

感度計算を用いた未知プラントのニューラルネットワーク制御の一構成法

浴, 百合雄

九州大学大学院システム情報科学府電気電子システム工学専攻 : 博士後期課程, (株)日立製作所大みか工場

平澤, 宏太郎

九州大学大学院システム情報科学府電気電子システム工学専攻

<https://doi.org/10.15017/1498345>

出版情報 : 九州大学大学院システム情報科学紀要. 3 (1), pp.95-100, 1997-12-22. 九州大学大学院システム情報科学研究所

バージョン :

権利関係 :

感度計算を用いた未知プラントの ニューラルネットワーク制御の一構成法

浴 百合 雄*・平澤宏太郎**

A New Architecture of Neural Network Controller of Unknown Plant Using Jacobian of Network

Yurio EKI and Kotaro HIRASAWA

(Received December 22, 1997)

Abstract : The plant model is able to be constructed by the neural networks, i.e., by identifying the plant model using the neural networks. But it is somehow difficult to obtain a control law from the plant model based on this neural network. The iterative inverse method has been proposed for this problem, but for this method it is necessary to determine two parameters (convergence coefficient and iterative number) and calculate the equations iteratively for obtaining a control law. The fact mentioned above is a big drawback for on-line control. This paper is related with improvement of the iterative inverse method using Jacobian of neural networks. It is shown that the proposed method is effective to control the thermal power plant by simulations.

Keywords : Neural network, Control, Identification, Non-linear, Plant Control

1. ま え が き

ニューラルネットワーク(NN)はパターン認識, 制御など多くの分野で応用されてきた。NNの制御への応用も種々の方法が提案されている^{1)~3)}。実際の制御対象を動作させながらNNの学習により, コントローラを構成することは一般的に難しい。制御対象のモデルを作ることができる場合はオフラインでNNによるコントローラを構成することができるが必ずしもモデルが構築できるとは限らない。とくに複雑で大規模なシステムでは難しい。本論文ではモデル作成が不可能な未知プラントのNNによる制御方式について検討した。今まで提案されている種々の方式について検討したが, 各々に問題があり, 次のような方式を開発した。つまり学習用のデータを実際のプラントより取り, オフラインでNNの学習(プラント同定)を行い, このNNから直接, 操作量を計算するアルゴリズムを開発した。この場合, NNによるプラントモデルは順モデルであり, 制御量を規定値にする操作量を求めるにはNNの出力から逆に入力を求めることになるがNNが非線形関数であり, 一般的に難しい。この解決策として文献3)では繰り返し逆変換法(Iterative Inverse Method)が提案されている。これはNN

の順モデルの重みを計算するのと同じ原理で操作量を繰り返し逆算するもので収束計算を行っている。この収束計算はオンライン制御においては好ましくない。これを改善する方法としてNNのヤコビアン(感度)を利用する方法を提案する。NNのヤコビアンは時々刻々プラントの線形化を行っていることに相当し, モデル規範形適応制御(MRAC)におけるプラントパラメータの係数と等価なものである。この方法は繰り返し逆変換法の欠点を改善したもので事例によるシミュレーションで良好な結果を得た。

2. ニューラルネットワークの応用

NNの学習法にはフィードバック誤差学習法¹⁾, 一般化学習法²⁾, 特殊化学習法²⁾, 順逆モデル学習法などが提案されている。図1(a)に示すフィードバック誤差学習法および図1(b)に示す特殊化学習法の場合, 実際のプラントを用いて学習するのは一般的にできないのでプラントモデルの構築が必要である。しかし, 複雑で大規模なシステムでは必ずしもプラントモデルが構築できるとは限らない。図1(c)に示す一般化学習法はプラントモデルがなくてもコントロールの構築が可能である。プラントの出力をNNの出力とし, 操作量を教師信号としてNNを学習させるいわゆる逆モデル学習法である。この場合, オフライン学習において学習誤差がどうしても発生し, オンライン実行時操作量の誤差が発生する。これに少し説明を加える。プラントモデルは種々の表現方法があるが非線形性を考慮し, 一般的につぎのような式で表せる。

平成9年12月22日 受付

* 電気電子システム工学専攻博士後期課程
(㈱日立製作所大みか工場)

** 電気電子システム工学専攻

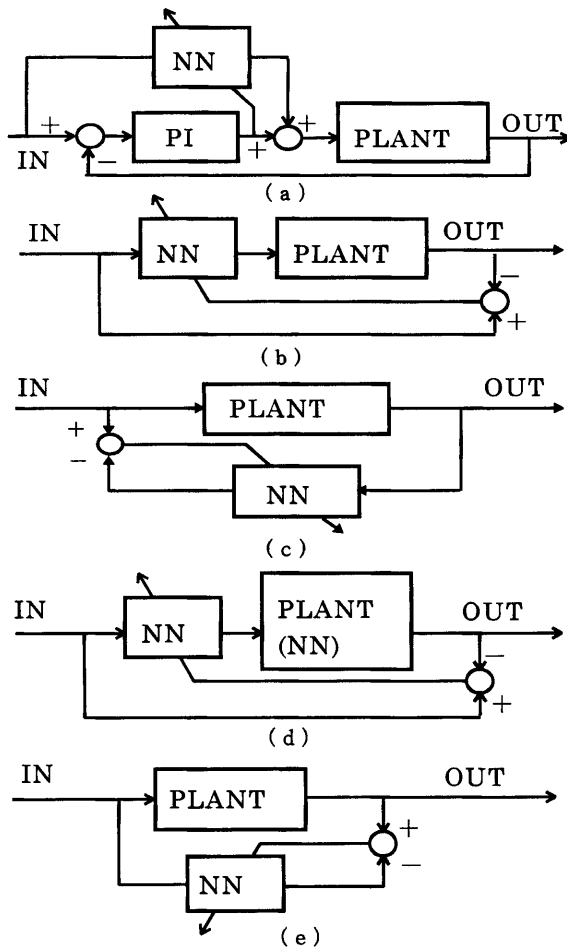


Fig. 1 NN learning method

y_t, y_{t-1}, \dots は制御量の時系列信号, u_t, u_{t-1}, \dots は操作量の時系列信号を示す.

$$y_{t+1} = f(y_t, y_{t-1}, \dots, u_t, u_{t-1}, \dots) \quad (1)$$

(1)式はプラントの順モデルあり, 逆モデルは次の(2)式を構築することになる.

$$u_t = g(y_t, y_{t-1}, \dots, y_t, u_{t-1}, \dots) \quad (2)$$

この場合, g は必ず学習誤差を伴い, ある時点で誤差を含む操作量 u_t で制御を行うと次のステップの操作量 u_{t+1} は u_t の関数となり, 次々に誤差が累積する. したがって良好な制御はできない. 図1(d)に示す順逆モデル学習法もプラントモデルがなくてもコントローラの構築ができる. この方式は先に述べた特殊化学習法(図1(b))のプラント部分を順モデルで学習し, 次のステップでは順モデル出力と設定値の差を順モデルを経由して逆伝搬し, NNコントローラの学習を行う方法である. 逆伝搬の方法として順モデルのヤコビアンを使用する. この場合, NNの学習が2段(順モデルを構成し, それを利用し

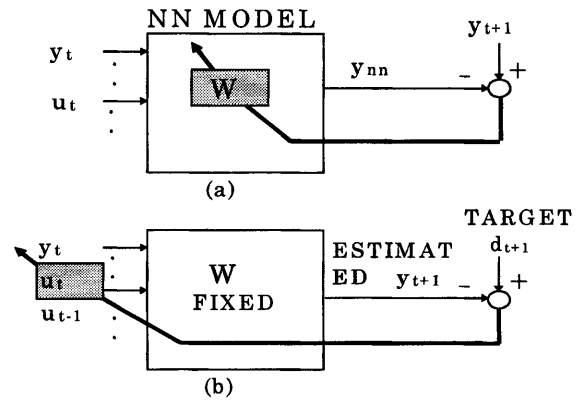


Fig. 2 Iterative inverse method

てさらに逆モデルを求める.)になっており, 学習誤差が重なる. このように従来提案されている種々の方式はプラントモデルが構築できないシステムの制御方式としては問題がある.

図1(e)に示す順モデル学習法はプラント出力とNN出力の偏差によってプラントモデルの学習を行う方式である. この順モデルから直接, 操作量を求めるにはこのNNの出力から入力を逆算する必要がある. 制御量を規定値にする操作量を求めるにはNNの出力から逆に入力を求めることになるがNNが非線形関数であり, 一般的に難しい. この解決策として文献3)では繰り返し逆変換法(Iterative Inverse Method)が提案されている. この方式について簡単に説明する. NNの出力 y_{t+1} が入力 y_t, \dots, u_t, \dots (y_t : 制御量, u_t : 操作量)と重みベクトル W で構成され ($y_{t+1} = f(y_t, \dots, u_t, \dots, W)$), 順モデルの学習で W が決定される. この固定された W に対し, W を求めた時と同じ原理で u_t を繰り返し計算で求める方法である. この方式を図2に示す. この方式は収束計算を行っているため繰り返し計算が必要で, 繰り返し回数, 学習係数を決定しなければならない. オフラインではこれらを試行錯誤的に決定することができるが, オンライン制御では好ましくない. 本論文では繰り返し逆変換法の問題点(繰り返し計算が必要など)を解決する方法としてNNのヤコビアンを用いる方法を提案する.

3. ヤコビアン方式

3.1 プラントモデルとNNの構成

プラントモデルは(1)式で表されるものとし, これをNNで構成することを考える. NNの構造方法も種々の方法(階層形NN, リカレントNN等)があるが図3のような階層型NNを考える. 図3において, 入力層の i 番目から中間層の j 番目への結合の重みを W_{ji} , 中間層の i 番目から出力層(1個)への結合の重みを W_i とする. 入力層の入力は $y_t, y_{t-1}, \dots, u_t, u_{t-1}, \dots$, 中間層の i 番目の出

力を O_{2i} , 出力層の出力を y_{t+1} とする.

3.2 NNの学習

学習方法も種々の方法(バックプロパゲーション法, ランダムサーチ法等)があるが, (1)式のようなモデルを構築するには経験的に階層形ネットワークにバックプロパゲーション法を利用するのが最適である. 入力方法についても(a)一連のデータを時系列的に投入して各ステップ毎に重みを計算する方法, (b)一連のデータをすべて投入完了後重みを計算する方法, (c)一定長の時系列データを一連のデータよりランダムに取り出し, 階層型のNNに次々に投入して各ステップ毎に重みを計算する方法がある. プロセス制御系のようなものはプロセス量の変動が緩やかで長い時間に亘るデータが特徴でこれらのデータを利用してオフラインで学習するには上記のうち(c)が最適である.

3.3 ヤコビアン計算

NNによる(1)式の学習が完了しても y_{t+1} を目標値にするため操作量を求めることはできない. これに対する方策として(1)式の瞬時線形化を考える.

$$y_{t+1} = a_0 y_t + a_1 y_{t-1} + \dots + b_0 u_t + b_1 u_{t-1} + \dots \quad (3)$$

ここで $a_0, a_1, \dots, b_0, b_1, \dots$ は(1)式のヤコビアンとみなすことができる. ヤコビアンは図3より次のようにして求められる.

$$\begin{aligned} a_{t-1} &= \frac{\partial y_{t+1}}{\partial y_t} = \sum_i \frac{\partial y_{t+1}}{\partial O_{2i}} \cdot \frac{\partial O_{2i}}{\partial y_t} \\ &= \sum_i \{f'(net_3) \cdot W_i \cdot f'(net_{2i}) \cdot W_{ip}\} \\ &= f'(net_3) \cdot \sum_i W_i \cdot f'(net_{2i}) \cdot W_{ip} \end{aligned} \quad (4)$$

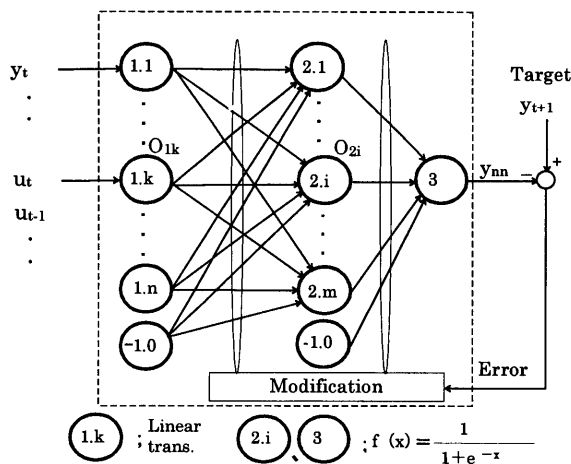


Fig. 3 Structure of neural network

$$\begin{aligned} b_{t-1} &= \frac{\partial y_{t+1}}{\partial u_t} = \sum_i \frac{\partial y_{t+1}}{\partial O_{2i}} \cdot \frac{\partial O_{2i}}{\partial u_t} \\ &= \sum_i \{f'(net_3) \cdot W_i \cdot f'(net_{2i}) \cdot W_{iq}\} \\ &= f'(net_3) \cdot \sum_i W_i \cdot f'(net_{2i}) \cdot W_{iq} \end{aligned} \quad (5)$$

但し, $net_{2i} = \sum_k O_{1k} \cdot W_{ik}$, $net_3 = \sum_i O_{2i} \cdot W_i$

(4)(5)式の W_{ip}, W_{iq} の p, q は各々 y_t, u_t の入力番号を示す.

3.4 操作量の決定

操作量 u_t の計算法を以下に示す. 評価関数は操作量を考慮し, 次式とする.

$$J = \frac{1}{2} \{ (d_{t+1} - y_{t+1})^2 + \rho \cdot u_t^2 \} \quad (6)$$

但し, d_{t+1} は y_{t+1} の目標値, ρ は操作料を制御性能のトレード・オフを示す係数である.

(6)式を u_t で微分すれば

$$\frac{dJ}{du_t} = (y_{t+1} - d_{t+1}) \cdot \frac{dy_{t+1}}{du_t} + \rho u_t \quad (7)$$

$\frac{dy_{t+1}}{du_t} = b_0$ であり, (7)式を0とおくと

$$\begin{aligned} b_0 \cdot (y_{t+1} - d_{t+1}) + \rho \cdot u_t &= 0 \\ b_0 \cdot (a_0 \cdot y_t + a_1 \cdot y_{t-1} + \dots + b_1 \cdot u_{t-1} + b_2 \cdot u_{t-2} + \dots - d_{t+1}) + b_0^2 \cdot u_t + \rho \cdot u_t &= 0 \\ u_t &= (d_{t+1} - a_0 \cdot y_t - a_1 \cdot y_{t-1} - \dots - b_1 \cdot u_{t-1} - b_2 \cdot u_{t-2} - \dots) / (b_0 + \rho/b_0) \end{aligned} \quad (8)$$

(8)式で操作量が求められる. この方式はMRACに類似しており, MRACの構成図を図4, 本方式の構成図を図5に示す. 要するにMRACはプラントパラメータを最小2乗法で求めているのに対し, 本方式ではNNモデルから求めている. 最小2乗法でパラメータを求める場合は持続的励振条件など満足しない場合もしばしば発生するが本方式ではこのような制約はない. 図5において破線はNNのオフライン学習を示し, 実線はオンライン制御を示す.

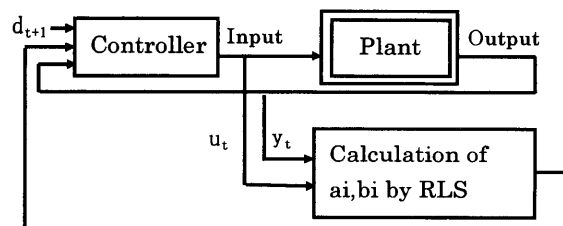


Fig. 4 Control by MRAC

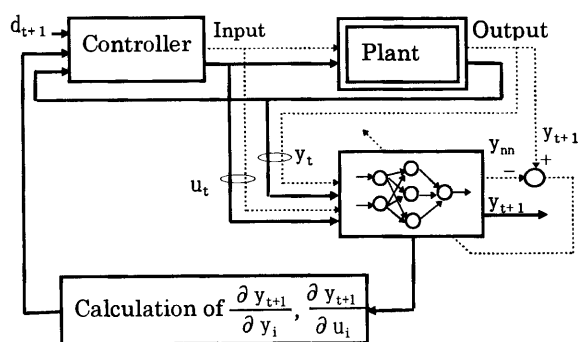


Fig. 5 Control by Jacobian Method

4. 適用例

4.1 プラントモデルと順モデルの構成

大容量事業用火力発電所のプラント制御システムを事例としてとりあげる。システム概要については文献4)を参照されたい。従来の主蒸気温度制御系の概要を図6に示す。

燃料制御信号は

- ① MWD(要求負荷信号)に見合う静的フィードフォワード信号
- ② 主蒸気温度設定値と実温度のフィードバックによるPI制御信号(図のPI)
- ③ フィードバック制御の遅れをカバーする過度フィードフォワード信号(図のFF)より成る。

本事例ではPIとFF部分、すなわち過渡時燃料信号(DFFC……Dynamic Feedforward Control)にNNを適用することを考える。目的は負荷上昇/下降時に温度変化を極小化するための最適なDFFCの決定することである。

つぎにプラントモデルは、遅れ演算子 z^{-1} を用いて表現すると次式で示される。

$$(1 + a_1z^{-1} + a_2z^{-2} + a_3z^{-3})y(t) = z^{-d}(b_0 + b_1z^{-1})u(t) + (c_0 + c_1z^{-1} + \dots + c_dz^{-d})w(t) \quad (9)$$

$w(t)$: 外乱・MWD

$u(t)$: 操作量・過渡時燃料(DFFC)

$y(t)$: 制御量・温度(設定温度-測定温度)

d : むだ時間

$1 = A(z^{-1})S(z^{-1}) + z^{-d}R(z^{-1})$ なるDiophantine方程式⁵⁾を導入し(9)式を変換すると(10)式となる。

$$y(t+d) = \{A(z^{-1})S(z^{-1}) + z^{-d}R(z^{-1})\}y(t+d) = R(z^{-1})y(t) + B(z^{-1})S(z^{-1})u(t) + C(z^{-1})S(z^{-1})W(t+d) \quad (10)$$

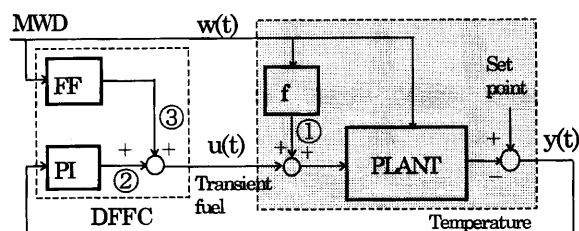


Fig. 6 Outline of main steam temperature control system

次数は $S: d-1, R: 2$ となり、 A, B, C の次数が(9)式より各々 $3, 1, d$ であるので BS, CS の次数は各々 $d, 2d-1$ となる。さらに、 $d=5$ (サンプリング周期10秒、使用したモデルのむだ時間50秒)とし、(10)式で(t)式を添字で表現すると(11)式になる。

$$y_{t+5} = f(u_t, u_{t-1}, u_{t-2}, u_{t-3}, u_{t-4}, u_{t-5}, w_{t+5}, w_{t+4}, w_{t+3}, w_{t+2}, w_{t+1}, w_t, w_{t-1}, w_{t-2}, w_{t-3}, w_{t-4}, y_t, y_{t-1}, y_{t-2}) \quad (11)$$

この場合(2)式の評価関数は次式となる。

$$J = (y_{t+5}^2 + \rho u_t^2) / 2 \quad (12)$$

目標値は0なので $d_{t+5} = 0$ である。

4.2 シミュレーション結果

図7にシミュレーション・フローを示す。火力プラント詳細解析モデルを実際のプラントと見なし、オフライン学習でプラントモデルを構成し、これを利用してオンラインで操作量を計算する。

図8は主蒸気温度(y)、MWD(w)、過渡時燃料・DFFC(u)の時系列データを示しており、このデータで(10)式の順モデル学習を行う。データの0~125%等は縦軸のフルスケースを示す。(以降、図9~13も全て同じ。)入力はDFFC: u_i , MWD: w_i , 温度: y_i , の時系列信号で出力は温度の予測値 y_{t+5} である。使用したNNは入力層、中間層、出力層が各々20, 20, 1の階層形で学習法はバックプロパゲーション法を使った。学習状況を図9に示す。図9で開始時点は約7°Cの温度偏差があったが学習回数1000回程度で急速に減少し、以後は漸減、最終的に約0.3°Cで重みも収束している。図10はNNによる温度の予測精度を検証するため、実温度とNNによる予測温度のシミュレーション結果を示す。(チャートの4, 5はスケールを拡大したもの)予測誤差は極めて小さい。(10)式のオフライン学習が完了するとアルゴリズムを制御装置に組み込みオンライン制御を実行する。図11は2%/分で適当に調整されたPI制御のみ($EF=0$)で負荷上昇、降下のシミュレーションを行ったものでかなり大きい温度変化となっている。次にこのNNで計算された重みを

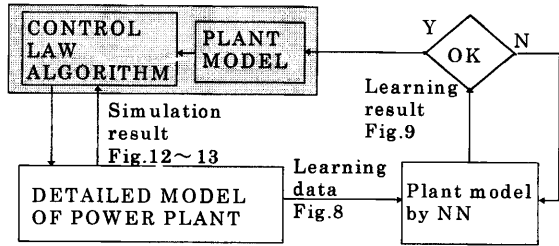


Fig. 7 Simulation flow

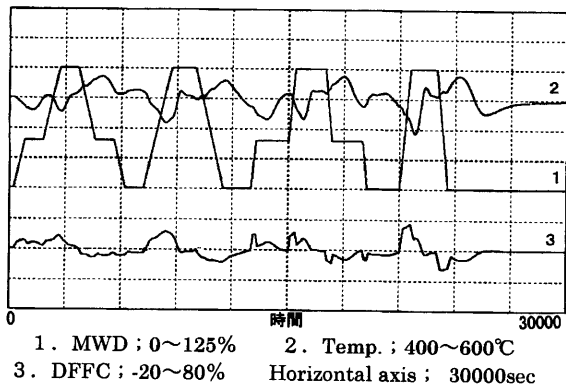


Fig. 8 Examples of Learning Data

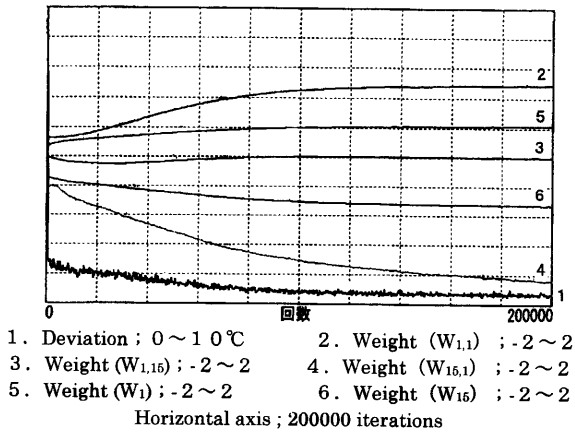


Fig. 9 Learning process of NN

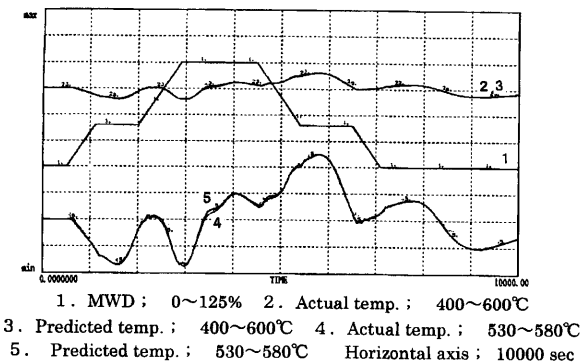


Fig. 10 Temperature prediction by NN

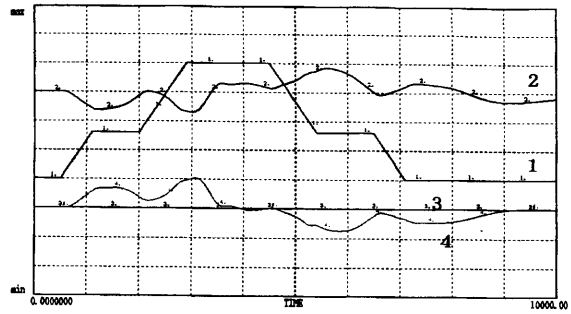


Fig. 11 Load change by PI control (2%/min.)

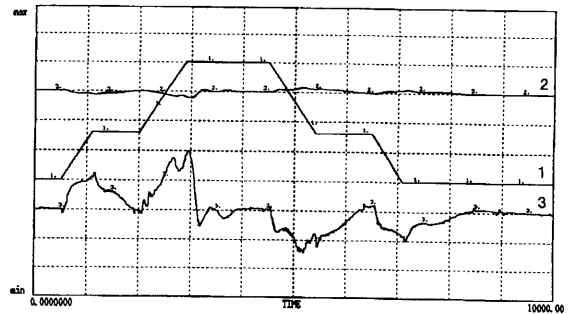


Fig. 12 Load change by proposed method (2%/min.)

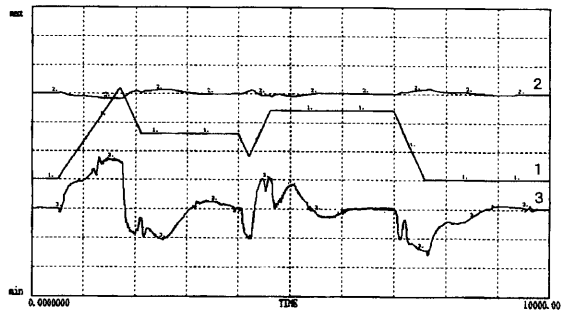


Fig. 13 Load change by proposed method (free load swing)

用いてヤコビアンを求め、それによって操作量(DFFC)を求める方式で負荷上昇、下降のシミュレーションを行った結果を図12(負荷変化率2%/分)に示す。図11と図12を比較すればNNによる方式の有効性が確認できる。また図12は図8の学習に使用した負荷変化と同じものであるがNNの汎化能力を検証するため、任意の負荷変化率、負荷変化中に行ったものを図13に示す。

5. ま と め

NNによるプラント制御においてプラント同定(順モ

デルの構成)結果より操作量を求めるのは非線形関数である NN の出力から入力を逆算することになり難点であった。この問題を解決するため繰り返し逆変換法などが提案されているが、繰り返し計算が必要で、繰り返し回数の設定、学習係数の決定が必要になり、オンライン制御においては好ましくない。本論文ではヤコビアン(感度)を利用した方式により上記の諸問題を解決した。またこの方式を火力プラントの温度制御の事例のシミュレーションにより良好な結果が得られることを確認した。この方式は一般のプロセス制御における NN 制御に広く適用できるもの考える。

参 考 文 献

- 1) 五味, 川人; フィードバック誤差学習による閉ループシステムの学習制御. システム制御情報学会論文集, Vol. 4, No. 1, pp. 37~47, 1991
- 2) D. Psaltis, A. Siders, A. Yamamura; A Multilayered Neural Network Controller. IEEE Control System Magazine, pp. 17~21, APRIL 1988
- 3) D. A. Hoskins, J. N. Hwang, J. Vagner; Iterative Inversion of Neural Networks and Its Application to Adaptive Control. IEEE TRANSACTION ON NEURAL NETWORKS, VOL. 3, No. 2, pp. 292~301, MARCH, 1992
- 4) 浴, 平澤他; 火力プラント主制御系の各種制御方式とその比較. 電気学会論文誌, Vol. 117-C, No. 3, pp. 287~294, 1997
- 5) I. D. Landau, 富塚; 適応制御システムの理論と実際. オーム社, 1982
- 6) J. Matyas; Random Optimization. Automation and Remote Control, Vol. 26, No. 2, pp. 244~251, 1965

