

ニューラルネットワークによる  
海中ロボット運動の表現と制御

1996年6月21日 提出  
指導教官 浦 環 教授

石 井 和 男

# 目次

図表一覧 .....	iii
第1章 序論 .....	1
1.1 はじめに .....	2
1.2 本研究の目的 .....	3
1.3 海中ロボットの動向 .....	4
1.4 海洋工学へのニューラルネットワーク適用 .....	9
1.5 本論文の内容 .....	10
第2章 対象とする海中ロボット .....	16
2.1 はじめに .....	17
2.2 航行型海中ロボット「PW45」 .....	17
2.3 汎用テストベッドロボット「Twin-Burger」 .....	18
2.3.1 諸元 .....	18
2.3.2 センサ .....	18
2.3.3 通信システム .....	20
2.3.4 アクチュエータ .....	20
2.3.5 コンピュータシステム .....	21
2.3.5 2号機の改良点 .....	23
2.3.6 ソフトウェア構造 .....	24
2.4 考察 .....	25
第3章 ニューラルネットワークによる海中ロボット運動の表現 .....	37
3.1 はじめに .....	38
3.2 ニューラルネットワークの基礎 .....	39
3.2.1 特徴 .....	39
3.2.2 ネットワークモデルの基本的記述 .....	39
3.2.4 誤差逆伝播学習法 .....	40
3.2.4 動的システムのモデル .....	42
3.3 運動の表現 .....	44
3.3.1 動的システムの同定 .....	44
3.3.2 学習手順 .....	45
3.4 海中ロボットへの適用 .....	47
3.4.1 PW45の縦運動への適用 .....	47
3.4.2 Twin-Burgerへの適用 .....	50
3.4.3 R-oneの海面におけるYaw運動への適用 .....	52
3.5 考察 .....	53



第4章 ニューラルネットワークを用いた運動モデルの信頼性評価	73
4.1 はじめに	74
4.2 随伴ネットワークを用いた信頼性評価	74
4.2.1 学習領域	74
4.2.1 随伴ネットワーク	74
4.2.3 随伴ネットワークの性質	75
4.2.4 随伴ネットワークによる動的モデルの信頼性評価	77
4.3 ネットワーク構造の最適化	78
4.3.1 中間層の出力による構造最適化	78
4.3.2 同定モデルへの適用	78
4.4 考察	79
第5章 ニューラルネットワークによる運動制御	93
5.1 はじめに	94
5.2 コントローラ生成法「Imaginary Training」	95
5.2.1 並列処理構造	95
5.2.2 オンラインコントローラ生成法「Imaginary Training」	96
5.4 海中ロボットへの適用	98
5.4.1 PW45の縦運動制御	98
5.4.2 Twin-Burgerの方位角制御	99
5.4.3 R-one Robotの方位角制御	100
5.5 考察	101
第6章 オンライン並列処理による外乱への適応	119
6.1 はじめに	120
6.2 オンライン並列処理	120
6.3 外乱への適応	122
6.3.1 長周期の外乱への適応	122
6.3.2 周期的な外乱への適応	123
6.4 潮流に対する適応	124
6.4.1 SSBLを用いた外乱の推定	124
6.4.2 外乱への適応法	125
6.4.3 潮流存在下での経路追従実験	126
6.5 考察	130
第7章 考察	169
第8章 結論	173
Appendix 結合荷重一覧	176
謝辞	190
参考文献一覧	192

# 図表一覽

## 第1章

Fig. 1-1 Underwater vehicle classes[8].	11
Fig. 1-2 Odyssey II vehicle (MIT).	12
Fig. 1-3 Cooperated mission of Odyssey II and Ocean Voyager II (FAU)[26].	12
Fig. 1-4 Ocean Voyager II Vehicle (FAU)[28].	13
Fig. 1-5 EAVE III vehicle (Northeastern Univ.).	13
Fig. 1-6 ABE vehicle (WHOI).	14
Fig. 1-7 Mission scenario of ABE[38].	14
Fig. 1-8 ORCA vehicle (NRL)[46].	15
Fig. 1-9 PTEROA150 (IIS, Univ. of Tokyo).	15

## 第2章

Fig. 2-1 Cruising type test-bed robot "PW45".	26
Fig. 2-2 Versatile test-bed robot "Twin-Burger".	27
Fig. 2-3 Newly developed test-bed robot "Twin-Burger 2".	27
Fig. 2-4 Front view of Twin-Burger 2 vehicle.	28
Fig. 2-5 Communication systems.	28
Fig. 2-6 Thruster units.	29
Fig. 2-7 TRAM (TRANsputer Module)	29
Fig. 2-8 Transputer based multiprocessor system of Twin-Burger vehicle.	30
Fig. 2-9 Block diagram of computer system and hardware interface mounted on Twin-Burger 1.	31
Fig. 2-10 Block diagram of computer system and hardware interface mounted on Twin-Burger 2.	32
Fig. 2-11 Distributed Vehicle Management Architecture developed by Fujii [51].	33
Fig. 2-12 Software architecture for Transputer based multiprocessor system of Twin-Burger ..	34
Table 2-1 Dimensions and specifications of Twin-Burger vehicles.	35
Table 2-2a Parameters of thruster calculation curve of Twin-Burger 1.	36
Table 2-2b Parameters of thruster calculation curve of Twin-Burger 2.	36

## 第3章

Fig. 3-1 Concept of identification with neural network.	54
Fig. 3-2 A unit of network.	54
Fig. 3-3 Logistic function.	55
Fig. 3-4 Fundamental structure of feedforward neural network.	55
Fig. 3-5 Jordan network.	56
Fig. 3-6 Elman network.	56
Fig. 3-7 Structure of Identification Network.	57



Fig. 3-8 Learning process to construct Identification Network. ....	58
Fig. 3-9 Parameters of underwater robot PW45. ....	59
Fig. 3-10 Network structure to represent longitudinal motion of PW45. ....	59
Fig. 3-11 Output of Identification Network and experimental results when elevators are trimmed in 0.5 Hz. ....	60
Fig. 3-12 Outputs of network and experimental results when elevators are trimmed in 0.1Hz. The synaptic weights are the same as those of Fig. 3-11. ....	61
Fig. 3-13 Outputs of network and experimental results when elevators are trimmed in 1.0 Hz. The synaptic weights are the same as those of Fig. 3-11. ....	61
Fig. 3-14 Outputs of network and experimental results when elevators are trimmed in eq. 3-28(a). The synaptic weights are the same as those of Fig. 3-11. ....	62
Fig. 3-15 Outputs of network and experimental results when elevators are trimmed in eq. 3-28(b). The synaptic weights are the same as those of Fig. 3-11. ....	62
Fig. 3-16 Frequency response of constructed network. ....	63
Fig. 3-17 Comparison of performance of constructed network as difference type and non-difference type network when elevators are trimmed in 0.1 Hz. ....	64
Fig. 3-18 Comparison of performance of constructed network as difference type and non difference type network when elevators are trimmed in 1.0 Hz. ....	64
Fig. 3-19 Comparison of performance between difference type network and non difference type network in frequency domain. ....	65
Fig. 3-20 Parameters of yawing motion of Twin-Burger vehicle. ....	66
Fig. 3-21 Network structure of Identification Network for yawing motion of Twin-Burger vehicle. ....	66
Fig. 3-22 Comparison of experimental results and outputs of Identification Network. ....	67
Fig. 3-23 Identification of surging motion of Twin-Burger vehicle. The results are compared with experimental results and equation of motion. ....	68
Fig. 3-24 Identification of swaying motion of Twin-Burger vehicle. The results are compared with experimental results and equation of motion. ....	68
Fig. 3-25 Identification of heaving motion of Twin-Burger vehicle. The results are compared with experimental results and equation of motion. ....	69
Fig. 3-26 Network structure to represent heading motion of Rone robot. ....	70
Fig. 3-27 Identification of heading motion of R-one robot when thrust signal is 120. ....	70
Fig. 3-28 Identification of heading motion of R-one robot when thrust signal is 150. ....	71
Fig. 3-29 Identification of heading motion of R-one robot when thrust signal is 180. ....	71
Table 3-1 Comparison of estimation error between Identification Network and ARX model. ...	72
Table 3-2 Comparison of estimation error between difference type network and non-difference type network. ....	72
Table 3-3 Parameters of vehicle motion of Twin-Burger. ....	72

## 第4章

Fig. 4-1 Mapping by a neural network from subset of input signal vectors to a subset of output signal vectors. ....	80
Fig. 4-2 Virtual learned region surrounding learning data points .....	80
Fig. 4-3 Teaching data for Mapping net. ....	81
Fig. 4-4 The structure of Mapping net and Descriptive net for the example. ....	81
Fig. 4-5 Learned region surrounding learning points for the example. ....	81
Fig. 4-6 Output of Mapping net. ....	82
Fig. 4-7 Output of Descriptive net ( $R_z = 1.0$ ). ....	82
Fig. 4-8 Output of Desc.N.N. ( $R_z = 0.5$ ). ....	83
Fig. 4-9 Output of Desc.N.N. ( $R_z = 2.0$ ). ....	83
Fig. 4-10 Small learned region. ....	84
Fig. 4-11 Output of Desc.N.N. ( $R_z = 1.0$ ) when the learned region is small. ....	84
Fig. 4-12 Output of Desc.N.N. when $R_z$ is changed according to the error density. ....	85
Fig. 4-13 The value of $R_z$ . ....	85
Fig. 4-14 Structure of Forecast Model Network and Descriptive Neural Network for representation of longitudinal motion of PW45. ....	86
Fig. 4-16 Output of network and experimental result when elevators are trimmed in 1.0 Hz (using same weight). ....	87
Fig. 4-15 Output of network and experimental result when elevators are trimmed in 0.1 Hz (using same weight). ....	87
Fig. 4-17 Simplification of network structure. ....	88
Fig. 4-18 Output of network after reducing a neuron of hidden layer. ....	89
Table 4-1 Output of Hidden layer's neuron. ....	90
Table 4-2 Synaptic weights of original network. ....	91
Table 4-3 Synaptic weights of optimized network. ....	92

## 第5章

Fig. 5-1(a) Direct adaptation approach by Yuh. ....	103
Fig. 5-1(b) Indirect adaptation approach by Fujii. ....	103
Fig. 5-2 Block diagram of proposed control system ITNN (Imaginary Training system with Neural Network). ....	104
Fig. 5-3 Concept of Imaginary Training. ....	105
Fig. 5-4 Overall structure to simulate longitudinal motion of PW45. ....	106
Fig. 5-5 Experimental results controlled by initialized controller and output .....	107
Fig. 5-6 Adaptation carried out in the Imaginary-World Part. ....	108
Fig. 5-7 Transition of evaluated value. ....	109
Fig. 5-8 Experimental results controlled by adjusted controller. ....	109
Fig. 5-9 Structure of Imaginary-World Part to deal with heading motion of Twin-Burger. ....	110



Fig. 5-10 Mapping function to initialize controller. ....	111
Fig. 5-11 Experimental results controlled by the initialized controller and the controller with the simple rule. ....	111
Fig. 5-12 Comparison of emulated motion controlled by the adjusted controller and the initialized one. ....	112
Fig. 5-13 Transition of evaluated value. ....	112
Fig. 5-14 Experimental results controlled by the adjusted controller (disturbance). ....	113
Fig. 5-15 Experimental results controlled by the adjusted controller (initial offset). ....	113
Fig. 5-16 Network structure to express the heading motion of R-one robot. ....	114
Fig. 5-17 Block diagram to simulate heading motion of R-one robot. ....	114
Fig. 5-18 Heading control simulation when thrust signal equal to 150 without disturbance. ....	115
Fig. 5-19 Heading control simulation when thrust signal equal to 150 with constant disturbance. ..	116
Fig. 5-20 Heading control simulation when thrust signal equal to 150 with sinusoidal disturbance. ....	117
Table 5-1 Evaluated value of control performance in various condisions. ....	118

## 第 6 章

Fig. 6-1 Process configuration in the on-line controller adaptaion experiments. ....	131
Fig. 6-2 Communication between processors. ....	132
Fig. 6-3 Timing of data processing and network updating in the on-line adaptation experiments. ..	133
Fig. 6-4 Heading control simulation using ITNN under constant disturbance. ....	134
Fig. 6-5 Output of Identification Network and equations of motion under constant disturbance. ...	135
Fig. 6-6 Output of Identification Network and equations of motion under constant disturbance ....	135
Fig. 6-7 Performance of the controller C_I in the simulation with equations of motion when the constant disturbance is given to the robot. ....	136
Fig. 6-8 Experiment results when the constant disturbance is given to the robot. ....	137
Fig. 6-10 Comparison of outputs of the FWD and experimental results when the constant moment is given to the robot and the FWD is updated. ....	138
Fig. 6-9 Comparison of outputs of the FWD and experimental results when the constant moment is given to the robot and the FWD is not updated. ....	138
Fig. 6-11 Performance of the controller C_I in the experiment when the constant disturbance is given to the robot. ....	139
Fig. 6-12 Result of the control simulation when the sinusoidal moments are given to the robot. ....	140
Fig. 6-13 Output of Identification Network and equations of motion under sinusoidal disturbance. ....	141
Fig. 6-14 Output of Identification Network and equations of motion under sinusoidal disturbance. ....	141
Fig. 6-15 Performance of the controller C_I in the simulation with equations of motion when the sinusoidal disturbance is given to the robot. ....	142
Fig. 6-16 Result of the control experiment when the sinusoidal moments are given to the robot. ...	143
Fig. 6-17 Output of Identification Network and experimental result under sinusoidal disturbance. ....	144
Fig. 6-18 Output of Identification Network and experimental results under sinusoidal disturbance ....	144

Fig. 6-19 Performance of the controller C_I in the experiment when the sinusoidal disturbance is given to the robot. ....	145
Fig. 6-20 SSBL positioning system for Twin-Burger. ....	146
Fig. 6-21 Calibration curve of SSBL positioning system. ....	147
Fig. 6-22 Trajectory of the robot by dead-reckoning, SSBL data on host PC, and transmitted data by command link. ....	148
Fig. 6-23 Estimated disturbance calculated as the difference between position from dead-reckoning and from SSBL. ....	149
Fig. 6-24 Block Diagram of compensation for Disturbance. ....	150
Fig. 6-25 Control results by initialized controller and target trajectory (surge). ....	151
Fig. 6-26 Control results by initialized controller and target trajectory (sway). ....	152
Fig. 6-27 Control results by initialized controller and target trajectory (heave). ....	153
Fig. 6-28 Configuration of Transputer array & processes assigned. ....	154
Fig. 6-29 Processes representing each motion located in a TRAM. ....	155
Fig. 6-30 Experimental results of surge motion. The current (0.08 [m/sec]) is given to the robot as the disturbance. ....	156
Fig. 6-31 Experimental results of sway motion. The current (0.08 [m/sec]) is given to the robot as the disturbance. ....	157
Fig. 6-32 Experimental results of heave motion. The current (0.08 [m/sec]) is given to the robot as the disturbance. ....	158
Fig. 6-33 Experimental results of heading motion. The current (0.08 [m/sec]) is given to the robot as the disturbance. ....	159
Fig. 6-34 Estimation error of dynamical model network and evaluated value of controller concerning to surge motion. ....	160
Fig. 6-35 Identification of surge motion. The input force consists of usual control force and force against the disturbance. ....	161
Fig. 6-36 Control results of surge motion in the imaginary-world part. Target trajectory is generated to deal with the disturbance (current). The Identification Network is used as the dynamical model. ....	161
Fig. 6-37 Estimation error of dynamical model network and evaluated value of controller concerning to sway motion. ....	162
Fig. 6-38 Identification of sway motion. The input force consists of usual control force and force against the disturbance. ....	163
Fig. 6-39 Control results of sway motion in the imaginary-world part. Target trajectory is generated to deal with the disturbance (current). The Identification Network is used as the dynamical model. ....	163
Fig. 6-40 Estimation error of dynamical model network and evaluated value of controller concerning to heave motion. ....	164
Fig. 6-42 Control results of heave motion in the imaginary-world part. Target trajectory is generated to deal with the disturbance (current). The Identification Network is used as the dynamical model. ....	165



Fig. 6-41 Identification of heave motion. The input force consists of usual control force and force against the disturbance. ....	165
Fig. 6-43 Estimation error of dynamical model network and evaluated value of controller concerning to yaw motion. ....	166
Fig. 6-45 Control results of heading motion in the imaginary-world part. Target trajectory is generated to deal with the disturbance (current). The Identification Network is used as the dynamical model. ....	167
Fig. 6-44 Identification of heading motion. The input force consists of usual control force and force against the disturbance. ....	167
Fig. 6-46 Trajectory of Twin-Burger. ....	168

## Appendix

Table A-1 Synaptic weights of dynamical model to represent the motion of PW45. Experimental data when $d_e$ is trimmed in 0.5Hz is used as teaching data. ....	177
Table A-2 Synaptic weights of dynamical model to represent the motion of PW45. Experimental data when $d_e$ is trimmed in 0.1Hz is used as teaching data and the difference type network is adopted as network model. ....	178
Table A-3 Synaptic weights of dynamical model to represent the motion of PW45. Experimental data when $d_e$ is trimmed in 0.1Hz is used as teaching data and the difference type network is NOT adopted as network model. ....	179
Table A-4b Synaptic weights of dynamical model to represent yawing motion of Twin-Burger 2. ....	180
Table A-4a Synaptic weights of dynamical model to represent yawing motion of Twin-Burger. ....	180
Table A-5a Synaptic weights of dynamical model to represent surging motion of Twin-Burger. ....	181
Table A-5b Synaptic weights of dynamical model to represent surging motion of Twin-Burger 2. ....	181
Table A-6a Synaptic weights of dynamical model to represent swaying motion of Twin-Burger. ....	182
Table A-6b Synaptic weights of dynamical model to represent swaying motion of Twin-Burger 2. ....	182
Table A-7a Synaptic weights of dynamical model to represent heaving motion of Twin-Burger. ....	183
Table A-7b Synaptic weights of dynamical model to represent heaving motion of Twin-Burger 2. ....	183
Table A-8 Synaptic weights of dynamical model to represent heading motion of R-one robot. ....	184
Table B-1a Synaptic weights of controller network to control longitudinal motion of PW45 (initialized controller). ....	185
Table B-1b Synaptic weights of controller network to control longitudinal motion of PW45 (adjusted controller). ....	185
Table B-2a Synaptic weights of controller network to control yawing motion of Twin-Burger (initialized controller). ....	186
Table B-2b Synaptic weights of controller network to control yawing motion of Twin-Burger (adjusted controller). ....	186
Table B-3 Synaptic weights of controller network to control surging motion of Twin-Burger. ....	187
Table B-4 Synaptic weights of controller network to control swaying motion of Twin-Burger. ....	187
Table B-5 Synaptic weights of controller network to control heaving motion of Twin-Burger. ....	188
Table B-6 Synaptic weights of controller network to control heading motion of R-one robot. ....	189

# 第1章

## 序論



## 1. 1 はじめに

近年の地球規模での環境観測に対する意識の高まりから、海洋に対する関心が大きくなってきている。海洋は地球表面の約7割を占めるにも係わらず、海底に眠る鉱物資源[1]や未知なる種多様性の宝庫である海洋生物[2]など未だにその大部分が神秘的なベールに包まれている。さらに、海洋の表層と深海域にはさまれた海洋中層域にも多様な海洋生物が存在することが判明している[3]。

海洋調査のプロジェクトとしては、国際深海掘削計画 (ODP: Ocean Drilling Program) [4]、気候変動や海洋変動の監視と予測等を目的とする世界海洋観測システム (GOOS: Global Ocean Observation System) の構築[5]や、海洋底が生成され拡大してゆく中央海嶺系における火山活動や生態系の調査を目的としたインターリッジ計画[6,7]など国際的な研究協力のもと推進されている。海洋調査を目的とした移動式の海中機器はFig.1-1のように分類され[8]、水中曳航体[9]や有人潜水艇、有索自航式の遠隔操縦型無人潜水機 (ROV: Remotely Operated Vehicle) を用いて海洋調査が行われている。しかしながら、これらの手法は海中の高水圧、暗黒、電波を伝えにくいこと等にも起因して、多額の費用、人的及び物質的資源を必要とし、支援設備も大規模なものとなる。

近年、コンピュータにおける飛躍的な処理能力の向上にも関連し、新たな海洋調査観測の道具として無索無人潜水艇 (UUS: Unmanned Untethered Submersible) が注目されている。特に自身で判断する機構を有し、環境に適応できる自律型の潜水艇を AUV (Autonomous Underwater Vehicle)、海中ロボットと呼び、米国やカナダを中心として開発が進められている[10]。海中ロボットを有索の潜水艇と比較すると、

- ・母船との索を持たないため自由に行動できる
- ・小規模な支援設備により運営でき、取り扱いが簡便である
- ・オペレータに必要とされる労力が少ない
- ・複数の艇による協調行動が有索潜水艇に比べて容易である

等の利点が挙げられるが、その反面、

- ・エネルギーを自身に搭載しなければならない、重作業に不向きである
- ・外環境を十分に把握するための信頼性の高いセンサが必要
- ・自律性を実現するためのシステムソフトウェアの構築

等の問題がある。

上記の特徴を有する海中ロボットを実現するには、ハードウェア及びソフトウェア両方面からのアプローチによる問題提起及びその解決法の探究を行い、様々な要素技術を確立する必要がある。この観点から、テストベッドとして Univ. of New Hampshire の EAVE (Experimental Autonomous VEHICLE) III や MIT の Sea Squirt、Odyssey、NPS (Naval Postgraduate School) の Phoenix、東京大学生産技術研究所の Twin-Burger 等が製作されている。さらに実用機を目指したものとして、東

京大学生産技術研究所のPTEROA150やISE(International Submarine Engineering) Ltd.のTheseus、WHOI(Woods Hole Oceanographic Institution)のABE(Autonomous Benthic Explorer)が開発され、実海域における自律航行に成功している。

このように海中ロボット開発はこの10年で無から有への立ち上がりを成し遂げており、現時点ではそれほど洗練されてはいないものの、確実に成果を挙げてきている。海中ロボットの実現には、エネルギーや外部環境把握のためのセンサの開発など克服しなければならない課題は多いが、次世代の環境調査手段として十分な可能性を秘めている。さらに、6自由度の運動を簡単に実現できるのは水中だけであり、運動学的に見ても大変興味深い。本論文が自律型潜水艇—海中ロボット—の発展に少しなりとも寄与することができれば嬉しく思う。

## 1.2 本研究の目的

先に述べたように、海中ロボットを開発するには、通信手段及び密度の乏しさや、外部環境把握が困難であること、その3次元的な運動等を考慮する必要がある。さらに、未知な動作環境において、自由に行動し目的を達成するには、得られた情報から状況を的確に判断し対処できる自律性が不可欠となる。海中ロボットの自律性を実現するには、様々な関連する技術を確立する必要があり、実際のロボット製作及び適用を通じて研究を進めることが重要である。

自律的な海中ロボットを実現するための重要課題の1つとして、運動制御システムの開発が挙げられる。海中ロボットは、自由な航行が可能なよう中性浮力に調整されるため、その運動は流体中での6自由度の運動となる。さらに、複雑な形状を有することも考えられ、運動モード間の相互影響や非線形な流体力を考慮する必要がある。また、与えられたミッションに依存して、搭載するセンサやアクチュエータを変更することが考えられ、これに起因する運動特性の変化に対して適応的に対処できる運動モデル及び制御手法が必要となる。海中ロボットの運動を表現する場合、水槽実験を通じて得られる運動方程式により表わすのが一般的であるが、正確な運動モデルを得るには多大な労力が必要とされることが知られており[11,12]、ロボットの動特性が変化した場合には再び水槽実験を行わなければならない。上記の海中ロボット固有の問題点を踏まえると、海中ロボットの運動制御システムは、運動に関して得られた情報から適応的に構築されることが望ましい。

近年、新しい情報処理の道具としてニューラルネットワーク[13]が注目されており、時変で非線形性が強いシステムや特徴抽出、画像処理へ適用されている[14-18]。これは、ニューラルネットワークが学習能力や非線形写像能力、自己組織化能力等の特徴を有するためであり、海中ロボットの運動制御システム構築に対しても適した特徴と言える。本論文では、ニューラルネットワークの特徴に注目し、ニューラルネットワークによる海中ロボットの運動の表現及び制御手法について考察する。さらに、実際に提案するシステムを海中ロボットへ適用し、実験を通じてその有効性の検討を行なう。



### 1. 3 海中ロボットの動向

海中ロボットに関する代表的な国際会議は、IEEEが主催するAUVシンポジウムとUniv. of New Hampshireが主催するUUST(Unmanned Untethered Submersible Technology)シンポジウムがあり、流体力学に関する技術から運動制御や超音波通信技術、コンピュータ・ネットワークに関する話題まで幅広く討論が行われる。例えば、UUST95(Univ. of New Hampshireで開催)の内容を分類すると

- ・潜水艇の設計コンセプト及びシステムの構成
- ・実海域における実験の成果
- ・魚の運動をモデルとしたアクチュエータ
- ・ポジショニングシステム
- ・コントローラ
- ・ナビゲーションシステム(障害物回避, ドッキング)
- ・画像処理による物体認識
- ・超音波画像等を用いた海底地形のマッピング
- ・海中におけるLEDを用いた通信
- ・シミュレータ
- ・超音波通信におけるInternet Protocol
- ・診断システム

等であり、実用的な海中ロボットを開発するに当たって必要不可欠な話題が網羅されている。以下に近年成果を挙げている海中ロボット及び関連技術について紹介する。

海中ロボットには、完全自律型の潜水艇と遠隔操縦によって航行する半自律型の潜水艇が存在する。完全自律型潜水艇は、搭載されたコンピュータにより運動の制御、障害物回避、超音波等により現在地点や目的地点を決定しながら自律航行する。半自律型潜水艇は、搭載されたコンピュータを用いて運動制御及びデータ収集を行いながら、母船上のオペレータからの超音波或いは電波を用いた指示に基づいて目標地点を決定し航行する。

#### ・完全自律型潜水艇

<Massachusetts Institute of Technology (MIT)>

MIT Sea Grantでは、AUVの知的な制御の研究を目的として、1989年に小型のテストベッド”Sea Squirt”が製作された[19]。その後、実海域や氷下で航行できる航行型海中ロボットとして1992年にOdyssey[20]、1994年にOdyssey II[21](Fig.1-2参照)が開発されている。最近開発されたOdyssey IIは、全長2.15 m、乾燥重量140 kg(海底調査用の時は160 kg)、コンピュータシステムとしてMotorola 40MHz 68030 CPUを採用しており、オペレーションシステムはリアルタイムで

マルチタスクが可能な OS-9 である。スラスタ等のアクチュエータや CTD (Conductivity, Temperature, Depth)、カメラはそれぞれ 8bit のマイクロプロセッサ (Motorola 68HC11) により独立して制御され、メインのコンピュータとは SAIL protocol[22] (通信速度 9600bps) を用いたシリアルバスにより通信を行う。電源として 48V の銀亜鉛電池を有しており、2.3 knots で 6~10 時間の航行が可能である。コンピュータや I/O インターフェイス、電源等は 2 つの Benthos 社のガラス製圧力容器に搭載されている。Odyssey II は 5 機製造され、WHOI や NERL 等に分配されており、共通のデータベースを構築している。運動の制御はエレベータ及びラダーを用いて行い、Brooks の提唱する Subsumption Architecture[23] に基づいて行動を決定する。ミッションとして氷の下での航行及び海底沿った航行を想定しており、実際にホーミングセンサを用いた実海域実験や後述する FAU の Ocean Voyager II との LBL を用いた協調行動に成功している [24~26] (Fig.1-3 参照)。

#### <Florida Atlantic University (FAU) >

FAU では、Low-Cost AUV による海中調査を目的として 1994 年に Ocean Voyager II [27,28] (Fig.1-4 参照)、1995 年に Ocean Explorer [29] を開発している。Ocean Voyager II は Odyssey II をベースに全長 2.4 m、乾燥重量 250 kg と Odyssey II よりペイロードを大きい設計となっている。コンピュータシステムは、Motorola 50MHz 68030 CPU を用いており、OS はリアルタイムでマルチタスクが可能な VxWorks である。電子計測機器類は Benthos 社のガラス製圧力容器に搭載されている。スラスタ及びエレベータ等のアクチュエータ、CTDO やドップラーログ等のセンサ類はモジュール化されており LONTalk による通信を行う。LONTalk は、シリアル通信を用いて簡単にコンピュータネットワークを構築できる LON システムのプロトコルであり、モジュール化された各ノード間において 1.25Mbps の高速通信を行なう。電源として 60 Volts の銀亜鉛電池及び鉛電池を搭載しており、速度 3 knots で 50~100 km の航行が可能である。Ocean Explorer は、センサ類が "plug and play"、つまり搭載機器の交換が簡単に行える Low-Cost AUV として設計・開発された。必要最小限のセンサ類とスラスタやエレベータ等のアクチュエータを艇体の最後尾に、計測機器類を前中部に配置し、搭載機器の大きさに従い艇の長さが 6~10 feet の間で可変な構造としている。この可変構造は Ocean Voyager II と同様に LONTalk を介した通信システムと電源の供給により実現されている。従って、高価である計測機器類の流用が可能となり、さらにコンピュータシステムの Upgrade が容易に行なうことができる。電源として 6 系統の 50 volts ニッカド電池を持ち、3 knots で約 10 時間の航行が可能である。これらの潜水艇は、フロリダ沖及びメキシコ湾での航行を想定して建造されており、複数の AUV による海洋調査を念頭において研究が進められている。



## &lt;University of New Hampshire&gt;

Univ. of New Hampshire の Marine Systems Engineering lab. (現在は Northeastern Univ. に移っている) は AUV 開発の先駆的存在であり、現在第3世代の EAVE (Experimental Autonomous VEhicle) III (Fig.1-5 参照) を2台保有している[30,31]。オープンフレームに5つのシリンダ形状の圧力容器を取り付けた構造である。コンピュータシステムとして MC68000 をベースとした lower level と MC68020 からなる upper level の2系統を有しており、lower level のシステムはスラスタの制御やセンサの入出力を扱う。Upper level のシステムには VxWorks が搭載されており、パスプランや障害物回避等の処理を行う。運動の制御は3軸方向に取り付けられた6つのスラスタを用いて行い、超音波障害物探知装置、CTD、LBL と SBL によるポジショニングシステムや Acoustic Modem (2400 bps) 等のセンサを有する。1993年の実験では2台の EAVE を用い、LBL によるポジショニングと Acoustic Modem を用いた通信に基づいた協調行動について検証している。複数の AUV の通信言語 COLA (COncceptual Language for AUVs) や障害物回避システムなど AUV 諸技術開発のテストベッドとして活躍している[32,33]。

## &lt;Woods Hole Oceanographic Institution (WHOI) &gt;

WHOI では、Mid-Ocean Ridge に沿った熱水鉱床や海底火山活動の長期観測を目的とした自律型潜水艇、ABE (Autonomous Benthic Explorer)[34]を1993年に開発している (Fig.1-6 参照)。ABE は安定して画像等のデータを採取できることが最優先されているため、定点での滞在や横方向、垂直方向への移動できるように設計されている。コンピュータシステムは、運動制御及びナビゲーション用に INMOS Transputer を2個、スラスタや I/O インターフェイス用に Motorola 68HC11 CPU を採用しており、SAIL Bus により通信を行う。さらに、長期滞在のため電源の管理をするコンピュータを用いており、Sleeping Mode での消費電力は 0.048 W である。これらは6個の Benthos 製ガラス球に配置され、上部2つの容器内に搭載されている。電源は48V鉛電池であり、下部の圧力容器に収納されている。これらのカウルはエネルギー消費と航続距離、速度の関係から最適な形状に設計されている[35]。姿勢制御はスラスタを用いて行い、前進方向に3個、横方向に2個、垂直方向に2個取り付けられている。センサとして CTD、磁気センサ、Video camera、超音波障害物検知装置などを搭載しており、LBL によりナビゲーションを行なう。また、指北方式のジャイロ[36]では磁場の影響で正確な値が得られないため、角速度センサとのハイブリッドにより方位を算出している。

1回のミッションは Fig.1-7 のように Descent、Survey、Ascent からなる。1995年に、the central Juan de Fuca Ridge (北緯46度、西経129度、深度約2400m) において5回のミッションをこなしており、合計35kmの自律航行に成功している。1回のミッションは、夕方に投機し翌朝回収する約10時間であり、海底では5~100

mの高度を維持しながら約3.5時間で7Kmのトラッキング(約1knot)を行なっている。磁場のベクトルや深度、高度、CTDの垂直分布を測定しており、20,000 nT(東京の磁場は60,000nT)を越える磁場が存在することが判明している[29,30]。このようにABEは近年著しい成果を挙げており、AUVによる海洋観測の可能性を世界に示している。

#### <International Submarine Engineering Ltd.>

カナダのISEは、自律型及び半自律型のUUSを開発しており、自律型潜水艇として1984年にARCS[8]を製作した。さらに北極圏での海底ケーブルの施設を目的としたTHESEUS[39]を1994年に開発している。THESEUSは、全長11 m、乾燥重量10 ton、エネルギー源は銀亜鉛電池であり、70時間の連続航行が可能である。実際に北極圏において実験を進めている。

他にも、Naval Postgraduate SchoolのテストベッドPhoenix[40,41]や欧州で共同開発されているMARIUS[42]、中国のExplorer-01[43]等、各国で開発が進められている。

#### ・半自律型潜水艇

代表的な半自律型(遠隔操縦型)の潜水艇として、NRaDのAUSS(Advanced Unmanned Search System)[44]やNaval Research Lab.のORCA(International Submarine Engineering Ltd.がDolphinとして製作、Fig.1-8 参照)[45,46]が存在する。

AUSSは、自身に搭載されたCPUとセンサを用いて姿勢制御を行う一方、超音波により潜水艇が捉えた映像を母船に送信し、母船に搭乗しているオペレータにより目的地点を決定するシステムである。

ORCAはディーゼルエンジンを搭載した半没水型の潜水艇であり、水面上のアンテナを介して母船と通信し、行動を決定する。10 knotsで24時間の航行が可能であり、超音波を用いて深度300 mまでの海底地形を調査する。

日本における海中ロボットの開発は、1970年代に製作されたOSR-Vに始まり、その後大きなプロジェクトとして、西日本流体技研の「ウォーターバード」、東京大学生産技術研究所の「プテロア計画」、東京大学生産技術研究所と三井造船の共同開発による「R-one計画」があり、近年KDDが海底ケーブルの調査を目的とした海中ロボットAQUA EXPLORER1000を開発している[8]。

#### <東京大学生産技術研究所>

東京大学生産技術研究所では、1989年にPTEROA150 (Fig.1-9 参照)[47]、その



テストベッドとして1990年にPW45[48]、1992年にALBAC[49,50]及びTwin-Burger[51-52]、1995年にTwin-Burger 2を開発している。

PTEROA150は航行型海中ロボットであり、海底の調査を目的として製作された。実海域において、グライディングにより潜降し、超音波障害物センサを用いて海底から一定高度保持しながら航行することに成功している[54]。CPUはIntel 80186を搭載しており、左右のエレベータと2つのスラスタにより姿勢制御を行なう。電源は48Vのニッカド電池であり、3 knotsで40分の航行が可能である。

ALBACは実用機を目指して設計されており、信頼性の向上のため稼働部分は全て圧力容器内に収納されている。従って、スラスタ等のアクチュエータを持たず、圧力容器内の重りを前後または左右方向に移動させることにより、重心位置を変化させ姿勢制御を行う。CPUは2つのNEC V50であり、障害物センサや温度計を搭載している。1995年、静岡県三津浜沖において、深度300m迄の鉛直方向の温度分布の調査を行なっている[50]。PW45及びTwin-Burgerについては、本研究内容の適用例として取り上げているので、第2章で詳しく述べる。

#### <KDD>

KDDでは、海底ケーブルの修繕を目的とした航行型海中ロボットAQUA EXPLORER 1000 (AE1000)を開発している。CPUはMotorola 68030であり、左右のエレベータ及びスラスタにより姿勢制御を行う。一対の磁気センサ、ビデオカメラ、障害物センサ等のセンサを搭載している。PTEROA150と同様、グライディングにより潜航し、海底に到着後重りを投棄する。その後、海底ケーブルが作り出す磁場からケーブルの位置を推定し、ケーブルに沿って航行しながら、超音波により海底画像を支援母船に通信する。1994年、深度200mの海底において、航行速度1 knot、2.6 kmに渡るケーブルトラッキング及びビデオ撮影に成功している[55]。

このように、実海域においても海中ロボットを用いた海底調査が始まっており、ロボットに与えられたミッションが比較的明瞭である場合には良好な成果を挙げつつある。今後の展開として、ある水準以上のロボットの知能化及び多機能化を行ない、単体のロボットだけでも充分な海洋調査が可能なシステムの開発、或いは必要最小限の機能を有するロボット群を用いた海洋調査システムが考えられる。前者のシステムでは、単体で多くの情報の収集が可能であるが、高度の知能と情報処理能力が必要であり、システムの巨大化や取り扱いが複雑化が問題になる。後者のシステムは、単体のロボットのメンテナンスや取り扱いが容易である一方、乏しい通信手段の中でシステム全体をどうやって協調させるか等の問題が存在する。今後は、これらの妥協点を見出す方向で海中ロボットが進んでいくものと思われる。以下に、最近の海中ロボット諸技術について簡単に紹介する。

海中ロボットの知能は、得られる情報量とその情報処理手段に大きく依存する。海中においても、情報源として画像が目立っており、得られた海底面の画像をオンラインでモザイク処理する研究[56]や、画像によるポジショニング[57]やトラッキング[58]等の研究が進められている。また、複数のロボットによる海中探査手法や通信手段についてもシミュレーション或いは実海域実験において検討されている[59,24,31]。興味深い発表としては、魚の運動をモデル化し考案されたアクチュエータが挙げられる。MITのTesting Tank Tuna[60,61]の構想とNortheastern Univ.のEel[62]に関する実験検証が行われており、構想段階ではあるが効率の良い推進器につながる可能性を秘めている。

これらの話題はInternetを通じても見聞することが可能であり、参考までに海中ロボットに関するWWWサイトを挙げておく。

・ IIS, Univ. of Tokyo	<a href="http://underwater.iis.u-tokyo.ac.jp/">http://underwater.iis.u-tokyo.ac.jp/</a>
・ MIT Sea Grant	<a href="http://web.mit.edu/afs/athena/org/s/seagrant/www/auv.html">http://web.mit.edu/afs/athena/org/s/seagrant/www/auv.html</a>
・ FAU	<a href="http://www.oe.fau.edu/AMS/AMS.html">http://www.oe.fau.edu/AMS/AMS.html</a>
・ NPS	<a href="http://www.cs.nps.navy.mil/research/auv/">http://www.cs.nps.navy.mil/research/auv/</a>
・ WHOI	<a href="http://www.dsl.whoi.edu/">http://www.dsl.whoi.edu/</a>
・ MBARI	<a href="http://www.mbari.org/">http://www.mbari.org/</a>
・ Heriot-Watt Univ.	<a href="http://www.cee.hw.ac.uk:80/subsea/">http://www.cee.hw.ac.uk:80/subsea/</a>

## 1. 4 海洋工学へのニューラルネットワーク適用

ニューラルネットワークを海中ロボットの運動制御へ応用した例は、藤井らによるフィードバックコントローラ[63-66]や、須藤らによる海中ロボットの定高度航行[67-70]、Yuhによる強化学習法を用いたコントローラ生成法[71,72]がある。藤井らの手法は、運動モデルを介して間接的にコントローラを調整するのに対し、Yuhの手法は制御誤差から直接的にコントローラを生成する。

船舶に適用した例として、R.S. Burnsによる教示型のコントローラ[73]や小川原らによる学習型フィードフォワードコントローラ[74]等がある。R.S. Burnsは、シミュレーションにおいて速度を変えた場合の制御性能を最適制御と比較しており、良好な結果を。小川原らは、PIDコントローラと学習型フィードフォワードコントローラを組み合わせて、風などの外乱を補償するシステムを構築している。また、山本らはスラスタに変わる手法として、船のラダーを左右方向に振動させて推力を得る手法を考案しており、推力の制御にニューラルネットワークを用いている[75,76]。

他にも、超音波通信等で問題になるマルチパスを考慮して、音源の方向を特定する手法[77]や、海底面と海面での温度、海底面でのマルチパスの影響を含んだ



到達時間等から、1年間の深度変化による音速への影響を出力するネットワークの構築[78]、プラントの故障診断[79,80]、パスプランニング[81]等にも応用されており、ニューラルネットワークは幅広い信号処理の道具として今後の発展が期待されている。

## 1. 5 本論文の内容

2章において、本論文で主として用いた海中ロボットについて述べる。3章では、ニューラルネットワークを用いた海中ロボットの運動の表現手法を提案し、提案する手法の具体的な航行型及び非航行型の海中ロボットの運動表現への適用を通じて有効性の確認を行う。4章では、ニューラルネットワークの問題点である知識の蓄積方法やネットワーク構造の最適化について提案する。

5章では、提案した運動表現手法の応用した運動制御システム及びそのコントローラ生成手法を提案する。提案する運動制御システムを海中ロボットの運動制御に適用し、その制御結果の検証を行なう。さらに、提案する運動制御システムを並列処理機能を有するシステムに適用し、オンラインでの制御実験を通じてシステムの適応性を検討する。

6章では、潮流などの外乱への対処を考え、SSBL及び超音波通信によるポジショニングシステムを利用した外乱に対する適応手法について考察し、実際に海中ロボットを用いた実験を通じて、その可能性の議論を行う。

最後に、本論文で提案する海中ロボットの運動の表現手法及び制御手法について考察を行い、その展望を述べる。

(潜水機)



Fig. 1-1 Underwater vehicle classes[8].





Fig. 1-2 Odyssey II vehicle (MIT).



Fig. 1-3 Cooperated mission of Odyssey II and Ocean Voyager II (FAU)[26].  
(Left: Odyssey II, Right: Ocean Voyager II)



Fig. 1-4 Ocean Voyager II Vehicle (FAU)[28].

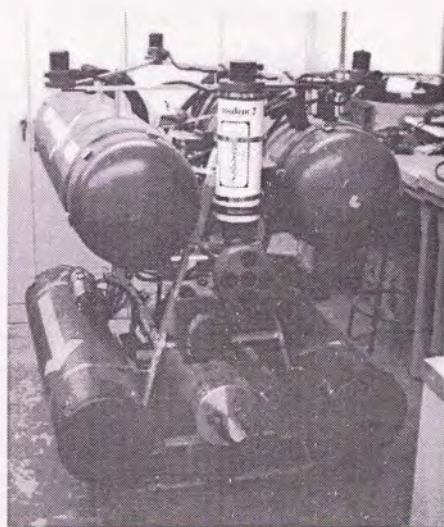


Fig. 1-5 EAVE III vehicle (Northeastern Univ.).





Fig. 1-6 ABE vehicle (WHOI).

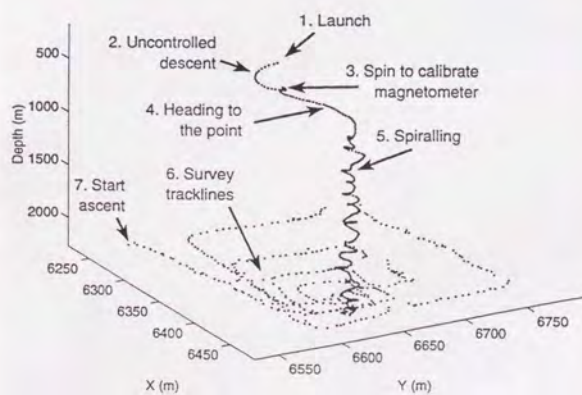


Fig. 1-7 Mission seenario of ABE[38].

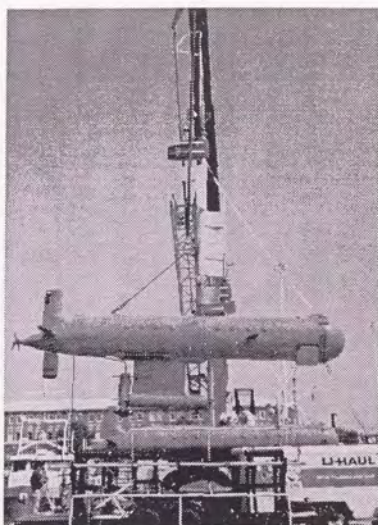


Fig. 1-8 ORCA vehicle (NRL)[46]

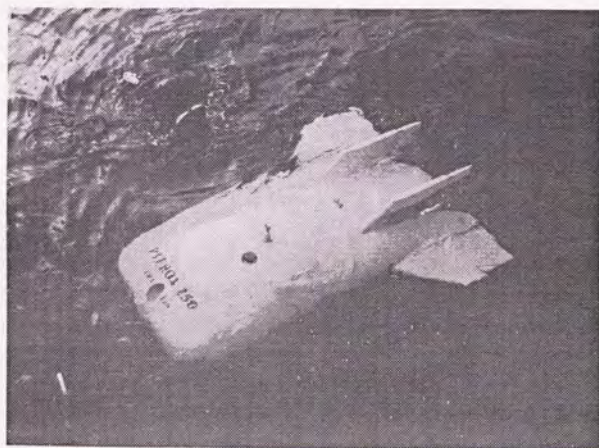


Fig. 1-9 PTEROA150 (IIS, Univ. of Tokyo).



## 第2章

# 対象とする海中ロボット

## 2. 1 はじめに

自律的な海中ロボットを実現するには、シミュレーションにより新しいアルゴリズムや理論を検証すると同時に、海中という環境においてロボットを実際に動作させることが重要である。その観点から、1章で述べたように様々な形態の海中ロボットが開発され、水槽実験や実海域実験を通じて海中ロボット固有の技術の研究が行われている。第2章では本論文に主として用いられている海中ロボットPW45[48]及びTwin-Burger[51,52] 1号機、2号機的设计思想とシステム構成について述べる。

## 2. 2 航行型海中ロボット「PW45」

PW45は、航行型海中ロボットの運動特性の把握及び縦運動に関する運動制御システムの検討を目的とした無索の海中ロボットであり、1990年に藤井らにより開発された。以下の点に留意した設計となっている[53]。

- ・索を有しない自律型のロボットとすること。
- ・数人で取り扱いが可能なように小型軽量を目指す。
- ・搭載物は運動制御に必要とされる最小限のものとする。
- ・メンテナンスを容易にするため搭載機器をモジュール化する。

外形は、Fig. 2-1に示すように、縦断面がNACA0030の翼形でありPTEROA150の中心線における縦断面と同じ形状を採用している。全長45 cm、全幅36 cm、乾燥重量約7 kgであり、アクチュエータとして胴体後部にマグネットカップリング式のスラスト1基と胴体左右に2つのエレベータを搭載している。センサとして、ロール角とピッチ角を出力する2軸の傾斜計と差圧方式の深度センサを有しており、左右のエレベータを用いてロール、ピッチ、深度に関して制御を行う。コンピュータシステムはNEC V50と専用のI/Oインターフェイスからなり、RS232を用いたシリアル通信によりHost PCからプログラムのダウンロード及びデータ収集を行なう。エネルギー源は7.2 Vニッカド電池であり、航行速度約0.3 m/secで2時間の航行が可能である。

運動制御システム開発用のテストベッドとして活躍しており、主にニューラルネットワークを用いた制御手法や運動の表現方法の検討に使用され、良好な成果が報告されている[48,82~84]。



### 2.3 汎用テストベッドロボット「Twin-Burger」

Twin-Burgerは、自律海中ロボットの自律性及び知的行動に関する研究を行うためのテストベッド、という位置づけで1992年に藤井らにより設計・開発された[51-53] (Fig. 2-2)。Twin-Burgerには、自身で外部環境を判断しながら、与えられたミッションを遂行するため

- ・ 内部状態と外部環境を認識するために必要なセンサ
- ・ 他のロボットやダイバーとの通信機能
- ・ 3次元での運動を可能にするためのアクチュエータ
- ・ 十分な処理能力を有するコンピュータシステム

を搭載している。さらに、実験を行う際の利便性を考えて、

- ・ 数人で扱える程度に小型軽量
- ・ ホバリング能力
- ・ コンピュータシステムの変更が容易な圧力容器
- ・ センサやアクチュエータの取り付け、変更が簡単に行えるオープンフレーム構造

等の特徴を有する設計となっている。

さらに、複数の海中ロボットによる協調行動の研究への発展を念頭において、1994年からTwin-Burgerの2号機であるTwin-Burger 2の開発を進め、1995年に筆者らにより完成された (Fig. 2-3)。以下では、両機に共通するロボットの諸元やセンサ類、アクチュエータ類、コンピュータシステム及び2号機における改良点について述べる。

#### 2.3.1 諸元

Twin-Burgerは、全長1.54 m、乾燥重量約120 kgであり、フレーム上部に取り付けられた2つのFRP製耐圧容器、フレーム下部のアルミニウム製シリンダ、フレームに取り付けられた計測機器と推進器から構成される。その主要寸法及び搭載機器をTable 2-1に示す。高速での航行は想定していないため、上部圧力容器の形状設計には流体力学的な特性は重視されておらず、取り扱いが容易な直方体に丸みを帯びた形状を有している。

比較的軽量であるコンピュータシステムとその周辺機器を上部圧力容器内に、バッテリーや水中カメラ等の重量の大きい装置を下部シリンダ内に配置し、重心位置と浮心位置の距離を大きくすることにより、ロールとピッチに関して静的な安定を確保している。

#### 2.3.2 センサ

搭載されているセンサ類は、ロボット自身の内部状態を把握するためのセンサと外部環境の状態を獲得するためのセンサに分類される。

## 1) 内部状態把握のセンサ

3軸方向の並進運動を計測するため、プロベラ式流速形と圧力式の深度計、さらに各軸方向にピエゾ抵抗型加速度計を搭載している。3軸周りの回転運動を計測するためのセンサとしては、ロール角、ピッチ角、方位角及びそれらの角速度を出力する Watson 社製の AHRS (Attitude and Heading Reference System) を1基搭載している。

プロベラ式流速計は、プロベラの回転により生成されるパルスを計測することにより対水速度の計測を行なうシステムである。搭載しているセンサは $\pm 2.0$  [m/sec]の範囲を測定することが可能であり、零点近傍を除く領域で対水速度に比例した回転数を出力する。

加速度計は、 $\pm 2.0$  Gの範囲を測定することが可能であるが、海中での運動を考慮すると通常の手法では思わしい結果を得ることが出来ない。そのため、実際に得られた計測値を用いるには至っていない。

深度計は、水槽での実験での使用を考えて10 [m]まで測定できるものを採用している。1号機では差圧式の深度計を用いていたため、容器内の温度上昇や圧力容器の体積変化に起因する深度変化が問題となったため、2号機では絶対圧式の深度計を用いている。

AHRS は、ロール角 $\pm 180$  [deg]、ピッチ角 $\pm 90$  [deg]、方位角 $\pm 180$  [deg]、角速度 $\pm 100$  [deg/sec]の範囲で計測することができる信頼性の高いセンサであり、速度を入力することにより計測値の補正をすることも可能である。

## 2) 外部環境把握のためのセンサ

海中という未知な環境において動作するため、150 [kHz]の8チャンネル超音波測距センサと CCD カメラシステムを搭載している (Fig. 2-4 参照)。

8チャンネルの測距センサは、前後方向、前方左右30度方向、左右方向及び上下方向の、8方向に関して障害物迄の距離を測定する。各々のセンサでは、1秒毎に1回の計測を行っている。

CCDカメラシステムはシリンドラ前部に取り付けられており、水中画像をNTSC信号として出力する。パン・チルト機構が備わっており、1号機ではパンは左右100 [deg]、チルトは真下から上方向15 [deg]まで稼働出来るシステムを自作している。2号機に搭載されたカメラシステムは、DEESEA社から購入したもので、パン・チルトは40 [deg/sec]の速さで160 [deg]まで迄行え、ズーム機能と焦点調整機能が備わっている。



### 2.3.3 通信システム

複数のロボットやダイバー等と協調して行動をする場合、何らかの方法でコミュニケーションをとる必要がある。そこでTwin-Burgerは超音波コマンドリンクシステム及びTwin-Burger前面に取り付けられた視覚通信システムの2系統の通信システムを搭載している。

超音波コマンドリンクシステムは、情報を発信する場合、受信先と送信元、8bitsのデータを約50 [kHz]の周波数帯の超音波を用いて水中に発信し、情報を受け取る場合は自分宛のデータだけ受信する通信システムである (Fig. 2-5 参照)。最大8台の間で交信が可能であり、現在1号機、2号機及びHost PCの計3台による通信が可能となっている。

視覚通信システムは、5枚のEL (Electro-Luminescent) パネルによる5 bitsのビットパターンにより通信するシステムであり、主としてダイバーとの通信手段として考案されたシステムである。ELパネルは、AC100 [V]を加えることにより面的に発行する。水中ライトと比較すると、距離による照射面積の変化が少ないので、点滅の認識が容易である。ロボット間の通信では、CCDカメラシステムを用いて認識することにより通信が可能である。

### 2.3.4 アクチュエータ

Twin-Burgerは、推進器として5基のスラスタを有しており、胴体左右に取り付けられた前後方向のスラスタを用いて前進及び回頭を行ない、中心部に取り付けられた左右方向のスラスタ1基と上下方向のスラスタ2基を用いて横方向及び上下方向の並進運動を行なう。上下方向の2基のスラスタによりロール運動も制御可能であるが、本論では考えていない。

搭載しているスラスタは、三井造船社製の40[W]スラスタであり、PW45と同様マグネットカップリングによる水密方式を採用している (Fig. 2-6 参照)。そのため、司令値の急激な変化による脱調を起こしやすいという問題が存在する。スラスタにはエンコーダが備わっており、スラスタの回転数からロボットの前進速度の算出やスラスタの状態の把握を行なうシステムの構築も可能である。

スラスタの動特性は、海中ロボットの運動に直接影響するため、Yoergerら[85]やHealeyら[86]により詳しく検討が行なわれている。本研究においては、海中ロボットの動特性にスラスタの特性が含まれるとして取り扱うので、ボラードブル状態で求めた制御電圧、回転角速度、推力の関係式から(3.1)式を用いて各スラスタにおける制御電圧を算出する。ここで、 $\Omega$ はスラスタの回転数、 $F$ は制御指令値、 $V$ は電圧値である。パラメータ $a$ 、 $e$ は定数であり、1995年7月において求めた値をTable 2-2に示す。

$$\Omega|\Omega| = a \cdot F + b$$

$$V = c \cdot \Omega^2 + d \cdot \Omega + e \quad (3.1)$$

### 2. 3. 5 コンピュータシステム

これまで述べてきたように、海中ロボットの自律航行を実現するには、様々なセンサから入力される情報の処理や状況判断、行動の決定等を並行して行なう必要がある。ハードウェアに近い低レベルの処理は、メインのコンピュータにおける処理とは独立して行われることが望ましい。近年開発されている海中ロボットのコンピュータシステムは、複数のマイクロプロセッサにより構成されるものが多く、Odyssey II[21]やABE[27]においてはメインとなるプロセッサの他に、スラスタやセンサを管理するプロセッサとしてMotorola 68HC11を駆使している。また、オペレーティングシステム(OS)は、並列処理が可能であり、タイマーによる時間管理や処理の切り替えをサポートしているものが採用されていることが多く、VxWorksやOS-9が用いられている。Twin-Burgerは並列処理の取り扱いが容易であり、演算能力の高いINMOS社製のトランスピュータ(Transputer)[87]から構成されるマルチプロセッサシステムを採用している。

#### 1) トランスピュータ

Twin-Burgerに搭載しているT800トランスピュータは、32 bits マイクロコンピュータ、64 bitsの浮動小数点演算機(FPU)、及び通信速度20 Mbpsのシリアル通信リンクを4本持つ。並列処理を前提としたプロセッサであり、1つのプロセッサ内において複数のプログラムを実行させることができる。さらに、シリアルリンクを用いて他のプロセッサと接続することにより、複数のプロセッサによる並列処理を容易に実現することができる。4本のシリアルリンクのうち、2本はハードウェアでバイプライン形状に接続されており、残りの2本はソフトウェアを用いて自由に他のプロセッサと接続することができる。トランスピュータには並列処理用の特別なOSが存在せず、プロセッサとそれに搭載するプログラムの関係をプログラムのコンパイル時に読み込ませることにより、複数のプロセッサによる並列処理を実現している。市販されているものは、Fig. 2-7に示されるように、プロセッサと外部RAMを組み合わせたTRAM (TRANsputer Module) 形状のものである。実際にコンピュータシステムを構築する際には、マザーボードに載せて使用する。

プログラムのソースコードはOccam2或いはINMOSが開発したANSI C準拠のC言語により記述可能であり、Twin-BurgerのソフトウェアはINMOS ANSI Cにより開発している。一般のC言語との主な違いは、プロセス管理用の関数と宣言、通信用のチャンネル関数と宣言、時間管理用のタイマー関数等が加わっている点である。SUN Workstation 或いはPCでコンパイル終了後、ISERVER Protocolに従って実行コードがトランスピュータ群に転送される。転送終了後、自動的にプログラムが起動され、ホストコンピュータとは独立して実行することが可能である。



## 2) コンピュータシステム

Twin-Burger のコンピュータシステムは、演算用の T800 シリーズとインターフェイス用の FPU を持っていない T400 シリーズから構成される。Twin-Burger 1 号機及び 2 号機のメインコンピュータシステムを Fig. 2-8 に示す。

1 号機では 16 枚分のスロットを有するダブルユーロカードサイズのマザーボード B012 上に、1Mbyte の外部 RAM を有する T800 TRAM を 8 枚、データログ用の 8Mbyte の外部 RAM を有する T800 TRAM を 1 枚、画像処理用の T800 を 1 枚の計 10 枚搭載している。更に、インターフェイス用の T425 TRAM を前方の圧力容器内に 2 枚、後方の（メインのコンピュータと同じ）圧力容器内に 2 枚搭載している。そのコンピュータシステムとセンサ、アクチュエータ間の関係を Fig. 2-9 に示す。後方の圧力容器（Hull #1）には AD Card と DI/DO Card が配置されており、AD Card には、AHRS、加速度計、深度計からの信号が入力される。AD 変換の後、メモリを介して T425 TRAM に取り込まれる。DI/DO Card には、8ch の測距センサへの入出力信号と EL パネルへの出力信号が接続されており、同様にメモリを介して T425 TRAM によりコントロールされる。前方の耐圧容器（Hull #2）には、Counter と DA 機能を持つ Servo Card と、DI/DO Card がある。Counter には、流速計及びスラスタからのパルスが入力され、DA からはスラスタ及びカメラシステムに対して司令電圧が出力される。DI/DO Card には超音波通信システムが接続されており、メモリを介して T425 TRAM に取り込む構成になっている。これらの計測信号は、T425 TRAM において物理量に変換された後、シリアルリンクを用いてメインコンピュータシステムに通信される。

2 号機のコンピュータシステムも基本的な設計は同じであり、B012 マザーボード上のメインコンピュータシステムと VME タイプのマザーボード 4 枚に搭載された I/O インターフェイスにより構成される。センサやアクチュエータの配置が 1 号機と同じ配置としてあるため、前方の圧力容器内に VME タイプのマザーボード 2 枚と後方の圧力容器内に B012 マザーボードと VME タイプのマザーボード 2 枚配置している。

B012 上には、1Mbyte の外部 RAM を有する T800 TRAM を 6 枚と、データログ用の SCSI TRAM を 1 枚、画像処理用の T800 TRAM を 1 枚の計 8 枚搭載している。長時間のデータ採取及び信頼性向上のため、2 号機では SCSI TRAM を採用しており 250 Mbyte の SCSI ハードディスクと接続されている。インターフェイス用の 4 枚の VME タイプのマザーボードは、各 1 枚の T800 TRAM と 3 枚の T400 搭載 DIO TRAM、AD TRAM、DA TRAM から構成されている。センサやアクチュエータと、I/O インターフェイスとの関係は基本的に 1 号機と同様であるが、AD TRAM 等の TRAM 形状のインターフェイスを採用したことにより、Fig. 2-10 に示すように、計測値の取り扱いが容易になっている。つまり、センサやアクチュエータの

変更を行なう場合でも、それらに応じたI/O TRAMの差し替えにより柔軟に対処できる。

### 2. 3. 5 2号機の改良点

これまで述べたように、1号機の反省点から、2号機は主にコンピュータシステムに改良が加えられている。改良点をまとめると以下のようになる。

- ・ハードディスクを搭載し、長時間の計測データ取得とその信頼性の向上を行なう。これは、計測データが揮発性のRAMに保存してあるため、ホストPCとTwin-Burger間の通信に障害が発生した場合に計測したデータが得られないという問題を解決し、記憶容量に起因するロボット行動の時間制限を無くすことを目的としている。
- ・TRAM形状のI/Oインターフェイスボードを採用し、センサやアクチュエータ等の搭載機器の変更に対して、柔軟に対処できるシステムとする。
- ・絶対圧式の深度計を採用し、圧力容器内の温度変化等に起因する内圧の変動による影響を排除する。
- ・カメラシステムの稼働部分を耐圧容器内に配置することにより、カメラシステムの信頼性を向上する。
- ・コンピュータシステムの取り扱いを容易にするため、簡単に取り外しできるようコンピュータシステムをモジュール化している。
- ・下部シリンダ内に搭載されるバッテリーの交換が容易にするため、車輪付きのレールを採用している。
- ・上部圧力容器の耐圧強度を上げるため、CFRP製の圧力容器とする。更に、圧力容器内の圧力を約0.9気圧として、水面における漏水対策を行なう。

上記以外にも、改良が考えられる項目として、

- ・センサ、アクチュエータ等にマイクロコンピュータを搭載し、シリアル通信により情報の伝達行ない、信号線を少なくする。これにより、メンテナンスが容易となると同時に、ケーブル類から生じる雑音の影響を小さくする事が可能である。
- ・センサ、アクチュエータ等の搭載機器の変更への対処を容易にするため、"Plug & Play"が可能な通信システムを搭載する。
- ・エネルギー密度の高いエネルギー源を採用し、重量の軽量化を図る。
- ・海中ロボットの自律行動を研究するためのテストベッドという位置づけから、演算能力の高いプロセッサを搭載する。この項目は現在進めており、PowerPCプロセッサとT800から構成されるParsytech PowerTRAMを自作のマザーボード上に搭載し、Twin-Burgerへの導入を検討中である。

等が挙げられ、実用機を開発するための様々な知見が得られている。



### 2. 3. 6 ソフトウェア構造

Twin-Burgerのコンピュータシステムは、複数のプロセッサから構成されるマルチプロセッサシステムである。そのため、通常のプログラミング手法とは異なった考え方、及びソフトウェア構造が必要となる。藤井らは、マルチプロセッサシステムによる並列分散処理を前提としたロボットの行動制御システムとして、分散型行動アーキテクチャ (DVMA: Distributed Vehicle Management Architecture) を提案している[43-45]。DVMAは、直接センサやアクチュエータの処理を行うハードウェアレベル、状況に応じた行動スクリプトが記述されている機能レベル、及びロボットの行動を決定する行動レベルから構成されている (Fig. 2-11 参照)。DVMAをTwin-Burgerの行動制御システムとして導入し、ロボットが外部環境に応じて行動することを水槽実験において示している。

本論では、藤井らの考え方を継承しつつ、Twin-Burgerに搭載されているマルチプロセッサシステムの取り扱いを容易にすることを考え、Fig. 2-12に示すソフトウェア構造を採用した。ソフトウェアの構造を考える場合、対象とするプロセッサの処理能力や機能が大きく影響する。トランスピュータは通信のためのシリアルリンクを4本有するが、自由に接続できるシリアルリンクは2本であり、プロセッサの配置がプログラミングに大きく影響する。本論文では、各プロセッサをバイブライン状に接続し、海中ロボットの行動に関する基本的な情報を共有するソフトウェア構造としている。各プログラムの固有の処理とは独立して、基本的な情報を共有するための処理プロセスが実行されており、ある一定の周期で情報の更新が行われる。その最新の情報に基づいて、各プロセス固有の処理が非同期に実行されるため、待ち時間が存在せず効率的な処理を行うことが可能となる。従って、複数のプロセッサの扱いが容易になり、システム開発に要する労力が軽減されている。残りの2本のシリアルリンクは、自由に接続できるため、ローカルな情報の通信に利用されており、研究内容に応じて変更を行っている。

## 2.4 考察

海中ロボットの自律的行動を実現するには、実際に海中ロボットの開発を行い、実環境において動作させる事が重要であるという認識から、これまで筆者は上記のロボット開発に携わってきた。そこで得られた知見及び経験は、今後実用機を開発する場合にも多いに役立つものと思われるので、ここで議論しておく。

海中ロボットを開発する上で重要な要素は、海中ロボットの開発目的を含めた設計思想と、搭載する機器類のバランスである。ロボットの開発目的により、ロボットの形状やコンピュータシステムの処理能力、エネルギー源が決定される。例えば、海底面に沿ってデータの収集を行う場合、搭載する計測装置に従って、航行速度やロボットの走行安定性などの要素が決まり、それらの要素が運動制御システムにおいて重要となる。逆にロボットの姿勢に因らず計測できる計測機器を搭載している場合は、航続距離や航行速度に重点が置かれた形状となるであろう。第一に、ロボットの設計思想を明確にすることが要求される。

搭載機器類のバランスとは、単体での精度、重量、容積、価格などの要素の関係と全体の搭載機器間におけるそれらの要素の関係の事を差している。海中ロボットに搭載する機器は、陸上のロボットの機器類よりも重量や容積の重要度が高い。精度が良くとも、重量や容積が軽量・小型でなければ搭載が困難である。また、ロボットの運動に関する計測データの精度が良くとも、外部環境を把握する計測機器の精度が十分でない場合、海中ロボットの自律航行は外部環境の計測精度に大きく影響される。耐圧容器やコネクタ類も同様である。

これらを踏まえてロボットの開発を行わなければ、投資した労力に見合う成果を挙げるのは困難であろう。



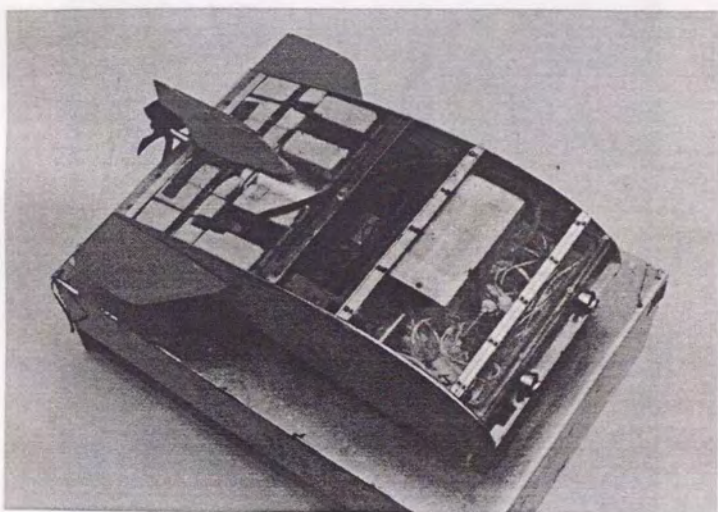


Fig. 2-1 Crusing type test-bed robot "PW45".

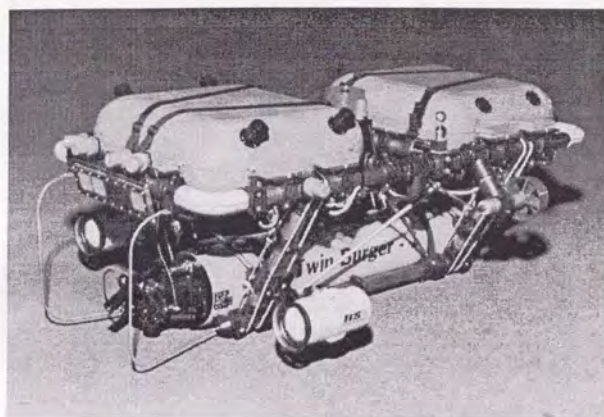


Fig. 2-2 Versatile test-bed robot "Twin-Burger".

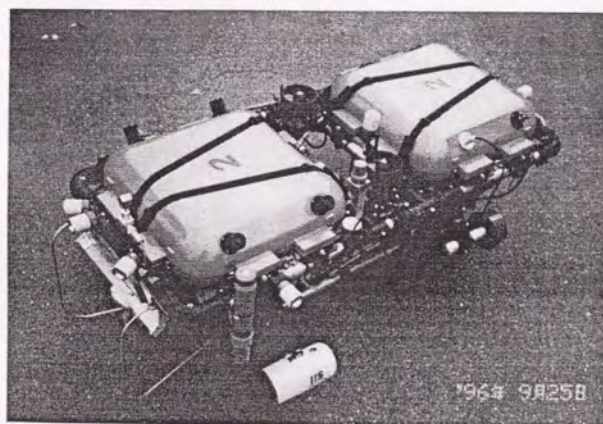


Fig. 2-3 Newly developed test-bed robot "Twin-Burger 2".



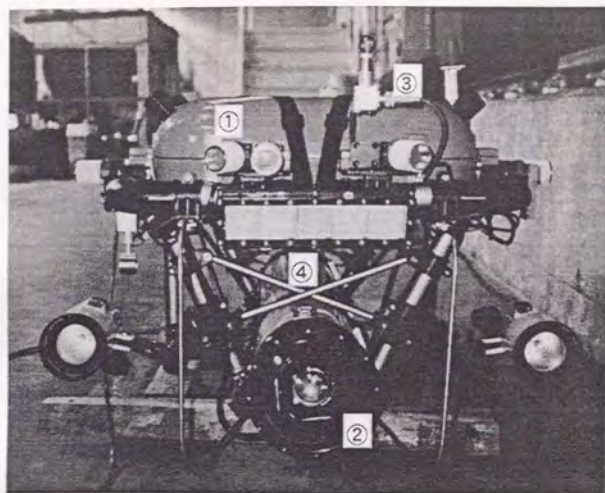


Fig. 2-4 Front view of Twin-Burger 2 vehicle.

①: Ultrasonic range finders and ②: CCD camera system are mounted to observe underwater environment. ③: Propeller type flow sensor.

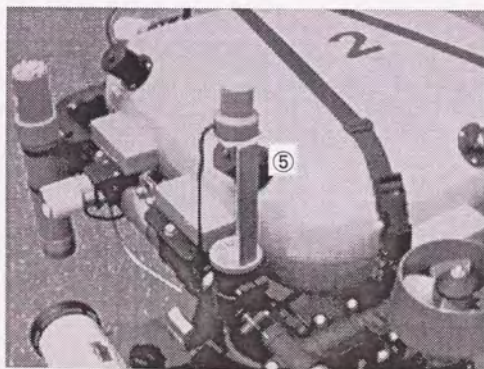


Fig. 2-5 Communication systems. ④: vision communication system with EL panels and, ⑤: Ultrasonic command link communication system.

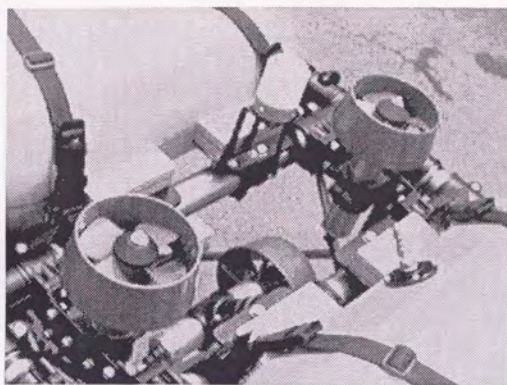


Fig. 2-6 Thruster units.

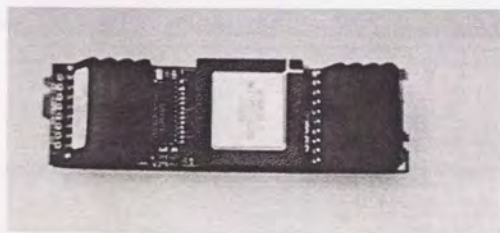


Fig. 2-7 TRAM (TRANsputer Module).



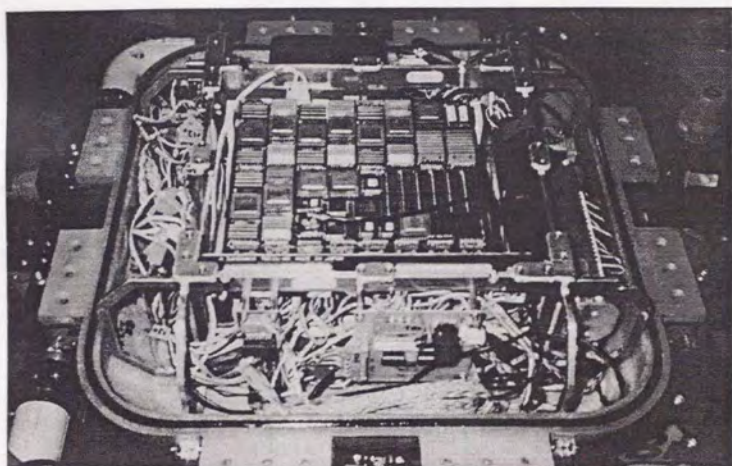


Fig. 2-8 Transputer based multiprocessor system of Twin-Burger vehicle.

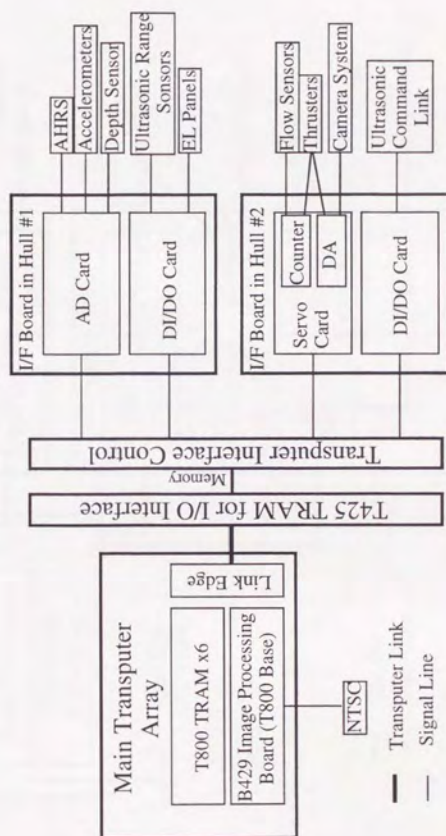


Fig. 2-9 Block diagram of computer system and hardware interface mounted on Twin-Burger 1.



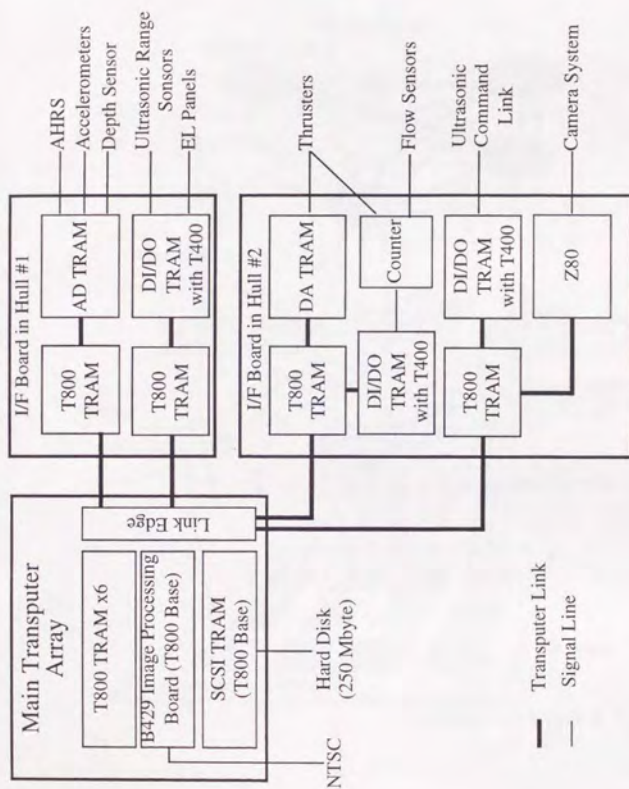


Fig. 2-10 Block diagram of computer system and hardware interface mounted on Twin-Burger 2.

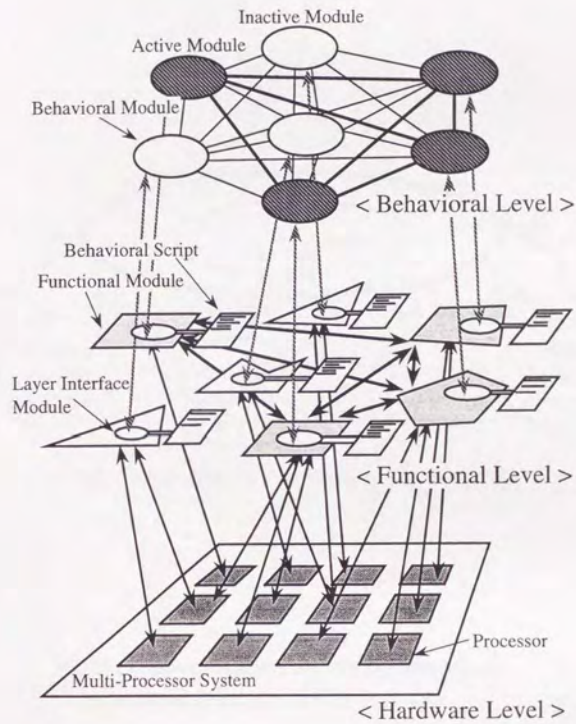


Fig. 2-11 Distributed Vehicle Management Architecture developed by Fujii [51].



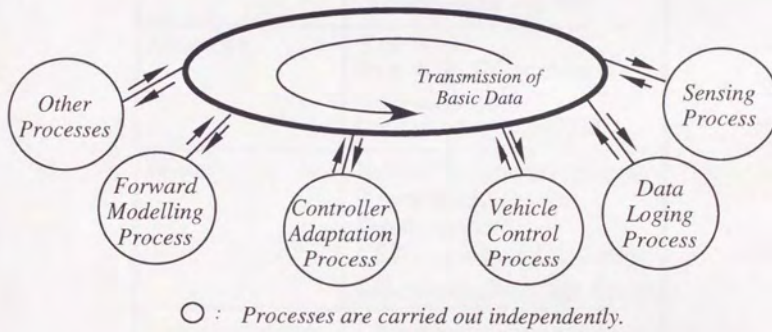


Fig. 2-12 Software architecture for Transputer based multiprocessor system of Twin-Burger.

Table. 2-1 Dimensions and specifications of Twin-Burger vehicles.

Length over all	1.54 m
Breadth over all	0.65m
Depth over all	0.50m
Dry weight	120kg
Operating depth	50m max.
Maximum speed	1knot
Duration	2hours (for CPUs)
Batteries	25.2V/10A and 28.8V/10A
Structure	Twin FRP hulls and a cylinder with a frame
Actuators	5 thrusters with 40W DC motors 2 servo motors for camera pan and tilt
Sensors	Inertial navigation system 2 axes speed sensor Depth sensor CCD color imaging sensor 8ch. ultrasonic range finder
Communication	Ultrasonic Command Link Electro-luminescent panels
Computer system	<b><u>Twin-Burger 1</u></b> < Main > 8x T800 TRAM with 1M RAM 1x T800 TRAM with 8M RAM 1x T800 TRAM for Image Proc. < Interface > 4x T425 TRAM with 1M RAM <b><u>Twin-Burger 2</u></b> 6x T800 TRAM with 1M RAM 1x T800 TRAM with SCSI 1x T800 TRAM for Image Proc. < Interface > 4x T800 TRAM with 1M RAM 3x T400 TRAM with DIO 1x AD TRAM 1x DA TRAM



Table 2-2a Parameters of thruster calculation curve of Twin-Burger 1.

		$a(= \times 10^3)$	$b(= \times 10^1)$	$c(= \times 10^{-5})$	$d(= \times 10^{-2})$	e
Thruster-L	Forward	1.414	5.797	7.687	1.051	1.268
	Backward	2.098	-12.75	-4.032	1.555	-0.942
Thruster-R	Forward	1.142	7.285	8.905	0.883	1.370
	Backward	2.012	-0.129	-4.854	1.608	-0.915
Thruster-H	Forward	2.133	5.574	7.604	0.287	1.376
	Backward	1.145	-7.681	-10.03	0.179	-1.343
Thruster-V1	Forward	1.130	-3.094	7.000	1.082	1.006
	Backward	2.039	-16.63	-4.838	1.251	-0.994
Thruster-V2	Forward	1.159	14.62	3.975	1.667	1.359
	Backward	2.173	-33.95	-5.162	1.159	-1.474

Table 2-2b Parameters of thruster calculation curve of Twin-Burger 2.

		$a(= \times 10^3)$	$b(= \times 10^1)$	$c(= \times 10^{-5})$	$d(= \times 10^{-2})$	e
Thruster-L	Forward	1.154	22.62	6.879	1.327	0.809
	Backward	2.056	-7.924	-5.681	1.327	-1.165
Thruster-R	Forward	1.108	27.73	9.279	1.439	1.126
	Backward	2.059	-0.319	-6.583	1.592	-1.238
Thruster-H	Forward	1.140	16.30	5.132	1.709	1.112
	Backward	1.993	-5.957	-4.395	1.595	-1.113
Thruster-V1	Forward	1.091	150.5	10.52	0.953	1.241
	Backward	1.981	-15.76	-7.435	1.028	-1.422
Thruster-V2	Forward	1.109	32.23	10.52	0.953	1.241
	Backward	2.015	-54.76	-7.435	1.028	-1.422

## 第3章

# ニューラルネットワークによる 海中ロボット運動の表現



### 3.1 はじめに

海中ロボットの自律航行を実現するには、第1章で述べたように、多くの確立すべき要素技術が存在する。本論文は、根幹の要素技術の1つである運動制御システムに関して取り扱う。運動制御システムを開発する場合、対象となるシステムの動特性を知る必要があることから、古くから実システムのモデルに関する研究がなされている[88-92]。実システムのモデルを構築するには、2つのアプローチがある。その1つは、物理・化学的自然法則に基づくシステム解析により、数理モデルを作成するモデリングと呼ばれる手法であり、他の1つは、与えられた入出力データに基づいてシステムの伝達関数を推定するブラックボックス的な手法で、システム同定と呼ばれている[90]。モデリングとは、対象システムの内部構造を記述するもので、対象の特性を熟知する必要がある、モデルの良し悪しは設計者の能力に大きく依存する。システム同定は、実システムの入出力データに基づいてシステムの外部特性である伝達関数を作成する手法であり、実システムと制御理論を繋げるための手法である。

システムの同定問題では、同定実験とデータ収集、モデル構造の選択、評価関数、推定アルゴリズム、モデルの低次元化など様々な要素を取り扱う。同定手法を分類すると、(i) ARXモデル等において予測誤差法や相関法を用いた時間領域における同定法、(ii) 周波数応答法等のノンパラメトリック同定法、(iii) 公称モデルとモデル化されないダイナミクスによる不確かさを評価し同定する手法等が存在する。特に、制御対象が不確かでも制御系の安定性を保つロバスト安定問題の取り扱いと関連して(iii)の手法が注目されている[92]。

海中ロボットの運動モデルを作成する場合は、制御対象の振舞いの不確かさを考慮する必要がある上に、搭載機器の変更による動特性の変化や、未知な動作環境における外乱にも対処可能な同定法が必要となる。本章では、モデル化が容易でない不確かな要素を含め、運動モデルを運動データから直接的に作成することを考え、ニューラルネットワーク[13,93]を用いた同定手法を提案する。動的システムのモデルとして、Fig. 3-1に示されるように、運動に関する状態量及び操作量入力に対して次の時間ステップの状態量を出力する差分方程式の構造を持つネットワークを検討し、海中ロボットの運動へ適用する。

### 3.2 ニューラルネットワークの基礎

#### 3.2.1 特徴

ニューラルネットワークには、連想記憶やパーセプトロン、ボルツマンマシン等の形態が存在するが[93]、本研究では一般的にニューラルネットワークと言われている階層型のフィードフォワードネットワークをベースとして動的なシステムの表現を試みる。階層型のフィードフォワードネットワークの特徴として、

- ・ シグモイド関数を入出力関数として採用しているため、非線形な応答が可能である。
- ・ 学習能力を有する。
- ・ 知識はパターンの静的なコピーではなく、ユニット間の結合荷重として蓄積する。
- ・ 補間能力を有する。

等があり、問題点として、

- ・ 出力の精度と信頼性。
- ・ ネットワークの構造の定式化が困難。
- ・ 追加学習による過去の知識への影響の取り扱いが難しい。
- ・ 学習は経験的であり、安定性の議論が不十分である。

等の問題点から、ブラックボックス的と評価される。しかしながら、上記の特徴からも分かるように未知の可能性を秘めており、非線形性が強い動的なシステムへの適用が期待されている。

#### 3.2.2 ネットワークモデルの基本的記述

ネットワークの構成要素である一つのユニットを Fig. 3-2 に示す。他のユニットからの入力荷重和により、ユニット  $i$  の活性値  $x_i$  が定まる。この活性値が、 $\pm\infty$  で飽和値をとる単調増加で微分可能である、シグモイド形出力関数  $g(x)$  により変換され、ユニット  $i$  での出力値  $y_i$  が得られる。本文ではシグモイド形状関数のなかでも Fig. 3-3 に示すロジスティック関数

$$g(x) = \frac{1}{1 + e^{-x}} \quad (3.1)$$

を用いた。この入出力関係を定式化すると

$$x_i = \sum_j w_{ji} y_j + \theta_i \quad (3.2)$$

$$y_i = g(x_i) \quad (3.3)$$

ただし、 $w_{ji}$  はユニット  $j$  からユニット  $i$  への結合荷重、 $\theta_i$  はユニット  $i$  のしきい値である。

このようなユニットから構成された3層のネットワークの構造を Fig. 3-4 に示す。入力層と出力層が外界と接しているのに対し、中間層は外界に接していない。



このネットワークに教師データを与え、式(3.2), (3.3)に示す活性度伝播により出力パターンを求める。なお、入力層においては入出力関数を  $g(x)=x$  とし、入力値がユニット出力値となる。

### 3. 2. 4 誤差逆伝播学習法

フィードフォワード型ニューラルネットワークの学習において、一般的に用いられている手法として誤差逆伝播学習法 (Back-Propagaton Method) がある。誤差逆伝播学習法は最急降下法の一つであり、評価関数のエネルギーを減少させる方向に結合荷重のベクトルを移動させることにより、極小値を探索する手法である。以下に誤差逆伝播学習法の学習アルゴリズムについて述べる。

ここでの学習とは入力パターンをネットワークに加え、活性度伝播により計算した出力パターンとその目標出力パターンができるだけ一致するよう、すなわち下記の自乗出力誤差和  $E$  が最小になるよう結合荷重を調整することである。

$$E = \sum_k E_k = \frac{1}{2} \sum_k \sum_i (y_i^{(k)} - y_i^{d(k)})^2 \quad (3.4)$$

ただし、 $E_k$  は  $k$  番目の訓練パターンに対する自乗出力誤差和、 $y_i^{(k)}$  は訓練パターン  $k$  に対する出力ユニット  $i$  の出力値、 $y_i^{d(k)}$  はその目標出力値である。

バックプロパゲーションは、基本的には最急降下法により結合荷重を修正する。結合荷重  $w_{ij}$  に関する評価  $E$  の偏微分を計算し、次式のように偏微分値に比例して結合荷重を修正する。

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} \quad (3.5)$$

ただし、 $\Delta w_{ij}$  は結合荷重の修正量、 $\eta$  は学習係数である。上式の偏微分は合成関数の偏微分により、次のように表現できる。

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial x_i} \frac{\partial x_i}{\partial w_{ij}} \quad (3.6)$$

ここで  $x_i$  はユニット  $i$  の活性値である。式(3.6)の右辺第2項は式(3.2)より

$$\frac{\partial x_i}{\partial w_{ij}} = y_j \quad (3.7)$$

ただし上式での  $y_j$  は、 $x_i$  の一つ下位階層のユニット  $j$  の出力値である。

次に、式(3.6)の右辺第一項を、

$$\delta_i = - \frac{\partial E}{\partial x_i} \quad (3.8)$$

とおく。この  $\delta_i$  をユニット  $i$  の誤差と呼び、ユニット  $i$  の活性値に対する自乗出力

誤差和の感度を表す。ここで、式(3.8)をさらに変形して、

$$\delta_i = - \frac{\partial E}{\partial y_i} \frac{\partial y_i}{\partial x_i} \quad (3.9)$$

ここで、 $y_i$  は活性値  $x_i$  に対応する出力値である。式(3.9)の右辺第2項は式(3.3)より、

$$\frac{y_i}{x_i} = g'(x_i) \quad (3.10)$$

である。ここでロジスティック関数の微分  $g'$  は次式で与えられる。

$$g'(x_i) = g(x_i) [1 - g(x_i)] \quad (3.11)$$

式(3.9)の右辺第一項は、出力値  $y_i$  が出力ユニットの出力値の場合と、隠れユニットの出力値の場合とで、分けて考える必要がある。 $y_i$  が出力ユニットの場合では、

$$\begin{aligned} - \frac{\partial E}{\partial y_i} &= \frac{1}{2} \frac{\partial \Sigma (y_i - y_i^d)^2}{\partial y_i} \\ &= (y_i^d - y_i) \end{aligned} \quad (3.12)$$

従って、出力ユニットの誤差  $\delta_i$  は式(3.9),(3.10),(3.12)より、

$$\delta_i = (y_i^d - y_i) g'(x_i) \quad (3.13)$$

次に出力値  $y_i$  が隠れユニットの出力値の場合には、一つ上位階層のユニット活性値  $x_j$  および、誤差  $\delta_j$  がすでに計算済みとする、

$$\frac{\partial E}{\partial y_i} = - \Sigma_j \frac{\partial E}{\partial x_j} \frac{\partial x_j}{\partial y_i} \quad (3.14)$$

ここで、右辺第一項は定義により  $\delta_j$  であり、右辺第2項は式(3.2)より  $w_{ij}$  であるので、

$$\frac{\partial E}{\partial y_i} = - \Sigma_j \delta_j w_{ij} \quad (3.15)$$

式(3.9),(3.10),(3.15)より、隠れユニットの誤差  $\delta_i$  は次式で表される。

$$\delta_i = g'(x_i) \Sigma_j \delta_j w_{ij} \quad (3.16)$$

すなわち、上位階層の誤差  $\delta_j$  を結合  $w_{ij}$  を介して当該階層に逆伝播させて得られる  $\Sigma_j \delta_j w_{ij}$  に当該階層の  $g'(x)$  をかけることにより、隠れユニットの誤差  $\delta_i$  を求められる。

以上をまとめると、結合荷重の修正量は次式で表現できる。

$$\Delta w_{ij} = \eta \delta_i y_j \quad (3.17)$$



式(3.16)から分かるように、出力層での誤差を結合を經由して、逆方向に伝播し下層での誤差を逐次求めるのが、バックプロパゲーションの特徴である。

なお、しきい値 $\theta$ の学習についてはふれていない。式(3.2)より、あるユニット $i$ のしきい値は、常に値1を取る仮想的ユニットからの結合荷重と考えることができるので、結合荷重と同様に学習が可能である。

実際の学習では、式(3.16)の代わりに、

$$\Delta w_{ij}(n) = \eta \delta_j + \alpha \Delta w_{ij}(n-1) \quad (3.18)$$

を用いる。ここで、 $\Delta w_{ij}(n-1)$ は一回前の結合荷重修正量、 $\alpha$ はモーメントムと呼ばれる0と1の間の数である。上式の右辺第2項は慣性項に相当する。例えば峡谷状の誤差局面の場合、式(3.17)を用いて結合荷重を修正すると、峡谷に沿った方向への速度が極めて小さくなることもある。このような場合、式(3.18)のように慣性項を導入すると、学習速度を大きくできる。

学習係数 $\eta$ は、小さすぎると学習に長時間を要して非効率である。逆に大きすぎると振動してしまいうまく学習できない。一般には振動しない範囲でできるだけ大きな学習係数を用いるとよいが、これには試行錯誤が必要である。

式(3.17)から分かるように、結合の始点ユニットの出力値が零であると、結合荷重の修正量が零になり、学習が行われぬ。注意すべきことは、3層以上のネットワークの場合、すべての結合に0や1などの同じ初期値を与えると、正しく学習できないことがある。なぜならば式(3.16)から推測されるように、各出力ユニットに入る結合荷重は常に等しく、また各入力ユニットから出る結合荷重も常に等しくなるからである。

### 3.2.4 動的システムのモデル

静的なシステムの同定に適用する場合、入力層から出力層への関係は静的であり、教示データの提示順序に本質的な意味はない。学習において、訓練パターン毎に結合荷重を修正する場合には、提示順序が若干の影響を与えるが、十分な学習の後にはその影響は小さくなる。他方、動的なシステムを取り扱う場合、教示データは時系列データであり、学習において訓練パターンの提示順序自体に意味がある。従って、動的なシステムを表現する場合には、過去の履歴を何らかの形でネットワーク内に保持することが必要である。過去の履歴の保持するネットワークとしてジョルダンネットワークとエルマンネットワークが提案されており、ここではこれらについて述べる[93]。

#### 1) ジョルダンネットワーク

ジョルダンネットワークは、入力層・中間層・出力層のほかに、回帰的結合を介して出力値を入力値とする状態層を有する (Fig. 3-5 参照)。この状態層の入力

値は、出力  $y_i$  の履歴として式 (3.19) により表現される。

$$y_i = y_{i-1} + dy_{i-2} + d^2y_{i-3} + d^3y_{i-4} + \dots \quad (3.19)$$

ただし、 $d$  は 1 時点前の状態の残存率である。このような状態層を用いることにより、過去の履歴を反映した系列の学習が可能である。適切な残存率を用いれば、過去の履歴を反映した状態層の活性値の違いにより、それぞれの後続系列を適切に学習できる。

## 2) エルマンネットワーク

エルマンネットワークは、Fig.3-6 に示すように 1 時点遅れの間層の出力を状態層として用いる。ジョルダンネットワークでは、学習が進展すると出力ベクトルが目標出力ベクトルに近づくため、状態層は目標出力情報を反映したものとなる。他方、中間層は一般に入力層と出力層の双方を反映し、学習が進展するにつれて中間層は過去の履歴を十分反映し得るため、エルマンネットワークの状態層は入出力の履歴を十分反映したものとなる。



### 3.3 運動の表現

ニューラルネットワークによる海中ロボット運動の表現手法として、前節で述べたジョルダン型、或いはエルマン型のネットワークを用いた手法が考えられる。藤井らは文献[48,64]において、航行型海中ロボットの縦運動の同定を行ない、両者を比較した結果、エルマン型のネットワークを採用した運動モデルの方が精度良く同定できることを確認している。この運動モデルは、制御系の調整を目的として考案されたもので、実験から得られた状態量と制御信号の入力に対して、制御結果となる状態量を出力する。つまり、運動の評価を行うためには、実験から得られる一連の状態量の時系列が必要となる。ここでは、エルマン型のネットワークをベースとし、制御系調整だけでなく海中ロボットの代替となる運動モデル、つまり制御信号の時系列から運動シミュレーションが行なえる新たな運動モデルの構築法を提案する。

#### 3.3.1 動的システムの同定

運動モデルを用いて海中ロボットの運動をシミュレーションするには、操作入力の時系列と状態量の初期値から状態量の時系列を生成する必要がある。つまり、運動モデルにおいて、入力となる状態量を自身の出力から得ることができれば、制御信号の時系列から状態量の時系列を出力することが可能である。本論文では、Fig. 3-7に示す構造の運動モデル（以後、Identification Networkと呼ぶ）を提案する[94'96]。

Identification Network は、シグモイド関数を入出力関数として持つニューロンよりなる3つの層と積分層、及び出力層及び中間層から入力層への2種類の回帰的結合(RC\_1, RC\_2)から構成されている。出力層から入力層へのリカレントRC\_1から入力となる状態量を回帰的に取得し、操作量及び得られた状態量の時系列から対象とする動的システムのシミュレーションを行える構造となっている。中間層から入力層へのリカレントRC\_2は、入力変数の簡素化を行なうと同時に、過去の時系列の影響をモデル内に蓄えるために導入している。出力は1時間ステップ後の状態量であり、制御対象となる状態量として使用することを念頭に置いている。また、学習の収束を速め、動的な教示データの写像関係を良く表現するために、差分型ネットワーク[68,70]の構造を採用している。差分型ネットワークは、出力層において入力層に入力された状態量の高次差分を出力するネットワークである。通常のネットワークを用いて動的なシステムのモデルを表現する場合、入力値と出力値の差が小さいため恒等関数に近い写像を学習することになり、学習の精度や時間において問題が存在する[70]。出力値として高次の差分値を学習することにより、同定モデルの精度が向上し、動的なシステムの高次の振舞いまで表現することが可能となる。従って、入力及び運動の評価対象となる状態量を $\Delta^2 Sf(k) (= \Delta Sf(k+\Delta k) - \Delta Sf(k))$ 、 $\Delta Sf(k) (= Sf(k+\Delta k) - Sf(k))$ 、 $Sf(k)$ とすると、第

3層における出力値は $\Delta^2 Sf(k)$ となる。Identification Networkは状態方程式をニューラルネットワークで表現することを念頭において設計した運動モデルである。差分型ネットワークを導入した場合としない場合の比較は、次節において実際に海中ロボットに適用した例を挙げて比較する。

Identification Networkの順方向の計算は、 $Sf(k)$ の時系列を得るために以下の式に基づいて行われる。ネットワークの写像関数を $f$ とすれば、第3層の出力は(3.20)式で与えられる。

$$\Delta^2 Sf(k+\Delta k) = f(\Delta Sf(k), \Delta Sf(k-\Delta k), \dots, u(k), u(k-\Delta k), \dots) \quad (3.20)$$

次に、各種分層において(3.21)、(3.22)式に基づいて $Sf(k)$ の時系列を生成する。

$$\Delta Sf(k+\Delta k) = \Delta Sf(k) + \Delta^2 Sf(k) \quad (3.21)$$

$$Sf(k+\Delta k) = Sf(k) + \Delta Sf(k) \quad (3.22)$$

### 3.3.2 学習手順

効率よく学習を行うため、(3.20)～(3.22)式を念頭に置き学習段階を分割することを考える。Identification Networkにおける教示データの入出力関係を「入力量, 出力量」で表すものとする。時間 $k$ において「 $[\Delta Sf(k) u(k)]$ ,  $[\Delta^2 Sf(k+\Delta k)]$ 」,  $[\Delta Sf(k) u(k)]$ ,  $[\Delta Sf(k+\Delta k)]$  及び「 $[\Delta Sf(k) u(k)]$ ,  $[Sf(k+\Delta k)]$ 」が存在する。これらの写像関係を同時に学習した場合、リカレントの影響で学習の収束が遅くなる。また、入力として1時間ステップ前の出力を用いるために、入力値に含まれる誤差の影響による学習の発散が起こる。そこで、Fig. 3-8に示すように学習段階を4分割した学習方法を用いる。以下では、実験から得られた状態量を $\Delta^2 Se$ ,  $\Delta Se$  及び  $Se$ 、学習における重み行列を $M_i$ 、誤差評価関数を $Ef$ とし、誤差逆伝播法を用いて学習を行うものとする。

第1段階：入力は $\Delta Se(k)$  及び  $u(k)$  とし、評価される出力は $\Delta^2 Sf(k)$  とする。 $Ef$ は(3.23)式で表す。

$$Ef = 1/2 \sum_k \{(\Delta^2 Sf(k) - \Delta^2 Se(k))^T \cdot M_1 \cdot (\Delta^2 Sf(k) - \Delta^2 Se(k))\} \quad (3.23)$$

第2段階：入力は $\Delta Sf(k)$  及び  $u(k)$  とし、評価される出力は $\Delta^2 Sf(k+\Delta k)$  とする。 $Ef$ は第1段階と同様に(3.23)式で表す。

第3段階：入力は $\Delta Sf(k)$  及び  $u(k)$  とし、評価される出力は $\Delta Sf(k+\Delta k)$  とする。 $Ef$ を(3.24)式で表す。

$$Ef = 1/2 \sum_k \{(\Delta Sf(k) - \Delta Se(k))^T \cdot M_3 \cdot (\Delta Sf(k) - \Delta Se(k))\} \quad (3.24)$$



第4段階: 入力は $\Delta Sf(k)$ 及び $u(k)$ とし, 評価される出力は $Sf(k+\Delta k)$ とする。 $Ef$ を(3.25)式で表す。

$$Ef = 1/2 \sum_k \{ (Sf(k) - Se(k))^T \cdot M_4 \cdot (Sf(k) - Se(k)) \} \quad (3.25)$$

第1段階の学習は、RC\_1が結合されていないネットワークの学習に対応し、制御対象の $\Delta^2 Sf$ 、 $\Delta Sf$ 及び $u$ の関係を学習する。第1段階における学習が充分収束した後、第2段階の学習に移行する。これは、ある程度学習を行なったネットワークを用いることにより、入力値に含まれる誤差に起因する学習の発散を防ぐためである。第2段階では、入力となる状態量をリカレントRC\_1から得られる値を用いて運動モデル内に閉ループを構成し、 $\Delta^2 Sf$ 、 $\Delta Sf$ 及び $u$ の関係を学習する。この学習により、ネットワークの写像の精度を向上させると共に、実験から得られたデータの近傍において入力領域が存在することになり、より安定した運動モデルが得られる。第3段階において出力 $\Delta Sf$ を実験値と一致させ、第4段階では $Sf$ を一致させるように学習することにより、 $Sf$ 及びその1次、2次の微係数を同時に精度良く表現することができる。

### 3. 4 海中ロボットへの適用

Identification Network を海中ロボットの運動同定に適用し、提案したモデルの有効性を検討する。海中ロボットの例として、2章で述べた航行型海中ロボットのテストベッドであるPW45、非航行型海中ロボットのテストベッドTwin-burger、及び最近開発されたR-one Robot[98]を考え、実験結果をもとに運動の同定を行う。さらに、差分型ネットワークによる精度向上を検証するため、差分型ネットワークを採用しない場合との比較、前節において提案した学習過程を経ることにより有効な同定モデルが得られること、及び最小2乗法により求めたパラメータを用いて行なったシミュレーションとニューラルネットワークによるシミュレーションの結果を比較する。

#### 3. 4. 1 PW45の縦運動への適用

##### 1) 同定モデル

PW45は運動制御システムを開発するためのテストベッドであり、胴体左右に取り付けられたエレベータを用いて縦運動及びロール度の制御を行うことが可能である。Fig. 3-9に示すように、状態量としてピッチ角 $\theta$ 及び深度 $D$ を取り、制御信号はエレベータ切り角 $\delta_e$ である。ここでは、縦運動について考え、推力は一定として前進速度については無視する。

PW45の動特性に関するデータを得るために、周波数応答実験を行った。実験では、最初の4.4秒間でロボットを安定な状態にするために目標深度1.0mになるように制御し、それ以後はエレベータの角度を周期的に正弦波状に変えることにより30秒間の運動データを得た。データのサンプリングレートは20Hzである。

縦運動を表現するためのネットワーク構造をFig. 3-10に示す。入力変数はエレベータ切り角 $\delta_e$ 、ピッチ角 $\theta$ 及び深度 $D$ とし、出力は第3層において、1時間ステップ後の状態量であるピッチ角速度 $\Delta\theta$ と深度変化速度 $\Delta D$ であり、積分を経てピッチ角と深度を出力する。中間層のニューロン数は10とした。また、中間層からの回帰的結合RC<sub>2</sub>により、中間層にある全てのニューロンからの出力を入力層に回帰させている。また、各ニューロンの応答関数としてシグモイド関数を用いているため、出力は全て[0, 1]の範囲になる。更に、学習の収束を進めるために、入力の範囲を同程度にすることが望ましい。従って本論文では、入出力を[0, 1]の範囲に正規化する。正規化した変数は $\hat{\cdot}$ を付けて表し、式(3.26)に基づいて行なう。

$$\hat{S} = (S - S_0) / S^* + 0.5 \quad (3.25)$$

ロボットの運動を考えて次のようにする。ここで、 $\theta^* = 0.5 \text{ rad}$ 、 $D^* = 1.0 \text{ m}$ 、 $\delta_e^* = 0.5 \text{ rad}$ 、 $\Delta\theta^* = 0.25 \text{ rad}$ 、 $\Delta D^* = 0.5 \text{ m}$ 、 $D_0$ : 4.4秒後の深度とする。正規化を行うことで各ニューロンの入出力を同等に比較検討できる事が可能となる。



## 2) 学習

ここでは、周波数応答実験においてエレベータ角を  $\delta_e = 0.1 \sin(\pi t)$  rad (振幅 0.1rad, 0.5Hz) としてを与えた場合の実験結果を教示データとした。誤差逆伝播法では、誤差の量と学習定数に比例して結合加重を調整するので、学習定数  $\eta$  を大きく取ると誤差が振動が発生し、それ以上収束しない。逆に、学習定数を小さくすると極値にとらえられ抜け出すことが困難である。従って、学習段階毎に学習係数を変化させ、全体として15万回の学習を行った。

学習した時系列データとIdentification Network及び(3.27)式に示すARXモデルによるシミュレーション結果の比較がFig. 3-11であり、上からエレベータ切り角、ピッチ角、深度を示している。Identification Networkの学習段階における重み行列は全て単位行列とし、ARXモデルの各パラメータは最小二乗法により求めている。同定の精度を(3.25)式に基づいて評価したものがTable 3-1である。図から分かるように、Identification Networkは与えられた時系列データの写像関係を良く学習しているおり、ARXモデルと比較しても十分な精度の運動モデルが得られている。0.5Hzで $\delta_e$ に制御信号を与えた場合の運動特性がネットワークに内包されていることがわかる。

$$\theta(k) + a_{11} \cdot \theta(k-1) + a_{12} \cdot D(k-1) = b_1 \cdot \delta_e \quad (3.27a)$$

$$D(k) + a_{21} \cdot \theta(k-1) + a_{22} \cdot D(k-1) = b_2 \cdot \delta_e \quad (3.27b)$$

ここで

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} -9.976 \times 10^{-1} & 1.399 \times 10^0 \\ -6.632 \times 10^{-3} & -9.958 \times 10^{-1} \end{bmatrix}$$

$$B = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} 1.222 \times 10^{-2} \\ -2.974 \times 10^{-2} \end{bmatrix}$$

## 3) 有効性の検討

得られたネットワークの有効性を検討する。ここでは、制御量 $\delta_e$ に学習していない周波数の入力を与え、実験データとの比較を行う。

## ・低周波特性

振幅0.1rad、0.1Hzの制御信号を与えた場合のシミュレーション結果をFig. 3-12に示す。ピッチ角については、0.04rad程度の定常偏差が残っているものの、良く表現できている。偏差は、初期条件の誤差により発生している。深度については、ネットワークが制御信号に対して応答量が小さい場合について学習しているため、誤差が生じている。

## ・高周波特性

振幅0.1rad、1.0Hzの制御信号を与えた場合のシミュレーションをFig. 3-13に示す。高周波の信号に対しては、追従できている。実験結果と比較すると、シミュレーションでは上向きにピッチしており、0.05rad程度の定常偏差が残っている。これに伴い、深度が浅くなっている。

## ・複合波

Figs. 3-14、3-15は、次に示す制御信号を与えてシミュレーションしたものである。

$$\delta_c = 0.2 \sin(2\pi \cdot 1.0 \cdot t) + 0.2 \sin(2\pi \cdot 0.25 \cdot t) \quad (3.28a)$$

$$\delta_c = 0.3 \sin(2\pi \cdot 1.0 \cdot t) + 0.1 \sin(2\pi \cdot 0.25 \cdot t) \quad (3.28b)$$

これらのシミュレーションから分かるように、 $\delta_c$ に複合波を入力した場合でもピッチ運動に関しては、実験結果と良く一致している。深度に関しては、教示データの特性を受け継いでおり全域にわたり応答が小さい。ピッチ角と深度における学習精度の違いは、操作量 $\delta_c$ に対する影響の違いにあると思われる。ロボットの運動を決定するのはピッチ角 $\theta$ やピッチ角速度 $\Delta\theta$ 、深度変化 $\Delta D$ であり、深度自体はさほど意味を持たない。差分型のネットワークを適用しているため、深度Dにおいても学習した周波数では表現できているが、他の周波数領域に対する汎化能力が小さい。これは、入力する状態量を再検討することでより汎化能力の高いモデルが得られると思われる。

## ・追加学習

得られた結合荷重に、振幅0.1 rad、周波数0.1 Hzの制御信号を与えた時系列データの追加学習を行う。ニューラルネットワークでは、過去に学習したことを忘れるという問題が存在する。ここでは単純に、振幅0.1rad、周波数0.5Hzの時系列データの追加学習を同じ回数行う。追加学習は約10万回行った。

Fig. 3-16は、追加学習以前と以後のネットワークの周波数応答を示している。この図より、単一周波数に対する応答を学習した最初のネットワークに比べ、追加学習を行うことでより低周波数の応答を学習したネットワークの方が、全体的な運動特性を平均的に表している。しかし、0.5Hzでの精度は下がっている。これは、追加学習を行ったことにより過去に学習した0.5Hzにおける知識が曖昧になったことに起因する。

## 4) 差分型ネットワークによる精度向上

差分型ネットワーク構造を導入したネットワークと導入していないネットワークに対して同じ教示データを与え、その効果の確認を行う。ここでは、教示データとしてエレベータを振幅0.1 rad、周波数0.1 Hzでトリムさせた場合の実験データを用いる。これは、操作量に対する状態量の応答が大きい教示データを採用した方がネットワークを評価しやすく、ネットワークの学習領域[83]が大きくなる



為である。差分型ネットワーク構築には約10万回、非差分型ネットワーク構築には試行錯誤を含めて約100万回の学習を行った。Fig. 3-17に学習した0.1Hzのデータに対する出力結果の比較を示す。差分型のネットワークの方が滑らかであり、誤差が少ないことが解る。学習した周波数より高い1.0Hzの教示データに対する応答の比較をFig. 3-18に示す。非差分型のネットワークではピッチ角に偏差が生じており、深度に関しても約15秒で一定出力となり有効な値を出力をしていない。一方、差分型のネットワークでは、ピッチ角の応答が実験値に比べて大きいものの、実験値をある程度追従できている。両者の周波数応答の比較をFig. 3-19に、誤差の比較をTable 3-2に示す。差分型のネットワークが動的なシステムの同定に有効であることが解る。

### 3. 4. 2 Twin-Burger への適用

非航行型海中ロボットのテストベッドであるTwin-Burgerの運動モデルの構築について検討する。Twin-Burgerの形状は、2つの压力容器とシリンダからなるオープンフレーム構造であり、航行型の海中ロボットとは違い流体力学的な考慮は成されていない。また、スラストにより直接的に運動を制御できるホロノミック[99]なロボットであり、複雑な経路も単純な1自由度の運動の組み合わせにより表現することが可能である。ここでは最初に、安定して絶対角及び角速度が得られるYaw運動について取り扱い、その学習過程及び運動モデルの精度について考察する。次に、Surge、Sway、Heaveに関する運動について各々独立のネットワークを構築する。

#### 1) Yaw 運動の同定

学習過程の有効性を含め、運動モデルの構築手法について検討するため、最初に取り扱いが簡単なYaw運動について考える。取り扱う変数をFig. 3-20に示す。PW45の運動モデルの作成には周波数応答実験から得られたデータを用いたが、その場合、教示データの周波数成分が支配的になり、他の周波数帯における汎化能力に問題がみられた。運動の同定を行なう場合、入力信号としてM系列[88]の信号を取り扱うことも考えられるが、実験の都合上、方位角に関するリミットサイクルを取り扱う。得られた実験結果を教示データとして学習することにより運動モデルを構成する。

運動モデルの構造をFig. 3-21に示す。操作量  $u(t)$  と回帰的結合 RC\_1、RC\_2 からの入力に対し、第3層で得られる角加速度  $\Delta^2\psi(t)$  を2つの積分層において積分することにより、評価値となる状態量  $(\psi(t), \Delta\psi(t))$  を出力する構造とした。実験結果と同定モデルの学習段階における出力の比較をFig. 3-22に示す。上から入力信号  $u$ 、角加速度  $\Delta^2\psi$ 、角速度  $\Delta\psi$ 、方位角  $\psi$  を表しており、点線が実験結果、細線が第2段階の学習を終えた後の同定モデルの出力、太線が第4段階の学習を終

えた後の最終的な出力を表している。一点鎖線は、海中ロボットの最も簡単な運動方程式として、慣性と速度 ( $=\dot{x}_i$ ) の2乗に比例する抗力のみを考慮した、

$$M_i \cdot \ddot{x}_i + C_i \cdot \dot{x}_i \cdot |\dot{x}_i| = F_i \quad (3.29)$$

の形の運動方程式[100]を用い ( $i$ は運動モード)、最小二乗法により算出したハラメータを用いてシミュレーションした結果であり、Identification Network による同定結果と比較するために示している。評価関数における重み行列は全て単位行列とし、PW45 の同定と同様、式 (3.26) を用いて入力変数を正規化している ( $\Delta^2\psi^*=1.0 [rad/s^2], \Delta\psi^*=1.0 [rad/s], \psi^*=1.0 [rad], M^*=10.0 [Nm]$ )。Fig. 3-22 から、運動モデルの出力が学習段階が進むにつれて実験結果に一致していく様子が分かる。教示データに雑音が含まれているにも係わらず、運動特性を良く抽出している。更に、第4段階を学習に含めることにより、 $\Delta\psi(t)$ だけでなく $\psi(t)$ を実験値に一致させることに成功している。(3.29) 式の運動方程式のハラメータは、抗力に相当する項を操作信号と同じ入力値、つまり外力として考え、方位角速度と制御信号の関係から ARX モデルを用いて最小二乗法により同定した。評価値として角速度の時系列を用いているため、第2段階の運動モデルの出力とほぼ一致している。しかし、方位角の評価が含まれていないため、第4段階後の Identification Network の出力と比較すると方位角において同定の精度が低い。

以上の結果から、運動モデルとして Identification Network を導入し、提案した学習法に基づいて運動を同定することによりシミュレータができることが示された。Identification Network により、制御対象から得られる状態量と同等の出力を得ることが可能である。

## 2) Surge, Sway, Heave 運動の同定

Twin-Burger は4自由度の運動を前提としており、ここでは残りの Surge, Sway, Heave に関する運動の同定を行う。更に、前述した (3.29) の運動方程式の結果と Identification Network による同定結果と比較する。運動モデルの構造は、Yaw 運動と同様に、Fig. 3-21 に示される構造とした。学習は、提案する学習手順に従って進めるものとする。各運動モードの (3.29) 式の運動方程式のハラメータを Table. 3-3 に、各運動を Identification Network を用いて同定した結果の比較を Figs. 3-23~25 に示す。提案した同定モデルと実験データは全ての運動モードにおいて良く一致しており、十分な精度を有する同定モデルが得られていることが分かる。特に、Heave 運動の教示データは、フィルタ後の値であるにも係わらず雑音が多く含まれている。雑音を含む教示データから運動特性を抽出することに成功している。一方、運動方程式の同定は、フィルタの仕方に大きく影響され、設計者のセンスに依るところが大きい。Fig. 3-23~25 における運動方程式の挙動は、非線形方程式を線形方程式の形に変形して同定した事にも起因して、さほど良い結果



は得られていない。

以上の結果から、ニューラルネットワークを用いた海中ロボットの運動表現が有効であることが分かる。現在は、各運動モード間の干渉を外乱と捉えて運動制御システムの開発を行っているが、今後、複数の状態量入力への展開を検討すべきである。

### 3. 4. 3 R-one の海面におけるYaw運動への適用

R-one Robot[98]は、インターリッジ計画[6]におけるプロジェクトの1つとして計画された潜水艇であり、24時間の航行を可能にするためCCDE (Closed Cycle Diesel Engine) を搭載した全長8 [m]、乾燥重量4 [ton]の実用機である。ここでは、海面における方位角制御への適用を考えて、R-one Robotの方位角運動の同定モデルを作成する。

ここではYaw運動に関係する状態量として角速度、入力信号としてラダー角とメインスラストを取り扱う。Yaw運動には、前進速度も取り扱う必要があるが、ここではスラスト指令値を前進速度の代わりに用いる。従って、入力変数はYaw角速度、ラダー切り角、スラスト指令値とし、角加速度を出力するFig. 3-26に示すネットワークの構築を行う。また、中間層のニューロン数は5とした。

教示データとして、スラスト指令値が120、150、180における $z$ 試験データを用いて学習を進めた。Fig. 3-27はスラスト指令値が120の場合の同定結果であり、Fig. 3-28はスラスト指令値150、Fig. 3-29はスラスト指令値が180の場合の結果である。スラスト指令値による、ラダー切り角の効果の変化特性を表現できており、良好な結果が得られている。

### 3. 5 考察

以上、具体的な海中ロボットの運動表現にニューラルネットワークを用いた手法の導入を行い、Identification Network 及びその学習手順を用いた海中ロボットの運動モデルを提案した。提案した運動モデルは、状態方程式をニューラルネットワークにより構築することを念頭においた差分型の構造を有している。更に、制御信号の時系列から対象システムの運動シミュレーションを行なうため、出力層から入力層への回帰的結合を備えている。また、提案した構造の運動モデルは、入力として回帰的に得られる状態量に誤差が含まれることにも起因して、単純に教示データを学習するだけでは構築が困難である。そのため、簡単なことから学習を進めるように学習の段階を4つに分けている。

実際に海中ロボットに適用し得られた結果は、海中ロボットの運動モデルとして十分満足できるものであり、同定モデルの1つとして十分に機能することを示している。ARXモデルや運動方程式による同定結果と比較すると、実験データに存在する非対称な動特性に対しても表現できており、数式化が困難な要素に関しても表現できる可能性を示している。

問題点として、

- ・ 学習係数及び回数の決定手法がなく設計者のセンスに依るところが多い
- ・ 学習した領域以外での挙動が補償が困難である
- ・ どのクラスのシステムまで表現できるか

等が存在し、これらの課題について、実践に基づいた解析を行ってきた。これらに対し、既存の手法を用いて表現困難な要素をニューラルネットワークを用いて学習する等の方法も考えられる。今後、ニューラルネットワークを用いた運動モデルが、非線形な要素の取り扱いを含め、どの程度表現できるか解析する必要がある。



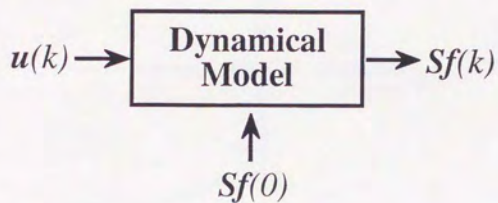


Fig. 3-1 Concept of identification with neural network.

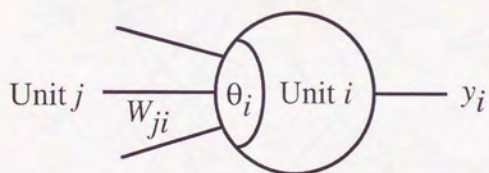


Fig. 3-2 A unit of network.

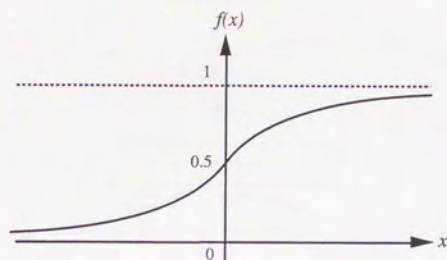


Fig. 3-3 Logistic function.

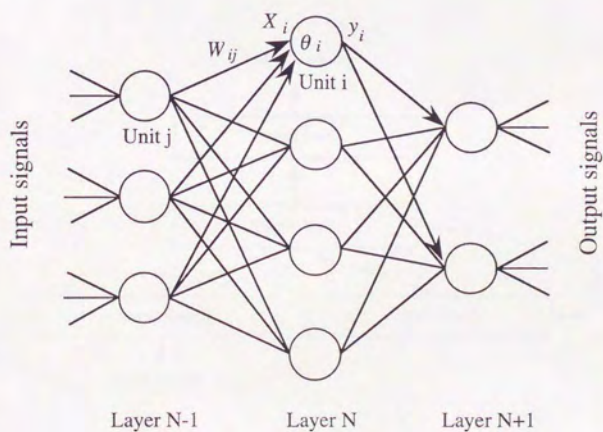


Fig. 3-4 Fundamental structure of feedforward neural network.



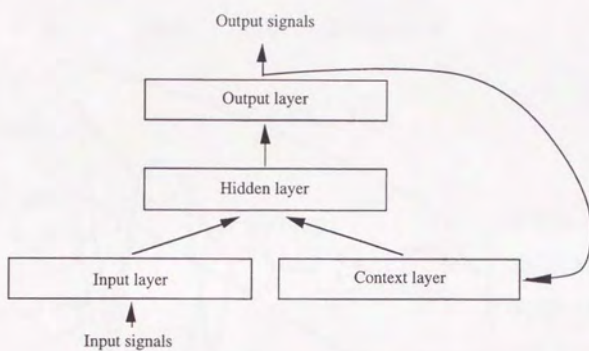


Fig. 3-5 Jordan network.

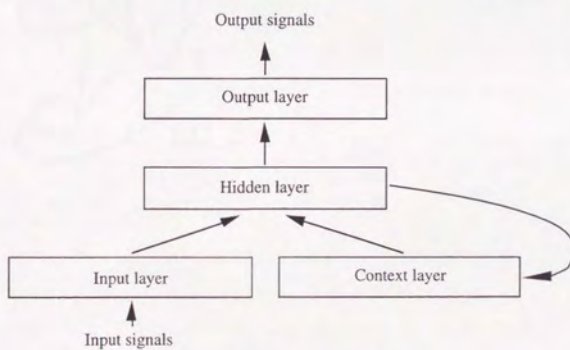


Fig. 3-6 Elman network.

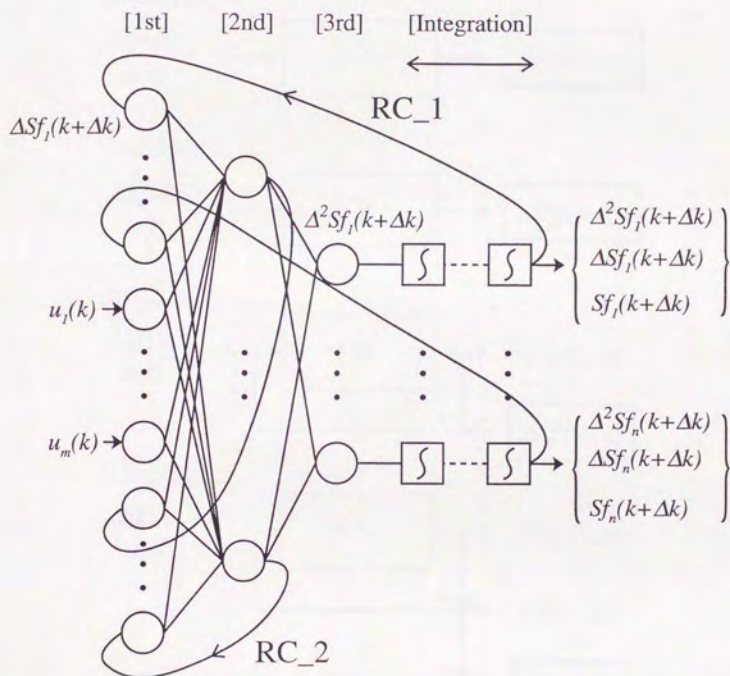
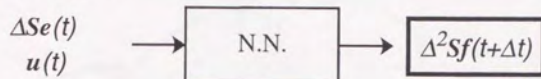


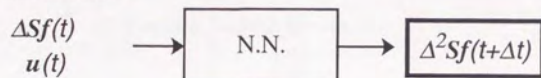
Fig. 3-7 Structure of Identification Network.



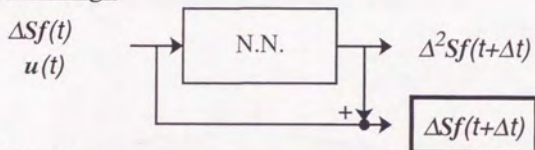
1st Stage



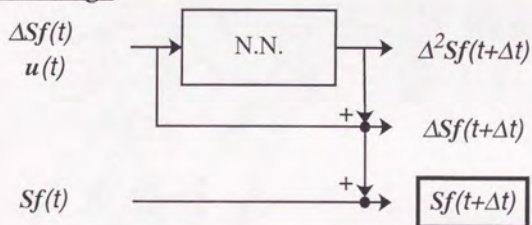
2nd Stage



3rd Stage



4th Stage



: Evaluated Signals

Fig. 3-8 Learning process to construct Identification Network.

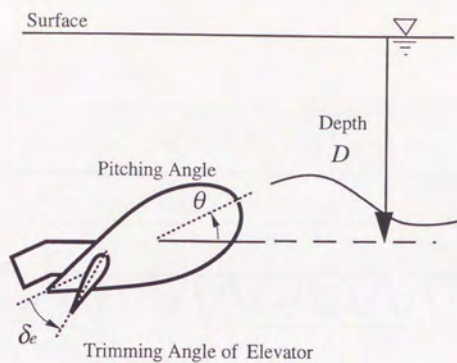


Fig. 3-9 Parameters of underwater robot PW45.

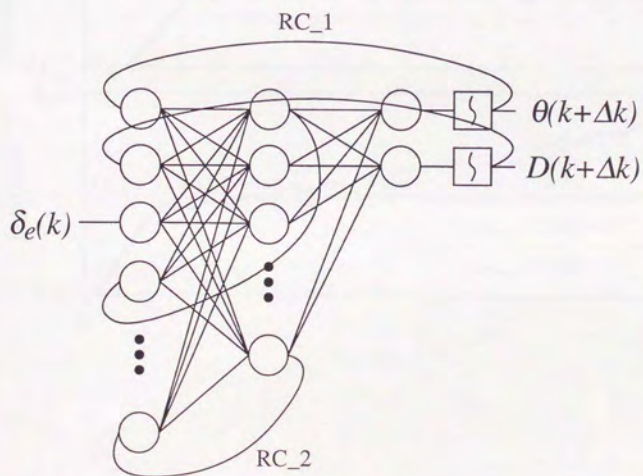


Fig. 3-10 Network structure to represent longitudinal motion of PW45.



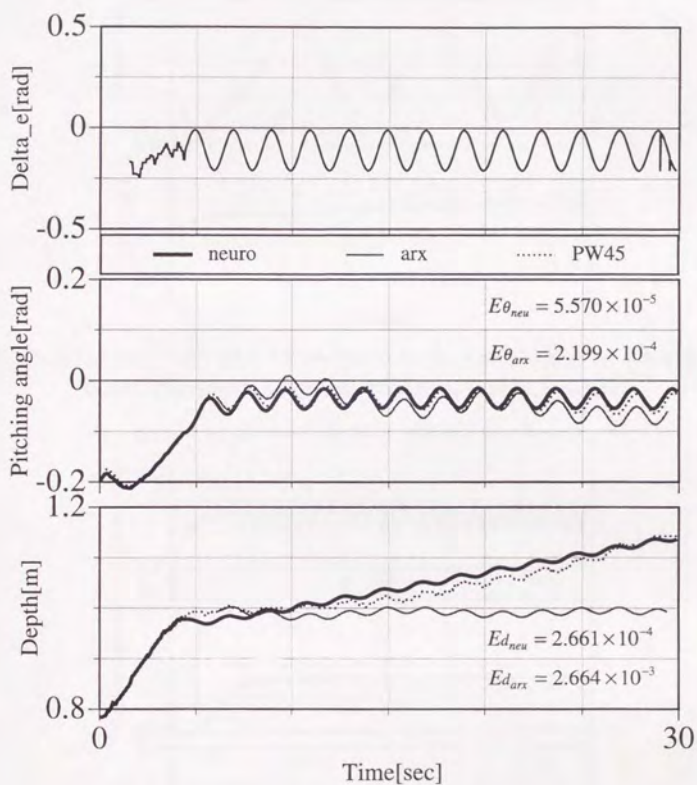


Fig. 3-11 Output of Identification Network and experimental results when elevators are trimmed in 0.5 Hz.

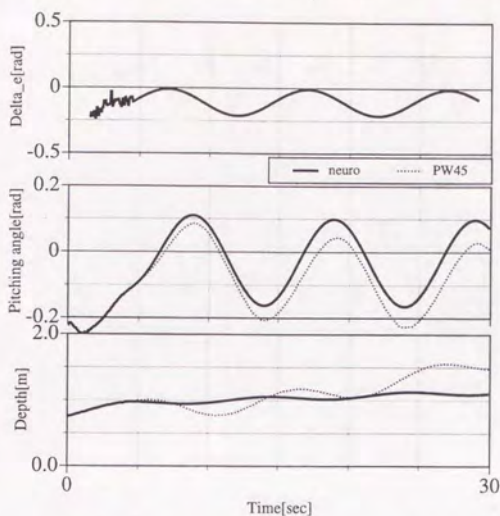


Fig. 3-12 Outputs of network and experimental results when elevators are trimmed in 0.1Hz. The synaptic weights are the same as those of Fig. 3-11.

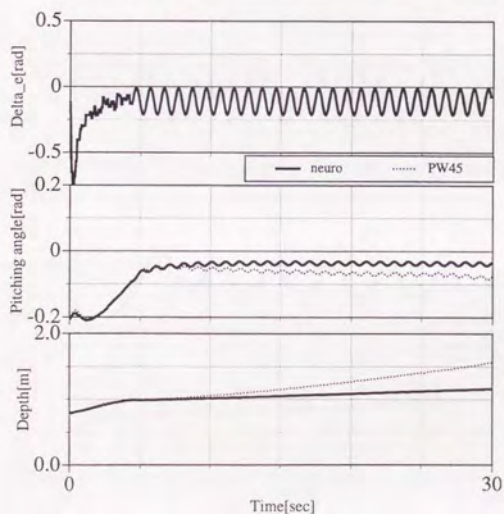


Fig. 3-13 Outputs of network and experimental results when elevators are trimmed in 1.0 Hz. The synaptic weights are the same as those of Fig. 3-11.

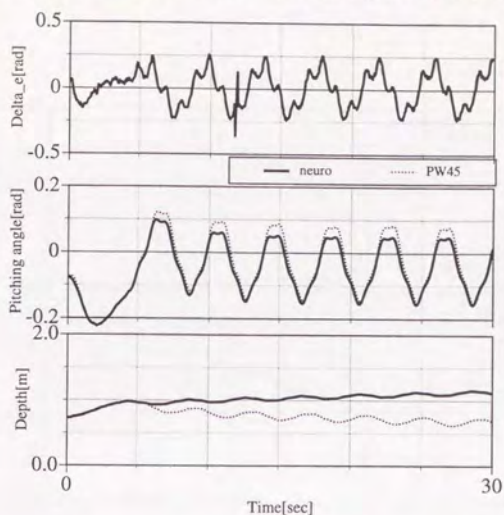


Fig. 3-14 Outputs of network and experimental results when elevators are trimmed in eq. 3-28(a). The synaptic weights are the same as those of Fig. 3-11.

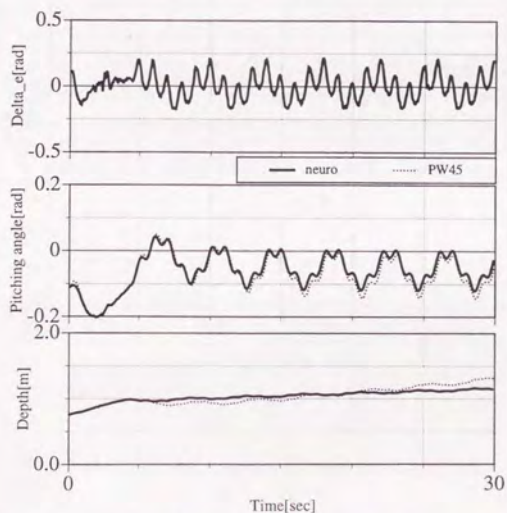


Fig. 3-15 Outputs of network and experimental results when elevators are trimmed in eq. 3-28(b). The synaptic weights are the same as those of Fig. 3-11.



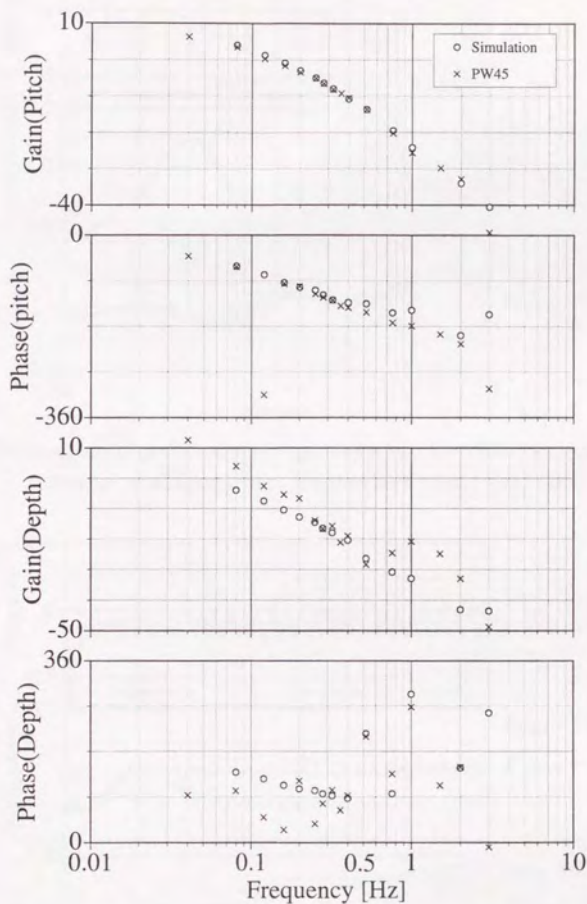


Fig. 3-16 Frequency response of constructed network.

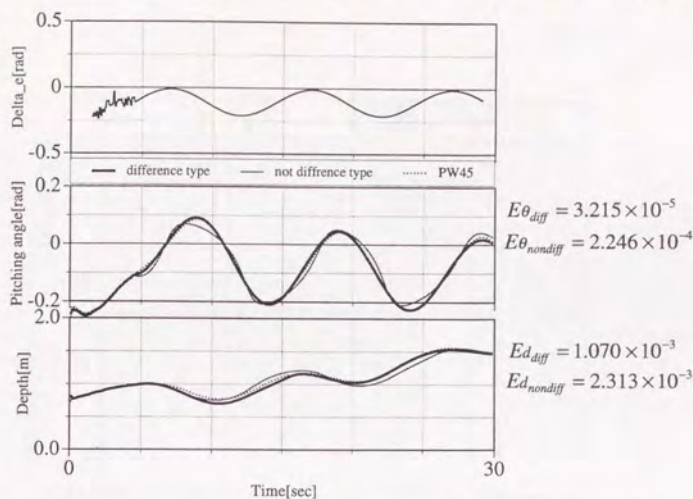


Fig. 3-17 Comparison of performance of constructed network as difference type and non-difference type network when elevators are trimmed in 0.1 Hz.

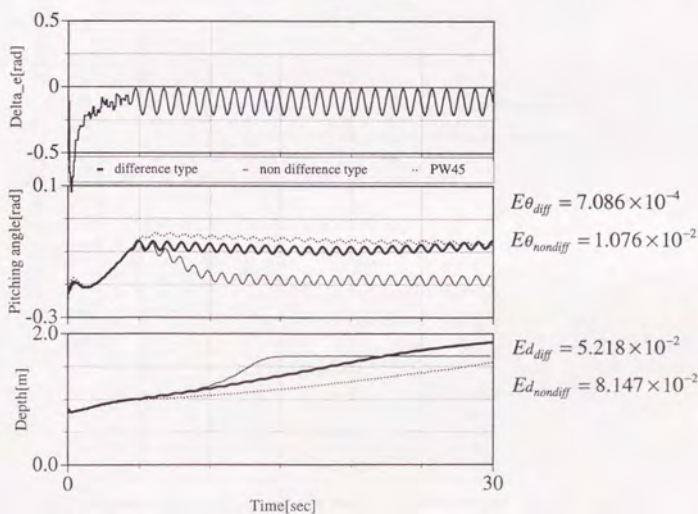


Fig. 3-18 Comparison of performance of constructed network as difference type and non-difference type network when elevators are trimmed in 1.0 Hz.

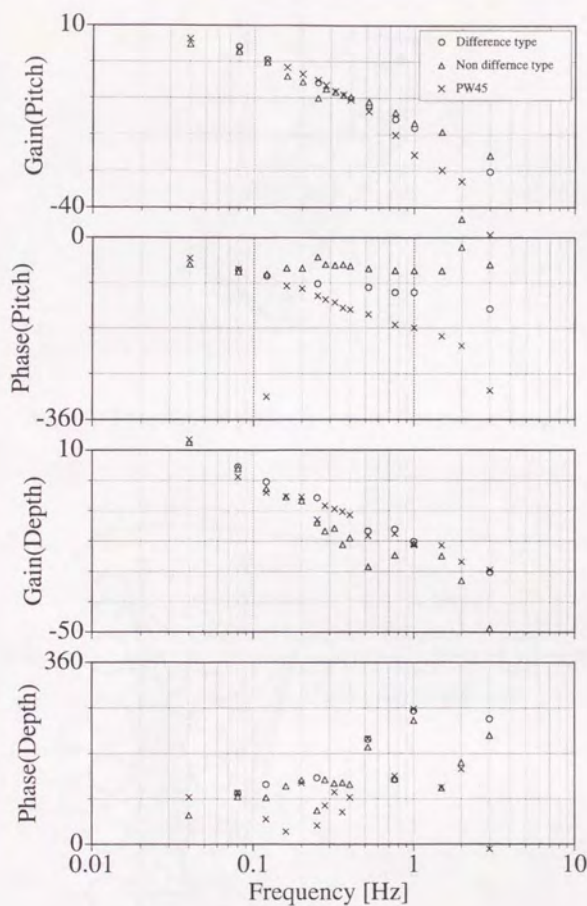


Fig. 3-19 Comparison of performance between difference type network and non-difference type network in frequency domain.



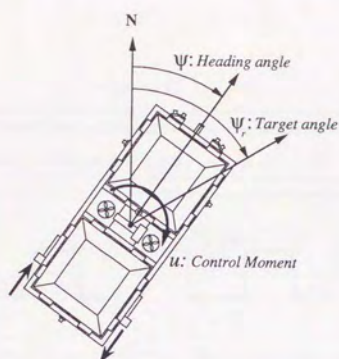


Fig. 3-20 Parameters of yawing motion of Twin-Burger vehicle.

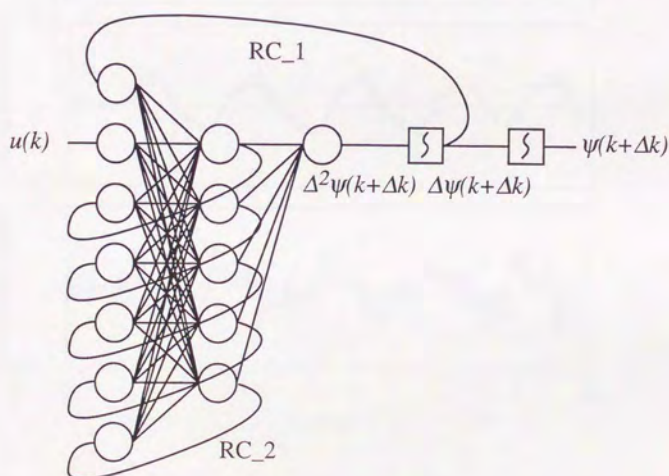


Fig. 3-21 Network structure of Identification Network for yawing motion of Twin-Burger vehicle.

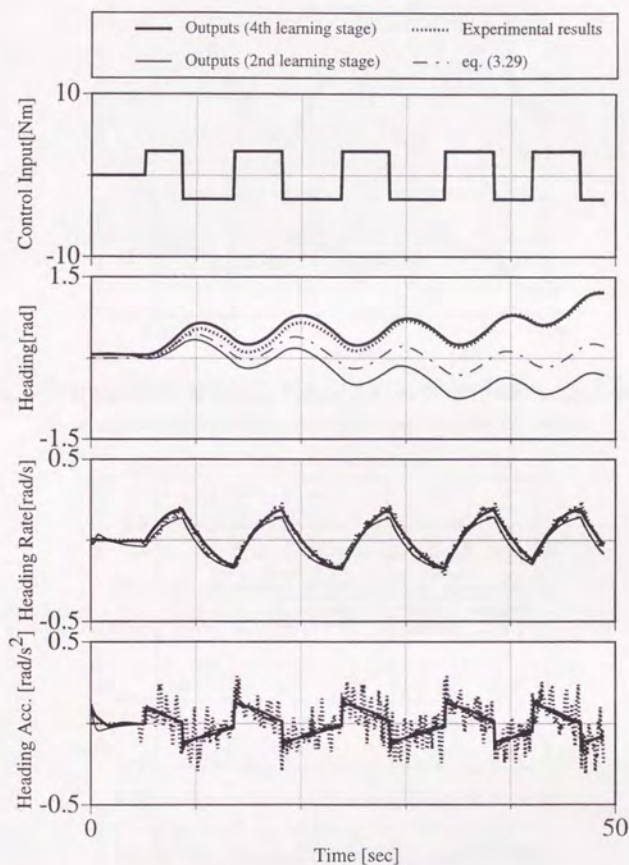


Fig. 3-22 Comparison of experimental results and outputs of Identification Network.

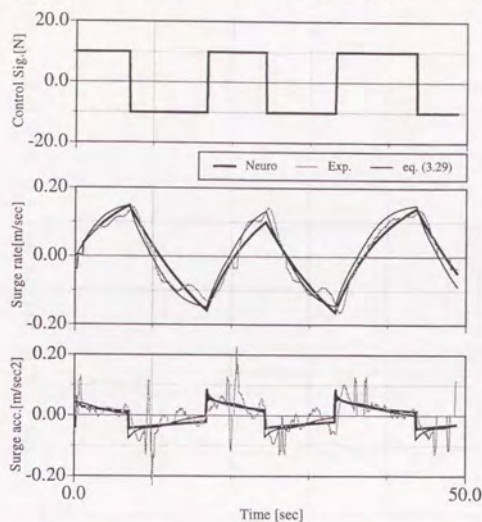


Fig. 3-23 Identification of surging motion of Twin-Burger vehicle. The results are compared with experimental results and equation of motion.

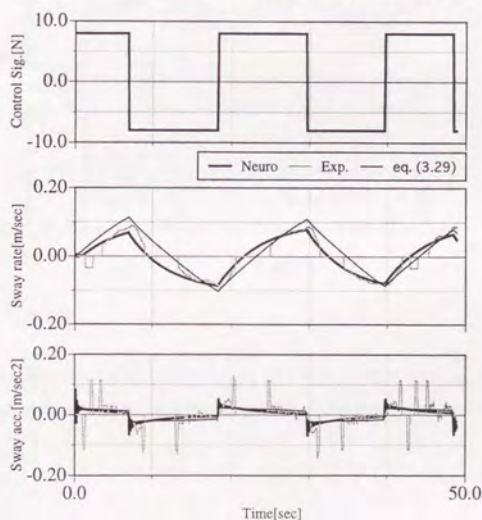


Fig. 3-24 Identification of swaying motion of Twin-Burger vehicle. The results are compared with experimental results and equation of motion.



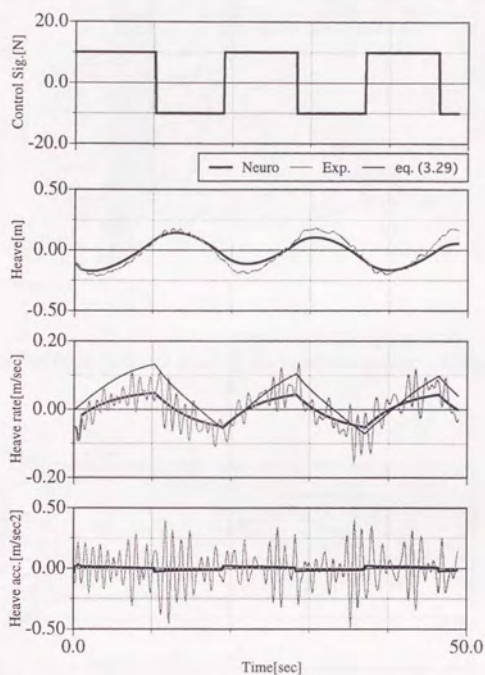


Fig. 3-25 Identification of heaving motion of Twin-Burger vehicle. The results are compared with experimental results and equation of motion.

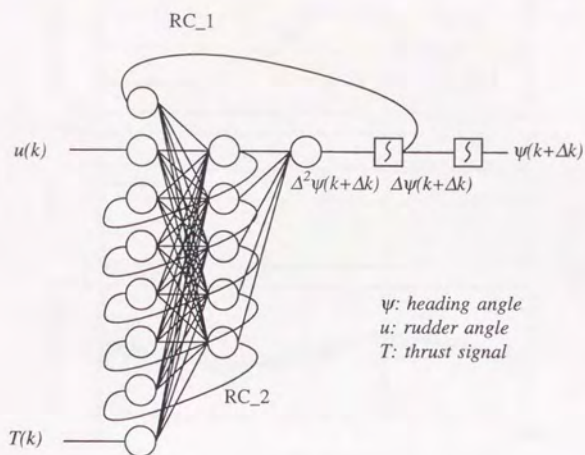


Fig. 3-26 Network structure to represent heading motion of R-one robot.

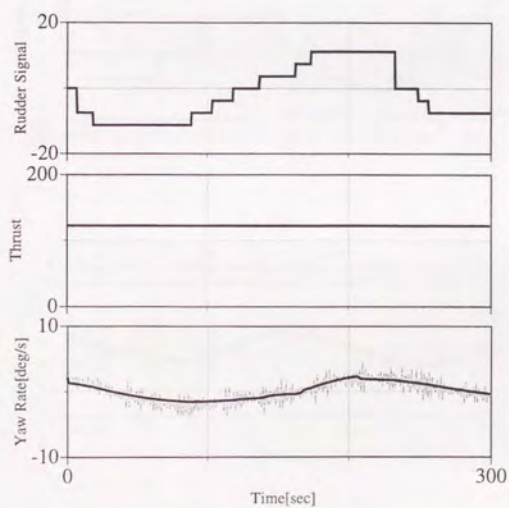


Fig. 3-27 Identification of heading motion of R-one robot when thrust signal is 120.

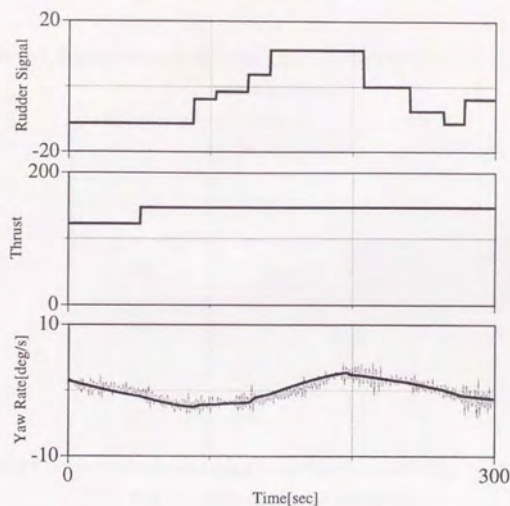


Fig. 3-28 Identification of heading motion of R-one robot when thrust signal is 150.

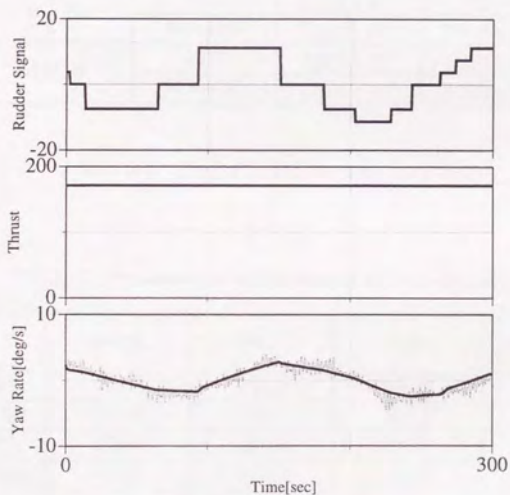


Fig. 3-29 Identification of heading motion of R-one robot when thrust signal is 180.



Table 3-1 Comparison of estimation error between Identification Network and ARX model.

	Pitching angle	Depth
Neuro	$5.570 \times 10^{-5}$	$2.661 \times 10^{-4}$
ARX	$2.199 \times 10^{-4}$	$2.664 \times 10^{-3}$

Table 3-2 Comparison of estimation error between difference type network and non-difference type network.

	Type	Pitching angle	Depth
0.1 Hz	Diff.	$3.215 \times 10^{-5}$	$1.070 \times 10^{-3}$
	Non-diff.	$2.246 \times 10^{-4}$	$2.313 \times 10^{-3}$
1.0 Hz	Diff.	$7.086 \times 10^{-4}$	$5.218 \times 10^{-2}$
	Non-diff.	$1.076 \times 10^{-2}$	$8.147 \times 10^{-2}$

Table 3-3 Parameters of vehicle motion of Twin-Burger.

		Surge	Sway	Heave	Yaw
M	TB1	202.3	214.9	480.5	28.6
	TB2	243.8	419.1	519.5	31.38
C	TB1	319.1	250.0	335.4	87.1
	TB2	414.7	230.1	414.7	136.5

## 第4章

ニューラルネットワークを用いた運動モデルの信頼性評価

#### 4. 1 はじめに

ネットワークの出力の評価は、非常に重要であるにもかかわらず、出力そのものを評価することは難しい。3章でも述べたように、入力変数が学習した領域内に存在する場合はある程度信頼性が高い出力を得られるが、学習した領域以外での保証がない。そこで、筆者らは出力の評価を入力変数を用いて評価する手法を考案し、“随伴ネット (Descriptive Neural Network)” を提案した。評価対象のネットワークにおいて、入力、つまり学習した領域については、既知であるので判別は可能である。評価対象のネットワーク (以後 Mapping-Net と呼ぶ) が十分に収束したとすると、出力の信頼性は入力の信頼性で近似できる。このことにより、入力 (学習領域) を確保する事により、信頼性を評価することが可能である。ここでは、随伴ネットを用いたネットワークの信頼性評価について述べる。

#### 4. 2 随伴ネットを用いた信頼性評価

##### 4. 2. 1 学習領域

$m$  個の入力信号からなる Connectionist Model を考える。入力信号ベクトルを

$$\mathbf{x} = (x_1, x_2, \dots, x_m) \quad (4.1)$$

で表し、すべての  $\mathbf{x}$  の集合を  $T$  で表す。 $T$  の  $n$  個の要素よりなる信号の集合を  $T_a$  とし、全ての出力信号の集合を  $S$  とする。教示型の Connectionist Model では  $T_a$  と  $S$  の部分集合  $S_a$  への写像関係を学習し、 $T$  から  $S$  への連続写像が形成される。このネットワークを  $Net\_A$  とする (Fig. 4-1 参照)。

学習が十分収束しており、目的の写像が連続であれば、 $T_a$  の各点の近傍で  $Net\_A$  はある精度で目的の写像を表しているはずである。しかし、その外側のどの範囲で有効であるかは、断定できない。ただ、 $T_a$  の密度が十分に高ければ、 $T_a$  を含むような  $T$  の部分集合については有効であろうと推定される。Fig. 4-2 に示されるように、その領域が (A) なのか (B) なのか (C) なのかは、定性的にしかいえず、(A) は、大胆、(C) は慎重といえる。

##### 4. 2. 1 随伴ネット

$Net\_A$  の入力データを  $T_a$  とするとき

- ・ 入力データ  $T_a$  と、この入力に対する出力 “1” の組 ( $n$  組)
- ・ 全入力領域  $T$  にランダムに分布された入力データと、この入力データに対する出力 “0” の組 ( $Z$  組)

を学習データとするニューラルネットワーク  $Net\_Z$  を考える。後者はバックグラウンドと呼んでおく。 $Net\_Z$  は、入力があると、それが過去に学習した領域に



あるかどうかを、 $[0, 1]$  の範囲で示すことになる。Net-Z は Ta の近傍では、0 より大きな値を取り、その外側では 0 に近づくネットワークになると思われる。

密度の定義はいろいろ考えられるが、ここでは 前者の密度を学習密度と呼び、Ta を含む  $m$  次元直方体で定義し、

$$Ma = \frac{n}{\prod_i (x_{i\max} - x_{i\min})} \quad (4.2)$$

とする。同じように、後者の密度をバックグラウンド密度と呼び、T を含む  $m$  次元直方体で定義し、

$$Mz = \frac{z}{\prod_i (x_{i\max} - x_{i\min})} \quad (4.3)$$

とする。両者の密度の比をバックグラウンド密度比と呼び

$$Rz = \frac{Mz}{Ma} \quad (4.4)$$

と定義する。このパラメータは

$Rz \geq 1.0$ : 学習密度の方がバックグラウンド密度より大きい。

$Rz = 1.0$ : 学習密度とバックグラウンド密度が等しい。

$Rz \leq 1.0$ : バックグラウンド密度の方が学習密度より大きい。

ということを意味する。

#### 4. 2. 3 随伴ネットの性質

随伴ネットの性質を調べるために、Mapping-Net の教示データとして次式から計算される値を元に (Fig.4-3 参照)、随伴ネットの出力について考察する。

$$f(x, y) = 0.2 * \sin(2\pi * 2.0 * \sqrt{x^2 + y^2}) + 0.5 \quad (4.5)$$

入出力は  $[0, 1]$  の範囲に正規化してあるものとする。Mapping-Net 及び随伴ネットの構造は (4.5) 式の関数を表現するために Fig.4-4 に示される構造とする。この入力領域のうち 学習領域は、Fig. 4-5 に示す中心部分の領域とする。Mapping-Net を用いて、学習領域の写像関係を誤差が十分に小さくなるまで学習する。学習後の Mapping-Net の出力が Fig. 4-6 である。学習領域内では、教示データの写像関係を良く表現しているが、学習領域外では (4.5) 式の写像関係が得られない。並行して作成した随伴ネットの出力を Fig. 4-7 に示す。中心部分、つまり学習した領域では信頼できるが、他の領域においては信頼度が乏しいことを示している。得られた随伴ネットの出力は、 $Rz = 1.0$  つまり、"0" と "1" の学習密度を等しくして学習させたものである。

Rzの違いによっておこる、随伴ネットの出力の違いを比較するために、Rz = 0.5 (Fig. 4-8) 及び Rz = 2.0 (Fig. 4-9) として随伴ネットに学習させた。これらの図を比較すると分かるように、Rz を小さくする (学習領域の密度を大きくする) とときには、"学習した"と思われる領域が拡大し、"大胆"な随伴ネットが出来上がり、逆に Rz を大きくすると"慎重"な随伴ネットとなる。

次に、学習領域が Fig. 4-10 において示される領域である場合について検討する。Rz を前述の場合と同じように、Rz = 1.0 として学習する場合、学習領域が Fig. 4-5 の場合に比べて 1/4 であり、学習領域内に存在する学習点数も 1/4 となる。このため、随伴ネットの出力がバックグラウンドの影響を強く受け、望ましい随伴ネットの出力が得られない。Fig. 4-11 はその例であり、学習した領域としていない領域の判別が困難である。

そこで以下のアルゴリズムを取り入れる。学習点の誤差の値を  $E$ 、学習した領域に対する教示データを  $A_{in}$ 、それ以外の領域の教示データを  $A_{out}$ 、定数  $e, b$  ( $0 < e, b < 1$ ) とすると、一連の教示データに対して

- 1)  $E > (A_{in} - A_{out}) * e$  となる入力データの個数を学習領域及びバックグラウンドにおいて算出し、その学習領域における個数を  $Ea$ 、学習していない領域での個数を  $Ez$  とする。
- 2) 誤差密度比  $Rer$  を求める。

$$Rer = \frac{Ez}{Vz} \frac{Va}{Ea} \quad (4.6)$$

- 3) バックグラウンド密度比と誤差密度比が等しくなるように、一定回数学習後に Rz を調整する。

$$\frac{Rer}{Rz} \geq 1.0 \text{ の場合}$$

$$Rz = Rz * (1.0 + \beta) \quad (4.7)$$

$$\frac{Rer}{Rz} \leq 1.0 \text{ の場合}$$

$$Rz = Rz * (1.0 - \beta) \quad (4.8)$$

のように、バックグラウンド密度を変化させる。このアルゴリズムを用いて得られた随伴ネットの出力が Fig. 4-12、その時の Rz の値の変化が Fig. 4-13 である。ここでは、1000 回の学習後に Rz の変更を行った。このように Rz を自動的に調整することにより、かなり望ましい随伴ネットの出力が得られることが分かる。

以上の考察より、随伴ネットの性質として

- ・ 学習領域近傍の学習していない領域に対しても、高い値を出力する。
- ・ Rz = 1.0 とすれば良いわけではなく、適切な値に調整する必要がある。
- ・ 誤差密度比により、バックグラウンド比を調整すれば、望ましい出力が得られる。

等の知見が得られた。



## 4. 2. 4 随伴ネットによる動的モデルの信頼性評価

随伴ネットを3章で述べた動的なモデルの信頼性評価に適用し、有効性の検討を行う。ここで扱うネットワークの構造を Fig. 4-14 に示す。随伴ネットの入力は回帰的な結合からの入力を見ると、13の入力変数を取り扱う必要がある。しかし、随伴ネットを作成するときに、13の全ての入力に関して対応させると、バックグラウンドとして入力するデータの個数を極めて大量にする必要がある。なぜなら、 $T_a$ の占める体積が $T$ の体積に比べて極めて小さくなるからである。この入力次元の増加に伴う入力数の増大を防ぎ、計算を速めるためにはためには、入力層の数を減らす必要がある。そこで、リカレント（中間層からの戻り値）の出力を調べたところ、ピッチ角及び深度のネットワークにおいて出力が同じような性質を示すことが判明した。従って、各ネットワークから1つのリカレントを取り出し、代表させることによって簡略化を行った。他の中間層からのリカレントについては、無視している。

ここでは、0.1 Hzの実験データを学習した同定モデルを例にとる。随伴ネットは、 $R_z = 0.008$  ( $0m < D < 2m$ ,  $-0.5rad < \theta < 0.5rad$ ,  $-0.5rad < \delta e < 0.5rad$ ,  $0 < \text{リカレント} < 1$ )で作成し、学習回数は500 M回とした。教示データに対するネットワークの出力を Fig. 4-15 に示す。全般に、ある程度信頼できることを示している。後半部で、出力が急激に下がっているのは、学習領域を次第に外れていき、境界上に存在していることを示している。

同じ結合荷重を用いて、エレベータ角を1.0 Hzの周期で変化させた場合の出力結果を Fig. 4-16 に示す。Fig. 4-16において、シミュレーション開始直後は随伴ネットの出力が大きく、入力領域が学習領域内に存在していると判断できる。しかし、運動の後半では随伴ネットの出力が下がり、Mapping-Net（ここでは運動モデル）の出力が疑わしいことを表現している。随伴ネットの出力が振動しているのは、制御量 $\delta e$ の影響と考えられる。

実際に適用する場合の工夫として、次の事が挙げられる。

- ・ 入力の爆発を防ぐために、リカレントの数の調整を行う。
- ・ 誤差密度比を用いて、バックグラウンド密度を適した値に制御する。
- ・ 学習点を中心に、近傍領域に"0"の点を一定距離おいて数点配置し、学習領域の境界を明確化する。そのときの距離は、はじめある程度大きく取り、収束しはじめてから、小さくしていく。
- ・ 全空間に"0"の点を配置する必要があるが、学習領域に近い、空間の中心近くでは、"0"の点を配置しないようにする。

これらの工夫で、精度の良い随伴ネットが得られ、ネットワークの信頼性評価を行うことが可能である。



### 4. 3 ネットワーク構造の最適化

ニューラルネットワークにおいて、ネットワークの構造は学習時間や精度、信頼性などに大きく影響する。ネットワーク構造は、入力層及び出力層のニューロン数はシステムの要求で必然的に決まるため、中間層とそのニューロン数に依存する。特に、回帰的な結合を持つネットワークでは、回帰的結合の数が中間層のニューロン数と等しくなるため、中間層の大きさが学習時間や精度に多大な影響を及ぼす。最適なネットワーク構造であると、学習時間、精度等向上するであろう。中間層の数をどう決定するかは色々な方法が(例えば[101])、ここでは中間層の出力を用いてネットワーク構造を最適化する。

#### 4. 3. 1 中間層の出力による構造最適化

2章でも述べたように、中間層から入力層へ回帰的な結合を持つネットワークでは、ネットワークが十分収束すると、入力層と出力層の知識が中間層に集約される。更に、本研究では動的なシステムの取り扱いを前提として入出力データを正規化した事により、各ニューロンの挙動が簡単に比較可能である。

フィードフォワード型のネットワークにおいて、学習が十分行われ、ある程度収束したとする。そのネットワークの中間層のニューロンの数が、最適な個数よりも多い場合、入力に対する応答が小さく、出力がほぼ一定であるニューロンが現われる。つまり、このニューロンはしきい値の役目をしているものと考えられる。従って、以下のアルゴリズムによって結合しているニューロンのしきい値を変更することにより、対象となっているニューロンを削除することが可能である。

入力層、中間層、出力層の3層からなるネットワークを考え、そのニューロン数を各々  $N_{in}$ 、 $N_{hid}$ 、 $N_{out}$  とし、ニューロン  $i$  とニューロン  $j$  の間の結合荷重を  $w_{ij}$  とする。ここで、中間層における  $k$  番目のニューロン  $Hid_k$  において、その出力  $y_k$  の変動が他のニューロンの出力の変動域と比較して十分に小さい値であるとする。その中間層ニューロン  $Hid_k$  及び  $Hid_k$  から回帰的に結合しているニューロン (Fig. 4-17の黒塗りのニューロン) は一定の値  $y_k$  を出力するため、中間層及び出力層のしきい値として扱うことが可能である。従って、中間層の各ニューロン  $i$  のしきい値  $\theta_i$  を式(4.9)を用いて更新する。

$$\theta_i = \theta_i + y_k \cdot w_{ki}^{rec} \quad (4.9)$$

同様に出力層のニューロンのしきい値も変更することにより、中間層ニューロン  $Hid_k$  を削除することが可能となる。

#### 4. 3. 2 運動モデルへの適用

この手法を既に得られた Fig. 3-17 に示される PW45 の運動モデルに適用する。この運動モデルは、周期0.1 Hz でエレベータを動作させた実験データを基に作成したモデルである。教示データを入力し運動のシミュレーションを行い、中間層

10個の出力を調べた結果、Table 4-1が得られた。この表から判断し、中間層の3、4、6、7、8、10番目の6個のニューロンを先に述べた手法に従って削除し、ネットワークの簡略化を行なった。簡略化したネットワークによる運動シミュレーションの結果をFig. 4-18、簡略化前後の結合荷重の一覧をTables 4-2、4-3に示す。構造を簡略化したネットワークにより、良好な結果が得られており、提案した手法が機能することが分かる。この手法によりネットワークの結合荷重は162個から46個となりネットワークの最適化が行なえている。

#### 4. 4 考察

ニューラルネットワークの信頼性を評価する方法として、学習領域の判定に基づいて信頼性の判断を行う随伴ネット、及び、中間層の出力からネットワーク構造の最適化を行い信頼性を上げる手法について考察した。

運動制御システムの構築を考えた場合、単一のネットワークにより全ての機能を実現するのは非現実的である。簡単なネットワークをモジュール化し、複数のネットワークを用いてシステム全体を構築する方法が現実的であり、機能が高いことが知られている[70]。随伴ネットは、学習した領域に入力データが存在するか否かを判定し、対象とするネットワークの信頼性を判定する手法であることから、複数のネットワークから構成されるシステムにおける、随伴ネットの評価に基づいたネットワークの切り替えや、シミュレーションの妥当性、追加学習の必要性等への適用が考えられ、今後の幅広い応用が期待される。例えば、運動について取り扱う場合、周波数やゲイン、位相遅れ等を入力として扱い、ネットワークが取り扱える領域に存在するかを判定することも可能である。

問題点として、入力の次元が多い場合において、ネットワークの収束に時間を要することが挙げられ、より効率の良い学習アルゴリズムを検討する必要がある。その解決法の1つとして、中間層の出力によるネットワーク簡略化について考察した。提案した手法は、中間層の出力を用いて中間層のニューロン数、つまり回帰結合のニューロン数を最適化するものであり、随伴ネットの作成において、回帰結合からの入力を選択することが可能となる。さらに、複数のネットワークを用いてシステム開発する場合、一般的に大規模なネットワークが構成されるが、本手法を用いてネットワークを簡略化することにより、システム全体が効率的に機能すると思われる。

今後、実際にネットワーク構造の適応アルゴリズムとして導入し、その可能性を見極める必要がある。これらの手法を用いることにより、より柔軟で適応性の高いシステムの構築が可能であると思われる。



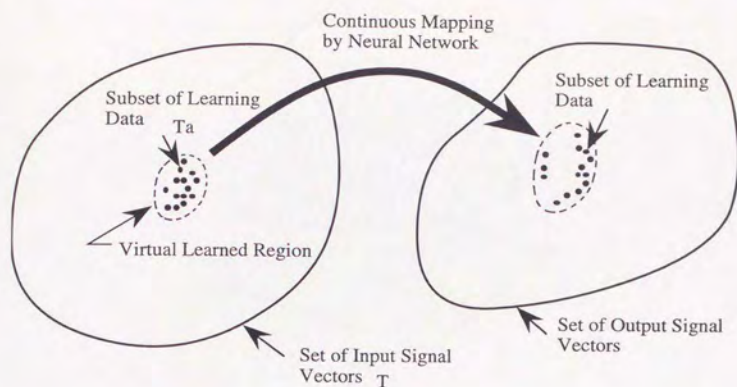


Fig. 4-1 Mapping by a neural network from subset of input signal vectors to a subset of output signal vectors.

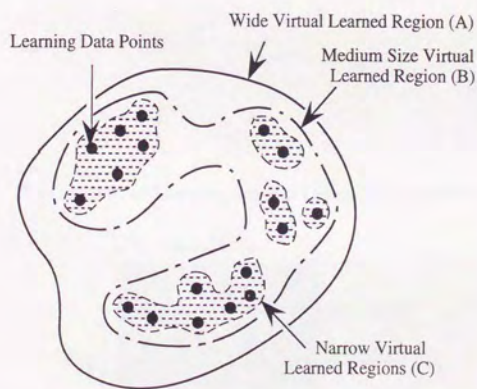


Fig. 4-2 Virtual learned region surrounding learning data points



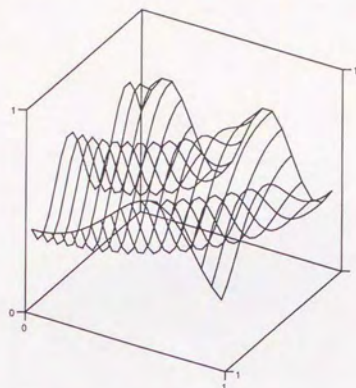


Fig. 4-3 Teaching data for Mapping net.

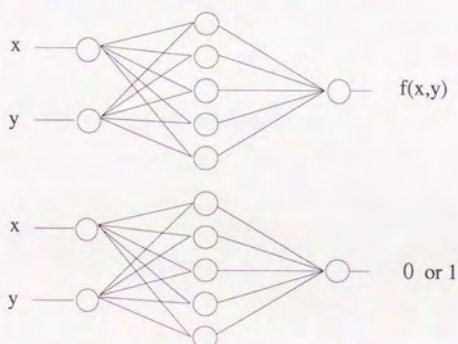


Fig. 4-4 The structure of Mapping net and Descriptive net for the example.

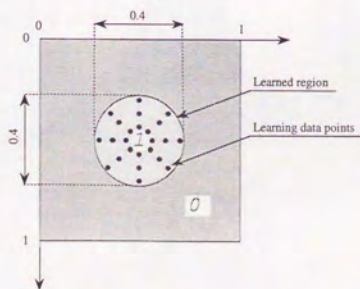


Fig. 4-5 Learned region surrounding learning points for the example.

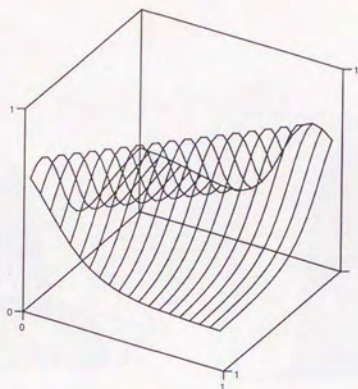


Fig. 4-6 Output of Mapping net.

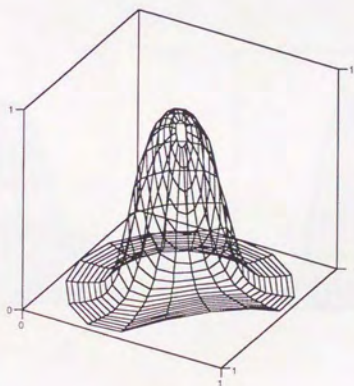


Fig. 4-7 Output of Descriptive net ( $R_z = 1.0$ ).

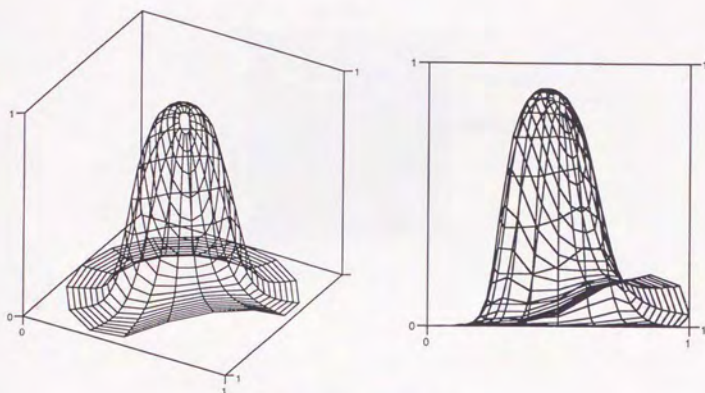


Fig. 4-8 Output of Desc.N.N.(  $R_z = 0.5$  ).

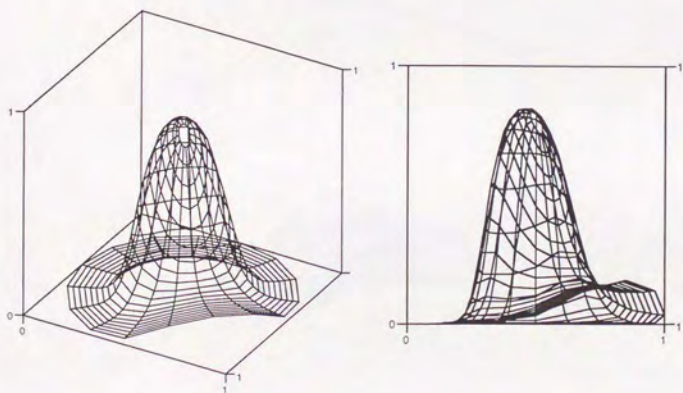


Fig. 4-9 Output of Desc.N.N. (  $R_z = 2.0$  ).



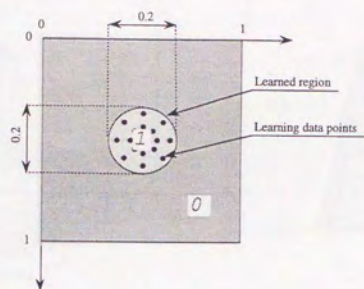


Fig. 4-10 Small learned region.

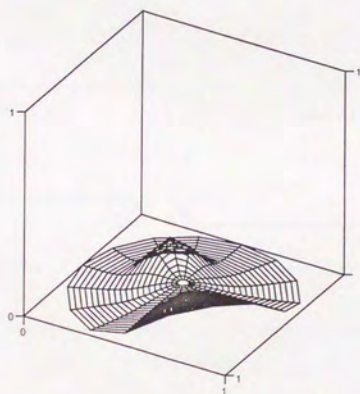


Fig. 4-11 Output of Desc.N.N. ( $R_z = 1.0$ ) when the learned region is small.

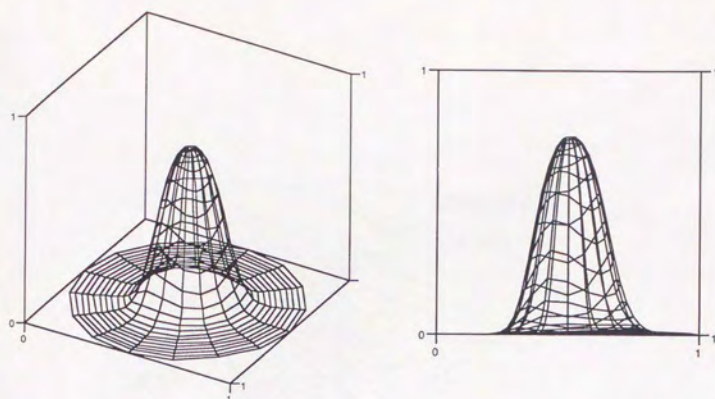


Fig. 4-12 Output of Desc.N.N. when  $R_z$  is changed according to the error density.

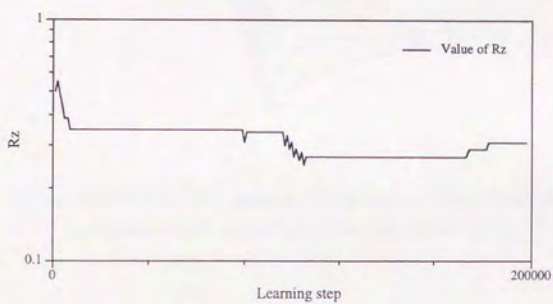


Fig. 4-13 The value of  $R_z$ .

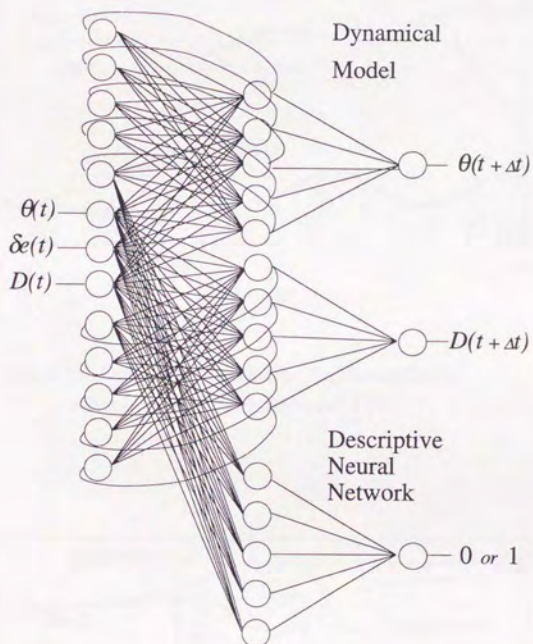


Fig. 4-14 Structure of dynamical model and Descriptive Neural Network for representation of longitudinal motion of PW45.



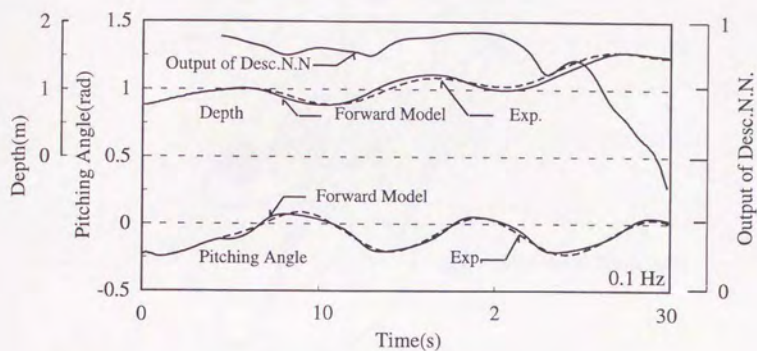


Fig. 4-15 Output of network and experimental result when elevators are trimmed in 0.1 Hz.

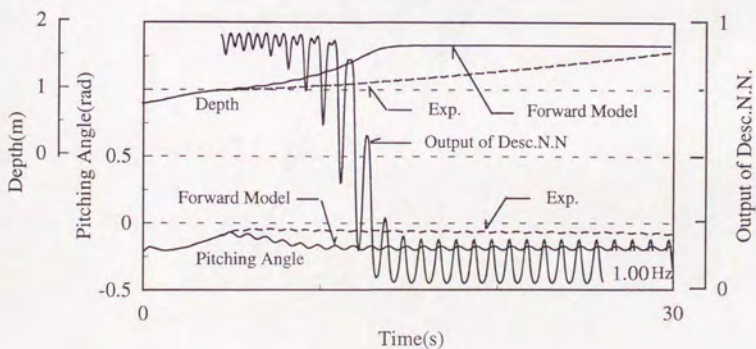


Fig. 4-16 Output of network and experimental result when elevators are trimmed in 1.0 Hz (using same weight).

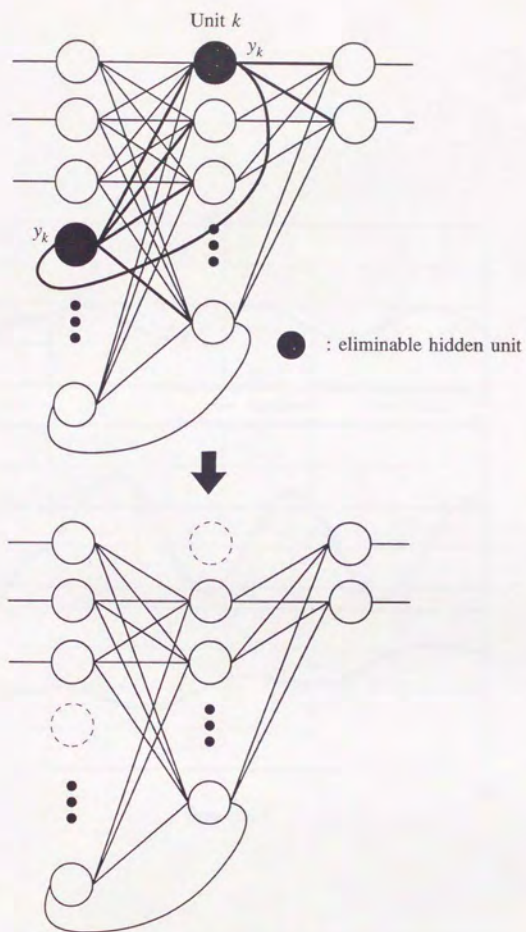


Fig. 4-17 Simplification of network structure.

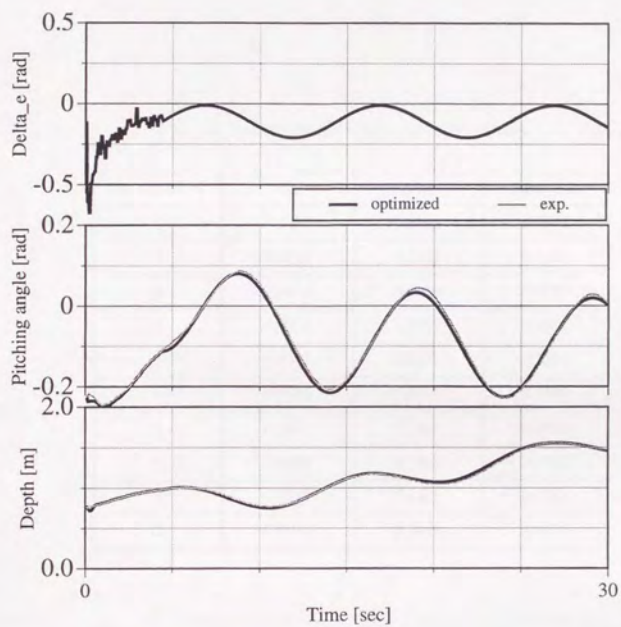


Fig. 4-18 Output of network after reducing 6 neurons of hidden layer.



Table 4-1 Outputs of Hidden layer's neuron.

No. of neuron	Max	Min	Max - Min
1	0.1227	0.0583	0.0592
2	0.9119	0.7465	0.1568
3	0.9999	0.9999	0.0000
4	0.9991	0.9959	0.0032
5	0.6957	0.3677	0.3280
6	0.9999	0.9999	0.0000
7	0.9996	0.9991	0.0005
8	0.9992	0.9990	0.0002
9	0.9877	0.8489	0.1367
10	0.9994	0.9963	0.0031

Table 4-2 Synaptic weights of original network.

		Unit Number of Hidden Layer									
		1	2	3	4	5	6	7	8	9	10
Input Layer	1	2.7891	1.8915	2.5633	3.3038	0.1417	3.0574	1.0334	3.4820	4.0510	1.5524
	2	0.9325	1.7572	3.8546	1.4373	0.8235	2.4199	1.4746	2.0517	2.3088	2.1743
	3	-1.7756	-2.8664	-2.2213	-3.3962	2.7871	-1.2543	-0.0907	-3.5368	-3.3466	-1.6933
Output Layer	1	-0.9193	-1.0496	0.2504	0.0400	1.9076	-0.1954	0.1038	1.1188	0.3647	0.7926
	2	-5.0986	3.1791	-0.7745	-0.5016	-0.6643	-0.3733	-1.3662	-1.4738	-4.1900	1.2528
Rec. (Input)	1	3.3953	-0.3872	0.1612	1.3014	-0.1317	1.4275	-0.4142	0.3366	0.4943	-0.5509
	2	-1.0696	0.9765	1.2232	-0.1241	-1.8787	1.5091	-0.5823	0.0410	0.3995	4.7343
	3	0.6686	-1.1377	7.0905	1.1160	-0.2701	2.9970	-0.4738	-0.4410	-0.7496	-1.9161
	4	1.1122	1.0309	2.1518	5.3656	0.3277	1.3315	-2.6550	1.7523	0.0652	0.3928
	5	-0.6183	-0.6822	-1.7698	-1.4972	1.3746	-1.5385	1.0284	-1.6434	-1.9849	-1.2993
	6	0.6006	-0.9632	2.1060	-0.1211	-0.2579	6.7327	-1.0336	-1.9308	-1.5490	-2.4728
	7	-0.5451	0.8305	-0.6912	-1.4890	0.4671	-1.0707	3.8184	1.9099	0.3657	0.0289
	8	-0.0171	1.1081	1.2872	3.3257	0.7318	0.5753	3.5501	5.7371	1.1115	1.0558
	9	-1.8226	1.0390	1.0108	1.5873	0.0993	0.4780	2.7282	4.4339	6.8663	1.4922
	10	-0.3359	1.0929	-0.4678	-0.8451	0.3983	-0.5299	1.4518	-0.5565	-0.5564	5.7418
Threshold	hid	-2.0018	-2.0634	-4.5896	-2.3862	-2.2601	-3.1145	-1.0236	-2.5811	-2.6553	-2.0052
	out	-2.5262	5.4319	-	-	-	-	-	-	-	-

Table 4-3 Synaptic weights of optimized network.

		Unit Number of Hidden Layer									
		1	2	3	4	5	6	7	8	9	10
Input Layer	1	2.7891	1.8915	-	-	0.1417	-	-	-	4.0510	-
	2	0.9325	1.7572	-	-	0.8235	-	-	-	2.3088	-
	3	-1.7756	-2.8664	-	-	2.7871	-	-	-	-3.3466	-
Output Layer	1	-0.9193	-1.0496	-	-	1.9076	-	-	-	0.3647	-
	2	-5.0986	3.1791	-	-	-0.6643	-	-	-	-4.1900	-
Rec. (Input)	1	3.3953	-0.3872	-	-	-0.1317	-	-	-	0.4943	-
	2	-1.0696	0.9765	-	-	-1.8787	-	-	-	0.3995	-
	3	-	-	-	-	-	-	-	-	-	-
	4	-	-	-	-	-	-	-	-	-	-
	5	-0.6183	-0.6822	-	-	1.3746	-	-	-	-1.9849	-
	6	-	-	-	-	-	-	-	-	-	-
	7	-	-	-	-	-	-	-	-	-	-
	8	-	-	-	-	-	-	-	-	-	-
	9	-1.8226	1.0390	-	-	0.0993	-	-	-	6.8663	-
	10	-	-	-	-	-	-	-	-	-	-
Thresh hold	hid	-0.5185	-0.1021	-	-	-0.8633	-	-	-	-3.9677	-
	out	-0.4162	2.1955	-	-	-	-	-	-	-	-