強化学習と小脳モデルを利用した
適応型制御システムの設計法

A Design Method
for An Adaptive Control System Based on
Reinforcement Learning and Cerebellar Model

平　成　２８　年　９　月

内　山　祥　吾

山口大学大学院理工学研究科

概要

現代制御理論は，制御対象の数式モデルを利用する設計を前提とした，システムの安定性を保証した制御方式に関するものである。しかし，現代社会に存在する多くのシステムは非線形システムであり，その時間的変化は通常，非線形の微分方程式で表わすことになる。その変化は複雑で数式（微分方程式）での表現（モデル化）が困難となる場合も多く，そのようなシステムを制御するためには，制御対象の数式モデルを必要とせずに制御器の設計が可能なモデルフリーな制御系設計法が必要となる。その一つがニューラルネットワーク(NN)やファジィ等を用いた計算手法であるソフトコンピューティングを利用した知的制御である。ソフトコンピューティングを利用した知的制御は，目的を達成するように設計パラメータを学習の繰り返しで調整しながら最適な制御系の設計を意図するもので，学習中に環境との相互作用の中で対象システムの性質を獲得していく制御方式であり，「対象システムに関する事前知識を必要としない」ことが本制御方式の大きな強みである。但し，制御系の安定性は保証されない。

そこで，近年，制御対象のモデル化が困難なシステムを対象として，現代制御理論とソフトコンピューティングの両者の利点を利用した，安定性を保証する両者の融合型制御方式の研究が盛んに行われている。

一方，人間の脳機能に関する脳科学に関する研究も進展し，特に脳の学習機能を模倣した強化学習を制御システムに展開する研究も盛んに行われてきている。強化学習も制御対象との相互作用による学習の過程で制御対象の性質を獲得する学習方式であり，制御対象の数式モデルを必要とせず，ソフトコンピューティングに含まれると言えるが，その学習の性質上，強化学習を利用して設計された制御系も安定性は保証されない。

本論文では，筆者が，以上述べた経緯に沿って研究を進め，最終的に，数式で表現できないシステムを対象として，安定性を保証し，かつシステムの変動にロバストで，可能な限り高い制御性能を持つ制御系の実現を目指して開発した設計法の開発過程をまとめたものである。以下，各章を要約する。

第 1 章では，本論文の背景と目的について述べる。

第 2 章では，ソフトコンピューティングと現代制御理論の両者の利点を利用した融合型制御として，強化学習制御システムとロバスト性が強い適応 H_∞ 制御システムの協働により性質の異なる状態を同時に制御する制御系設計法を提案する。本方法により，通常では制御が困難な制御対象を容易に制御可能としている。頭上走行クレーンを制御対象システムとした計算機シミュレーションにより，提案法の有効性を示す。

第 3 章では，第 2 章での設計方法がオフライン学習を利用しているのに対し，システムの不確定性に頑健なロバスト制御と強化学習をオンライン的に利用した，両者の融合型である，オンライン強化学習制御システム「リアルタイム強化学習制御システム(Real-time

Reinforcement Learning Control System，RRLCS)」を提案する。リヤプノフ関数による安定性解析により，システムの安定性を証明し，台車付き倒立振子の制御対象システムとした計算機シミュレーションにより提案システムの有効性を示す。

第 4 章では，第 3 章で提案した RRLCS が持つ確率推論による性能の不安定さを解決するため，小脳モデルを利用した制御系設計法について提案する。小脳モデルに関する先行研究として，Albus が提案した CMAC(Cerebellar Model Articulation Controller)と呼ばれる小脳パーセプトロンモデルがある。CMAC の利点は，ルックアップテーブル(Look Up Table，LUT)を扱う単純な構造の局所的な NN であるため，学習が速く，高い収束性を持ち，幅広い分野で用いられている。しかし，CMAC は局所的表現の LUT を用いているため，その汎化能力は学習（訓練）された行動のごく近傍に限られる。そこで，本論文では LUT を用いない，新たな小脳パーセプトロン改良モデル(Cerebellar Perceptron improved model，CP)を提案し，それを基にした「小脳パーセプトロン改良モデル利用型ロバスト制御システム(Cerebellar Perceptron Robust Control System，CPRCS)」を提案する。リヤプノフ関数による安定性解析により，システムの安定性を証明し，台車付き倒立振子による計算機シミュレーションにより，RRLCS より高い追従性能を示し，提案システムの有効性を示す。

第 5 章では，フィードバック制御である CPRCS を，一般的に応答が高速であるフィードフォワード制御に拡張するために CP を改良する。改良型 CP を基に構築した「自己融合小脳パーセプトロン改良モデル利用型ロバスト制御システム(Auto-Fusion Cerebellar Perceptron Robust Control System，AFCPRCS)を提案する。フィードバック誤差学習による学習則により最適な CP を構築するシステムであり，台車付き倒立振子を制御対象システムとした計算機シミュレーションにより，AFCPRCS が CPRCS より高い追従性能を示し，提案システムの有効性を示す。

第 6 章では，AFCPRCS をマルチエージェントシステムの合意問題に適用可能な制御システムに拡張する。多入力多出力システムで構成される 6 体のエージェントによる合意問題の計算機シミュレーションにより，高い合意性能と CP の汎化性能を示す。

第 7 章では，結論であり本論文を総括し，今後の展望を述べる。

Abstract

Modern control theory is control system which guarantees the stability of the system that assumes a design that utilizes a mathematical model of control system. However, many of the systems that exist in modern society is a non-linear system the time change will be usually be expressed by differential equations of non-linear. The change often to be complex and difficult to express in formulas. Such in order to control the system, the model-free control system design method capable of designing a controller without the need for mathematical model of the controlled system is necessary. One is the intelligent control using the soft computing is a calculation method using a neural network (NN) and fuzzy. Intelligent control using the soft computing contemplates optimal control system design while adjusting the repetition of learning design parameters to achieve the purpose. It is a control system to continue to acquire the nature of the target system in the interaction with the environment during the learning. "It does not require prior knowledge about the target systems" It is a major advantage of this control method. However, there is no guarantee the stability of the control system it is a major strength of this control scheme. However, the stability of the control system is not guaranteed.

In recent years, the study of both the converged control scheme that ensures stability to take advantage of modern control theory and Soft Computing both targeting difficult system model of the controlled system have been actively.

On the other hand, research to expand the research on brain science of human brain function also progress in particular reinforcement learning that mimics the learning function of the brain to control systems have also been actively carried out. Reinforcement learning is also a learning method to win the nature of the control target in the course of learning due to the interaction of the control object. Without the need for a mathematical model of the controlled object, it said to be included in the soft computing, but the stability of the control system designed using reinforcement learning is not guaranteed.

In this paper, research along the background described above, the author summarizes the development process of the design method, which was developed with the goal of systems with high control performance to guarantee stability and robust to changes in the system. The following summarizes each chapter.

In chapter 1, we describe the background and purpose of this paper.

In chapter 2, as a fusion-type control utilizing both the advantages of the soft computing and modern control theory, we propose reinforcement learning control

system and the robustness is strong adaptive control system designed method to control the different states of nature at the same time by the cooperation. By this method, the control is difficult to control the object you are easily controllable. Through the computer simulation of the overhead traveling crane, we show effectiveness of the proposed system.

In chapter 3, while the design method in Chapter 2 is using the offline learning, we propose online reinforcement learning control system type using robust control to uncertainty of the system and online reinforcement learning control system. This control system named Real-time Reinforcement Learning Control System (RRLCS). We show stability of the proposed system by stability analysis of Lyapunov function. Through the computer simulation for controlling an inverted pendulum system, we show the effectiveness of the proposed system.

In chapter 4, we propose a control system design method using the cerebellar model to solve the instability of the performance by the probability inference that RRLCS proposed in chapter 3. As previous research on the cerebellum model, there is a cerebellum perceptron model Albus called the proposed CMAC (Cerebellar Model Articulation Controller). The advantage of CMAC have been used in various fields having a lookup table (Look Up Table, LUT) local learning high speed convergence since the NN simple structure for handling. However, CMAC is because of using the LUT of the local representation, its generalization ability is limited to the immediate vicinity of behaviors learned. Therefore, the new cerebellar Perceptron improved model that does not use a LUT In this paper Cerebellar Perceptron improved model (CP) is proposed, and we propose Cerebellum Perceptron improved model using robust control system (Cerebellar Perceptron Robust Control System (CPRCS). We show the stability of the system by the stability analysis of Lyapunov function. Through the computer simulation for controlling an inverted pendulum system, we show the effectiveness of the proposed system than RRLCS.

In chapter 5, generally responding CPRCS a feedback control to improve the CP in order to extend the feed-forward control is fast. We propose Auto-Fusion Cerebellar Perceptron Robust Control System (AFCPRCS) based on improved CP. The control system structure optimal CP using learning rule by the feedback error learning. Through the computer simulation for controlling an inverted pendulum system, we show the effectiveness of the proposed system than CPRCS.

In chapter 6, to extend the applicable control system to consensus problem of multi-agent system AFCPRCS. Through computer simulation of the agreement issue by six bodies agent composed of multi-input multi-output system, it shows the consensus

performance of high performance agreement and the CP.

In chapter 7, summarizes the present paper be a conclusion, we describe the future prospects.

目次

第 1 章　序論　1

1.1　研究背景 1

1.2　本研究の目的 3

1.3　論文構成 5

第 2 章　強化学習制御と適応 H_∞ 制御の協働型制御方式　8

2.1　はじめに 8

2.2　提案システム 9

2.2.1　対象システム 9

2.2.2　強化学習の概要 9

2.2.3　Actor-Critic 制御システムの構成 10

2.2.4　適応 H_∞ 制御システムの構成 12

2.2.5　強化学習制御と適応 H_∞ 制御の協働型制御システムの構成 14

2.3　計算機シミュレーション 16

2.3.1　制御対象 16

2.3.2　シミュレーション条件 18

2.3.3　シミュレーション結果 19

2.3.4　ロバスト性の検証 23

2.3.5　考察 24

2.4　まとめと今後の展開 24

第 3 章　H_∞ 追従性能補償器を備えたリアルタイム強化学習制御システム　25

3.1　はじめに 25

3.2　制御対象の定式化 26

3.3　H_∞ 追従性能補償器を備えたリアルタイム強化学習システム 28

3.3.1　H_∞ 追従性能補償器 29

3.3.2　Actor の構成 30

3.3.3　Critic の構成と TD 誤差 32

3.3.4　学習アルゴリズム 34

3.4　提案システムの安定性解析 35

3.4.1　H_∞ 追従性能 35

3.4.2　安定性解析 36

3.4.3　強化学習信号の有界性 39

3.5　計算機シミュレーション 40
3.5.1　台車付き倒立振子 40
3.5.2　シミュレーション結果 41
3.6　まとめ 47

第 4 章　フィードバック制御における
小脳パーセプトロン改良モデル利用型ロバスト制御システム　48
4.1　はじめに 48
4.2　制御対象の定式化 49
4.3　提案する小脳パーセプトロン改良モデル 49
4.3.1　自己構造アルゴリズム 52
4.4　小脳パーセプトロン改良モデル利用型ロバスト制御システム 53
4.4.1　小脳パーセプトロン改良モデルの近似 54
4.4.2　安定性解析 56
4.5　計算機シミュレーション 58
4.5.1　台車付き倒立振子 58
4.5.2　シミュレーション結果 58
4.6　まとめ 70

第 5 章　フィードフォワード制御における
自己融合小脳パーセプトロン改良モデル利用型ロバスト制御システム　71
5.1　はじめに 71
5.2　フィードバック誤差学習 72
5.3　提案する小脳パーセプトロン改良モデル 74
5.3.1　自己融合アルゴリズム 75
5.4　自己融合小脳パーセプトロン改良モデル利用型ロバスト制御システム 77
5.4.1　学習アルゴリズム 78
5.5　計算機シミュレーション 79
5.5.1　台車付き倒立振子 79
5.5.2　シミュレーション結果 80
5.6　まとめ 92

第 6 章　小脳パーセプトロン改良モデルの合意問題への適用　93
6.1　はじめに 93
6.2　マルチエージェントシステム 94
6.2.1　グラフ理論 94

6.3　制御対象の定式化 95
6.4　自己融合小脳パーセプトロン改良モデル利用型制御システム 95
6.4.1　学習アルゴリズム 97
6.5　計算機シミュレーション 98
6.5.1　合意問題 98
6.5.2　シミュレーション結果 100
6.6　まとめ 112

第 7 章　結論 **113**

謝辞 **115**

参考文献 **116**

第 1 章　序章

1.1　研究背景

人間の脳の構造をモデル化し，その働きを制御などの工学的に応用する研究が広くなされており，近年の計算機の発展も手伝って，多くの成果がもたらされるようになってきた。その中で，ソフトコンピューティングを用いた知的制御は，人間の知能，生体を模倣した手法であり，人間の脳の仕組みを模倣したニューラルネットワーク(NN)[1-5]，人間のあいまいな知識を活かすファジィ理論[6-7]，人間の学習能力をコンピュータで実現する強化学習[8-12]などが挙げられる。それらは目的を達成するように設計パラメータを学習の繰り返しで調整しながら制御系を設計するもので，学習範囲外での動作の安定性の保証が得られにくいが，学習中に環境との相互作用の中で対象システムの性質を獲得していくので，対象システムに関する事前知識を必要としないことが大きな強みである。

ソフトコンピューティングを用いた知的制御の必要性は，現代制御理論における精度の良い対象のモデルが得ることの困難にある。現代制御理論の高度な制御手法として，ロバスト制御と適応制御が誕生した。ロバスト制御は，モデルを正確に得ることが困難であることを前提として，精度の悪いモデルでもそれにもとづいた設計で所定の性能を達成する。適応制御は，制御対象のモデルなどの特性変動に応じて制御系の特性をオンライン的に自動調整し，制御系としての性能を常に良好に保つような制御方式である。つまり，両制御方式は対象システムのモデルを必要とするが，安定性を保証した制御方式である。しかし，現代社会に存在する多くのシステムは非線形システムであり，その時間的・空間的変化は通常，非線形の微分方程式で表わすことになる。その変化は複雑で微分方程式でのモデル化が困難となる場合も多く，また，表現できてもそのモデルには多くの不確かさが存在する[13-16]。そのようなシステムの制御のために，制御対象の数式モデルを必要としない制御器の設計が可能なモデルフリーな制御が必要となり，その一つがソフトコンピューティングを用いた知的制御である。

近年，現代制御理論とソフトコンピューティングの両者の利点を利用した融合型制御方式の研究が盛んに行われている。適応制御とファジィ理論を融合したあいまいさのファジィルールを適応的に変化させる適応ファジィ制御(Adaptive Fuzzy Control，AFC)[17-30]，適応制御とファジィニューラルネットワークを融合した NN のニューロンを適応的に増減させる自己構造型ファジィニューラルネットーク制御(Auto-Structure Fuzzy Neural Network，ASFNN)[31-35]，ロバスト制御と強化学習を融合した環境との相互作用で制御系設計が可能なロバスト強化学習制御(Robust Reinforcement Learning，RRL)[36-43]が挙げられる。これらの制御システムはモデルの非線形性や不確かさに対応することができるオンライン制御である。RRL において，オフライン学習によるさらなる非線形性や未知構

造を持つ制御対象を制御する研究も行われている[44-49]。この研究は，AFC や ASFNN のようにモデルを単入力単出力(SISO)システムを前提としていることに対し，オフライン学習は，より複雑な単入力多出力(SIMO)システムや多入力多出力(MIMO)システムに対応させることも可能にしている。その例として，SIMO システムである頭上走行クレーン[44-45]や MIMO システムであるモバイルロボット[48]の制御も可能にしている。現代制御理論とソフトコンピューティングの両者の利点を利用した融合型制御は，SISO システムなど特定の条件を選ぶ場合やオンライン性が失われる場合があるが，高い追従性能をもちモデルフリー制御を実現できるため，非線形性の強いロボット工学において盛んに研究されている。

その融合型制御方式において，人間の小脳の機能に着目し，それをモデル化する研究も行われている。人間の小脳の機能は「身体で覚える」記憶である。このような運動技能などの記憶を手続き記憶といい，小脳の重要な機能の一つである。繰り返し動作を行うことで，次第に適切な動作を行う回路記憶が獲得される。身体で覚えた数多くの手続き記憶は，人間の脳に長期的に記憶として記銘される。そして，状況に応じた記憶を想起することで，これまで覚えた身体の動作を無意識で，かつスムーズに行うことが可能になる。これは，脳内に「内部モデル」が存在し，身体の動作や環境の変化を基に内部モデルが学習を行っていることがそれを可能にしていると考えられている。この内部モデルを操作することで高速かつ高精度な制御を実現することができる[50-53]。そのため小脳は，適応性や即応性に優れた制御システムと言える。

1.2 本研究の目的

筆者は，これまでの研究で，SIMO システムである頭上走行クレーン制御用の強化学習制御器と適応H_∞制御器の協働型制御方式を提案した[44-45]。両者が協働して制御を行い，通常では困難な制御対象に適応可能にしている。クレーンの角度や角速度などの 1 変数（変数の微分を含む）を対象とした制御系設計法では困難な，性質の異なる多様な状態遷移を計画的に行わせる，即ち，多様な計画行動をロバスト制御と強化学習の協働型制御システムで実現することを目的とする。すなわち，強化学習制御（システム）と適応H_∞制御（システム）の協働により性質の異なる状態を同時に制御する方法を提案する。

しかし，ここで用いた強化学習制御器は，通常の強化学習と同様，事前にオフラインによる多数回の繰り返し学習を必要とする。さらに，適応H_∞制御のみの制御システムに比べて，安定性が劣る欠点も有している。そこで，モデルフリーで強化学習制御器の安定性を保証し，事前にオフライン学習を必要としないリアルタイム強化学習システム(Real-time Reinforcement Learning Control System，RRLCS)を提案する[4043]。そこでは，リヤプノフ関数による安定性解析を行い，追従性能の点で，優れた制御性能を示す。

しかし，RRLCS はシステムを構成する NN の結合荷重の設定範囲を仮定したため，システムの安定性はある条件下での保証に限定される。また，SISO システムを前提としているため，SIMO システムや MIMO システムへの対応が困難である。そこで，人間の小脳の適応性や即応性に着目し，高い追従性能やオンライン性を持つ制御システムで，RRLCS で仮定した NN の結合荷重の設定範囲を制限しない完全なモデルフリーの制御システムを考える。さらに，その制御対象に SISO システムだけでなく，より複雑な MIMO システムへの対応も考慮する。そこで，運動に関しての学習や記憶を司る小脳の機能に着目する。

小脳モデルに関する先行研究として，Albus が提案し，NN の 1 種である小脳パーセプトロンモデルで知られる CMAC(Cerebellar Model Articulation Controller)[54]と呼ばれる神経回路モデルがある[54-73]。CMAC の利点は，ルックアップテーブル(Look Up Table, LUT)を扱う単純な構造の局所的な NN であるため，学習が速く，高い収束性を持ち，ハードウェアによる実行が容易であることである[53]。また，優れた汎化能力を持つため，非線形システム[54]，カオスシステム[56-63]，画像処理[64-65]，パターン認識[66]，船体運動[67]，マニピュレータ[68]，車輪倒立振子[69]，モバイルロボット[70]，プロセス制御[71]，PID チューニング[72]，航空機の自動着陸システム[73]など様々な分野に用いられている。また，Almeida らは，CMAC の特徴である LUT をそのまま用いてネットワーク入力強度を表わす線形パラメトリック式やファジィ理論を用いたパラメトリック CMAC(Parametric CMAC, PCMAC)[74]を提案している。その大きな強みは，線形項の追加で得られる CMAC 以上の高い近似能力にある。

しかし，CMAC および PCMAC は局所的表現の LUT を用いるため，その汎化能力は訓練された行動のごく近傍に限られる。また，状態数が多い場合，これに対処するため膨大

なメモリ(ニューロン数)が必要になる。そこで，本論文では LUT を用いない，新たな小脳パーセプトロン改良モデル(Cerebellar Perceptron improved model, CP)を提案する。即ち，先述した小脳の記憶の機能に着目し，その概念を PCMAC に導入することで，制御システムの汎化能力を高め，メモリ数を抑える。安定性解析においては，RRLCS で仮定した NN の結合荷重の設定範囲を制限しない完全なモデルフリーの制御システムを提案する。

フィードバック制御システムにおいて，制御対象に適切な動作を行わせるため，制御対象の状態に応じて，制御に必要な情報を想起させる，即ち，パーセプトロンにおいては必要なニューロン(メモリ)のみを連結して結合荷重を強化させる。また，対応するニューロンがない場合，制御対象の状態の情報をもとに新たに追加し，逆に，一定時間参照されないニューロンは削除する(以下，自己構造と呼ぶ)。最終的に，強化されたニューロンのみを記憶・想起させることで，制御対象に適切な動作をスムーズに行わせるネットワークを構築する。このような CP を用いる，「小脳パーセプトロン改良モデル利用型ロバスト制御システム(Cerebellar Perceptron Robust Control System，CPRCS)」[75-77]を提案する。

さらに，上記の提案システムをフィードフォワード制御システムに拡張する。フィードバック制御は 1 時刻前の状態を基に制御を行うことによる遅延があるため，近似能力の低下が考えられたため，川人の生体の運動制御に関する学習機構であるフィードバック誤差学習(Feedback Error Learning，FEL)を導入し，フィードフォワード制御を実現する。そして，時間変化する目標信号に対応するため，自己構造メカニズムの代わりに自己融合メカニズムを導入する。よりスムーズで，様々なダイナミクスに対応可能な制御システムを目指した「自己融合小脳パーセプトロン改良モデル利用可制御システム(Auto-Fusion Cerebellar Perceptron Robust Control System，AFCPRCS)」を提案する。

また，近年，制御対象システムの大規模化・複雑化に伴い，単一の制御対象よりも，複数の制御対象に対する制御方式が重要になっている。その中で，それぞれが自律的に意思決定を行い，複数のシステム(エージェント)から構成されるマルチエージェントシステム(MAS)の協調制御が注目を集めている[78-96]。MAS とは複数のエージェントがある共通の目的のために，各エージェントが互いの情報を得ることで協調して，単一では困難な課題をシステム全体で合意を達成することである[78-87]。その応用として，自律ロボット群の協調制御[88]，1 体のエージェントに他の全エージェントが追従するリーダー・フォロワー制御[89-94]，水艇・自動車などのフォーメーション制御[95-96]などが挙げられる。しかし，MAS は非線形性や未知環境との相互作用などのため，不確かなダイナミクスを持ったエージェントの制御系の設計は困難とされていた。この問題を解決し，かつ合意問題において優れた性能を示す MAS に適用可能な AFCPCS も提案する[86-87]。

CP を安定性解析が必要なフィードバック制御システム，フィードバック誤差学習におけるフィードフォワード制御システム，MAS の合意問題の制御システムに適用可能であることを示し，その有効性を広げる。また，オンライン性能や追従性能から従来システムより優れることを示す。

1.3 論文構成

第 2 章では，ソフトコンピューティングと現代制御理論の両者の利点を利用した融合型制御として, 強化学習制御システムとロバスト性が強い適応 H_∞ 制御システムの協働により性質の異なる状態を同時に制御する制御系設計法を提案する。本方法により，通常では制御が困難な制御対象を容易に制御可能としている。頭上走行クレーンを制御対象システムとした計算機シミュレーションにより，提案法の有効性を示す。

第 3 章では，第 2 章での設計方法がオフライン学習を利用しているのに対し，システムの不確定性に頑健なロバスト制御と強化学習をオンライン的に利用した，両者の融合型である，オンライン強化学習制御システム「リアルタイム強化学習制御システム(Real-time Reinforcement Learning Control System，RRLCS)」を提案する。リヤプノフ関数による安定性解析により，システムの安定性を証明し，台車付き倒立振子の制御対象システムとした計算機シミュレーションにより提案システムの有効性を示す。

第 4 章では，第 3 章で提案した RRLCS が持つ確率推論による性能の不安定さを解決するため，小脳モデルを利用した制御系設計法について提案する。小脳モデルに関する先行研究として，Albus が提案した CMAC(Cerebellar Model Articulation Controller)と呼ばれる小脳パーセプトロンモデルがある。CMAC の利点は，ルックアップテーブル(Look Up Table，LUT)を扱う単純な構造の局所的な NN であるため，学習が速く，高い収束性を持ち，幅広い分野で用いられている。しかし，CMAC は局所的表現の LUT を用いているため，その汎化能力は学習（訓練）された行動のごく近傍に限られる。そこで，本論文では LUT を用いない，新たな小脳パーセプトロン改良モデル(Cerebellar Perceptron improved model，CP)を提案し，それを基にした「小脳パーセプトロン改良モデル利用型ロバスト制御システム(Cerebellar Perceptron Robust Control System，CPRCS)」を提案する。リヤプノフ関数による安定性解析により，システムの安定性を証明し，台車付き倒立振子による計算機シミュレーションにより，RRLCS より高い追従性能を示し，提案システムの有効性を示す。

第 5 章では，フィードバック制御である CPRCS を，一般的に応答が高速であるフィードフォワード制御に拡張するために CP を改良する。改良型 CP を基に構築した「自己融合小脳パーセプトロン改良モデル利用型ロバスト制御システム(Auto-Fusion Cerebellar Perceptron Robust Control System，AFCPRCS)を提案する。フィードバック誤差学習による学習則により最適な CP を構築するシステムであり, 台車付き倒立振子を制御対象システムとした計算機シミュレーションにより，AFCPRCS が CPRCS より高い追従性能を示し，提案システムの有効性を示す。

第 6 章では，AFCPRCS をマルチエージェントシステムの合意問題に適用可能な制御システムに拡張する。多入力多出力システムで構成される 6 体のエージェントによる合意問題の計算機シミュレーションにより，高い合意性能と CP の汎化性能を示す。

第 7 章では，結論であり本論文を総括し，今後の展開を述べる。論文構成を Fig.1.1 に示す。

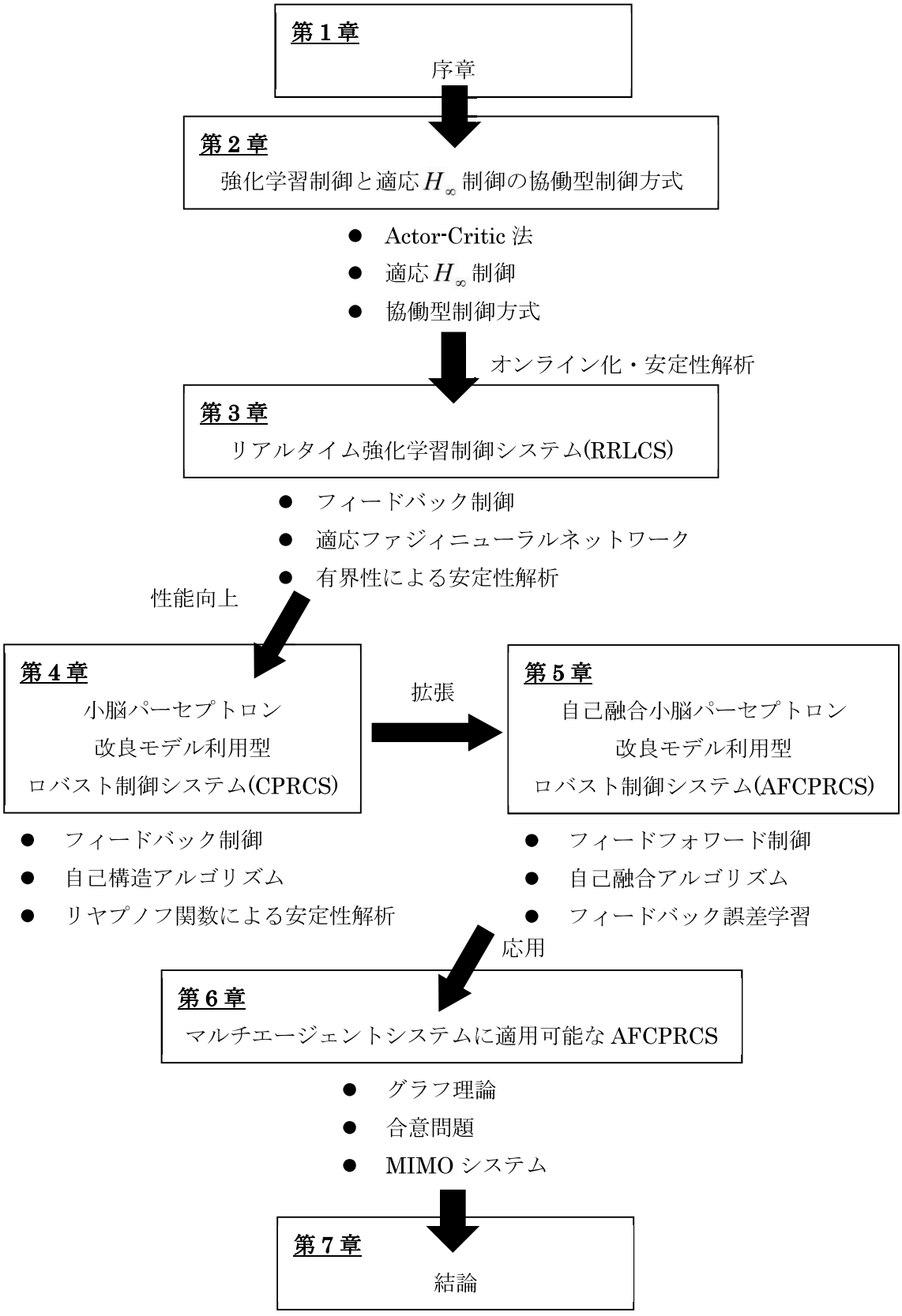

Fig.1.1. Structure of paper

第 2 章　強化学習制御と適応 H_∞ 制御の協働型制御方式

2.1　はじめに

ソフトコンピューティングを用いた知的制御は，人間の知能，生体を模倣した手法であり，その代表例として人間の脳の仕組みを模倣したニューラルネットワーク，人間のあいまいな知識を活かすファジィ理論，人間の学習能力をコンピュータで実現する強化学習などが挙げられる。それらは目的を達成するように設計パラメータを学習の繰り返しで調整しながら制御系を設計するもので，学習範囲外での動作の安定性の保証が得られにくいが，学習中に環境との相互作用の中で対象システムの性質を獲得していくので，対象システムに関する事前知識を必要としないことが大きな強みである。

また，現代制御理論は，時間領域における制御理論で安定性を保証し，精度が高い制御系設計が可能で，かつ，多入力多出力系を容易に取り扱うことができるという点が強みであるが，現代制御理論が使えるかどうかは精度の良い対象のモデルが得られるかどうかにかかわっている。そこで，誕生したのがロバスト制御である。ロバスト制御は，モデルを正確に得ることが困難であることを前提として，精度の悪いモデルでもそれにもとづいた設計で所定の性能を達成する。つまり，対象システムのモデルを必要とするが，安定性を保証した制御方式である。

近年, ソフトコンピューティングと現代制御理論の両者の利点を利用した融合型制御の研究が盛んに行われている。代表的な制御問題として，台車付き倒立振子の制御が挙げられるが，その振子の角度のみを考慮し，台車の位置の制御は考慮されていない。すなわち，上記融合型制御理論は 1 変数を対象としたものであり，台車の位置，振子の角度のように性質の異なる状態を同時に制御することはこれらの制御の枠組みでは困難となっている。

一方，実世界のロボットは，どの地点まで移動するか，そこで何をするかなどの多様な状態表現と状態遷移が考えられる。本論文では，上述の振子の角度のみ，すなわち，角度や角速度などの 1 変数（変数の微分を含む）を対象とした制御系設計法では困難な，性質の異なる多様な状態遷移を計画的に行わせる，即ち，多様な計画行動をロバスト制御と強化学習の協働型制御システムで実現することを目的とする。すなわち，強化学習制御システムと適応 H_∞ 制御システムの協働により性質の異なる状態を同時に制御する方法を提案する。

2.2　提案システム

2.2.1　対象システム

x_1に関するm階非線形微分方程式((2.1)式)，及びx_2に関するn階非線形微分方程式((2.2)式)で表現され，共通の制御入力uで駆動されるシステム，すなわち，１入力２出力の制御システムを対象とする。

$$x_1^{(m)} = f_1(\mathbf{x}_1, \mathbf{x}_2) + g_1(\mathbf{x}_1, \mathbf{x}_2) \cdot u \tag{2.1}$$

$$x_2^{(n)} = f_2(\mathbf{x}_1, \mathbf{x}_2) + g_2(\mathbf{x}_1, \mathbf{x}_2) \cdot u \tag{2.2}$$

ここで，$\mathbf{x}_1 = [x_1, \dot{x}_1, \cdots, x_1^{(m-1)}]^T \in R^m, \mathbf{x}_2 = [x_2, \dot{x}_x, \cdots, x_2^{(n-1)}]^T \in R^n$はシステムの部分的な状態ベクトル，$u \in R$は制御入力，$f_1(\mathbf{x}_1, \mathbf{x}_2), f_2(\mathbf{x}_1, \mathbf{x}_2), g_1(\mathbf{x}_1, \mathbf{x}_2), g_2(\mathbf{x}_1, \mathbf{x}_2)$は未知の連続関数である。$\mathbf{x}_1, \mathbf{x}_2$とも全て観測可とし，制御の目的は状態$\mathbf{x}_1, \mathbf{x}_2$をそれぞれ$\mathbf{x}_{1r} = [x_{1r}, \dot{x}_{1r}, \cdots, x_{1r}^{(m-1)}]^T, \mathbf{x}_{2r} = [x_{2r}, \cdots, \dot{x}_{2r}, \cdots, x_{2r}^{(n-1)}]^T$へ一致させることである。本章ついて説明する。両者は(2.1)(2.2)式のいずれかの制御をそれぞれ担当する。

2.2.2　強化学習の概要

強化学習とは，学習者であるエージェントが自己のおかれた環境との相互作用によって，得られた報酬をベースとして目的の行動戦略を獲得していく学習方式である。エージェントは環境中を行動しながら，できるだけ多くの報酬を獲得することで目的の行動戦略を獲得する。その概要は次の通りである。

(1)　ある特定の環境におかれたエージェントは状態を観測する。
(2)　ある行動決定政策に従って行動を決定し実行する。
(3)　その行動により，エージェントの観測環境はある状態から新しい状態へと推移する。
(4)　エージェントは環境から報酬を得る。

以上の処理を繰り返すことで，エージェントは報酬の期待値を最大化する方策を見つけていく。

本論文では強化学習モデルとして代表的な Actor-Critic 法を利用する。

2.2.3 Actor-Critic 制御システムの構成

Actor-Critic 法は(2.1)式，即ち，m 階微分非線形システムを制御対象とする。Actor-Critic 法とは，TD（Temporal Difference）強化学習法の一種で，状態価値と方策を陽に表現するために，両者が独立した構造を持つ学習方式である。Fig.3.1 に Actor-Critic 制御システムの構成を示す。Actor は system への制御入力 u_r を出力し，状態価値 V が大きくなるように学習を行う。制御信号 u_r を発生させる関数近似器として，RBF (Radial Basis Function)ネットワークを用い，状態及び制御入力が連続なシステムの制御を扱えるように構成する。制御信号は u_r は(2.3)式で計算される。

$$u_r = U_{\max} \cdot \frac{1-\exp\left(-\left\{\sum_{j=1}^{J} w_j b_j(\mathbf{x_1}) + n(t)\right\}\right)}{1+\exp\left(-\left\{\sum_{j=1}^{J} w_j b_j(\mathbf{x_1}) + n(t)\right\}\right)} \tag{2.3}$$

ここで，$U_{\max}$ は出力の最大値，J は中間層のノード数，w_j は出力と中間層 j 番目ノード間の重み，$b_j(\mathbf{x_1})$ は中間層 j 番目ノードの基底関数，$n(t)$ は探索ノイズである。

Critic は，状態価値 V を計算し，TD 誤差 δ を出力する。そして，TD 誤差が小さくなるように学習を行う。TD 誤差は(2.4)式で表される。

$$\delta(t) = r(t) + \gamma_a V(t+1) - V(t) \tag{2.4}$$

ここで，$\gamma_a (0 \le \gamma_a \le 1)$ は減衰係数，$r(t)$ は報酬。この TD 誤差が学習により小さくなるように Critic を構成する。Actor と同様，関数近似器には RBF ネットワークを用いる。このとき，状態価値 $V(t)$ は，(2.5)式で計算される。

$$V(t) = \sum_{j=1}^{J} w_j b_j(\mathbf{x_1}) \tag{2.5}$$

Actor と Critic を構成するネットワークの中間層には RBF を基底関数として用いるが，これを利用したニューラルネットワークでは局所的な関数近似を行うことができ，sigmoid 関数を基底関数として用いたネットワークより学習の収束が速いことで知られている[6]。ここで，$b_j(\mathbf{x_1})$ は(2.6)式で表される。

$$b_j(\mathbf{x_1}) = \exp\left\{-\sum_{i=1}^{I} \frac{(x_i - c_{ji})^2}{\sigma_{ji}^2}\right\} \tag{2.6}$$

ただし，I は入力層ノードの数，x_i は状態変数ベクトル $\mathbf{x}_1$ の i 番目の要素，c_{ji} は中間層 j 番目ノードの基底関数における i 番目入力に対する中心，σ_{ji}^2 は同分散である。

学習において，Critic は TD 誤差を小さくするようにパラメータの学習を行い，Actor はシステムの状態価値を高くするような信号（制御入力）を出力するようにパラメータの学習

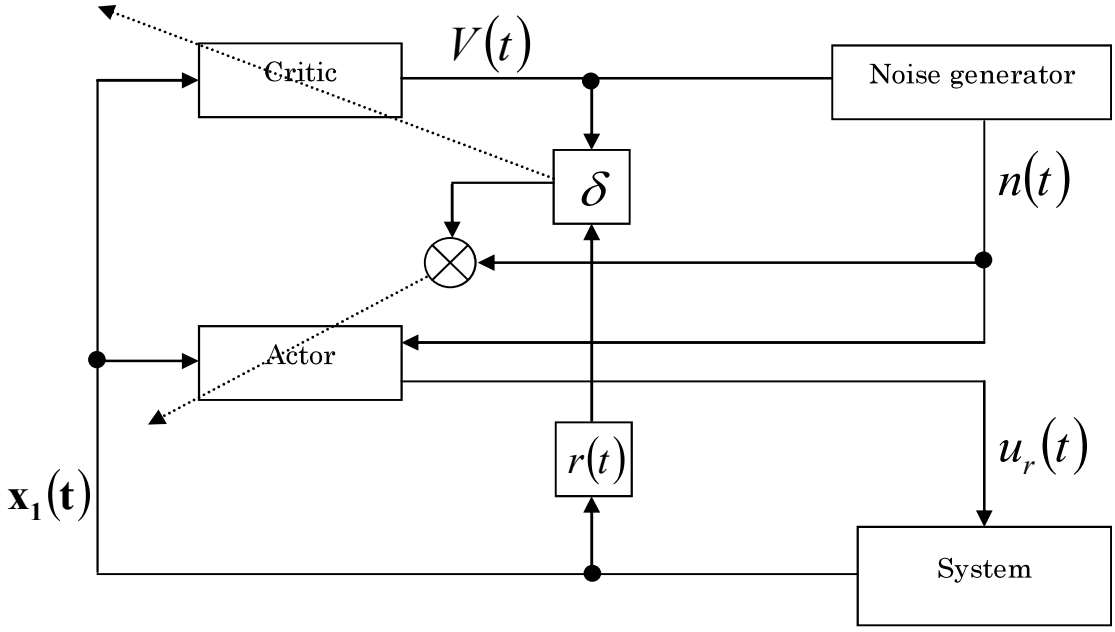

Fig.2.1. Construction of Actor-Critic control system.

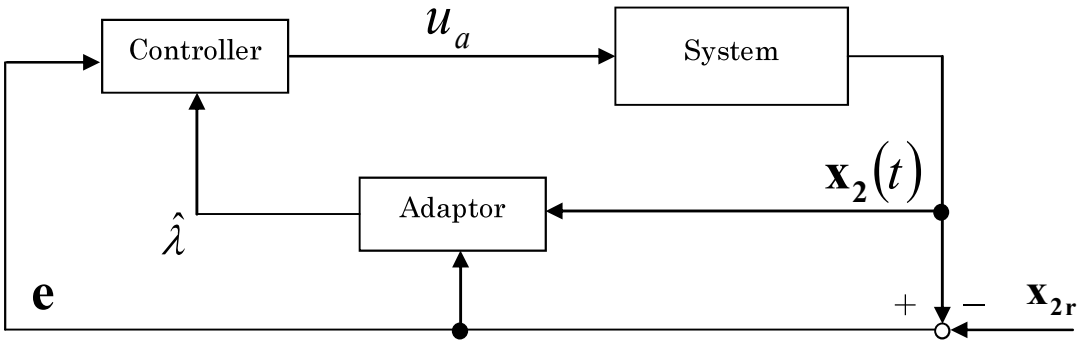

Fig.2.2. Construction of adaptive H_∞ control system.

を行う。その結果，Actor は実際に高い状態価値が得られる出力特性を獲得する。本論文では，RBF ネットワークは一般的な Back Propagation(BP)法を用いて学習する。具体的には，以下の式に従って関数近似器の重みを更新する。

Critic の学習

$$\Delta w_j^{cri} = -\eta_w^{cri} \cdot \frac{\partial V}{\partial w_j^{cri}} \cdot \frac{\partial}{\partial V} \cdot \frac{1}{2} \cdot \delta^2 \tag{2.7}$$

Actor の学習

$$\Delta w_j^{act} = \eta_w^{act} \cdot \frac{\partial u}{\partial w_j^{act}} \cdot \delta \cdot n \tag{2.8}$$

ここで，*cri* は Critic に関する，*act* は Actor に関するパラメータを表す。また，$\eta_w^{cri}, \eta_w^{act}$ は学習率である。

2.2.4 適応 H_∞ 制御システムの構成

本節での議論は主に文献(2)を参考にしている。以下，その概要を述べる。Fig.2.2 に本論文で採用する適応 H_∞ 制御システムの構成を示す。適応 H_∞ 制御は，ある程度の外乱が存在してもそれを許容しつつ，制御対象の所定の性能を達成させるシステムであり，その目的は，状態変数 $\mathbf{x_2}$ を目標の状態 $\mathbf{x}_{2\mathbf{r}}$ に追従させることである。追従誤差 $\mathbf{e}$ は(2.9)式で表される。

$$\mathbf{e} = \mathbf{x_2} - \mathbf{x}_{2\mathbf{r}} \tag{2.9}$$

追従誤差は適応部(adaptor)と入力部（controller）に送られる。本論文では(2.2)式，すなわち，(2.10)式のような n 階微分非線形システムを制御対象とする。但し，$f_2(\mathbf{x}_1, \mathbf{x_2}), g_2(\mathbf{x}_1, \mathbf{x_2})$ とも未知の連続関数で，$g_2(\mathbf{x}_1, \mathbf{x_2})$ の符号は既知で非零とする。

$$x_2^{(n)} = f_2(\mathbf{x}_1, \mathbf{x}_2) + g_2(\mathbf{x}_1, \mathbf{x}_2) \cdot u_a \tag{2.10}$$

ここで，$\mathbf{x_2} = \left[x_2, \dot{x}_2, \ldots, x_2^{(n-1)}\right]^{\mathrm{T}} = \left[x_{m+1}, x_{m+2}, \ldots, x_{n+m}\right]^{\mathrm{T}}$ は適応 H_∞ 制御システムで扱う状態変数ベクトルで全ての状態を観測可とする。u_a は制御信号である。

このとき(2.10)式は次式に書き換えることができる。

$$\dot{\mathbf{x}}_2 = \mathbf{A}\mathbf{x} + \mathbf{B}\left[f_2(\mathbf{x}_1, \mathbf{x}_2) + g_2(\mathbf{x}_1, \mathbf{x}_2)u_a\right] \tag{2.11}$$

ここで,

$$\mathbf{A}=\begin{bmatrix}0 & 1 & 0 & \cdots & 0\\ 0 & 0 & 1 & \cdots & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & 0 & \cdots & 1\\ 0 & 0 & 0 & \cdots & 0\end{bmatrix},\quad \mathbf{B}=\begin{bmatrix}0\\ 0\\ \vdots\\ 0\\ 1\end{bmatrix}$$

である。

また，設計仕様として正定数γが与えられているものとすると次式のリカッチ(Riccati)方程式を満たす正定対称行列$\mathbf{P}$が存在する。

$$\mathbf{A}^T\mathbf{P}+\mathbf{P}\mathbf{A}-\gamma\cdot\mathbf{P}\mathbf{B}\mathbf{B}^T\mathbf{P}+\mathbf{Q}=\mathbf{0} \tag{2.12}$$

適応部(adaptor)の機能は，制御信号u_aの大きさを調整するパラメータ推定量$\hat{\lambda}$を出力することである。関数近似器にRBFネットワークを用いる。中間層ノードの出力$s_l(\mathbf{x_2})$は(2.13)式で計算される。

$$s_l(\mathbf{x_2})=\exp\left\{-\sum_{n=1}^{N}\frac{(x_{n+m}-\mu_{\ln})^2}{\eta_{\ln}^2}\right\} \tag{2.13}$$

また，出力層ノードの出力H_{nn}は(2.14)式で計算される。

$$H_{nn}=\sum_{l=1}^{L}s_l^2(\mathbf{x_2}) \tag{2.14}$$

ここで，Nは入力層ノードの数，Lは中間層ノードの数，x_{n+m}は状態変数の$n+m$番目の要素，$\mu_{\ln}$はl番目の基底関数のn番目の入力に対する中心，$\eta_{\ln}^2$は同分散である。

パラメータ推定量$\hat{\lambda}(\le c)$は，(2.15)式で計算される。

$$\dot{\hat{\lambda}}=\begin{cases}\Gamma\bar{\phi} & if\quad (\hat{\lambda}>c\quad and\quad \phi>0),\\ \Gamma\phi & otherwise,\end{cases} \tag{2.15}$$

ただし，Γは適応ゲイン，cは$\hat{\lambda}$の上限値である。また，ϕと$\bar{\phi}$はそれぞれ(2.16)式と(2.19)式で定義されるものとし，ρは減衰率，$\lambda(>c)$は定数，δ_1は調整パラメータである。

$$\phi=\Phi(\mathbf{x_2})\mathbf{e}^T\mathbf{P}\mathbf{B}\mathbf{B}^T\mathbf{P}\mathbf{e}+\psi(\mathbf{x_2})\left\|\mathbf{B}^T\mathbf{P}\mathbf{e}\right\| \tag{2.16}$$

$$\Phi(\mathbf{x_2})=\gamma+\frac{1}{\rho^2}+(1+H_{nn}) \tag{2.17}$$

$$\psi(\mathbf{x_2})=1+(H_{nn})^{1/2} \tag{2.18}$$

$$\bar{\phi}=\left(1+\frac{c-\lambda}{\delta_1}\right)\phi \tag{2.19}$$

入力部(controller)は，system への制御入力u_aを出力する。制御信号u_aは(2.20)式で計算される。

$$u_a = -\hat{\lambda}\Phi(\mathbf{x_2})\mathbf{B}^T\mathbf{P}\mathbf{e} - \hat{\lambda}\psi(\mathbf{x_2})\mathrm{sgn}(\mathbf{B}^T\mathbf{P}\mathbf{e}) \tag{2.20}$$

ここで，sgn は符号関数で，次式で定義される。

$$\mathrm{sgn}(\sigma) = \begin{cases} +1 & (\sigma > 0) \\ -1 & (\sigma < 0) \end{cases} \tag{2.21}$$

(2.20)式の右辺第 1 項はH_∞制御入力であり，外乱を抑え，出力に影響を与えないようにする。第 2 項は可変構造系の特性を持つ制御信号で，σの符号で切り替えることによりシステムの安定化を図るものである。これらの導出の詳細は文献[19]を参照されたい。

2.2.5 強化学習制御と適応H_∞制御の協働型制御システムの構成

Fig.3.3 に提案システムである協働型制御システムを示す。1 変数を対象とする非線形制御方式では制御が困難な(2.1)(2.2)式の 2 変数の多階非線形微分方程式で表わされるシステムを制御対象とし，(2.1)式の$\mathbf{x}_1$の制御を強化学習システムにおける Actor の制御信号u_rが，(2.2)式の$\mathbf{x}_2$の制御を適応H_∞制御信号u_aがそれぞれ担当し，互いに協働相手の状態の影響を受けながら，system への単一の制御信号uを(2.22)式のように構成し，協働で制御する方式を提案する。ここで，2.2.3 節における制御入力u_rは状態変数$\mathbf{x}_1$のみ，2.2.4 節における制御入力u_aは$\mathbf{x}_2$のみを用いて構成していることに注意する。すなわち，(2.1)(2.2)式で表わされる制御対象システムは，本提案方法のように，システムの状態を分割して制御する方法を用いずに，全体を一括して制御する方法をとれば，1 入力 2 出力の非線形制御問題を解く必要がある。

ここでは，この問題をこれまでしばしば提案され，取り扱いが容易な 1 入力 1 出力非線形制御系設計問題に関する知見を利用して制御系を設計する。すなわち，(2.1)(2.2)式をそれぞれ，x_1, x_2に関する単独のダイナミックスとみなし，(2.1)におけるx_2，(2.2)式におけるx_1をそれぞれ外乱としみなし制御系設計を行なう。但し，それらは外乱ではあるが誤差として無視し，これらの誤差の影響は，外乱に耐性のある適応H_∞制御過程，強化学習過程においてその誤差が吸収されること，及びαを適切に設定することで高い制御性能の実現を期待する。ここでは，2 変数を 1 変数の二つのサブシステムで対処する方法として，一つは強化学習制御，もうひとつは適応H_∞制御で代表させたが，他の組み合わせ，すなわち，強化学習制御部分も適応H_∞制御とし，両サブシステムとも適応H_∞制御の組み合わせも考えられる。そこで，以後，前者の構成を提案法 1，後者の構成を提案法 2 と呼ぶ。両方のサブシステムが強化学習制御で構成されたシステム構成も可能であるが，この場合は 2.2.3 節の計算機シミュレーションで述べるように，強化学習そのものが多くの試行回数を必要としな

がらもそれほど高い制御性能が期待できないので本論文では言及しない。ここで,

$$u = \alpha u_a + (1-\alpha) u_r \tag{2.22}$$

ただし，$\alpha(0 \le \alpha \le 1)$ は平滑化定数である。

状況に応じて α の値を調整することにより，$\mathbf{x}_1$ 優先制御か $\mathbf{x}_2$ 優先制御か，制御信号の目的を明確にすることができる。

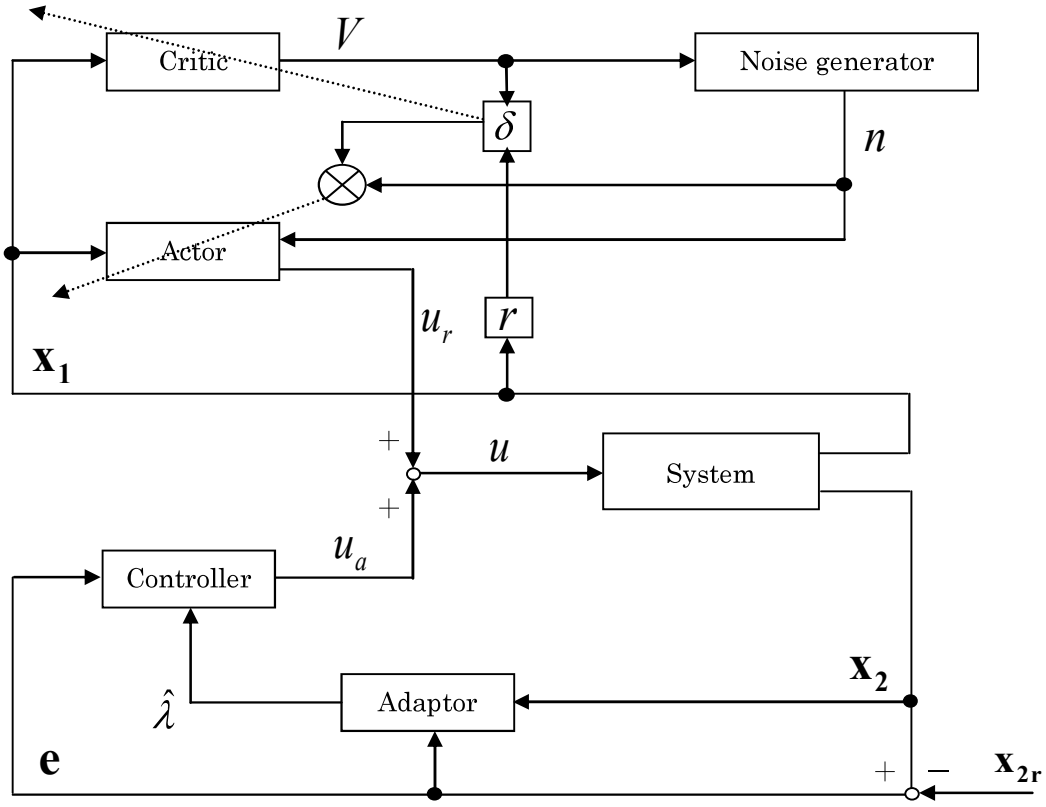

Fig.2.3.　Proposed cooperated control system.

2.3　計算機シミュレーション

2.3.1　制御対象

制御対象は，台車に振子がぶら下がる形に取り付けられた頭上走行クレーンを考える。クレーンシステムの概略を Fig.2.4 に示す。ダイナミクスは次式で与えられる。

$$\ddot{\theta} = -\frac{g}{l}\sin\theta + \frac{m}{m+M}\left\{\cos^2\theta - \frac{1}{2}\sin 2\theta\cdot\dot{\theta} - \frac{\cos\theta}{m\cdot l}\cdot F\right\} \tag{2.23}$$

$$\ddot{x} = -\frac{ml}{m+M}\cos\theta + \frac{ml}{m+M}\sin\theta\cdot\dot{\theta} + \frac{1}{m+M}\cdot F \tag{2.24}$$

また，F は入力信号，θ は角度，x は位置，x_r は目標地点，m は積荷の質量，M は台車の質量，g は重力加速度，l は振子の長さである。システムが観測する状態は時間ステップ t において $\mathbf{x}_t \equiv (\theta_t, \dot{\theta}_t, x_t, \dot{x}_t)$ であり，シミュレーション（制御）の目的は，振子の角度を 0 に保ちつつ，台車を目標地点に移動させることである。計算機上で提案システムである協働型制御システムの性能を検証するため，従来法として Actor-Critic 法のみで台車及び振子の両方を制御するシステム（以下従来法と呼ぶ）との性能比較シミュレーションを行った。

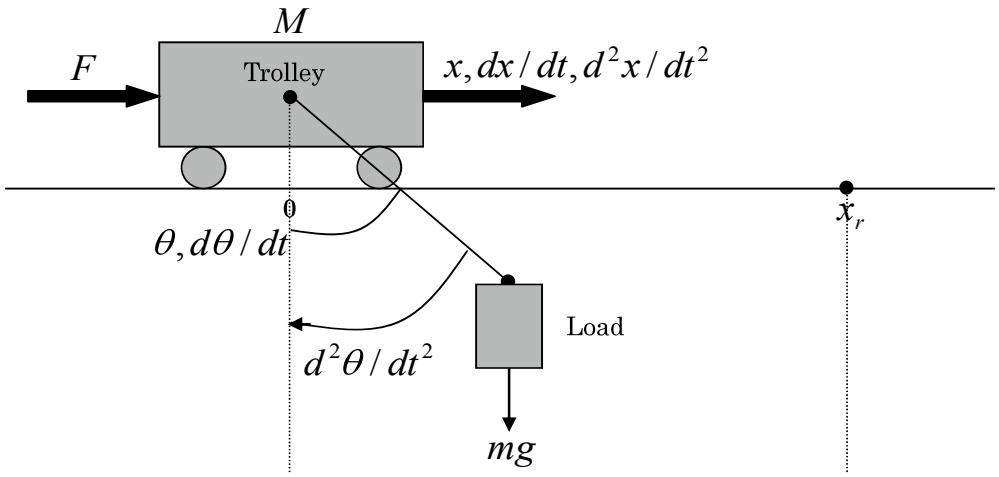

Fig.2.4.　Overhead traveling crane system.

クレーンシステムのシステム定数は $m=0.5[kg]$，$M=1.0[kg]$，$l=0.5[m]$，$g=9.8[m/s^2]$ とした。Table 2.1 にシミュレーションで使用したパラメータ値を示す。提案法 1 のパラメータは，従来法と共通部分である Actor-Critic 法に関するパラメータ以外に，適応 H_∞ 制御に関するパラメータが追加されている。提案法 2 のパラメータは，両サブシステムとも適応 H_∞ 制御に関するパラメータであるが, u_a を求める場合と u_r を求める場合とでは一部異なっている。これらのパラメータは共通にできる部分は同じ値としているが，基本的には，試行錯誤により，2 手法とも最良の結果を得る値を用いている。

Table 2.1. Parameters used in simulation.

Parameters	Proposed method		Actor-Critic method (Conventional method)
	1	2	
I	2	-	4
J	35	-	49
N	2	2	-
L	25	25	-
γ	0.5	0.5	-
Q	diag {0.4,0.4}	diag {0.24,0.24} for u_a	-
		diag {0.01,0.01} for u_r	
Γ	0.002	0.002	-
c	0.1	0.1	-
ρ	0.5	0.5	-
λ	1.0	1.0	-
δ_1	0.01	0.01	-
$U_{\max}$	3.0	-	3.0
γ_a	0.95	-	0.95
α_r^θ	1.0	-	0.2
α_r^θ	1.0	-	20.0
α_r^x	4.0	-	4.0
α_r^v	20.0	-	20.0
η_w^{cri}	0.05	-	0.02
η_w^{act}	0.05	-	0.02

2.3.2　シミュレーション条件

初期状態は，初期角度が 10 度，初期角速度，初期位置，初期速度が 0 の状態で，サンプリングタイムは 0.02 秒とした。制御時間は 40 秒とし，振子の角度の絶対値が 45 度を超えた場合，また，台車の位置の絶対値が 10m を超えた場合は失敗とみなし，その試行は打ち切る。以上を 1 試行とする。適応 H_∞ 制御の場合はオンライン，すなわち，制御しながら制御器のパラメータは調整されるが，強化学習の場合は試行の繰り返しにより制御器のパラメータを少しずつ調整することで制御器の性能向上を図る。なお試行中の最後の 5 秒間，台車が目標地点の±0.25[m]を維持できた場合を制御成功とみなす。

提案法 1

提案法 1 は，Actor-Critic 法で位置と速度の制御を行うため，$\mathbf{x_1} = [x_1, x_2]^T = [x, \dot{x}]^T$ とし，$\mathbf{x_{1r}} = [x_{1r}, \dot{x}_{1r}]^T = [x_r, 0]^T$ とする。また，適応 H_∞ 制御は角度と角速度の制御を行うため，2.2.4 節の記述に従えば，$\mathbf{x}_2 = [x_3, x_4]^T = [\theta, \dot{\theta}]^T$ であり，その目標値は $\mathbf{x}_{2r} = [x_{2r}, \dot{x}_{2r}]^T = [0,0]^T$ となる。(2.22)式の制御入力 u は平滑化定数 α の割合で振れ角の制御入力 u_a を採用し，$1-\alpha$ の割合で台車位置の制御入力 u_r を採用することを意味している。(2.25)式はシステムの状態の違いによる α の指定方法を規定しており，振子の振れ角 θ が目標値零より 1 度以上大きい（$|\theta| > 1$），もしくは台車位置 x が目標値 x_r に近い（0.1 以内）場合に α は s とし，そうでなければ $1-s$ とする。ここでは $s = 0.8, 0.5, 0.2$ の 3 ケースについてシミュレーションを行う。なお，振れ角 θ の初期値は 0 としており，かつ，台車の位置は目的位置より離れているため，初期状態の α は $1-s$ の値を取る。

$$\alpha = \begin{cases} s \quad if \quad (|\theta| > 1) \quad \text{または} \quad (x_r - 0.1 < x < x_r + 0.1) \\ 1-s \quad otherwise \end{cases} \tag{2.25}$$

提案法 1 における Actor-Critic 法の報酬は，(2.26)式のように定義する。

$$r(t) = \exp\left\{ -\left(\frac{(x - x_r)^2}{2\left(\alpha_r^x\right)^2} + \frac{(\dot{x})^2}{2\left(\alpha_r^v\right)^2} \right) \left(|\theta| + \alpha_r^\theta\right)^2 \left(|\dot{\theta}| + \alpha_r^{\dot{\theta}} \right)^2 \right\} \tag{2.26}$$

ここで，$\alpha_r^\theta, \alpha_r^{\dot{\theta}}, \alpha_r^x, \alpha_r^v$ はそれぞれ，角度，各速度，位置，速度に関する正の重み定数である。Actor-Critic 法は位置の制御を行うので，目標地点で制止するとき，つまり $(x, \dot{x}) = (x_r, 0)$ のとき最大報酬 1 を与える。また，(2.26)式はある時間の位置，速度と別の時間の位置と速度が同じ値のとき，角度，角速度が目標である 0 に近いほど大きい報酬を与えるようにしている。

提案法 2

提案法 2 は, 位置と速度の制御システム, 角度と角速度の制御システム, 両者とも適応 H_∞ 制御で行うため，2.2.4 節の適応 H_∞ 制御システムの構成に従って設計する。提案法 1 の適

応 H_∞ 制御部分と同様にして 2 サブシステムを構成する。詳細は紙面の都合上省略する。

比較対象である従来法（Actor-Critic 法）は，角度，角速度，位置，速度すべてを制御するので $\mathbf{x}=\left[\theta,\dot{\theta},x,\dot{x}\right]$ とし，これを Actor および Critic の入力とした。

従来法における Actor-Critic 法の報酬は，(2.27)式のように定義する。

$$r(t)=\exp\left\{-\frac{\theta^2}{2\left(\alpha_r^{\theta}\right)^2}-\frac{\dot{\theta}^2}{2\left(\alpha_r^{\dot{\theta}}\right)^2}-\frac{\left(x-x_r\right)^2}{2\left(\alpha_r^{x}\right)^2}-\frac{\dot{x}^2}{2\left(\alpha_r^{v}\right)^2}\right\} \tag{2.27}$$

Actor-Critic 法のみで，状態変数すべての制御を行うため，振子の角度，角速度が 0 で，かつ目標地点 x_r で静止するとき，つまり $\left(\theta,\dot{\theta},x,\dot{x}\right)=\left(0,0,x_r,0\right)$ のとき最大報酬 1 を与える。提案システムの報酬の式との違いは，提案システムでの Actor-Critic 法はあくまで位置と速度のみの制御なので，角度と角速度がどの値でも $(x,\dot{x})=(x_r,0)$ さえ満たせば最大報酬 1 を与える（(2.26)式）。それに対し，従来法の Actor-Critic は Actor-Critic のみで角度，角速度，位置，速度を制御するため，すべての状態変数が理想値である $\left(\theta,\dot{\theta},x,\dot{x}\right)=\left(0,0,x_r,0\right)$ を満たしたときに最大報酬 1 を与える((2.27 式))。

2.3.3 シミュレーション結果

3 種類の平滑化定数 α に対する提案法 1，2 における角度と位置の推移をそれぞれ，Fig.2.6~2.8 に示す。目標地点は 5m 地点とした。強化学習制御を用いる場合は 1000 回の試行を行い，1000 回目の学習により得られた制御系を，以後，評価対象システムとする。提案法 1 は Fig.2.5, Fig.6 から $s=0.8$ が最良の結果となった。$s=0.8$ の場合，制御初期は振れ角の初期値 $\theta_0=10$ であるため，(2.25)式から $\alpha=s=0.8$ となり， u_a に重きを置く，即ち，振れ角制御を重視することになっていることから，振れ角が 0 に近づく。その後，α の切り替えにより台車の制御へ比重が移り，速やかに 5m の位置へ収束している。提案法 2 の結果である Fig.2.7, Fig.2.8 では最良は $s=0.5$ となった。3 つの s の中で $s=0.5$ が最も優れているのは，$s=0.5$ の場合，$s=0.8$ と比較し，台車の立ち上がりの制御，振れ角の収束に優れている点にある。Fig.2.9, Fig.2.10 は提案法 1，2 の最良の結果と従来法の三者の比較図である。三者の比較では，角度は，収束が見られない従来法に対し，提案システム 1, 2 とも 0 に近い値に収束し(Fig.2.9)，位置は従来法よりも速く目標地点である 5[m]地点に収束している(Fig.2.10)。これらより，提案法 1, 2 とも従来法と比べて明らかに制御性能が良いことが確認できる。提案法 1 と 2 の比較では，提案法 2 では $s=0.5$ であるため，振れ角の制御と台車の制御の比重が等しく制御が同時に開始されている。従って，制御入力が分散され，振れ角，台車の応答は提案法 1 より劣る結果となっている。三者の中では，提案法 1 が最も優れていると言える。

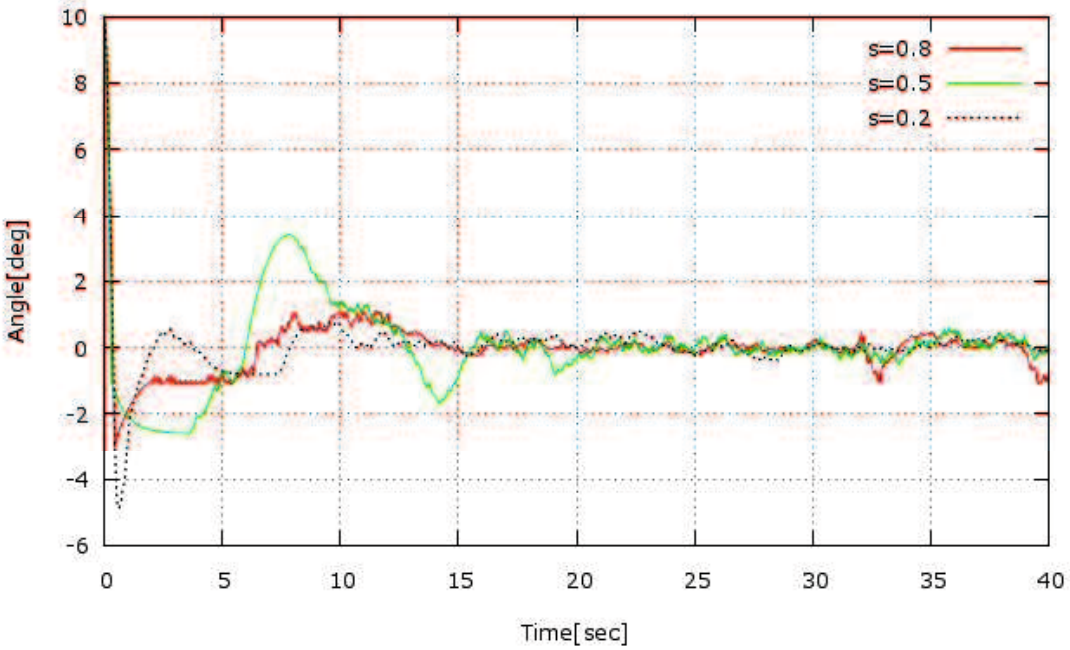

Fig.2.5. Control results of the angle by the proposed method 1

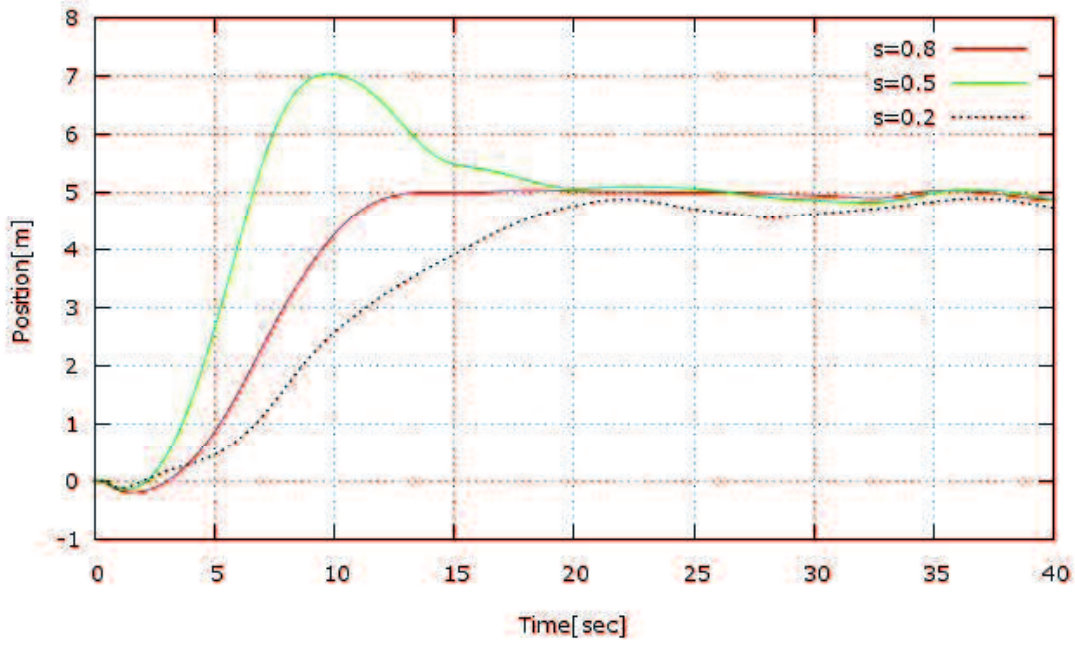

Fig.2.6. Control results of the position by the proposed method 1

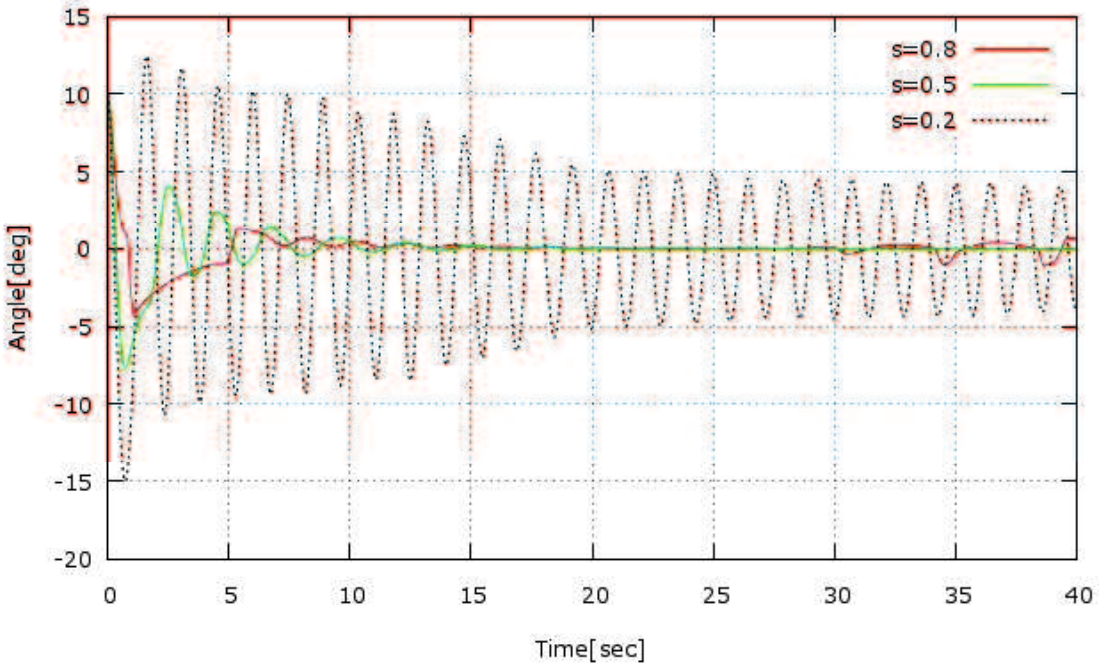

Fig.2.7. Control results of the angle by the proposed method 2

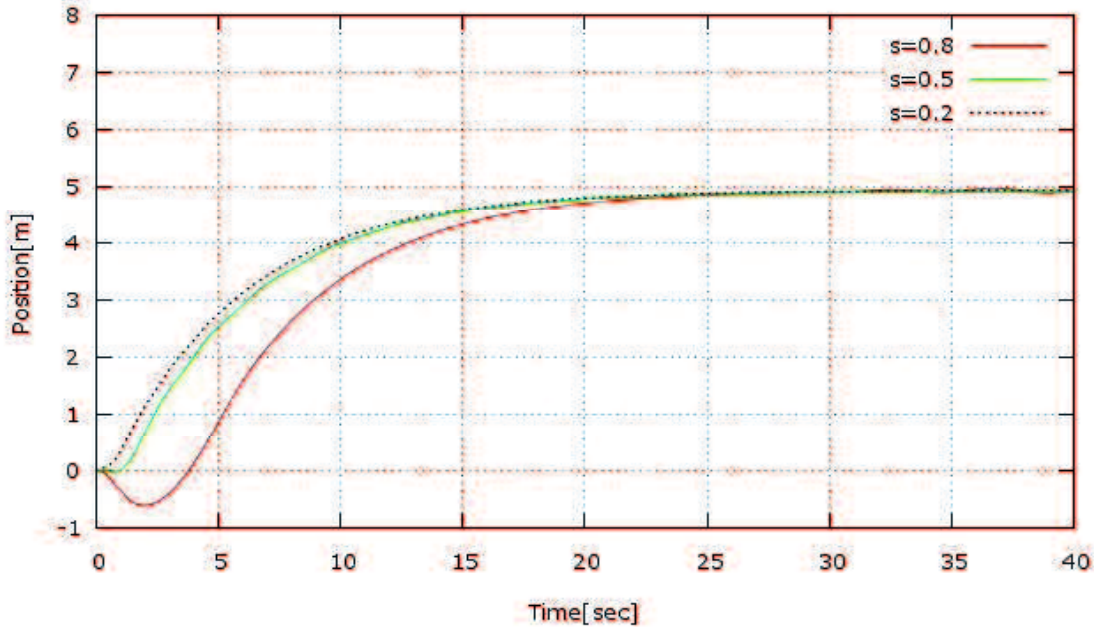

Fig.2.8. Control results of the position by the proposed method 2

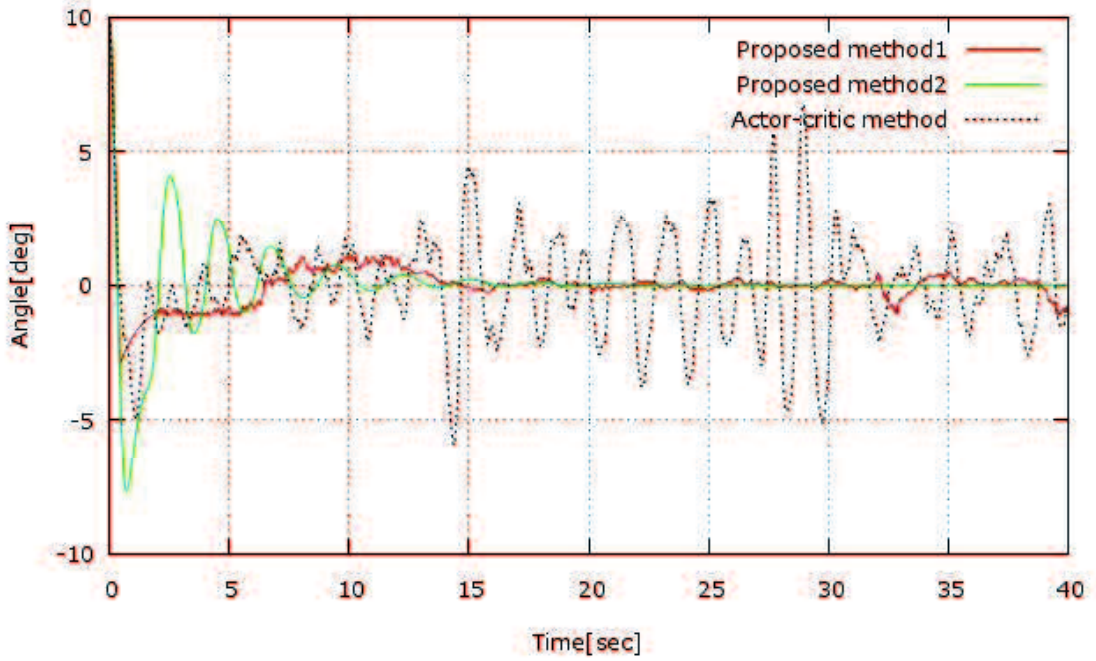

Fig.2.9. Comparison of control results of the angle among the three

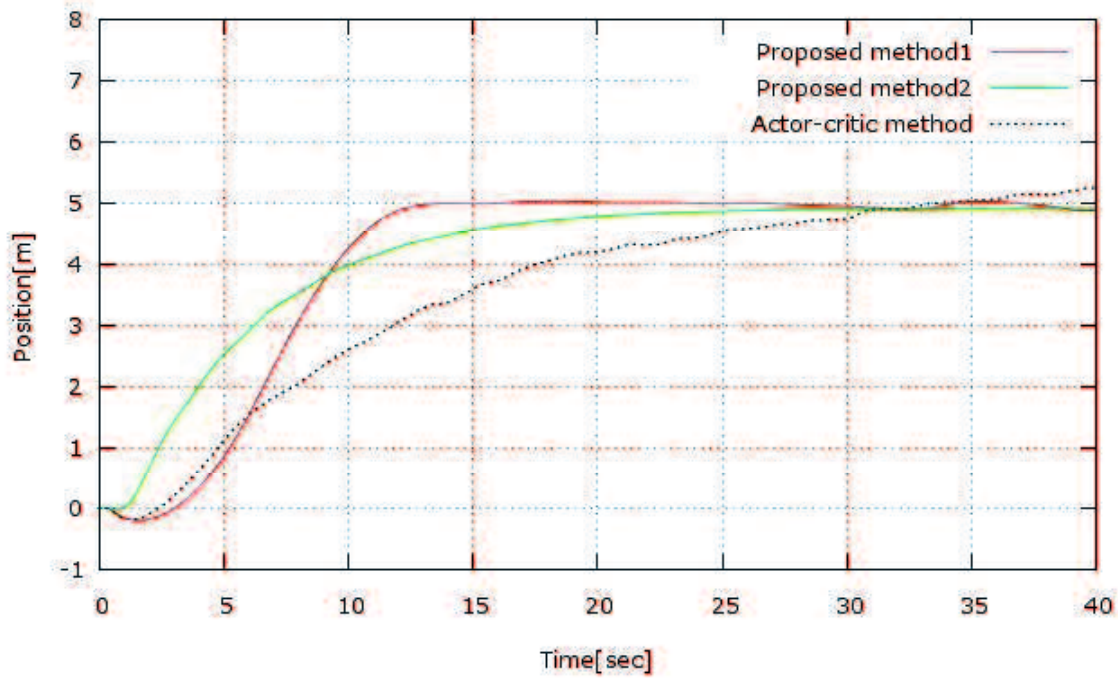

Fig.2.10. Comparison of control results of the position among the three

2.3.4　ロバスト性の検証

提案法 1，2 及び従来法の三者のロバスト性を検証するために，学習で得られた制御システムを用いて，クレーンシステムのパラメータと目標地点，初期角度を変えて検証を行った。Table 2.2 は制御成功限界を与える各システムのパラメータの値である。各パラメータにおいて，最大の場合は大きい方が，最小の場合は小さい方がロバスト性が良いとした検証結果を示す。三者の内，荷の最大質量は提案法 1 が，それ以外は全てのパラメータの上限値から下限値までの制御成功範囲において提案法 2 が圧倒的に優れたロバスト性を示した。

Table 2.2.　Results of robustness validation.

	Proposed method 1	Proposed method 2	Actor-Critic method
Maximum weight of the load	2.10	1.49	0.98
Minimum weight of the load	0.00	0.00	0.48
Maximum weight of the trolley	23.05	48.48	23.75
Minimum weight of the trolley	0.00	0.00	0.98
Maximum length of the pendulum	1.33	1.48	0.95
Minimum length of the pendulum	0.21	0.01	0.47
Maximum objective point	5.24	10.00	5.40
Minimum objective point	4.82	-10.00	5.00
Maximum initial angle	14.08	30.00	13.32
Minimum initial angle	-14.08	-30.00	-13.32

2.3.5 考察

提案法 1，2，従来法の三者の比較では，学習終了時の制御では，強化学習制御と適応H_∞制御の協働制御が優れ，ロバスト性においては，提案法 2 が非常に優れ，他を圧倒した。強化学習を制御系設計に含めると，学習の繰り返しで目的の正確な実現には優れているがロバスト性に劣ることがわかる。このことはロバスト性の強化の仕掛けを強化学習過程へ導入していないことからも予想できる。適応H_∞制御同士の組み合わせのロバスト性に強い原因はそれぞれがロバスト性に強いことからも妥当な結果と言える。但し，逆に，ロバスト性の強さが制御性能の劣化を引き起こす結果になったともと考えられる。

2.4 まとめと今後の課題

1 変数を対象とする非線形制御方式では制御が困難な 2 変数の多階非線形微分方程式で表わされるシステムを制御対象とし，これまで多くの実績があり，取り扱いが容易な，1 変数を対象とした強化学習制御，適応H_∞制御による協働型制御系設計法を提案し，クレーン制御システムの計算機シミュレーションにより提案法の有効性を示した。その中で，適応H_∞制御系を含む協働型制御方式では，強化学習制御系に比較し，若干の制御性能の劣りが見られるものの，ロバスト性に優れた制御系となること，学習による制御系は制御性能に優れるものの，ロバスト性に劣ることを示した。今後の展開として，より複雑な制御対象システムに対する本提案法の検証が考えられる。

第 3 章　H_∞追従性能補償器を備えた

リアルタイム強化学習制御システム

3.1　はじめに

現代社会に存在する多くのシステムは非線形システムであり，その時間的変化は通常，非線形の微分方程式で表すことになる。しかしながら，その変化は複雑で微分方程式でのモデル化が困難になる場合も多く，また，表現できてもそのモデルには多くの不確かさが存在する。そのようなシステムの制御には，制御対象の数式モデルを必要としない制御器の設計が可能なモデルフリーな制御が必要となる。このモデルフリーな制御として，システムの不確定性に頑健なロバスト制御とモデルを必要としない制御系設計法である強化学習の両者を融合することで，環境との相互作用で制御系設計が可能なロバスト強化学習制御[36]～[38]が有効である。ロバスト学習制御に関する研究は，未だ多くは見られない。その一つである文献[36]のロバスト学習制御は，強化学習には，連続の選択・決定のための計算量を最小限にして制御可能とする Actor-Critic 法を用いている。その特徴は，行動方策を司る Actor と PI 制御器などの制御器である Nominal Controller(NC)を並列に配置していることである。初期の制御ステップでは Actor は未学習なため，NC が制御を受け持ち，その後 Actor は，学習により，次第に最適な Actor を構築していく。しかしながら，その短所として，Actor の動作を評価する Critic を離散値表現の Q-table で構築しているため，大規模で状態数が無限に考えられる連続値系の制御対象の制御が困難となる。その場合，あらかじめ設計者が決めた状態変数のみを考慮した学習を行うことで対応している。文献[37]は，外乱生成器を付与した強化学習で，最悪外乱を出力してそれに耐えうる学習を行うことで環境の変動に強い強化学習を実現している。文献[44]は，制御対象の状況に応じて適応H_∞制御と強化学習制御器の両者が協働して制御を行うシステムを提案している。しかしながら，適応H_∞制御器との協働制御を行っている文献[44]における強化学習制御器は，通常の強化学習と同様，事前にオフラインによる多数回の繰り返し学習を必要とする。さらに，適応H_∞制御のみの制御システムに比べて，ロバスト性が劣る欠点も有している。

また，モデルフリーを意図した先行研究として，文献[17],[32]がある。文献[17]は，H_∞追従性能を保証するH_∞追従性能補償器(H_∞ Tracking performance Compensator, HTC)と適応ファジィ制御器の二つの制御器で構成している。その特徴は文献[1]と同様，初期の制御ステップは，HTC でシステムの制御を行うことである。その間，適応ファジィ制御器が学習のより構築し，それが完了すると HTC に代わってシステムの制御を担う。文献[32]は，ノード数を自ら追加・削除を行う適応的な自己構造化ファジィニューラルネットワーク制御器(ASFNCS)と不安定なシステムを安定化する可変構造制御器の 2 つの制御器で構

築している。しかしながら，可変構造制御器の性質から，高周波振動であるチャタリングが発生する欠点を有している。

本論文では，文献[36]の特徴を活かし，その短所を改善し，学習機能を備えた制御システムを構築する。即ち，未知の非線形の微分方程式で表現されるシステムに対応でき，効率の良い学習・ロバスト性・システムの安定性を保証した，事前にオンライン学習を必要としない「リアルタイム強化学習制御システム(Real-time Reinforcement Learning Control System，RRLCS)」を提案する。RRLCS は，文献[36]の NC の代わりに用いる HTC[17]と強化学習の一種である Actor-Critic 法とを組み合わせた制御システムである。更に，追従誤差の L_2-ゲインを有界にする H_∞ 制御の概念とリヤプノフ関数を用いて，提案システムの安定性を証明し，HTC の動作情報を得て学習することで追従性能を最適化する Actor を実現する。また，Critic を Q-table ではなく，環境適応型の多層 NN で構築することで，文献[36]では対応が困難であった連続値行動である無限の状態数にも対応可能とする。最後に，台車付き倒立振子による計算機シミュレーションにより，従来のモデルフリー制御方式[17],[32]と性能比較を行い，提案システムの有効性を検証する。

3.2 制御対象の定式化

次の n 次非線形システムを制御対象とする。

$$x^{(n)} = f(\mathbf{x}) + g(\mathbf{x})u \tag{3.1}$$

ここで，$\mathbf{x} = \left[x, \dot{x}, \cdots, x^{(n-1)}\right]^T = \left[x_1, x_2, \cdots, x_n\right]^T \in \mathrm{R}^n$ はシステムの状態変数，f, g は未知の連続関数で，$g > 0$ とする。u は入力信号である。式(3.1)を状態空間表現に書き換えると(3.2)式で表わされる。

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\left(f(\mathbf{x}) + g(\mathbf{x})u\right) \tag{3.2}$$

ここで,

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

である。

このシステムは，状態変数 $\mathbf{x}$ を目標の状態 $\mathbf{r}$ に追従させることを目的としている。状態変数 $\mathbf{x}$ と目標信号 $\mathbf{r}$ との追従誤差ベクトル $\mathbf{e}$ を次式で表わす。

$$\mathbf{e} = \mathbf{r} - \mathbf{x} \tag{3.3}$$

ここで，$\mathbf{e}=\left[e,\dot{e},\cdots,e^{(n-1)}\right]^T\in \mathrm{R}^{\mathrm{n}}$, $\mathbf{r}=\left[r,\dot{r},\cdots,r^{(n-1)}\right]^T\in \mathrm{R}^{\mathrm{n}}$ である。

関数 f,g が既知であるとき，システムの最適入力 u^* は(3.4)式で表わされる。

$$u^*=g^{-1}\left(-f+r^{(n)}+\mathbf{k}^T\mathbf{e}\right) \tag{3.4}$$

ここで，$\mathbf{k}=\left[k_n,k_{n-1},\cdots,k_1\right]^T\in \mathrm{R}^{\mathrm{n}}$ はフィードバックゲイン，$r^{(n)}$ は n 階微分の目標信号である。この最適入力 u^* を(3.1)式に代入すると次式を得る。

$$e^{(n)}+k_1e^{(n-1)}+\cdots+k_{n-1}\dot{e}+k_ne=0 \tag{3.5}$$

ここで，$k_i\,(i=1,2,\cdots n)$ は(3.5)式に対し，フルビッツの安定性を満足するように決定する。そのように決定した場合，式(3.5)より，式(3.4)を用いれば，$\lim_{t\to\infty}e_t=0$ になることが分かる。

しかし，実際には関数 f,g は未知なのでニューラルネットワークを用いて，最適入力 u^* を近似する。

3.3 H_∞追従性能補償器を備えた

リアルタイム強化学習制御システム

提案するH_∞追従性能補償器を備えたロバスト強化学習システムの構造を Fig.3.1 に示す。

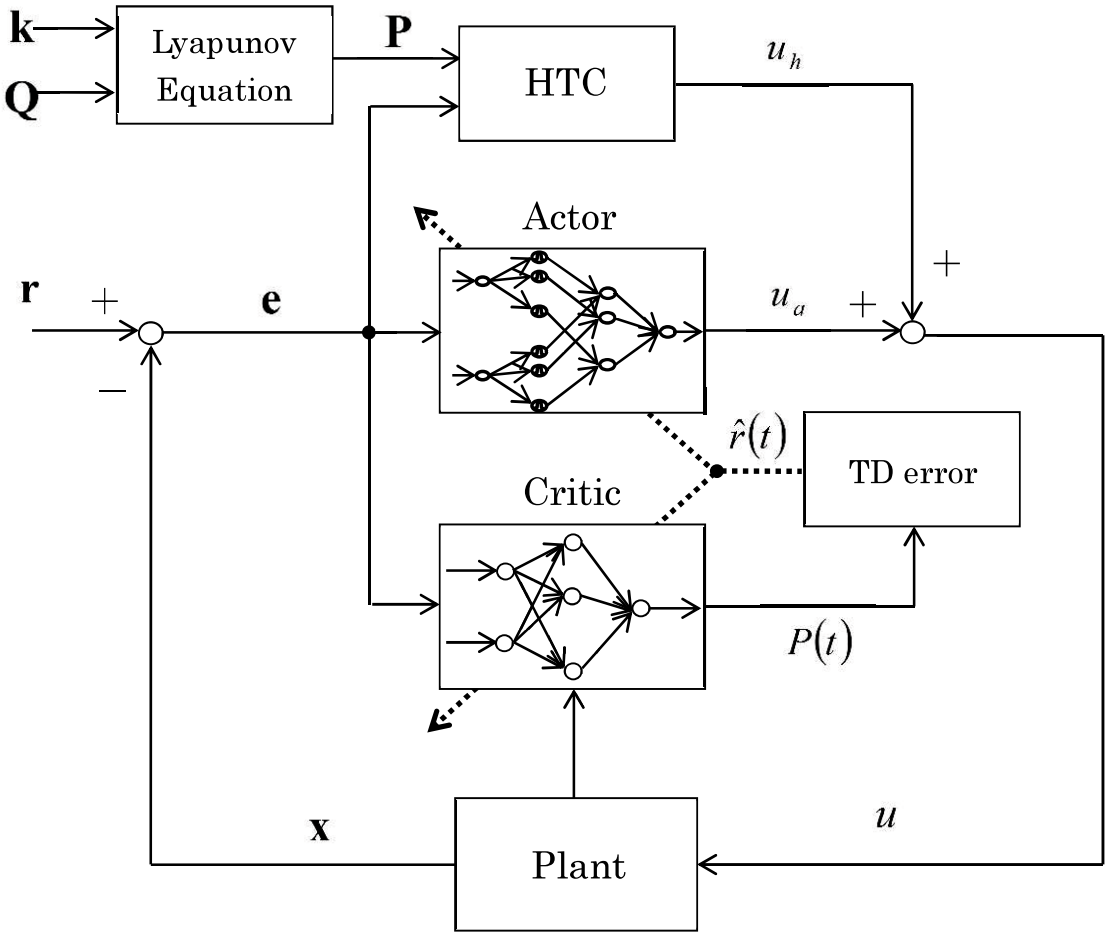

Fig.3.1. Structure of real-time reinforcement learning control system

ここで，**x** は状態変数ベクトル，$\mathbf{r_f}$ は目標信号ベクトル，**e** は追従誤差ベクトル，**k** はフィードバックゲイン，**Q** は正定対称行列，**P** はリヤプノフ方程式の解，u_r は強化学習出力信号，u_h は HTC 出力信号，u は制御対象への入力信号，$r(t)$ は報酬，$P(t)$ は予測報酬，$\hat{r}(t)$ は TD 誤差である。

提案システムの各構成は次節以降で説明する。

3.3.1 H_∞追従性能補償器

<リヤプノフ方程式>

フルビッツの安定性を満足するように，設計者が任意に決めたフィードバックゲイン$\mathbf{k}$と正定対称行列$\mathbf{Q}$で次のリヤプノフ方程式を解く。

$$\mathbf{\Lambda}^T\mathbf{P}+\mathbf{P}\mathbf{\Lambda}=-\mathbf{Q} \tag{3.6}$$

ただし，$\mathbf{\Lambda}$は

$$\Lambda=\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ -k_n & -k_{n-1} & \cdots & -k_2 & -k_1 \end{bmatrix}$$

であり，フルビッツの安定行列である。リヤプノフ方程式の解$\mathbf{P}$を用いてHTC出力信号を構成する。

<HTC>

HTCは，Actorの観測対象の制御器である。H_∞追従性能を保証するHTC出力信号u_hを次のように設計する[17]。

$$u_h=\frac{1}{8\tau^2}\mathbf{e}^T\mathbf{P}\mathbf{B} \tag{3.7}$$

ただし，τは減衰定数δによって決まる定数である。Actorが最適な制御器を見つけるまでの間，このHTCで制御を行う。この制御器は，H_∞追従性能を満たすように設計した。その証明および，定数τの導出は第3.4節で説明する。

3.3.2　Actor の構成

HTC の動作結果を観測することにより, 追従性能に優れた制御器を学習により構築する。Actor は，ASFNN[32]で構成される。Actor の ASFNN の構造を Fig.3.2 に示す。

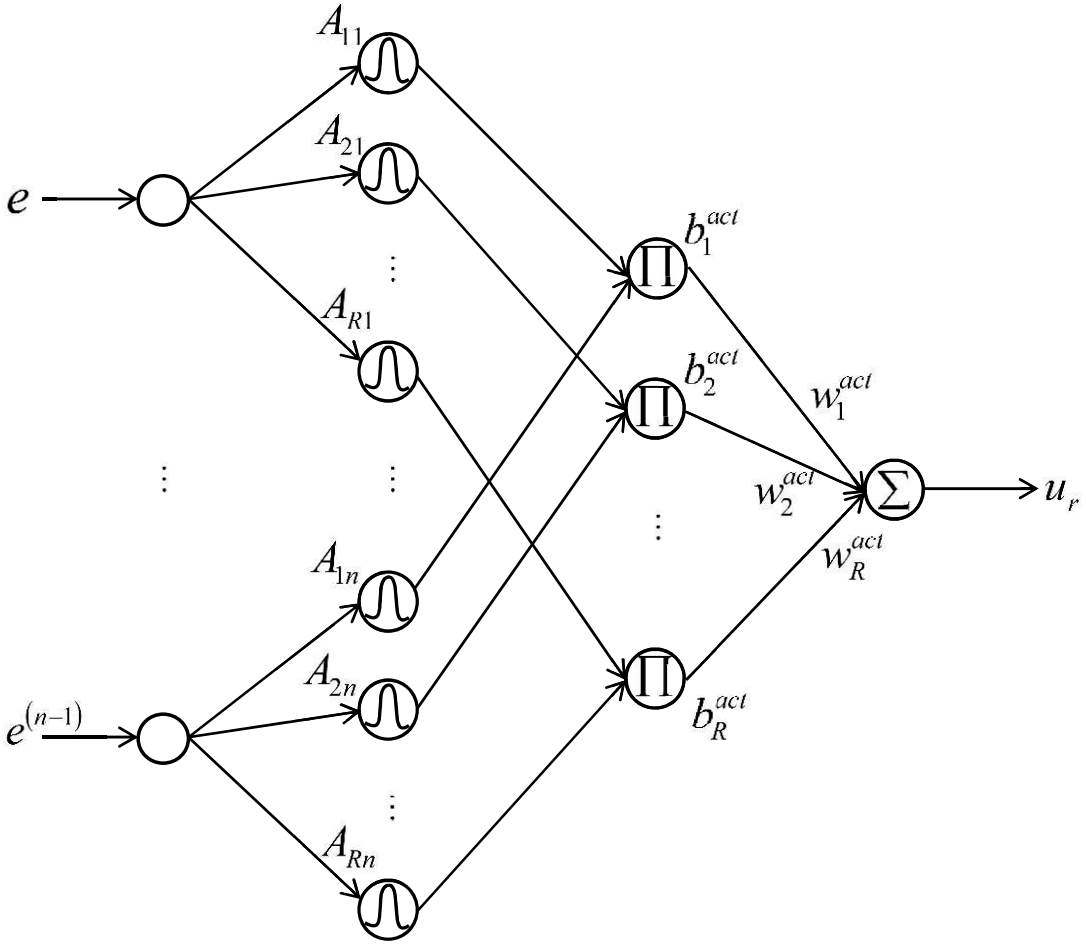

Fig.3.2.　Auto-structure fuzzy neural network

ここで，n はシステムの次数，R は Actor の NN のノード数，A_{ji}（$j=1,2,\cdots,R$，$i=1,2,\cdots,n$)はメンバーシップ関数，b_j^{act} はノード j の適合度，w_j^{act} は中間層 j ノード-出力間の結合荷重，u_r は強化学習出力信号である。

Actor の方策決定には ε - *greedy* 法を用いるため，強化学習出力信号 u_r は次式で表わされる。

$$u_r = \begin{cases} \sum_{j=1}^{R} w_j^{act} b_j^{act} & with \quad probability \quad 1-\varepsilon \\ \sum_{j=1}^{R} w_j^{act} b_j^{act} + n_s & with \quad probability \quad \varepsilon \end{cases} \tag{3.8}$$

ただし，ε は正の定数，n_s は探索ノイズである。これは 1-ε の確率で，ASFNN の出力信号をシステムの入力信号とし，ε の確率で，ASFNN の出力信号に探索ノイズ n_s を加えた信号をシステムの入力信号とすることを意味している。また，ノード j の適合度 b_j^{act} は次式で表わされる。

$$b_j^{act} = \prod_{i=1}^{I} A_{ji}\left(e^{(i-1)}\right) \tag{3.9}$$

ただし，I は入力層ノード数である。メンバーシップ関数 A_{ji} は次式のガウシアン型で表わされる。

$$A_{ji}\left(e^{(i-1)}\right) = \exp\left\{-\frac{\left(e^{(i-1)} - c_{ji}^{act}\right)^2}{\left(\sigma_{ji}^{act}\right)^2 + \varpi}\right\} \tag{3.10}$$

ただし，c_{ji}^{act} は中間層 j 番目のノードの基底関数における i 番目の入力に対する中心，$\left(\sigma_{ji}^{act}\right)^2$ は同広がり，ϖ は分母が 0 になることを防止する小さな正定数である。

提案システムは，Fig.3.2 の Actor を自己構造化し，自動的に NN のノードの追加・削除を行うことにより，効率的に最適入力を構築することを期待している。自己構造化メカニズムは次のようになる[31]~[35]。

<ノードの追加>

Actor の中間層ノードの適合度の最大値は次式で表わされる。

$$\Gamma_{\max} = \max\left(b_j^{act}\right), \quad j = 1,2,\cdots,R(t) \tag{3.11}$$

ただし，$R(t)$ は時刻 t におけるノード数である。ノードの追加条件は次式で表わされる。

$$\Gamma_{\max}(t) \le \Gamma_{th} \tag{3.12}$$

ただし，$\Gamma_{th} \in (0,1)$ はノード追加閾値である。(5.12)式を満たした時，すべてのノード適合度が，Γ_{th} 以下であることから，ASFNN のすべてのノードの出力が有効ではないと判断し，新たなノードを追加・生成する。生成された新しい $R+1$ 番目のノードの重みと，中心，分散は次のように与える。

$$R \leftarrow R+1 \tag{3.13}$$

$$w_R^{act} = w_c \tag{3.14}$$

$$c_{Ri}^{axt} = e^{(i-1)} \tag{3.15}$$

$$\sigma_{Ri}^{act} = \sigma_c \tag{3.16}$$

ただし，w_c, σ_c は規定定数である。

3.3.3 Critic の構成と TD 誤差

Critic の機能は，予測報酬 $P_v(t)$ を計算し，TD 誤差 $\hat{r}(t)$ を出力する。そして，TD 誤差が小さくなるように学習をする。その原理を以下に示す。

将来に渡って得られる報酬の和を $V(t)$ とし，以下のように定義する。

$$V(t) \equiv \sum_{n=0}^{\infty} \gamma^n \cdot r(t+n) \tag{3.17}$$

ただし，γ は割引率であり，$0 < \gamma \leq 1$ の定数である。(3.17)式は，次のように変形できる。

$$V(t) = r(t) + \gamma V(t+1) \tag{3.18}$$

ここで，$V(t)$ の予測値を $P_v(t)$ とすると，予測誤差は

$$\hat{r}(t) = r(t) + \gamma P_v(t+1) - P_v(t) \tag{3.19}$$

と計算できる。この誤差は TD 誤差とも呼ばれ，TD 誤差が学習により小さくなるように Critic は学習する。

次に，Critic の構造を示す。Critic は NN で構成され，Fig.3.3 に Critic の構造を示す。

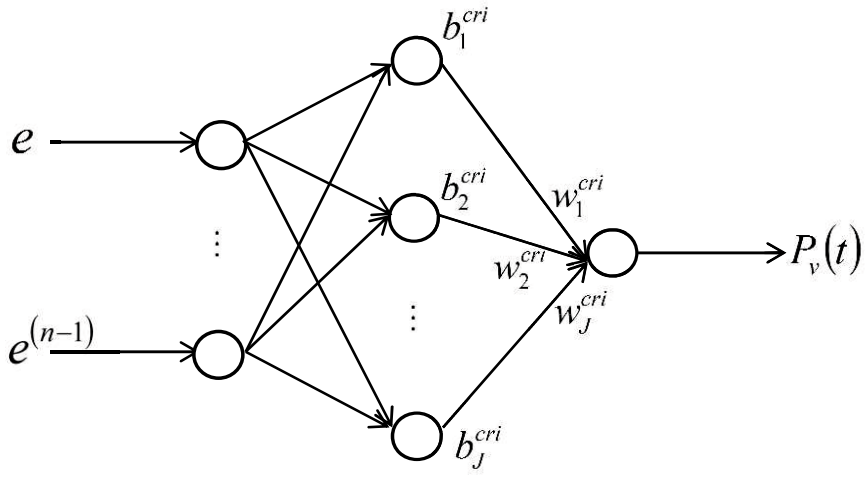

Fig.3.3.　Structure of Critic

ただし，J は Critic の中間層ノードの数，w_j^{cri} ($j=1,2,\cdots,J$)は中間層 j ノード-出力間の結合荷重，b_j^{cri} は中間層ノードの出力である。

予測報酬 $P_v(t)$ は次式で計算される。

$$P_v(t)=\sum_{j=1}^{J} w_j^{cri} b_j^{cri} \tag{3.20}$$

また，Critic の基底関数 b_j^{cri} は tanh 関数で構成する。tanh 関数 $f_{\tanh}(x)$ は次式で表わされる。

$$f_{\tanh}(x)=\frac{1-\exp(-x)}{1+\exp(-x)} \tag{3.21}$$

提案システムは NN への入力がベクトルであるため，実際の基底関数は次式で表される。

$$b_j^{\tanh}\left(e^{(i-1)}\right)=\frac{1-\exp\left(-\sum_{i=1}^{I} v_{ji}^{cri} e^{(i-1)}\right)}{1+\exp\left(-\sum_{i=1}^{I} v_{ji}^{cri} e^{(i-1)}\right)} \tag{3.22}$$

ただし，I は入力層の数，v_{ji}^{cri} ($j=1,2,\cdots,J$, $i=1,2,\cdots,I$)は i 番目入力-中間層 j ノード間の結合荷重である。

3.3.4 学習アルゴリズム

Critic は，予測誤差を小さくするように学習を行い，Actor は予測報酬を大きくするように学習を行う。その結果，Actor は，実際に高い報酬を得られる出力特性を獲得する。本研究では，NN の学習では一般的な Back Propagation(BP)を用いて学習する。具体的には，以下の式に従って関数近似器の重みを更新することにより，学習を行う。

Actor の学習

$$\Delta w_j^{act} = \eta_w^{act} \cdot \frac{\partial u}{\partial w_j^{act}} \cdot \hat{r} \tag{3.23}$$

$$\Delta c_{ji}^{act} = \eta_c^{act} \cdot \frac{\partial u}{\partial c_{ji}^{act}} \cdot \hat{r} \tag{3.24}$$

$$\Delta \sigma_{ji}^{act} = \eta_\sigma^{act} \cdot \frac{\partial u}{\partial \sigma_{ji}^{act}} \cdot \hat{r} \tag{3.25}$$

Critic の学習

$$\Delta w_j^{cri} = -\eta_w^{cri} \cdot \frac{\partial P_v}{\partial w_j^{cri}} \cdot \frac{\partial}{\partial P_v}\left(\frac{1}{2}\hat{r}^2\right) \tag{3.26}$$

$$\Delta v_{ji}^{cri} = -\eta_v^{cri} \cdot \frac{\partial P_v}{\partial v_{ji}^{cri}} \cdot \frac{\partial}{\partial P_v}\left(\frac{1}{2}\hat{r}^2\right) \tag{3.27}$$

ただし，*cri* は Critic に関する，*act* は Actor に関するパラメータを表し，$\eta_w^{act}, \eta_c^{act}, \eta_\sigma^{act}, \eta_w^{cri}, \eta_v^{cri}, \eta_c^{cri}, \eta_\sigma^{cri}$ は学習率である。

Critic については，$\hat{r}^2/2$ を評価関数として通常の BP を行っているが，Actor は $\hat{r}$ を誤差信号とみなす事により BP を行っている。ここには，フィードバック誤差学習の概念を利用している。即ち，学習が進むにつれて $\hat{r}(t)$ が小さくなってゆき，Actor の関数近似器の学習が収束することを期待している。

3.4　提案システムの安定性解析

3.4.1　H_∞追従性能

追従誤差ベクトルを$\mathbf{e}$，近似誤差や外乱などの不確かさの総和をε_t，制御の最終時刻をt_fとすると，システムのL_2-ゲインは，

$$\|G\|_\infty = \sup_{\varepsilon_t \in L_2[0,t_f]} \frac{\|\mathbf{e}\|_2}{\|\varepsilon_t\|_2} \tag{3.28}$$

H_∞制御問題は，システムのL_2-ゲインを 0 でない正定数δ以下にする制御器を見つける問題であるため，

$$\sup_{\varepsilon_t \in L_2[0,t_f]} \frac{\|\mathbf{e}\|_2}{\|\varepsilon_t\|_2} \le \delta \tag{3.29}$$

これは，追従誤差が正定数δで抑えられることを意味しており，δが小さいほど小さい誤差で有界になることが分かる。これがH_∞追従性能を表わす。

<定理 1>

次式はH_∞追従性能(3.29)式を定義することができ，H_∞追従性能を表わす[17],[18]。

$$\int_0^{t_f} \mathbf{e}^T \mathbf{Q} \mathbf{e} dt \le \mathbf{e}^T(0)\mathbf{P}\mathbf{e}(0) + \delta^2 \int_0^{t_f} \varepsilon_t^T \varepsilon_t dt \tag{3.30}$$

ここで，t_fは制御の最終時刻，ε_tは近似誤差や外乱などの不確かさの総和，δは 0 でない正定数，$\mathbf{Q}$は正定対称行列，$\mathbf{P}$は重み行列で$\mathbf{P} = \mathbf{P}^T > 0$である。

<定理 1 の証明>

論文[18]と同様の方法でこの定理 1 を証明する。L_2-ノルムおよび$\mathbf{Q}$が正定対称行列であることから次の式が成り立つ。

$$\|\mathbf{e}\|_{\mathbf{Q}2}^2 = \int_0^{t_f} \mathbf{e}^T \mathbf{Q} \mathbf{e} dt \tag{3.31}$$

$$\|\varepsilon\|_2^2 = \int_0^{t_f} \varepsilon_t^T \varepsilon_t dt \tag{3.32}$$

ここで，システムの初期状態を$\mathbf{e}(0) = \mathbf{0}$とし，(3.31)式，(3.32)式を(3.30)式に代入すると

$$\|\mathbf{e}\|_{\mathbf{Q}2}^2 \le \delta^2 \|\varepsilon_t\|_2^2 \tag{3.33}$$

整理すると

$$\sup \frac{\|\mathbf{e}\|_{\mathbf{Q}2}}{\|\varepsilon_t\|_2} \le \delta \tag{3.34}$$

ただし，$\varepsilon_t \in L_2\lfloor 0, t_f \rfloor$である。したがって，(3.30)式は$H_\infty$追従性能(3.29)式を定義でき，<定理 1>が証明された。□

3.4.2 安定性解析

<定理 2>

HTC の出力信号を(3.7)式と定義したとき，提案システムはH_∞追従性能を満たし，システムの安定性が保証される。

<定理 2 の証明>

論文[17]と同様な方法で以下，提案システムの安定性解析をH_∞追従性能とリヤプノフ関数を用いて行う。(3.4)式より

$$r_f^{(n)} = f + gu^* + d - \mathbf{k}^T\mathbf{e} \tag{3.35}$$

また，(3.1)式，(3.35)式より，n 階微分の追従誤差は，

$$\begin{aligned} e^{(n)} &= r_f^{(n)} - x^{(n)} \\ &= \left(f + gu^* + d - \mathbf{k}^T\mathbf{e}\right) - \left(f + gu + d\right) \\ &= -\mathbf{k}^T\mathbf{e} + g\left(u^* - u\right) \end{aligned} \tag{3.36}$$

となる。これを状態空間表現すると

$$\dot{\mathbf{e}} = \Lambda\mathbf{e} + \mathbf{B_1}\left(u^* - u\right) \tag{3.37}$$

ただし，

$$\Lambda = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ -k_n & -k_{n-1} & \cdots & -k_2 & -k_1 \end{bmatrix}, \quad \mathbf{B_1} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ g \end{bmatrix}$$

である。提案システムの制御入力は次式で表わされる。

$$\boldsymbol{u} = \boldsymbol{u}_r + \boldsymbol{u}_h \tag{3.38}$$

u_rは強化学習出力信号であり，u_hはHTCの出力信号である。これを(3.37)式に代入すると，

$$\dot{\mathbf{e}} = \Lambda\mathbf{e} + \mathbf{B_1}\left(u^* - u_r - u_h\right) \tag{3.39}$$

ここで，強化学習信号u_rと最適入力u^*の近似誤差を

$$\boldsymbol{\varepsilon}_u = \boldsymbol{u}^* - \boldsymbol{u}_r \tag{3.40}$$

と定義する。これを(3.39)式に代入すると，

$$\dot{\mathbf{e}} = \Lambda\mathbf{e} + \mathbf{B_1}\left(\varepsilon_u - u_h\right) \tag{3.41}$$

リヤプノフ関数として

$$V = \mathbf{e}^T\mathbf{P}\mathbf{e} \tag{3.42}$$

を考える。両辺を時間微分すると

$$\dot{V} = \dot{\mathbf{e}}^T\mathbf{P}\mathbf{e} + \mathbf{e}^T\mathbf{P}\dot{\mathbf{e}} \tag{3.43}$$

(3.41)式を代入すると

$$\begin{aligned}\dot{V} &= \left(\mathbf{e}^T\mathbf{P}\Lambda\mathbf{e} + \mathbf{e}^T\mathbf{P}\mathbf{B_1}\left(\varepsilon_u - u_h\right)\right) + \left(\mathbf{e}^T\Lambda^T\mathbf{P}\mathbf{e} + \mathbf{B_1}^T\left(\varepsilon_u - u_h\right)\mathbf{P}\mathbf{e}\right) \\ &= \mathbf{e}^T\left(\mathbf{P}\Lambda + \Lambda^T\mathbf{P}\right)\mathbf{e} + 2\mathbf{e}^T\mathbf{P}\mathbf{B_1}\left(\varepsilon_u - u_h\right)\end{aligned} \tag{3.44}$$

リヤプノフ方程式(3.6)式を用いると

$$\begin{aligned}\dot{V} &= -\mathbf{e}^T\mathbf{Q}\mathbf{e} + 2\mathbf{e}^T\mathbf{P}\mathbf{B_1}\left(\varepsilon_u - u_h\right) \\ &= -\mathbf{e}^T\mathbf{Q}\mathbf{e} - 2\mathbf{e}^T\mathbf{P}\mathbf{B_1}u_h + 2\mathbf{e}^T\mathbf{P}\mathbf{B_1}\varepsilon_u \\ &= -\mathbf{e}^T\mathbf{Q}\mathbf{e} - 2\mathbf{e}^T\mathbf{P}\mathbf{B}gu_h + 2\mathbf{e}^T\mathbf{P}\mathbf{B}g\varepsilon_u\end{aligned} \tag{3.45}$$

HTCの出力信号(3.7)式を代入すると

$$\begin{aligned}\dot{V} &= -\mathbf{e}^T\mathbf{Q}\mathbf{e} - \frac{g}{4\tau^2}\left(\mathbf{e}^T\mathbf{P}\mathbf{B}\right)^2 + 2\mathbf{e}^T\mathbf{P}\mathbf{B}g\varepsilon_u \\ &= -\mathbf{e}^T\mathbf{Q}\mathbf{e} - \left\{\left(\frac{\sqrt{g}}{2\tau}\right)\mathbf{e}^T\mathbf{P}\mathbf{B} - 2\tau\left(\frac{g}{\sqrt{g}}\right)\varepsilon_u\right\}^2 + \left(2\tau\frac{g}{\sqrt{g}}\right)\varepsilon_u^T\left(2\tau\frac{g}{\sqrt{g}}\right)\varepsilon_u\end{aligned} \tag{3.46}$$

したがって，

$$\dot{V} \le -\mathbf{e}^T\mathbf{Q}\mathbf{e} + \left(2\tau\frac{g}{\sqrt{g}}\right)\varepsilon_u^T\left(2\tau\frac{g}{\sqrt{g}}\right)\varepsilon_u \tag{3.47}$$

両辺を時間領域$[0, t_f]$で積分すると

$$V(t_f) - V(0) \le -\int_0^{t_f}\mathbf{e}^T\mathbf{Q}\mathbf{e}dt + \left(2\tau\frac{g}{\sqrt{g}}\right)^2\int_0^{t_f}\varepsilon_u^T\varepsilon_u dt \tag{3.48}$$

$V(t_f) \ge 0, V(0) = \mathbf{e}^T(0)\mathbf{P}\mathbf{e}(0)$より，

$$\int_0^{t_f}\mathbf{e}^T\mathbf{Q}\mathbf{e}dt \le \mathbf{e}^T(0)\mathbf{P}\mathbf{e}(0) + \left(2\tau\frac{g}{\sqrt{g}}\right)^2\int_0^{t_f}\varepsilon_u^T\varepsilon_u dt \tag{3.49}$$

H_∞追従性能の(3.30)式と比較して

$$\delta = 2\tau\frac{g}{\sqrt{g}} \tag{3.50}$$

とすれば，(3.49)式はH_∞追従性能を表わしている。これは，追従誤差ベクトル$\mathbf{e}$が近似誤差ε_uと減衰定数δを掛けた値が有界になることを示している。つまり，近似誤差と減衰定数が有界であれば，追従誤差も有界である。よってシステムの安定性が証明された。□

しかし，gは未知の連続関数なので次のように仮定する。

$$0 < g_{\min} \le g \le g_{\max} \tag{3.51}$$

ここで，$g_{\min}$はgの下限，$g_{\max}$はgの上限である。このとき，(3.50)式は

$$\delta' = 2\tau\frac{g_{\max}}{\sqrt{g_{\min}}} \ge 2\tau\frac{g}{\sqrt{g}} = \delta \tag{3.52}$$

なる関係がある。

3.4.3　強化学習信号の有界性

(3.49)式の安定性解析は，近似誤差ε_uと減衰定数δが有界であるという前提で成り立つ。減衰定数δは，(3.51)式，(3.52)式より，有界である。近似誤差ε_uは，(3.40)式で表わされ，最適入力u^*は，(3.4)式，(3.5)式より有界である。しかし，強化学習信号u_r((3.8)式)をより，有界にするためには，ノードjの適合度b_j^{act}は(5.9)式より$0 \le b_j^{act} \le 1$なので，中間層jノード-出力間の結合荷重w_j^{act}が有界でなければならない。

w_j^{act}を有界にするために，射影法[19],[21]を導入し，結合荷重の範囲を定める。学習により更新される量をΔw_j^{act}とする。

$$w_j^{act} = \begin{cases} w_j^{act} + \Delta w_j^{act} & if \quad \left| w_j^{act} + \Delta w_j^{act} \right| < D \\ w_j^{act} & otherwise \end{cases} \tag{3.53}$$

ただし，Dは結合荷重w_j^{act}の有効範囲である。この式は，学習後の結合荷重w_j^{act}が絶対値D未満であれば，更新することを意味している。

3.5　計算機シミュレーション

3.5.1　台車付き倒立振子

制御対象は，次のダイナミクスで表わされる台車付き倒立振子である。

$$f = \frac{g_r \sin\theta - \dfrac{mL\dot{\theta}_2^2 \sin\theta\cos\theta}{m_c + m}}{L\left(\dfrac{4}{3} - \dfrac{m\cos^2\theta}{m_c + m}\right)} \tag{3.47}$$

$$g = \frac{\dfrac{\cos\theta}{m_c + m}}{L\left(\dfrac{4}{3} - \dfrac{m\cos^2\theta}{m_c + m}\right)} \tag{3.48}$$

ここで，θは振子の角度，$\dot{\theta}$は振子の角速度，m_cは台車の質量，m_pは振子の質量，Lは振子の長さ，g_rは重力加速度である。また，状態変数を$\mathbf{x} = [x_1, x_2]^T = [\theta, \dot{\theta}]^T$とする。

制御の目的は，制御入力uを台車に与え，状態変数$\mathbf{x}$を目標信号$\mathbf{r}$に追従させることである。提案システムの追従性能を検証するため，論文[17]のH_∞追従性能補償器を備えた適応ファジィ制御(以下 AFC と呼ぶ)，及び論文[32]の自己構造型ファジィニューラル制御システム(ASFNCS と呼ぶ)との性能比較シミュレーションを行った。なお，従来のロバスト強化学習システムは，状態数が多くなることや，学習の速さの理由で倒立振子シミュレーションを行うことができない。なお，提案システムの Critic の基底関数は tanh 関数を用いる。以降のシミュレーションでもそれは同様である。

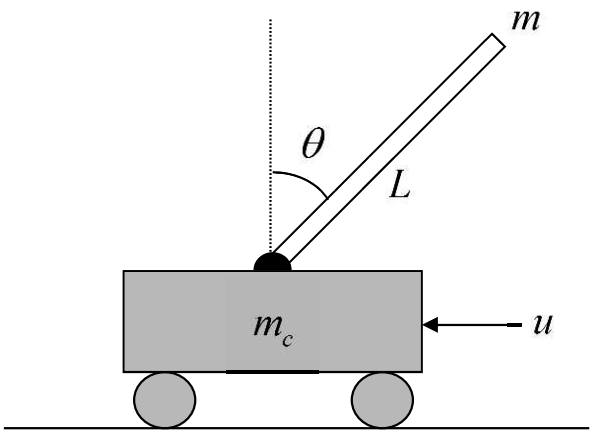

Fig.3.4.　Cart-Pole system

台車付き倒立振子のパラメータの値は文献[56]同様，m_c=1.0[kg]，m_p=0.1[kg]，L=1.0[m]，g_r=9.8[m/s^2]とした。また，初期角度-45 度(≒-0.785[rad])，初期角速度 0[rad/s]の状態で，制御時間は 30.0 秒，サンプリングタイムは 0.01 秒，目標信号は$r=\sin(t)$とした。

システムの性能評価として，制御中の角度の総誤差面積に対する，制御の単位時間当たりの誤差面積(平均誤差面積)と角度の 5～30 秒までの総誤差面積に対する，制御の単位時間当たりの定常偏差(平均定常偏差)を求め，検証した。

$$\int_0^{t_f} |e_t| dt \Big/ t_f \tag{3.49}$$

$$\int_{t_i}^{t_f} |e_t| dt \Big/ (t_f - t_i) \tag{3.50}$$

ただし，t_fは制御時間で，t_iは定常偏差の測定開始時刻である。ここでは，t_f=30，t_i=5 である。

シミュレーションで使用した提案システムのパラメータは，比較する文献[17]，[32]と同様に，k_1=5.0, k_2=1.0, $\mathbf{Q}$=diag{10.0, 10.0}, δ=0.3, $g_{\min}$=0.6, $g_{\max}$=1.5，I=2, J=5, Γ_{th}=0.3, σ_c=0.2, I_c=1.0, P_{th}=0.1, β=0.01, I_{th}=0.01，ϖ=0.001, γ=0.95, η_w^{cri}=0.01, η_v^{cri}=0.01, η_w^{act}=10.0, η_c^{act}=0.01, η_σ^{act}=0.01, α_θ=5.0, $\alpha_{\dot\theta}$=1.0, α_ε=0.1, D=500.0, n_s=[-10.0，10.0]の一様乱数，w_j^{cri}, v_{ji}^{cri}(i=1,…, I),(j=1,2,…, J)の初期値は [-1.0, 1.0]の一様乱数とした。また，(3.50)式よりτ ≒0.077 となる。Actor の初期ノード数はR=1 とし，そのノードのパラメータw_j^{act}，c_{ji}^{act}，σ_{ji}^{act}(i=1,…, I),(j=1,2,…, R)の初期値は(3.10)式により決定した。

3.5.2 シミュレーション結果

Fig.3.5 に各システムの振子の角度θの推移, Table.3.1 に各システムの角度の平均誤差面積と平均定常偏差，Fig.3.5 に各システムの角度の推移，Fig.3.6 に各システムの角度の追従誤差の推移，Fig.3.7 に Fig.3.6 の 0～5 秒間の拡大図を示す。さらに，Fig.3.8 に提案システムの TD 誤差，Fig.3.9 に Fig.3.8 の拡大図，Fig.3.10 に提案システムの Actor と HTC の出力の推移，Fig.3.11 に提案システムの Actor のノード数の推移，Fig.3.12 に提案システムの Actor の結合荷重のノルムの推移，Fig.3.13 に Actor のガウシアン型ノードの中心及び広がり(いずれもノルム)の推移を示す。従来法との性能比較において，Fig.3.6，Fig.3.7 より提案システム(RRLCS)は，AFC，ASFNCS に比べ，オーバーシュートが小さく，急速に，目標信号$\mathbf{0}$に近づいたため，誤差が小さくなった。これは，優れたH_∞追従性能を持つ HTC と学習により最適化された強化学習システムの Actor との補完しあう制御器の実現によるものと考えられる。即ち，Fig.3.8，Fig.3.9 の TD 誤差の減少により最適な学習が行われていることが分かる。さらに Fig.3.10 より，提案システムは制御初期段階では，HTC が振子の制御をしており，およそ 2 秒後に HTC の出力が 0 になり，Actor が振子の制御をしていることが分かる。

この結果から，Actor が 2 秒間で最適な NN の構築に成功したと言える。定量的な観点では，Table.2.1 より，角度の平均誤差面積・平均定常偏差は，提案システムが最も小さく，続いて ASFNCS，AFC の順であり，提案システムが従来法より優れていることが分かる。また，Fig.3.11～Fig.3.13 より，Actor のノード数や各 NN に関するパラメータのノルムが制御開始後，速やかに定常値へ収束していることから，学習が早い段階で完了し，Actor が最適な制御器の構築に成功したことが分かる。しかし，Fig.3.11 の Actor の出力でチャタリングが発生していることから，ノイズによる最適探索が続いていることが分かる。さらにそれは，Fig.3.11 の Actor のノード数や Fig.3.12, Fig.3.13 のノルムの増減が続いている事からも言える。

Table 3.1. Comparison of error areas and steady state errors of the angle among three methods

	RRLCS (Proposed)	AFC[17]	ASFNCS[32]
Error area of the angle	7.0×10^{-3}	1.3×10^{-2}	7.8×10^{-3}
Steady state errors of the angle	5.1×10^{-4}	1.2×10^{-3}	2.7×10^{-3}

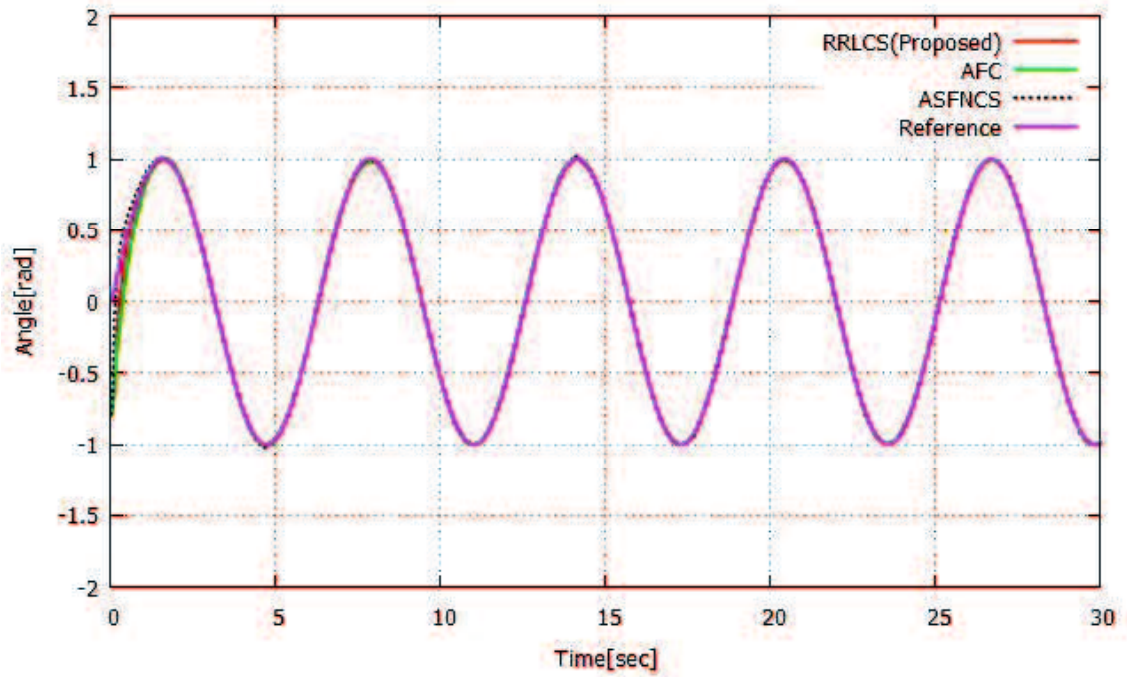

Fig.3.5. Comparison of control results of the angle among three methods

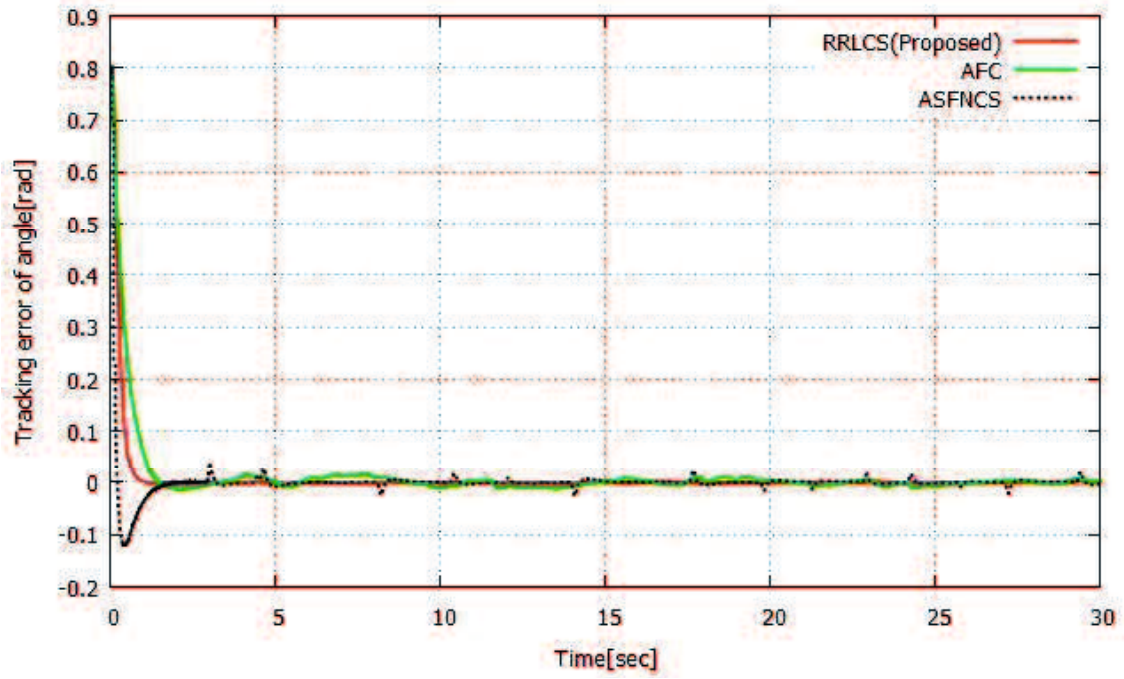

Fig.3.6. Comparison of the tracking errors of the angle among three methods

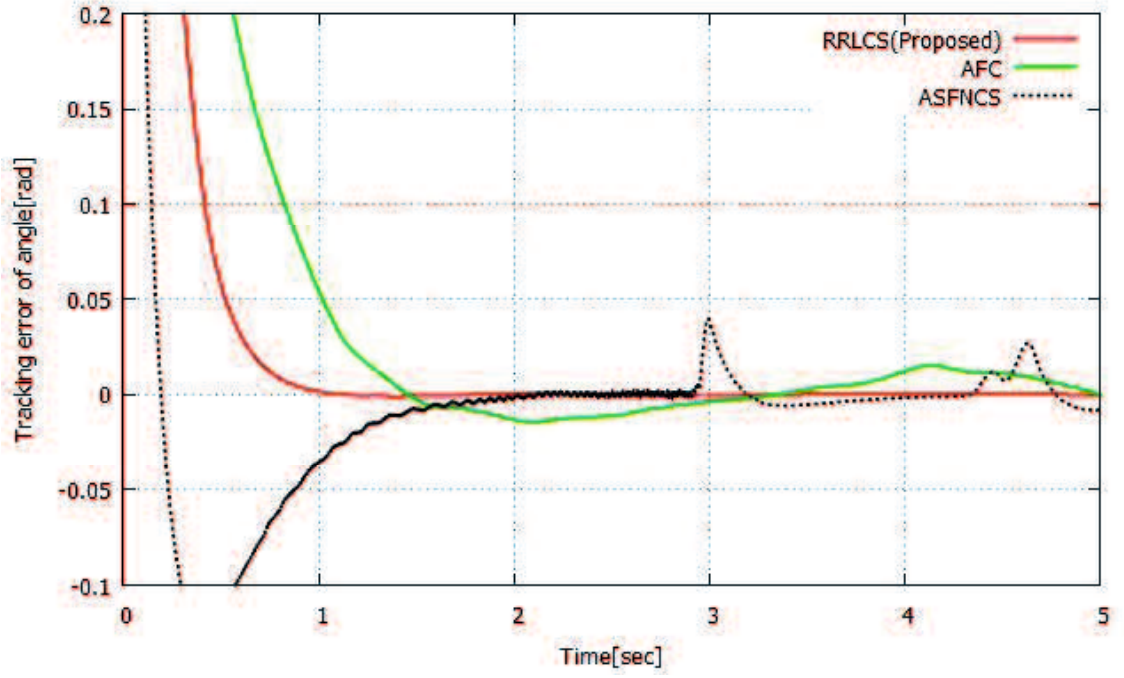

Fig.3.7. The enlarged view of Fig.3.6

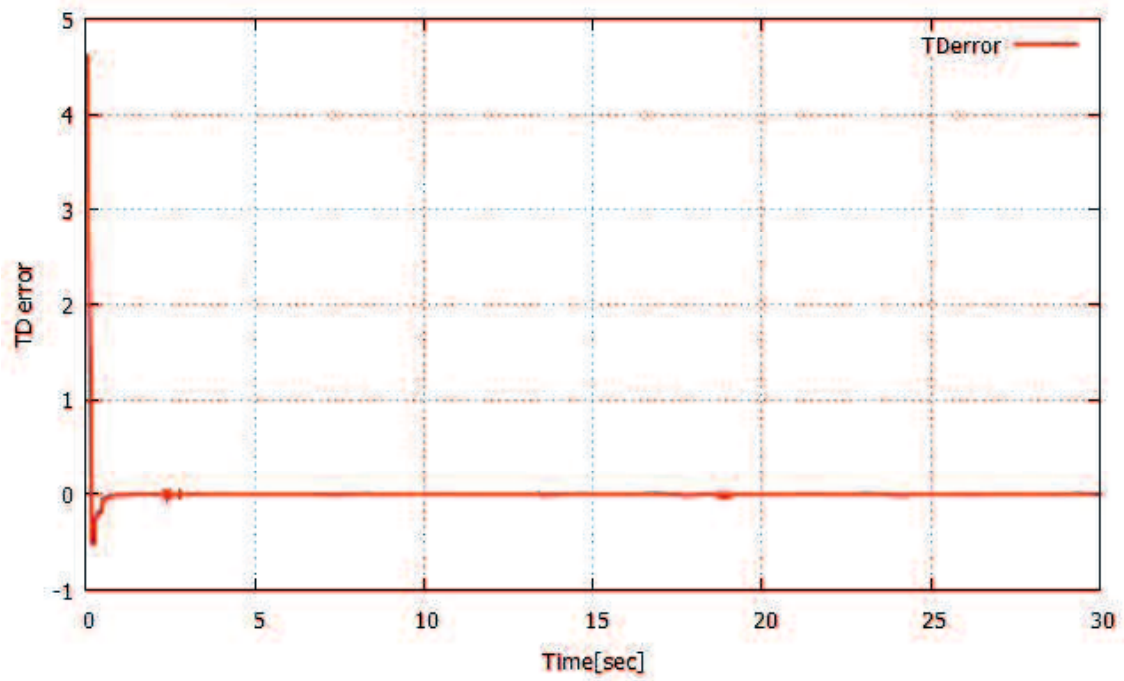

Fig.3.8. TD error of the proposed system

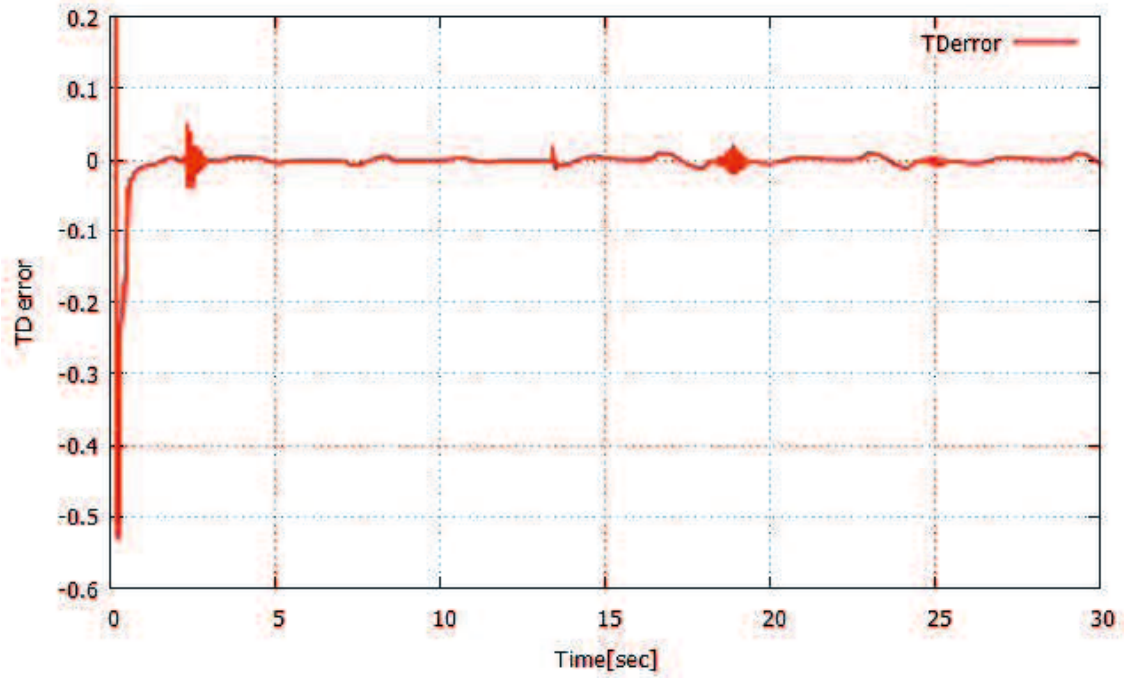

Fig.3.9. The enlarged view of Fig.3.8

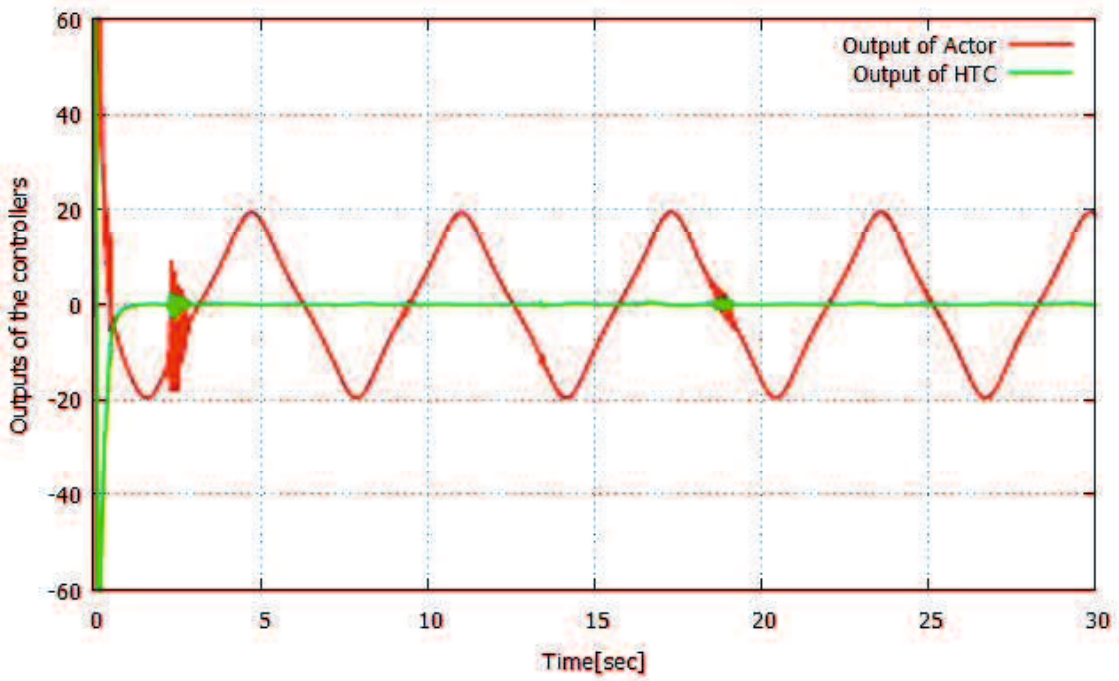

Fig.3.10. The outputs of the Actor and HTC of the proposed system

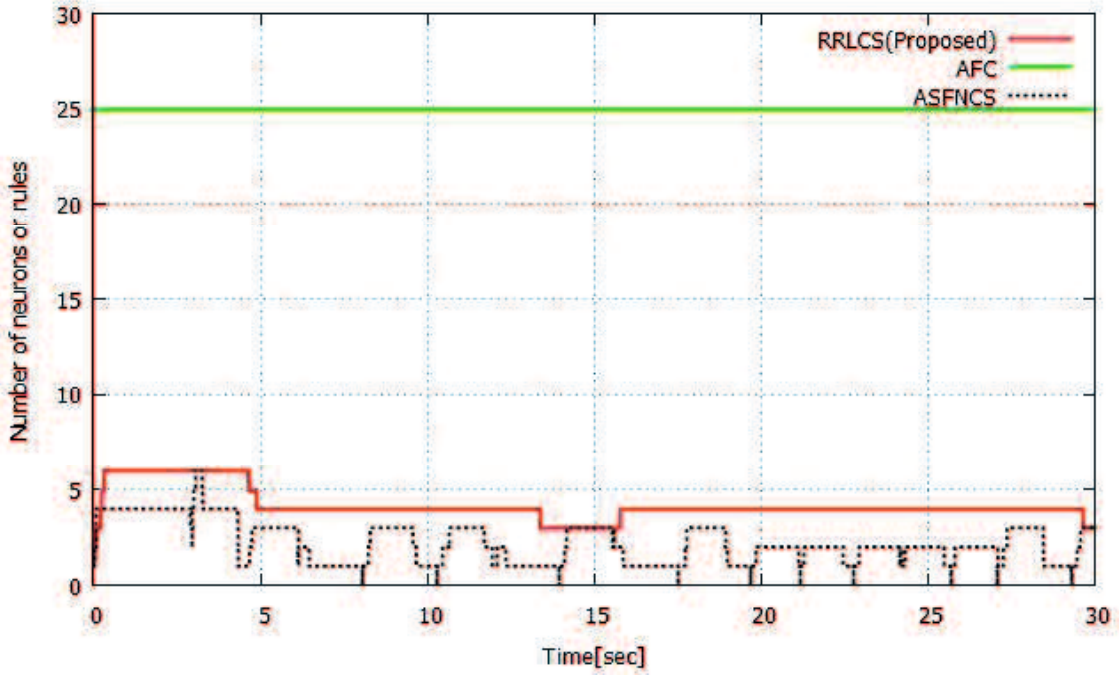

Fig.3.11. Comparison of the number of the neurons among three methods

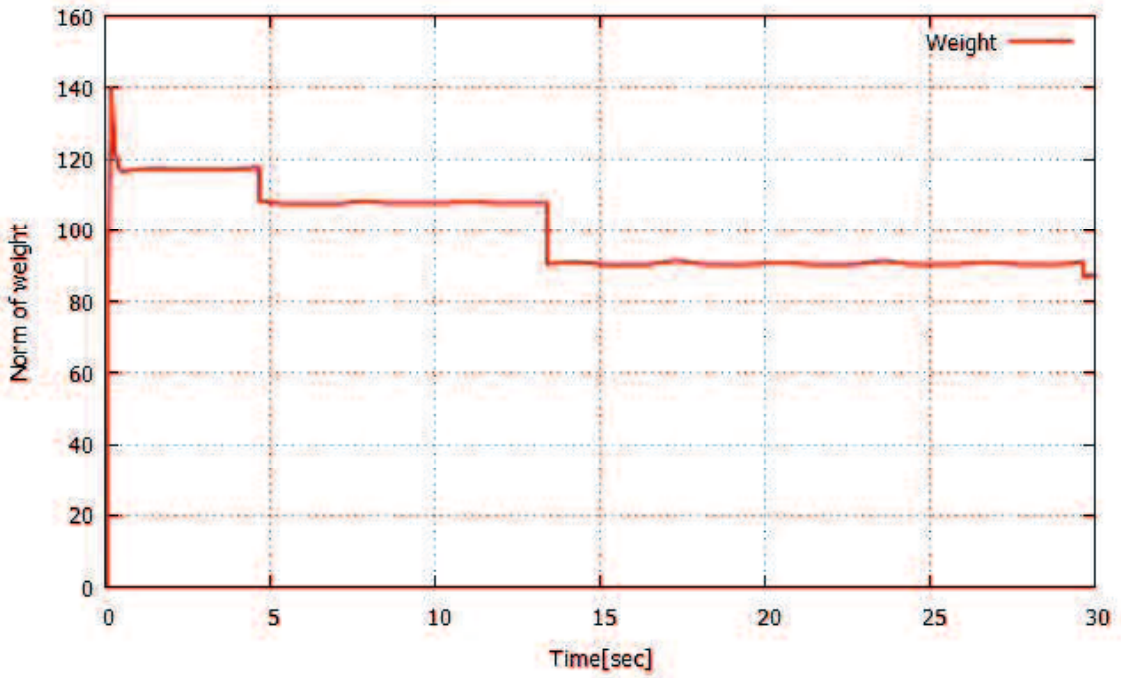

Fig.3.12. The norm of weight vector of Actor of proposed system

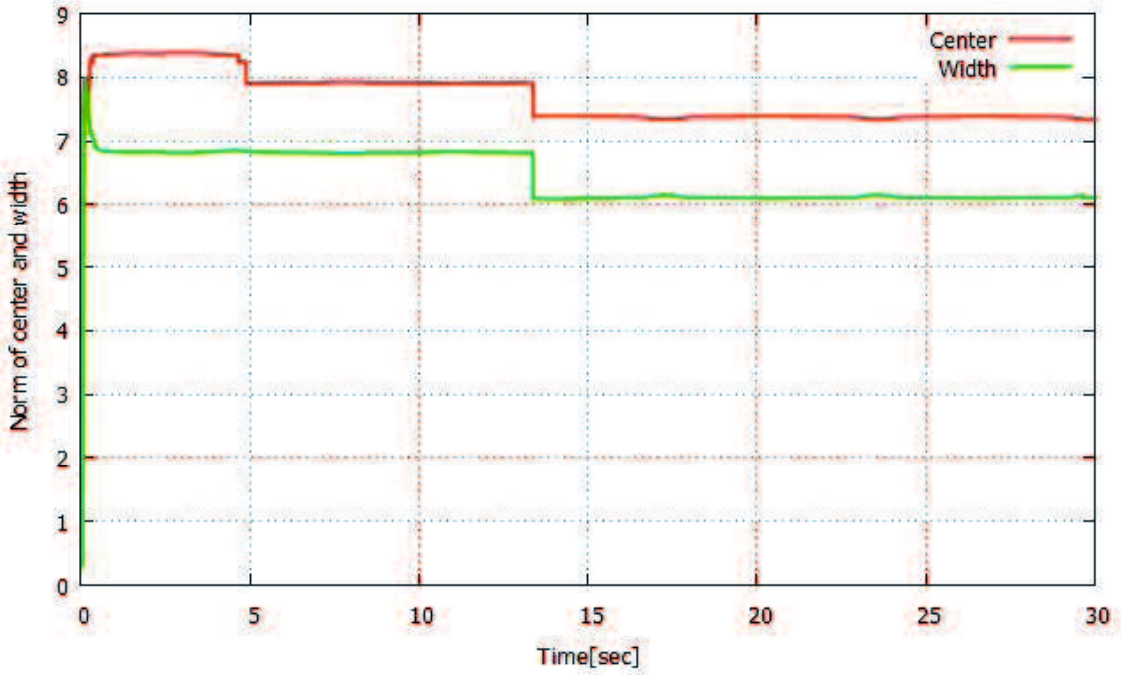

Fig.3.13. The norm of center and width vector of Actor of proposed system

3.6 まとめ

H_∞追従性能補償器を備えたリアルタイム強化学習システムを提案した。さらに，H_∞制御理論の概念とリヤプノフ方程式を用いて，提案システムの安定性も示した。そして，台車付き倒立振子シミュレーションにより，追従性能により提案システムの有効性を示した。

今後の展開として，本論文では提案システムの安定性を証明した際，未知関数の動作範囲を仮定しており，完全な未知関数を持つシステムを対象にしたとは言えない。そこで，未知関数に関する一切の情報を必要としないシステムに対する安定性を保証する制御が必要である。さらに，本研究の倒立振子シミュレーションは移動距離(台車の位置)を考慮していないため，汎用性を高めるべく，移動距離を考慮した台車付き倒立振子システムにおける，角度と距離を考慮した制御法の開発が考えられる。

第 4 章　フィードバック制御における

小脳パーセプトロン改良モデル利用型ロバスト制御システム

4.1　はじめに

小脳モデルに関する先行研究として，Albus が提案し，小脳パーセプトロンモデルで知られる CMAC(Cerebellar Model Articulation Controller)[54]と呼ばれる神経回路モデルがある。CMAC の利点は，ルックアップテーブル(LUT)を扱う単純な構造の局所的なニューラルネットワーク(NN)であるため，学習が速く，ハードウェアによる実行が容易であることである[50]。また，Almeida らは，CMAC の特徴である LUT をそのまま用いてネットワーク入力強度を表わす線形パラメトリック式やファジィ理論を用いたパラメトリック CMAC(PCMAC)を提案している[74]。その大きな強みは，線形項の追加で得られる CMAC 以上の高い近似能力にある。

しかし，CMAC および PCMAC は局所的表現の LUT を用いるため，その汎化能力は訓練された行動のごく近傍に限られる。また，状態数が多い場合，これに対処するため膨大なメモリ(ニューロン数)が必要になる。Lu らは，メモリの自己生成機能を導入したロバスト PCMAC (RPCSGD)を提案し[56]，従来の PCMAC のメモリを抑え，必要に応じて自動的に増加させることに成功している。しかし，メモリが増加し過ぎた場合，LUT が大きくなるためデータ領域を圧迫し，また，入力データが LUT の範囲外の場合，近似誤差が大きくなりやすい欠点を有している。

筆者らは，メモリ数を最低限に抑え，制御性能を向上させ，かつ，モデルフリーでロバスト性とシステムの安定性を保証し，事前にオフライン学習を必要としないリアルタイム強化学習制御システム(RRLCS)を既に提案している[43]。そこでは，RRLCS が，追従誤差の点で，H_∞追従性能補償器を備えたファジィ適応制御器，及び自己構造型ファジィニューラルネットワーク制御システムより優れた制御性能・ロバスト性を示している。しかしながら，システムを構成する NN の結合荷重の設定範囲を仮定したため，完全なモデルフリーと言えず，システムの安定性はある条件下での保証に限定されている。本論文では，Lu らが提案した PCMAC において，LUT の代わりに，小脳パーセプトロン(CP)を用いることを提案する。即ち，先述した小脳の記憶の機能に着目し，その概念を PCMAC に導入することで，制御システムの汎化能力を高め，メモリ数を抑える。安定性解析においては，RRLCS で仮定した NN の結合荷重の設定範囲を制限しない完全なモデルフリーの制御システムを提案する。

また，制御対象に適切な動作を行わせるため，制御対象の状態に応じて，制御に必要な

情報を想起させる，即ち，パーセプトロンにおいては必要なニューロン(メモリ)のみを連結して結合荷重を強化させる。また，対応するニューロンがない場合，制御対象の状態の情報をもとに新たに追加し，逆に，一定時間参照されないニューロンは削除する(以下，自己構造型と呼ぶ)。最終的に，強化されたニューロンのみを記憶・想起させることで，制御対象に適切な動作をスムーズに行わせるネットワークを構築する。このような CP を用いる，「小脳パーセプトロン改良モデル利用型ロバスト制御システム(CPRCS)」を提案する。そして，台車付き倒立振子による計算機シミュレーションにより，従来システムの RRLCS[43]及び RPCSGD[56]と提案システムとの性能比較を行い，提案システムの有効性を示す。

4.2 制御対象の定式化

次のn次非線形システムを制御対象とする。

$$x^{(n)} = f(\mathbf{x}) + g(\mathbf{x})u \tag{4.1}$$

ここで，$x^{(n)}$はxのn階の時間微分，$\mathbf{x} = \left[x, \dot{x}, \cdots, x^{(n-1)}\right]^T = \left[x_1, x_2, \cdots, x_n\right]^T$はシステムの状態変数ベクトル，$f, g$は未知でスカラーの連続関数(ただし$g > 0$)，$u$は制御入力である。

このシステムは，状態変数ベクトル$\mathbf{x}$を目標信号ベクトル$\mathbf{r} = \left[r, \dot{r}, \cdots, r^{(n-1)}\right]^T$に追従させる，つまり，追従誤差ベクトル$\mathbf{e} = \mathbf{r} - \mathbf{x}$を$\mathbf{0}$にすることを目的としている。ここで，関数$f, g$が既知であるとき，システムの最適入力は次式で表わされる。

$$u^* = g^{-1}\left(-f + r^{(n)} + \mathbf{k}^T\mathbf{e}\right) \tag{4.2}$$

ここで，$\mathbf{k} = \left[k_n, k_{n-1}, \cdots, k_1\right]^T$はフィードバックゲインベクトル，$r^{(n)}$は目標信号$r$の$n$階時間微分である。この最適入力を(3.1)式に代入すると次式を得る。

$$e^{(n)} + k_1 e^{(n-1)} + \cdots + k_{n-1}\dot{e} + k_n e = 0 \tag{4.3}$$

ここで，k_i (i=1,2,…,n)は(3.3)式に対し，フルビッツの安定性を満たすように決定する。そのように決定した場合，(3.2)式を用いれば，(3.3)式より，$\lim_{t\to\infty} e_t = 0$になることが分かる。しかし，実際には$f, g$は未知なので，CP を用いて，最適入力$u^*$を近似する。

4.3 提案する小脳パーセプトロン改良モデル

提案する CP 改良モデル(以下 CP)を Fig.3.1 に示す。従来システムでは，入力変数I_i (i=1,2,…,n)を量子化したデータを用いている。しかし，提案する CP では，入力変数を量子化せず，直接入力することで，高精度のシステムを実現する。Fig.4.1 の点線内を 1 つのニューロンとみなし，それぞれが受容野空間(Receptive-Field Space)とパラメータ空間(Parameter Space)を持つものとする。また，同図において，グレーで塗られたニューロンは，

入力に対応した連結されたニューロン群を表わす。提案する CP の特徴として，この連結処

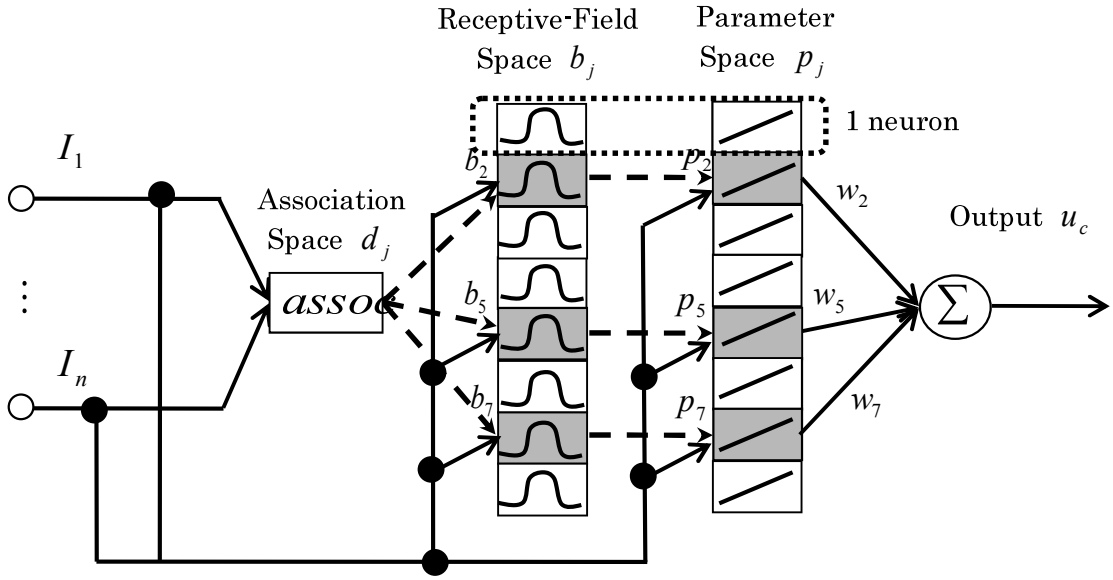

Fig.4.1.　Structure of the cerebellar perceptron improved model

理を行う連結空間(Association Space)で，入力変数を用いて，現在の制御対象の状態を観測することにより，現状態に対応する連結ニューロン群を決定する。

CP の一連の処理概要は次の通りである。まず入力変数を用いて，連結空間にて，受容野空間のガウシアン関数の中心c_{ij}との距離を計算し，連結ニューロンを決定する。そして，連結したニューロン，その受容野空間のガウシアン関数b_j・パラメータ空間のパラメトリック式p_j・シナプス荷重w_jの積和によって CP の出力信号u_cを出力する。

以下，具体的に CP の各部の構成について説明する。

連結空間: 入力変数I_iとj番目のガウシアン関数におけるi番目入力に対する中心c_{ij}との距離d_jを(4.4)式で定義する。

$$d_j = \sum_{i=1}^{n} \left| I_i - c_{ij} \right|, \quad j = 1,2,\cdots,m \tag{4.4}$$

ここで，mは全ニューロン数(メモリ数)である。距離d_jが小さいほど，受容野空間においてニューロンの重要度が高くなる((4.5)式参照)。そのため，距離d_jを小さい順に並べ，その距離が小さい方からa $(a \le m)$個のニューロンを連結する。その連結した受容野のニューロン群(a 個)を用いて，出力信号u_cが生成される。

また，連結空間の尺度(4.4)式は，制御目的に応じて変えることができる。本論文では，フィードバック制御システムで提案する CP を用いるため，追従誤差$e^{(i-1)}$を入力変数I_iとする。その場合，追従誤差は時間経過するにつれ 0 に近づくため，制御最終時刻には中心c_{ij}が 0 近傍のニューロン群が必要になる。そのため，ガウシアン関数の中心のみを考慮した(4.4)

式で定義している。

受容野空間：受容野とは，感覚系のニューロンの神経応答に変化が生じるような刺激が提示される空間領域である。連結されたニューロン群の j 番目受容野 b_j^{assoc} を次式のガウシアン関数で表現する。

$$b_j^{assoc} = \prod_{i=1}^{n} \exp\left\{ -\frac{\left(I_i - c_{ij}^{assoc}\right)^2}{\left(\sigma_{ij}^{assoc}\right)^2} \right\}, \quad j = 1,2,\cdots,a \tag{4.5}$$

ここで, $c_{ij}^{assoc}, \sigma_{ij}^{assoc}$ はそれぞれ連結された j 番目のニューロンのガウシアン関数における i 番目の入力に対する中心・分散であり，$assoc$ は連結されたニューロンを意味する。b_j^{assoc} の値は，ニューロンが発火した時のパルス信号の大きさに対応する。なお，連結された中心・分散の集合は

$$\mathbf{c} = \left[c_{11}^{assoc},\cdots,c_{n1}^{assoc},c_{12}^{assoc},\cdots,c_{n2}^{assoc},\cdots,c_{1a}^{assoc},\cdots,c_{na}^{assoc} \right]^T \tag{4.6}$$

$$\boldsymbol{\sigma} = \left[\sigma_{11}^{assoc},\cdots,\sigma_{n1}^{assoc},\sigma_{12}^{assoc},\cdots,\sigma_{n2}^{assoc},\cdots,\sigma_{1a}^{assoc},\cdots,\sigma_{na}^{assoc} \right]^T \tag{4.7}$$

である。

パラメータ空間：連結されたニューロン群のうち j 番目のパラメータ空間は次式の線形パラメトリック式 p_j^{assoc} で表現する。

$$p_j^{assoc} = p_{0,j}^{assoc} + p_{1,j}^{assoc} I_1 + \cdots + p_{n,j}^{assoc} I_n, \; j = 1,2,\cdots,a \tag{4.8}$$

ここで，$p_{0,j}^{assoc}, p_{1,j}^{assoc}, \cdots, p_{n,j}^{assoc}$ は連結されたニューロンの線形パラメトリック式の係数である。この係数も適応的に変化させることにより，制御効率が高まることを期待している。なお，連結された線形パラメトリック式の係数の集合は,

$$\mathbf{p}_i = \left[p_{i,1}^{assoc}, p_{i,2}^{assoc}, \cdots, p_{i,a}^{assoc} \right]^T, \; i = 0,1,\cdots,a \tag{4.9}$$

である。

出力信号：CP の出力信号 u_c は，次式で表わされる。

$$u_c = \mathbf{b}^T \mathbf{P} \mathbf{w} = \boldsymbol{\psi}^T \mathbf{w} \tag{4.10}$$

ただし,

$$\boldsymbol{\psi}^T = \mathbf{b}^T \mathbf{P} \tag{4.11}$$

$$\mathbf{b} = \left[b_1^{assoc} \quad b_2^{assoc} \quad \cdots \quad b_a^{assoc} \right]^T \quad \mathbf{b} \in R^{a \times 1} \tag{4.12}$$

$$\mathbf{P}=\begin{bmatrix} p_1^{assoc} & 0 & \cdots & 0 \\ 0 & p_2^{assoc} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & p_a^{assoc} \end{bmatrix} \quad \mathbf{P}\in R^{a\times a} \tag{4.13}$$

$$\mathbf{w}=\begin{bmatrix} w_1^{assoc} & w_2^{assoc} & \cdots & w_a^{assoc} \end{bmatrix}^T \quad \mathbf{w}\in R^{a\times 1} \tag{4.14}$$

である。ここで，$b_j^{assoc}, p_j^{assoc}, w_j^{assoc}$(j=1,2,..., a)はそれぞれ，連結された受容野・パラメータ・シナプス荷重のニューロンの集合$\mathbf{b}, \mathbf{P}, \mathbf{w}$ の 1 つの要素である。

例えば, Fig.4.1 のグレーで色づけされたニューロンが連結されたニューロン群とし, Fig.1 の点線内に位置するニューロンの各空間をb_1, p_1, w_1とする。そして，上から順に番号を与えた場合，グレーのニューロンが連結されたニューロン数はa=3 であり，連結された受容野・パラメータ・シナプス荷重はそれぞれb_2, b_5, b_7， p_2, p_5, p_7， w_2, w_5, w_7 となる。この結果，(4.12)式の集合$\mathbf{b}$ の各要素は$b_1^{assoc}=b_2$， $b_2^{assoc}=b_5, b_3^{assoc}=b_7$，(4.13)式の集合$\mathbf{P}$ の各要素は $p_1^{assoc}=p_2$， $p_2^{assoc}=p_5, p_3^{assoc}=p_7$，(4.13)式の集合の $\mathbf{w}$ 各要素は $w_1^{assoc}=w_2$， $w_2^{assoc}=w_5$, $w_3^{assoc}=w_7$ となるため，出力信号u_c は(4.10)式より

$$u_c = b_2 p_2 w_2 + b_5 p_5 w_5 + b_7 p_7 w_7 \tag{4.15}$$

となる。

4.3.1 自己構造アルゴリズム

提案する CP は，新しいニューロンの生成と不要となったニューロンの消去によって，ニューロン数の増減処理を行っている。これにより，現状態に対応した CP が構成され，効率の良い制御を可能にする。自己構造メカニズムは次のようになる。

ニューロンの生成：連結空間処理時得られた距離の最小値を$d_{\min}$ とする。このとき，距離の最小値は次式で計算される。

$$d_{\min}=\min\left(d_j\right), \quad j=1,2,\cdots,m \tag{4.16}$$

ニューロンの生成条件$d_{\min}\geq d_{th}$を満たした時，新たなニューロンを生成する。ただし，d_{th} は生成閾値である。生成された新たなニューロンの各空間のパラメータは次のように与えられる。

$$\begin{aligned} &w_{m+1}=0, c_{i(m+1)}=I_i, \sigma_{i(m+1)}=\sigma_i^{pre} \\ &p_{0,m+1}=p_0^{pre}, p_{i,m+1}=p_i^{pre} \end{aligned} \tag{4.17}$$

ここで，$\sigma_i^{pre}, p_0^{pre}, p_i^{pre}$ (i=1,2,..., n)は正の定数である。ニューロン数m は，$m\leftarrow m+1$ で

更新される。

ニューロンの消去：各ニューロン(j =1,2,..., m)に重要度指数 S_j を与える。 S_j の初期値は1.0であり，次のように更新を行う。

$$S_j(t+1)=\begin{cases} S_j(t)\cdot\exp(-\beta_1) & if \text{ not associate} \\ S_j(t)\cdot\left[2-\exp\left(-\beta_2\left(1-S_j(t)\right)\right)\right] & if \text{ associate} \end{cases} \quad j=1,2,\cdots,m \tag{4.18}$$

ここで，$\beta_1,\beta_2\geq 0$ は設計定数である。連結空間で連結されたニューロンの重要度を上げ，連結されなかったニューロンの重要度を下げる。

ニューロンの消去条件 $S_j\leq S_{th}$ を満たす時，ニューロン j を不要なニューロンだと判断し，消去する。ただし， S_{th} は消去閾値であり， $m>a$ を満たす時のみ消去を行う。制御初期は，ニューロン数は増減を繰り返すが，定常状態になると (4.18)式より，ニューロン数 m は a 個になる。ニューロン数と連結数が一致するため，連結されるニューロンは固定される。つまり，この a 個のニューロンが「制御対象を目標信号 $\mathbf{r}$ に追従させる」記憶となるため，これを想起する形で制御対象を最適化する制御入力 u_c を構築できる。

ニューロン自己構造メカニズムにおいて，文献[40]，[56]との違いは，文献[40]は，重要度の更新条件(4.18)式をガウシアン関数の出力値により判断していることで，文献[56]は，このニューロンの消去を行っていないことである。

4.4 小脳パーセプトロン改良モデル利用型ロバスト制御システム

提案する小脳パーセプトロン改良モデル利用型ロバスト制御システム(CPRCS)を Fig.4.2 に示す。提案システムは，Plant の初期の制御はスライディング変数を用いたロバスト制御器(RC)で行い，その間に CP が最適な制御器を構築し，(4.2)式の最適入力 u^* を近似する。定常状態になると，構築された CP が Plant の制御を担い，RC は最適入力 u^* と CP により最適入力を近似した u_c との近似誤差を打ち消し，性能を向上させる。CPRCS の制御信号 u を次式で表わす。

$$u=u_c+u_r \tag{4.19}$$

ここで， u_r は RC の出力である。また，スライディング変数 s を次式で定義する。

$$s=e^{(n-1)}+k_1e^{(n-2)}+\cdots+k_n\int_0^t e(\tau)d\tau \tag{4.20}$$

ここで， k_i (i=1,2,...,n)はフィードバックゲインである。更に，近似誤差を打ち消すロバスト制御器を次式で定義する。

$$u_r=\frac{\left(\delta^4+1\right)}{2\delta^3}s \tag{4.21}$$

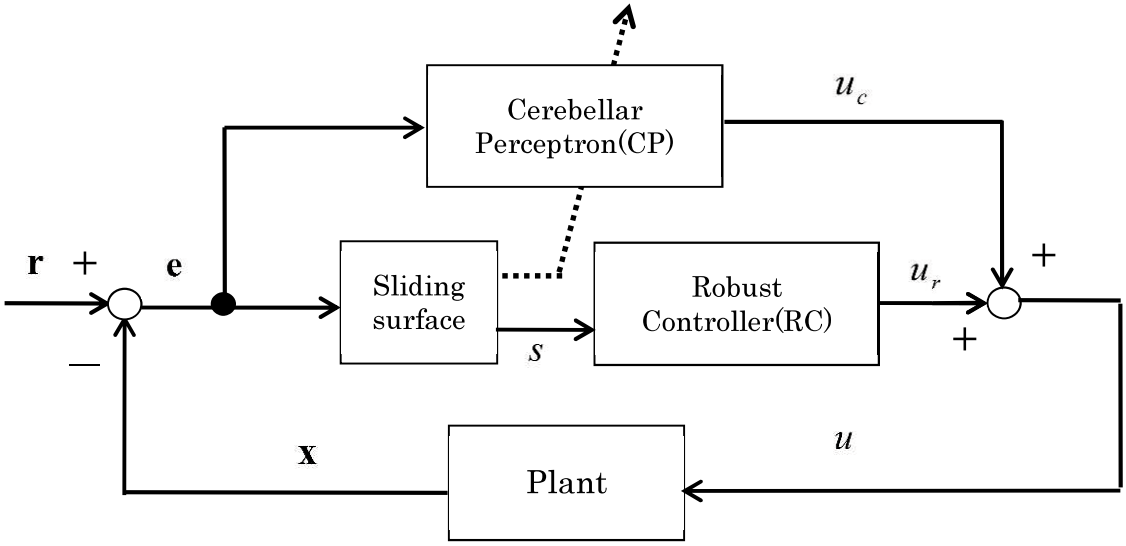

Fig.4.2.　A cerebellar perceptron-based robust control system

ここで，δ は減衰定数である。

4.4.1　小脳パーセプトロン改良モデルの近似

最適な CP の制御信号 u_c^* は最適入力 u^* を近似することができ，それは次式で表わされる。

$$u^* = u_c^* + \Delta = \boldsymbol{\psi}^{*T}\mathbf{w}^* + \Delta \tag{4.22}$$

ここで，$\boldsymbol{\psi}^*, \mathbf{w}^*$ はそれぞれ $\boldsymbol{\psi}, \mathbf{w}$ の最適ベクトル，Δ は近似誤差である。しかし，実際に最適ベクトルを決定することは困難であり，一意にならない場合がある。そこで，最適入力を模した CP 推定器を次のように定義する。

$$\hat{u}_c = \hat{\boldsymbol{\psi}}^T\hat{\mathbf{w}} \tag{4.23}$$

ここで，$\hat{\boldsymbol{\psi}}, \hat{\mathbf{w}}$ はそれぞれ $\boldsymbol{\psi}^*, \mathbf{w}^*$ の推定ベクトルである。そして，最適入力 u^* と最適 CP 推定器 $\hat{u}_c$ の推定誤差を次のように定義する。

$$\begin{aligned}\tilde{u} &= u^* - \hat{u}_c \\ &= \boldsymbol{\psi}^{*T}\mathbf{w}^* - \hat{\boldsymbol{\psi}}^T\hat{\mathbf{w}} + \Delta \\ &= \hat{\boldsymbol{\psi}}^T\tilde{\mathbf{w}} + \tilde{\boldsymbol{\psi}}^T\hat{\mathbf{w}} + \tilde{\boldsymbol{\psi}}^T\tilde{\mathbf{w}} + \Delta\end{aligned} \tag{4.24}$$

ここで，$\tilde{\boldsymbol{\psi}} = \boldsymbol{\psi}^* - \hat{\boldsymbol{\psi}}, \tilde{\mathbf{w}} = \mathbf{w}^* - \hat{\mathbf{w}}$ である。次に，$\tilde{\boldsymbol{\psi}}$ のテイラー級数展開を求める。

$$\begin{aligned}\tilde{\boldsymbol{\psi}} &= \mathbf{D}_{\mathbf{c}}^{T}\tilde{\mathbf{c}} + \mathbf{D}_{\boldsymbol{\sigma}}^{T}\tilde{\boldsymbol{\sigma}} + \mathbf{D}_{0}^{T}\tilde{\mathbf{p}}_{0} + \cdots + \mathbf{D}_{n}^{T}\tilde{\mathbf{p}}_{n} + \mathbf{h} \\ &= \mathbf{D}_{\mathbf{c}}^{T}\tilde{\mathbf{c}} + \mathbf{D}_{\boldsymbol{\sigma}}^{T}\tilde{\boldsymbol{\sigma}} + \sum_{i=0}^{n}\mathbf{D}_{i}^{T}\tilde{\mathbf{p}}_{i} + \mathbf{h}\end{aligned} \tag{4.25}$$

ここで，$\mathbf{h}$は高次項ベクトル，$\tilde{\mathbf{c}} = \mathbf{c}^{*} - \hat{\mathbf{c}}$，$\tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}^{*} - \hat{\boldsymbol{\sigma}}$，$\tilde{\mathbf{p}}_{i} = \mathbf{p}_{i}^{*} - \hat{\mathbf{p}}_{i}$ (i=0,1,...,n)であり，

$$\tilde{\boldsymbol{\psi}} = \begin{bmatrix}\tilde{\psi}_{1} & \tilde{\psi}_{2} & \cdots & \tilde{\psi}_{a}\end{bmatrix}^{T} \tag{4.26}$$

$$\mathbf{D}_{\mathbf{c}} = \left[\begin{array}{cccc}\dfrac{\partial\psi_{1}}{\partial\mathbf{c}} & \dfrac{\partial\psi_{2}}{\partial\mathbf{c}} & \cdots & \dfrac{\partial\psi_{a}}{\partial\mathbf{c}}\end{array}\right]_{\mathbf{c}=\hat{\mathbf{c}}} \tag{4.27}$$

$$\mathbf{D}_{\boldsymbol{\sigma}} = \left[\begin{array}{cccc}\dfrac{\partial\psi_{1}}{\partial\boldsymbol{\sigma}} & \dfrac{\partial\psi_{2}}{\partial\boldsymbol{\sigma}} & \cdots & \dfrac{\partial\psi_{a}}{\partial\boldsymbol{\sigma}}\end{array}\right]_{\boldsymbol{\sigma}=\hat{\boldsymbol{\sigma}}} \tag{4.28}$$

$$\mathbf{D}_{i} = \left[\begin{array}{cccc}\dfrac{\partial\psi_{1}}{\partial\mathbf{p}_{i}} & \dfrac{\partial\psi_{2}}{\partial\mathbf{p}_{i}} & \cdots & \dfrac{\partial\psi_{a}}{\partial\mathbf{p}_{i}}\end{array}\right]_{\mathbf{p}_{i}=\hat{\mathbf{p}}_{i}}, i = 0,1,\cdots,n \tag{4.29}$$

である。(4.25)式を(4.24)式に代入すると次式を得る。

$$\tilde{u} = \hat{\boldsymbol{\psi}}^{T}\tilde{\mathbf{w}} + \tilde{\mathbf{c}}^{T}\mathbf{D}_{\mathbf{c}}\hat{\mathbf{w}} + \tilde{\boldsymbol{\sigma}}^{T}\mathbf{D}_{\boldsymbol{\sigma}}\hat{\mathbf{w}} + \sum_{i=0}^{n}\tilde{\mathbf{p}}_{i}^{T}\mathbf{D}_{i}\hat{\mathbf{w}} + \varepsilon \tag{4.30}$$

ここで，近似誤差$\varepsilon = \Delta + \mathbf{h}^{T}\hat{\mathbf{w}} + \tilde{\boldsymbol{\psi}}^{T}\tilde{\mathbf{w}}$である。

また，(4.20)式のスライディング変数sの両辺を時間微分すると次式を得る。

$$\dot{s}(t) = e^{(n)} + k_{1}e^{(n-1)} + \cdots + k_{n}e \tag{4.31}$$

ここで，(4.19)式を(4.1)式に代入すると次式を得る。

$$x^{(n)} = f + g(u_{c} + u_{r}) \tag{4.32}$$

(4.31)式に，(4.2)式，(4.32)式を用いると次式を得る。

$$\begin{aligned}\dot{s}(t) &= \left(r^{(n)} - x^{(n)}\right) + k_{1}e^{(n-1)} + \cdots + k_{n}e \\ &= g\left(u^{*} - u_{c} - u_{r}\right)\end{aligned} \tag{4.33}$$

ここで，gは未知関数であるが，$0 < g$としても一般性は失われない。そこで，$0 < g \le g_{\max}$と仮定する。$g_{\max}$はgの上限定数である。(4.1)式の制御対象のダイナミクスgに対して，十分大きな値に$g_{\max}$を設定すれば，システム構成に影響を与えない。CP の出力信号u_{c}はCP 推定器(4.23)式と同義であるため，(4.33)式に(4.30)式を代入すると次式を得る。

$$\dot{s}(t) = g_{\max}\left(\hat{\boldsymbol{\psi}}^T \tilde{\mathbf{w}} + \tilde{\mathbf{c}}^T \mathbf{D}_{\mathbf{c}} \hat{\mathbf{w}} + \tilde{\boldsymbol{\sigma}}^T \mathbf{D}_{\boldsymbol{\sigma}} \hat{\mathbf{w}} + \sum_{i=0}^{n} \tilde{\mathbf{p}}_i^T \mathbf{D}_i \hat{\mathbf{w}} + \varepsilon - u_r \right) \tag{4.34}$$

以上の近似を用いて，安定性解析を行う[31],[56]。

4.4.2 安定性解析

リヤプノフ関数として次式を考える。

$$V = \frac{1}{2}\left(\frac{s^2}{\rho} + \frac{\tilde{\mathbf{w}}^T \tilde{\mathbf{w}}}{\eta_w} + \frac{\tilde{\mathbf{c}}^T \tilde{\mathbf{c}}}{\eta_c} + \frac{\tilde{\boldsymbol{\sigma}}^T \tilde{\boldsymbol{\sigma}}}{\eta_\sigma} + \sum_{i=0}^{n} \frac{\tilde{\mathbf{p}}_i^T \tilde{\mathbf{p}}_i}{\eta_i} \right) \tag{4.35}$$

ここで，$\eta_w, \eta_c, \eta_\sigma, \eta_i (i=0,1,\ldots,n)$はそれぞれ正の値を持つ学習係数，$\rho$は設計定数である。両辺を時間微分すると次式を得る。

$$\dot{V} = \frac{s\dot{s}}{\rho} + \frac{\tilde{\mathbf{w}}^T \dot{\tilde{\mathbf{w}}}}{\eta_w} + \frac{\tilde{\mathbf{c}}^T \dot{\tilde{\mathbf{c}}}}{\eta_c} + \frac{\tilde{\boldsymbol{\sigma}}^T \dot{\tilde{\boldsymbol{\sigma}}}}{\eta_\sigma} + \sum_{i=0}^{n} \frac{\tilde{\mathbf{p}}_i^T \dot{\tilde{\mathbf{p}}}_i}{\eta_i} \tag{4.36}$$

(4.34)式を上式に代入すると次式を得る。

$$\begin{aligned} \dot{V} = {} & \tilde{\mathbf{w}}^T \left(\frac{s g_{\max}}{\rho} \hat{\boldsymbol{\psi}} + \frac{\dot{\tilde{\mathbf{w}}}}{\eta_w} \right) + \tilde{\mathbf{c}}^T \left(\frac{s g_{\max}}{\rho} \mathbf{D}_{\mathbf{c}} \hat{\mathbf{w}} + \frac{\dot{\tilde{\mathbf{c}}}}{\eta_c} \right) \\ & + \tilde{\boldsymbol{\sigma}}^T \left(\frac{s g_{\max}}{\rho} \mathbf{D}_{\boldsymbol{\sigma}} \hat{\mathbf{w}} + \frac{\dot{\tilde{\boldsymbol{\sigma}}}}{\eta_\sigma} \right) \\ & + \sum_{i=0}^{n} \tilde{\mathbf{p}}_i^T \left(\frac{s g_{\max}}{\rho} \mathbf{D}_i \hat{\mathbf{w}} + \frac{\dot{\tilde{\mathbf{p}}}_i}{\eta_i} \right) + \frac{s g_{\max}}{\rho} \left(\varepsilon - u_r \right) \end{aligned} \tag{4.37}$$

ここで，(4.21)式のロバスト制御器u_rの減衰定数δを次のように定義する。

$$\delta = \frac{\rho}{g_{\max}} \tag{4.38}$$

ここで，減衰定数δは定数$\rho, g_{\max}$によって従属的に決定する定数であるため，減衰定数δは$0 < g \le g_{\max}$の仮定の下で有界である[17],[43]。そして，次のように適応則を与える。

$$\dot{\hat{\mathbf{w}}} = -\dot{\tilde{\mathbf{w}}} = \frac{\eta_w s}{\delta} \hat{\boldsymbol{\psi}} \tag{4.39}$$

$$\dot{\hat{\mathbf{c}}} = -\dot{\tilde{\mathbf{c}}} = \frac{\eta_c s}{\delta}\mathbf{D_c}\hat{\mathbf{w}} \tag{4.40}$$

$$\dot{\hat{\boldsymbol{\sigma}}} = -\dot{\tilde{\boldsymbol{\sigma}}} = \frac{\eta_\sigma s}{\delta}\mathbf{D_\sigma}\hat{\mathbf{w}} \tag{4.41}$$

$$\dot{\hat{\mathbf{p}}}_i = -\dot{\tilde{\mathbf{p}}}_i = \frac{\eta_i s}{\delta}\mathbf{D}_i\hat{\mathbf{w}}, \quad i = 0,1,\cdots,n \tag{4.42}$$

適応則(4.39)式~(4.42)式を(4.37)式に代入すると次式を得る。

$$\dot{V} = \frac{s}{\delta}\left(\varepsilon - u_r\right) \tag{4.43}$$

(4.21)式のロバスト制御器u_rを代入すると次式を得る。

$$\begin{aligned}\dot{V} &= \frac{s}{\delta}\varepsilon - \frac{\left(\delta^4+1\right)}{2\delta^4}s^2 \\ &= -\frac{s^2}{2} - \frac{1}{2}\left(\frac{s}{\delta^2} - \delta\varepsilon\right)^2 + \frac{1}{2}\delta^2\varepsilon^2 \\ &\le -\frac{s^2}{2} + \frac{1}{2}\delta^2\varepsilon^2\end{aligned} \tag{4.44}$$

ここで，両辺を時間領域$[0,t_f]$で積分する。なお，t_fは制御の最終時刻である。

$$V\left(t_f\right) - V\left(0\right) \le -\frac{1}{2}\int_0^{t_f} s^2 dt + \frac{1}{2}\delta^2\int_0^{t_f}\varepsilon^2 dt \tag{4.45}$$

ここで，$V\left(t_f\right) \ge 0$なので

$$\frac{1}{2}\int_0^{t_f} s^2 dt \le V\left(0\right) + \frac{1}{2}\delta^2\int_0^{t_f}\varepsilon^2 dt \tag{4.46}$$

となる。$V\left(0\right)$及び近似誤差εが有界ならば，スライディング変数sは$t_f \to \infty$で 0 に収束する[17-18],[31],[56]。よって，提案システムの安定性は証明された□

4.5　計算機シミュレーション

4.5.1　台車付き倒立振子

制御対象は，第 3 章と同様のダイナミクスで表わされる台車付き倒立振子である。制御の目的は，制御入力uを台車に与え，状態変数$\mathbf{x}$を目標信号$\mathbf{r}$に追従させることである。提案する小脳パーセプトロン改良モデル利用型ロバスト制御システム(CPRCS)の追従性能を検証するため，従来法のH_∞追従性能補償器を備えたリアルタイム強化学習制御システム(RRLCS)[43]と提案システムと同様のロバスト制御器を持つメモリの自己生成機能を導入したロバスト PCMAC (RPCSGD)[56]との性能比較シミュレーションを行った。台車付き倒立振子のパラメータの値は文献[56]同様，m_c=1.0[kg]，m_p=0.1[kg]，L=1.0[m]，g_r=9.8[m/s^2]とした。また，初期角度-45 度(≒-0.785[rad])，初期角速度 0[rad/s]の状態で，制御時間は 30.0 秒，サンプリングタイムは 0.01 秒，目標信号は① $r=\sin(t)$ (ref 1) 及び② $r=0.5\sin(t)+0.5\cos(2t)$(ref 2)の 2 ケースとした。

システムの性能評価として，(3.49),(3.50)式で定義される制御中の角度の総誤差面積に対する制御の単位時間当たりの誤差面積(平均誤差面積)と角度の 5～30 秒までの総誤差面積に対する制御の単位時間当たりの定常偏差(平均定常偏差)という 2 つの指標を用いて，検証した。

シミュレーションで使用した提案システムのパラメータは経験により，n=2，a=3，k_1=4.0，k_2=8.0，d_{th}=0.2，β_1=0.15，β_2=0.15，S_{th}=0.1，σ_i^{pre}=0.2，p_0^{pre}=0.5，p_1^{pre}=1.0，p_2^{pre}=1.5，η_w=1.0，η_c=0.01，η_σ=0.01，η_0=1.0，η_1=0.5，η_2=0.1 とした。そして，$g_{\max}$=10.0，ρ=3.0 とし，(4.38)式よりδ=0.3 となる。CP の初期ニューロン数は$m=a$=3 とし，そのニューロンのパラメータの初期値は(4.17)式により決定した。ただし，中心c_{ij}はc_{i0}=0.0，c_{i1}=0.5，c_{i2}=-0.5(i=1,2)とした。また，CP の入力変数I_i(i=1,2)は$I_i=e^{(i-1)}$である。

4.5.2　シミュレーション結果

結果 1：目標信号①$r=\sin(t)$(ref 1)

Table 4.1 に目標信号①(ref 1)の各システムの平均誤差面積と平均定常偏差，Fig.4.3 に振子の角度θの推移，Fig.4.4 に各システムの振子の角度の追従誤差の推移eの推移，Fig.4.5 に Fig.4.6 の拡大図を示す。

Fig.4.3 より各システムは角度がそれぞれの目標信号に追従していることから制御に成功したことが分かる。従来法との性能比較において，まず目標信号①の観点は，Fig.4.3 の矢印より，角度の追従誤差が 0.02 以下になったという点で，0 に近づいた速度が速いシステムは順に提案システム(CPRCS)，RRLCS，RPCSGD であった。Table.4.1 の定量的な観点か

ら定常状態は，提案システム，RRLCS，RPCSGD の順に優れていることが分かる。以上の結果から提案システムは，追従速度の向上と定常状態の安定化が行え，これは CP の連結空間において，状態に応じた適切なニューロンのみを最低限使用しているため，適応的な最適入力の構築が可能になったものと考えられる。

目標信号①のときのさらなる検証として，Fig.4.6 に提案システムの CP と RC の出力の推移，Fig.4.7 に各システムのニューロン(メモリ)数 m の推移，Fig.4.8 に提案システムの連結された 3 つニューロンの距離((4.4)式参照)の推移，Fig.4.9 に提案システムの連結されたシナプス荷重ベクトル $\mathbf{w}$ のノルムの推移，Fig.4.10 に連結されたガウシアン関数の中心ベクトル $\mathbf{c}$ ・分散ベクトル $\boldsymbol{\sigma}$ のノルムの推移，Fig.4.11 に連結されたパラメトリック式の係数ベクトル $\mathbf{p}_i$ (i=0,1,2)の推移を示す。

Fig.4.6 からも，1 秒で CP が最適入力を構築したと言える。また，同じロバスト制御器を持つ RPCSGD と Table.1(a)の定常偏差に差が生じた理由としては，入力データの量子化を行わなかった点や，不必要なニューロンを連結空間で呼び出される前に削除できた点が挙げられる。Fig.4.7 のニューロン数やメモリ数の推移においても，提案システムが最もニューロン数を抑えることができた。しかし，矢印の時刻 19 秒において，ニューロンが 1 つ増加しているが，これは Fig.4.8 の入力データとの距離が最も小さいニューロンが矢印の時刻 19 秒で，生成閾値 0.2 を上回ったためである。つまり時刻 19 秒までの CP は準最適入力であり，さらなる制御器構築のための増加し，不要になったニューロンがすぐに削除されている。最後に，Fig.4.9～4.11 より，制御開始 1 秒で各パラメータのノルムが定常変動に収束しているが，ノミナル荷重 $\mathbf{w}$ とパラメトリック式の係数 $\mathbf{p}_0$ のノルムは収束していると言えない。$\mathbf{p}_0$ は適応則(4.42)式を解くと，追従誤差の値が考慮されない更新式になるため，パラメータの更新が常に行われ，その影響がノミナル重みに影響したと考えられる。これらを定常値に収束させることができれば，さらなる性能向上が望めると考えられる。

Table 4.1　Comparison of error areas and steady state errors of the angle among three methods for ref 1

	CPRCS (Proposed)	RRLCS (Chapter 2)	RPCSGD[56]
Error area of the angle	3.9×10^{-3}	7.0×10^{-3}	9.6×10^{-3}
Steady state errors of the angle	4.5×10^{-4}	5.1×10^{-4}	1.6×10^{-3}

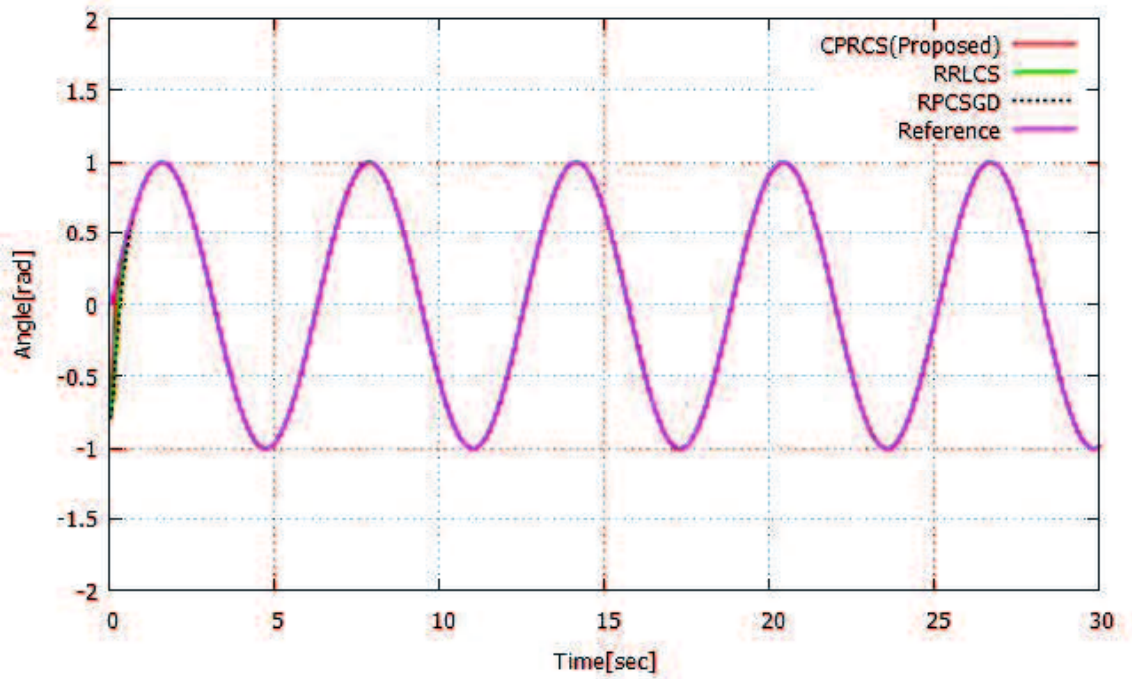

Fig.4.3. Comparison of control results of the angle among three methods for ref 1

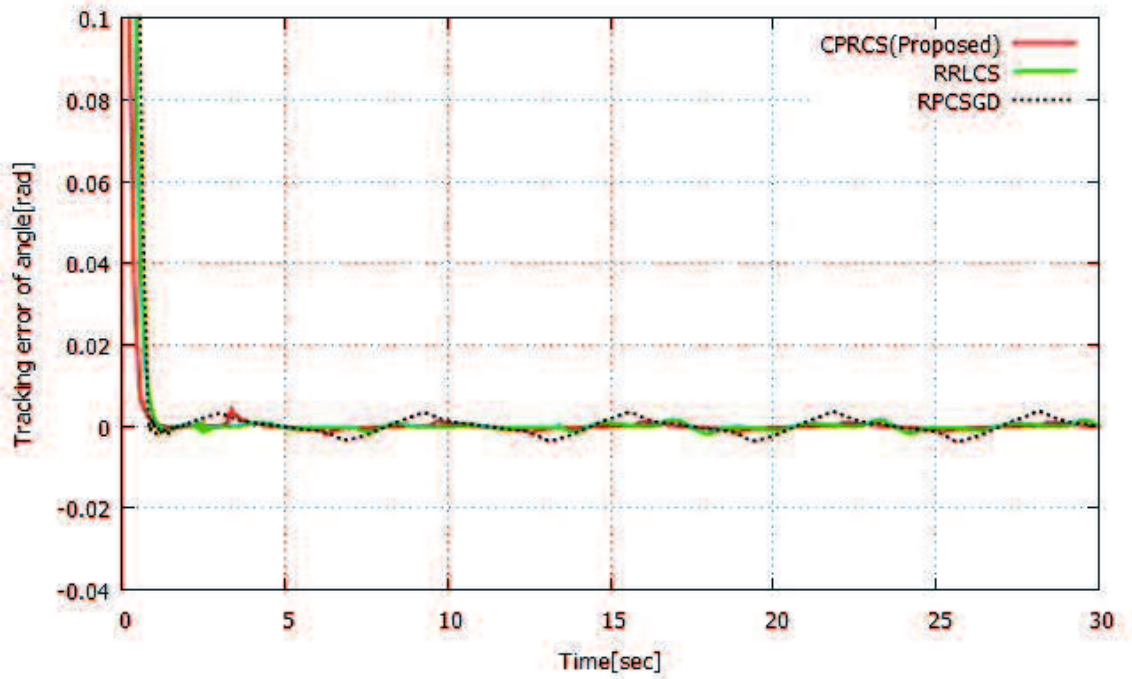

Fig.4.4. Comparison of tracking errors of the angle among three methods for ref 1

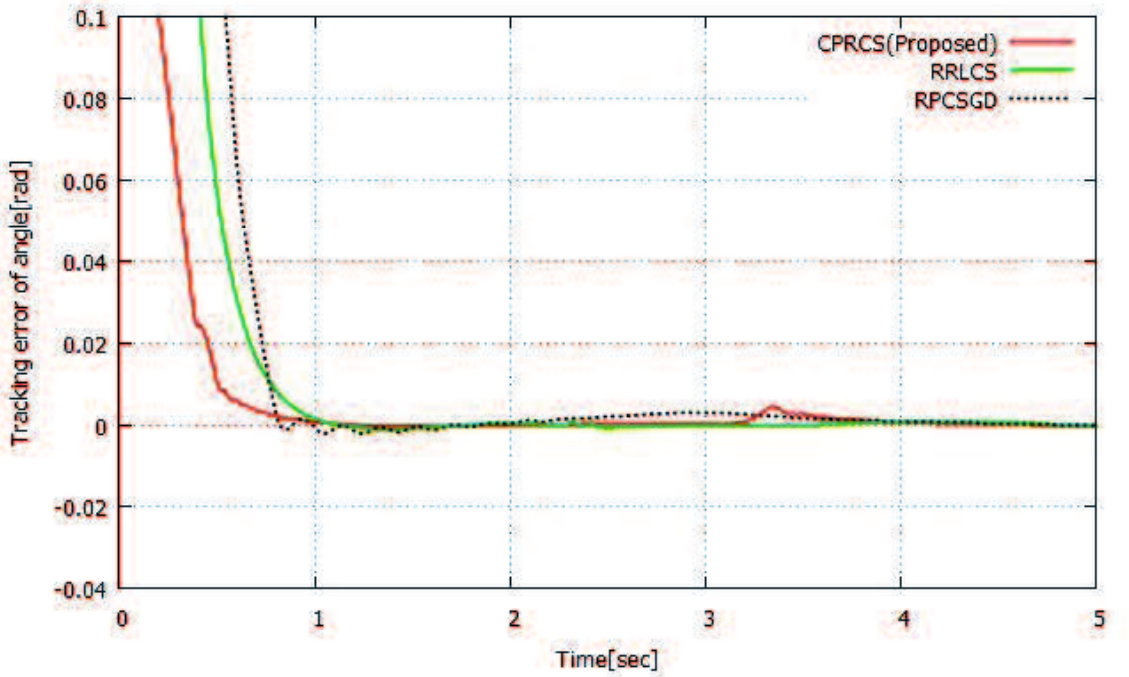

Fig.4.5. The enlarged view of Fig.4.5

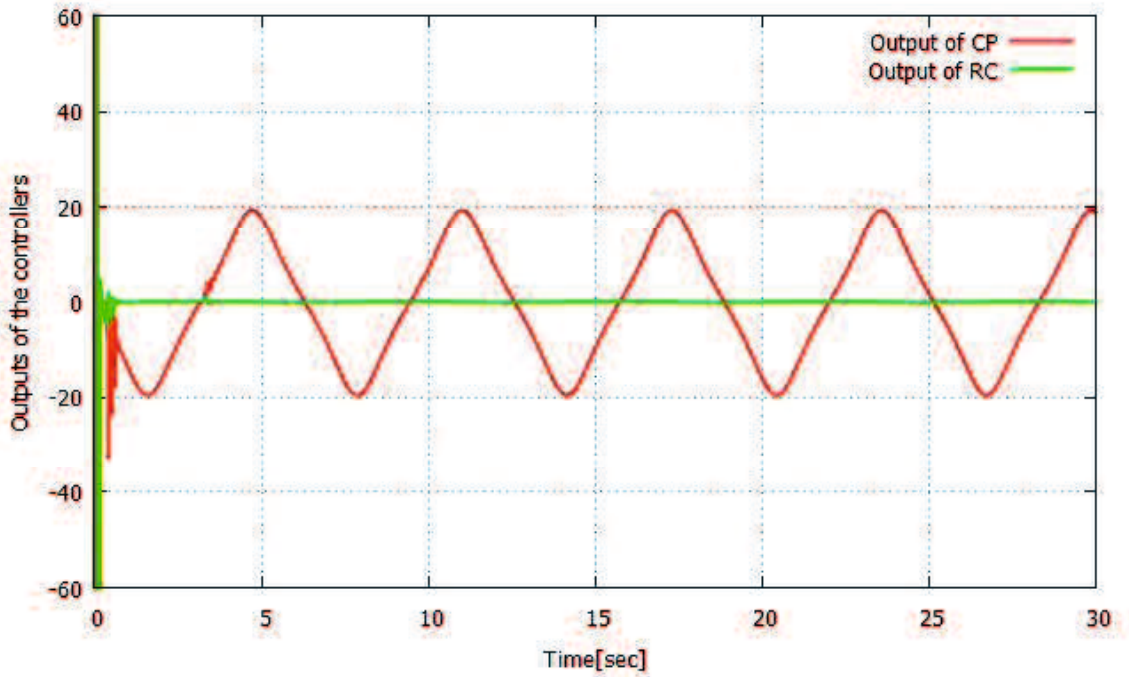

Fig.4.6. The outputs of the controller of the proposed system for ref 1

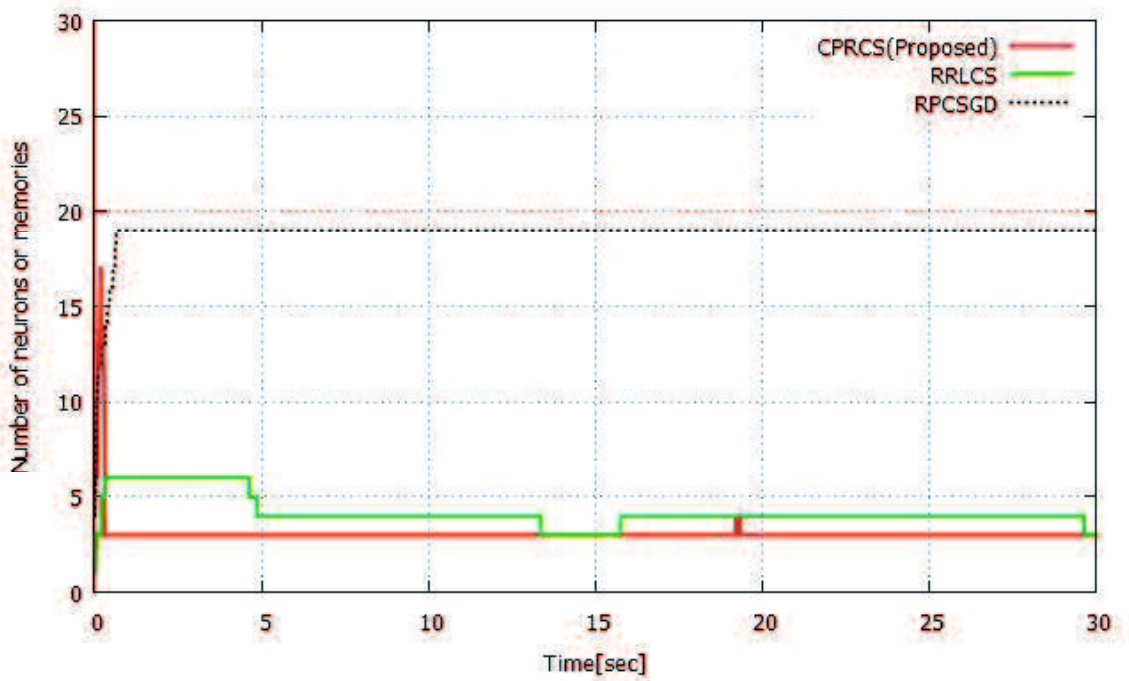

Fig.4.7. The number of neurons or memories of the each system for ref 1

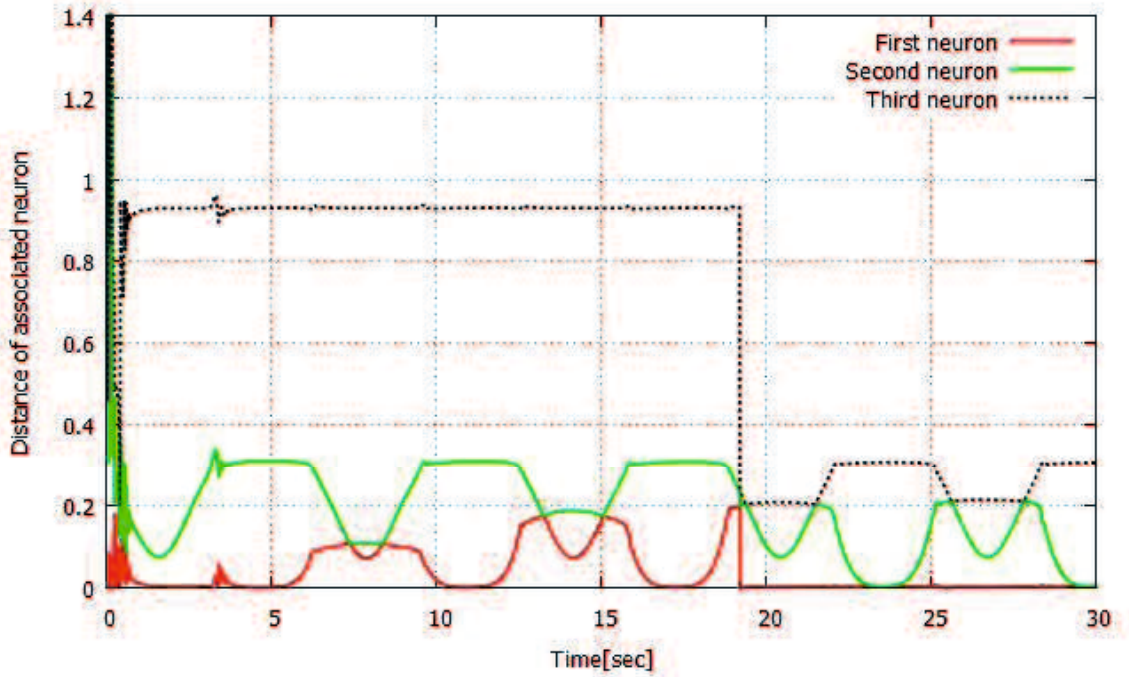

Fig.4.8. The distance between the input the center of the associated neuron of the proposed system for ref 1

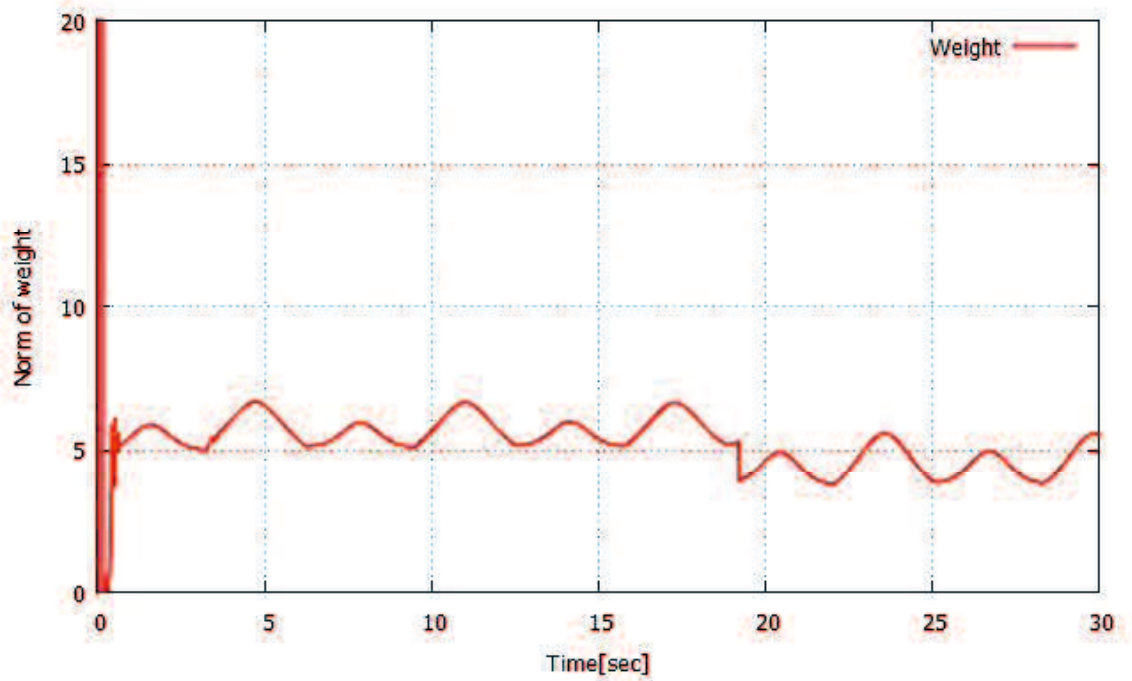

Fig.4.9. The norm of weight vector of the proposed system for ref 1

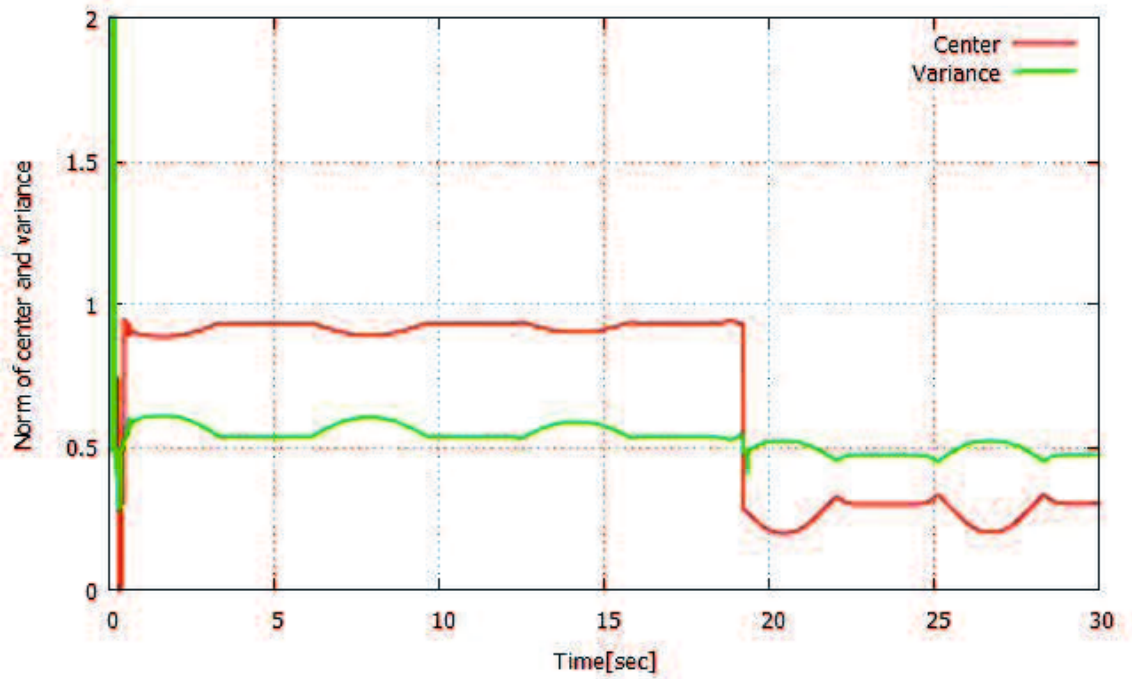

Fig.4.10. The norm of center and variance vector of the proposed system for ref 1

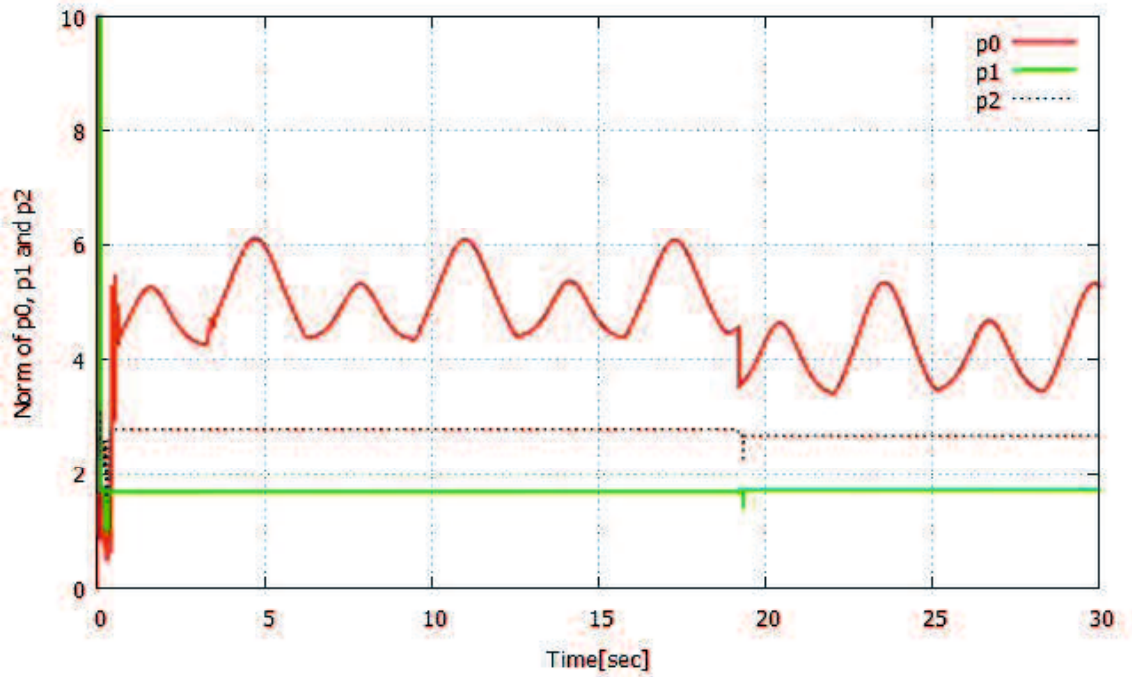

Fig.4.11. The norm of parametric parameter vectors of the proposed system for ref 1

結果 2：目標信号②$r=0.5\sin(t)+0.5\cos(2t)$(ref 2)

Table 4.2 に目標信号②(ref 2)の各システムの平均誤差面積と平均定常偏差，Fig.4.12 に振子の角度θの推移，Fig.4.13 に各システムの振子の角度の追従誤差の推移eの推移，Fig.4.14 に Fig.4.13 の拡大図を示す。

Fig.4.12 より各システムは角度がそれぞれの目標信号に追従していることから制御に成功したことが分かる。従来法との性能比較において， 0 に到達した速度が速いシステムは順に提案システム(CPRCS)，RRLCS，RPCSGD であったが，提案システムはオーバーシュートが発生したため，収束の速さから性能は評価できない。

Table.4.2 の定量的な観点から誤差面積・定常偏差は，提案システム，RRLCS，RPCSGD の順に優れていることが分かる。以上の結果から提案システムは，複雑な目標信号の場合，オーバーシュートが発生するが，CP がその状態に応じた適切なニューロンのみを最低限使用しているため，再び 0 に収束し，優れた追従性能を示した。

目標信号②のときのさらなる検証として，Fig.4.15 に提案システムの CP と RC の出力の推移，Fig.4.16 に各システムのニューロン(メモリ)数mの推移，Fig.4.17 に提案システムの連結された 3 つニューロンの距離((4.4)式参照)の推移，Fig.4.18 に提案システムの連結されたシナプス荷重ベクトル$\mathbf{w}$のノルムの推移，Fig.4.19 に連結されたガウシアン関数の中心ベクトル$\mathbf{c}$・分散ベクトル$\boldsymbol{\sigma}$のノルムの推移，Fig.4.20 に連結されたパラメトリック式の係

数ベクトル$\mathbf{p}_i$ (i=0,1,2)の推移を示す。

Fig.4.15 からも，1 秒で CP が最適入力を構築したと言える。Fig.4.16 のニューロン数やメモリ数の推移においても，提案システムが最もニューロン数を抑えることができた。最後に，Fig.4.17 より，制御時刻約 5 秒で各連結ニューロンの距離d_jが定常変動に収束し，Fig.4.18～Fig.4.20 より，各パラメータのノルムも同様の結果が見られる。これらの結果から制御時刻約 5 秒で最適な CP を構築できたと言える。

Table 4.2.　Comparison of error areas and steady state errors of the angle among three methods for ref 2

	CPRCS (Proposed)	RRLCS (Chapter 2)	RPCSGD[56]
Error area of the angle	8.5×10^{-3}	1.3×10^{-2}	1.8×10^{-2}
Steady state errors of the angle	3.9×10^{-4}	1.1×10^{-3}	6.8×10^{-3}

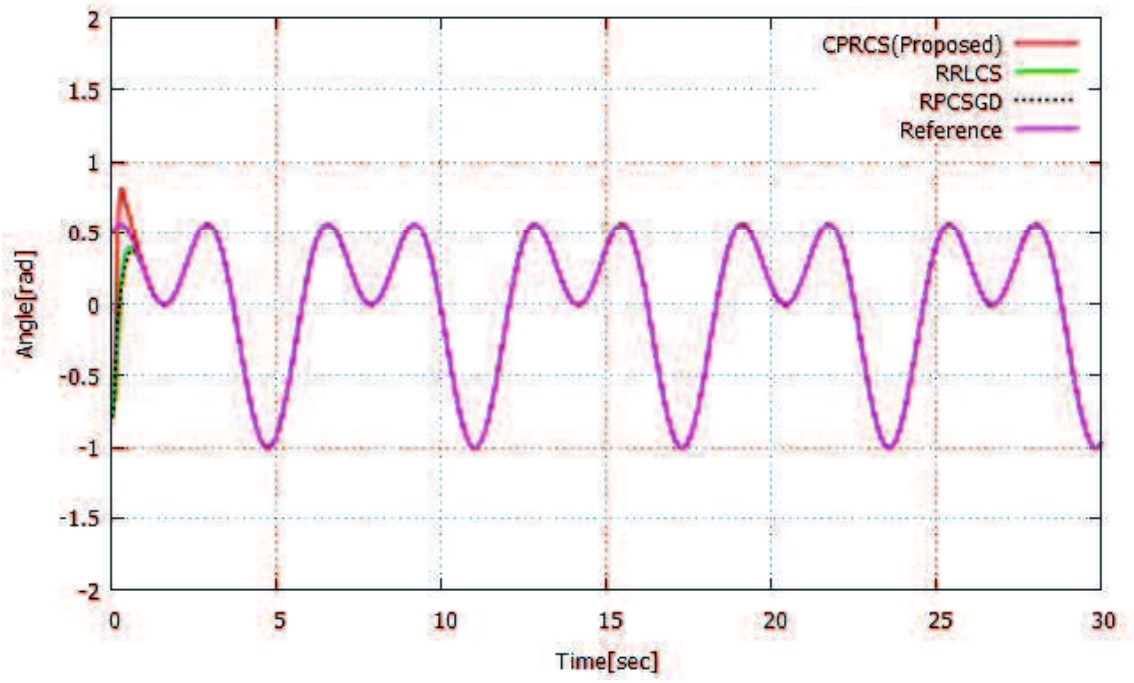

Fig.4.12.　Comparison of control results of the angle among three methods for ref 2

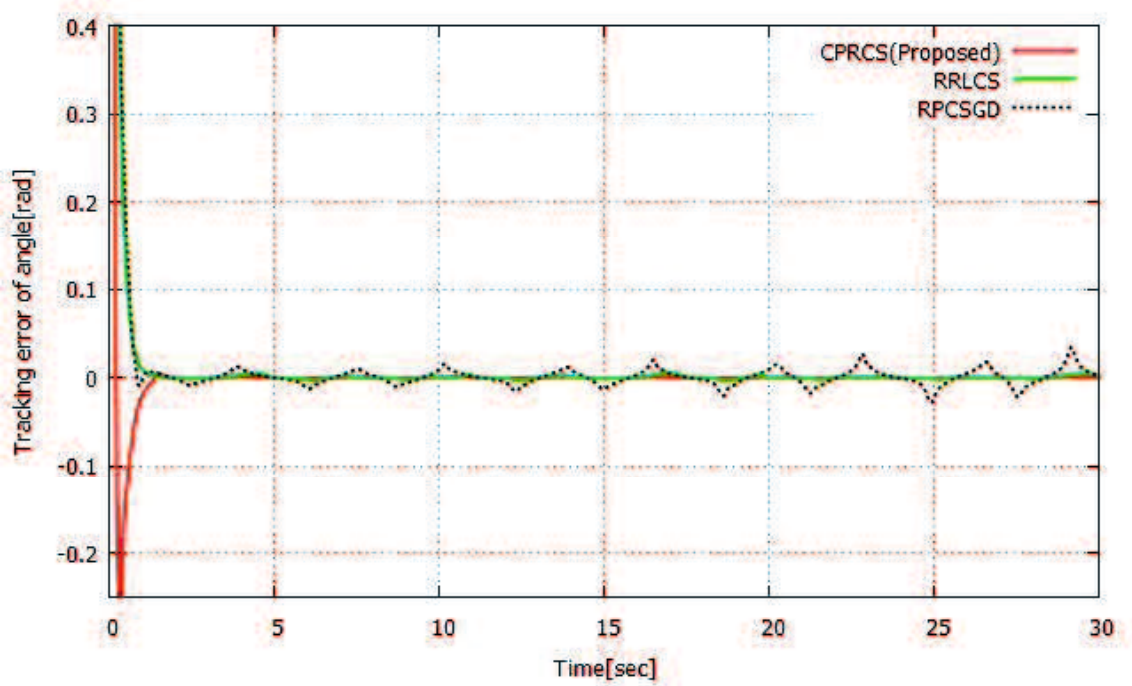

Fig.4.13. Comparison of tracking errors of the angle among three methods for ref 2

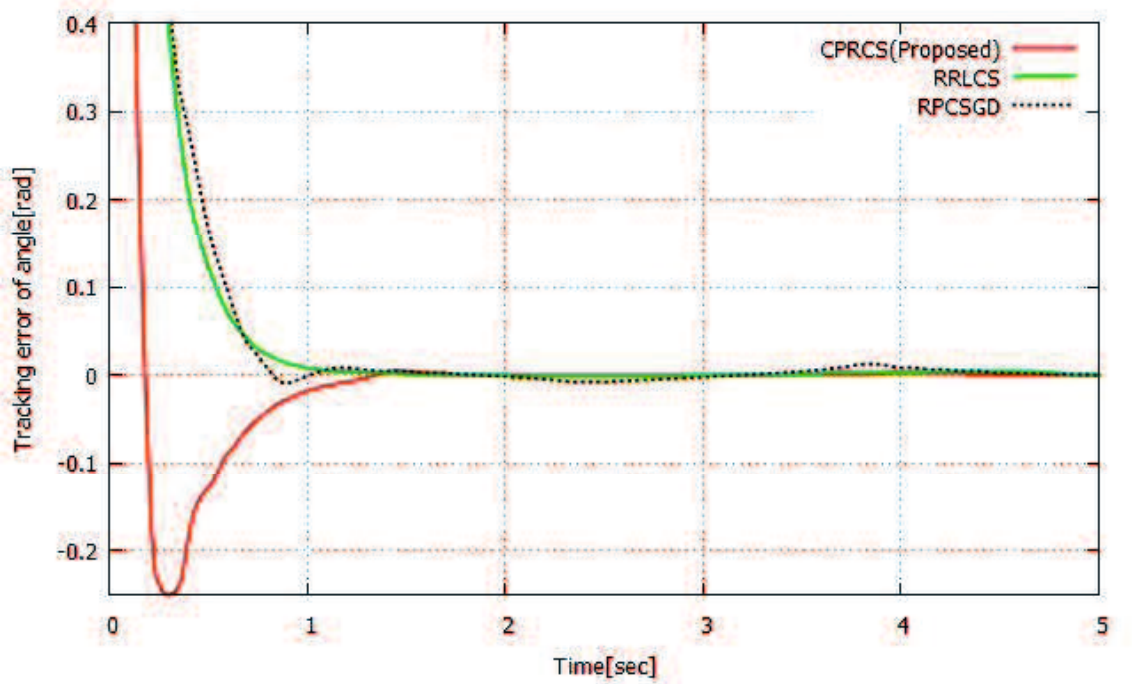

Fig.4.14. The enlarged view of Fig.4.14

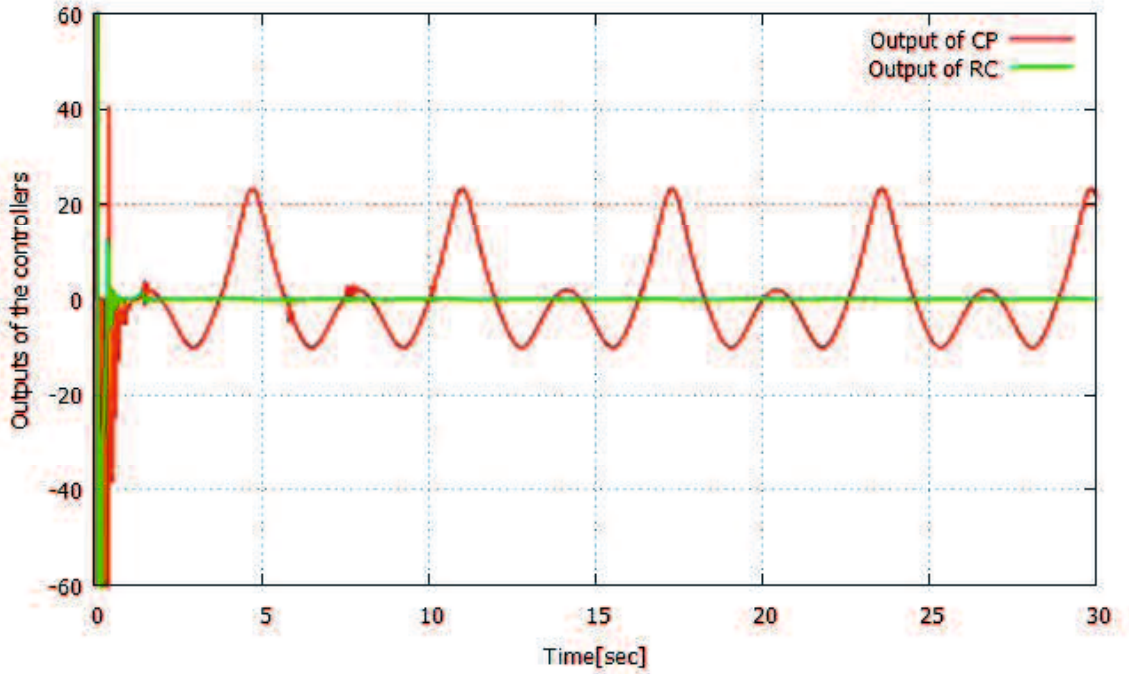

Fig.4.15. The outputs of the controller of the proposed system for ref 2

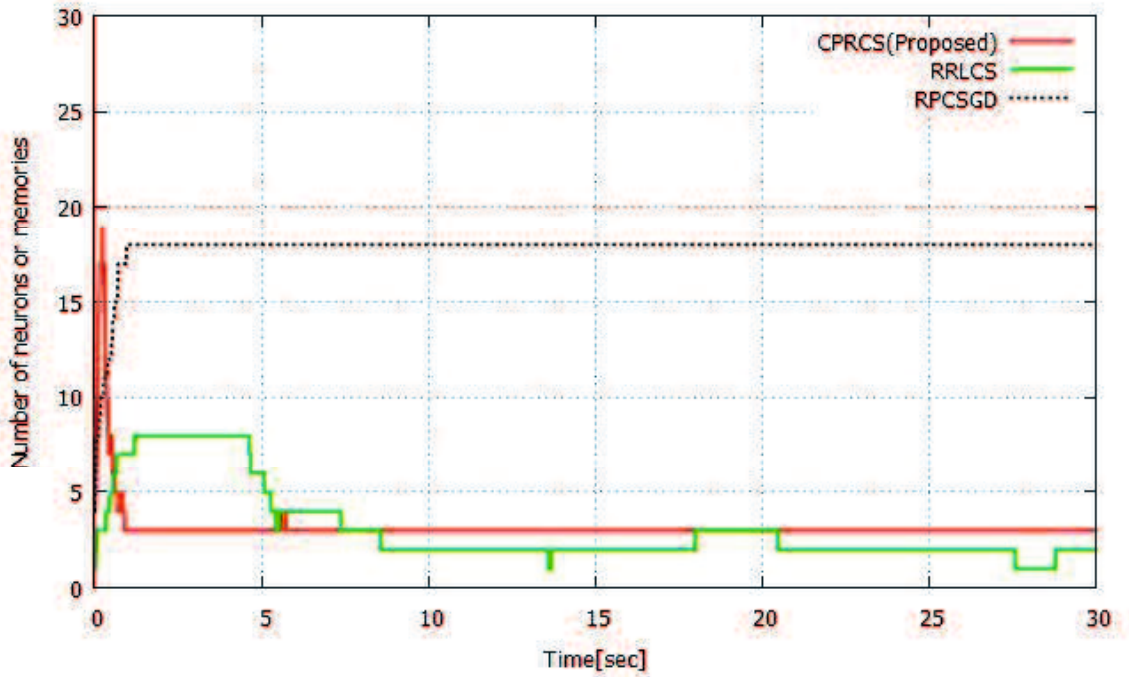

Fig.4.16. The number of neurons or memories of the each system for ref 2

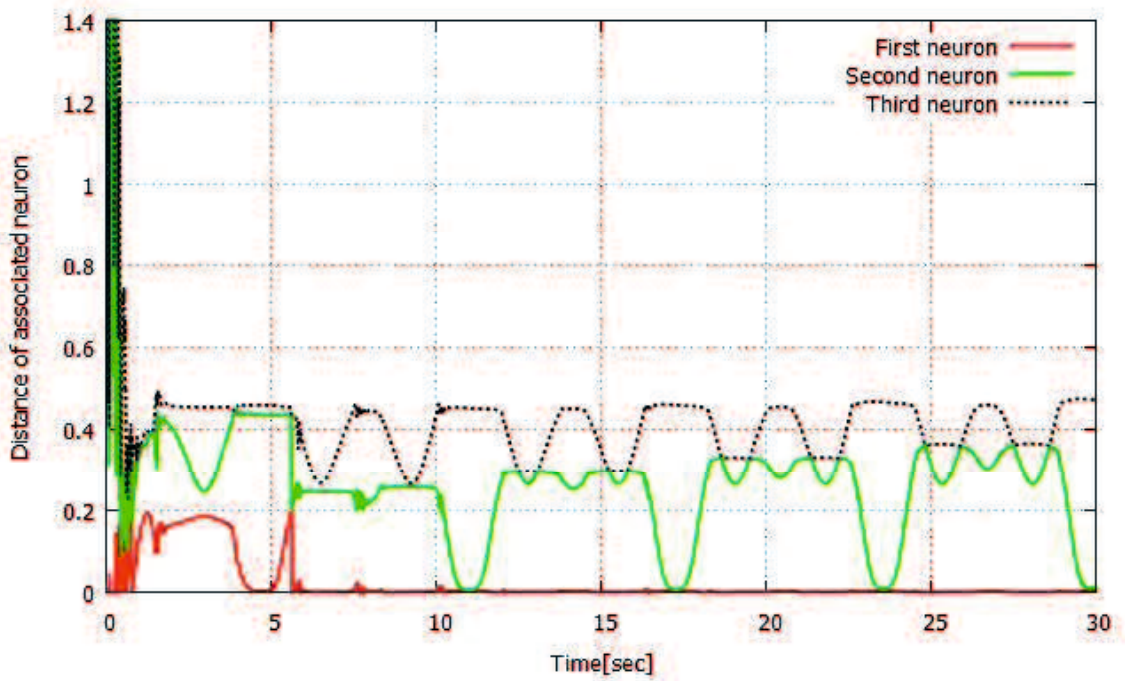

Fig.4.17. The distance between the input the center of the associated neuron of the proposed system for ref 2

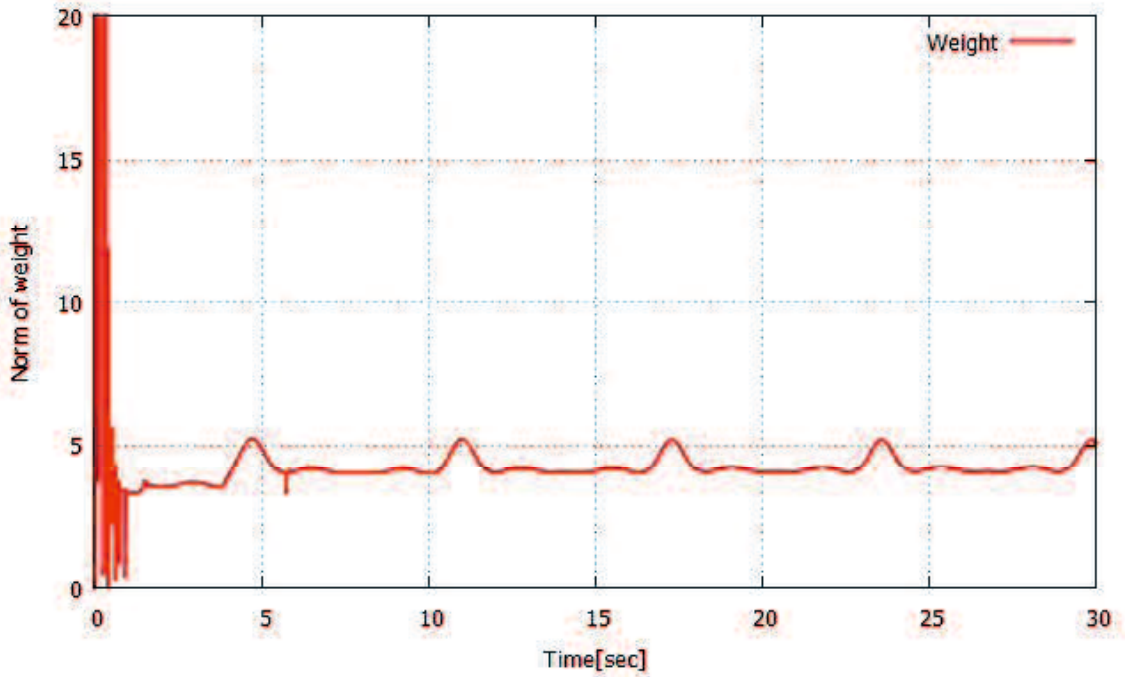

Fig.4.18. The norm of weight vector of the proposed system for ref 2

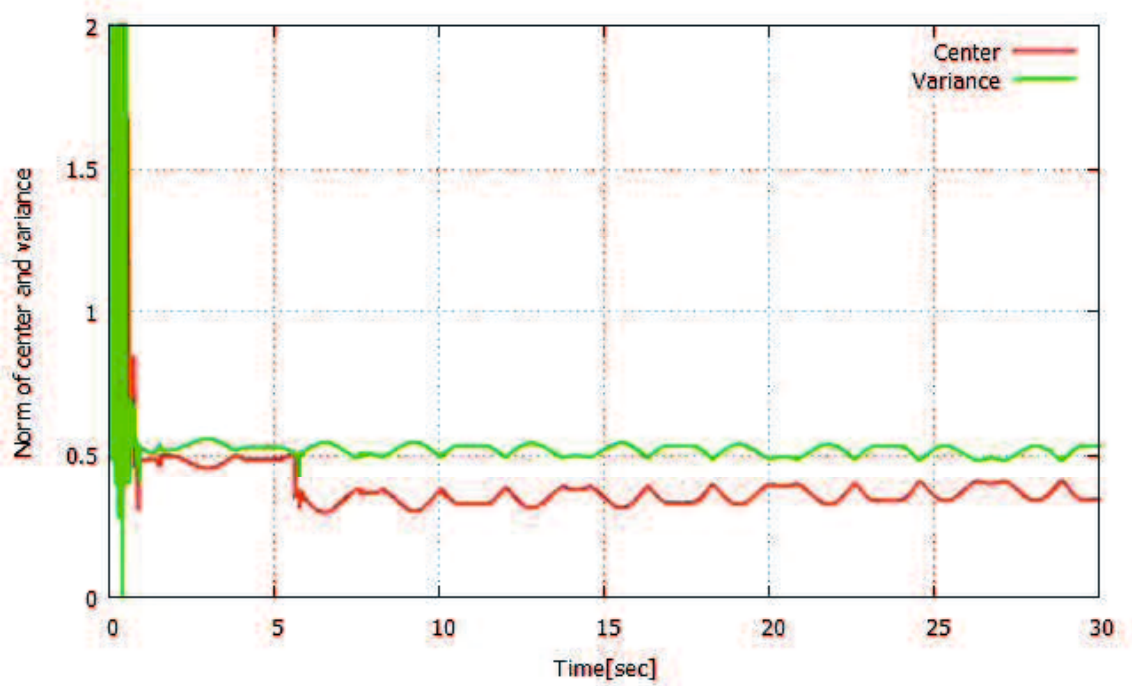

Fig.4.19. The norm of center and variance vector of the proposed system for ref 2

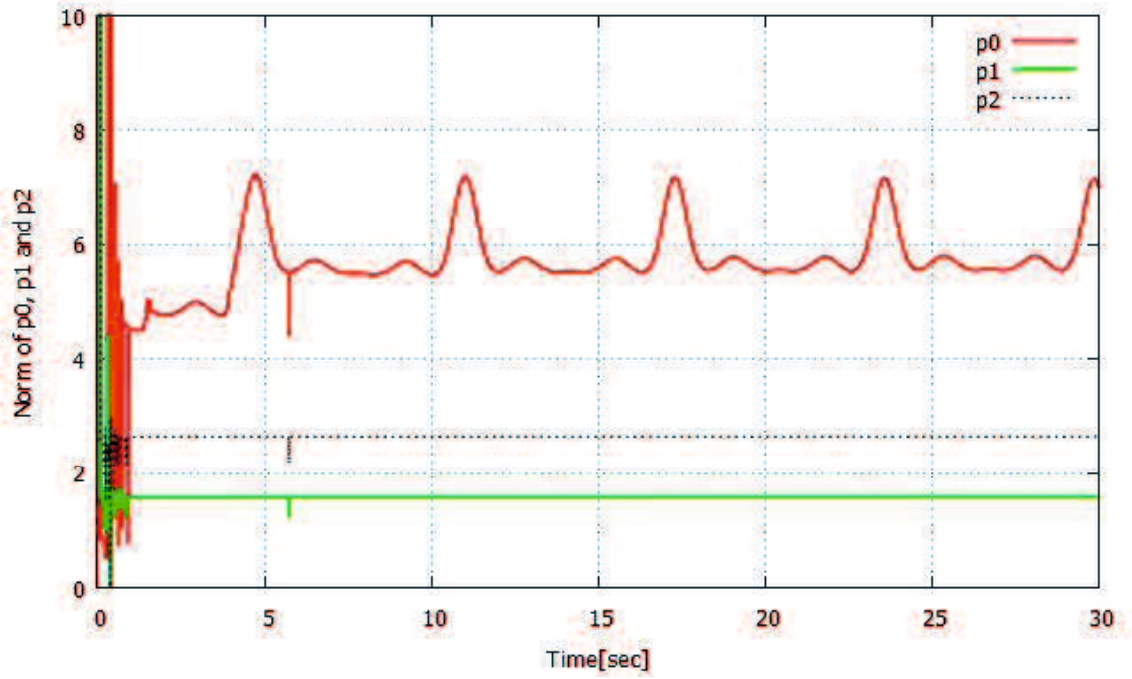

Fig.4.20. The norm of parametric parameter vectors of the proposed system for ref 2

4.6 まとめ

小脳の記憶の概念に着目し，小脳パーセプトロン改良モデルを提案した。さらに，これをロバスト制御システムに導入し，制御対象がさらに意図した動作をスムーズに行わせる制御システムである小脳パーセプトロン改良モデル利用型ロバスト制御システムを提案し，台車付き倒立振子シミュレーションにより，その有効性を示した。

今後の展開として，本論文では提案システムのニューロンの連結数を一定にしていたが，制御対象の状態に合わせて連結数を適応的に変えていく手法が考えられる。

第 5 章　フィードフォワード制御における自己融合

小脳パーセプトロン改良モデル利用型ロバスト制御システム

5.1　はじめに

第 4 章では，フィードバック制御において CMAC に環境変化への対応を可能とする，ニューロンの増減を行う自己構造メカニズムを付加した，自己構造小脳パーセプトロン改良モデルによるロバスト制御システムを提案した。本提案は従来システムより優れた性能を示したが，依然として定常誤差が残った。その原因として 2 つ考えられる。1 つはフィードバック制御により 1 時刻前の状態を基に制御を行うことによる遅延で，他の 1 つは自己構造メカニズムによるニューロンの増減で生じる最適な制御器が構築されるまでの制御入力の変動である。これらにより，制御システムの近似能力の低下や一時的な不安定化が考えられる。

1 つ目の遅延の解決法として，川人の生体の運動制御に関する学習機構であるフィードバック誤差学習(FEL)[50-53]を導入する。FEL では，運動の目標軌道とプラントの実現軌道間の誤差を用いて小脳の内部モデルの修正を行う。その設計法は，システム全体の大域漸近安定性保証に不可欠な従来のフィードバック制御器(Conventional Feedback Controller，CFC)と小脳の内部モデルを実現するニューラルネットワーク(NN)の 1 種である多層パーセプトロン(Multi Layer Perceptron，MLP)によるフィードフォワード制御器(Neural Network Feedforward Controller，NNFC)で構成される[51]。CFC を誤差関数として学習することで最適な NN が構成され，最終的に CFC の出力は 0 になる。

2 つ目のニューロン増減問題の解決法として，自己構造メカニズムの代わりに自己融合メカニズム[76]を導入する。本メカニズムは各ニューロンをある時刻においての重要度を 3 段階に分類し，第 2 章では削除した重要度が低いニューロンを削除しない。代わりに，その近傍のニューロンと融合させ，削除するニューロンの役割を残す。これにより，ニューロンの増減による制御システムの一時的な不安定を抑える方法である。

FEL を適用した先行研究[43,51-52]として，Sabahi ら[51]や Farivar ら[52]は，CFC として PID 制御器，NNFC としてファジィニューラルネットワークを用い，Sabahi らは電力システムに対して，Farivar らはマニピュレータに対して適応的な制御を可能にした。しかしながら，NN のニューロン数やそれらの初期値に性能が左右されやすいという問題を有している。そこで，筆者らは，フィードバック制御においてH_∞追従性能補償器と適応性に優れた自己構造型ファジィニューラルネットワーク(ASFNN)[31-35]によるリアルタイム強化学習制御システム(RRLCS)を提案し，ニューロン数の問題を解決し，優れた追従性能とロバスト性を示した[43]。しかしながら，制御システムはフィードフォワード制御を行っていないた

め，目標達成まで多くの時間を要する。一方，適応制御の概念を用いてフィードフォワード制御を実現した先行研究[53]として，Topalov らは CFC として PD 制御器，NNFC として MLP を用い，その学習法として，スライディングモード制御の概念を用いた適応則で行うニューロ適応型スライディングモード制御(NASCS)を提案した[53]。彼らは，フィードフォワードを実現した安定性解析を行い，ロボットマニピュレータ制御に成功した。しかしながら，文献[51-52]と同様の問題が考えられる。

本論文では, NNFC に第 3 章で用いた最適なネットワークを構築する小脳パーセプトロン改良モデル(CP)を利用する。このモデルに，FEL と自己融合メカニズムを導入することで，よりスムーズで, 様々なダイナミクスに対応可能な制御システムを目指す。そのため, NNFC を CP で構築した「自己融合小脳パーセプトロン改良モデル利用型制御システム(AFCPCS)」を提案する。CFC にはスライディングモード制御の概念を用いたロバスト制御器を用いて，これを誤差関数として FEL を用いた学習により, NNFC である CP は制御対象の逆モデルを生成し，フィードフォワード制御を実現する。そして，台車付き倒立振子による計算機シミュレーションにより，従来システムの RRLCS[43]及び NASCS[53]と提案システムとの性能比較を行い，提案システムの有効性を示す。

5.2 フィードバック誤差学習

次の n 次非線形システムを制御対象とする。

$$x^{(n)} = f(\mathbf{x}) + g(\mathbf{x})u \tag{5.1}$$

ここで，$x^{(n)}$ は x の n 階の時間微分，$\mathbf{x} = \left[x, \dot{x}, \cdots, x^{(n-1)}\right]^T = \left[x_1, x_2, \cdots, x_n\right]^T$ はシステムの状態変数ベクトル，f, g は未知の連続関数(ただし $g > 0$)，$u \in R$ は制御入力である。

このシステムは，状態変数ベクトル $\mathbf{x}$ を目標信号ベクトル $\mathbf{r} = \left[r, \dot{r}, \cdots, r^{(n-1)}\right]^T$ に追従させる，つまり，追従誤差ベクトル $\mathbf{e} = \mathbf{r} - \mathbf{x}$ を $\mathbf{0}$ にすることを目的としている。

運動制御に用いられる内部モデルは，脳科学の研究により小脳に存在すると考えられている[(10)]。ここで，Fig.5.1 のようなフィードフォワード制御を考える。制御対象のダイナミクス f の逆モデル f^{-1} が小脳の内部モデルであり，フィードフォワード制御によって運動指令 u_c を出力している。この逆モデルを用いると，目標信号 r を入力した時の制御対象の出力信号である状態変数 x は次のように表わされる。

$$x = f(u_c) = f\left(f^{-1}(r)\right) = r \tag{5.2}$$

状態変数 x は目標信号 r に一致し，追従誤差 e は 0 になるため，理想的な制御が速やかに実現できる。つまり，小脳は制御対象の逆モデルを内部モデルで実現し，フィードフォワード制御を行っている。

従来のフィードバック誤差学習による制御システムを Fig.1 に示す。このシステムは，目

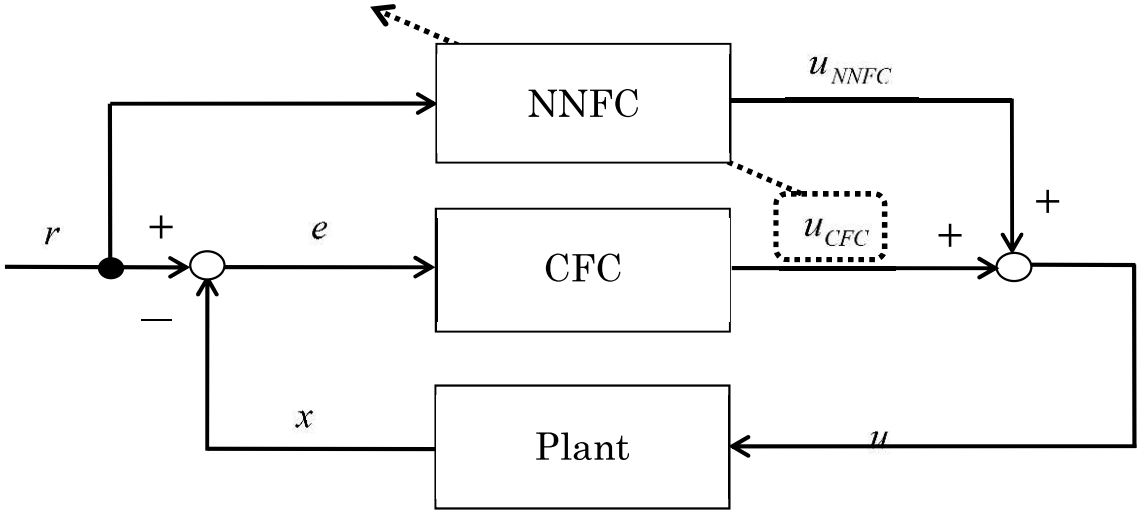

Fig.5.1. Structure of feedback error learning

標信号rをNNの入力とするフィードフォワード制御器(NNFC)u_{NNFC}と目標信号rと状態変数xの誤差$e=r-x$を入力とするフィードバック制御器(CFC)u_{CFC}で構成される[51]。そして，制御対象(Plant)への制御入力uは

$$u = u_{NNFC} + u_{CFC} \tag{5.1}$$

となる。u_{NNFC}の学習アルゴリズムの構成法は，最適な制御信号をu^*とし，学習のための誤差関数を次式で定義する。

$$E = \frac{1}{2}\left(u^* - u_{NNFC}\right)^2 \tag{5.2}$$

NNの結合荷重wに対する最急降下法による学習則は次式で表わされる。

$$\begin{aligned} \Delta w &= -\eta \cdot \frac{\partial E}{\partial w} \\ &= \eta \cdot \frac{\partial u_{NNFC}}{\partial w} \cdot \left(u^* - u_{NNFC}\right) \end{aligned} \tag{5.3}$$

ここで，ηは正の学習定数である。$u^* - u_{NNFC}$が0のとき，フィードフォワード制御器u_{NNFC}は，理想的な制御器が構築できており，フィードバック制御器の出力u_{CFC}が0になることを意味する。したがって$u^* - u_{NNFC}$はu_{CFC}に近似することができる。(5.3)式の$u^* - u_{NNFC}$をu_{CFC}に置き換えると

$$\Delta w = \eta \cdot \frac{\partial u_{NNFC}}{\partial w} \cdot u_{CFC} \tag{5.4}$$

となる。これが川人の提案した小脳における運動制御の学習モデルであるフィードバック誤差学習である[50]。このシステムの NNFC を提案する小脳パーセプトロン改良モデルで構築し，MAS の合意問題を実現するシステムを提案する。

5.3　提案する小脳パーセプトロン改良モデル

利用する小脳パーセプトロン改良モデル(CP)を Fig.5.2 に示す。まず入力変数 I_i ($i = 1,2,\cdots,z$)を用いて，入力層(Input Layer)で，中間層(Hidden Layer)の価値 q_j を計算し，中間層の連結ニューロンを決定する(Fig.5.2 ではグレーで色付けの部分)。そして，そのガウシアン関数 b_j と，その中間層の出力と出力層(Output Layer)間の結合荷重 w_j の積和とパラメトリック式 p_j によって CP の出力 u_c を求めることができる。以下に各層の構成を説明する。

入力層：入力変数 I_i を入力した時の中間層 j 番目のニューロンの価値 q_j を計算する。

$$q_j = \prod_{i=1}^{z} \exp\left\{-\frac{\left(I_i - c_{ij}\right)^2}{\sigma_{ij}^2}\right\}, \quad j = 1,2,\cdots,m \tag{5.5}$$

ここで，m はニューロン数(メモリ数)，c_{ij} は中間層 j 番目のノードの基底関数における i 番目の入力に対する中心，σ_{ij}^2 は同広がりである。計算したガウシアン関数値が大きい方から a ($a \le m$)個の中間層ニューロンを連結し，その連結した中間層を用いて出力を計算する。

中間層：入力変数 I_i を用いて，連結された中間層のニューロンのガウシアン関数 $b_{j'}^{assoc}$ とパラメトリック式 $p_{j'}^{assoc}$ を計算する。

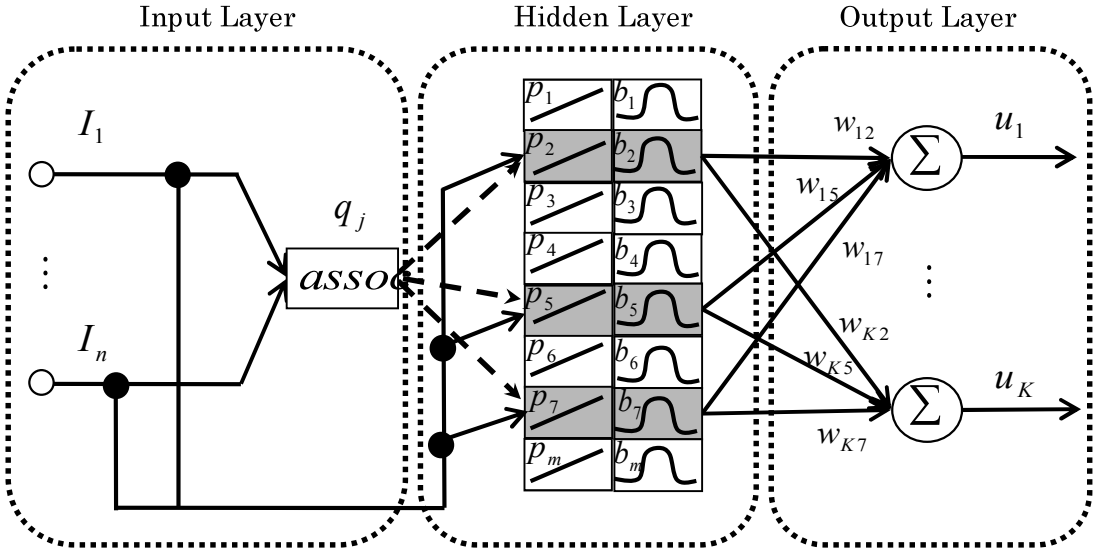

Fig.5.2.　Structure of the cerebellar perceptron improved model

$$b_{j'}^{assoc} = \prod_{i=1}^{z} \exp\left\{ -\frac{\left(I_i - c_{ij'}^{assoc}\right)^2}{\left(\sigma_{ij'}^{assoc}\right)^2} \right\}, \quad j' = 1,2,\cdots,a \tag{5.6}$$

$$p_{j'}^{assoc} = p_{0,j'}^{assoc} + p_{1,j'}^{assoc} I_1 + \cdots + p_{z,j'}^{assoc} I_z, \; j' = 1,2,\cdots,a \tag{5.7}$$

ここで，$b_{j'}^{assoc}$は連結されたニューロンの j' 番目ガウシアン関数であり，神経発火した時のパルス信号の大きさである。(5.6)式では全てのガウシアン関数を計算し，(5.7)式では連結されたニューロンのみを計算対象としている。また，$p_{j'}^{assoc}$は連結されたニューロンのパラメトリック式であり，さらに，入力変数I_iを線形結合することで，制御信号を滑らかにする。また，$c_{ij'}^{assoc}, \left(\sigma_{ij'}^{assoc}\right)^2$ はそれぞれ，連結されたニューロン j' 番目のガウシアン関数における i 番目の入力に対する中心・広がり，$p_{0,j'}^{assoc}, p_{1,j'}^{assoc}, \cdots, p_{z,j'}^{assoc}$は連結されたニューロンの線形パラメトリック式の係数である。$assoc$ は連結されたニューロンを意味する。

出力層：CP の出力u_cは，ガウシアン関数の出力と結合荷重の積$w_{kj'}^{assoc} \cdot b_{j'}^{assoc}$とパラメトリック式$p_{j'}^{assoc}$の和で求める。

$$u_k = \sum_{j'=1}^{a} \left(w_{kj'}^{assoc} b_{j'}^{assoc} + p_{j'}^{assoc} \right), \quad k = 1,2,\cdots,K \tag{5.8}$$

ここで，K は出力次元数，$w_{kj'}^{assoc}$は連結された中間層 j' 番目－出力層 k 番目間の結合荷重である。本提案モデルと第 2 章の CP との違いは，それぞれ$w_{kj'}^{assoc} \cdot b_{j'}^{assoc}$と $p_{j'}^{assoc}$の和で計算するか，$w_{kj'}^{assoc} \cdot b_{j'}^{assoc}$と $p_{j'}^{assoc}$の積で計算するかである。本論文において和で計算する理由は，$p_{j'}^{assoc}$は変動による影響を受けないためである。そのため $p_{j'}^{assoc}$を CP のもう一つの入力信号ととらえる。そして，学習過程によって生じる CP の変動を抑えるため，(5.8)式で定義している。

5.3.1　自己融合アルゴリズム

提案する CP は，新しいニューロンの生成と一時的に不要となったニューロン同士の融合によって，CP の構築を行う。自己融合メカニズムは次のようになる。

ニューロンの生成：中間層のガウシアン関数の集合を$\Gamma = \{b_1, b_2, \cdots, b_m\}$とする。このとき，ガウシアン関数の最大値は次式で計算される。

$$b_{\max} = \max_j \left(b_j \right), \quad j = 1,2,\cdots,m \tag{5.9}$$

ニューロンの生成条件$b_{\max} \le b_{th}, b_{th} \in (0,1)$を満たした時，新たなニューロンを生成する。ただし，b_{th}は生成閾値である。生成された新たなニューロンの各空間のパラメータは次のように与えられる。

$$w_{m+1}=0,\ c_{i(m+1)}=I_i,\ \sigma_{i(m+1)}=\sigma_i^{pre}$$
$$p_{0,m+1}=p_0^{pre},\ p_{i,m+1}=p_i^{pre} \tag{5.10}$$

ここで，$\sigma_i^{pre}, p_0^{pre}, p_i^{pre}$ (i=1,2,...,z)は正の定数である。ニューロン数mは，$m \leftarrow m+1$で更新される。

ニューロンの融合：一時的に不要と判定されたニューロンは削除ではなく他のニューロンとの融合を行う理由は，目標信号が時間変化する信号の場合，ある時刻では不必要であるが，削除したニューロンが別時刻で再び必要になる可能性があることにある。

各ニューロン(j =1,2,...,m)に重要度指数S_jを与え,「連結されたニューロンの集合Γ_{assoc}」「不必要なニューロンの集合Γ_{unnec}」「どちらでもないニューロンの集合Γ」の 3 つの集合に分ける。なお，Γ_{assoc}・Γ_{unnec}はそれぞれガウシアン関数の値が大きい・小さいa個の集合である。ここで，S_jの初期値は 1.0 であり，次のように更新を行う。

$$S_j(t+1)=\begin{cases} S_j(t)\cdot\left[2-\exp\left(-\beta_1\left(1-S_j(t)\right)\right)\right] & if \quad b_j\in\Gamma_{assoc} \\ S_j(t)\cdot\exp\left(-\beta_2\right) & if \quad b_j\in\Gamma_{unnec} \quad j=1,2,\cdots,m \\ S_j(t) & otherwise \end{cases} \tag{5.11}$$

ここで，β_1,β_2は設計定数である。上式は，連結された中間層の集合Γ_{assoc}に属するニューロンは重要度を上げ，Γ_{unnec}では重要度を下げ，その他のニューロンの重要度は変えないことを意味する。

集合Γ_{unnec}のj番目のニューロンの重要度が，削除条件$S_j \le S_{th}$を満たす時，このニューロンをM_1とし，このときのガウシアン関数の中心・広がり$\left(c_i^f,\left(\sigma_i^f\right)^2\right)$を他のニューロンに融合させるべきニューロンが存在するかを調査する。即ち，融合対象ニューロンは，集合Γと集合Γ_{unnec}の中からガウシアン関数の中心が最も近傍にあるニューロンとする。即ち，次式の最小距離dを導出した際の中心c_{ij}を持つニューロンj (M_2)が融合対象となる。

$$d=\min_j\sum_{i=1}^{I}\left|c_{ij}-c_i^f\right|,\quad b_j\notin\Gamma_{assoc},\quad j=1,2,\cdots,m \tag{5.12}$$

このとき，融合条件$d \le d_{th}$を満たすとき，ニューロンM_1は，ニューロンM_2に融合される。つまり，削除条件を満たしても，融合条件を満たさなければ，融合は行われない。融合条件を満たすとき，ニューロンN_2の中心・広がり$\left(c_{ij},\left(\sigma_{ij}\right)^2\right)$は次のように融合される。

$$c_{ij}\leftarrow\left(c_{ij}+c_i^f\right)/2 \tag{5.13}$$

$$\sigma_{ij}\leftarrow\sqrt{\sigma_{ij}^2+\left(\sigma_i^f\right)^2} \tag{5.14}$$

このとき，融合したニューロンの重要度指数S_jは 1.0 に戻し，ニューロン数mは融合処理により 1 減少する。このように CP を設計し，目標信号が時間変化する信号の場合でも対応

できる，適応的な制御システムを実現する。

なお，(5.11)式の集合Γ_{assoc}，Γ_{unnec}はそれぞれa個の集合で定義し，ニューロンの融合はニューロン数mが$2a$個以上の時($m \geq 2a$)のみ行う。この条件による，Γ_{assoc}の集合のニューロンのみ残ることを防ぐ。

5.4　自己融合小脳パーセプトロン改良モデル利用型

ロバスト制御システム

提案する自己融合小脳パーセプトロンモデル利用型ロバスト制御システム(AFCPRCS)をFig.5.3 に示す。制御初期は CFC であるロバスト制御器(RC)が制御を行い，その時の RC の出力u_rを誤差関数として，CP の学習を行う。RC の出力u_rが 0 になるとき，CP の学習が完了し，CP の出力u_cは，理想的な制御器u^*になったことを意味する。AFCPRCS の制御信号は次式で表わす。

$$u = u_c + u_r \tag{5.15}$$

ここで，RC の出力u_rを次のように定義する。

$$u_r = \frac{\left(\delta^4 + 1\right)}{2\delta^3} s \tag{5.16}$$

ここで，δは減衰定数，sはスライディング変数であり，

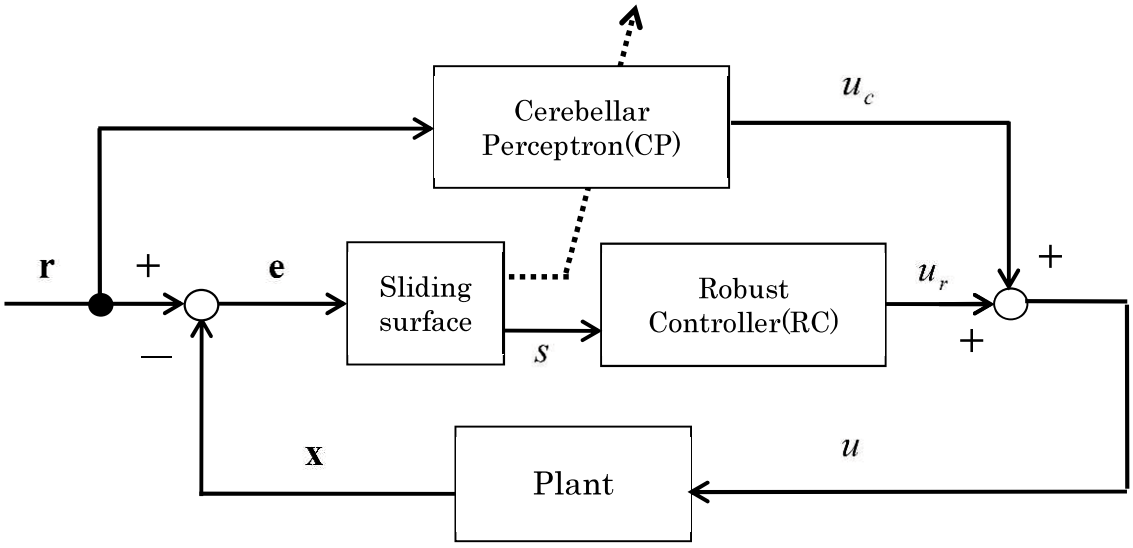

Fig.5.3.　Auto-fusion cerebellar perceptron-based robust control system

$$s = e^{(n-1)} + k_1 e^{(n-2)} + \cdots + k_n \int_0^t e(\tau) d\tau \tag{5.17}$$

で定義される。ここで，$k_i\,(i=1,2,\ldots,n)$はフィードバックゲインであり，フルビッツの安定性を満たすように決定する。

5.4.1 学習アルゴリズム

提案システム(AFCPRCS)におけるフィードバック誤差学習は，RC の出力であるu_rを誤差関数として学習を行い，この出力が 0 になるとき，CP は最適な制御器が構築される。RC の出力u_rを 0 にすることは(5.16)式より，(5.17)式のスライディング変数sを 0 にすることと同義である。したがって，u_rの代わりに，sを誤差関数として用いて，次の学習則でパラメータの更新を行う。

$$\Delta w_{kj}^{assoc} = \eta_w \cdot \frac{\partial u_c}{\partial w_{kj}^{assoc}} \cdot s \tag{5.18}$$

$$\Delta c_{ij}^{assoc} = \eta_c \cdot \frac{\partial u_c}{\partial c_{ij}^{assoc}} \cdot s \tag{5.19}$$

$$\Delta \sigma_{ij}^{assoc} = \eta_\sigma \cdot \frac{\partial u_c}{\partial \sigma_{ij}^{assoc}} \cdot s \tag{5.20}$$

$$\Delta p_i^{assoc} = \eta_i \cdot \frac{\partial u_c}{\partial p_i^{assoc}} \cdot s \tag{5.21}$$

ここで，$\eta_w, \eta_c, \eta_\sigma, \eta_i\,(i=0,1,\ldots,n)$は正の学習定数である。学習が進むにつれて$s$が小さくなってゆき，CP の学習が収束することを期待している。

5.5　計算機シミュレーション

5.5.1　台車付き倒立振子

制御対象は，第 3 章と同様のダイナミクスで表わされる台車付き倒立振子である。提案する自己融合小脳パーセプトロンモデル利用型ロバスト制御システムの追従性能を検証するため，第 3 章の小脳パーセプトロンモデル利用型ロバスト制御システム(CPRCS)と PD 制御器と多層パーセプトロン制御器によるニューロ適応型スライディングモード制御(NASCS)[51]との性能比較シミュレーションを行った。台車付き倒立振子のパラメータの値は第 2 章同様，m_c=1.0[kg]，m_p=0.1[kg]，L=1.0[m]，g_r=9.8[m/s^2]とした。また，初期角度 -45 度(≒-0.785[rad])，初期加速度 0 の状態で，制御時間は 30.0 秒，サンプリングタイムは 0.01 秒, 目標信号は①$r=\sin(t)$(ref 1)及び②$r=0.5\sin(t)+0.5\cos(2t)$(ref 2)の 2 ケースとした。

システムの性能評価として，(3.49),(3.50)式で定義される制御中の角度の総誤差面積に対する制御の単位時間当たりの誤差面積(平均誤差面積)と角度の 5～30 秒までの総誤差面積に対する制御の単位時間当たりの定常偏差(平均定常偏差)という 2 つの指標を用いて，検証した。

シミュレーションで使用した提案システムのパラメータは試行錯誤により，n=2，a=3，k_1=5.0，k_2=10.0，b_{th}=0.2，β_1=0.05，β_2=0.05，S_{th}=0.1，d_{th}=0.5，σ_i^{pre}=0.2，p_0^{pre}=0.5，p_1^{pre}=1.0，p_2^{pre}=1.5，η_w=35.0，η_c=1.0，η_σ=1.0，η_0=1.0，η_1=1.0，η_2=1.0，δ=0.3 とした。パラメータ p_0^{pre}，p_1^{pre}，p_2^{pre}，η_w，η_c，η_σ，η_0，η_1，η_2 は，大きい程追従は速いが定常偏差が残り，小さい程定常偏差を減らせるが追従が遅くなる。また，δ は定常偏差を抑えられることを表しているため，δ は小さい程システム性能が高くなる。しかし，δ を小さく設定し過ぎると制御入力$\mathbf{u}_i^f$が大きくなり，学習を妨げてしまい制御性能は低下する。逆に大きくし過ぎると制御入力$\mathbf{u}_i^f$が足りずオーバーシュートが大きくなり，定常偏差も大きく残ってしまう。このようにシステムの状態に合ったδを設定することが制御性能を向上させるのに必要になる。

CP の初期ニューロン数は$m=a$=3 とし，そのニューロンのパラメータの初期値は(5.10)式により決定した。ただし，中心c_{ij}はc_{i0}=0.0，c_{i1}=0.5，c_{i2}=-0.5(i=1,2)とした。また，CP の入力変数I_i(i=1,2)は$I_i=r^{(i-1)}$である。

5.5.2　シミュレーション結果

結果 1：目標信号①$r=\sin(t)$(ref 1)

Table.5.1 に各システムの平均誤差面積と平均定常偏差，Fig.5.4 に各システムの振子の角度θの推移, Fig.5.5 に各システムの振子の角度の追従誤差eの推移, Fig.5.6 に Fig.5.5 の 25~30 秒間の拡大図を示す。さらに，Fig.5.7 に提案システムの CP と RC の出力の推移，Fig.5.8 に各システムのニューロン(メモリ)数mの推移, Fig.5.9 に提案システムの連結された 3 つニューロンの距離((5.5)式参照)の推移，Fig.5.10 に提案システムの結合荷重ベクトル$\mathbf{w}$のノルムの推移，Fig.5.11 にガウシアン関数の中心ベクトル$\mathbf{c}$・分散ベクトル$\boldsymbol{\sigma}$の推移，Fig.5.12 にパラメトリック式の係数ベクトル$\mathbf{p}_i\,(i=0,1,2)$の推移を示す。

Fig.5.4 より各システムは角度が$\sin(t)$に追従していることから制御に成功したことが分かる。従来法との性能比較において，まず角度の観点は，Fig.5.5 より，角度の追従誤差が 0.04 以下になったという点で，0 に近づいた速度が速いシステムは CPRCS が最も速く，次に AFCPRCS(提案システム)で，NACSC が最も遅かった。さらに，提案システムは 0 に近づいた後，チャタリングが発生している。このことは，Fig.5.7 の制御入力の推移からも確認できる。初期時刻と時刻 12 秒の CP の最適制御器構築のためのチャタリングが著しい。その影響が，Fig.5.5 の追従誤差の推移に現れている。しかし，Fig.5.6 より，提案システムは定常状態以降，全システムの中で最も低い誤差で収束していることが分かる。このことから，フィードフォワードの最適制御器の構築には成功し，その性能は他システムより優れた。誤差を抑える制御のためにチャタリングが発生したと考えられる。

定量的な観点では, Table.5.1 より, 誤差面積は, CPRCS, 提案システム(AFCPRCS), NASCS の順に優れている。定常状態において，提案システムの追従誤差の方が CPRCS より，0 近傍にあることが言える。同様のことは，Table.1 の定常偏差の結果に対しても言える。このことから，提案システムは最適制御器の構築に CPRCS と同等の時間がかかるが，一旦モデルが構築できれば，その誤差はフィードバック制御で構築された CPRCS よりも小さくなることが分かる。

Fig.5.8 のニューロン数の推移は，提案システムは初期時刻に 17 個まで増加し，時刻 2 秒まで減少を続け，最終的に最小数の$2a-1$，本シミュレーションでは$a=3$であることから$2a-1=5$個に抑えることができた。これは，1 度ニューロンを多数増加させ，その中から融合を繰り返し，徐々にニューロン数を減少させることにより，最適なニューロンを構築したと考えられる。しかし，メモリの最小数が 3 で定められている CPRCS の方が，時刻 1 秒以降において，3 個のニューロン数で構築されている。しかし，時刻 19 秒で 4 個に変動をしていることから最適入力の探索が長引いていることが言える。そして，NACSC はニューロン数が固定の MLP で構築されているため，本シミュレーションで設定した 20 個のままである。不必要なニューロンが多くあるため，適応的な NN と比べて性能は劣る。

最後に，Fig.5.10 より，初期時刻にノルムの量が急激に増加しているが，これは Fig.5.8 のニューロン数の急激な増加のためである。そして，ニューロン数が 5 個になるまでは徐々にノルムが減少し，5 個になった時刻 2 秒以降には各ノルムは収束している。このことから，ニューロンが 5 個になった後は学習が完了しているため，ノルムがほとんど増減しなかったと言える。

Table 5.1. Comparison of error areas and steady state errors of the angle among three methods for ref 1

	AFCPRCS (Proposed)	CPRCS (Chapter 3)	NASCS[51]
Error area of the angle	6.5×10^{-3}	3.9×10^{-3}	1.4×10^{-2}
Steady state errors of the angle	3.2×10^{-4}	4.5×10^{-4}	5.9×10^{-4}

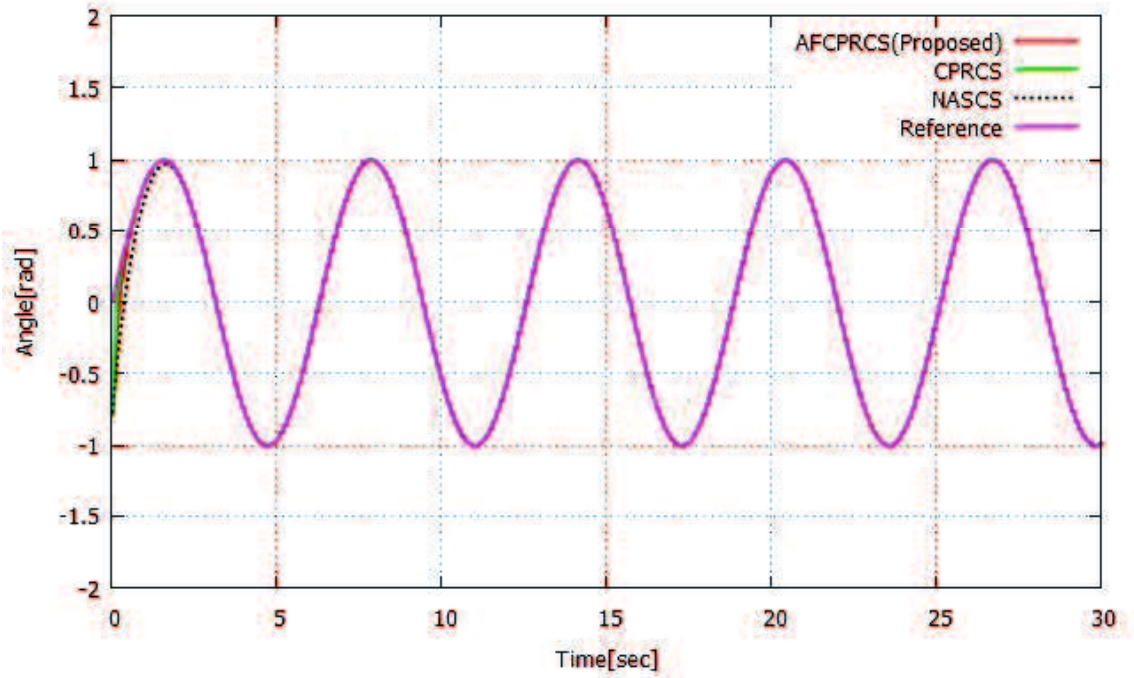

Fig.5.4. Comparison of control results of the angle among three methods for ref 1

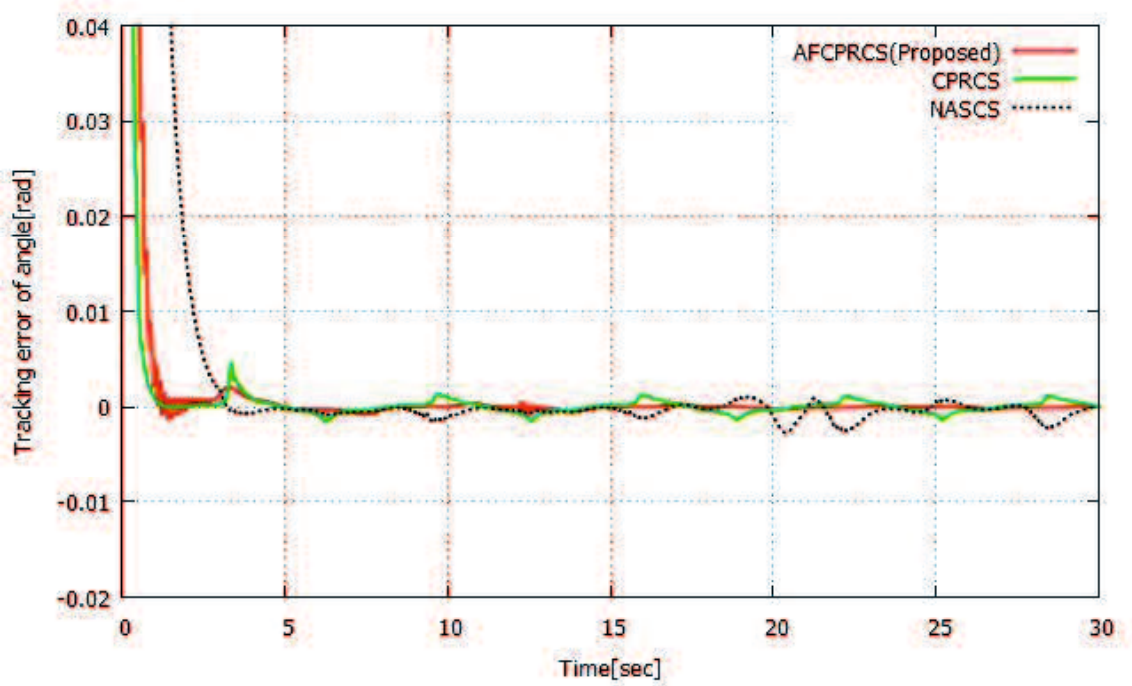

Fig.5.5. Comparison of tracking errors of the angle among three methods for ref 1

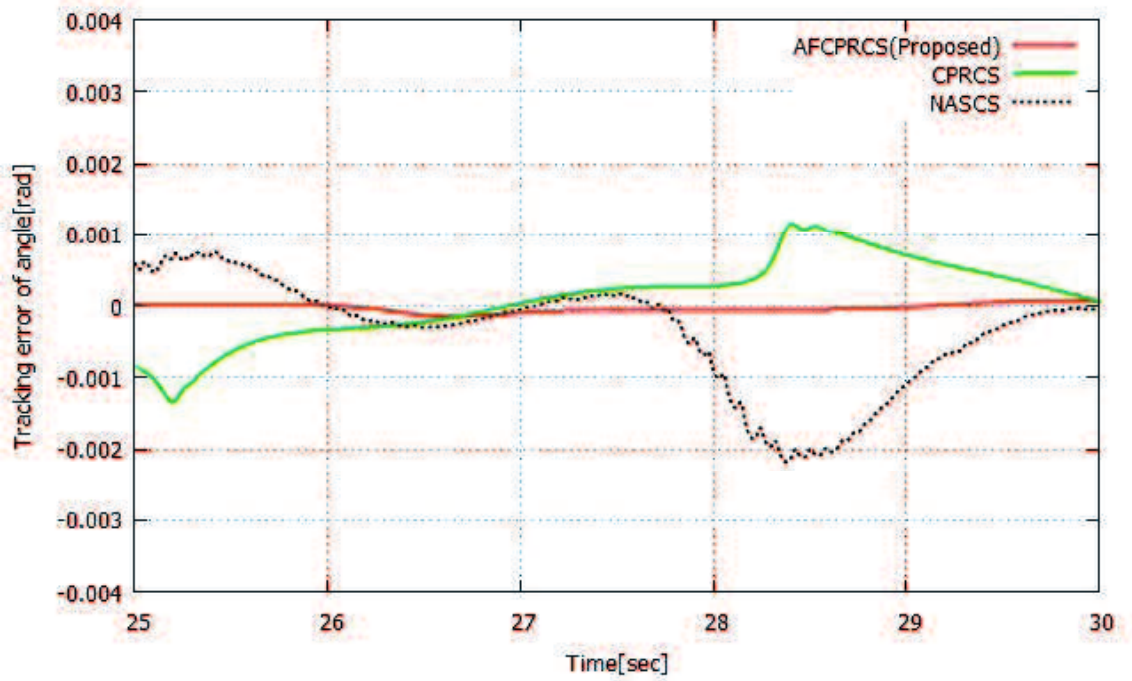

Fig.5.6. The enlarged view of Fig.5.5

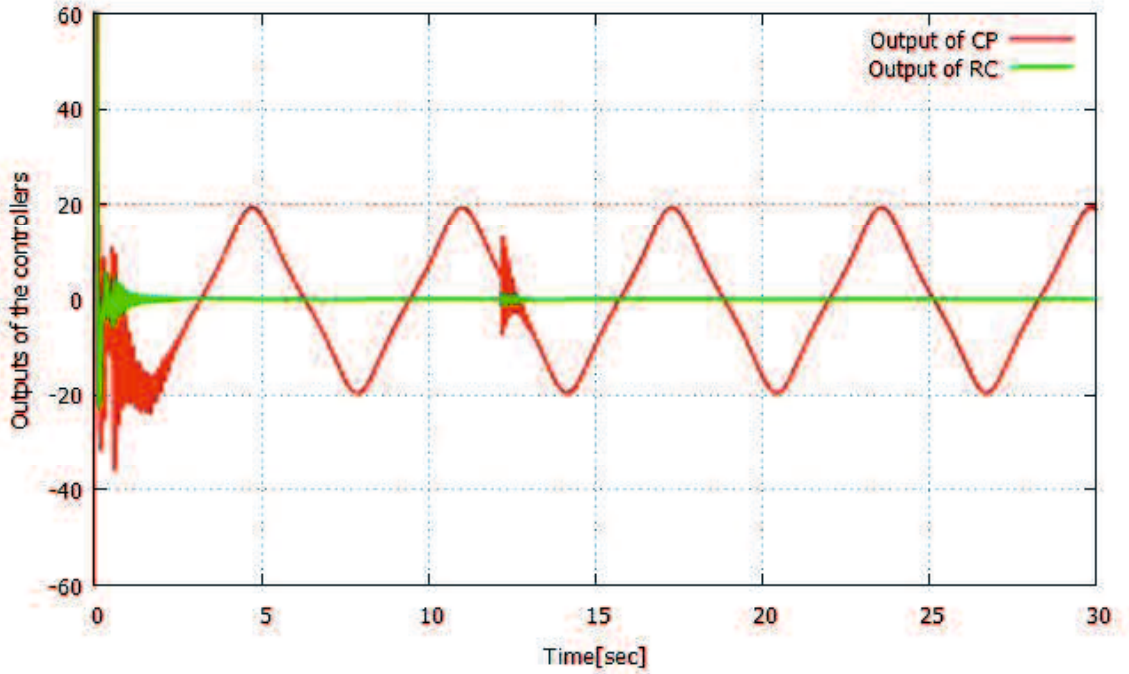

Fig.5.7. The outputs of the controllers of the proposed system for ref 1

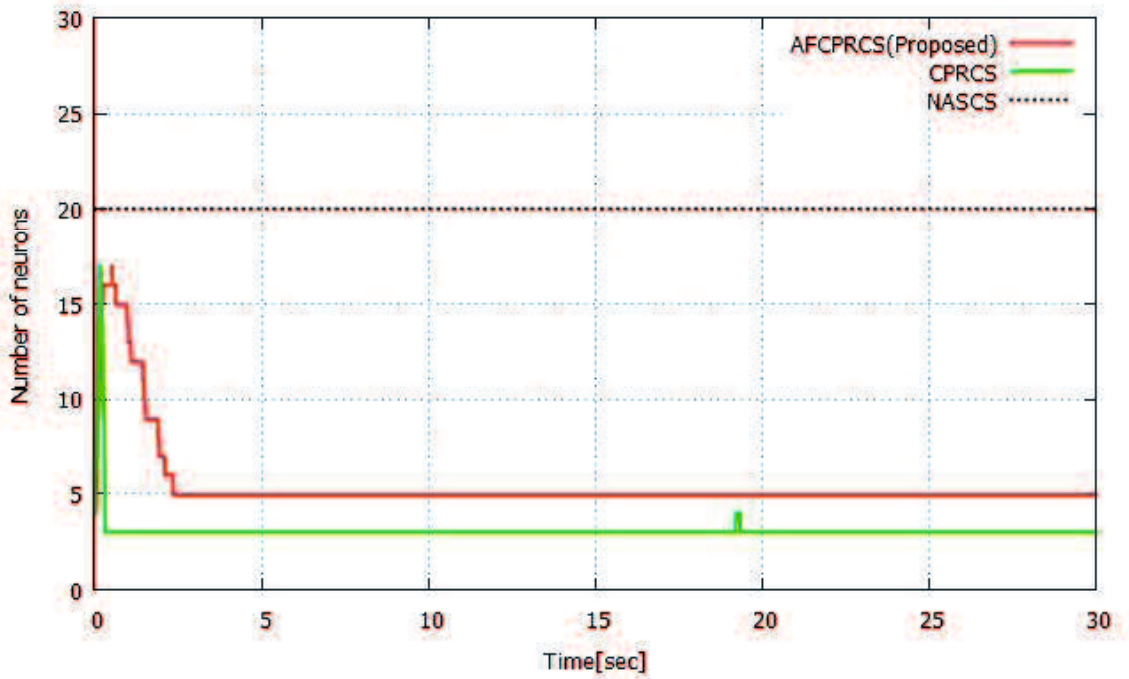

Fig.5.8. The number of neurons or memories of the each system for ref 1

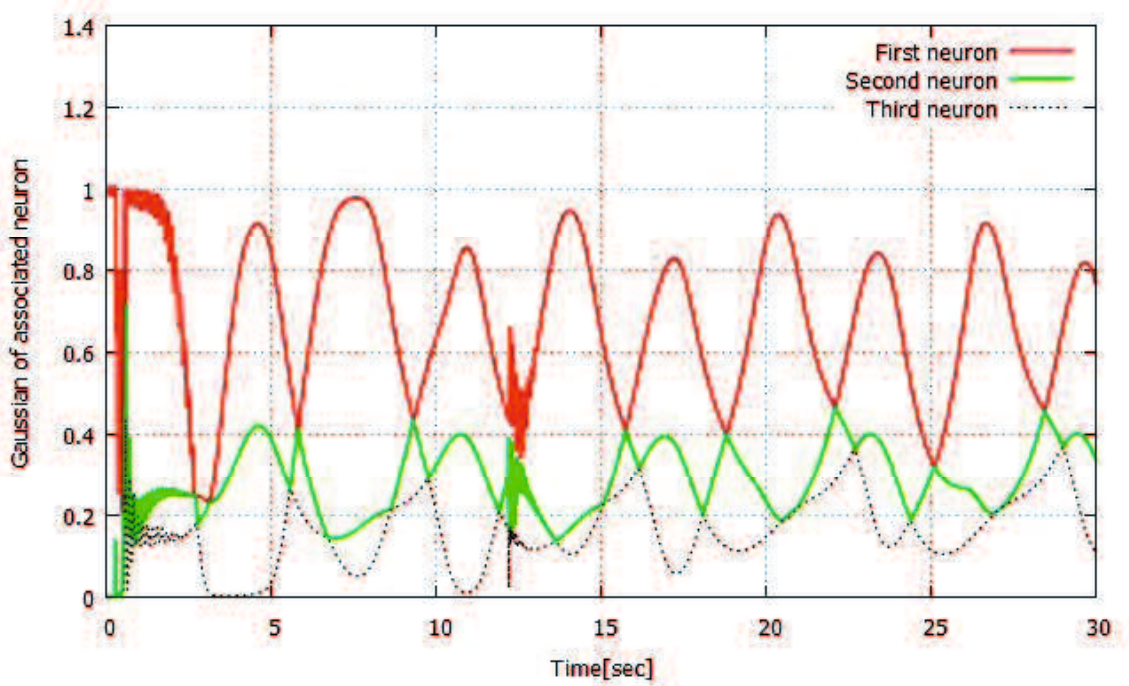

Fig.5.9. The distance between the input and the center of the associated neuron of the proposed system for ref 1

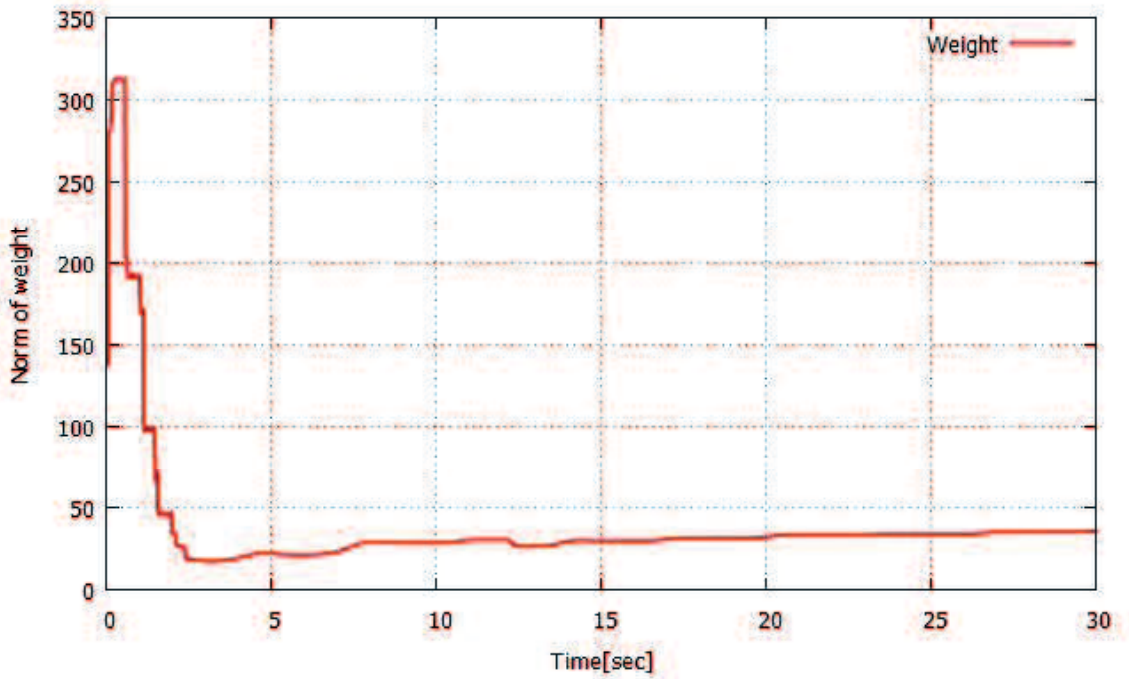

Fig.5.10. The norm of weight vector of the proposed system for ref 1

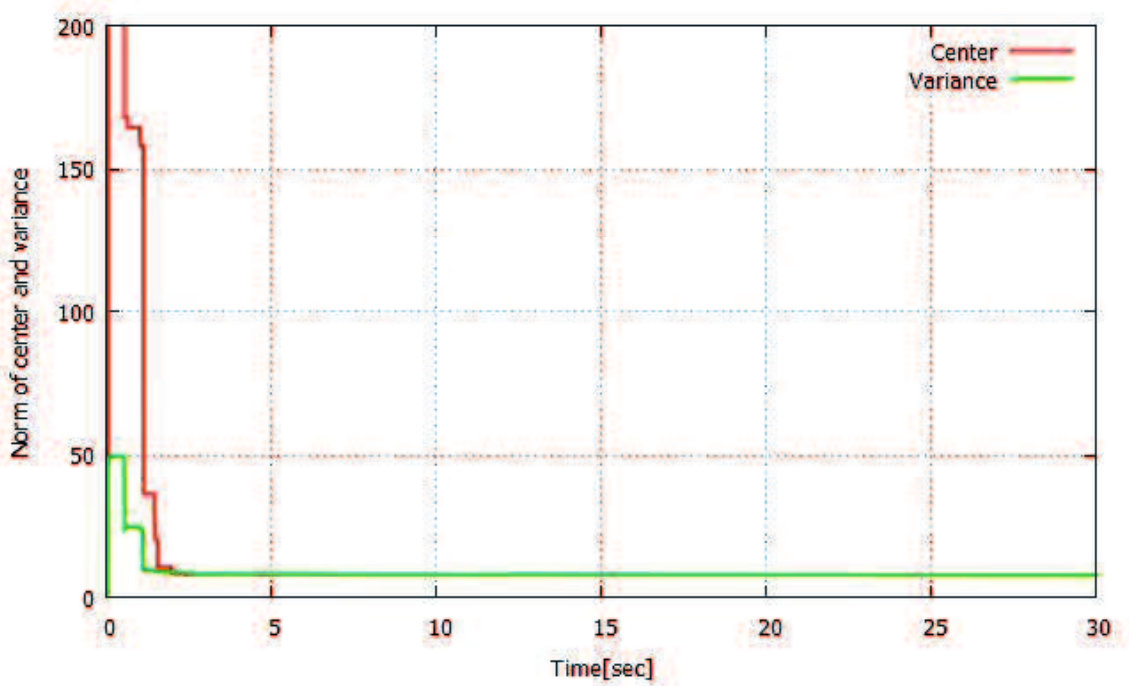

Fig.5.11. The norm of center and variance vector of the proposed system for ref 1

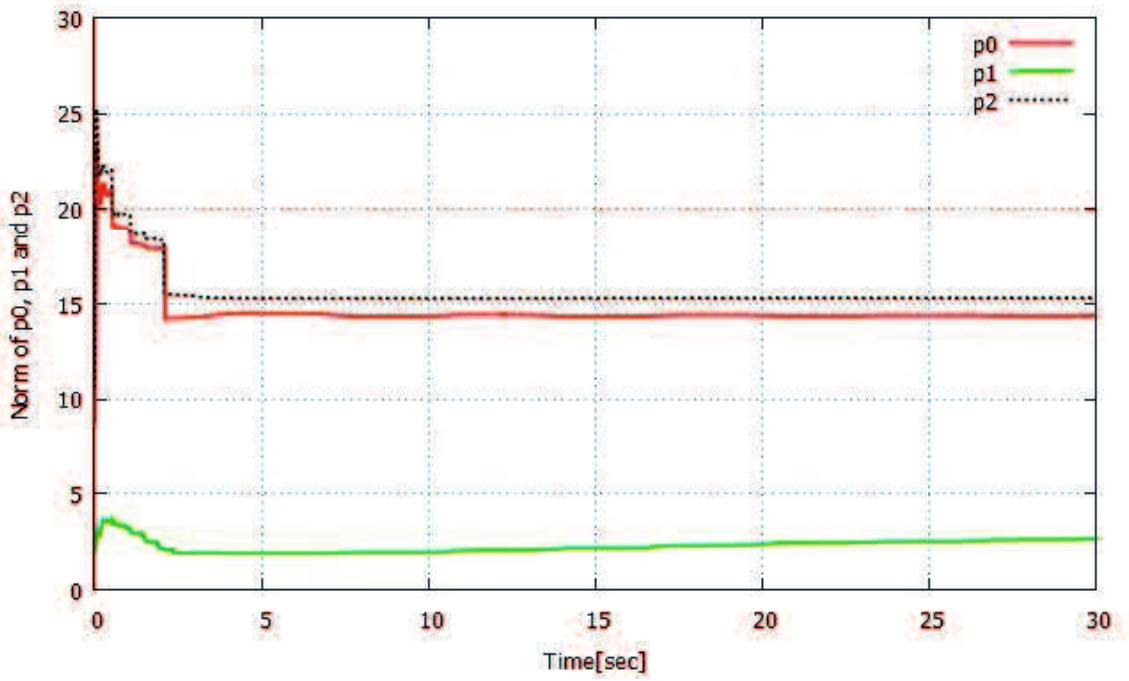

Fig.5.12. The norm of parametric parameter vectors of the proposed system for ref 1

結果 2：目標信号②$r=0.5\sin(t)+0.5\cos(2t)$(ref 2)

Table.5.2 に各システムの平均誤差面積と平均定常偏差，Fig.5.13 に各システムの振子の角度θの推移，Fig.5.14 に各システムの振子の角度の追従誤差eの推移，Fig.5.15 に Fig.5.16 の 25~30 秒間の拡大図を示す。さらに，Fig.5.17 に提案システムの CP と RC の出力の推移，Fig.5.18 に各システムのニューロン(メモリ)数mの推移，Fig.5.19 に提案システムの連結された 3 つニューロンの距離((5.5)式参照)の推移，Fig.5.20 に提案システムの結合荷重ベクトル$\mathbf{w}$のノルムの推移, Fig.5.21 にガウシアン関数の中心ベクトル$\mathbf{c}$・分散ベクトル$\boldsymbol{\sigma}$の推移, Fig.5.22 にパラメトリック式の係数ベクトル$\mathbf{p}_i\,(i=0,1,2)$の推移を示す。

Fig.5.13 より各システムは角度が$0.5\sin(t)+0.5\cos(2t)$に追従していることから制御に成功したことが分かる。しかし, 従来法である CPRCS はオーバーシュートが発生し, NASCS は他のシステムと比べて，追従に時間がかかる。それは，Fig.5.14 からも確認できる。さらなる従来法との性能比較において， Fig.5.14 より，角度の追従誤差が 0.4 以下になったという点で，0 に近づいた速度が速いシステムは CPRCS が最も速く，次に AFCPRCS(提案システム)で，NACSC が最も遅かった。また，Fig.5.16 より，提案システムは初期時刻と目標信号$0.5\sin(t)+0.5\cos(2t)$の-1 に追従に必要な制御入力 20 を発生する際の CP のチャタリングが著しい。しかし，Fig.5.15 より，提案システムは定常状態以降，全システムの中で最も低い誤差で収束していることが分かる。このことから，フィードフォワードの最適制御器の構築には成功し，その性能は他システムより優れた。誤差を抑える制御のためにチャタリングが発生したと考えられる。

定量的な観点では, Table.5.2 より, 誤差面積は, CPRCS, 提案システム(AFCPRCS), NASCS の順に優れている。定常状態において，提案システムの追従誤差の方が CPRCS より，0 近傍にあることが言える。同様のことは，Table.4.2 の定常偏差の結果に対しても言える。このことから，提案システムは最適制御器の構築に CPRCS と同等の時間がかかるが，一旦モデルが構築できれば，その誤差はフィードバック制御で構築された CPRCS よりも小さくなることが分かる。

Fig.5.17 のニューロン数の推移は，提案システムは初期時刻に 17 個まで増加し，時刻 2 秒まで減少を続け，最終的に最小数の$2a-1$，本シミュレーションでは$a=3$であることから$2a-1=5$個に抑えることができた。これは，1 度ニューロンを多数増加させ，その中から融合を繰り返し，徐々にニューロン数を減少させることにより，最適なニューロンを構築したと考えられる。しかし，メモリの最小数が 3 で定められている CPRCS の方が，時刻 1 秒以降において，3 個のニューロン数で構築されており，初期時刻のニューロンの増加も CPRCS の方が少ない。これは，CP の入力変数I_iが，CPRCS は時間経過するにつれ単調減少する追従誤差e_iに対し，提案システム(AFCPRCS)は時間変動する目標信号rであるため，CP の構築に時間がかかることが考えられる。そして, NACSC はニューロン数が固定の MLP で構築されているため，本シミュレーションで設定した 20 個のままである。不必要なニュ

ーロンが多くあるため，適応的な NN と比べて性能は劣る。

最後に，Fig.5.19~Fig.5.21 より，初期時刻にノルムの量が急激に増加しているが，これは Fig.5.17 のニューロン数の急激な増加のためである。そして，ニューロン数が 5 個になるまでは徐々にノルムが減少し，5 個になった時刻 4 秒以降には各ノルムは収束している。時刻 24 秒で 1 度ニューロンが増加したが，ノルムに大きな影響を与えなかったことから，ニューロンが 5 個になった後は学習が完了していると言える。

Table 5.2. Comparison of error areas and steady state errors of the angle among three methods for ref 2

	AFCPRCS (Proposed)	CPRCS (Chapter 3)	NASCS[51]
Error area of the angle	9.0×10^{-3}	8.5×10^{-3}	3.0×10^{-2}
Steady state errors of the angle	1.9×10^{-4}	3.8×10^{-4}	8.0×10^{-3}

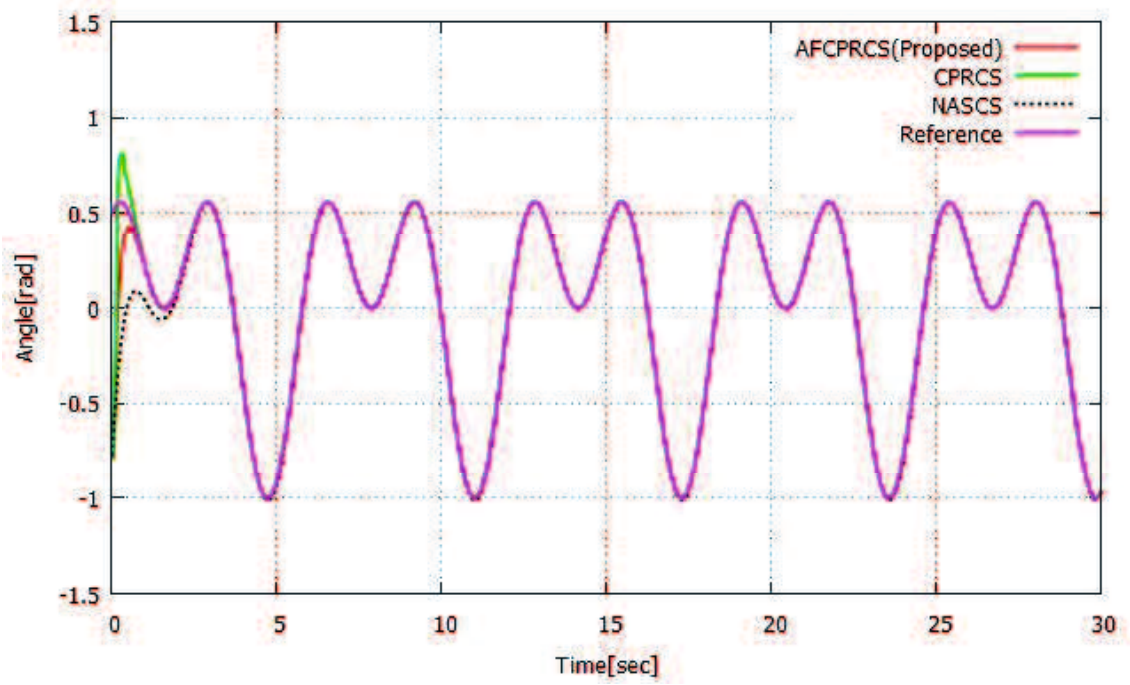

Fig.5.13. Comparison of control results of the angle among three methods for ref 2

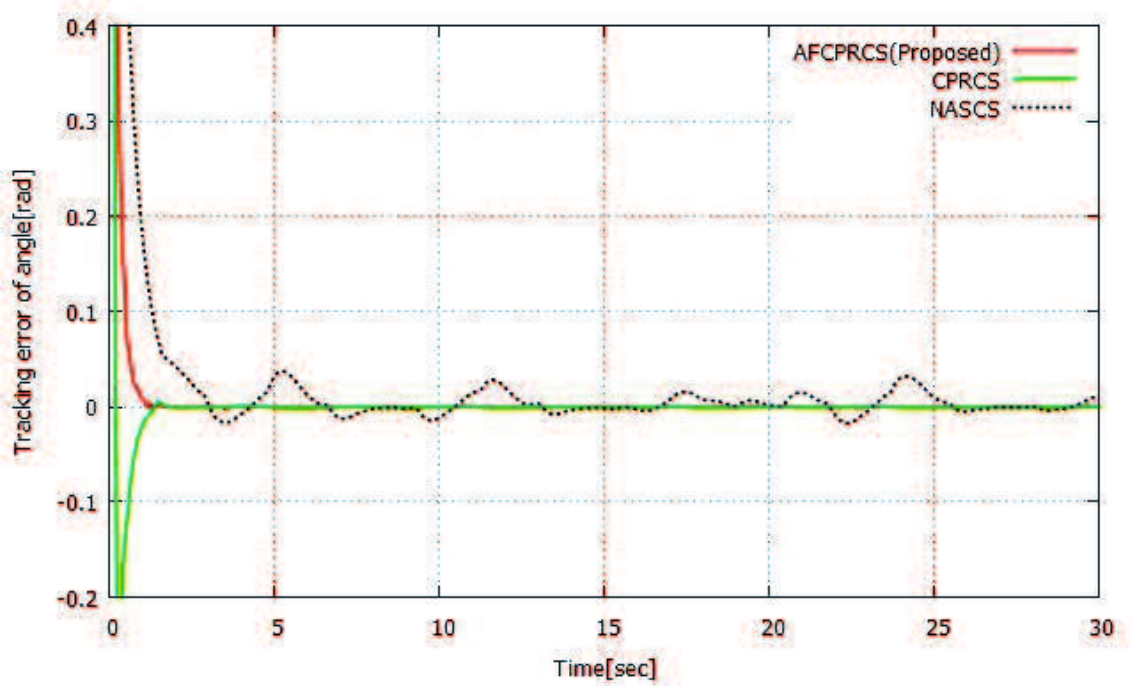

Fig.5.14. Comparison of tracking errors of the angle among three methods for ref 2

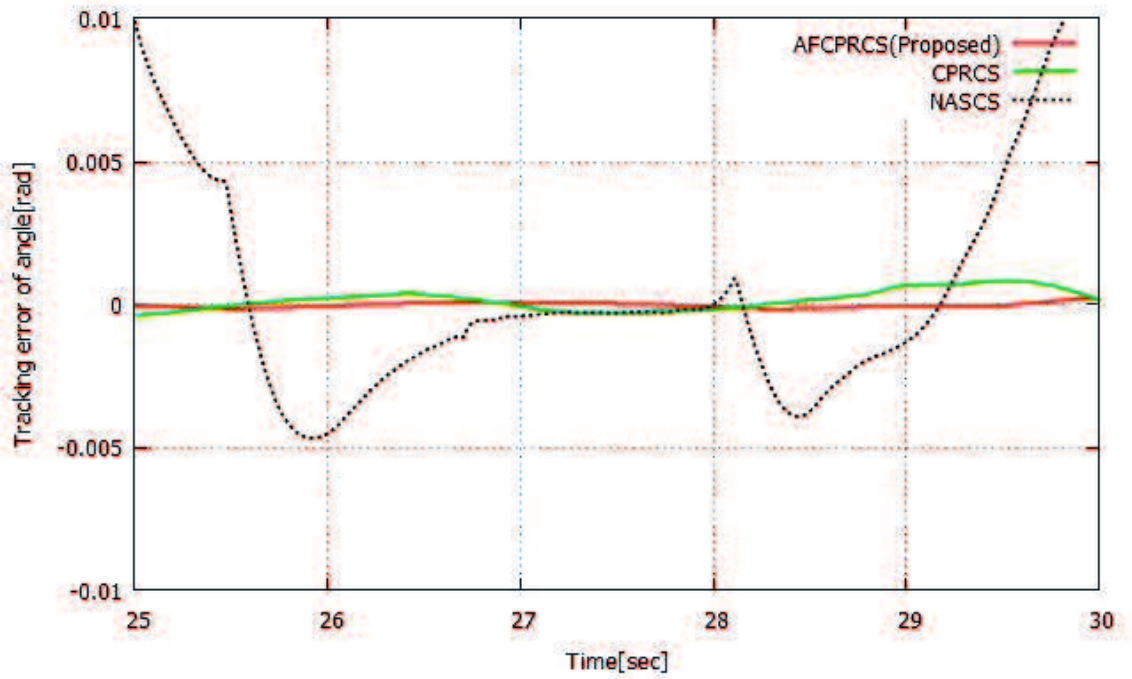

Fig.5.15. The enlarged view of Fig.5.5

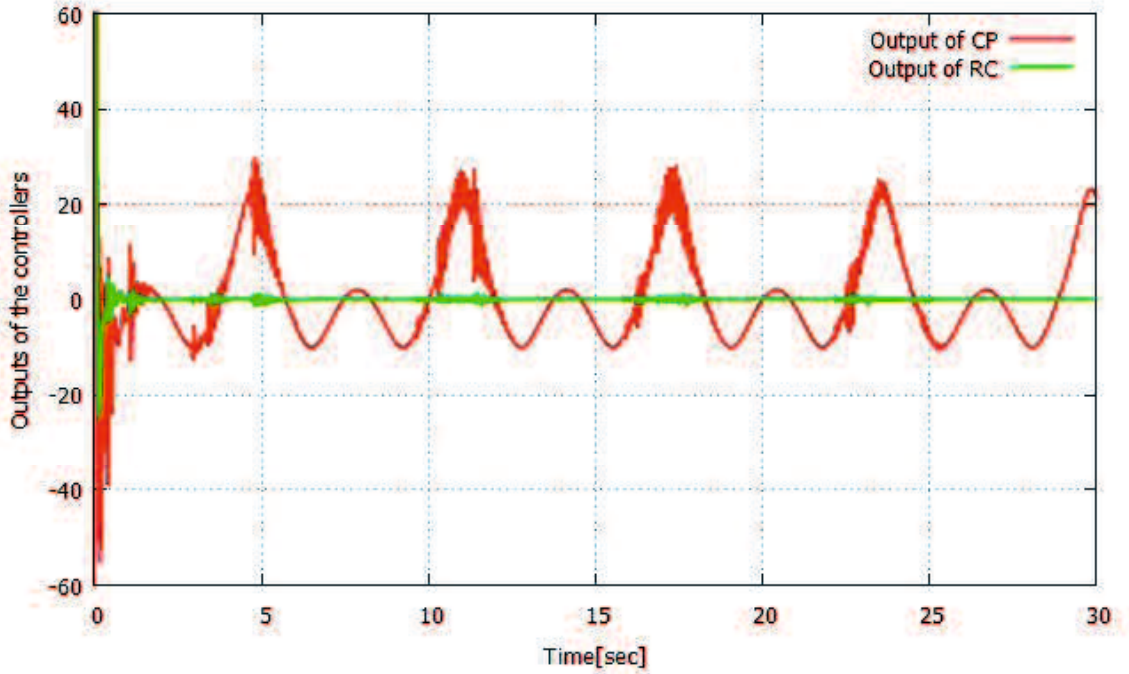

Fig.5.16. The outputs of the controllers of the proposed system for ref 2

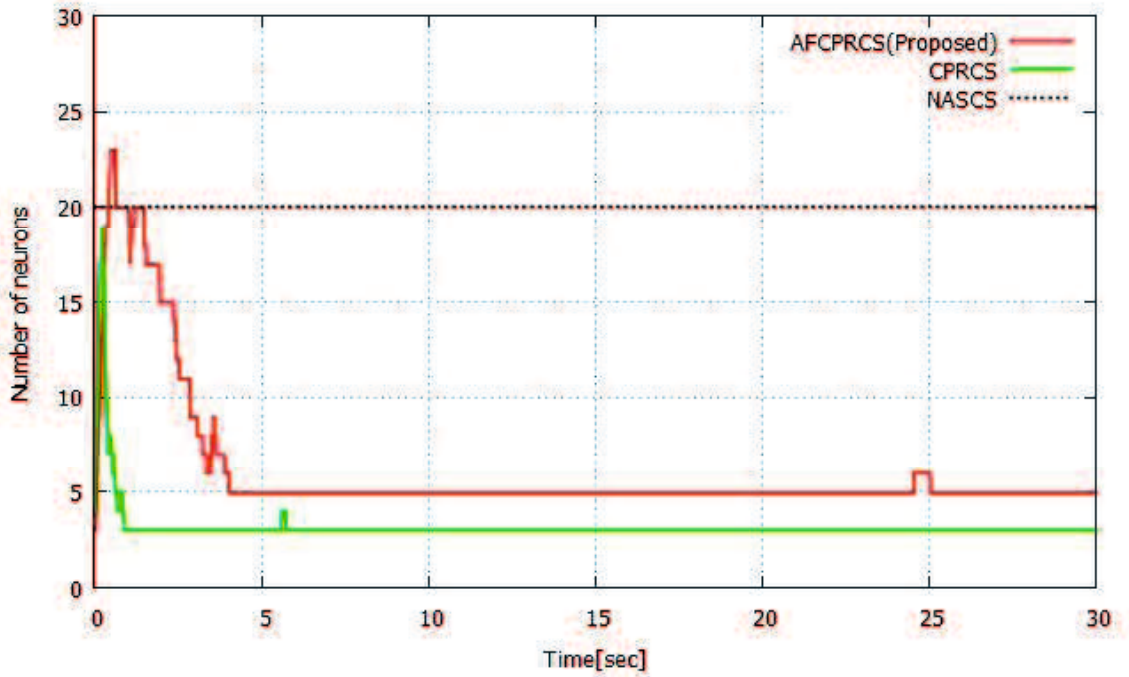

Fig.5.17. The number of neurons or memories of the each system for ref 2

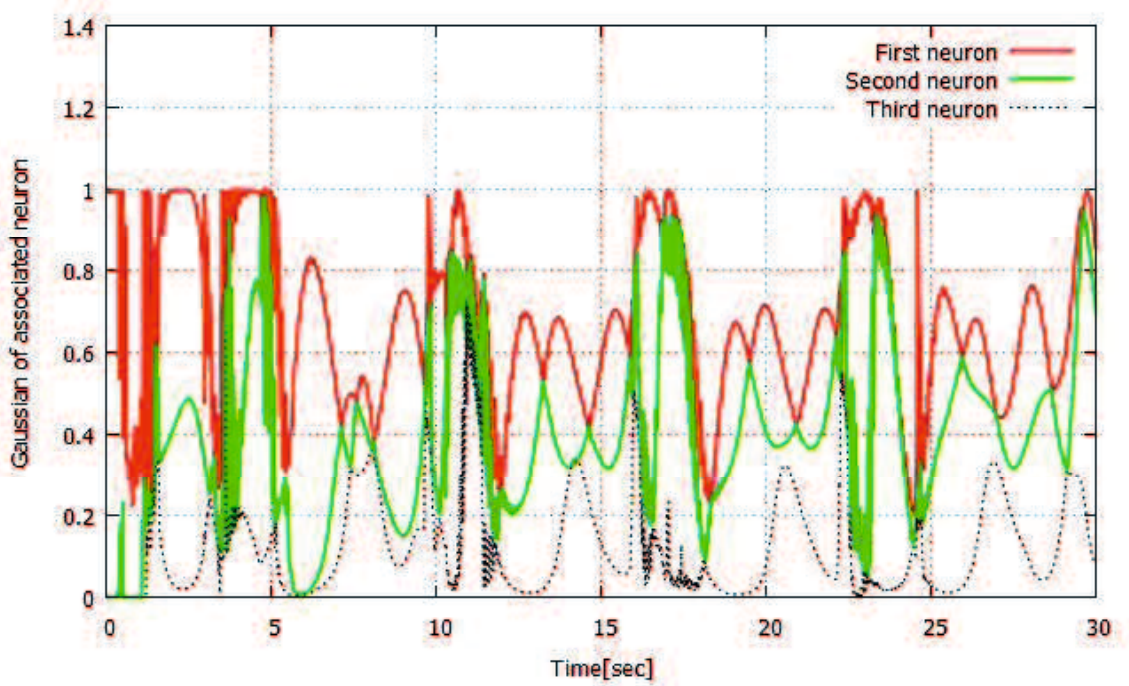

Fig.5.18. The distance between the input and the center of the associated neuron of the proposed system for ref 2

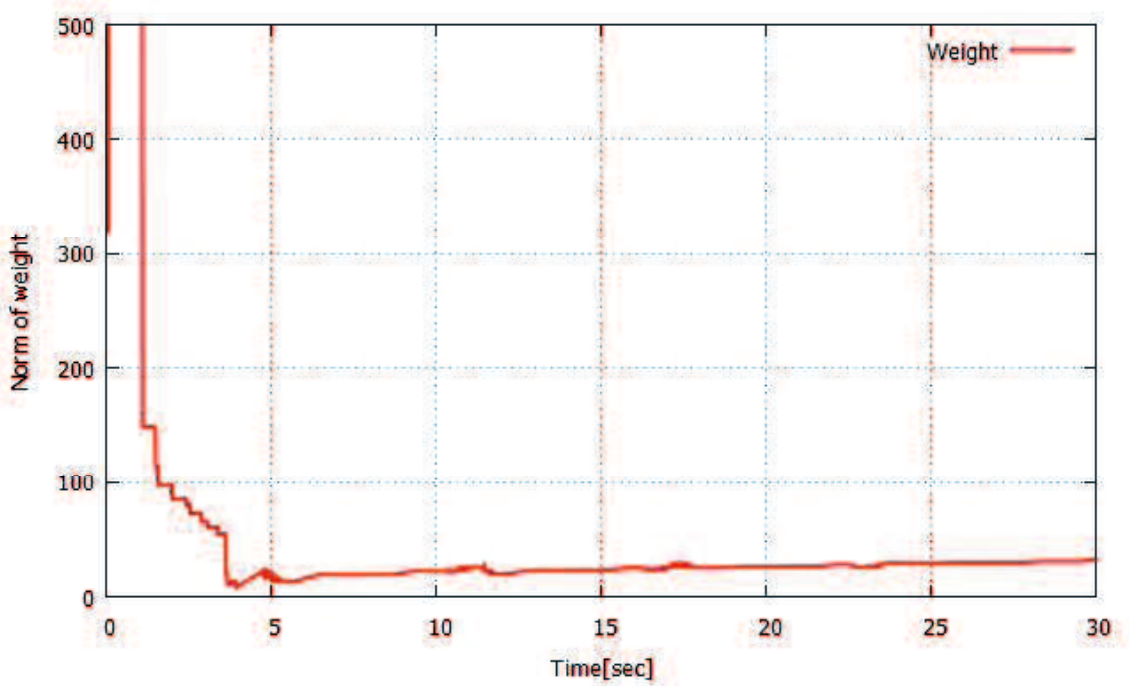

Fig.5.19. The norm of weight vector of the proposed system for ref 2

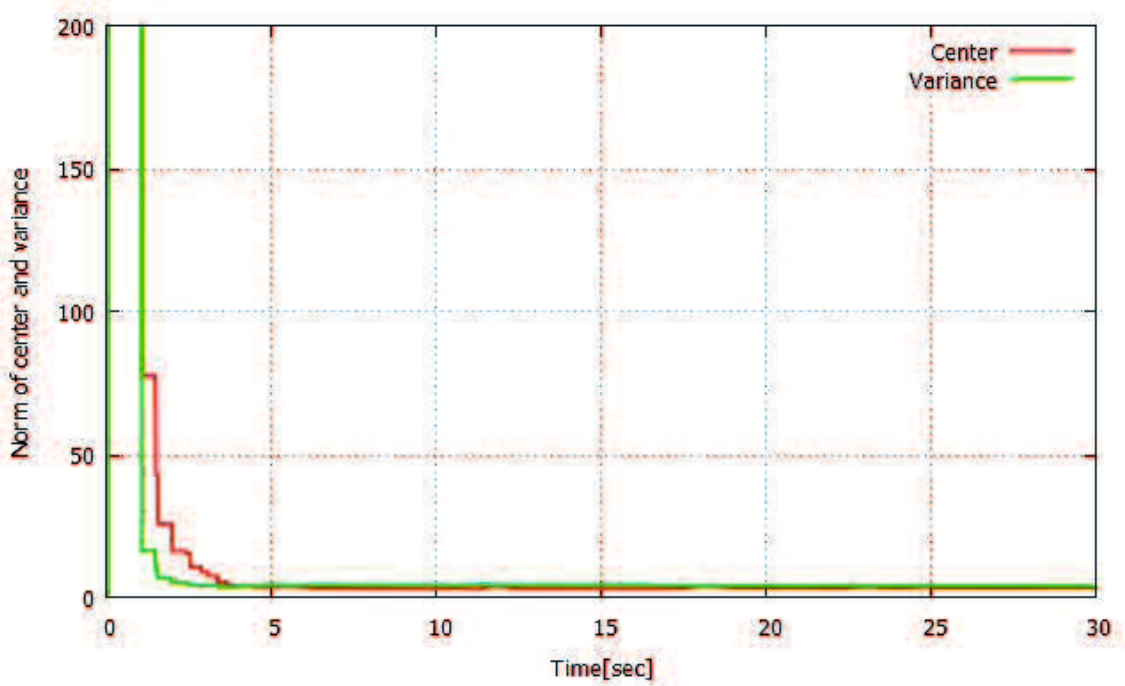

Fig.5.20. The norm of center and variance vector of the proposed system for ref 2

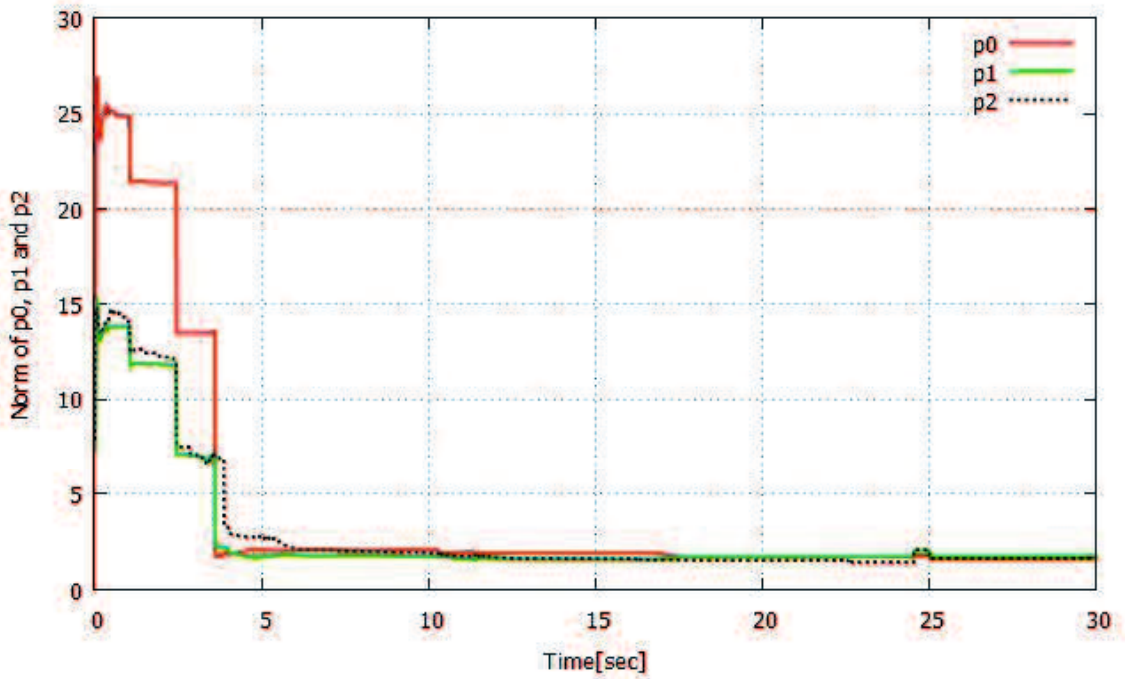

Fig.5.21. The norm of parametric parameter vectors of the proposed system for ref 2

5.6 まとめ

フィードバック誤差学習の設計に基づき，時間変動する目標信号の追従問題に対応した自己融合小脳パーセプトロンモデル利用型ロバスト制御システムを提案し，台車付き倒立振子シミュレーションにより，その有効性を示した。第 2 章の提案システムと比べて，誤差面積から CP の最適構築の速度が劣るが，定常偏差からより最適な CP の構築が可能だと言える。また，チャタリングが著しいが，オーバーシュートを無くすことができた。

今後の展開として，$\sin(t)$のように，正負の符号が反転する目標信号に対応可能な制御器の設計が考えられる。これにより，入力信号のチャタリングが抑えられ，制御性能の向上が期待できる。また，角度の振幅±5°の制御信号$0.087\sin(t)$のように，微小の振幅波形の制御も考えられる。この制御により，本提案システムのさらなる優位性を示すことができる。他にも，本シミュレーションでは$\sin(t)$や$0.5\sin(t)+0.5\cos(2t)$のように 1 つの制御信号を目標に追従制御を行ったが，まず，$\sin(t)$の学習結果のニューロンを融合されないように保存し，次は，$0.5\sin(t)+0.5\cos(2t)$の追従を行い，そのニューロンの情報も保存する。そして，これらのニューロンの情報の保存を考えれば，オフラインになるが，2 種類の追従問題を扱うことができる。最後に，ニューロンの情報の保存を考えれば，移動距離を考慮した台車付き倒立振子システムにおける，角度と距離のように，複数の状態が考えられる制御対象の制御も可能である。

第 6 章　小脳パーセプトロン改良モデルの合意問題への適用

6.1　はじめに

第 3~5 章の計算シミュレーションでは，単入力単出力(SISO)システムを制御対象に用いた。しかし，より複雑なシステムである多入力多出力(MIMO)システムの制御対象を扱っていない。第 4 章の提案システムである「小脳パーセプトロン改良モデル利用型ロバスト制御システム(CPRCS)」は SISO システムを前提として，安定性解析を行っているため，このシステムを MIMO システムに適応させることは困難である。しかし，第 5 章の提案システムである「自己融合小脳パーセプトロン改良モデル利用型ロバスト制御システム(AFCPRCS)」は，学習アルゴリズムを拡張すれば MIMO システムにも適用可能である。

また，近年，制御対象システムの大規模化・複雑化に伴い，単一の制御対象よりも，複数の制御対象に対する制御方式が重要になっている。その中で，それぞれが自律的に意思決定を行い，複数のシステム(エージェント)から構成されるマルチエージェントシステム(MAS)の協調制御が注目を集めている[78-96]。MAS とは複数のエージェントがある共通の目的のために，各エージェントが互いの情報を得ることで協調して，単一では困難な課題をシステム全体で達成することである。しかし，MAS は非線形性や未知環境との相互作用などのため，不確かなダイナミクスを持ったエージェントの制御系の設計は困難とされていた[78,88,95]。

これらの解決法として, Hou らは NN の導入により非線形性や不確かさが含まれる制御対象のダイナミクスでも，制御可能にした分散型ロバスト適応制御システム(Decentralized Robust Adaptive Control System，DRACS)を提案し，合意問題やフォーメーション制御を行っている[78]。しかし，NN の隠れ層のニューロン数が固定化されているため，エージェントのダイナミクスの複雑さの程度に応じたニューロン数設定の事前検討が必要であることや，エージェントの移動範囲によってニューロンの初期設定が異なる欠点を有する。そのため，各エージェントは環境変化に適切に対応することは困難だと考えられる。この問題を解決し，かつ合意問題において優れた性能を示す MAS に適用可能な AFCPRCS を提案する。

提案システムの有効性を示すため，MIMO システムで構築されるエージェント 6 体による合意問題において, 優れた結果を示している前述の DRACS と提案システムとの性能比較シミュレーションにより，その有効性を示す。MAS を制御対象にした理由は 2 つある。

1 つは，従来法である CPRCS と提案システムとの単一のエージェントによる計算機シミュレーションでは，大きな性能の差が見られないことである。しかし，単一のエージェントも MAS も，エージェントの制御対象システムのダイナミクスが非線形で，かつ未知であることは共通である。ただし，CPRCS は単入力単出力システムを制御対象としているため，

MAS の計算機シミュレーションの比較対象にはしていない。

他の 1 つは，自己融合メカニズムの有効性を示すためである。自己融合メカニズムは，従来法である CPRCS の問題点の制御システムの一時的な不安定を取り除く手法である。この有効性を示すために，制御対象の状態変数が不安定になりやすい MAS の合意問題を適用した。

6.2 マルチエージェントシステム

6.2.1 グラフ理論

マルチエージェントシステム(MAS)において, エージェント同士が位置情報などを互いに通信するネットワーク構造は，グラフ $G=(V,E)$ で表わされる。ここで，$V=(v_1,v_2,\cdots,v_n)$ は頂点集合，n はエージェント数(頂点数)，$E\subseteq V\times V$ は辺集合である。頂点 v_i の近傍集合 N_i は $N_i=\{v_j\in V:(v_j,v_i)\in E\}$ として表わされる。また，グラフの頂点 v_i を i 番目エージェントの状態変数 $\mathbf{x}_i$ とする。ただし，制御入力の設計は，エージェント i の近傍エージェント群 N_i の情報のみを用いる。

グラフを代数的に表現するために知られているものとして，隣接行列 A，次数行列 D，グラフラプラシアン行列 L がある。隣接行列 $A=\lfloor a_{ij}\rfloor$ は次式で表現される。

$$a_{ij}=\begin{cases}1 & (v_j\in N_i)\\ 0 & (v_j\notin N_i)\end{cases} \tag{6.1}$$

また，次数行列 D は，v_i の次数が i 行目の対角要素である行列である。即ち，

$$D=diag(d_1,d_2,\cdots,d_n) \tag{6.2}$$

ただし，

$$d_i=\sum_{j=1}^{n}a_{ij}$$

である。また，グラフラプラシアン行列 $L=\lfloor l_{ij}\rfloor$ は，隣接行列 A と次数行列 D を用いて，

$$L=D-A \tag{6.3}$$

で定義される。このグラフラプラシアン行列の i 行 j 列目の要素 l_{ij} をフィードバックの制御入力の設計に用いる((6.10)式参照)。

6.3　制御対象の定式化

本研究では，n 個のエージェントで構成される MAS を考える。各エージェントは次の 1 階非線形微分方程式で表わされる。

$$\dot{\mathbf{x}}_i = \mathbf{f}_i(\mathbf{x}_i) + \mathbf{g}_i(\mathbf{x}_i)\mathbf{u}_i \tag{6.4}$$

ここで，$\mathbf{x}_i = [x_{i1}, x_{i2}, \cdots, x_{1m}]^T$ は i 番目エージェントの状態変数ベクトル，m は状態の次元数，$\mathbf{f}_i, \mathbf{g}_i$ は未知の連続関数ベクトル(ただし $\mathbf{g}_i > 0$)，$\mathbf{u}_i$ は制御入力ベクトルである。また，$a_{ij} = 1$ を満たす状態変数行列を X_j とする(Fig6.1 参照)。

制御目的は，MAS 合意状態を達成すること，即ち次式を満たす i 番目のエージェントに対する制御入力ベクトル $\mathbf{u}_i$ を設計することである。

$$\lim_{t\to\infty} \left\| \mathbf{x}_i(t) - \mathbf{x}_j(t) \right\| = 0, \quad i, j = 1,2,\cdots,n \tag{6.5}$$

また，グラフ G を無向グラフ，即ち $\forall i, j$ について $a_{ij} = a_{ji}$ であると仮定する。そのとき，制御システムは全エージェントの状態変数の時間微分が $\dot{\mathbf{x}}_i = 0$ (i=1,2,..., n)になるように制御する。ここで，そのときの合意値ベクトル $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_n]^T$ は全エージェントの状態変数の平均値であり，次式で表わされる。

$$\boldsymbol{\alpha} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i(0) \tag{6.6}$$

ここで，$\mathbf{x}_i(0)$ は $\mathbf{x}_i$ の $t = 0$ における初期値である。

合意には 2 つ定義があり，1 つは(6.5)式を満足する状態を初期値に依存した値に，合意が達成されたといい，他の 1 つは(6.6)式を満足した場合で平均合意が達成されたという。

6.4　自己融合小脳パーセプトロン改良モデル利用型制御システム

MAS において，各エージェントに搭載する自己融合小脳パーセプトロンモデル利用型制御システム(AFCPCS)を Fig.6.1 に示す。このシステムは，MAS が合意を達成するための i 番目エージェントに対応する制御器は，i 番目エージェントの近傍エージェントの情報のみを利用し，設計する。制御初期は CFC であるフィードバック合意制御器(FCC)が制御を行い，その時の FCC の出力 $\mathbf{u}_i^f$ を誤差関数として，CP の学習を行う。FCC の出力 $\mathbf{u}_i^f$ が 0 になるとき，CP の学習が完了し，CP の出力 $\mathbf{u}_i^c$ は，理想的な制御器 $\mathbf{u}_i^*$ になったことを意味する。制御対象システム((6.4)式)への入力 $\mathbf{u}_i$ を次式で表わす。

$$\mathbf{u}_i = -\left(\mathbf{u}_i^c + \mathbf{u}_i^f\right) \tag{6.7}$$

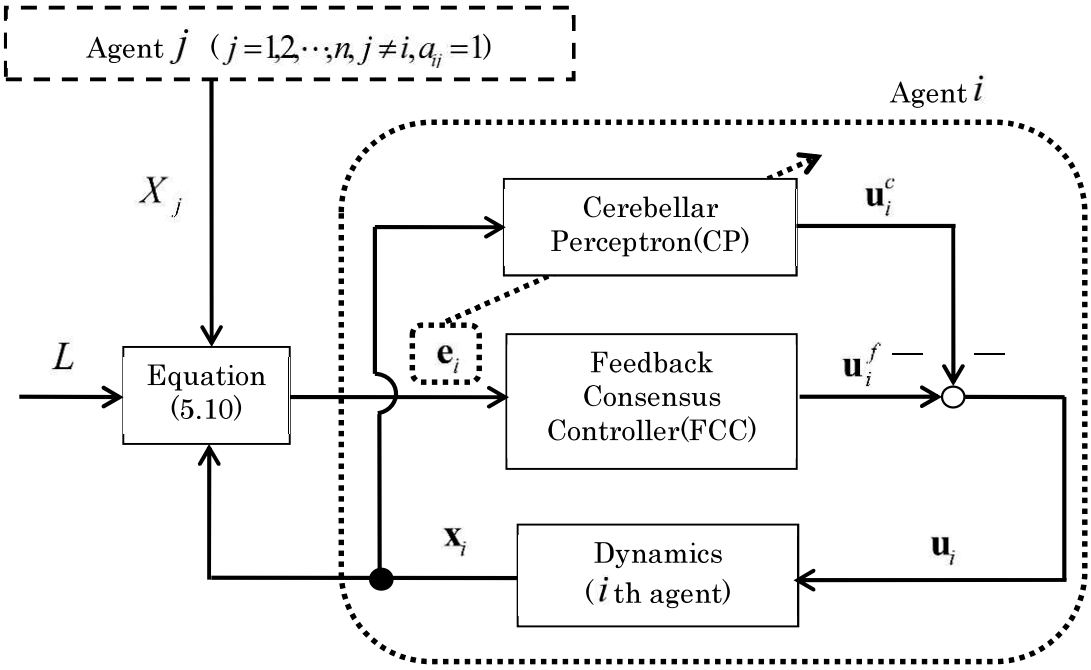

Fig.6.1. Structure of the proposed system for i th agent

$$\mathbf{u}_i^c = \left(\mathbf{w}_i^{assoc}\right)^T \mathbf{b}_i^{assoc} + \mathbf{p}_i^{assoc} \tag{6.8}$$

$$\mathbf{u}_i^f = \frac{\left(\delta^4+1\right)}{2\delta^3}\mathbf{e}_i \tag{6.9}$$

ただし，合意誤差ベクトル$\mathbf{e}_i$は次式で計算される。

$$\mathbf{e}_i = \sum_{j=1}^{n} l_{ij}\mathbf{x}_j \tag{6.10}$$

ここで，$\mathbf{u}_i^c$はi番目エージェントのCPの出力ベクトルであり，各エージェントが持つ未知の連続ベクトル関数$\mathbf{f}_i/\mathbf{g}_i$を近似する制御入力である。$\mathbf{u}_i^f$は$i$番目エージェントのフィードバック合意制御入力であり，CPによる近似誤差を補償する。ただし，δは減衰定数であり，小さいほど性能が優れる。(6.7)式を負の制御入力としている理由は，各制御器が$\mathbf{u}_i^c = \mathbf{f}_i/\mathbf{g}_i$，$\mathbf{u}_i^f = 0$に近似したときに，(6.4)式より$\mathbf{x}_i = 0$となり，各エージェントの動作が完了するからである。また，$\mathbf{w}_i^{assoc}$は連結された結合荷重ベクトル，$\mathbf{b}_i^{assoc}$は連結されたガウシアン関数ベクトル，$\mathbf{p}_i^{assoc}$は連結されたパラメトリック式ベクトルである。合意問題における$\mathbf{b}_i^{assoc}$((4.6)式参照)及び$\mathbf{p}_i^{assoc}$((4.7)式参照)の入力変数は$I_z = x_{iz}$である。

6.4.1 学習アルゴリズム

提案システムにおけるフィードバック誤差学習は, (5.4)式より FCC の出力 $\mathbf{u}_i^f$ を誤差関数として学習を行い，この出力が 0 になるとき，CP は最適な制御器が構築される。FCC の出力 $\mathbf{u}_i^f$ を 0 にすることは(6.9)式より，(6.10)式の合意誤差を 0 にすることと同義である。したがって，$\mathbf{u}_i^f$ の代わりに，$\mathbf{e}_i$ を誤差関数として用いて，次の学習則でパラメータの更新を行う。

$$\Delta \mathbf{w}_i^{assoc} = \eta_w \cdot \frac{\partial \mathbf{u}_i^c}{\partial \mathbf{w}_i^{assoc}} \cdot \mathbf{e}_i \tag{6.11}$$

$$\Delta \mathbf{c}_i^{assoc} = \eta_c \cdot \frac{\partial \mathbf{u}_i^c}{\partial \mathbf{c}_i^{assoc}} \cdot \mathbf{e}_i \tag{6.12}$$

$$\Delta \boldsymbol{\sigma}_i^{assoc} = \eta_\sigma \cdot \frac{\partial \mathbf{u}_i^c}{\partial \boldsymbol{\sigma}_i^{assoc}} \cdot \mathbf{e}_i \tag{6.13}$$

$$\Delta \mathbf{p}_i^{assoc} = \eta_i \cdot \frac{\partial \mathbf{u}_i^c}{\partial \mathbf{p}_i^{assoc}} \cdot \mathbf{e}_i \tag{6.14}$$

ここで，$\mathbf{c}_i^{assoc}$は連結されたガウシアン関数の中心ベクトル，$\left(\boldsymbol{\sigma}_i^{assoc}\right)^2$ は連結されたガウシアン関数ベクトルの広がりベクトル，$\eta_w, \eta_c, \eta_\sigma, \eta_i (i=0,1,\ldots,n)$は正の学習定数である。学習が進むにつれて$\mathbf{e}_i$ が小さくなってゆき，CP の学習が収束することを期待している。

6.5 計算機シミュレーション

6.5.1 合意問題

MIMO システムで構成される 6 体のエージェントの合意問題を考える。全エージェントは 2 次元平面上を移動するものとする。エージェントのダイナミクスは次式で表わされる。

$$\frac{d}{dt}\begin{pmatrix} x_{i1}(t) \\ x_{i2}(t) \end{pmatrix} = \begin{pmatrix} x_{i2}(t)\sin\left(k_{i1}x_{i1}(t)\right) \\ x_{i1}(t)\cos\left(k_{i2}x_{i2}^2(t)\right) \end{pmatrix} + \mathbf{u}_i \tag{6.15}$$

ここで，k_{i1}, k_{i2} は設計パラメータである。

提案システム(AFCPCS)の追従性能を検証するため，従来法のニューロン数固定型 NN を用いた分散型ロバスト適応制御システム(DRACS)[78]との性能比較シミュレーションを行った。

制御性能を評価する指標として MAS 全体の距離誤差と合意誤差を利用する。各エージェントの座標と合意値 $\boldsymbol{\alpha}$ 間の距離を次式で定義する。

$$x_i^{\alpha} = \sqrt{\left(x_{i1} - \alpha_1\right)^2 + \left(x_{i2} - \alpha_2\right)^2} \tag{6.16}$$

MAS 全体の距離誤差(distance error)は次式で表わされる。

$$x^{\alpha} = \sum_{i=1}^{n} x_i^{\alpha} \tag{6.17}$$

合意誤差の計算は次のように行われる。各エージェントについての合意誤差の計算は次式で表わされる。

$$x_i^e = x_{i1}^e + x_{i2}^e = \left|x_{i1} - x_1^{ave}\right| + \left|x_{i2} - x_2^{ave}\right| \tag{6.18}$$

ただし，

$$x_1^{ave} = \frac{1}{n}\sum_{i=1}^{n} x_{i1}, \quad x_2^{ave} = \frac{1}{n}\sum_{i=1}^{n} x_{i2}$$

であり，MAS 全体の合意誤差(consensus error)は次式で表わされる。

$$x^e = \sum_{i=1}^{n} x_i^e \tag{6.19}$$

制御目的は，非線形ダイナミクス(6.15)に制御入力(6.7)を与えることで，6 体($n = 6$)のエージェントの位置を合意値 $\boldsymbol{\alpha}$ へと収束させ，MAS が合意を達成することである。サンプリ

ングタイムを 0.01[s]，制御時間を 10.0[s]とする。従来システム(DRACS)との比較のため，文献[77]と同様のエージェントのパラメータとネットワーク構造を用いる。エージェントのダイナミクス中のパラメータ k_{i1}, k_{i2} を以下に示す。

$$(k_{11},k_{12})=(0.6,0.3),(k_{21},k_{22})=(-0.6,0.4),(k_{31},k_{32})=(7.0,-5.0)$$
$$(k_{41},k_{42})=(-10.0,-11.0),(k_{51},k_{52})=(10.0,11.0),(k_{61},k_{62})=(0.01,10.0)$$

また，エージェントの初期値を以下に示す。

$$(x_{11},x_{12})=(6.0,0.0),(x_{21},x_{22})=(3.0,3\sqrt{3}),(x_{31},x_{32})=(-3.0,3\sqrt{3})$$
$$(x_{41},x_{42})=(-6.0,0.0),(x_{51},x_{52})=(-3.0,-3\sqrt{3}),(x_{61},x_{62})=(3.0,-3\sqrt{3})$$

(6.6)式より，合意値 $\boldsymbol{\alpha}$ は全てのエージェントの平均座標である(0,0)となる。MAS 全体の合意誤差 x^e が 0.4 以下になるとき，合意を成功とする。この基準は，エージェントの初期値により初期の合意誤差 x^e が約 45 となり，その約 0.1%以下であることや，制御を続けてもこの値を超えないことから，この値を合意成功の閾値とした。また，エージェント同時の情報通信を表すグラフの隣接行列 A は次のように表され，ネットワーク構造は Fig.6.2 に示す無向グラフである。

$$A=\begin{bmatrix}0.0 & 0.1 & 0.0 & 0.0 & 0.0 & 0.6\\ 0.1 & 0.0 & 0.2 & 0.0 & 0.0 & 0.0\\ 0.0 & 0.2 & 0.0 & 0.3 & 0.0 & 0.0\\ 0.0 & 0.0 & 0.3 & 0.0 & 0.4 & 0.0\\ 0.0 & 0.0 & 0.0 & 0.4 & 0.0 & 0.5\\ 0.6 & 0.0 & 0.0 & 0.0 & 0.5 & 0.0\end{bmatrix}$$

本シミュレーションのエージェント 6 体の通信は，グラフ理論的には，強連結で平衡で

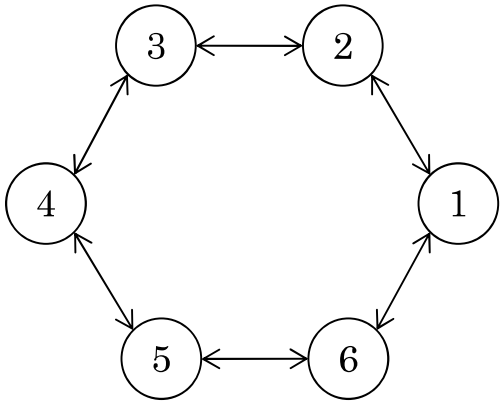

Fig.6.2.　Information communication network structure

ある。この場合，適切に制御されれば平均合意が達成可能である。

シミュレーションで使用した提案システムのパラメータは試行錯誤により，a=3，b_{th}=0.3，β_1=0.01，β_2=0.01，S_{th}=0.1，d_{th}=1.0，σ_i^{pre}=1.0，p_0^{pre}=0.01，p_1^{pre}=0.1，p_2^{pre}=0.15，η_w=0.1，η_c=0.01，η_σ=0.01，η_0=0.03，η_1=0.02，η_2=0.01，δ=0.3 とした。CP の初期ニューロン数は$m=a$=3 とし，そのニューロンのパラメータの初期値は(4.10)式により決定した。ただし，中心c_{ij}はc_{i0}=0.0，c_{i1}=5.0，c_{i2}=-5.0(i=1,2)とした。また，CP および，従来法(DRACS)の NN の入力変数I_z (z=1,2)は$I_z = x_{iz}$であり，従来法の NN のニューロン数は 36 個で固定である[78]。

6.5.2　シミュレーション結果

Fig.6.3 に提案システム(AFCPCS)の全エージェントの軌跡，Fig.6.4 は合意地点の拡大図，Fig.6.5 に従来システムの全エージェントの軌跡，Fig.6.6 は合意地点の拡大図である。なお，Fig.6.5,Fig.6.6 は，拡大個所は異なるが，x_1軸・x_2軸は同じ拡大率で示している。また，Fig.6.7 に両システムによる各エージェントの合意までの移動距離，Fig.6.8 に合意後の移動距離，Fig.6.9 に両システムの全エージェントと合意値$\boldsymbol{\alpha}$の距離誤差の推移，Fig.6.10 に MAS 全体の誤差の推移，Fig.6.11 に誤差範囲[0:0.4]の拡大図を示す。

Fig.6.3，Fig.6.5 より，両システムは合意値である(0,0)に近づき，合意しているが，両図を比べて提案システムの方が合意までの迂回が大きく，(0,0)近傍でも円を描いて合意している。それは，Fig.6.7 の合意までの移動距離を見ても，提案システムで制御した各エージェントが無駄な動きが含まれていることが分かる。しかし，合意値(0,0)に黒丸(●)を記している Fig.6.4，Fig.6.6 より，提案システムの方が合意値(0,0)付近で合意していることが分かる。それに対し DRACS は，6 体のエージェント内で合意はしているが，制御目的の合意値$\boldsymbol{\alpha}$から離れている。それらは，Fig.6.9 の MAS 全体の距離誤差を見ても，提案システムは 0 に収束し，DRACS は 5 に収束していることからも言える。以上のことから，提案システムは(19)式の合意と(20)式の平均合意のどちらも満たしている。それに対し，従来法は平均合意が達成可能な合意問題でありながら，単に合意が達成されたため，制御性能は提案システムの方が優れている。

また，Fig.6.6 より，DRACS の全エージェントが振動しているため，Fig.6.7 より，合意後の移動距離は，明らかに DRACS の方が大きい。提案システムは，各エージェントの合意後の移動距離が 0.2 以下であったことから，合意後，全エージェントは静止していると言える。また，Fig.6.10 より，合意は提案システムよりも DRACS の方が速い。さらに，Fig.6.11 より，およそ時刻 0.6 秒で合意成功とした MAS 全体の合意誤差 0.4 以下になったことからも DRACS の方が速いことからも言える。しかし，時刻 3 秒で提案システムが DRACS よりも MAS 全体の合意誤差が小さくなり，時刻 8 秒で 0 に収束している。以上の結果から，DRACS

は合意誤差までの速さと移動距離の点で優れているが，提案システムは合意値への収束性と合意後に各エージェントが静止する点で優れている。

提案システムのさらなる検証として，Fig.6.12 に提案システムによる各エージェントのニューロン数の推移，Fig.6.13 に提案システムによる各エージェントの x_1 の追従誤差の推移，Fig.6.14 に提案システムによる各エージェントの x_2 の追従誤差の推移，Fig.6.15 に提案システムによる各エージェントの x_1 の制御入力の推移，Fig.6.16 に提案システムによる各エージェントの x_2 の制御入力の推移，Fig.6.17 に提案システムによる各エージェントのシナプス荷重ベクトル $\mathbf{w}$ のノルムの推移，Fig.6.18 に提案システムによる各エージェントのガウシアン関数の中心ベクトル $\mathbf{c}$ のノルムの推移，Fig.6.19 に提案システムによる各エージェントのガウシアン関数の分散ベクトル $\boldsymbol{\sigma}$ のノルムの推移，Fig.6.20 に提案システムによる各エージェントのパラメトリック係数ベクトル $\mathbf{p}_0$ のノルムの推移，Fig.6.21 に提案システムによる各エージェントのパラメトリック係数ベクトル $\mathbf{p}_1$ のノルムの推移，Fig.6.22 に提案システムによる各エージェントのパラメトリック係数ベクトル $\mathbf{p}_2$ のノルムの推移を示す。

Fig.6.12 より，提案システムの各エージェントに搭載した CP のニューロン数が最大 8~9 個になり，最終的には 5 個で制御を可能にしたことが分かる。従来法である DRACS が各エージェントに用いたニューロン数 36 個と比べ，大きく減らす事が出来き，提案システムはニューロン数の初期設定なしで適切な設計が出来るとと言える。また，Fig.6.13,5.14 より，各エージェントの x_1, x_2 の誤差が時刻約 3 秒で 0 になり，Fig.6.15,5.16 より，各エージェントの x_1, x_2 への制御入力もその時刻で 0 になったことから，時刻約 3 秒で制御が成功し，各エージェントが静止したことが言える。最後に，Fig.6.17 のノルムの推移より，およそ 2.5 秒(図中の矢印)でノルムが収束しているため，速い学習が実現でき，環境適応が速かったと言える。それが，提案法は従来法である DRACS との比較において，合意値での正確な合意に繋がったと考えられる。そして，Fig.6.18~Fig.6.22 より，ガウシアン関数の中心ベクトル $\mathbf{c}$ ・分散ベクトル $\boldsymbol{\sigma}$ ・パラメトリック式の係数ベクトル $\mathbf{p}_h (h = 0,1,2)$ のノルムも Fig.6.17 と同様の結果が得られた。

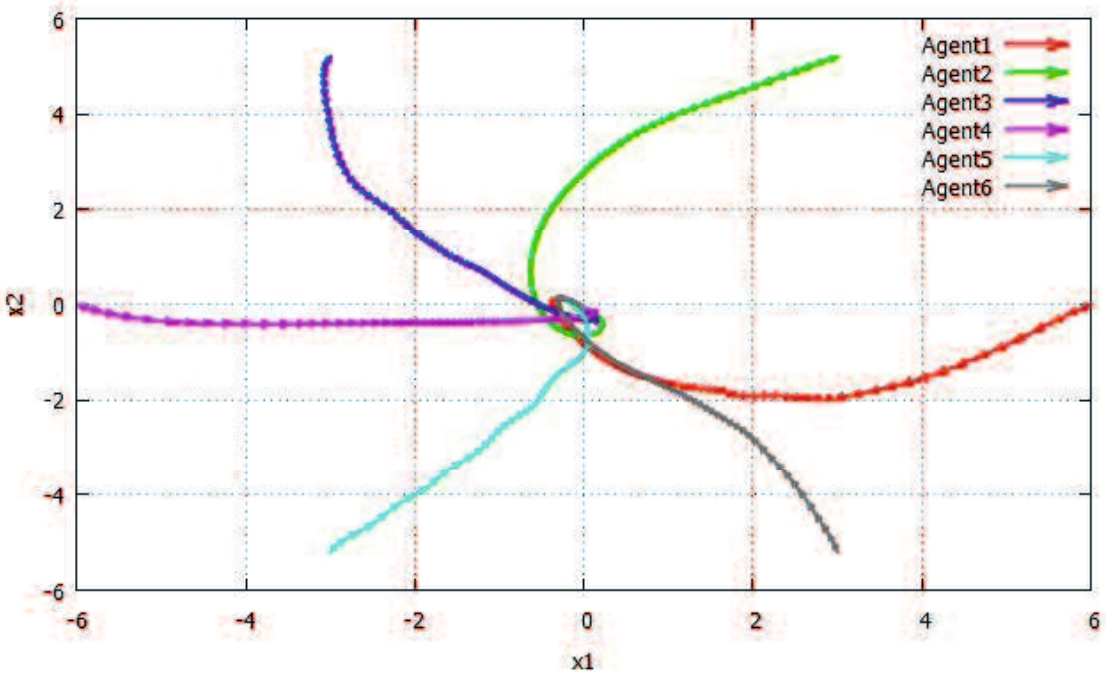

Fig.6.3. Trajectory of all agents by the proposed sytem

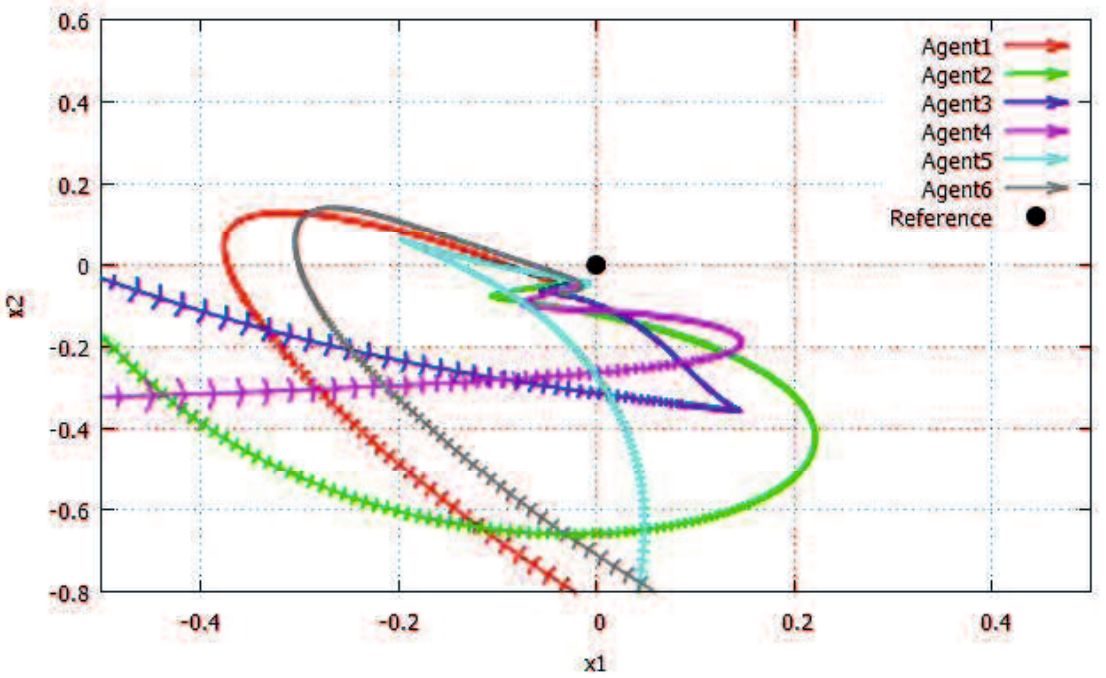

Fig.6.4. The enlarged view of Fig.6.3

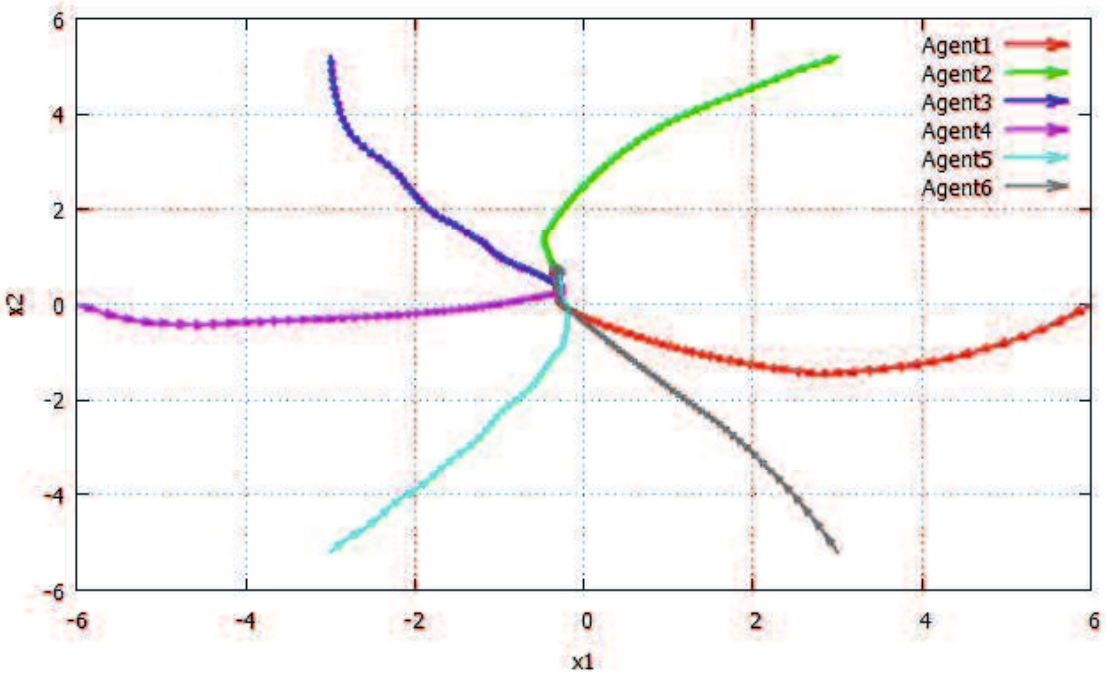

Fig.6.5. Trajectory of all agents by the conventional system

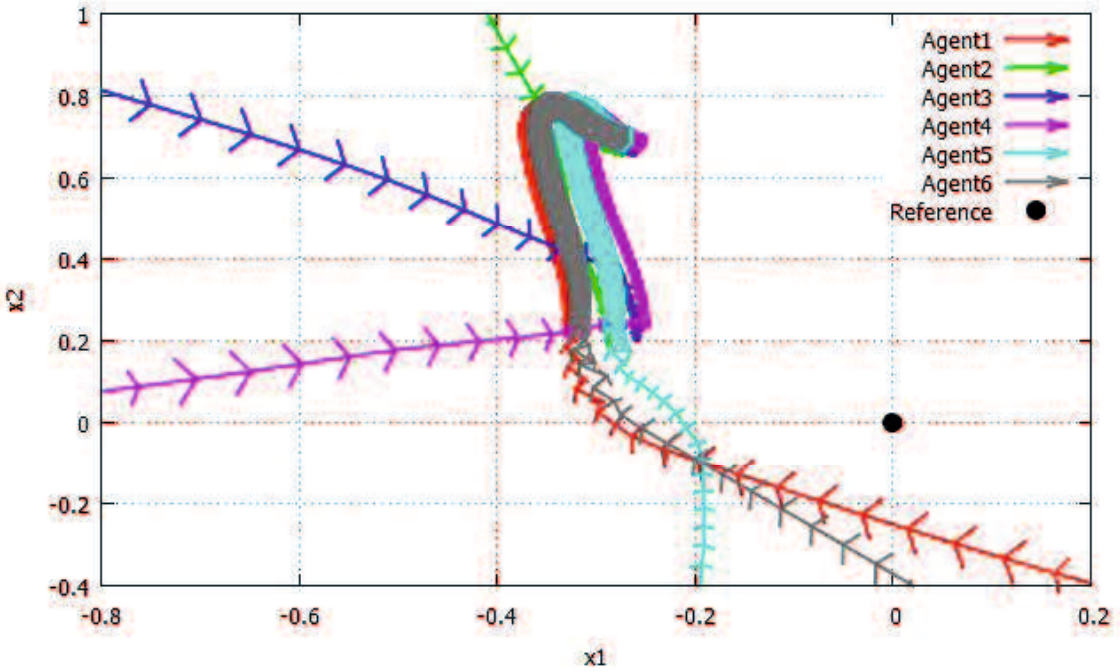

Fig.6.6. The enlarged view of Fig.6.5

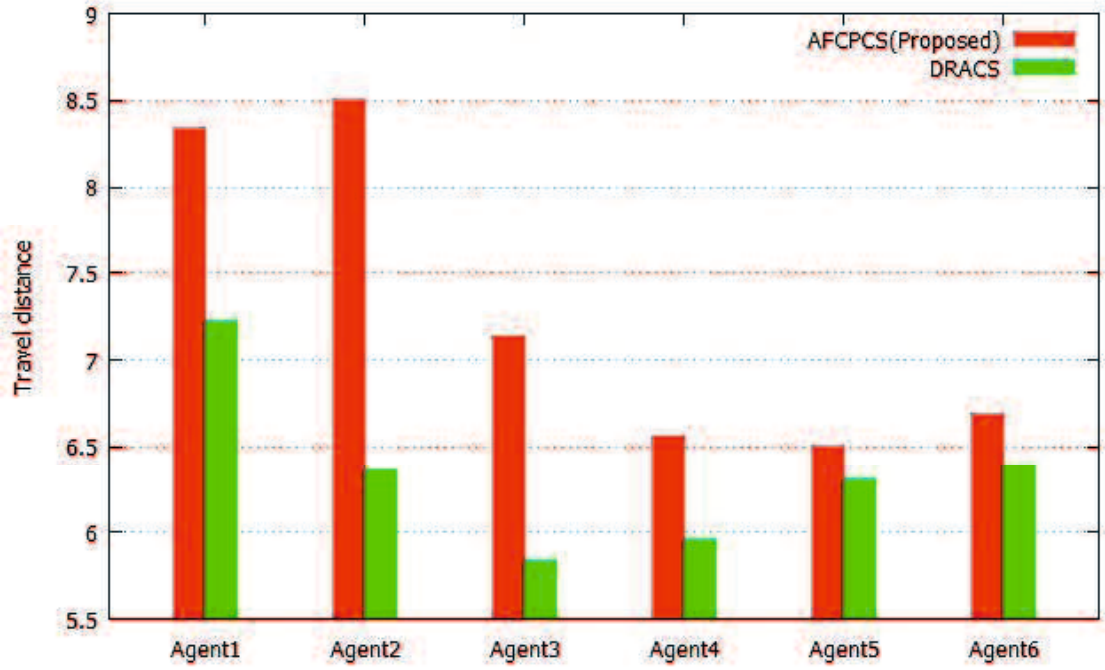

Fig.6.7. Comparison of moved distance of each agent until consensus by the proposed AFCPCS and conventional DRACS

Fig.6.8. Comparison of moved distance of each agent after consensus by the proposed AFCPCS and conventional DRACS

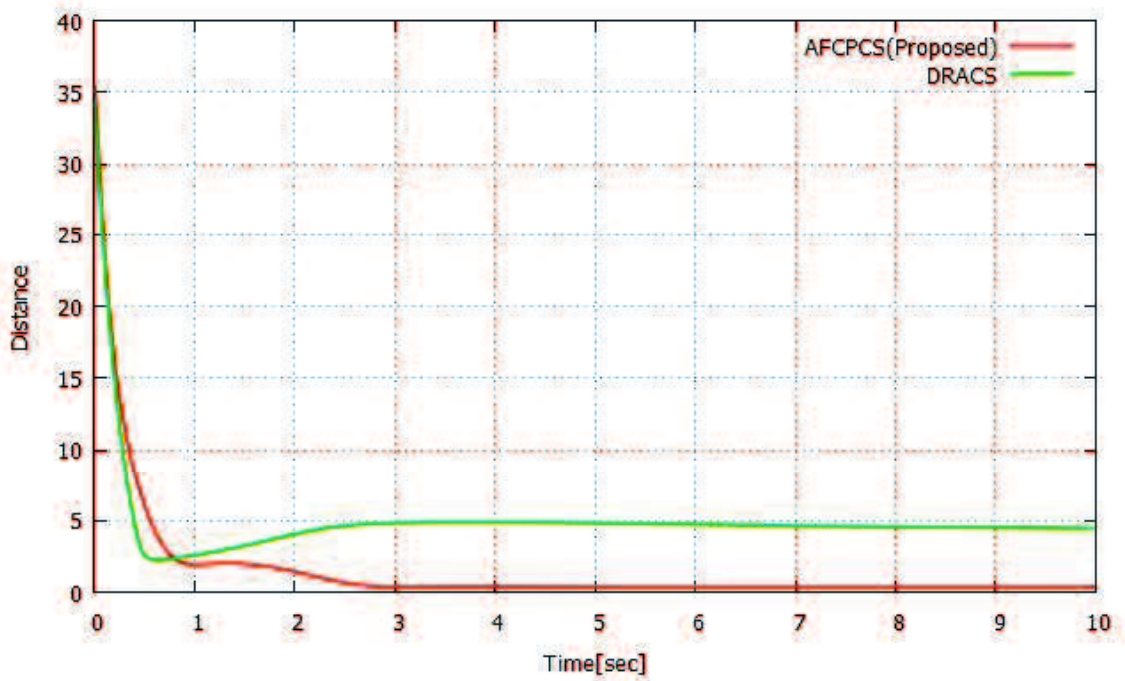

Fig.6.9. Comparison of the distance error of all agents by the proposed AFCPCS and conventional DRACS

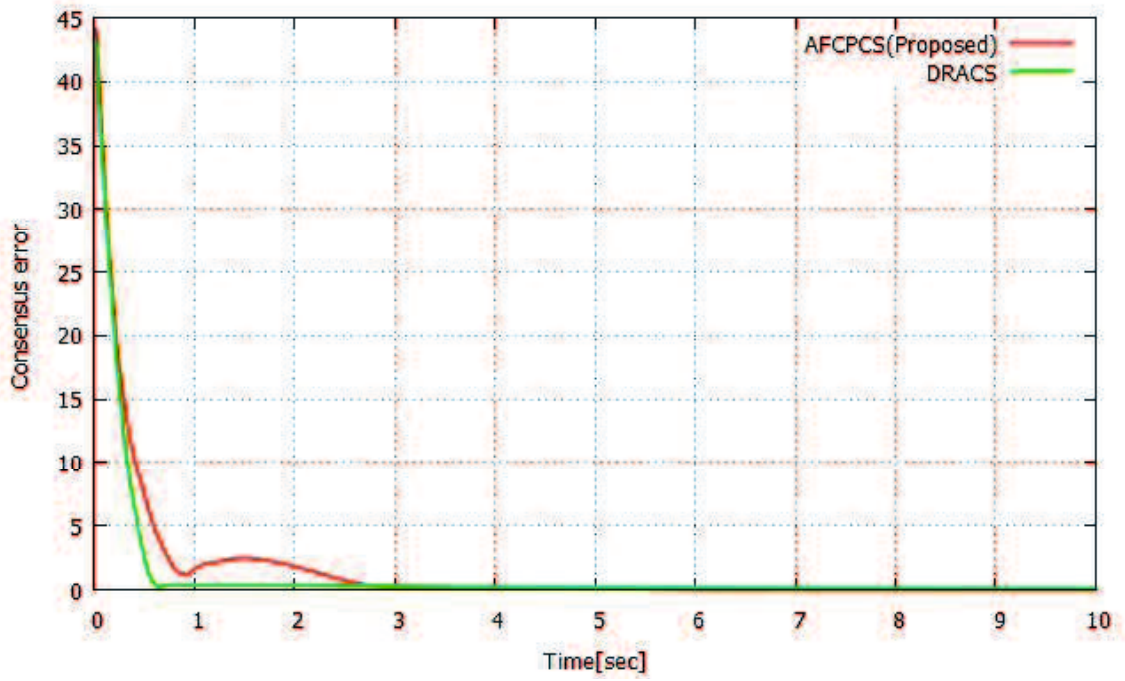

Fig.6.10. Comparison of consensus error of all agents by the proposed AFCPCS and conventional DRACS

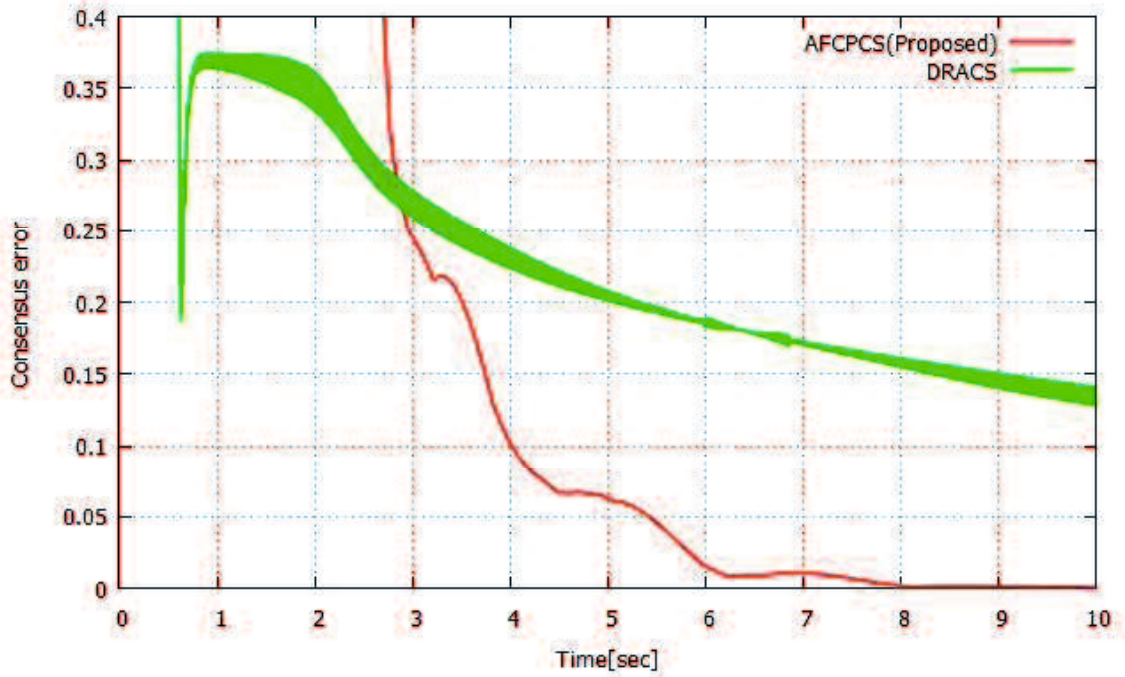

Fig.6.11. The enlarged view of Fig.6.10

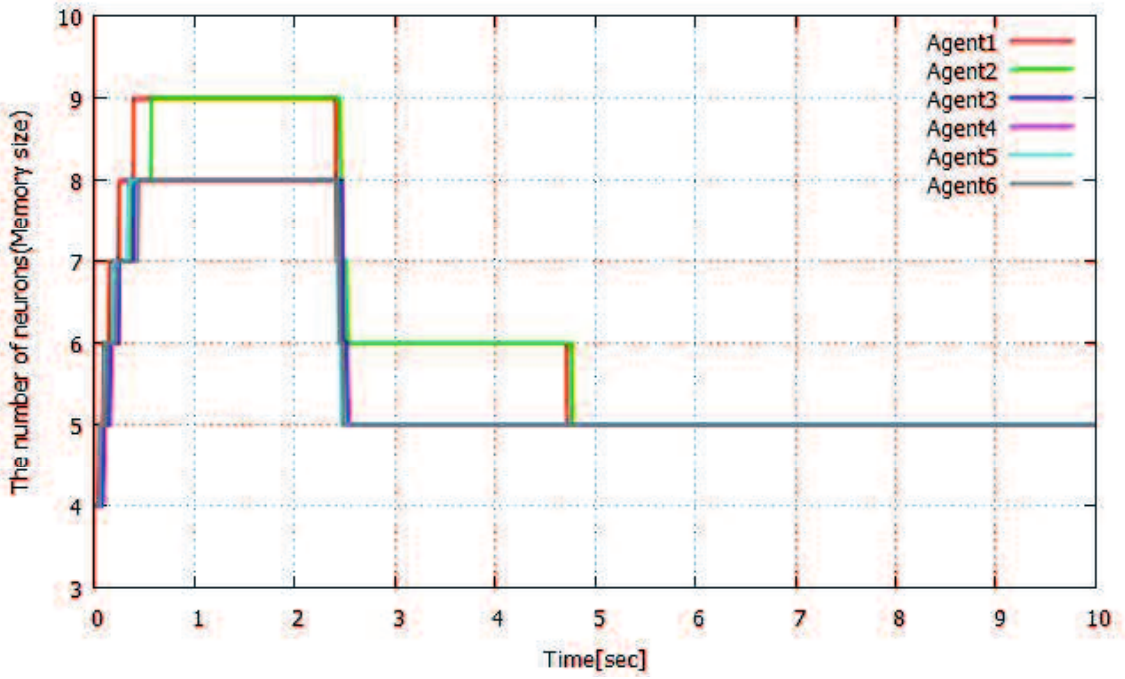

Fig.6.12. The number of neurons of each agent of the proposed system

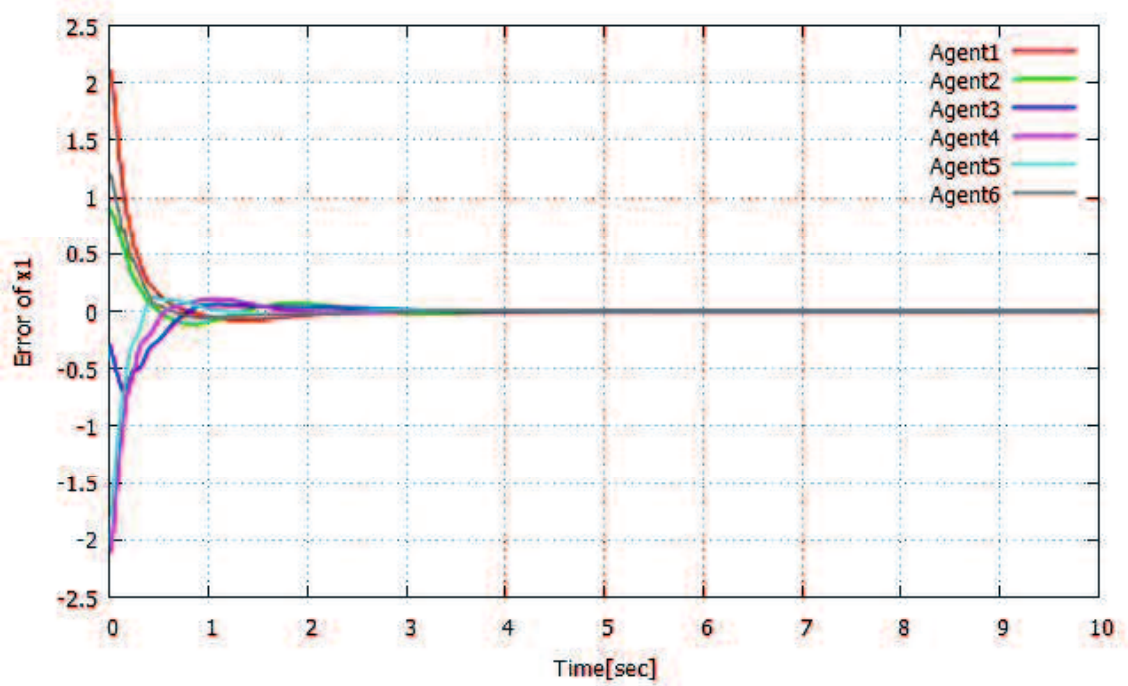

Fig.6.13. The error of x_1 of each agent of the proposed system

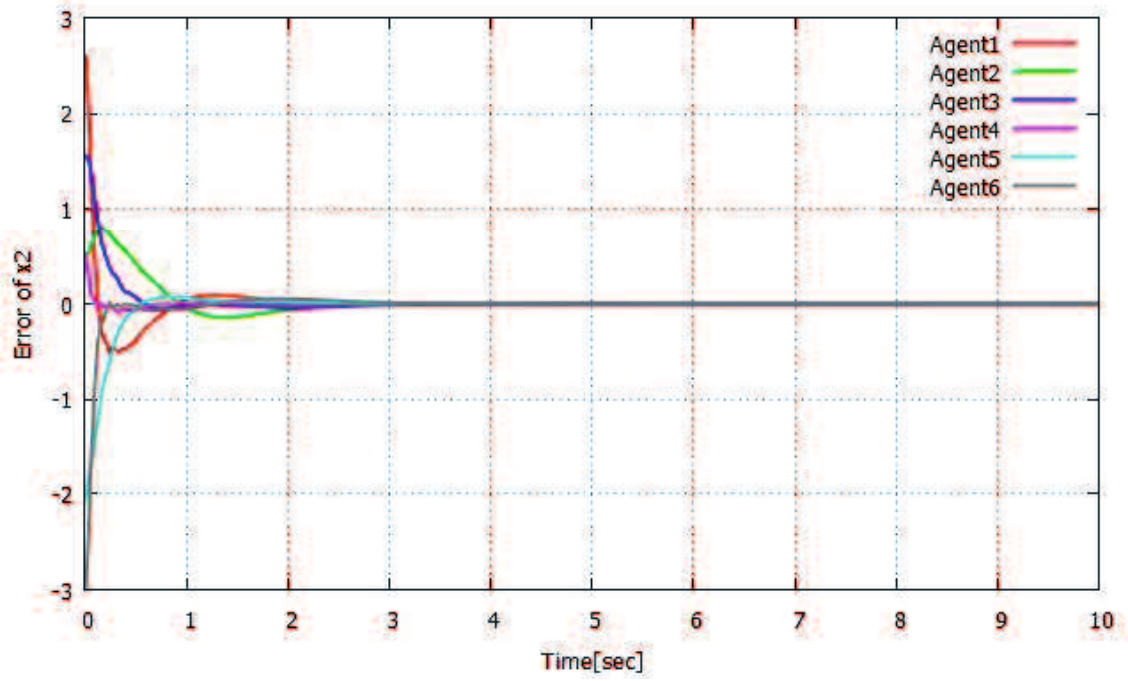

Fig.6.14. The error of x_2 of each agent of the proposed system

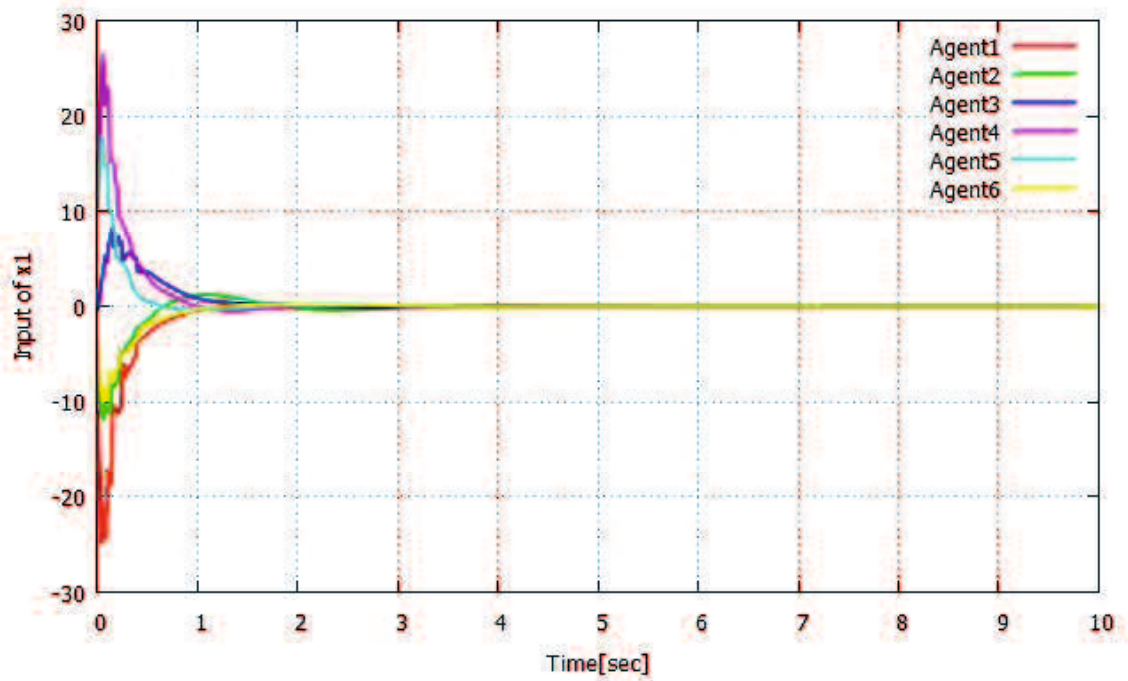

Fig.6.15. The input to x_1 of each agent of the proposed system

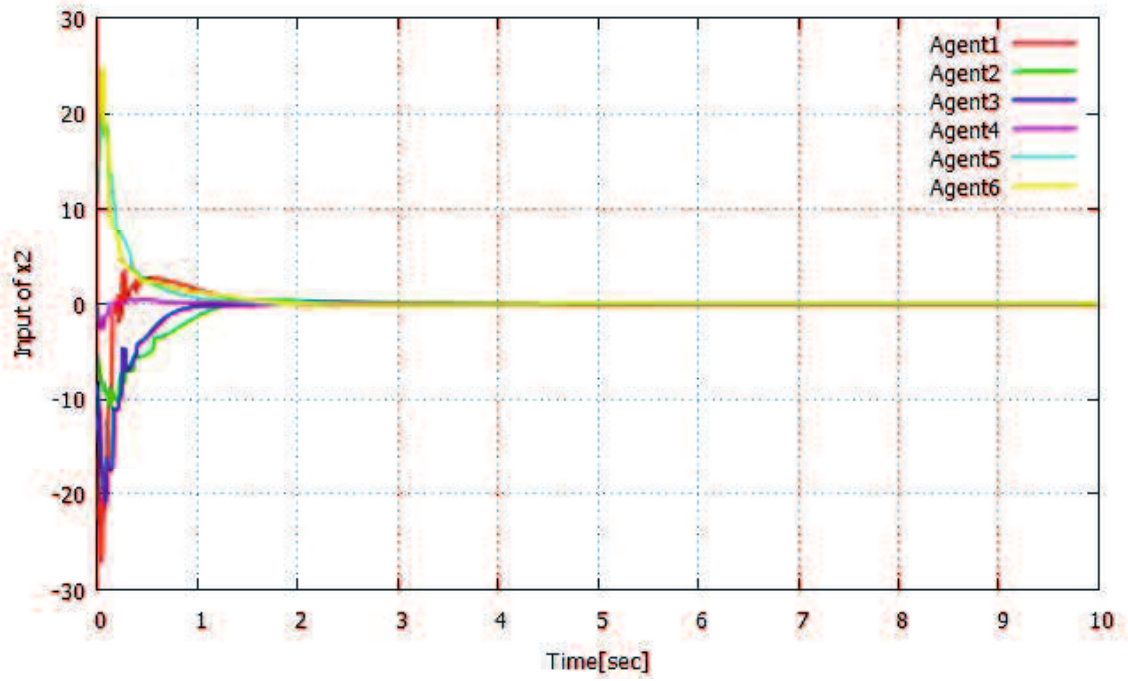

Fig.6.16. The input to x_2 of each agent of the proposed system

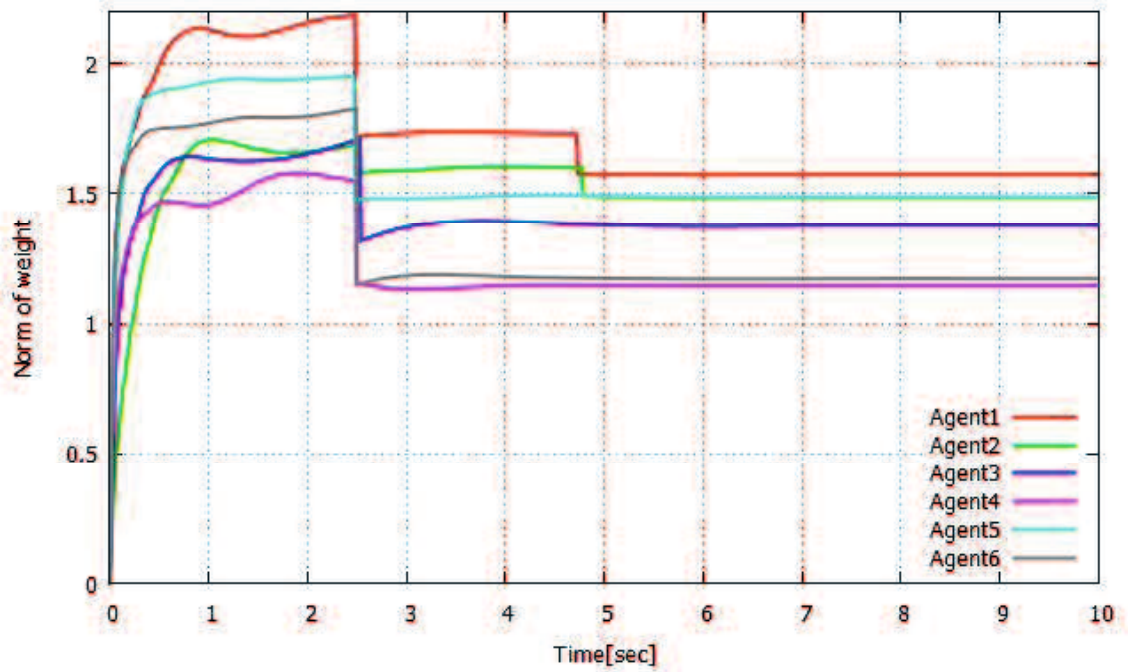

Fig.6.17. The norm of weights of each agent of the proposed system

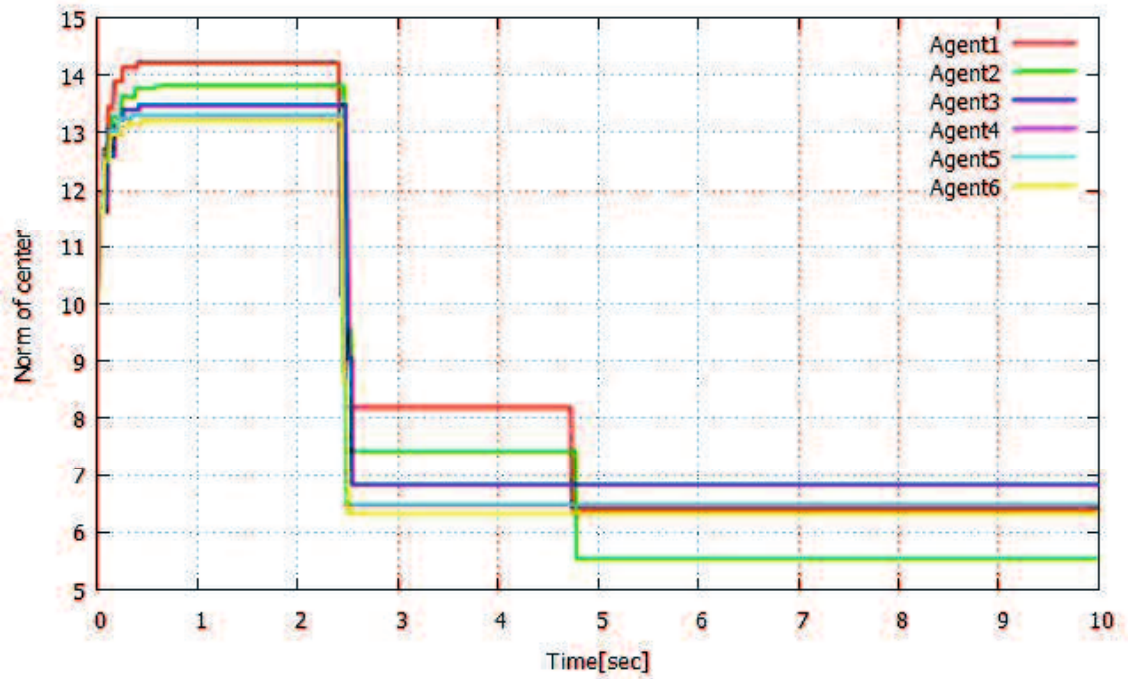

Fig.6.18. The norm of centers of each agent of the proposed system

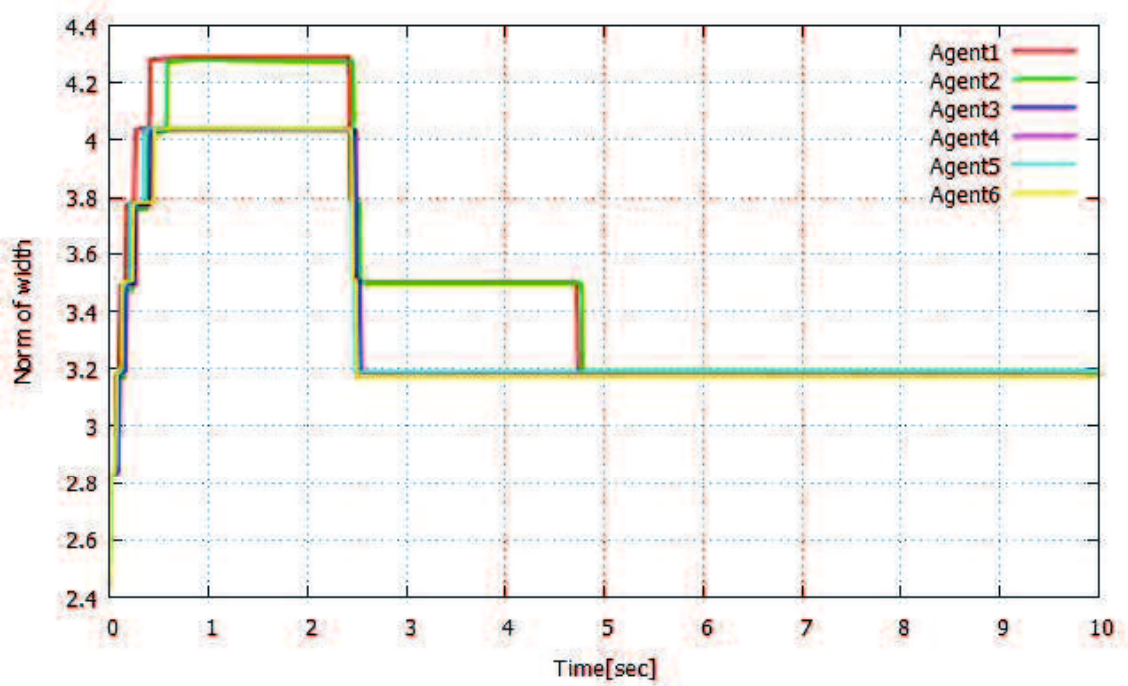

Fig.6.19. The norm of widths of each agent of the proposed system

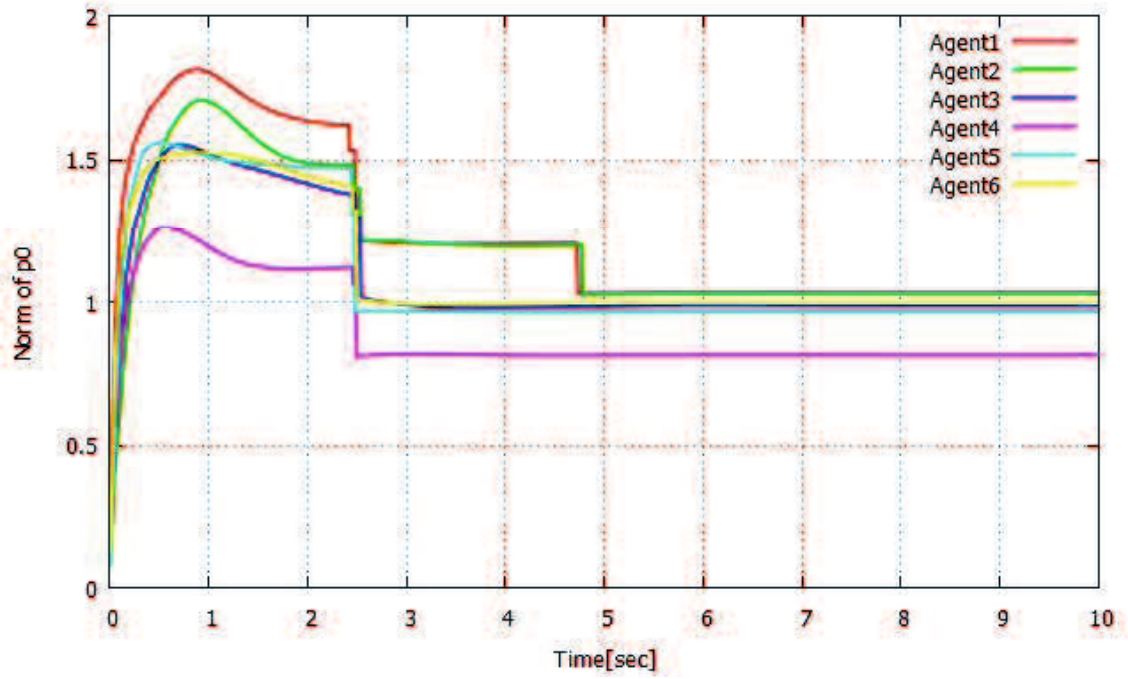

Fig.6.20. The norm of p_0 of each agent of the proposed system

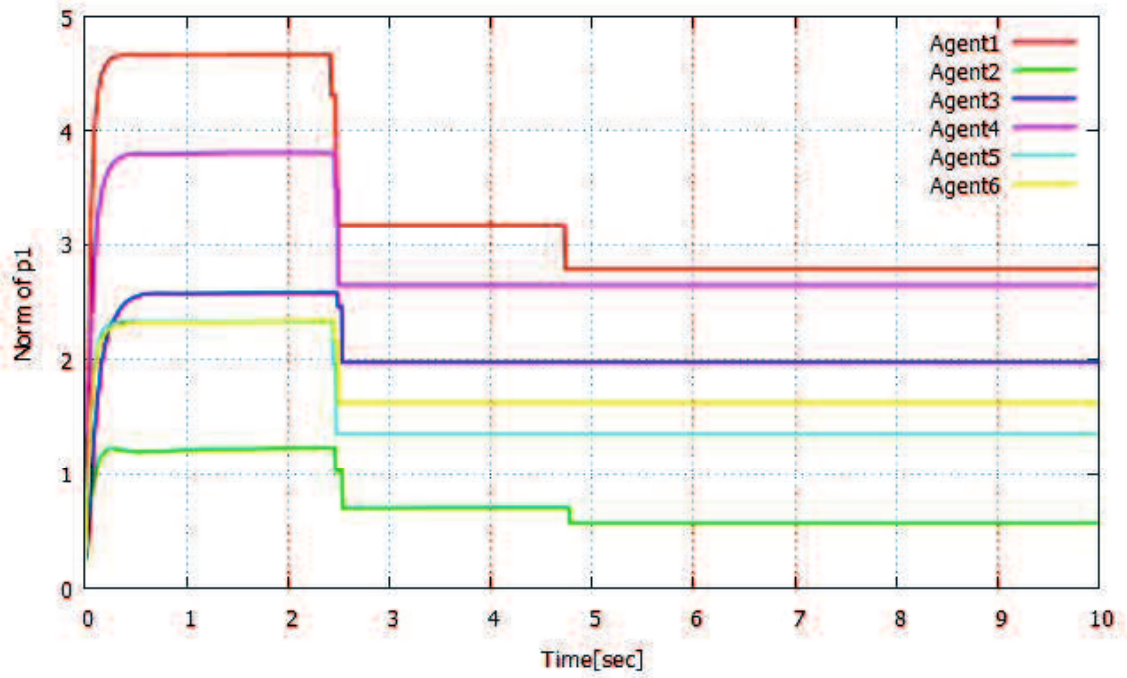

Fig.6.21. The norm of p_1 of each agent of the proposed system

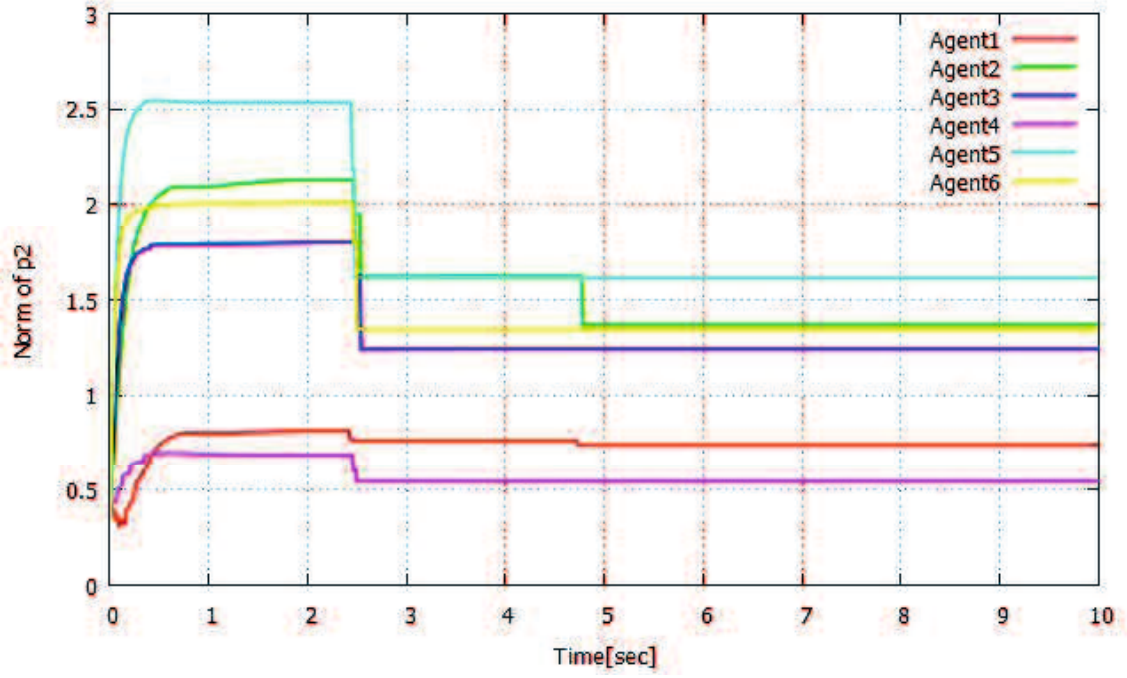

Fig.6.22. The norm of p_2 of each agent of the proposed system

6.6 まとめ

フィードバック誤差学習の設計に基づき，時間変動する目標信号の追従問題に対応した自己融合小脳パーセプトロンモデル利用型制御システムを提案し，マルチエージェントシステムの合意問題の計算機シミュレーションにより，その有効性を示した。

今後の展開として，合意問題の応用であるリーダー・フォロワーによる目標追従合意問題への適用が考えられる。

第 7 章　結論

本論文では，人間の脳の機能である学習や記憶に着目し，それらを模倣した制御システムを構築した。学習に着目し，ロバスト強化学習システムを提案した。また，記憶と小脳モデルである CMAC に注目し，新たな小脳パーセプトロン改良モデル(Cerebellar Perceptron improved model，CP)を提案し，それを基に制御システムを構築した。

第 2 章では，強化学習制御（システム）と適応 H_∞ 制御（システム）の協働により性質の異なる状態を同時に制御する方法を提案した。そして，台車の位置，振子の角度を考慮したクレーンによる計算機シミュレーションで，1 変数を対象とした制御系設計法では困難な，性質の異なる多様な状態遷移を計画的に行わせる，即ち，多様な計画行動をロバスト制御と強化学習の協働型制御システムで実現した。

第 3 章では，システムの不確定性に頑健なロバスト制御とモデルを必要としない制御系設計法である強化学習の両者を融合したオンライン強化学習である「リアルタイム強化学習制御システム(Real-time Reinforcement Learning Control System，RRLCS)」を提案する。リヤプノフ関数による安定性解析により，システムの安定性を証明し，台車付き倒立振子による計算機シミュレーションにより提案システムの有効性を示した。

第 4 章では，フィードバック制御において自己構造型の CP を提案し，それを基に構築した「小脳パーセプトロン改良モデル利用型ロバスト制御システム(Cerebellar Perceptron Robust Control System，CPRCS)」を提案した。リヤプノフ関数による安定性解析により，システムの安定性を証明し，台車付き倒立振子による計算機シミュレーションにより，最低限のニューロン(メモリ)数で高い追従性能を示した。

第 5 章では，CPRCS をフィードフォワード制御に拡張するため，自己融合型の CP を提案し，それを基に構築した「自己融合小脳パーセプトロン改良モデル利用型ロバスト制御システム(Auto-Fusion Cerebellar Perceptron Robust Control System，AFCPRCS)を提案した。フィードバック誤差学習による学習則により最適な CP を構築するシステムであり，台車付き倒立振子による計算機シミュレーションにより，定常状態における残差を減らすことに成功した。

第 6 章では，AFCPRCS をマルチエージェントシステム(Multi-Agent System，MAS)の合意問題に適用可能な制御システムに拡張した。多入力多出力(Multi Input Multi Output，MIMO)システムで構成される 6 体のエージェントによる合意問題の計算機シミュレーションにより，高い合意性能を示した。

各章の提案システムをまとめると，以下のようになる。

第 2 章　強化学習制御と適応 H_∞ 制御の協働型制御方式

- 多様な計画行動をするロバスト制御と強化学習の協働型制御システム
- 性質の異なる状態を同時に制御する方式

第 3 章　H_∞ 追従性能補償器を備えたリアルタイム強化学習制御システム

- オンライン強化学習制御システムの開発
- 有界性による確率推論である強化学習の安定性の保証

第 4 章　フィードバック制御における
小脳パーセプトロン改良モデル利用型ロバスト制御システム

- フィードバック制御において，リヤプノフ関数に基づく安定性の証明
- 自己構造アルゴリズムにより最低限のニューロンで制御可能

第 5 章　フィードフォワード制御における
自己融合小脳パーセプトロン改良モデル利用型ロバスト制御システム

- フィードバック誤差学習と自己融合アルゴリズムによるフィードフォワード制御への適用
- 定常状態におけるフィードバック制御以上の定常偏差の減少

第 6 章　小脳パーセプトロン改良モデルの合意問題への適用

- MAS 合意問題における高い合意性能
- 合意問題や MIMO システムへの適用可能による CP の汎用性

以上，本論文では人間の小脳の機能を模倣した CMAC に着目し，離散的な空間ではなく連続的な空間に対応可能にした。ここで，提案してきた各制御システムは，CP をロバスト制御や適応制御と結びつけたものである。制御対象の状況に応じて，制御に必要なニューロンのみを連結して結合荷重を強化させるシステムは，ニューロン数の減少だけでなく計算処理時間の減少にもつながるため，非常に有用なシステムである。また，単入力単出力(Single Input Single Output，SISO)システムや MIMO システムなどの制御対象や，追従問題や合意問題などの制御目的に対しても制御可能なことからその汎用性も期待できる。本論文では，各提案システムのニューロンの連結数を一定にしていたが，制御対象の状態に合わせて連結数を適応的に変えていく手法により，さらなるニューロン数や計算処理時間の減少に繋がると考えられる。また，全体を通じて計算機シミュレーションにより各種の実験を行ったが，実環境への拡張も今後の課題である。

人間の脳は，未だ解明されていないメカニズムが多く存在する。それらが解明されれば，より高度な学習・記憶の機能を持つ小脳モデルを構築でき，制御工学においてさらなる発展が期待できる。そのため，制御工学における小脳モデルの研究は，研究課題として非常に重要で興味深く，長期的な課題ではあるが今後の工学的応用が期待されている。

謝辞

本論文をまとめるにあたり，本学教授 大林正直先生には終止懇切なるご指導と様々な有益なるご助言および温かいご激励を賜りました。この 5 年間の研究生活は，大林先生の公私に渡る温かなご指導が無ければ成り立ちませんでした。ここに心より感謝の意を表すと共に謹んでお礼申し上げます。

そして本論文をまとめるにあたり，本学教授 田中幹也先生，准教授 田村慶信先生，准教授 松藤信哉先生，准教授 松元隆博先生，准教授 山口真悟先生，准教授 若佐裕治先生，山口大学理工学研究科キャリアパス形成支援室 浜田純夫先生には，有益な御教授および温かいご激励を賜り，深く謝意を表します。

また，日頃から様々なご助言とご討論を賜りました本学助教授 呉本尭先生，助教授 間普真吾先生，愛知県立大学情報科学部情報科学科准教授 小林邦和先生に深く感謝いたします。そして，ご協力頂きました本学工学部知能情報工学科生体情報システム工学講座の学部生，大学院生ならびに卒業生の関係諸氏に心よりお礼申し上げます。

最後に，私を産み，育て，そして精神面，肉体面，社会面，その他のすべてにおいて私を支えてくださった両親，祖父母，妹に感謝いたします。

参考文献

NN に関する参考文献

[1] F. Rosenblatt: Principles of Neurodynamics, Spartan (1961)

[2] C.T. Lin, C.P. Jou, and C.J. Lin: GA-based Reinforcement Learning for Neural Networks, International Journal of Systems Science, Vol. 29, No.3, pp.233-247 (1988)

[3] E.D. Karnin: A simple procedure forpruning back-propagation trained neural net-works, IEEE Trans. Neural Networks., Vol.1, pp.239-242 (1990)

[4] J.J. Hopfield: Neural Networks and Physical Systems with Emergent Collective Computational Properties, Proc. of the National Academy of Sciences U.S.A, Vol.81, pp.2554-2558 (1982)

[5] G.E. Hinton and R.R. Salakhutdinov: Reducing the Dimensionality of Data with Neural Networks, Science 313, pp.504-507 (2006)

ファジィに関する参考文献

[6] S. Cavalieri and M. Russo: Improving Hopfield Neural Network Performance by Fuzzy Logic-based Coefficient Tuning, Neurocomputing, Vol.18, pp.107-126 (1998)

[7] M. Obayashi, T. Kuremoto, and K. Kobayashi : A Self-Oraganized Fuzzy-Neuro Reinforcement Learning System for Continuous State Space for autonomous Robots, Proc. of International Conference on Computational Intelligence for Modeling, Control and Automation (CIMCA), pp.551-556 (2008)

強化学習に関する参考文献

[8] A.G. Barto, R.S. Sutton and C.W. Anderson: Neuronlike adaptive elements that can solve difficult learning control problems, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 13, pp.834-846 (1983)

[9] Y.H. Kuo, J.P. Hsu and C.W. Wang: A Parallel Fuzzy Inference Model with Distributed Prediction Scheme for Reinforcement Learning, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 22, No. 2, pp.160-172 (1998)

[10] C.T. Lin, C.P. Jou, and C.J. Lin: GA-based Reinforcement Learning for Neural Networks, Intertnational Journal of Systems Science, Vol. 29, No.3, pp.233-247 (1998)

[11] J. Lee, S. Oh, and D. Choi: TD Based Reinforcement Learning Using Neural Networks in Control Problems with Continuous Action Space, IJCMM'98,

pp.2028-2033 (1988)
[12] K. Umesako, M. Obayashi, and K. Kobayashi: Reinforcement Learning for Dynamic System in Incomplete Observation Environment, CDROM, ICONIP'00 (2000)

現代制御に関する参考文献

[13] K.S. Narendra, A.M.Annaswamy：Stable Adaptive System, Prentice Hall (1985)
[14] M.W. Spong and M. Vidyasagar：Robot Dynamics and Control, John Wiley & Sons
[15] K.Zhou, J.C.Doyle and K.Glover：Robust and Optimal Control, Prentice Hall (1995)
[16] H.Kimura：Chain-Scatterting Approach to H_∞ Control, Birkhauser (1996)

適応ファジィ制御に関する参考文献

[17] Y.-C. Hsueh and S.-F. Su: Compensate Controller Design for Solving The Parameter Drift Problem of Learning Fuzzy Control Systems, IEEE International Conference on Fuzzy Systems, pp.1112-1117 (2008)
[18] B.-S. Chen, C.-H. Lee, Y.-C. Chang: H_∞ Tracking Design of Uncertain Nonlinear SISO Systems: Adaptive Fuzzy Aproach, IEEE TRANSACTIONS FUZZY SYSTEMS Vol.4, No.1, (1996)
[19] Y.-S. Yang and X.-F. Wan: Adaptive H_∞ tracking control for a class of uncertain nonlinear systems using radial basis function neural networks, Neurocomputing, Vol.70, pp.932-941 (2007)
[20] S. Tong, H.-X. Li, and W. Wang: Observer-based adaptive fuzzy control for SISO nonlinear systems, Fuzzy Sets and Systems 148, pp.355-376 (2004)
[21] T.-C. Lin, C.H. Wang, and H.-L. Liu: Observer-based indirect adaptive fuzzy-neural tracking control for nonlinear SISO systems using VSS and H_∞ approach, Fuzzy Sets and System, Vol.143, pp.211-232 (2004)
[22] C.-C. Kung and T.-H. Chen: Observer-based indirect adaptive fuzzy sliding mode control with state variable filters for unknown nonlinear dynamical systems, Fuzzy Sets and Systems, Vol.155, pp.292-308 (2005)
[23] P. A. Phan and T. J. Gale: Direct adaptive fuzzy control with a self-structuring algorithm, Fuzzy Sets and Systems, Vol.159, pp.871-899, (2005)
[24] C.-K. Lin: Robust adaptive critic control of nonlinear systems using fuzzy basis function networks: An LMI approach, Information Sciences, Vol.177, pp.4934-4946 (2007)
[25] W.-Y. Wang, M.-L. Chan, C.-C. J. Hsu, and T.-T. Lee: H_∞ Tracking-Based Sliding Mode Control for Uncertain Nonlinear Systems via an Adaptive Fuzzy-Neural Approach, IEEE Transaction on System, Man and Cybernetics, pp.483-492 (2002)

[26] Q. Kang, W. Wang: Adaptive fuzzy controller design for a class of uncertain nonlinear MIMO systems, Nonlinear Dyn Vol.59, pp.579-591 (2010)
[27] T.-C. Lin, M. Roopaei: Based on interval type-2 adaptive fuzzy H_∞ tracking controller for SISO time-delay nonlinear systems, Commun Nonlinear Sci Numer Simulat 15, pp.4065-4075 (2010)
[28] L. Yu, S. Fei, and X. Li: Robust adaptive neural tracking control for a class of switched affine nonlinear systems, Neurocomputing 73, pp.2274-2279 (2010)
[29] Kuo-Ho Su, Minh-Hoang To, Chan-Yun Yang: Robust Tracking Controller Design and Its Application to Wheeled Robot, IEEE International Conference on System Science and Engineering, pp263-268 (2013)
[30] Xiru Wu, Yaonan Wang, Xuanju Dang: Robust adaptive sliding-mode control of condenser-cleaning manipulator using fuzzy wavelet neural network, Fuzzy Sets and Systems 235, pp.62-82 (2014)

自己構造ファジィニューラルネットワークに関する参考文献

[31] C.-F. Hsu, P.-Z. Lin, T.-T. Lee, and C.-H. Wang: Adaptive asymmetric fuzzy neural network controller design via network structuring adaptation, Fuzzy Sets and Systems 159, pp.2627-2649 (2008)
[32] K.-H. Cheng: Auto-structuring fuzzy neural system for intelligent control, Journal of The Franklin Institute 346, pp.267-288 (2009)
[33] R.-J. Wai, C.-M. Liu, and Y.-W. Lin: Robust path tracking control of mobile robot via dynamic petri recurrent fuzzy neural network, Soft Comput. 15 (4), pp.743-767 (2011)
[34] D. Lin and X. Wang: Self-organizing adaptive fuzzy neural control for the synchronization of uncertain chaotic systems with random-varying parameters, Neurocomputing 74, pp.2241-2249 (2011)
[35] Chun-Fei Hsu: Intelligent control of chaotic systems via self-organizing Hermite-polynomial-based neural network, Neurocomputing 123, pp.197-206 (2014)

ロバスト強化学習に関する参考文献

[36] R. M. Kretchmar, P. M. Young, C. W. Anderson, D. C. Hittle, M. L. Anderson, and C. C. Delnero: Robust Reinforcement Learning Control with Static and Dynamic Stability, Technical Report CS-00-102 (2000)
[37] J. Morimoto and K. Doya: Robust Reinforcement Learning, Neural Computation 17, pp.335-359 (2005)

[38] C. W. Anderson, P. M. Young, M. R. Buehner, J. N. Knight, K. A. Bush, and D. C. Hittle: Robust Reinforcement Learning Control Using Integral Quadratic Constraints for Recurrent Neural Networks, IEEE Transactions on Neural Networks, Vol.18, No.4 (2007)

[39] C.-K. Lin: H_∞ reinforcement learning control of robot manipulators using fuzzy wavelet networks, FUZZY sets and systems 169, pp.1765-1785 (2009)

[40] 内山祥吾，大林正直，呉本尭，小林邦和：H_∞追従性能補償器を備えたオンライン型強化学習制御システム，平成 23 年度(第 62 回)電気・情報関連学会中国支部連合大会，pp.373-374 (2011)

[41] S. Uchiyama, M. Obayashi, T. Kuremoto, and K. Kobayashi: Robust Reinforcement Learning Control System with H_∞ Tracking Performance Compensator, International Conference on Control, Automation and Systems, pp.248-253 (2011)

[42] S. Uchiyama, M. Obayashi, T. Kuremoto, and K. Kobayashi: H_∞ Robust Reinforcement Learning Control System with Auto-Structuring Fuzzy Neural Network, Proceedings of the 3rd International Symposium on Digital Manufacturing, pp.95-100 (2011)

[43] 内山祥吾，大林正直，呉本尭，小林邦和：H_∞追従性能補償器を備えたリアルタイム強化学習制御システム，電学論 C，Vol.132，No.6，pp.1008-1015 (2012)

オフライン学習に関する参考文献

[44] 大林正直，内山祥吾，呉本尭，小林邦和：強化学習制御と適応 H_∞制御の協働型制御方式，電学論 C，Vol.131，No.8，pp.1467-1474 (2011)

[45] 内山祥吾，大林正直，呉本尭，小林邦和：H_∞制御と強化学習の融合によるロバストな計画行動制御方式の開発，平成 22 年電気学会電子・情報・システム部門大会，pp.1518-1523 (2010)

[46] W. Zuo and L. Cai: A New Iterative Learning Controller Using Variable Structure Fouier Neural Network, IEEE Transaction on System, Man and Cybernetics, Part B: Cybernetics, Vol.40, No.2, pp.458-468 (2010)

[47] Chiang-Ju Chien: A Combined Adaptive Law for Fuzzy Iterative Learning Control of Nonlinear Systems with Varying Control, Expert Systems with Applications, Vol.37, pp.545-558 (2009)

[48] R. Syam, K. Watanabe, K. Izumi: Adaptive actor-critic learning for the control of mobile robots by applying predictive models, Soft Comput Vol.9, pp.835-845 (2005)

[49] D. Vrabie, F. Lewis: Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems, Neural Networks Vol. 22, pp.237-246 (2009)

フィードバック誤差学習に関する参考文献

[50] 川人光男：脳の計算理論，産業図書株式会社，(1996)

[51] K. Sabahi, M. Teshnehlab, and M.A. Shoorhedeli: Recurrent fuzzy neural network by using feedback error learning approaches for LFC in interconnected power system, Energy Conversion and Management 50, pp.938-946 (2009)

[52] F. Farivar, M.A. Shoorehdeli, M. Teshnehlab and M.A. Nekoui: Hybrid Control of Flexible Manipulator, Journal of Applied Sciences 9 (4), pp.639-650 (2009)

[53] A.V. Topalov, O. Kayak and G. Aydin: Neuro-adaptive sliding-mode tracking control of robot manipulators, INTERNATIONAL JOURNAL OF ADAPTIVE CONTROL AND SIGNAL PROCESSING, (2007)

CMAC

[54] J.S. Albus：New approach to manipulator control: The cerebellar model articulation (CMAC), Journal of Dynamic Systems, Measurement and Control, Transaction of the ASME 97, Ser G (3), pp.228-233 (1975)

CMAC：非線形に関する参考文献

[55] P.E.M. Almeida and M.G. Simoes: Parametric CMAC Networks: Fundamentals and Applications of a Fast Convergence Neural Structure, IEEE Transactions On Industry Applications, Vol.39, No.5 (2003)

CMAC：カオスシステムに関する参考文献

[56] H.-C. Lu and C.-Y. Chuang: Robust parametric CMAC with self-generating design for uncertain nonlinear systems, Neurocomputing 74, pp.549-562 (2011)

[57] C.-M. Lin and C.-H. Chen: CMAC-based supervisory control for nonlinear chaotic systems, Chaos Solitons and Fractals 35, pp.40-58 (2008)

[58] C.-J. Lin and C.-Y. Lee: A novel parametric fuzzy CMAC network and its applications, Applied Soft Computing 9, pp.775-785, (2009)

[59] C.-F. Hsu, C.-M. Chung, C.-M. Lin, and C.-Y. Hsu: Adaptive CMAC neural control of chaotic systems with a PI-type learning algorithm, Expert Systems with Applications 36, pp.11836-11843 (2009)

[60] C.-F. Hsu: Adaptive dynamic CMAC neural control of nonlinear chaotic systems with L_2 tracking performance, Engineering Applications of Artifical Intelligence 25, pp.997-1008 (2012)

[61] C.-F. Hsu, C.-J. Chiu, and J.-Z. Tsai: Indirect adaptive self-organizing RBF neural controller design with a dynamical training approach, Expert Systems with

Applications 39, pp.564-573 (2012)
[62] C.-M. Lin and H.-Y. Li: Self-organizing adaptive wavelet CMAC backstepping control system design for nonlinear chaotic systems, Nonlinear Analysis: Real World Applications, (2012)
[63] Y.-F. Peng: Robust intelligent backstepping tracking control for uncertain non-linear chaotic systems using H_∞ control technique, Chaos, Solitons and Fractals 41, pp.2081-2096 (2009)

CMAC：画像処理に関する参考文献
[64] C.-J. Lin, J.-H. Lee, and C.-Y. Lee ： A novel hybrid learning algorithm for parametric fuzzy CMAC networks and its classification applications, Expert Systems with Applications 35, pp.1711-1720 (2008)
[65] H.C. Lu and T. Tao: The treatment of image boundary effects CMAC netwoks, Proceedings of the IEEE International Joint Conference on Neural Networks, July 25-29, vol.2, pp.867-872 (2004)

CMAC：パターン認識に関する参考文献
[66] H.T He and Y. Li: The research on flatness pattern recognition based on CMAC neural network, Proceedings of the International Conference on Machine Learning and Cybernetics, August 19-22, Vol.5, pp.2745-2748 (2007)

CMAC：船体運動に関する参考文献
[67] Z. Shen, C. Guo and Ning Zhang: A general fuzzified CMAC based reinforcement learning control for ship steering using recursive least-squares algorithm, Neurocomputing 73, pp.700-706 (2010)

CMAC：マニピュレータに関する参考文献
[68] S.I. Han, K.S. Lee, M.G. Park and J.M. Lee: Robust adaptive deadzone and friction compensation of robot manipulator using RWCMAC network, Journal of Mechanical Science and Technology 25 (6), pp.1583-1594 (2011)

CMAC：車輪倒立振子に関する参考文献
[69] C.-H. Chiu, Y.-F. Peng, and Y.-W. Lin: Robust intelligent backstepping tracking control for wheeled inverted pendulum, Springer Verlag (2011)

CMAC：モバイルロボットに関する参考文献

[70] J. Peng, Y. Wang, and W. Sun: Trajectory-Tracking Control for Mobile Robot Using Recurrent Fuzzy Cerebellar Model Articulation Controller, Neural Information Processing – Letters and Reviews, pp.15-23 (2007)

CMAC：プロセス制御に関する参考文献

[71] T. Yamamoto and M. Kaneda: Intelligent controller using CMACs self-organaized structure and its application for a process system, IEICE Trance. Fundamentals, Vol.E82-A, No.5, pp.856-860 (1999)

CMAC：PID チューニングに関する参考文献

[72] 黒住亮太，山本透：小脳演算モデルを用いたインテリジェント PID 制御系の一設計，電学論 C，Vol.125，No.4，pp.607-615 (2005)

CMAC：航空機の自動着陸システムに関する論文

[73] Jih-Gau Juang, Shuai-Ting Yu: Disturbance encountered landing system design based on sliding mode control with evolutionary computation and cerebellar model articulation controller, Applied Mathematical Modelling 39, pp.5862-5881 (2015)

PCMAC に関する参考文献

[74] P.C. Parks and J. Militzer: A Comparison of Five Algorithms for the Training of CMAC Memories for Learning Control Systems, Automatica, Vol.28, No.5, pp.1027-1035 (1992)

小脳パーセプトロン改良モデルに関する参考文献

[75] 内山祥吾，大林正直，呉本尭，小林邦和：小脳パーセプトロン利用型ロバスト制御システム，システム研究会，pp.1-6 (2011)

[76] 内山祥吾，大林正直，呉本尭，小林邦和：フィードバック誤差学習に基づく自己融合小脳パーセプトロンモデル利用型ロバスト制御システム，平成 24 年電気学会電子・情報・システム部門大会，pp.575-580 (2012)

[77] 内山祥吾，大林正直，呉本尭，小林邦和：小脳パーセプトロン改良モデル利用型ロバスト制御システム，電学論 C，Vol.133，No.6，pp.1251-1258 (2013)

マルチエージェント(グラフ理論) に関する参考文献

[78] Z.-G. Hou, L.Cheng, and M. Tan: Decentralized Robuts Adaptive Control for the Multiagent System Consensus Problem Using Neural Networks, IEEE

TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, PART B: CYBERNETICS, Vol.39, No.3 (2009)

[79] R.O. Saber, R.M. Murray: Agreement Problems in Networks with Directed Graphs and Switching Topoligy, IEEE Conference on Decision and Control, (2003)

[80] R.O. Saber, J.A. Fax, and R.M. Murray: Consensus and Cooperation in Networked Multi-Agent Systems, Proc. Of the IEEE, Vol.95, No.1, (2007)

[81] N.R. Jennings: Controlling cooperative problem solving in industrial multi-agent systems using joint intentions, Artificial Intelligence 75, pp.195-240 (1995)

[82] A. Das and F.L. Lewis: Cooperative adaptive control for synchronization of second – order systems with unknown nonlinearities, INTERNATIONAL JOURNAL OF ROBUST AND NONLINEAR CONTROL, (2010)

[83] A. Das and F.L. Lewis: Distributed adaptive control for synchronization of unknown nonlinear networked systems, Automatica 46, pp.2014-2021 (2010)

[84] W. Ren: Multi-vehicle consensus with a time-varying reference state, Systems & Control Letters, Vol.56, No.7-8, pp.474-483 (2007)

[85] W. Ren, R. W. Beard: Distributed Consensus in Multi-vehicle Cooperative Control Theory and Applications, Springer (2007)

[86] 内山祥吾，大林正直，呉本尭，小林邦和：小脳パーセプトロンモデル利用型ロバスト制御システムの提案とその合意問題への応用，第 21 回計測自動制御学会中国支部学術講演会，pp.92-93 (2012)

[87] 内山祥吾，大林正直，呉本尭，小林邦和，間普真吾：自己融合小脳パーセプトロン改良モデル利用型制御システムとその合意問題への適用，電学論 C，Vol.134，No.7，pp.990-998 (2014)

マルチエージェントシステム：自律ロボットの協調制御に関する参考文献

[88] M. Defoort, T. Floquet, A. Kokosy and W. Perruquetti: Sliding-Mode Formation Control for Cooperative Autonomous Mobile Robots, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, Vol.55, No.11, (2008)

リーダーフォロワーに関する参考文献

[89] L.Cheng, Z.-G. Hou, M. Tan, Y. Lin, W. Zhang: Neural-Network-Based Adaptive Leader-Following Control for Multiagent Systems With Uncertainties, IEEE Transactions on Neural Networks, Vol.21, No.8, pp.1351-1358 (2010)

[90] R. Vatankhah, S. Etemadi, A. Alasty, G. Vossoughi: Adaptive critic-based neuro-fuzzy controller in multi-agents: Distributed behavioral control and path tracking, Neurocomputing 88, pp.24-35 (2012)

[91] R. Cui, S.S. Ge, B.V.E. How, Y.S. Choo: Leader-follower formation control of underactuated autonomous underwater vehicles, Ocean Engineering 37, pp.1491-1502 (2010)

[92] Z. Meng, Z. Zhao, and Z. Lin: On global leader-following consensus of identical linear dynamic systems subject to actuator saturation, Systems & Control Letters 62, pp.132-142 (2013)

[93] Q. Song, J. Cao, W. Yu: Second-order leader-following consensus of nonlinear multi-agents systems via pinning control, Systems & Control Letters 59, pp.553-562 (2010)

[94] G. Wen, A. Rahmani, and Y. Yu: Consensus Tracking for Multi-Agent Systems with Nonlinear Dynamics under Fixed Communication Topologies, Proceedings of the World Congress on Engineering and Computer Science, (2011)

フォーメーション制御(水艇)に関する参考文献

[95] J. Ghommanm, N. Haddad and G. Poisson: Formation control for a class of output-feedback systems, 3rd National Conference on “Control Architectures of Robots”, (2008)

[96] M. Obayashi, Y. Yokoji, S. Uchiyama, L. Feng, T. Kuremoto, and K. Kobayashi: Intelligent Tracking Control Method of A Target by Group of Agents with Nonlinear Dynamics, International Conference on Control Automation and Systems, pp.928-933 (2011)