広島大学総合科学部紀要Ⅳ理系編，第24巻，11－26ページ，1998年12月

ニューラルネットワークを用いた不確定要素とむだ時間を含む大規模システムのロバスト分散制御

日浦　章博*・水上　孝一**

*広島大学工学研究科
**広島大学総合科学部情報行動基礎研究講座

Robust Decentralized Control of Large-Scale System with Delay in Interconnections and System Uncertainties Using Neural Networks

Akihiro HIURA* and Koichi MIZUKAMI**

Graduate School of Engineering, Hiroshima University, Higashi-Hiroshima 1-4-1 739-0046, Japan
***Department of Information and Behavioral Sciences, Faculty of Integrated Arts and Sciences, Hiroshima University, Higashi-Hiroshima 1-7-1 739-0046, Japan*

Abstract : In this paper, we consider the decentralized robust stabilization problem of large-scale interconnected systems with delays, by making use of the theorem given by Lee and Radvic. If the mathematical system equation is complete, we can design decentlized control gains which satisfying the condition of the theorem so that the overall interconnected system is stable by using local state feedback. However, when the mathematical system equation includes model uncertainties, it is obvious that we can not make such interconncted systems stability by this controller. Thus, we propose the decentlized robust controller combining this controller into the multilayerd neural networks (MNN) to overcome system uncertainties.

Keywords : Large-scale interconnected systems, Robust stabilization, Neural networks, System uncertainties, Time delay

1．ま　え　が　き

近年、大規模システムに対する研究が多くなされてきた[1]－[8]。大規模システムとは、一般的に状態変数の次元が非常に大きいものをいうが、この次元の大きさだけが大規模システムの特色というわけではない。すなわち、実際のシステムは、多くの場合、物理的(または生物学的、社会的)な意味をもったいくつかの部分（サブシステム）から成り立っているものである。たとえば、電力系統であれば距離的に近くかつ特性があい似た発電機がそれぞれ１グループとして振る舞い、全体の応答はこれらのグループ間の干渉の結果としてうまくとらえられる。このように、サブシステムが相互に干渉し合って全体のシステムを構成しているという点も大規模システムの重要な特色である。そのようなシステムに対して、制御系の保守、試験運転、システムの改編などを考慮すると、

1998年10月２日受付；1998年11月２日受理

サブシステムごとの独立な制御系の設計が必要になってくる。これらの点で、分散型の制御系設計法に関する研究が多く行われている[1][2]。このような制御系の設計法としてリカッチ方程式を用いて分散状態フィードバックゲインを計算して分散状態フィードバック制御を行うものが提案されている[1][2]。

また、システムの不確定性を克服するためのロバスト制御に関する研究も盛んになされている[9]−[12]。制御問題において、線形制御理論では制御対象のダイナミクスが既知の場合には有効である。しかし、制御対象のダイナミクスが未知あるいは非線形特性を含む場合には制御器の設計が困難である。そこで、多層ニューラルネットワーク (Maltilayerd Neural Networks, MNN) の制御への応用が注目されている[13][14]。MNNは人間の脳の構造をモデルにした、神経回路網状のネットワークメカニズムで、1986年に Rumelhart らによって、その学習法（誤差逆伝播法）が発表されて以来様々な分野で応用されている。制御への応用方法としては、MNNが非線形写象を学習できることを利用して、プラントの動特性を学習させたMNNを用いる方法などが報告されている。そのなかで、MNNを線形制御理論である最適レギュレータに統合し、システムの不確定要素を克服するような非線形レギュレータが提案されている[13]。

本論文では、むだ時間を含む大規模システムに対して、サブシステムに不確定要素が存在するとき、従来のリカッチ方程式を用いた分散制御の方法と新たにMNNを統合することにより、不確定要素に対してよりロバスト性の高い分散型制御系を設計することを提案する。通常、サブシステムに不確定要素が存在する大規模システム対して、ある程度不確定要素の上限が大きくなると従来法では安定化できない。しかし、そのような場合においてもMNNを統合することにより安定化させることができることを示す。したがって、本論文の手法を利用することによって、安定化制御が可能となる大規模システムのクラスを拡張することができる。

2. 問 題 設 定

不確定要素を含む N 個の内部結合サブシステム S_i,(i =1,2,$\cdots$,N)を持つ、線形時不変離散値系複合システム S を考える。

$$S_i : x_i(k+1) = A_i x_i(k) + B_i u_i(k) + \sum_{j=1}^{N} A_{ij} x_j(k - h_{ij}) + \omega(x_i(k), u_i(k)) \qquad (1)$$

$$y_i(k) = C_i x_i(k) \qquad (2)$$

ここで、$x_i \in R^{n_i}, u_i \in R^{m_i}, y_i \in R^{r_i}$は、それぞれサブシステム$S_i$の状態ベクトル、入力ベクトル、出力ベクトルを表し、

$$\sum_{i=1}^{N} n_i = n, \quad \sum_{i=1}^{N} m_i = m, \quad \sum_{i=1}^{N} r_i = r \qquad (3)$$

とする。$A_i\, B_i\, C_i\, A_{ij}$は、適当な次元の定数行列とする。$h_{ij} \in R$は内部結合間に含まれるむだ時間を表す。またシステムの係数行列A_{ij}は以下のように分解できると仮定する。

$$A_{ij} = B_i L_{ij} C_j, \quad i, j = 1, 2, \cdots, N \qquad (4)$$

さらに、システム(1)に含まれている $x_i(k), u_i(k)$ の非線形関数$\omega(x_i(k), u_i(k))$ はシステムの不確定要素を表す。

本論文の目的は、大規模な不確定要素を含むシステム (1) が漸近安定となる制御則を設計することである。従来、大規模な不確定要素を含まないノミナルなシステム (5) に対して、漸近安定となる制御則は文献 [2] で提唱されている。一方、不確定要素を含む通常の状態空間表現されたシステムに対して、ロバスト非線形レギュレータが文献 [13] で提唱されている。本論文で扱っている問題は文献 [2] と文献 [13] の両方の問題を合わせ持つ。つまり、大規模な不確定要素を含むシステム (1) に対してロバスト非線形レギュレータを構築するのが本論文の目的である。

2.1. ニューラルネットワークを用いた大規模システムの分散制御

本論文の基本的な考えは以下の通りである。文献 [2] に従ってシステムに不確定要素を含まないノミナルな以下のシステム (5)

$$S_i : x(k+1) = A_i x(k) + B_i u_i(k) + \sum_{j=1}^{N} A_{ij} x_j(k - h_{ij}), \quad i = 1, 2, \cdots, N \tag{5}$$

$$y_i(k) = C_i x_i(k) \tag{6}$$

に対して、システム (5) を安定にする分散状態フィードバックは (7) で与えられる[2]。

$$\bar{u}_i(k) = K_i x_i(k), \quad i = 1, 2, \cdots, N \tag{7}$$

ここで、K_i ,$(i=1, 2, \cdots, N)$は

$$K_i = -(I_i + B_i^T P_i B_i)^{-1} B_i^T P_i A_i \tag{8}$$

であり、P_i ,$(i=1,2,\cdots, N)$は以下の離散値系リカッチ方程式 (9) の解である。

$$P_i = A_i^T P_i A_i - A_i^T P_i B_i (I_i + B_i^T P_i B_i)^{-1} B_i^T P_i A_i + C_i^T Q_i C_i \tag{9}$$

以下の判定条件(10)を満足するようにQ_i $(i=1, 2, \cdots, N)$ を設定すればシステム(5)を安定にする分散状態フィードバックの1つに(7)を得ることができる。

$$Q_i - \sum_{j=1}^{N} N_j L_{ji}^T (I_i + B_j^T P_j B_j) L_{ji} > 0, \quad i = 1, 2, \cdots, N \tag{10}$$

ただし、I_i, $(i=1, 2, \cdots, N)$ は単位行列,Q_i $(i=1, 2, \cdots, N)$ は正定対称行列 ($Q_i = Q_j^T > 0$) であり、各サブシステムに関して集合を

$$M_i = \{j | L_{ij} \neq 0, j = 1, 2, \cdots, N\}$$

$$M_i' = \{j | L_{ji} \neq 0, j = 1, 2, \cdots, N\}$$

と定義したときM_i, M_i' の基数をそれぞれ N_i, N_i',

$$N_i = k(M_i), N_i' = k(M_i'), \quad i = 1, 2, \cdots, N$$

とする。

(注意1) 条件式(10)を満たすようにQ_iが設定できれば(7)式の分散状態フィードバックを入力してむだ時間を含む複合閉ループシステムを安定化できる。しかし、システムの状態方程式(1)のように不確定要素を含んでいる場合、(7)をフィードバックした制御系では安定化できない場合がある。
(注意2) (8)の分散フィードバックK_iと(9)のリカッチ方程式の形は、従来からある離散値系最適レギュレータ問題と比較してみると、この制御系は

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k), \quad i = 1, 2, \cdots, N \tag{11}$$

のもとで、評価関数

$$J_i = \sum_{k=0}^{\infty} (x_i^T(k) C_i^T Q_i C_i x_i(k) + u_i^T(k) I_i u_i(k)), \quad i = 1, 2, \cdots, N \tag{12}$$

を最小化するような最適レギュレータと一致している。したがって、容易に分散状態フィードバックK_iを得ることができる。

まず、グローバルな制御則として(7)式の分散フィードバックK_iを用意する。次に、不確定要素による不安定性を克服するためにニューラルネットの補正入力

$$\hat{u}_i(k) = \delta u_i(k), \quad i = 1, 2, \cdots, N \tag{13}$$

を準備する[13]。したがって、先に紹介した分散状態フィードバック(7)とニューラルネットの補正入力(14)を統合した以下の制御入力(14)

$$u_i(k) = \bar{u}_i(k) + \hat{u}_i(k) = K_i x_i(k) + \delta u_i(k), \quad i = 1, 2, \cdots, N \tag{14}$$

を利用することによってシステム全体(1)を安定化する。

2.2. ニューラルネットワークの構成

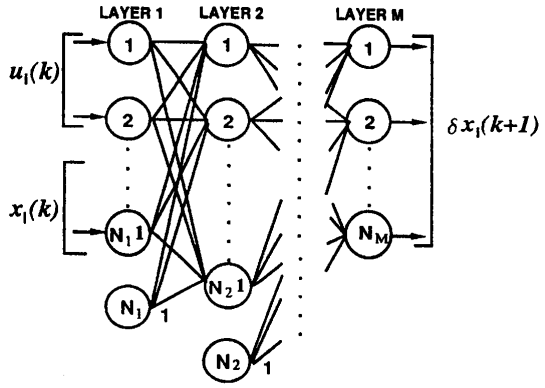

図1 モデリングのためのMNN(MNNA$_i$)
Fig. 1 MNN for Modeling (MNNA$_i$)

MNNは、サブシステムごとに、モデリング部と制御部でそれぞれ用いられる。モデリング部と制御部のMNNをそれぞれMNNA$_i$,MNNB$_i$, (i=1,2,$\cdots$, N) とする。次節ではモデリング部と制御部のMNNの構成について説明する。

2.2.1 モデリング部

まず $\bar{x}_i(k+1)$ を

$$\bar{x}_i(k+1) = A_i x_i(k) + B_i u_i(k) + \sum_{j=1}^{N} A_{ij} x_j(k-h_{ij}), \quad i = 1,2,\cdots,N \tag{15}$$

と定義する。しかし、このモデルは非線形部分を除いて考えているため$\bar{x}_i(k+1)$と実際の制御対象の状態 $x_i(k+1)$ は、完全には一致しない。そこで各サブシステムごとに、その不確定要素を補償するために、式(15)に n_i+m_i+1次元ベクトル $(u_i(k), x_i(k), 1)$ を入力、n_i次元ベクトル $\delta\ \bar{x}_i(k+1)$ を出力とするMNNA$_i$ (i=1, 2,$\cdots$, N)(図1)を並列に接続する。

つまり、MNNA$_i$ の非線形マッピングを

$$\delta x_i(k+1) \triangleq g_i(u_i(k), x_i(k)) \tag{16}$$

と表すと、制御対象の推定値 $\hat{x}_i(k+1)$ は,

$$\hat{x}_i(k+1) = \bar{x}_i(k+1) + g_i(u_i(k), x_i(k)) \tag{17}$$

である。そして、$\hat{x}_i(k+1)$ と $x_i(k+1)$ が一致するように、つまり評価関数

$$E_i^M(k+1) = \frac{1}{2}(x_i(k+1) - \hat{x}_i(k+1))^T (x_i(k+1) - \hat{x}_i(k+1)) \tag{18}$$

を最小にするようにMNNA$_i$を学習させる。MNNA$_i$のリンクの重み a_{ij}^n は、以下の式(19)で更新される。

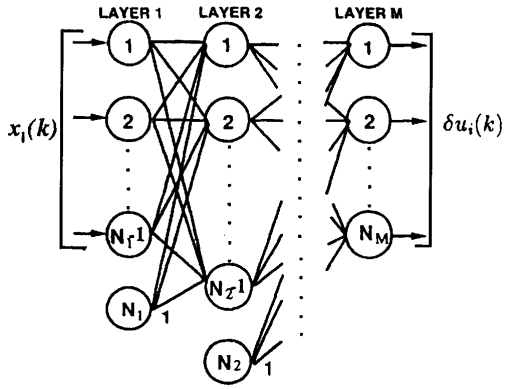

図2　制御のためのMNN (MNNB$_i$)

Fig. 2　MNN for Control (MNNB$_i$)

$$a_{ij}^n(k+1) = a_{ij}^n(k) - \mu_M \frac{\partial E_i^M(k+1)}{\partial a_{ij}^n} \tag{19}$$

ここで、

$$\begin{aligned}\frac{\partial E_i^M(k+1)}{\partial a_{ij}^n} &= -(x_i(k+1) - \hat{x}_i(k+1))^T \frac{\partial \hat{x}_i(k+1)}{\partial a_{ij}^n} \\ &= -(x_i(k+1) - \hat{x}_i(k+1))^T \frac{\partial}{\partial a_{ij}^n}(\bar{x}_i(k+1) + g_i(u_i(k), x_i(k)) \\ &= -(x_i(k+1) - \hat{x}_i(k+1))^T \frac{\partial g_i(u_i(k), x_i(k))}{\partial a_{ij}^n}\end{aligned} \tag{20}$$

である。

2.2.2 制御部

各サブシステムについてシステム方程式(15)の係数行列から条件(10)を満たすQ_iが設定できたとする。このQ_iを用いて(15)の分散制御ゲインK_iを計算する。

ところが(15)式は、実際の制御対象を完全に記述してはいないので、この制御系では、実際のプラントを安定化できないことが考えられる。そこで各サブシステムごとに、n_i+1次元ベクトル$(u_i(k),1)$を入力、m_i次元ベクトル$\delta u_i(k)$を出力とするMNNB$_i$(i=1,2,$\cdots$, N) (図2)を並列接続する。つまり、MNNB$_i$の非線形マッピングを、

$$\delta u_i(k) \triangleq h(x_i(k)) \tag{21}$$

と表すと、制御入力$u_i(k)$は、

$$u_i(k) = \bar{u}_i(k) + h(x_i(k)) \tag{22}$$

である。このMNNB$_i$は、より最適な制御入力を発生させるために(12)式の評価関数を最小化するように学習される。しかし、直接(12)式を最小化するのは難しいので以下の補題を利用する[13]。

[補題] 線形システム方程式

$$x(k+1) = Ax(k) + Bu(k) \tag{23}$$

によって$x(k)$が与えられ$x_i(k+1)$が一般化されているとする。そのとき

$$\frac{1}{2}(x^T(k+1)Px(k+1) + u^T(k)Ru(k)) \tag{24}$$

を最小化する制御入力$u(k)$は、

$$u(k) = Kx(k) = -(R + B^TPB)^{-1}B^TPAx(k) \tag{25}$$

で与えられる。つまり、式(24)を最小化することと

$$J = \sum_{k=0}^{\infty}(x^T(k)Qx(k) + u^T(k)Ru(k)) \tag{26}$$

を最小化することは等価である。

この補題では、状態の未来値 $x_i(k+1)$ が必要となるので、各サブシステムごとにMNNA$_i$でモデリングした $\hat{x}_i(k+1)$ を利用する。(12)式を最小化するために実際にはMNNB$_i$は次の評価関数(27)を最小化するように学習される。

$$E_i^C(k) = \frac{1}{2}(\hat{x}_i^T(k+1)P_i\hat{x}_i^T(k+1) + u_i(k)^T I_i u_i(k)) \tag{27}$$

そして、次の勾配法でリンクの重み b_{ij}^n を更新する。

$$b_{ij}^n(k+1) = b_{ij}^n(k) - \mu_C \frac{\partial E_i^C(k)}{\partial b_{ij}^n} \tag{28}$$

ここで、

$$\begin{aligned}
\frac{\partial E_i^C(k)}{\partial b_{ij}^n} &= \hat{x}_i^T(k+1)P_i\frac{\hat{x}_i(k+1)}{\partial b_{ij}^n} + u_i(k)^T I_i\frac{\partial u_i(k)}{\partial b_{ij}^n} \\
&= \hat{x}_i^T(k+1)P_i\frac{\partial}{\partial b_{ij}^n}[A_i x_i(k) + B_i u_i(k) + \sum_{j=1}^{N} x_j(k-h_{ij}) \\
&\quad + g_i(x_i(k), u_i(k))] + u_i^T I_i\frac{\partial u_i(k)}{\partial b_{ij}^n} \\
&= \hat{x}_i^T P_i[B_i\frac{\partial u_i(k)}{\partial b_{ij}^n} + \frac{\partial g_i(u_i(k), x_i(k))}{\partial u_i(k)} \cdot \frac{\partial u_{ij}(k)}{\partial b_{ij}^n} \\
&\quad + \frac{\partial g_i(u_i(k), x_i(k))}{\partial x_i(k)} \cdot \frac{\partial x_i(k)}{\partial b_{ij}^n}] + u_i^T(k) I_i\frac{\partial u_i(k)}{\partial b_{ij}^n} \\
&= [\hat{x}_i^T P_i(B_i + \frac{\partial g_i(u_i(k), x_i(k))}{\partial u_i(k)}) + u_i^T(k) I_i]\frac{\partial u_i(k)}{\partial b_{ij}^n}
\end{aligned} \tag{29}$$

である。

2.3. ニューラルネットワークを統合する制御系の構成

サブシステムが2つの場合の全体のデータの流れを図3に示す。但し、"MNNA$_1$、MNNA$_2$"、"MNNB$_1$、MNNB$_2$"はそれぞれ第1、2サブシステムのモデリングのためのニューラルネット、制御のためのニューラルネットを表す。Z^{-k}は k 単位時間の遅延を表す。
以下に学習のためのアルゴリズムを示す。
step1 : モデリング部のMNNの重みによる評価関数の偏微分を計算する。

$$\frac{\partial E_i^M(k+1)}{\partial a_{ij}^n} = -(x_i(k+1) - \hat{x}_i(k+1))^T\frac{\partial \hat{x}_i(k+1)}{\partial a_{ij}^n}$$

step2 : モデリング部のMNNの学習を行なう。

$$a_{ij}^n(k+1) = a_{ij}^n(k) - \mu_M\frac{\partial E_i^M(k+1)}{\partial a_{ij}^n}$$

step3 : $\bar{u}_i(k)$ を設計する。

$$\bar{u}_i(k) = K_i x_i(k)$$

step4 : 制御部のMNNを用いて $\bar{u}_i(k)$ を補正し、$u_i(k)$ を設計する。

$$u_i(k) = \bar{u}_i(k) + h(x_i(k))$$

step5 : $\bar{x}_i(k+1)$ を設計する。

$$\bar{x}_i(k+1) = A_i x_i(k) + B_i u_i(k) + \sum_{j=1}^{N} A_{ij} x_j(k - h_{ij})$$

step6 : モデリング部のMNNを用いて $\bar{x}_i$ を補正し、$\hat{x}_i(k+1)$ を設計する。

$$\hat{x}_i(k+1) = \bar{x}_i(k+1) + g(u_i(k), x_i(k))$$

step7 : 制御部のMNNの重みによる評価関数の偏微分を計算する。

$$\frac{\partial E_i^C(k)}{\partial b_{ij}^n} = [\hat{x}_i^T(k+1) P_i (B_i + \frac{\partial g(u_i(k), x_i(k))}{\partial u_i(k)}) + u_i^T(k) I_i] \frac{\partial h(x_i(k))}{\partial b_{ij}^n}$$

step8 : 制御部のMNNの学習を行なう。

$$b_{ij}^n(k+1) = b_{ij}^n(k) - \mu_C \frac{\partial E_i^C(k)}{\partial b_{ij}^n}$$

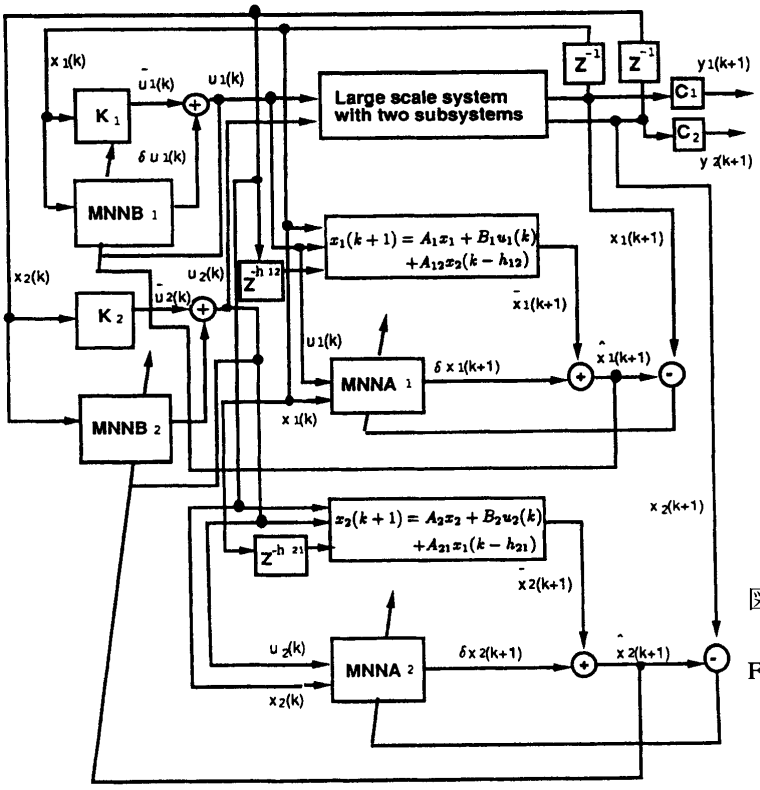

図3 MNNを用いた分散制御系のブロック図

Fig.3 Block Diagram of Decentralized controller using MNN

3．シミュレーション

次のようなむだ時間と不確定要素が存在する2つのサブシステムを持つシステム $P(\varepsilon_1, \varepsilon_2)$を考える。

サブシステム1

$$\begin{aligned}\begin{bmatrix} x_{11}(k+1) \\ x_{12}(k+1) \end{bmatrix} &= \begin{bmatrix} 0.0 & 1.0 \\ -0.01 & -0.02 \end{bmatrix}\begin{bmatrix} x_{11}(k) \\ x_{12}(k) \end{bmatrix} + \begin{bmatrix} 0.0 \\ 1.0 \end{bmatrix} u_1(k) \\ &\quad + \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.5 & 0.5 & 0.5 \end{bmatrix}\begin{bmatrix} x_{21}(k-1) \\ x_{22}(k-1) \\ x_{23}(k-1) \end{bmatrix} \\ &\quad + \varepsilon_1 \begin{bmatrix} \sin(2x_{11}(k)) + \sin(u_1(k)) \\ \sin(2x_{12}(k)) + \sin(u_1(k)) \end{bmatrix} \end{aligned} \tag{30}$$

$$y_1(k) = \begin{bmatrix} 1 & 1 \end{bmatrix}\begin{bmatrix} x_{11}(k) \\ x_{12}(k) \end{bmatrix} \tag{31}$$

サブシステム2

$$\begin{aligned}\begin{bmatrix} x_{21}(k+1) \\ x_{22}(k+1) \\ x_{23}(k+1) \end{bmatrix} &= \begin{bmatrix} 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \\ -0.03 & -0.02 & -0.01 \end{bmatrix}\begin{bmatrix} x_{21}(k) \\ x_{22}(k) \\ x_{23}(k) \end{bmatrix} + \begin{bmatrix} 0.0 \\ 0.0 \\ 1.0 \end{bmatrix} u_2(k) \\ &\quad + \begin{bmatrix} 0.0 & 0.0 \\ 0.0 & 0.0 \\ 0.5 & 0.5 \end{bmatrix}\begin{bmatrix} x_{11}(k-1) \\ x_{12}(k-1) \end{bmatrix} + \varepsilon_2 \begin{bmatrix} \sin(2x_{21}(k)) + \sin(u_2(k)) \\ \sin(2x_{22}(k)) + \sin(u_2(k)) \\ \sin(2x_{23}(k)) + \sin(u_2(k)) \end{bmatrix} \end{aligned} \tag{32}$$

$$y_2(k) = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}\begin{bmatrix} x_{21}(k) \\ x_{22}(k) \end{bmatrix} \tag{33}$$

ここで、ε_1, ε_2が掛かっている項が制御設計時に知られていなかった不確定要素を表す。まず、$[A_1, B_1]$、$[A_2, B_2]$、$[A_1, C_1]$、$[A_2, C_2]$は可制御、可観測である。このとき線形部分において、

$$L = \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix} = \begin{bmatrix} 0.0 & 0.5 \\ 0.5 & 0.0 \end{bmatrix} \tag{34}$$

とすると基数N_i, (i=1, 2)は、

$$N_1 = N_2 = 1 \tag{35}$$

となる。またA_{12}, A_{21}は(4)式

$$A_{12} = B_1 L_{12} C_2, \quad A_{21} = B_2 L_{21} C_1 \tag{36}$$

を満たしている。ここで

$$Q_1 = Q_2 = 1.0 \tag{37}$$

として(9)式のリカッチ方程式を解くと、

$$P_1 = \begin{bmatrix} 1.000 & 0.9963 \\ 0.9963 & 1.604 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 1.001 & 0.9902 & 0.9849 \\ 0.9902 & 1.651 & 1.474 \\ 0.9849 & 1.474 & 1.889 \end{bmatrix} \tag{38}$$

となる。不等式(10)の左辺に代入してみると

$$\begin{aligned} Q_1 &- \sum_{j=1}^{2} N_j L_{j1}^T (I_1 + B_j^T P_j B_j) L_{j1} \\ &= 1.0 - 1 \times 0.5(1.0 + \begin{bmatrix} 0.0 & 0.0 & 1.0 \end{bmatrix} \begin{bmatrix} 1.001 & 0.9902 & 0.9849 \\ 0.9902 & 1.651 & 1.474 \\ 0.9849 & 1.474 & 1.889 \end{bmatrix} \begin{bmatrix} 0.0 \\ 0.0 \\ 1.0 \end{bmatrix})0.5 \\ &= 1.0 - 0.722250 > 0 \end{aligned} \tag{39}$$

$$\begin{aligned} Q_2 &- \sum_{j=1}^{2} N_j L_{j2}^T (I_1 + B_j^T P_j B_j) L_{j2} \\ &= 1.0 - 1 \times 0.7 \times (1.0 + \begin{bmatrix} 0.0 & 1.0 \end{bmatrix} \begin{bmatrix} 1.000 & 0.9963 \\ 0.9963 & 1.604 \end{bmatrix} \begin{bmatrix} 0.0 \\ 1.0 \end{bmatrix})0.5 \\ &= 1.0 - 0.651000 > 0 \end{aligned} \tag{40}$$

となって、このQ_1、Q_2は、不確定要素部分を除いては不等式(10)を満たしている。(8)式より分散制御ゲインK_1、K_2は、

$$K_1 = \begin{bmatrix} 0.00616 & -0.3703 \end{bmatrix} \tag{41}$$

$$K_2 = \begin{bmatrix} 0.01962 & -0.3278 & -0.5037 \end{bmatrix} \tag{42}$$

と計算される。

まず, P(0.0, 0.0) についてシミュレーションを行った。この場合は, 不等式 (10) を満たす線形システムなので理論的に証明されている通りに分散安定化することができた。k=1 から k=200 までの200ステップの各状態変数の応答を図4に示す。ただし, 各状態変数の初期値は1.0とした。次に P (0.1, 0,1) について実験を行った。この場合でも、この線形制御系が本来もつロバスト性のみで安定化することができた。結果を図5に示す。次に非線形項の大きさが P (0.2, 0.2) に対して実験を行った。この場合では、システムは安定ではあるが状態を零点に戻すことができなかった。(図6参照)そこで、このような場合での本論文で提案するMNNを統合する分散制御系の有効性を調べるために実験を行った。ただし、MNNは三層構造とし, 各MNNの中間層のニューロン 30、また学習率は0.05とした。また, 重みの初期値は、すべて0.0とした。結果を図7に示す。図7から全ての状態変数が0に収束していることがわかる。以上の実験結果からシステムに不確定要素を含む場合, 従来の制御則 (7) では安定化できない場合でもMNNを統合することによって安定化することができた。したがって, 本論文の有用性をシュミレーションによって示すことができた。

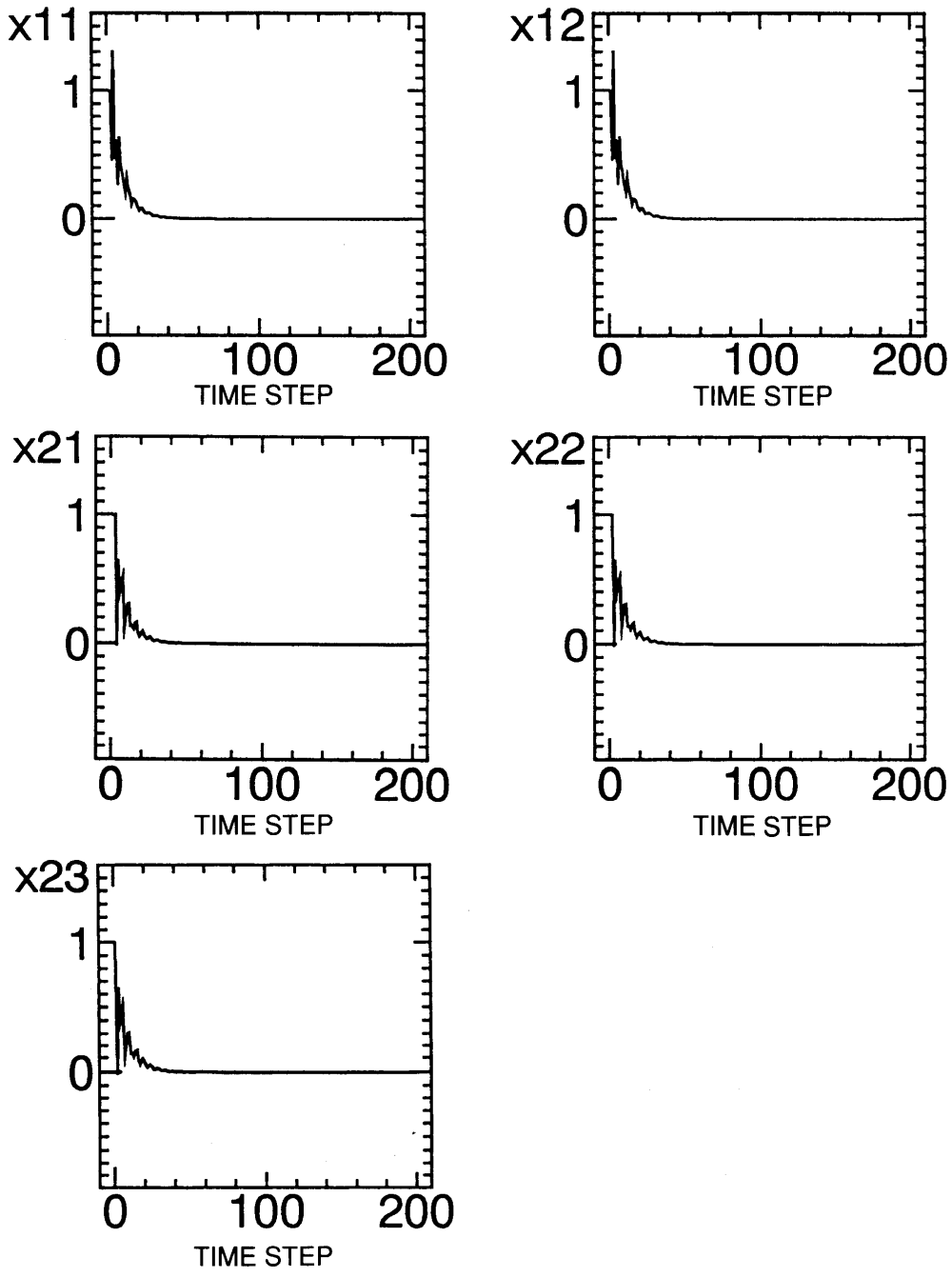

図 4　P (0.0,0.0)に線形制御系を用いた実験結果

Fig.4　Simulation results for P (0.0,0.0) by using linear controller

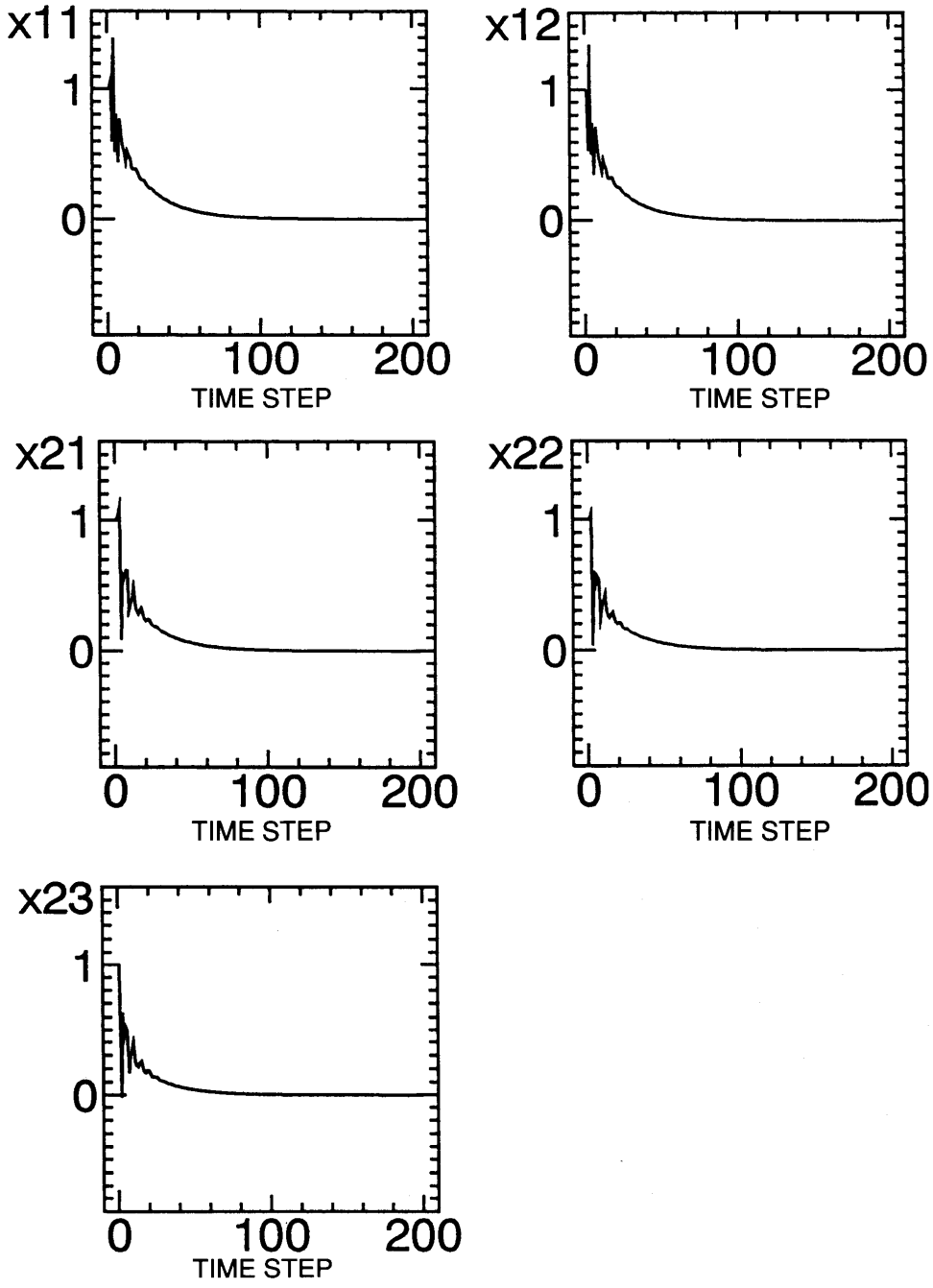

図 5　P (0.1,0.1) に線形制御系を用いた実験結果

Fig.5　Simulation results for P (0.1,0.1) by using linear controller

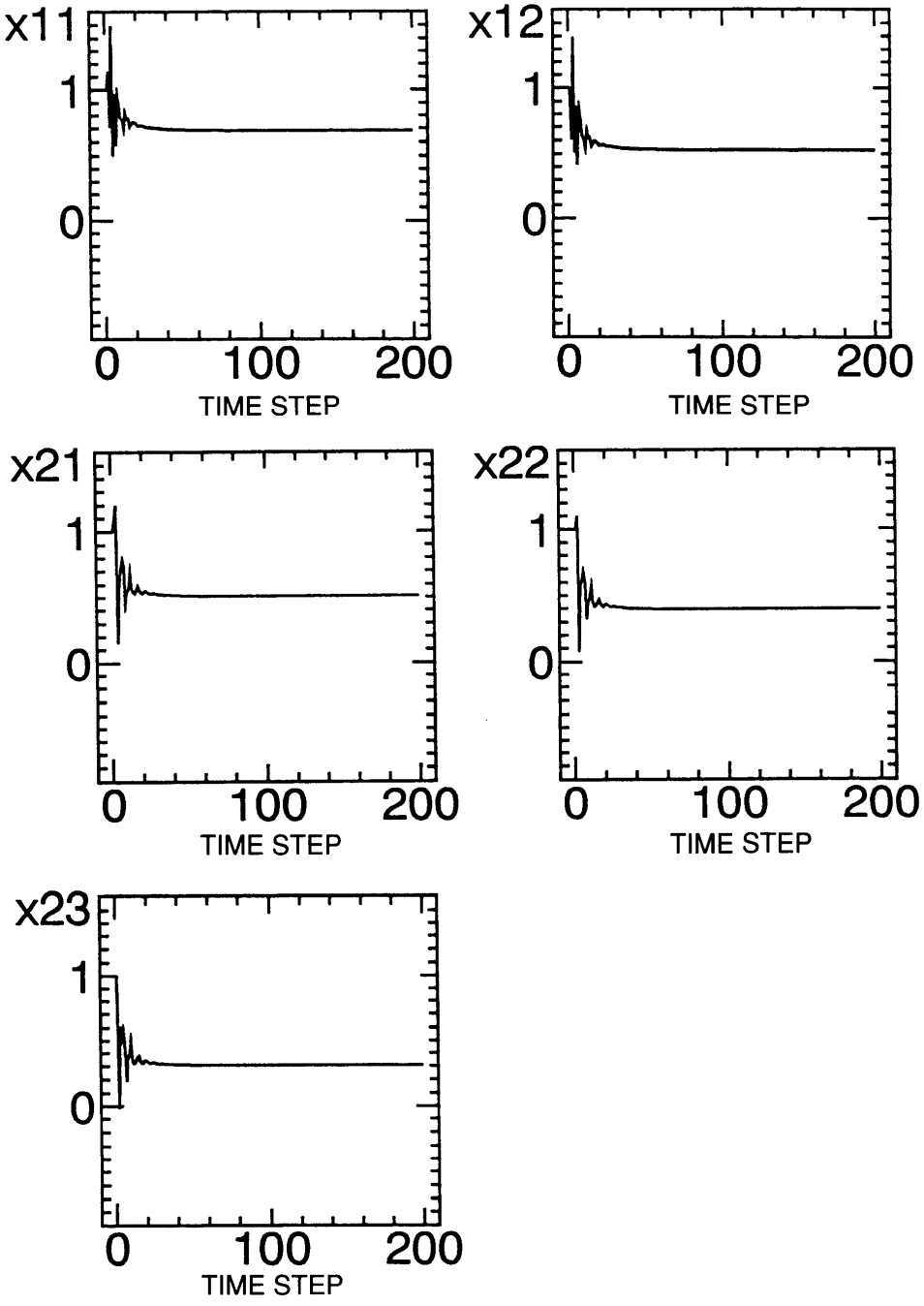

図 6　P (0.2,0.2) に線形制御系を用いた実験結果

Fig.6　Simulation results for P(0.2,0.2) by using linear controller

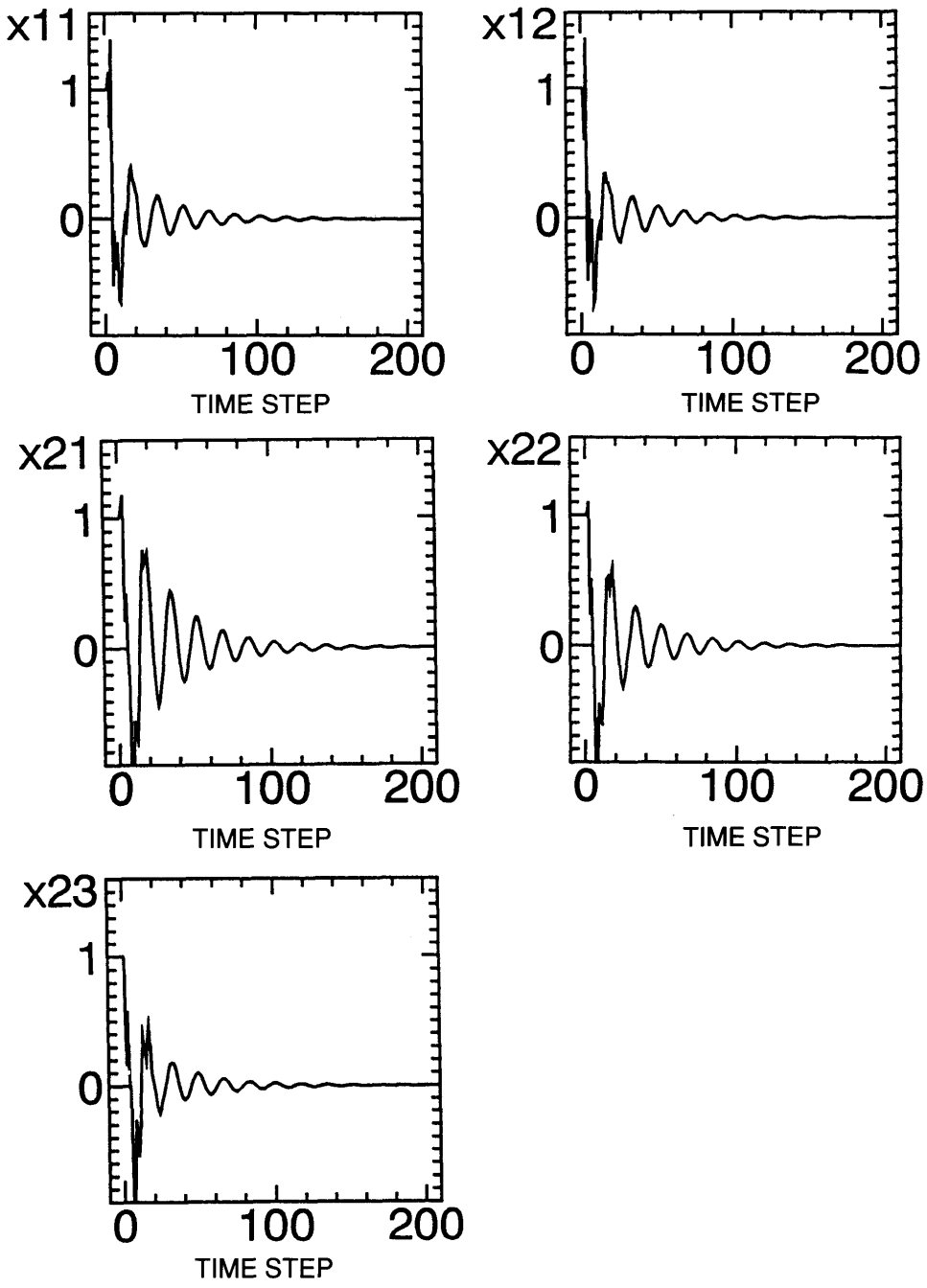

図 7　$P(0.2,0.2)$ にMNN(学習率0.05 ニューロン数30個)を統合する制御系を用いた実験結果

Fig.7　Simulation results for $P(0.2,0.2)$ by using controller with MNN (learning rate 0.05 , neuron 30)

4. むすび

本論文では、分散安定条件が最適レギュレータと一致していることに注目し、新たにニューラルネットを用いてレギュレータの性能を高める方法を統合することによって、分散制御系のロバスト性を向上させることを検討した。制御対象が不確定要素を含んでいる場合、従来の設計方法では安定化できない場合があった。そこで、本論文で提案するニューラルネットを用いた分散制御系を適用すれば安定化できることを計算機シミュレーションによって示した。

本手法では、制御対象がある程度線形化されているものを扱っており、その線形システムに多少の変動があってもそれをニューラルネットで補償するというものである。その変動が大きくなると本手法でも収束しないケースも存在することが考えられる。そこで今後の課題としては、さらに大きな変動にも対処できるようなニューラルネットの学習アルゴリズムの改善があげられる。また本手法では、各サブシステムの状態 $x_i(k)$ が観測可能として、状態フィードバックを行った。しかし、現実には出力 $y_i(k)$ しか観測できない場合が多い。今後は、分散オブザーバを用いた推定状態フィードバックに拡張することが課題になる。

参考文献

[1] J.C.Geromel and A.Yamakami : Stabilization of Continuous and Discrete Lenear Systems Subject to Control Structure Constrains, INT.J.CONTROL, Vol.63, No.3, 429/444 (1982)

[2] T.N.Lee and U.N.L.Radovic : Decentlized Stabilization of Linear Continuous and Discrete-Time Systems with Deleys in Interconnections, IEEE Transaction on Automatic Control, Vol.33, No.8, 757/761 (1988)

[3] I.H.Suh and Z.Bien : On Stabilization by Local State Feedback for Discrete-Time Large-Scale System with Delays in Intercnnections, IEEE Transaction on Automatic Control, Vol.27, No.3, 744/746 (1982)

[4] M.Ikeda and D.D. Šiljak : On Decentrally Stabilizable Large-Scale Systems, AUTOMATICA, Vol.16, 331/334 (1980)

[5] M.Ikeda, D.D. Šiljak and K.Yasuda : Optimality of Decentralized Control for Large-Scale Systems, AUTOMATICA, Vol.19, No.3, 309/316 (1983)

[6] S.J.Ho, I.R.Horng and J.H.Chou : Decentralized stabilization of large-scale systems with structured uncertainties, INT. J. SYSTEM SCI, Vol.23, No.3, 425/434 (1992)

[7] 田村 : 大規模システム、昭晃堂 (1986)

[8] D.D. Šiljak : LARGE-SCALE DYNAMIC SYSTEM : Stability and Structure, NORTH-HOLLAND SERIES IN SYSTEM SSIENCE AND ENGINEERING (1978)

[9] T.J.Su , T.S.Kuo and Y.Y.Sun : Robust stability of lenear purturbed discrete systems with saturating actuators, INT. J. SYSTEM SCI, Vol.21, No.7, 1273/1279 (1990)

[10] K.M.Sobel , S.S.Banda and H.H.Yeh : Robust control for linear systems with structureed state space uncertainty, INT. J. SYSTEM SCI, Vol.20, No.5, 1991/2004 (1989)

[11] H.Wu and K.Mizukami : Robust stabilization of uncertain linear dynamical systems, INT. J. SYSTEM SCI, Vol.24, No.2, 265/276 (1993)

[12] A.T.Neto , J.M.Dion and L.Dugard : On the Robustness of LQ Regulators for Discrete-Time Systems, IEEE Transaction on Automatic Control, Vol.38, No.10, 1564/1568 (1993)

[13] Iiguni, Sakai and Tokumaru : Nolinear Regulator Design in the Presence of System Uncertainties Using Multilayered Neural Networks, IEEE Transaction on Neural Networks, Vol.2, No.4, 410/417 (1991)

[14] K.S.Narenda and K.Parthasarathy : Identification and Control of Dynamical Systems Using Neural Networks, IEEE Transaction on Neural Networks, Vol.1, No.1, 4/27 (1990)