ニューラルネットワークによる電気自動車 実験装置の走行負荷トルク制御

正員 松村 茂 憲 (徳島大)

正員 大松 繁 (徳島大)

正員 樋笠 博 正 (四国総研)

Running Load Torque Control for Electric Vehicle Testing System by Neural Networks

Shigenori Matsumura, Member, Sigeru Omatu, Member (University of Tokushima), Hiromasa Higasa, Member (Shikoku Research Institute Inc.)

In order to develop an efficient driving system of electric vehicles (EVs), a testing system has been devised, which enables us to simulate the driving performance of EVs. In this system, running resistance of the EV is realized by torque control of the motor operated as a generator. Nowadays, torque control is performed by using conventional PID controllers, and reasonably good results have been obtained. However, when using conventional PID controllers, it generally needs much effort and time to tune the PID gains.

In this paper, instead of conventional PID controllers, three types of control systems by neural networks are considered. At first, the testing system and its conventional PID controller are described. Then self tuning PID type neurocontroller is discussed from the viewpoints of how to self-tune the PID gains automatically and parallel and series types are examined to improve the control performance. Finally, we show numerical results of the effectiveness of the presented methods through simulations and experiments, compared with the conventional PID control.

キーワード：ニューロコントローラ，セルフチューニングPID制御，トルク制御，電気自動車走行模擬装置

1. はじめに

エネルギー源の多様化ならびにCO₂問題などの観点から、エンジン自動車の代わりに電気自動車（以下、EVと略記）が注目されている。EVの電源として、鉛電池が用いられることが多い。この場合、1回の充電でEVの走行可能な距離は短く、かつこの距離は走行パターンによって大きく左右される。

そこで、適正な走行パターンを検討するために、著者らは既に、実験室内で様々な走行条件下でのEV走行パターンが模擬可能な試験装置を構築した。本装置では、電動機を発電機として運転し、その負荷トルク

でEVの走行抵抗トルクを模擬しており、望ましい負荷トルクを得るために後述するPID制御が組み込まれている。ただし、PIDゲインは実機運転の試行錯誤的なチューニングによって得られた値を用いている。このようなチューニング法は多くの労力と時間を要し、また得られたPIDゲインが最適に近い値であるか否か不明である。

ところで、ニューラルネットワーク（以下、NNと略記）を用いるニューロコントローラ（以下、NCと略記）は、制御パラメータを適応的に調整し、制御性能を目標の仕様に適合させるための優れた制御手段の一つである^{(1)~(3)}。上記の問題へこのNCを適用する

ことは、制御対象に適合したPIDゲインを自律的に決定し、制御すべき走行抵抗トルクを望ましい値へ精度良く追従させる目的のためである。

本論文では、走行抵抗トルク発生装置をNNによってモデル化し、そのモデルを制御対象とみなしたNCを新たに構成するとともに、その有効性について考察する。

まず、制御対象となる試験装置の概要とその走行抵抗トルクを実現する負荷用電動機のトルク制御系について述べる。ここで考察する制御対象は時定数が小さいため、サンプリング間隔($T=1.5$ s)に対して静的モデルで近似できる。ここでは、静的モデルとして回帰式を採用するが、そこで生ずるモデル化誤差を更に減少させるために、NNを並列的に用いた制御対象の実験モデルを作成する。この実験モデルに対して、セルフチューニングPID型制御系、並列型NCおよび直列型NCを構成し、それらの有効性を従来のPID制御による制御結果と比較し、本手法の有効性を定量的に検討する。更に、これらの手法が実機への適用においても有効であることを示す。

2. 試験装置の概要

図1は実験装置の概要を示している。この装置は負荷側と駆動側からなり、パソコン(UFAS)を介して様々な情報伝達が行われている。負荷側直流電動機M1を発電機として運転し、その負荷トルクをEVの走行パターンに応じたトルク値に制御することにより、EVの走行抵抗が模擬されている。

また、駆動側電動機M2はEVの走行パターンに応じた回転数で運転することにより、EVの駆動源を模擬している⁽⁴⁾。

特に本実験では、負荷側に直流電動機M1を、駆

動側に誘導電動機M2を用いる。その理由は、直流電動機はトルク制御を正確に行うことができ、誘導電動機はEV駆動源として頑強で、しかもその速度制御が容易に行えるためである。

本論文の実験において想定した車両モデルおよび走行パターンは以下のようなものである。

(i) 車両モデル このモデルは重量が約1,000 kgの普通乗用車クラスのEVで、電動機4台を搭載した車両の1/4モデルである。

(ii) 走行パターン 市街走行を代表する走行パターンを基準とした、以下のパターンを適用する。加速域：加速度は4 km/h/s、定速域：走行速度は60 km/hとし走行時間は10~30 s、減速域：2 km/h/sの割合で減速、停止期間：15 sとする。

3. 走行負荷トルク制御

本論文では、EV試験装置を用いて種々の走行パターンを実験室内で生成し、EV走行時の効率などを検討するために必要となる負荷側のトルク制御について考察する。

負荷側の電動機は直流電動機であるため、電機子電流を制御することによって、容易にトルク制御が実現できる。

ここで用いるEV実験装置では、走行パターンの加速・減速時に対応した発電機および電動機の役割を一つのM1で実現することが必要である。

この場合、図2に示すようにM1の特性は電動機(負の出力域)と発電機(正の出力域)では、出力/制御入力(負の出力域)と発電機(正の出力域)では、出力/制御入力の比が異なっている。従って、M1の特性を一つの数式モデルで記述することは困難である。更に、M1には種々のノイズが介在し、電動機固有の非線形特性もあり、備付けのDDCのみで正確なトルク制御を実現することは困難である。そこで、本EV試験装置はPIDコントローラがDDCの前に設置され、パソコンで制御されるようになっている。従って、トルク

図1 EV試験装置の構成
Fig. 1. Structure of EV testing system.

図2 PIDトルク制御例
Fig. 2. Example of PID torque control result.

図 3 PID トルク制御系
Fig. 3. PID torque control system.

制御系のブロック線図は図3のようになっている。ここで、PID ゲインは種々の実験を通じて試行錯誤的に決定されているのが現状である。

図3で負荷側模擬装置はDDCを含みハード的に構成されており、調整可能なものはPID コントローラのゲインのみである。

そこで本論文では、負荷側模擬装置に含まれる非線形特性を考慮したモデルをNNによって構築する。更に、そこで得られたモデルをシステムのエミュレータとして用いたセルフチューニングPID型NC、並列型NCおよび直列型NCをNNで構成し、負荷側のトルク制御を行う。

4. 試験装置の数式モデルと実験モデルの作成

〈4・1〉 モデルの記述 入出力データを用いて制御対象のモデルを構築するためには、その応答の過渡特性を把握することが必要である。図4はDDCコントローラ、サイリスタ変換装置およびモータ電機子などの構成要素からなる負荷側模擬装置のブロック線図を示している。入力 u から出力 y までのその伝達関数 $G(s)$ は次式のように求められる。

$$Y(s)=G(s)U(s) \quad \cdots \cdots \cdots (1)$$

$$G(s)=M(b_2s^2+b_1s+b_0)/(s^5+a_4s^4+a_3s^3+a_2s^2+a_1s+a_0) \quad \cdots \cdots \cdots (2)$$

ここに、

$$\begin{aligned} a_0 &= 7.71 \times 10^{11}, a_1 = 2.0 \times 10^{10}, a_2 = 1.92 \times 10^8, \\ a_3 &= 9.07 \times 10^5, a_4 = 1.83 \times 10^3, \\ b_0 &= 5.14 \times 10^{11}, b_1 = 6.68 \times 10^9, b_2 = 6.17 \times 10^6 \end{aligned}$$

ここで、 M は入力 u の値と回転数に依存して変化するモータ定数である。 $U(s)$, $Y(s)$ はトルクの入力 u および出力 y のラプラス変換である。

次に、(2)式の伝達関数 $G(s)$ はその極を用いて、次のように展開できる。

$$G(s)=M\{a_1/(s+\lambda_1)+a_2/(s+\lambda_2)+a_3/(s+\lambda_3)+a_4/(s+\lambda_4)+a_5/(s+\lambda_5)\} \quad \cdots \cdots \cdots (3)$$

ここに、

図 4 負荷側模擬装置のブロック線図
Fig. 4. Block diagram of EV load simulation system.

表 1 回帰分析表
Table 1. Table of regression analysis.

変動要因	平方和	自由度	不偏分散	分散比
全変動	36286.60	334		
回帰の変動	36156.70	2	18078.40	46211
残差の変動	129.88	332	0.39	

重相関 = 0.998 寄与率 = 0.996

回帰式: $y=0.00242u^2+1.17u+1.98$

u :制御入力(Nm), y :出力(Nm)

$$\begin{aligned} \lambda_1 &= 83.3, \lambda_2 = 124 + j99.8, \lambda_3 = 124 - j99.8, \\ \lambda_4 &= 1.19 \times 10^3, \lambda_5 = 306, \\ a_1 &= 2.45, a_2 = 4.615 + j102, a_3 = 4.615 - j102, \\ a_4 &= 1.16, a_5 = -12.8 \end{aligned}$$

これらの式より、制御系の時定数は約 10 ms (= $1/\lambda_1$) 以下であることがわかる。従って、サンプリング周期が 0.5~1.5 s では、サンプリング時における出力はほぼ定常状態となり、その入出力は静的な関係が支配的であるとみなされる。

そこで、測定が容易な定常状態のデータを用いて、計算の簡単な回帰式により入力と出力の関係を近似する。モータ定数 M に非線形要素が介在しているため、回帰式は適宜二次以上の項も含める。次に、回帰式で近似しきれない残差、すなわち回帰式に反映できない非線形要素およびダイナミクスに起因する残差を NN を用いて補正する。

〈4・2〉 モデルの妥当性の検証⁽⁵⁾⁽⁶⁾ まず、回帰式を求めるため、EV システムの入出力データの定常値を測定し、その回帰式の結果は表 1 のようにまとめられた。

次に、表 1 の回帰式で与えられた二次の近似しきれない残差を NN を用いて補正する。この NN を以後 NNM と記述する。この場合、制御対象の入出力特性を近似するために、図 2 に示した代表的な時系列データを用いて NNM を学習する。ただし、NNM は図

図 5 NNM の構造
Fig. 5. Structure of NNM.

図 6 負荷側模擬装置のモデル構成
Fig. 6. Model structure of EV load simulation system.

表 2 NNM 学習パラメータ
Table 2. Learning parameter of NNM.

学習率 (ε)	0.001
慣性項定数 (α)	0.9
減速項定数 (β)	-0.1

5のような3層の階層構造とし、近似精度と構造の簡単さを勘案し以下のような設定を行った。入力層は6ユニットとし、その入力、過去2刻み前までの制御対象の入力 $\{u(k), u(k-1), u(k-2)\}$ と出力 $\{y(k-1), y(k-2)\}$ および現時点の回転数 $\{R(k)\}$ とした。ただし、 $u(k)=u(kT)$, $y(k)=y(kT)$ とし、 T はサンプリング周期を示す。また、中間層は10ユニットとし、出力層は1ユニットとした。回帰式も含めたモデルの

図 7 NNM の学習結果
Fig. 7. Learning results by NNM.

図 8 回帰式と NNM による推定誤差
Fig. 8. Estimation error by regression method and NNM.

概要を図6に示す。学習はバックプロパゲーション法を用い、本論文で用いた(19)式に含まれる学習パラメータは表2に示すとおりである。また、ユニットの出力関数 $f(x)$ は(4)、(5)式のように設定した。

$$f(x) = \{1 - \exp(-x)\} / \{1 + \exp(-x)\} \quad (\text{中間層}) \quad (4)$$

$$f(x) = x \quad (\text{出力層}) \quad (5)$$

x : ユニットへの入力の総和

NNMの学習結果を図7(a), (b)に示す。また、モデルの推定誤差を回帰式のみを用いる場合とNNMも合せて用いる場合を比較して図8に示す。学習が進むにつれて、実測値と推定値との誤差が減少していることが図7(a), (b)からわかる。なお、(b)図より、モデルはトルクの最大値が約17 Nmに対し、誤差が0.5 Nmの範囲内で近似できており、図8より出力の変化部の補正効果がNNMを用いることによって顕著であることが示されている。

実機運転でオンライン的にNNを学習し、そこで設計されたNCを実機に適用することは危険性を伴っている。そこで本論文では、本章で得られた実験モデルを制御対象とみなし、NNを用いたセルフチューニングPID型NC、並列型NCおよび直列型NCの設計を行い、シミュレーションおよび実験によりこれらの手法の有効性を検証する。

5. セルフチューニングPID型NC⁽¹⁾

試行錯誤を通して得られた実機運転に対するPIDゲインによっても、かなり良好な制御は可能である。

しかしこの場合、試行錯誤に要する時間と労力は多大なものであり、実機の運転中にこの値をオンラインで選定することは困難である。

そこで、図9に示すようなNNを用いた制御系を構成し、PIDゲインの適切な値へのチューニングを自動的に行うとともに、その制御性能の改善を図る。

制御系は制御対象とこれに直列に接続したPIDコントローラとから構成されている。更に、PIDコントローラは、PIDゲインを出力するNN(以後、NNCと記述)とそれらのゲインを用いて制御入力を演算する要素とからなっている。また、EV試験装置シミュレータは前章で構成した回帰式とNNMとからなっている。

〈5・1〉PIDコントローラ⁽⁷⁾ 図9に示すPIDコントローラは、従来のPIDゲインが一定の場合の計

図9 セルフチューニングPID型NC
Fig. 9. Self tuning PID type NC.

算式から次式となる。

$$\begin{aligned} u(k) = & u(k-1) + (K_P(k) + K_I(k) \\ & + K_D(k))E(k) \\ & - \{K_P(k) + 2K_D(k)\}E(k-1) \\ & + K_D(k)E(k-2) \dots\dots\dots (6) \end{aligned}$$

ただし、

$$\begin{aligned} K_P(k) = & C_1 O_P(k), \quad K_I(k) = C_1 O_I(k), \\ K_D(k) = & C_1 O_D(k) \dots\dots\dots (7) \end{aligned}$$

$$E(k) = r(k) - y(k) \dots\dots\dots (8)$$

ここで、 $u(k)$:制御入力、 $K_P(k)$ 、 $K_I(k)$ 、 $K_D(k)$:比例、積分、微分の各ゲイン、 $O_P(k)$ 、 $O_I(k)$ 、 $O_D(k)$:NNCの出力、 C_1 :スケール調整定数、 $r(k)$:トルク設定値、 $y(k)$:トルク出力

〈5・2〉PID型ニューロコントローラの設計

図9のNNCの構造は3層の階層構造とし、本論文では、入力を $r(k)$ としたので入力層のユニット数は1、中間層のユニット数は3とする。出力層はPIDゲインを出力するため、ユニット数は3である。NNCの学習はバックプロパゲーション法により行い、次式に示す評価関数を最小化するものとする。

$$J = e_m(k)^2 / 2 \dots\dots\dots (9)$$

$$e_m(k) = C_2 \{r(k-1) - y(k)\} \dots\dots\dots (10)$$

ここで、 C_2 :スケール調整定数

この場合、NNCの重みを修正するために用いる評価関数 J の重みに関するこう配は以下ようになる。

$$\begin{aligned} \partial J / \partial W_{ij}^l(k-1) &= e_m(k) \cdot \partial e_m(k) / \partial W_{ij}^l(k-1) \\ &= -C_2 e_m(k) \cdot \partial y(k) / \partial W_{ij}^l(k-1) \dots\dots\dots (11) \end{aligned}$$

ここで、 $W_{ij}^l(k-1)$ は時刻 $k-1$ における $l-1$ 層 j 番ユニットと l 層 i 番ユニットとの結合の重みを表す。

$$\begin{aligned} \partial y(k) / \partial W_{ij}^l(k-1) &= \partial y(k) / \partial u(k-1) \partial u(k-1) / \partial W_{ij}^l(k-1) \\ &\dots\dots\dots (12) \end{aligned}$$

(12)式の第1項 $\partial y(k) / \partial u(k-1)$ は、出力の入力に対するこう配として、回帰式の微分およびNNMのバックプロパゲーション法により求めることができる。

(12)式の第2項 $\partial u(k-1) / \partial W_{ij}^l(k-1)$ は次式で求められる。

$$\begin{aligned} \partial u(k-1) / \partial W_{ij}^l(k-1) &= \partial u(k-1) / \partial K(k-1) \partial K(k-1) / \partial W_{ij}^l(k-1) \dots\dots\dots (13) \end{aligned}$$

表 3 NNC 学習パラメータ
Table 3. Learning parameter of NNC.

学習率 (ε)	0.001~0.05
慣性項定数 (α)	0.9
振動項定数 (β)	-0.1

ここで、 $\mathbf{K}(k-1)$ は NNC の出力を表し、 $\mathbf{K}(k-1) = \{K_P(k-1), K_I(k-1), K_D(k-1)\}^t$ である。第 2 項

$$\begin{aligned} & \partial \mathbf{K}(k-1) / \partial W_{ij}^l(k-1) \\ &= \{\partial K_P(k-1) / \partial W_{ij}^l(k-1), \\ & \quad \partial K_I(k-1) / \partial W_{ij}^l(k-1), \\ & \quad \partial K_D(k-1) / \partial W_{ij}^l(k-1)\}^t \quad \dots\dots\dots (14) \end{aligned}$$

は NNC に関するバックプロパゲーション法により求めることができる。(13)式の第 1 項 $\partial u(k-1) / \partial \mathbf{K}(k-1) = \{u_{d1}(K-1), u_{d2}(K-1), u_{d3}(K-1)\}$ の各成分は(6)式を用いて以下のように計算される。

$$\begin{aligned} u_{d1}(k-1) &= \partial u(k-1) / \partial K_P(k-1) \\ &= E(k-1) - E(k-2) \quad \dots\dots\dots (15) \end{aligned}$$

$$\begin{aligned} u_{d2}(k-1) &= \partial u(k-1) / \partial K_I(k-1) = E(k-1) \\ &\quad \dots\dots\dots (16) \end{aligned}$$

$$\begin{aligned} u_{d3}(k-1) &= \partial u(k-1) / \partial K_D(k-1) \\ &= E(k-1) - 2E(k-1) + E(k-3) \\ &\quad \dots\dots\dots (17) \end{aligned}$$

以上の(10)~(17)式から、評価関数の NNC の重みに関するこう配は次式のように整理される。

$$\begin{aligned} & \partial J / \partial W_{ij}^l(k-1) \\ &= -C_2 \{r(k-1) - y(k)\} \partial y(k) / \\ & \quad \partial u(k-1) \{u_{d1}(k-1) \partial K_P(k-1) / \\ & \quad \partial W_{ij}^l(k-1) + u_{d2}(K-1) \partial K_I(k-1) / \\ & \quad \partial W_{ij}^l(k-1) + u_{d3}(K-1) \partial K_D(k-1) / \\ & \quad \partial W_{ij}^l(k-1)\} \quad \dots\dots\dots (18) \end{aligned}$$

この $\partial J / \partial W_{ij}^l(k-1)$ を用いて重みの修正は次式により行う。

$$\begin{aligned} \Delta W_{ij}^l(k-1) &= -\varepsilon \cdot \partial J / \partial W_{ij}^l(k-1) \\ & \quad + \alpha \Delta W_{ij}^l(k-2) + \beta \Delta W_{ij}^l(k-3) \\ & \quad \dots\dots\dots (19) \end{aligned}$$

$$W_{ij}^l(k) = W_{ij}^l(k-1) + \Delta W_{ij}^l(k-1) \quad \dots\dots (20)$$

ここで、 ε は学習率、 α は慣性項定数、 β は振動項定数であり、本論文では表 3 に示す値を用いた。

6. 並列型 NC および直列型 NC

<6・1> 並列型 NC⁽¹⁾ 並列型 NC の構成を図 10 に示す。この場合、従来の制御は PID 制御を用いており、制御入力 $u(k)$ は次式により求められる。

図 10 並列型 NC
Fig. 10. Parallel type NC.

図 11 直列型 NC
Fig. 11. Series type NC.

$$u(k) = C_{1P} O_P(k) + u_{PID}(k) \quad \dots\dots\dots (21)$$

$$\begin{aligned} u_{PID}(k) &= u_{PID}(k-1) + (K_P + K_I + K_D) E(k) \\ & \quad - (K_P + 2K_D) E(k-1) + K_D E(k-2) \\ & \quad \dots\dots\dots (22) \end{aligned}$$

ここで、 $O_P(k)$: NNC の出力、 C_{1P} : スケール調整定数、 K_P, K_I, K_D : 固定された比例、積分、微分の各ゲイン

また、 $E(k)$ は(8)式と同じ制御偏差である。なお、(4)式の間層出力関数は、試行の結果そのダイナミックゲインを 5 とした。

<6・2> 直列型 NC⁽¹⁾ 直列型 NC の構成を図 11 に示す。この場合制御入力 $u(k)$ は次式により求める。

$$u(k) = C_{1s} O_s(k) \quad \dots\dots\dots (23)$$

ここで、 $O_s(k)$: NNC の出力、 C_{1s} : スケール調整定数

中間層出力関数は並列型の場合と同様のものを用いる。

7. シミュレーションによる検討

<7・1> シミュレーションの条件 ここでは、第 3 章で作成したモデルを制御対象とし、コントローラの学習を行いながらシミュレーションを実施した。シミュレーションで用いた各コントローラの構造および

表 4 コントローラの構造
Table 4. Structure of controller.

		セルフチューニング	並列型	直列型
		PID 型		
NN	入力	1	10	10
ユニット	中間	3	10	10
ト数	出力	3	1	1
PID 定数		—	$K_P=0.03$ $K_I=0.66$ $K_D=0.09$	—
スケール調整定数		$C_1=0.1$	$C_{1P}=1$	$C_{1S}=1$
		$C_2=0.1$		
入力ユニット数と符号		1: $r(k)$	3: $r(k)-r(k-2)$ 3: $u(k)-u(k-2)$ 3: $y(k)-y(k-2)$ 1: 目標値 $R(k)$	

表 5 並列型・直列型 NC 学習パラメータ
Table 5. Learning parameter of parallel and series type NNC.

学習率 (ϵ)	0.0001-0.01
慣性項定数 (α)	0.9
振動項定数 (β)	-0.1

(a) 初期

(b) 終期

図 12 セルフチューニング PID 型 NC 学習結果 (シミュレーション)

Fig. 12. Training results of self tuning PID type NC by model.

(a) 初期

(b) 終期

図 13 セルフチューニング PID 型 NC 学習による PID ゲイン (シミュレーション)

Fig. 13. PID gains by self tuning PID type NC (simulation).

定数は、表 4 に示すとおりである。構造はシンプルなものを基本とし、目的とする制御性能が達成できるようなユニット数を設定した。スケール調整定数も、コントローラの構造と誤差の収束状況を勘案して試行錯誤的に決定した。

各コントローラの学習は誤差の減少がほぼ飽和傾向を示すまで実施した。また、この学習で用いた重み修正時のパラメータの値は、表 5 に示すとおりである。

〈7・2〉 セルフチューニング PID 型 NC による制御 図 12(a), (b) はセルフチューニング PID 型 NC の学習結果を示す。(a) 図は学習の初期、(b) 図は学習終了時の結果を示している。誤差は、学習が進むにつれて減少していることがわかる。また、この学習過程における NNC の出力、すなわち PID ゲインの変化を図 13(a), (b) に示す。各ゲインとも学習の進行につれて、望ましい値に漸近している様子がうかがえる。この学習の結果得られた PID ゲインは比例ゲイン $K_P=0.03$ 、積分ゲイン $K_I=0.66$ 、微分ゲイン $K_D=0.09$ であった。この K_I の値は、試験装置の実験で試行錯誤により得られた K_I の値にほぼ等しい値となっている⁽⁴⁾。

図 14 制御誤差の比較 (シミュレーション)
Fig. 14. Comparison of control error (simulation).

表 6 コントローラのタイプと誤差の比較 (シミュレーション)

Table 6. Controller types and comparison of error (simulation).

コントローラ		誤差 (Nm)		備 考
		平均	最大	
セルフチューニング PID型	可変ゲイン	0.389	1.15	
	固定ゲイン	0.400	1.14	$K_p=0.03$ $K_i=0.66$ $K_d=0.09$
並列型 NC		0.181	0.510	並列PIDゲイン:同上
直列型 NC		0.190	0.540	
従来型 I 制御		0.800	2.98	$K_p=0$ $K_i=0.7$ $K_d=0$

〈7・3〉 並列型 NC および直列型 NC による制御

並列型 NC の PID コントローラの固定ゲインは, NNC の学習によって最終的に得られた, $K_p=0.03$, $K_i=0.66$, $K_d=0.09$ とする。直列型 NC は並列型で用いた NN と同じ構造とし, 並列型 NC において PID コントローラを除いた構成となっている。

〈7・4〉 固定ゲイン PID 制御 PID 型 NC を用いて, より適正なゲインが得られることを検証するため, 従来の PID 制御の固定ゲインとして, 次の二つのケースについてシミュレーションを行う。(1) NNC によって得られたゲイン, $K_p=0.03$, $K_i=0.66$, $K_d=0.09$ および, (2) 運転経験によって得られたゲイン, $K_p=0$, $K_i=0.7$, $K_d=0$ 。

〈7・5〉 結果の比較と考察 各タイプの NC および固定ゲイン PID 制御について, 150 秒間の制御性能の比較を図 14(a)~(e) および表 6 に示す。これらの結果より, 以下のことがわかる。

まず, 平均誤差は並列型 NC および直列型 NC が最も小さく, 次いでセルフチューニング PID 型制御およびその学習結果を用いた固定ゲイン PID 制御が小さい。従来型 I 制御はこれらに比べ大きい。

次に, 最大誤差も同様に並列型 NC および直列型 NC が最も小さく, 次いでセルフチューニング PID 型が小さい。運転経験の結果によるゲインを用いた従来型 I 制御は最も大きな値となっており, 作成したモデルに対しては目標値の急変部におけるチューニングが不十分であると思われる。

制御結果からみた場合, 並列型 NC および直列型 NC が最も優れていることがわかる。一方, PID ゲインという具体的なパラメータのチューニングのためには, セルフチューニング PID 型 NC が有効であり, 並列型 NC における PID 制御の適正な固定ゲインを得るために欠かせない。直列型 NC は両者の性能を

一つの NN で実現しているものと考えられる。

8. 実機による制御結果

前章のシミュレーションで用いた NC をそれぞれ実験装置に適用して、それらの性能を確認する。

〈8・1〉 セルフチューニング PID 型 NC 図 15 (a), (b) はセルフチューニング PID 型 NC の学習の経過を示している。初期は比較のため、モデルにより得られたゲインに固定した PID 制御を行っている。その後、前章のモデルで得られた NC に切り換えて、実機での学習が実行された。学習開始の直後は、固定ゲイン PID 制御に比べ制御誤差は大きい。学習が進むにつれて減少していることがわかる。学習後の制御結果を図 16(a) および表 7 に示す。この結果得られた PID ゲインは、 $K_p=0$, $K_i=0.7$, $K_d=0$ となっており、試行錯誤により得られた値にほぼ一致している。従って、実機ではセルフチューニング PID 型の固定ゲインによる制御結果は従来型 I 制御の結果に一致し、その結果を図 16(b) および表 7 に示す。このことはセルフチューニング PID 型 NC が、PID ゲインのチューニングに対して有効であることを意味して

図 15 セルフチューニング PID 型 NC 学習結果

Fig. 15. Training results of self tuning PID type NC.

いる。

〈8・2〉 並列型 NC 前節の学習により得られた各ゲインを固定ゲインとして用いる PID 制御に、NN を並列に挿入する。この場合、NN の重みの初期値は乱数によって決定した。学習後の結果を図 16(c) および表 7 に示す。制御誤差は、図 16(a), (b) および表 7 に示すセルフチューニング PID 型

図 16 制御誤差の比較

Fig. 16. Comparison of control error.

表 7 コントローラのタイプと誤差の比較
Table 7. Controller types and comparison of error.

コントローラ		誤差(Nm)		備 考
		平均	最大	
セルフチューニング PID型	可変ゲイン	0.398	1.37	
	固定ゲイン	0.362	1.28	$K_p=0$ $K_i=0.7$ $K_d=0$
並列型 NC		0.227	0.610	並列PIDゲイン:同上
直列型 NC		0.210	0.869	
従来型 I 制御		セルフチューニングPID型: 固定ゲインに同じ		$K_p=0$ $K_i=0.7$ $K_d=0$

NC に比べ、最大値、平均値ともほぼ半減しており、並列型 NC の有効性がわかる。

〈8・3〉 直列型 NC 学習後の結果を図 16(d)および表 7 に示す。制御誤差は並列型 NC とほぼ同程度の値を示しており、直列型 NC の有効性がわかる。

9. ま と め

まず、制御対象のモデル作成について、回帰式と NN を用いる方法を提案した。次に、従来の PID 制御を NN で構成したセルフチューニング PID 型 NC を提案した。ここで得られた制御性能を更に改善するために、並列型 NC を併用する手法を提案し、更に直列型 NC についても検討した。最後に、従来型の制御方式と比較して、これらの手法の有効性をシミュレーションおよび実験により定量的に検討した。これらの結果から以下のことが示された。

- (1) 不確定要素や非線形要素を含む制御対象のモデル化に対して、NN は有効な手法である。
 - (2) 本論文で用いたセルフチューニング PID 型 NC は、PID ゲインを精度良くチューニングする機能を有している。
 - (3) 並列型 NC は、従来の固定ゲイン PID 制御と併用することにより、その制御性能を改善することが可能である。
 - (4) 直列型 NC もオンライン学習を繰返すことによって、その制御性能を改善することが可能である。
- 以上のことから、NC は従来の制御機能では実現が困難であった部分を補う手法として効果的であるものと思われる。

本研究は、四国電力(株)からの委託研究の一環で実施したものであり、関係各位の御指導に謝意を表したい。

(平成 4 年 10 月 19 日受付, 同 5 年 2 月 8 日再受付)

文 献

- (1) 大松:「ニューラルネットワークによる適応制御」, 電学誌, 112, 101 (平 4-7)
- (2) D. Psaltis, et al.: "A Multilayered Neural Network Controller", *IEEE System Magazine*, 8, No. 2, 17 (1988)
- (3) B. Barvarian: "Introduction to Neural Networks for Intelligent Control", *ibid.*, 8, No. 2, 22 (1988)
- (4) S. Matsumura, et al.: "Neural Networks for Torque Control in Electric Vehicle", ISKIT '92
- (5) 船橋:「ニューラルネットワークの capability について」, 信学技報, MBE 88-52 (昭 63)
- (6) W. Thomas Miller, III, Richard S. Sutton & J. Werbos: *Neural Networks for Control* (1990) MIT press
- (7) 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, No. 12, 769 (平 4)

松 村 茂 憲 (正員)

昭和 23 年 9 月 6 日生。47 年京都大学工学部電気工学科卒業。同年四国電力(株)入社。現在、(株)四国総合研究所および徳島大学大学院工学研究科博士後期課程において、電気自動車に関する技術の研究に従事。

大 松 繁 (正員)

昭和 21 年 12 月 16 日生。49 年大阪府立大学大学院工学研究科博士課程修了。同年徳島大学工学部情報工学科助手, 63 年同知能情報工学科教授となり、現在に至る。工学博士。適応制御、分布定数系の推定と制御、ニューラルネットなどの研究に従事。平成 3 年電気学会論文賞受賞。計測自動制御学会, SIAM, IEEE 会員。

樋 笠 博 正 (正員)

昭和 6 年 1 月 8 日生。26 年旧制新居浜工業専門学校卒業。同年四国電力(株)入社。63 年(株)四国総合研究所負荷平準化技術研究室長。第 13, 30 回オーム技術賞受賞。情報処理学会会員。