

# 複雑な形の海底面上を航行する海中ロボットの訓練 ——その1：自己生成型ニューラルネットワーク制御システムの適用——

Unsupervised Learning System For Vehicle Guidance Constructed With Neural Network

浦 環\*・須藤 拓\*  
Tamaki URA and Taku SUTO

## 1. はじめに

航行型の海中ロボットが海底面およびその近くの情報を得るために航行するとき、海底面に十分に近づく必要があるが、同時に衝突を避けなければならない。そのため地形情報はロボットの前方および下方への海底面までの距離を超音波によって測ることで得られる。この地形情報を基に目的を遂行するようにロボットの運動は制御されるのだが、ロボットのダイナミクスや地形情報をすべて考慮し、その全ての場合を含んだようなアルゴリズムをあらかじめ決めておく事は困難である。そこで目的に対して状況に応じて制御が適応的に変わっていくようなコントローラが必要となる。

SONCS<sup>1)~4)</sup>(Self-Organizing Neural-net-Controller System)は本研究室で開発され、艇の運動の適応的な制御に用いられた。ここではSONCSのアルゴリズムに基づいて超音波による距離情報を取り込み、複雑な地形の上をロボットが自力で定高度航行をするようなシステムを構築する。

## 2. ロボットの諸元

ロボットのモデルとしてPTEROA150<sup>5)6)</sup>を選び、その縦運動をシミュレーション上で制御する。その運動は実験から得た運動方程式<sup>7)8)</sup>から計算する。PTEROA150は全長150cm、乾燥重量220kgの航行型無索無人潜水艇である。図1のように鉛直面内に真下から正面まで30°おきの方向に4つの超音波測距装置を有し、それぞれに対応する装置で測られた距離を $k_0, k_1, k_2, k_3$ とする。Aは高度を表し、 $k_0, k_1$ に対応するビームが海底面に反射する点を結んだ線と艇との距離と定義する。 $\theta, \delta_e$ はそれぞれピッチ角、エレベータトリム角である。艇の運動はエレベータトリム角を動かす事だけで制御する。推力は50Nに固定し、その時の平均速度は約5 knotsである。また地球に固定された慣性座標系 $X_e-Z_e$ は図1のようにとる。

\*東京大学生産技術研究所 第2部

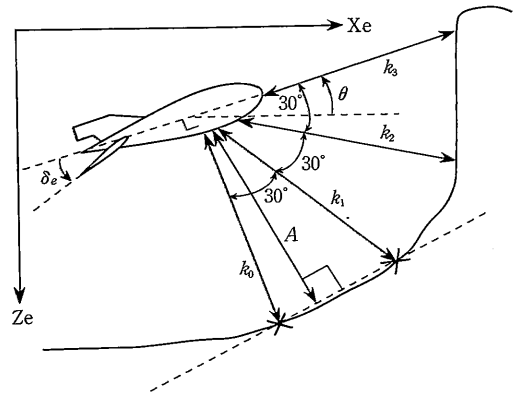


図1 ロボットの縦運動

コントロールの目的は連続した三角形断面を有する地形の上を定高度で航行する事である。

## 3. ネットワークの構成

図2はSONCSを構成するニューラル・ネットワークの部分である。左のネットワークがコントローラ・ネットワークで右のネットワークがロボットの運動を表すフォワードモデル・ネットワークである。

コントローラ・ネットワークは3層コネクショニストモデル<sup>9)</sup>で入力層に12ニューロン、中間層に5ニューロン、出力層に1ニューロンを有す。入力としては制御量に関係を持つであろう図2のようなものを使う。ここで $\Delta$ は現在の値と前の時間刻みの値の差を表す。出力はエレベータトリム角である。

フォワードモデル・ネットワークも同様に3層のコネクショニストモデルであるが入力層と中間層の間に図のような回帰的な結合を有す。信号はここを1時間刻み遅れて伝わるので結果として過去の情報の履歴をネットワーク内に保持することとなる。フォワードモデル・ネッ

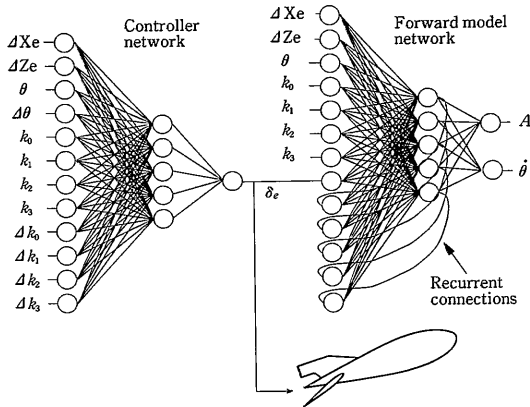


図2 ネットワークの構成

ネットワークは入力層に13ニューロン、中間層に5ニューロン、出力層に2ニューロンを持ち、入力層の13ニューロンのうち5ニューロンは回帰的結合のために用いられ残りの入力図2のとおりである。フォワードモデル・ネットワークには回帰的結合があるので過去の履歴がネットワーク内に内在する。このためコントローラ・ネットワークの入力にある  $\Delta k_0, \Delta k_1, \Delta k_2, \Delta k_3$  を入れていない。コントロール結果の評価量としての出力は高度とピッチ角速度とした。

入力はおおよそ  $(-1.0, 1.0)$  に、出力はおおよそ  $(0.0, 1.0)$  になるように線形変換を施した。

4. コントローラ・ネットワークの初期化とフォワードモデル・ネットワークの形成

コントローラおよびフォワードモデル・ネットワークの初期化は図3のようにCRTディスプレイに表示されるロボットの動きを見ながら筆者が手動で泳がせた運動を基に行った。地形は図中のtrajectoryの所にあるように底辺が100m、高さが10mの三角形で構成されている。このとき時間刻み0.1秒でデータを取り、1セット570データを初期化のために取った。

コントローラ・ネットワークの初期化は、このデータセットに対して、8000回の学習により行った。図4はその結果で教示データとほぼ同じような傾向を示すようになっている。

フォワードモデル・ネットワークの形成は高度10mという目標に対しては図3のものでも充分なので同じデータセットを用いて8500回の学習を行った。図5はその結果であり、高度の最大誤差は1.5m程度でロボットの運動をほぼ正しく表現できるようになっている。この学習をさせているデータの高度の範囲は2~13mであるが、この範囲を越える範囲で評価値を求めようとする場合には、

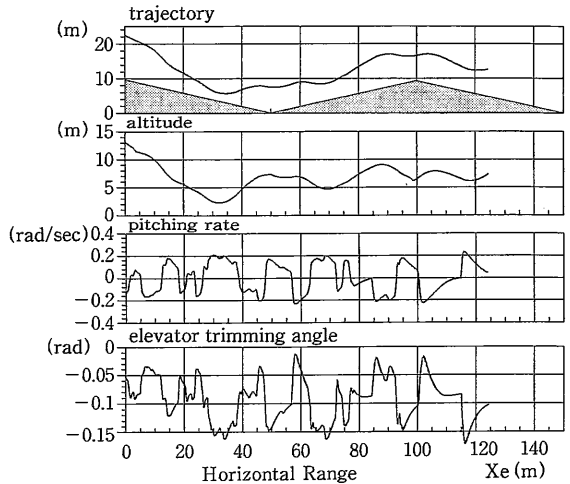


図3 初期化の為のデータ

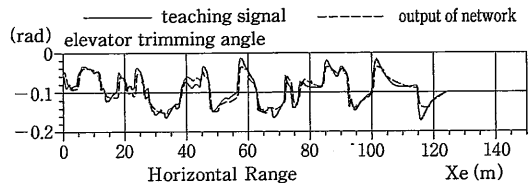


図4 初期化したコントローラの出力

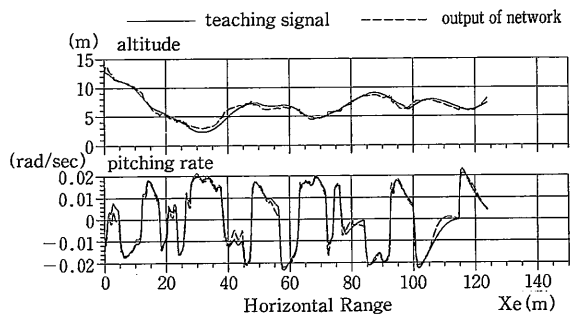


図5 フォワードモデルの出力

ネットワークの出力は外挿されたものとなり、その場合には一般的に精度が落ちることとなる。このときコントローラ・ネットワークのアップデートと平行してフォワードモデル・ネットワークの追加学習も進める事が望ましい。今回は、上記の範囲で泳がせているのでフォワードモデル・ネットワークの追加学習は行っていない。

5. 訓練

初期化されたコントローラ・ネットワークの出力  $\Delta e$  をロボットとフォワードモデル・ネットワークの入力につなぎ、目的に適ったコントローラを得るためにバック・プロパゲーション法<sup>11)</sup>によるアップデートを行う。航行試験(以後訓練と呼んでおく)は1000ステップ(100

研究速報

表1 アダプテーションの諸条件

total adaptation times	0	2	5	13	54
weights for $A$ and $\dot{\theta}$	—	1,0.1	1,0.1	1,0.1	1,0.1
$\eta$ for back propagation	—	0.05	0.005	0.004	0.004
$\alpha$ for back propagation	—	0.2	0.2	0.2	0.2
root mean square error for altitude (m)	3.56	4.10	4.25	3.60	2.36
root mean square error for pithing rate (rad/sec)	0.045	0.045	0.042	0.042	0.027

秒)間か、もしくは海底面に衝突するまでを1回とし、1回の訓練に対して

$$E = \sum_n \{a_1(A - A_0)^2 + a_2\dot{\theta}^2\}$$

により、コントローラ・ネットワークの重みを調整する。ここで $a_1$ 、 $a_2$ はそれぞれ高度とピッチング角速度の重み付けで、 $n$ は各データのインデックスである。目標高度 $A_0$ を以後10mと定める。図6は訓練の過程を示したものである。地形とロボットの初期状態は4節で述べたものと同じである。表1は学習の条件を示す。

訓練開始直後のコントローラは図3のtrajectoryを真似ようとするが、72m程進んだときに海底面に衝突してしまう。しばらく訓練とアダプテーションを続けることにより、この衝突は避けられるようになり、さらにアダプテーションが進むとロボットは徐々に小さなピッチング角速度で長い距離を泳ぐことができるようになる。ここで、コントローラのアダプテーションは、方法を教えるのではなく、ポテンシャル関数だけから行われている事に注目せねばならない。最終的には艇は高度10mで航行するという目的に対して誤差3m以内で泳げるように

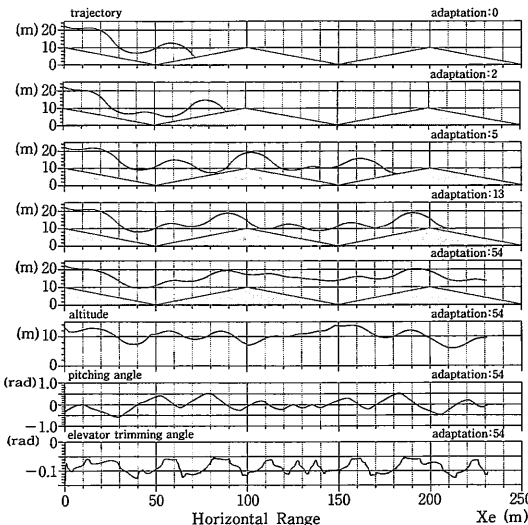


図6 訓練の過程

なっている。

6. さまざまな地形への適用

前の地形に対して作られたコントローラ・ネットワークが別の地形に対しても有効か、あるいは速やかにアダプテーションが行われるかどうかを検証する。

図7は前節で最終的に得られたコントローラに対して2倍の高さを有する三角形からなる地形に対してコントローラ・ネットワークの訓練とアダプテーションを進めていった例である。アダプテーションをする前でも、それなりに泳ぐのだけが頂点に接近する事がある。167回のアダプテーションの後、ロボットは大体目標の高度を保って泳げるようになる。山頂と谷底では地形が折れ曲がっており、高度が $k_0$ と $k_1$ でのみ求められている事を考えると図7の結果は評価できる。

図8は最初の地形に対して底辺の長さが半分の三角形

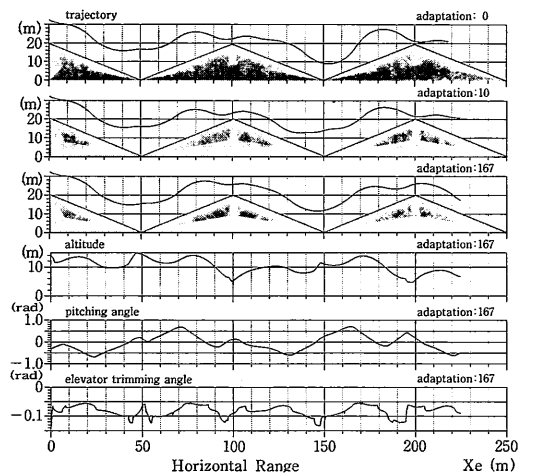


図7 険しい山での訓練

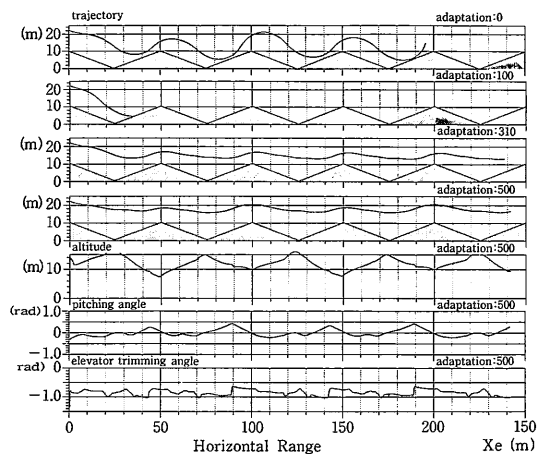


図8 小さな山での訓練

研究速報

からなる地形について訓練を進めたものである。図 8 の地形について作られたコントローラ・ネットワークでは頭上げを行うタイミングが遅れるために危険な状態に陥ってしまう。この後約 100 回の訓練の間、ロボットは海底にぶつかってしまう。これは高度の定義が  $k_0$  と  $k_1$  を用いてなされているために、 $k_2$ ,  $k_3$  が小さくなくても頭上げをしようとする意志が起こらないからであり、評価関数の定義に問題があることによる。しかしこの高度も衝突時には小さくなるので図 8 ではアダプテーションが進むにつれて、徐々に改良され 310 回目のように泳ぐようなコントローラ・ネットワークができる。この種の衝突に対して安全に、かつ速やかに訓練を進めるためには  $k_2$ ,  $k_3$  をも取り込んだ形でポテンシャル関数を定める必要がある。

## 7. 結 論

自己生成型ニューラルネットコントローラシステム SONCS によって複雑な地形に沿ってロボットを航行させるための適応的なコントローラを作れる事が示された。ここではフォワードモデル・ネットワークの入力として海底面に対する距離情報を用い、地形に関する情報をその中に取り込ませて、出力として高度を出させた。このような SONCS の柔軟な特性を発展させていけば、ニューラル・ネットワークによる制御は海中ロボットの実際の適応的な制御方法と成り得ると結論できよう。

(1991年7月15日受理)

## 参 考 文 献

- 1) 藤井輝夫, 浦環, 黒田洋司: “自己生成型ニューラルネットコントローラシステムの開発と潜水機の運動制御への適用”, 日本造船学会論文集, Vol. 168 (1990), pp. 275-281
- 2) 藤井輝夫, 浦環, 黒田洋司, 能勢義昭: “自己生成型ニューラルネットコントローラシステムの開発と潜水機の運動制御への適用 (その 2)”, 日本造船学会論文集, Vol. 169 (1991), pp. 477-486
- 3) Fujii, T. and Ura, T.: “Development of Motion Control System for AUV Using Neural Nets”, Proc. of AUV90., Washington D.C. (1990), pp. 81-86
- 4) Fujii, T. and Ura, T.: “Neural Network Based Adaptive Control Systems for AUVs”, IJEAAL, Special Issue in Intelligent Autonomous Vehicle Research (Sep. 1991), to be published
- 5) Ura, T.: “Free Swimming Vehicle PTEROA for Deep Sea Survey”, Proc. ROV'89, San Diego (1989), pp. 263-268
- 6) 浦環, 能勢義昭, 坂巻隆: “グライダー型潜水艇の設計に関する研究 (その 7)”, 生産研究, Vol. 43, No 2 (1991), pp. 39-42
- 7) 浦環, 大坪新一郎: “航行型無索無人潜水艇に関する研究”, 日本造船学会論文集, Vol. 162 (1987), pp. 117-124
- 8) Ura, T. and Otsubo, S.: “Gliding Performance and Longitudinal Stability of Free Swimming Vehicle”, Proc. PACON88, Honolulu (1988), pp. OST1/10-18
- 9) Rumelhart, D.E., McClelland, J.L. and The PDP Research Group: “Parallel Distributed Processing Volume 1: Foundations”, MIT Press, Cambridge (1986)