

EBP学習アルゴリズムによる非線形関数の学習と その適応制御への応用

山本祥弘・古木功資^{*1}・坂本和洋^{*2}
 知能情報工学科・^{*1}ミノルタ株式会社・^{*2}鳥取大学大学院

Learning of a Nonlinear Function using a EBP Learnig Algorithm and its Application to Adaptive Control

Yoshihiro YAMAMOTO, Atsushi FURUKI ^{*1} and Kazuhiro SAKAMOTO ^{*2}

Department of Information and knowledge Engineering, Faculty of Engineering
 Tottori University, Tottori, 680 Japan
 E-mail: yamamoto@ike.tottori-u.ac.jp

^{*1} Minolta Corporation Ltd, Osaka, 541 Japan

^{*2} Graduate School of Engineering, Tottori University, Tottori, 680 Japan

Abstract : It has been shown that the EBP Learning algorithm proposed by the authors is very effective for a learning of logic functions. In this paper, the algorithm is applied for a learning of sinusoidal function and is used as an adaptive algorithm of a nonlinear adaptive control system. In the latter problem, two methods of configuration of neural network are examined in simulation studies.

Key words: Error back propagation, Learnig algorithm, Neural network, Nonlinear function, Adaptive control, Nonlinear system

1. はじめに

近年、ニューラルネットワーク(NN)が多くの分野で利用されていることは衆知のことであり、制御関係[1]~[3]においても例外でない。NNによる学習の最も魅力的なことの一つは、NNが非線形関数を任意の精度で近似できることである。従って、非線形性を考慮した制御システムの同定と解析に関して多くの研究が行われている。しかし、その学習には時間が多くかかることから、適応制御のようなオンラインでの実行には、かなり無理があるように思われる。現在までの多くの方法は、学習を行い、その結果を(適応)制御に利用する考え方であるが、見方を変えれば、適応制御は元々学習を行っていると考えられる。すなわち、パラメータ推定は一種の学習であり、その学習結果を適応的に制御に用いているのが適応制御であると解釈できる。ただ、そのパラメータ推定が、パラメータに関して線形なシステムに限定されているのが現状である。従って、NNを利用して非線形推定をおこなえば、非線形系の適応制御とな

るはずである。問題は、オンラインでの実行に耐えうるだけの高速な学習アルゴリズム(非線形適応アルゴリズム)を開発することである。

本論では、適応制御の一つの形式を非線形系に拡張した形に、新しく提案しているEBP学習アルゴリズム[4]を応用した結果を報告す。まづ第2章でEBP学習アルゴリズムを述べのべ、この方法を非線形関数の一つである正弦関数の学習に応用した結果を第3章で記す。第4章では、正弦関数を非線形項としてもつ非線形系の適応制御にNNを利用し、提案するアルゴリズムを応用した結果を述べる。この問題で最も難しい点は、どのようなNNの構成にするかである。本論でも、2つの構成法を提案しているが、NNを制御に応用した研究はすでに数多く発表されており、どの形式が最も一般的かは今後の検討課題である。

2. EBP学習アルゴリズム

入力を b_k 、出力を c_k とする3層ニューラルネットワークを次式で表す。

$$c_k = f(z_k), \quad z_k = w_{k-1}^T a_k, \quad (1a)$$

$$a_k = f(y_k), \quad y_k = V_{k-1}^T b_k \quad (1b)$$

ここに添え字 k は時刻を表す。また、ベクトル、行列のサイズは適切とする。このとき d_k を b_k に対する教師信号(目標信号)として以下のような学習アルゴリズム[4]が提案されている。

[学習アルゴリズム]

$$1) \Delta A_k = w_{k-1} (w_{k-1}^T w_{k-1})^{-1} \{ f^{-1}(d_k) - w_{k-1}^T A_k \}^T \quad (2a)$$

$$2) A_{Nk} = A_k + \Delta A_k \quad (2b)$$

$$3) A_{Nk} = H * A_{Nk} \quad (2c)$$

$$4) \Delta V_{k-1} = B_k (B_k^T B_k)^{-1} \{ f^{-1}(A_{Nk}) - V_{k-1}^T B_k \}^T \quad (2d)$$

$$5) V_k = V_{k-1} + \Delta V_{k-1} \quad (2e)$$

$$6) \Delta w_{k-1} = A_k (A_{Nk}^T A_{Nk})^{-1} \{ f^{-1}(d_k) - w_{k-1}^T A_k \}^T \quad (2f)$$

$$7) w_k = w_{k-1} + \Delta w_{k-1} \quad (2g)$$

ただし、このアルゴリズムは(1)式のデータを M 組一括した場合であり、

$$A_k = (a_k, a_{k-1}, \dots, a_{k-M+1}) \quad (3a)$$

$$B_k = (b_k, b_{k-1}, \dots, b_{k-M+1}) \quad (3b)$$

$$d_k = (d_k, d_{k-1}, \dots, d_{k-M+1}) \quad (3c)$$

である。(2c)式の H は、関数 f の値域を越えた分をカットする操作を表す。

このアルゴリズムを論理関数の学習に用いると、文献[5]にみられるように良好な結果を得ている。また別の特徴としては、 NN を a を入力とする(1a)式の2層 NN とすれば、アルゴリズムは(2f)式のみとなり、さらにユニットを線形、すなわち $f = 1$ とすれば、(2f)式は文献[5],[6]で与えられている TL SM 適応アルゴリズムと一致する。すなわち、提案する学習アルゴリズムは、従来の適応アルゴリズムの非線形系への自然な拡張であり、学習アルゴリズムは、その用い方によっては適応アルゴリズムと見なすことができる。そこで本論では、このアルゴリズムを非線形系の適応制御に応用した結果を以下に記す。

3. 非線形関数の学習

ここでは代表的な非線形関数の一つである正弦関数を提案するアルゴリズムを用いて学習させる。すなわち、教師信号として

$$y_p = 0.9 \sin(x_p) \quad (4)$$

を考える。学習データとしての入力 x_p は、以下の二通りを行った。

[方法1] 閉区間 $[0, 2\pi]$ を等間隔に $T-1$ 分割した T 点、すなわち、

$$x_i = \frac{2\pi i}{T-1} \quad (5)$$

とする。このときの学習誤差を

$$E = \frac{1}{2T} \sum_{i=0}^{T-1} (y_i - \hat{y}_i)^2 \quad (6)$$

とし、学習の終了条件を $E < 0.001$ とする。また、汎化能力の評価としてのテスト誤差を

$$E_t = \frac{1}{200} \sum_{i=0}^{99} (\hat{y}_i - y_i)^2 \quad (7)$$

とする。 $E_t < 0.0015$ のとき十分に汎化できているとする。

[方法2] 閉区間 $[0, 2\pi]$ の値をとる疑似乱数とする方法。このときのテスト誤差は()式の E_t とし、終了条件を $E_t < 0.0015$ とする。

用いる NN は1入力1出力の3層 NN とし、中間層ユニット数を25とした。出力層の非線形関数 f は線形とし、中間層の各 f は双曲線関数

$$f(b) = \frac{1 - \exp(-b)}{1 + \exp(-b)} \quad (8)$$

とする。このときのシミュレーション結果は以下のとおりである。まづ方法1により $M=3$ で学習点数 T にたいするテスト誤差の結果を Table 1 に示す。この結果から、汎化能力の向上のためには学習点数を増すことが必要である。本例では $T=8$ 以上あれば充分である。しかし、 T をさらに増やしても、学習誤差、テスト誤差ともに単調減少となっていない。ただしこのことは、学習終了時点での結果であり、学習終了条件をさらに厳しくすれば、単調減少の範囲は広がっていく。学習回数にかんしても $T=6$ までは大きく増加し、その後急速に減少するが、やはり単調減少となってい

Table 1 Simulation result of method1 with M=3

T	学習回数	学習誤差	テスト誤差
4	391	0.0010	0.0093
5	15318	0.0010	0.0069
6	81509	0.0010	0.0018
7	1750	0.0010	0.0015
8	1008	0.0009	0.0011
9	441	0.0008	0.0009
10	160	0.0005	0.0004
11	110	0.0002	0.0003
12	156	0.0009	0.0010
13	234	0.0010	0.0010
14	322	0.0005	0.0532

ないことは興味深い。Mの値にたいする最適なTの値が存在すると思われる。この中で最小の学習回数であるT=11の場合の学習が収束する様子を Fig. 1 に示す。繰り返しの初期のうちは図中左

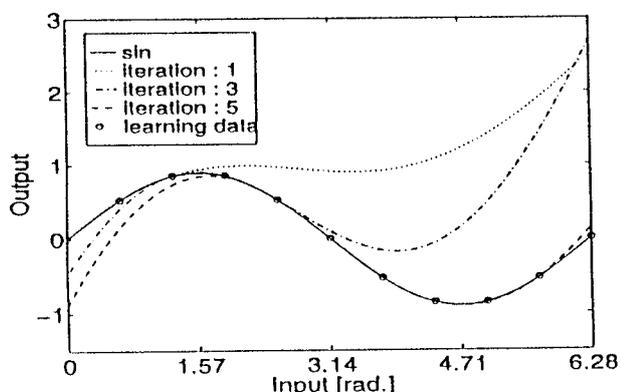


Fig.1.1 Outputs of NN with M=3 and T=11(No.1)

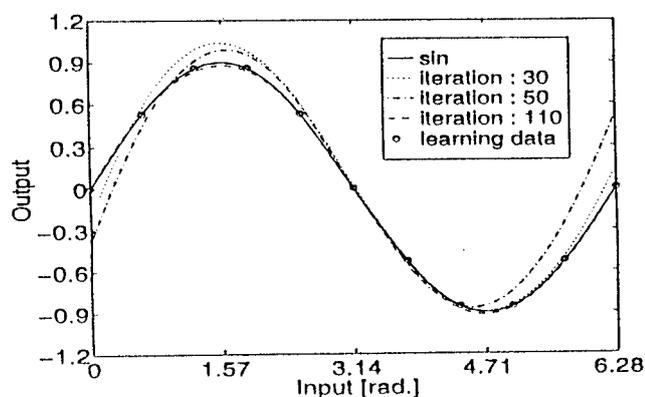


Fig.1.2 Outputs of NN with M=3 and T=11(No.2)

側から収束しているのは、学習点に起因している。すなわち、学習する3点では教師信号とほぼ一致

Table 2 Simulation result of method 1 with T=11

M	学習回数	学習誤差	テスト誤差
1	(110)	(0.572)	(0.494)
2	(110)	(0.357)	(0.388)
3	110	0.0002	0.0003
4	(110)	(1.374)	(1.497)
5	110	0.0010	0.0011

している。次に、Mにたいする変化を Table 2 に示す。表中括弧付きは収束しなかった場合であり、学習回数 110 のときの値を参考に記している。このときのグラフを Fig. 2 に示す。

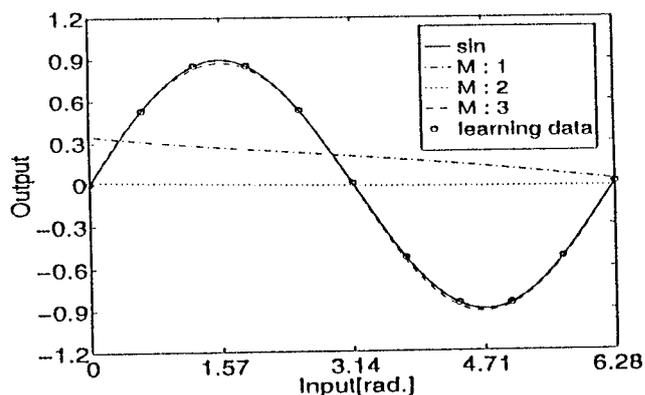


Fig.2.1 Outputs of NN with T=11 and M=1,2,3

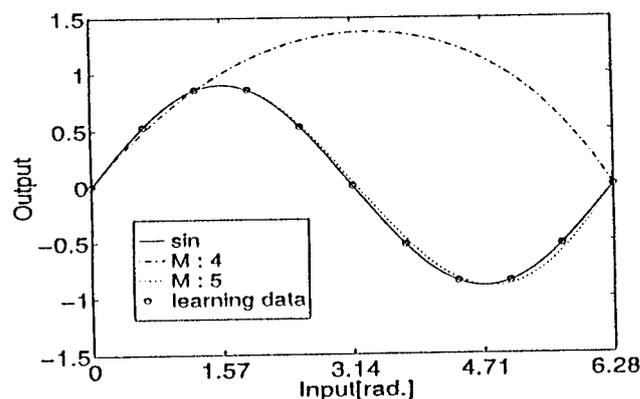


Fig.2.2 Outputs of NN with T=11 and M=4,5

Table 3 Simulation result of method 2

M	3	4	5	6	7
N	14634	146	77	6	6
E	0.147	0.133	0.075	0.004	0.002

次に、乱数により入力点を選択する方法2の結果を記す。Table 3は各Mにたいする学習結果であり、Mの増加とともに回数、誤差ともに急激に

減少している。M=3を除けば、方法1よりも良好であるといえる。M=5のときの出力を Fig. 3 に示す。

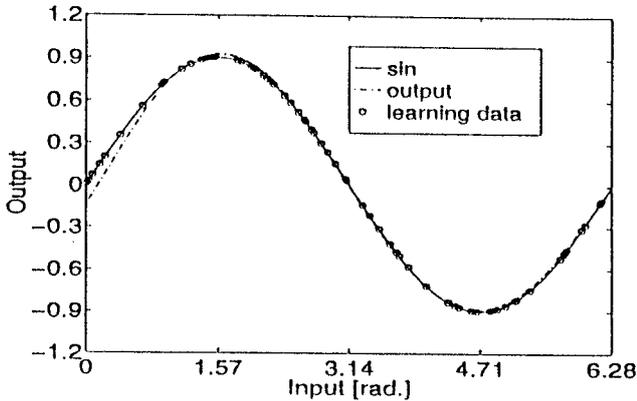


Fig.3 Output of NN with M=5 (Method 2)

4. 適応制御

ここでは簡単のため、次の2次系を考える。より一般化が可能であることは当然である。

$$y_{k+1} = -p_1 y_k - p_2 y_{k-1} + r_1 u_k + r_2 u_{k-1} + f(y, u) \quad (9)$$

ここに y_k は出力、 u_k は入力であり、最後の非線形項の変数の添え字は省略する。また、希望出力を $y_{d, k+1}$ とする。この希望出力は

$$y_{d, k+1} = -p_{d1} y_{dk} - p_{d2} y_{d, k-1} + r_{d1} u_{dk} + r_{d2} u_{d, k-1} \quad (10)$$

なる規範モデルの出力としてもよい。このとき、制御入力 u_k は、システムの入力 y_{k+1} が希望出力 $y_{d, k+1}$ と一致するように、

$$u_k = r_1^{-1} \{ y_{d, k+1} + p_1 y_k + p_2 y_{k-1} - r_2 u_{k-1} - f(y, u) \} \quad (11)$$

とすればよい。ただし、 $r_1 \neq 0$ を仮定する。

ここで、パラメータ (p_1, p_2, r_1, r_2) と非線形関数 $f(y, u)$ が未知の場合、これらをそれぞれ推定値で置き換えることにより、システム(9)式と同定器をNNで構成することになるが、その方法として以下の2つを考える。

4.1 方法1

システム(9)式を次のように表現する。

$$y_{k+1} = \theta^T v_k \quad (12)$$

ただし、

$$\theta^T = (p_1, p_2, r_1, r_2, f(y, u)) \quad (13a)$$

$$v_k^T = (-y_k, -y_{k-1}, u_k, u_{k-1}, 1) \quad (13b)$$

とする。この(7)式に対する同定器を

$$\hat{y}_{k+1} = \theta_k^T v_k \quad (14)$$

とし、これをNN1と呼ぶことにする。すなわち、

$$\theta_k = (\theta_{1k}, \theta_{2k}, \theta_{3k}, \theta_{4k}, \theta_{5k}) \quad (15)$$

であり、 $\theta_{1k} \sim \theta_{4k}$ は p_1, p_2, r_1, r_2 の推定値である。 θ_{5k} は非線形関数 $f(y, u)$ の推定値であり、以下に示すNN2の出力 c_k である。もし c_k が定数であれば、すでに知られている任意の適応アルゴリズムを用いることができる。しかし、 c_k は非線形関数 $f(y, u)$ の推定値であり、定数でないので、NN1の推定(学習)としては、提案している学習アルゴリズムを $M=1$ として用いることにする。このとき、NN1に対する教師信号は $y_{d, k+1}$ である。一般に、このような推定では、Mの値が大きいほうが安定した推定値が得られる。従って、サンプリング時間や応答の速さにもよるが、 $M=2, 3, \dots$ として、局所的に c_k を定数と近似するのも一つの方法と思われる。

次に、NN1で更新した θ_{5k+1} を d_k とし、この d_k を仮の教師信号として、 d_k の推定値を与えるのが(1)式であり、これをNN2と呼ぶことにする。すなわち、NN2は非線形関数 $f(y, u)$ の同定器の役割を果たすことになる。いま仮に、パラメータ (p_1, p_2, r_1, r_2) の値が既知であれば、 d_k はNN2に対する真の教師信号となるが、 d_k が推定値であるので、真の教師信号 $y_{d, k+1}$ から眺めると、NN1とNN2とを合わせた4層のNNであるとも解釈できる。以上をまとめると次のアルゴリズム1となる。

[アルゴリズム1]

- 1) $\theta_{5k} = c_k$ とした(12)式から、 θ_k の修正を行う。その結果を θ_{Nk} と記す。
- 2) θ_{Nk} の第5成分を $\theta_{5Nk} = d_k$ とする。
- 3) d_k を仮の教師信号として、NN(1)式の学習を行い、その出力を c_{k+1} とする。
- 4) c_{k+1} を θ_{k+1} の第5成分、1~4までは θ_{Nk}

と同じとする。

- 5) θ_{k+1} を用いて、制御(11)式、および同定出力(14)式を求める。

4.2 方法2

システム(9)式を、次のように表現する。

$$y_{k+1} = \theta^T v_k \quad (16)$$

ただし、

$$\theta^T = (p_1, p_2, r_1, r_2, 1) \quad (17a)$$

$$v_k^T = (v_{1k}, v_{2k}, v_{3k}, v_{4k}, v_{5k}) \quad (17b)$$

とする。従って、

$$\begin{aligned} v_{1k} &= -y_k, v_{2k} = -y_{k-1}, v_{3k} = u_k, \\ v_{4k} &= u_{k-1}, v_{5k} = f(y, u), \end{aligned} \quad (18)$$

と対応している。ここで以下のようにNNを構成する。最初に出力層を

$$\hat{y}_{k+1} = \theta_k^T v_k \quad (19)$$

とする。ただし、

$$\theta_k = (\theta_{1k}, \theta_{2k}, \theta_{3k}, \theta_{4k}, 1) \quad (20)$$

である。次の層を2つに分けて考える。

$$\begin{aligned} v_{1k} &= -g_{1k} y_k, v_{2k} = -g_{2k} y_{k-1}, \\ v_{3k} &= g_{3k} u_k, v_{4k} = g_{4k} u_{k-1}, \\ g_{jk} &= 1 \quad (j = 1, 2, 3, 4) \end{aligned} \quad (21)$$

および

$$v_{5k} = c_k \quad (22)$$

とし、(22)式の c_k 以下は(1)式とする。結局、全体としては4層のNNとなり、アルゴリズムを以下のようにする。

[アルゴリズム2]

- 出力誤差から次の層への仮の教師信号を決定する。これは(2a)式に相当する。言い換えれば、(19)式に対して、 θ_k でなく v_k の修正を行う。その結果を v_{nk} と記す。
- (21)式より g_{jnk} ($j = 1, 2, 3, 4$)が求まり、これか

ら $\theta_{jk+1} = \theta_{jk} * g_{jnk}$ ($j = 1, 2, 3, 4$)とする。

- $v_{5nk} = d_k$ とし、これを(1)式に対する仮の教師信号として、学習アルゴリズムを実行する。

- その出力 c_k を θ_{5k+1} として制御(11)式、および同定出力(19)式を求める。

(注意)M=1のときはこれでよいが、Mが2以上のときは v_{nk} が行列で与えられ θ_{jk+1} がM通り与えられる。このときにはそれらの平均値を θ_{jk+1} とする。

以上2つのアルゴリズムの違いは、2以上のMに対して顕著である。アルゴリズム1ではNN2の仮の教師信号を定数と近似するのに対して、アルゴリズム2では、線形部のパラメータを平均化している。

4. シミュレーション結果

数値例として

$$\begin{aligned} p_1 &= -1.6, p_2 = 0.6, r_1 = 0.29, r_2 = 0, \\ f(y, u) &= 0.29K \sin(y_{k-1}) \end{aligned} \quad (23)$$

を用いている。非線形関数のKは非線形の強さを可変にするために導入している。このときシステム(9)式の平衡点は0と π であり、 $K = 1$ のときの平衡点近傍での線形近似式の極zは

$$\begin{aligned} z &= 0.8 \pm 0.5j \quad (y = 0) \\ z &= 1.37, 0.23 \quad (y = \pi) \end{aligned} \quad (24)$$

である。従って、これは実際のシステムを対象としたモデルではないが、倒立振子の主要な特徴を備えている。

次に(1)式の構成が重要となる。非線形関数がまったく未知の場合には、NNの入力bとしては、システムの入手可能な入出力変数を多く用いることが必要となる。同時に、中間層ユニットaの個数も影響を与える。Fig.4は、M=1, K=1とし、bを y_{k-1} の1入力、aの個数を10としている。また、(1a)式のユニットは線形、(1b)式には双曲線関数をもちいている。重みパラメータの初期値は、NN1の線形部は真値*0.1とし、NN2は乱数を用いている。また、希望出力は

$$\begin{aligned} p_{d1} &= -0.8, p_{d2} = 0.15, \\ r_{d1} &= 0.3125, r_{d2} = 0.0375 \end{aligned} \quad (25)$$

とした(10)式であり、 u_a は0と π の矩形関数とした。その結果、出力応答はかなりうまくいっているように見えるが、これは線形部の補償によるものが大である。実際、非線形部を無視した適応制御でも同程度の出力応答が得られ、 $K = 2, 3, \dots$ とすると、応答はどちらの場合も同様に悪化する。この理由の一つとして、フィードバック補償がうまくいくことにより出力応答はほとんどが目標値0, π の近傍に集中しており、非線形関数の学習としてのデータが偏っていることも理由の一つと考えられる。方法2の結果は類似であるので省略する。

5. まとめ

本論では、EBP学習アルゴリズムによる非線形関数の学習とその非線形系の適応制御の一つの試みを示した。正弦関数の学習は成功しているが、初期値にたいする問題は残っている。一方、非線形系の適応制御にたいしては、期待しただけの結果は得られなかった。その最も大きな理由は、NNを用いた制御系の構成にあると思われる。線形部のパラメータも同時に推定しているため、非線形部に対する教師信号が近似的なものとなり、学習がうまく行われぬものと推定される。第2の理由としては、学習アルゴリズムとして逐次修正法を用いていることである。文献[4]からも明らかかなように、一括処理法のほうがすぐれているが、その適応制御に適した形式については現在検討中である。3番目として、(9)式の表現に問題があるのかもしれない。非線形部が線形部を含むかど

うかで、(9)式の表現に一意性がない。たとえば、 $\sin(y)$ は y を線形部として含んでいる。今後は、これらの点を考慮して、改善していきたい。

参考文献

- 1) K.S.Narendra : Adaptive Control of Dynamical Systems using Neural Networks, in Handbook of Intelligent Control, D.A.White & D.A.Sofge Eds. , pp141-184, Van Nostrand, 1992.
- 2) 大松：ニューロコントロールと適応制御、システム／制御／情報、36巻、12号、pp769-775, 1992.
- 3) 田中：ニューロコントロール～適応制御からのアプローチ、システム／情報合同シンポジウム講演論文集、pp121-125, 1997.
- 4) 山本、佐藤：EBP学習アルゴリズムの逐次修正法と一括処理法、第17回適応制御シンポジウム資料、pp37-40, 1996.
- 5) 山本：修正最小2乗法による適応アルゴリズム、計測自動制御学会論文集、26巻、12号、pp22-27, 1990.
- 6) 山本：最小二乗法の一般化による適応アルゴリズムとその液位プラントへの応用、システム制御情報学会論文誌、7巻、3号、pp77-83, 1994.
- 7) 山本：一般化適応アルゴリズムの代数的導出とその性質、第16回適応制御シンポジウム資料、pp23-26, 1996.

(受理 平成9年8月30日)

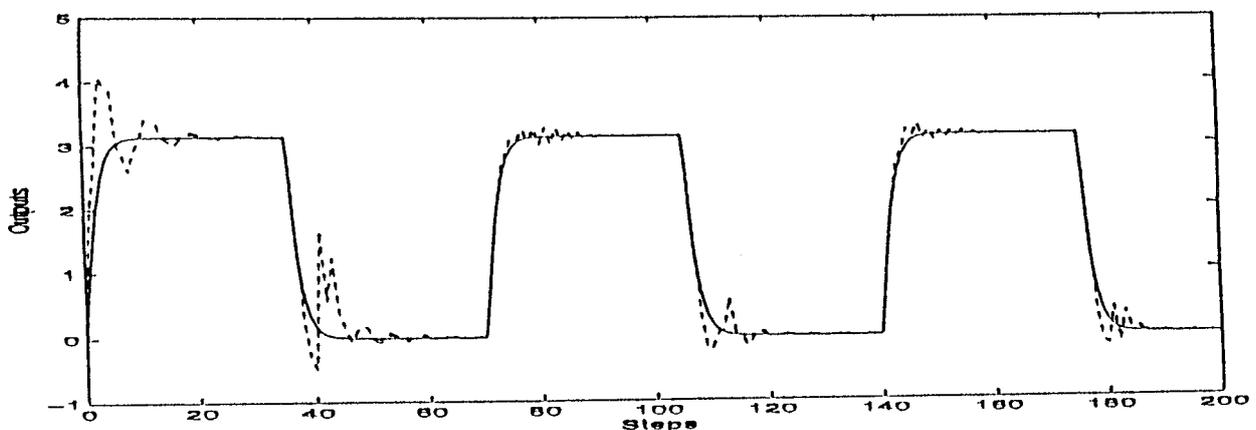


Fig. 4 Nonlinear adaptive control using neural network
(solid line: desired output, dotted line: system output)