正オイラー法により数値計算を行う。

歩行軌道を計算するにあたり時間短縮のため、いくつかの仮定をする。すなわち実際の生物の動作を観察すると、通常の歩行動作において、前へ振り出す足が地面と接触しつつ動かすことの無いことか

ら次のような仮定を行う。

- 地面に接触しているリンクは、接地していないリンクが運動を停止するまで地面から離れることはない。
- 接地していないリンクは地面を擦りながら動くことはない。

## 3. 制御システムの構成

### 3.1 同定器モジュール

式(3)を変形すると

$$\text{状態変数 } Z = [X, \dot{X}]^T = [\theta, \phi, \dot{\theta}, \dot{\phi}]^T \quad (4)$$

$$\text{入力 } U = [t_1, t_2]^T \quad (5)$$

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

となり、状態の増分が現在の状態  $Z$  と制御入力  $U$  の関数として表現される。つまり、 $Z$  と  $U$  を入力したときの1サンプリング時間後の状態の増分  $\Delta Z$ , つまり制御対象の入出力関係が同定できれば、制御対象をオイラーの数値計算で解くことが可能となる。そこで、Fig.2 に示す構造の同定器モジュール(以下IM)を構築する。

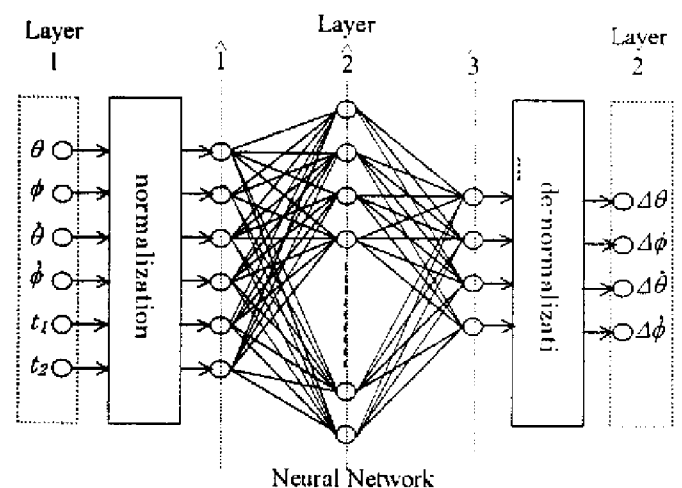


Fig.2 Structure of identification module

Layer1 に状態  $Z$  を入力し、内部に汎化能力をもつ3層の全結合 NN によって入出力関係を同定し、Layer2 に  $\Delta t$  後の状態の変分  $\Delta Z$  を出力する。時刻  $t$

を離散時間  $k(t=k\Delta t)$  を用いて表す。時刻  $k$  における Layer1, Layer2 の入力を、それぞれ  $U^1, U^2$  とすると、学習を十分に行った IM においては次式が成立する。

$$u^1[k] = \begin{bmatrix} Z[k] \\ U[k] \end{bmatrix} = \begin{bmatrix} X[k] \\ \dot{X}[k] \\ U[k] \end{bmatrix} \quad (7)$$

$$u^2[k] = \Delta Z[k] = \begin{bmatrix} \Delta X[k] \\ \Delta \dot{X}[k] \end{bmatrix} \quad (8)$$

時刻  $k$  における第  $l$  層ユニットの入力を  $u_l$ 、出力を  $v_l$  とし、次のように定義をする。

$$u^l[k] = [u_1^l[k], u_2^l[k], \dots, u_{n^l}^l[k]]^T \quad (9)$$

$$v^l[k] = [v_1^l[k], v_2^l[k], \dots, v_{n^l}^l[k]]^T \quad (10)$$

$l=1$  のとき,  $k=k_0, \dots, k_f$

$l=2, \hat{1}, \hat{2}, \hat{3}$  のとき,  $k=k_0, \dots, k_f - 1$

$n^l$  は第  $l$  層ユニットの個数を示し,  $n^1=6, n^2=4, n^{\hat{1}}=6, n^{\hat{2}}=J, n^{\hat{3}}=4$  である。  $J$  は NN の隠れ層の個数である。

このとき IM の各ユニットの入力と出力は次式で表される。

$$u^l[k] = \omega^{l-1,l} v^{l-1}[k] + \xi^l \quad (l=\hat{2}, \hat{3}) \quad (11)$$

$$u^{\hat{1}}[k] = \omega^{\hat{1},\hat{1}} v^{\hat{1}}[k] + \xi^{\hat{1}} \quad (12)$$

$$u^2[k] = \omega^{\hat{3},2} v^{\hat{3}}[k] + \xi^2 \quad (13)$$

$$v_l^l[k] = \begin{cases} \frac{2}{1 + \exp(-u_l^l[k])} - 1 & (l=\hat{2}, \hat{3}) \\ u_l^l[k] & (l=1, \hat{1}, 2) \end{cases} \quad (14)$$

第 1 層 - 第  $\hat{1}$  層は NN に入力するために  $(-1, 1)$  へ正規化する働きを持ち、第  $\hat{3}$  層 - 第 2 層においては、現実の座標系への写像を行う。第  $\hat{1}$  層 - 第 2 層 - 第  $\hat{3}$  層は全結合型の 3 層ニューラルネットワークであり、式(14)はユニットの応答関数 (シグモイド関数) である。各層間の結合係数  $\omega$  と閾値  $\xi$  は次式による。

$$\omega^{1,\hat{1}} = \text{diag}[\omega_1^{1,\hat{1}}, \omega_2^{1,\hat{1}}, \dots, \omega_6^{1,\hat{1}}] \quad (15)$$

$$\omega^{l-1,l} = \begin{bmatrix} \omega_{11}^{l-1,l} & \omega_{12}^{l-1,l} & \dots & \omega_{16}^{l-1,l} \\ \vdots & \vdots & \ddots & \vdots \\ \omega_{j1}^{l-1,l} & \omega_{j2}^{l-1,l} & \dots & \omega_{j6}^{l-1,l} \end{bmatrix} \quad (l=2,3) \quad (16)$$

$$\omega^{\hat{3},2} = \text{diag}[\omega_1^{\hat{3},2}, \omega_2^{\hat{3},2}, \dots, \omega_4^{\hat{3},2}] \quad (17)$$

$$\xi^l = [\xi_1^l, \xi_2^l, \dots, \xi_{n^l}^l]^T \quad (l=1,2,3) \quad (18)$$

$\omega^{1,\hat{1}}, \omega^{\hat{3},2}, \xi^{\hat{1}}, \xi^2$  は外部空間の座標系と NN 内部の座標系との関係を示すものであり、経験的に求められる。 $\omega^{\hat{1},\hat{2}}, \omega^{\hat{2},\hat{3}}, \xi^{\hat{2}}, \xi^{\hat{3}}$  はバックプロパゲーション法により修正される。このときの学習パターンは入力  $(Z, U)$ 、教師  $\Delta Z$  の組である。バックプロパゲーション法によって獲得された結合係数としきい値は、学習によって得られた知識として利用する。

### 3.2 繰り返し構造ネットワーク

式(6)の微分方程式をオイラー法で解くと

$$\begin{aligned} Z[k+1] &= Z[k] + F(Z[k], U[k])\Delta t \\ &= Z[k] + \Delta Z[k] \end{aligned} \quad (19)$$

となる。

よって、 $\Delta Z[k]$  を出力する IM をカスケード型に配置した繰り返し構造型ネットワーク (Fig.3) を用いて運動開始から運動終了までの制御対象の状態  $Z[k]$  の推定を行うことができる。ここで時刻  $k+1$  における IM への入力は次式で表現される。

$$\begin{bmatrix} u_1^1[k+1] \\ \vdots \\ u_6^1[k+1] \end{bmatrix} = \begin{bmatrix} v_1^1[k] \\ \vdots \\ v_6^1[k] \end{bmatrix} + v^2[k] \quad (20)$$

この神経回路モデルでは、初期値  $Z[k_0]$  と運動指令  $U[k]$  を入力する順方向計算により、制御対象の軌道を推定することができ、評価関数から決定される誤差信号を逆伝播する逆方向計算により、運動指令  $U$  を生成することができる。よって、このシステムに

においては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、最適な軌道生成と制御入力を同時に獲得することができる。さらに、制御対象からフィードバックされる状態  $Z$  を現在の離散時間に対応したモジュールに入力することで、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。

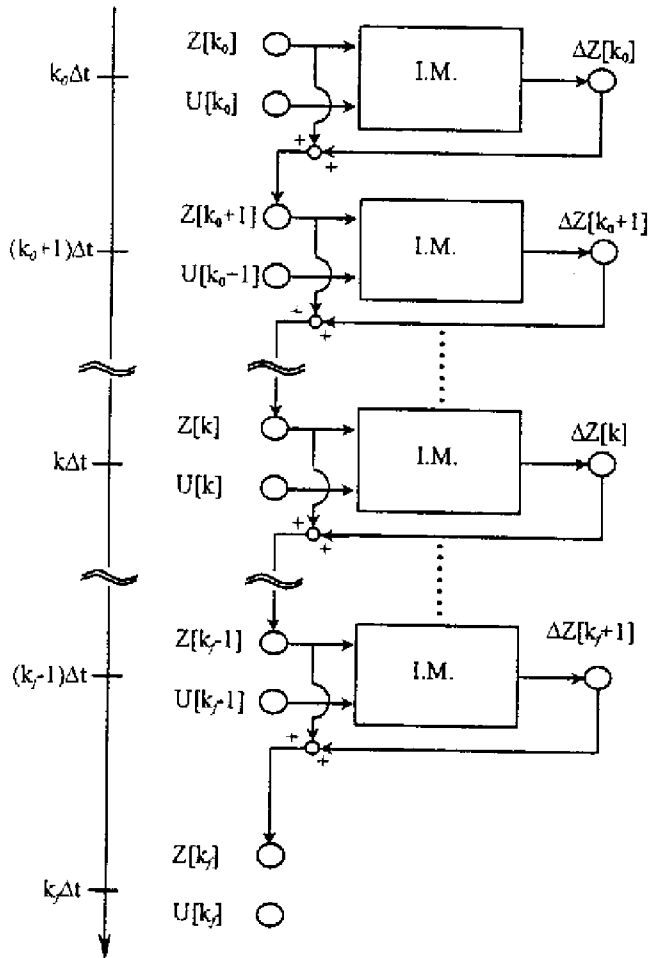


Fig.3 Cascaded network

## 4. 最適軌道生成システム

### 4.1 運動評価アルゴリズム

運動を決定するためには、最終目標状態を達成するのみならず、制御対象を円滑に運動させるため、運動指令変化を最小にする事なども考慮にいれなくてはならない。さらに、人間の関節運動を考慮に入れた場合、加速度の時間変化率である躍度を最小にするように軌道を計画する必要もある。

よって本研究においては、これらの要素を評価関数  $L$  として次式に定義をする。

$$L = E_i + E_u + E_a \quad (21)$$

$$E_i = \frac{1}{2} \sum_{k=k_0}^{k_f} e_i[k]^T W[k] e_i[k] \quad (22)$$

$$E_u = \frac{1}{2} \sum_{k=k_0}^{k_f-2} e_u[k]^T \lambda_u e_u[k] \Delta t \quad (23)$$

$$E_a = \frac{1}{2} \sum_{k=k_0}^{k_f-1} e_a[k]^T \lambda_a e_a[k] \Delta t \quad (24)$$

ここで、

$$e_i[k] = Z^*[k] - Z[k]$$

$$e_u[k] = U[k+1] - U[k]$$

$$e_a[k] = \Delta \dot{X}[k] - \Delta \dot{X}[k]$$

$E_i$  は目標状態における評価関数、 $E_u$  は運動指令変化最小に関する評価関数、 $E_a$  は加速度変化最小に関する評価関数を表す。 $Z^*[k]$  は目標状態、 $W[k]$ 、 $\lambda_u$ 、 $\lambda_a$  は評価係数を示し、

$$W[k] = \text{diag}[w_1[k], w_2[k], \dots, w_6[k]]$$

$$\lambda_a = \text{diag}[\lambda_{a1}, \lambda_{a2}], \quad \lambda_u = \text{diag}[\lambda_{u1}, \lambda_{u2}]$$

となる。

評価関数  $L$  により、運動指令ユニット値を次の修正式により最急降下方向に調節する。

$$\frac{dU_i[k]}{dt} = -\varepsilon_u \frac{\partial L}{\partial U_i[k]} \quad (25)$$

ここで

$$U_i[k] : i=1,2, \quad k=k_0+1, \dots, k_f-1, \quad \varepsilon_{u_i} : \text{修正係数}$$

### 4.2 制御システム構成

システム概略図を Fig.4 に示す。

制御対象の初期状態と目標状態、各種評価係数を制御コントローラに入力することで、制御器内部のネットワークが評価関数  $L$  を最小化するように動作し、運動の条件を満たす運動指令  $U$  が生成される。内部の IM は制御対象の動特性を獲得しているので、評価関数  $L$  が最小になり、ネットワークの計算が収束した状態では、運動方程式(3)を満足し、かつ、目標状態の条件も満たしている。

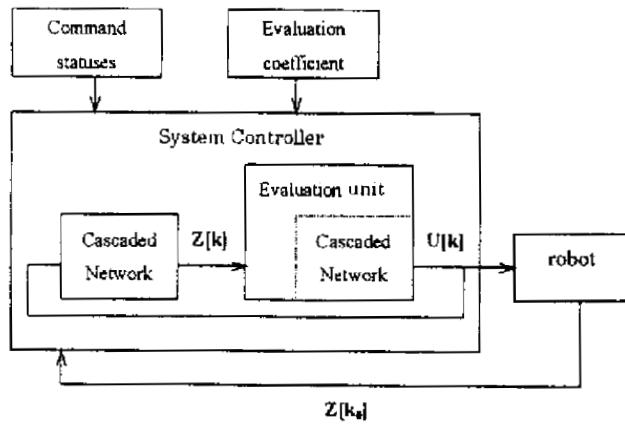


Fig.4 Control system

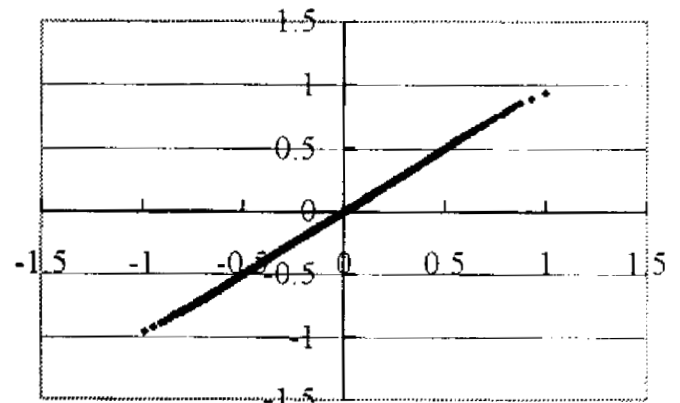
## 5. シミュレーション

本研究における制御システムの有用性を、シミュレーションにより示す。制御対象となる2リンクロボットのパラメータを Table 1 に示す

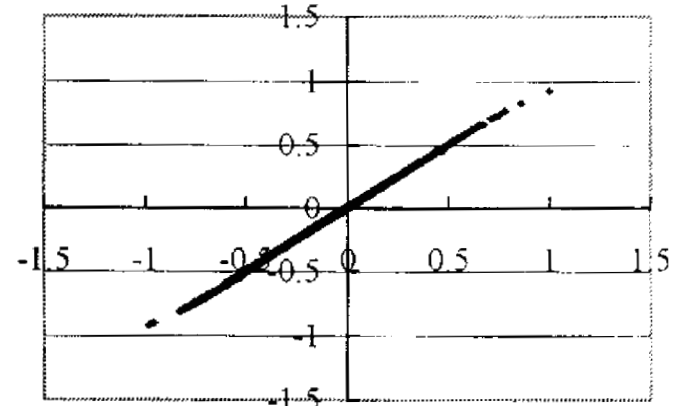
Table 1 parameter of 2link robot

	リンク 1	リンク 2
重さ	0.2[kg]	0.2[kg]
長さ	0.5[m]	0.5[m]
関節粘性抵抗	0.1[Nm·s/rad]	0.1[Nm·s/rad]

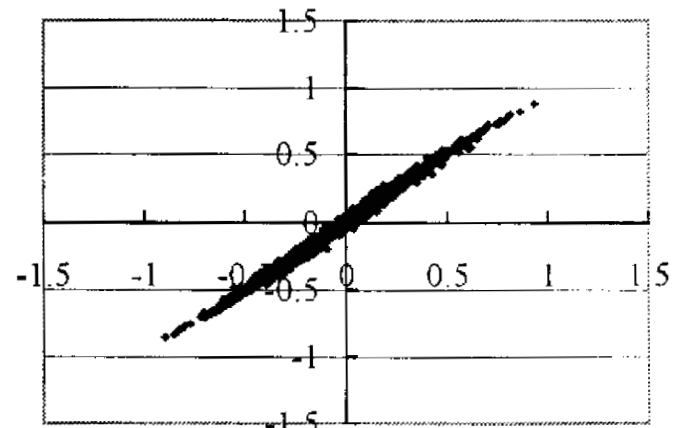
IMの動特性獲得には、計算機シミュレーションにより、刻み時間 0.01[s]として数値計算を行った制御対象の 9000 パターンの入出力データを生成し、IMの動特性学習データとして使用した。IM内部のNNは中間層の数を64個とし、学習はバックプロパゲーション法を用い、学習回数は自乗誤差が収束するまでの200000回行った。Fig.5の(a)-(d)はIMの動特性獲得状況を示す。図の値はそれぞれ、 $\Delta\theta$ 、 $\Delta\phi$ 、 $\Delta\dot{\theta}$ 、 $\Delta\dot{\phi}$ を正規化したものを示す。横軸は運動方程式(3)を数値計算して得られた値であり、縦軸はIMによって推定された値である。よって、動特性推測能力が高ければ、一つ一つの点は傾き1の直線の上に乗ることになる。Fig.5より(c),(d)の角速度の変化量出力に関しては、多少のずれが見られるものの、誤差は10%以内である。多くのパターンについては、傾き1の直線付近に集まっており、動特性推測能力、もしくは順ダイナミクスを表現するのに



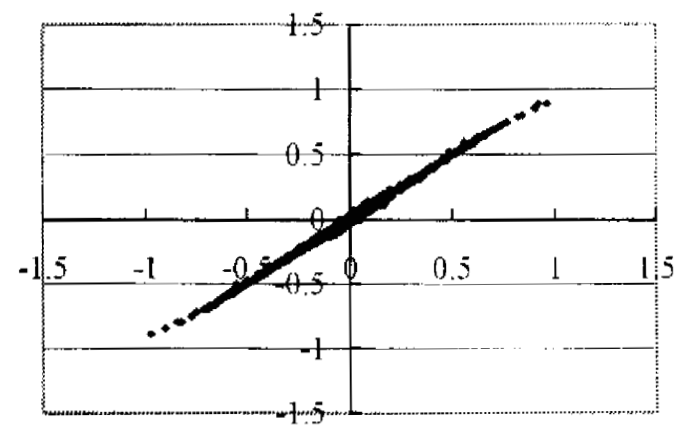
(a)  $\Delta\theta$



(b)  $\Delta\phi$



(c)  $\Delta\dot{\theta}$



(d)  $\Delta\dot{\phi}$

Fig.5 Neural network

十分な精度を保持していると考えられる。

### 5.1 順方向計算

制御対象の動特性を IM によって獲得可能であることから、繰り返し型ネットワークを構築することができる。初期状態  $Z[k_0]$  と、離散時間の運動開始時  $k_0$  から運動終了時  $k_f$  までの運動指令  $U[k]$  を入力することで、制御対象の状態  $Z[k]$  を推測することができる。順方向計算により、制御対象の状態  $Z[k]$  が推測可能な事を示すため、下記の条件で計算機シミュレーションを行った。

初期状態

$$Z[k] = (0.3, 0.0, 0.0, 0.0)$$

運動指令

$$U[k] = (0.0, 0.0)$$

IM の個数

$$k = 500$$

この運動は、第 1 リンクの角度を  $0.3[\text{rad}]$  とし、静止状態から自由運動させた場合と同じである。

繰り返し構造型ネットワークから出力された状態  $Z[k]$  の第 1 リンク角度  $\theta$  について、状態推測結果を Fig.6 に示す。

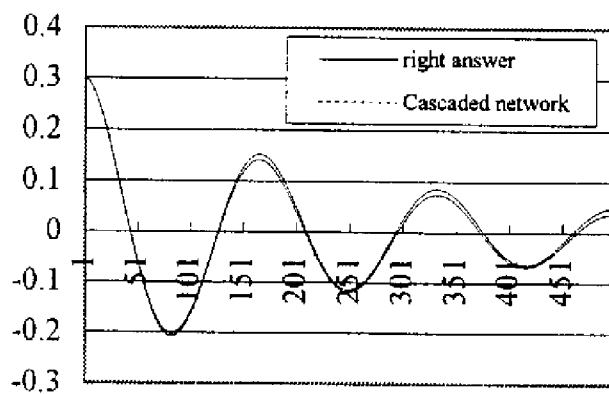


Fig.6 guessed of  $\theta$

Fig.6 は第 1 リンク角度について、カスケード構造が出力した値と、(3)の運動方程式を数値計算した値を plot した結果である。グラフより繰り返し構造型ネットワークは、変化量の大きい所で多少のずれが見られるものの、IM の最後まで精度よく状態を推測していることが分かる。

### 5.2 運動軌道と制御入力の生成

学習済みの IM を用いて繰り返し構造型ネットワークを構築し、逆方向計算により誤差を逆伝播させ、評価関数  $L$  を最小にするような軌道と制御入力を決定する。ここでは、次に示す 4 つの場合についてシミュレーションを行った。

1. 目標状態到達モデル  $L = E_f$
2. 加速度指令変化最小モデル  $L = E_f - E_i$
3. 運動指令変化最小モデル  $L = E_f - E_i$
4. 併用モデル  $L = E_f - E_i - E_a$

今回の計算機シミュレーションにおいては、歩幅を事前に決定しているものとする。歩行動作の前半と後半の運動は、基本的に同様のアルゴリズムで運動を生成していると仮定し、歩行動作における半周期分のデータを示した。制御パラメータを下記に示す。

$$Z[k_0] = [2.79, -2.44, 0.0, 0.0]$$

$$Z[k_f] = [2.35, 1.57, 0.0, 0.0]$$

$$W[k] = \text{diag}[0.5, 0.5, 0.5, 0.5]$$

$$U[0] = [0.0, 0.0], U[k_f] = [0.0, 0.0]$$

$$k = 100, \Delta t = 0.01[\text{s}], \varepsilon_u = 0.1$$

$$\lambda_u = \text{diag}[0.001, 0.001], \lambda_a = \text{diag}[0.005, 0.005]$$

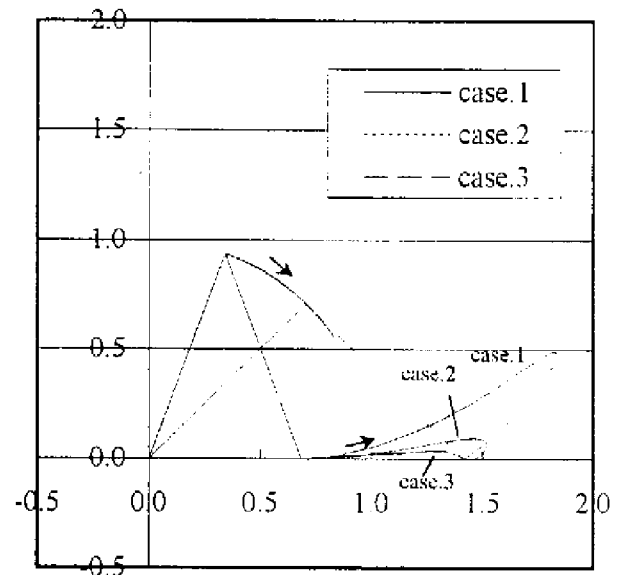


Fig7 Path

Fig.7 に生成された軌道を示す。case.1, case2, case3 はそれぞれ前に示した 4 つの評価関数に対応する。case1 では、最終目標位置しか条件に与えていないため最終到達位置の条件は満たしているものの、中

間の軌道には無駄がある。case.2,3 は運動中間の軌道生成に関しても条件を与えられているため、case.1 に比べ明らかに無駄な運動は抑制されている。Fig.8 に併用モデルで軌道生成を行った結果を示す。

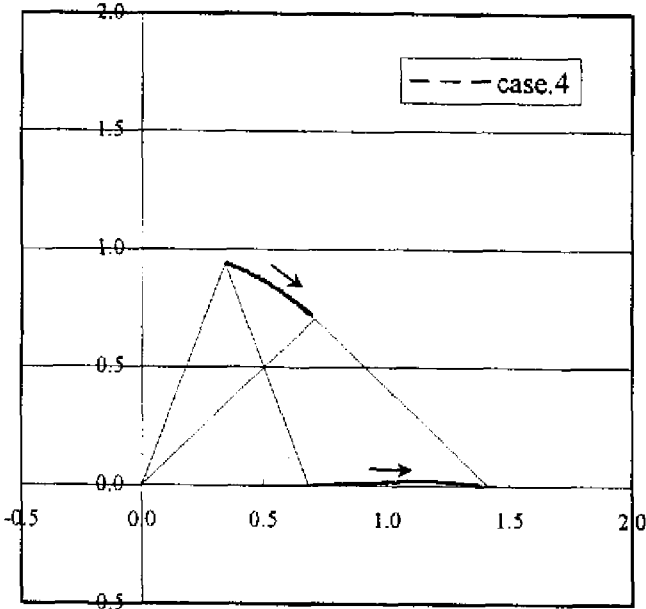


Fig.8 Path

与えられた運動指令の結果を Fig.9, Fig.10 に示す。Fig.9 は第 1 リンクに入力するトルクを表し、Fig.10 に第 2 リンクに入力するトルクを表す。最終目標位置のみを条件として与えられている case.1 では、Link1,2 両方の制御入力に関して変化量大きい。case.2,3 の場合、運動の途中にも拘束条件があるため、運動途中の制御入力の変化は小さい。次に、最終目標位置近くでの制御入力に着目する。case.2 では運動開始から制御入力量が大きくなり、制御入力量を残したまま制御入力運動を終了している。case.3 の場合、運動の最初と最後は制御入力を与えない条件  $U[0]=[0.0,0.0]$ ,  $U[k_f]=[0.0,0.0]$  を 2 点境界値問題として解いた形の制御入力を生成している。case.2,3 の運動中間付近では、case.2 の方が制御入力の変化量が小さいように受け取れるが、運動開始から終了までを考慮して制御入力を見ると case.3 の方が滑らかな関数の形で制御入力表現しており、運動指令変化最小の効果が見受けられる。運動開始と終了時の制御入力を考慮すると、case.3 の場合の方が、自然な制御入力を生成してると考えられる。case.4 は case.2 と case.3 の中間的な特性を持つように変化

している。生成された軌道から推測すると case.4 は最も優れた制御入力を生成していると考えられる。

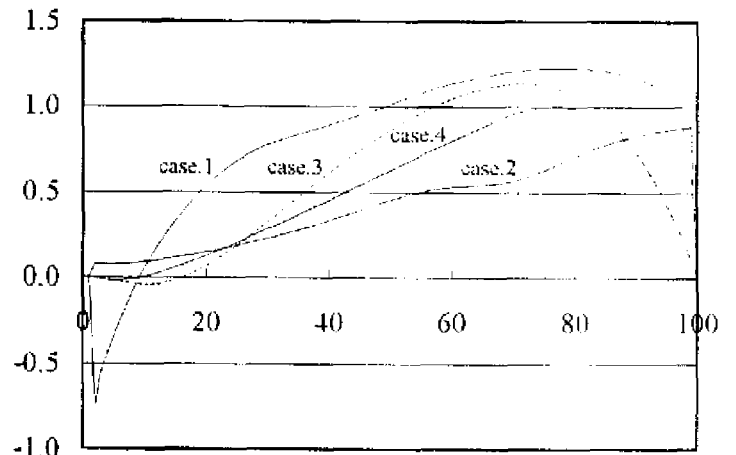


Fig.9 Link1 torque

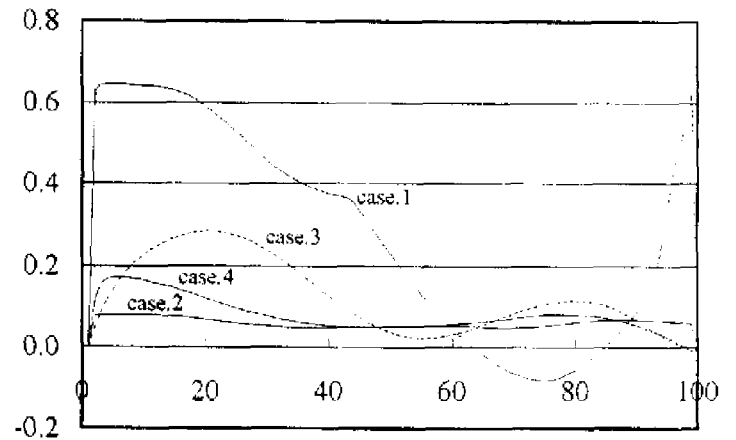


Fig.10 Link2 torque

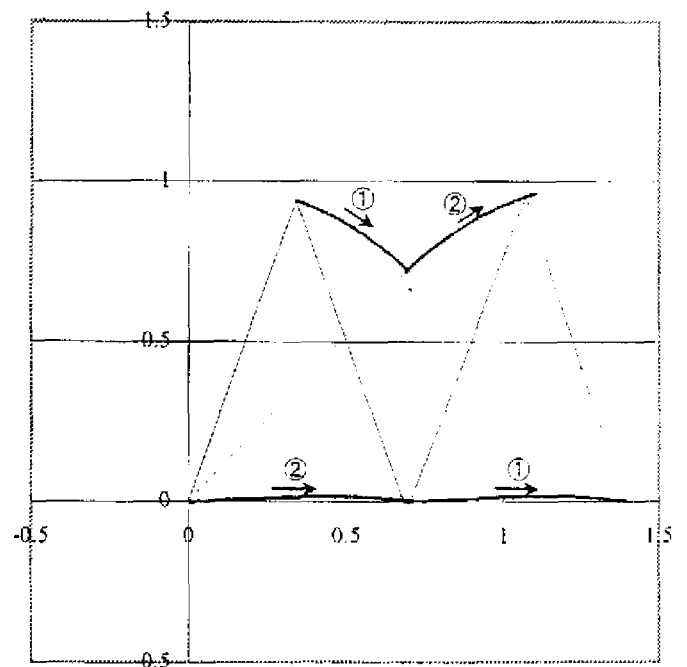


Fig.11 Path

case4 の評価関数を用いて、1歩行分の運動指令を生成し、実際のモデルに入力した結果を Fig.11 に示す。ただし、事前に生成した運動指令に関係なく、遊脚が地面に接触した段階で後半の運動指令を入力する。

## 6. まとめと今後の検討

本研究では、制御対象の動特性をニューラルネットワークを内部に含んだ IM を学習することにより獲得し、学習済みの IM を継続接続した繰り返し構造型ネットワークによって目的を満たす軌道と制御入力を同時に生成した。そして、簡単な 2 リンク歩行ロボットと、運動の条件を決定する評価関数を定義し、評価関数を最小にすることで条件を満たす軌道と制御入力を得られることを計算機シミュレーションによって確認し、アルゴリズムの有用性を確かめてきた。

計算機シミュレーションの結果より、制御対象の動特性は事前に観測した制御対象の入出力データを用いて IM を学習させることにより、獲得できることを確認した。さらに、学習済みの IM を用いて繰り返し構造型ネットワークを構築することで、指定終了時間までの運動を推定することが可能となった。運動を決定する評価関数を定義し、その評価関数を最小にするように、誤差を繰り返し構造型ネットワークを逆伝播させることで、与えられた要求を満たす軌道と制御入力を獲得し、評価関数の種類によって生成させる軌道と制御入力異なることを検討した。

今後は、環境の変化の応じた動作を生成するために、運動全体を監視する観測器を設け歩幅を可変にすることなどが考えられる。

## 参考文献

[1] 小平, 大友, 田中, 岩月, 大内: “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”, 電子情報通信学会論文誌, J79-D-II, 1, pp.91-100 (1996)

[2] 宇野洋二, 川人光男, 鈴木良治: “ロボットマ

ニピュレータの最適軌道に対する繰り返し学習制御”, 計測自動制御学会論文集, 24-8, 837/843

[3] 矢野晃一, 石川正俊: “最適軌道の探索と逆モデルの学習を行う運動の習熟機構” 電子情報通信学会論文誌, J80-D-II, 5, pp.1231-1238

[4] 新矢和行, 斎藤春雄: “マニピュレータの移動障害物自動回避軌道計画法の研究”, 日本機械学会論文集 (C 編) 61, 583, No.94-0497

[5] 川人光男 “脳の計算理論” 産業図書, 1996