力学系に基づく自律移動ロボットの行動学習

福村 直博^{*1} 谷 淳^{*2}

Learning Goal-directed Behaviour as Dynamical System for Sensory Motor System

Naohiro Fukumura^{*1} and Jun Tani^{*2}

This paper describes an experimental results based on our prior-proposed scheme: learning of sensory-based, goal-directed navigation. The emphasis is that learning a task-based behavior could be formulated as an embedding problem of dynamical systems: desired trajectories in a task space should be embedded into an adequate sensory-based internal state space so that an unique mapping from the internal state space to the motor command could be established. In this study, two types of neural architecture were tested for constructing such an adequate internal state space based on sensory information. A simple homing task was conducted on Feed Forward Neural Network that is fed with a regression of sensory sequence as time-delayed manner. A task of complex cyclic routing was conducted on Recurrent Neural Network (RNN). In both cases, sufficient supervised training of robots generated rigid internal structure of attractor dynamics which realized desired navigations in robust manner, even against limited environmental changes as well as miscellaneous noises in the real world. It was confirmed that RNN architecture has more flexibility adapting to various tasks.

Key Words: Autonomus Mobile Robot, Navigation, Neural Network, Dynamical System, Embedding

1. はじめに

現在移動ロボットの研究において、センサを使って外界の情報を獲得し、環境に適応した行動を決定するロボットの自律化の研究が盛んである。自律ロボットが目標に向かって移動していくためには、ロボット自身が環境に関する何らかのモデルを持っていることが必要である。多くの研究においてはデッドレコニング及びレンジイメージなどのセンサ情報を統合して絶対座標系に基づく詳細な環境地図を作成し、それを用いて位置推定をしながら目標に移動していく方法が試みられている(例えば[1][2])。しかし、これらの手法は実世界でのノイズの影響を受けやすく、絶対座標系での環境モデルの作成やそれに基づく位置推定には限界があると考えられる。

これらとは異なったアプローチとして、MataricはBrooksのsubsumption architecture[3]の考えを拡張し、行動に基づく(Behaviour Based)移動ロボットのナビゲーション法を提案している[4]。これは超音波センサによるセンサデータを用い

て壁などを認識し、それらをランドマークとする。ロボットは詳細な環境地図を作るのではなく、ランドマークをノードとする有限状態オートマトンの形に簡略化された環境のモデルを獲得していく。しかし、この方法でもランドマークの検出のために移動を壁面に沿った動きに限定するなどのいくつかの仮定が必要であるので、複雑なナビゲーションは不可能である。

我々はBrooksらのセンサベースの行動という考え方に加えて、新たに力学系の枠組でロボットの行動学習を考えることを提案し、シミュレーションによってその有効性を確かめた[5]。この方法では、ロボットが目的のタスクを達成する過程を作業座標系上のあるアトラクタに収束するダイナミクスとして捉え、この作業座標系上の軌道を埋め込める適当な内部空間をセンサ時系列情報を基に構成する。その内部状態からモータコマンドへの写像をニューラルネットワークを使った教師付学習を通して獲得することにより、センサ情報に基づいたナビゲーションが実現できる。Mataricの環境モデルの記号的な表現と我々の提案する力学系による表現を比べると、記号的な表現は明示的であり、かつコンパクトである。しかし、ノイズを含む現実の世界を単に記号的に表現するには限界があると考えられる。一方力学系による表現は冗長であるが、それに基づく学習・行動のプロセスははるかにロバストである。

本論文では特に、センサ時系列情報に基づく内部空間の構成

原稿受付 1994年3月9日

^{*1}SONY 中央研究所

^{*2}SONY コンピュータサイエンス研究所

^{*1}Research Center, SONY Corp.

^{*2}SONY Computer Science Laboratory Inc.

を(1)時系列回帰の構造をアプリアリに与える, (2)学習を通して必要な回帰構造を自己組織化させる, の両者のアプローチについて, 実際の移動ロボットを使った実験を通して比較, 考察を行う。

2. 力学系の導入

本章では我々がすでに提案した, 力学系に基づくロボットナビゲーション法の概念を説明する[5]。

我々は以下の条件のもとにロボットのナビゲーションを行う。

- ロボットは地図情報や, 絶対座標系上での自分の位置などの情報を持たず, 局所的に得られるレンジセンサのデータのみに依存して自律移動する。
- 移動する範囲のジオメトリは任意であり, 特別なランドマークを必要としない。
- ロボットは何処からスタートしても目標のタスクを達成できる。
- 環境が多少変動してもタスクを達成できる。

具体的なタスクとしては

- ある決まったゴールに回帰する
- ある決められた軌道を周回し続ける

をとりあげる。

ロボットがこれらのタスクを達成する過程は作業座標系上ではあるアトラクタに収束するダイナミクスとしてとらえることができる。

ロボットがセンサ情報のみに頼ってタスクを達成するためには, この作業座標系(Fig. 1(a))における軌道ベクトルをセンサ情報に基づく適当な内部空間に(交差がなく)埋め込めることが必要になる。つまりロボットが作業座標系においてゴールに収束していくとき, センサ情報に基づく内部の状態も同様にあるアトラクタに収束していく(Fig. 1(b))。しかし, 一般にこ

Fig. 1 Embedding task trajectories into sensory-space

のような埋め込みができるかは保証されていない。例えば, ある軌道上の1点でのセンサ情報のみから作られる空間を内部空間としてマッピングしたとき, 作業座標系での軌道上の位置が異なるにも関わらず, センサ入力が同じである点が存在することがあり, 内部空間上での軌道はこの点では交差している(Fig. 1(c))。このような交差を解消するためには, 内部空間の次元を上げる必要がある。ここではセンサ情報の履歴に注目し, 内部空間の次元を上げることを試みる。埋め込みが出来れば, 各時刻の内部状態からロボットのタスクを達成するために最適な軌道の方向(すなわちモータコマンド)が直接決定できる。

本研究ではロボットのタスクの学習は適切なガイダンスの基に行われ, この学習を通して上述のセンサ情報に基づく内部空間からモータコマンドへのマッピングが形成される。結果として, ロボットがタスクを達成するための適切な行動は自己組織化されたアトラクタダイナミクスに従って実現される。

3. ナビゲーション

本章では具体的なナビゲーションの方法について, 説明する。

移動ロボットのナビゲーションは Fig. 2 のようにコントロールレベルとナビゲーションレベルの階層構造により実現される。コントロールレベルのメカニズムはあらかじめ与えられており, ダイナミクスは常に固定である。一方, ナビゲーションレベルは, 学習モードにおいて教師信号が与えられ, 適したコマンドが生成されるようにニューラルネットワークの学習が行われる。学習終了後の実行モードでは, そのニューラルネットワークが生成するコマンドにより, ロボットは目的とするタスクを達成する。

3.1 コントロールレベル

コントロールレベルは, ポテンシャル法に従い障害物を回避するスムーズな軌道を生成する。ロボットの姿勢は絶対座標系における位置とロボットの向き (x, y, θ) で表すことができる(Fig. 3)。ロボットはレンジセンサを持っていて, $-\alpha < \phi < \alpha$ の範囲で距離 R_ϕ を測定する。ただし ϕ はロボット正面を原点とする角度である。各時刻でのポテンシャル曲線 P_ϕ は, レンジセンサにより測定された距離 R_ϕ から

$$P_\phi = \begin{cases} k_3(R_\phi - R_p)^3 + k_2(R_\phi - R_p)^2 + k_1(R_\phi - R_p) & \text{if } R_\phi < R_p \\ 0.0 & \text{otherwise} \end{cases} \quad (1)$$

のように計算される。 R_p はレンジセンサが測定できる距離の範囲内での適当な閾値, 式(1)は測定された距離 $R_\phi < R_p$ において単調減少になるように定数 k_1, k_2, k_3 を決め, $R_\phi \geq R_p$ では

Fig. 2 Architecture of robot navigation

Fig. 3 Mobile robot with range sensor

値が0になる。すなわち、障害物との距離が遠い方向はポテンシャルが低くなる (Fig. 3 の右図)。ロボットはこのポテンシャル曲線のある極小点 ϕ_f に注目する (以降、この注目する極小点をフォーカスポイントと呼ぶ)。ロボットはフォーカスポイントをトラッキングするように、次式のダイナミクスに従って制御される。

$$\dot{\theta} = k_p \phi_f + \varepsilon \text{ obst}(\theta_{\min}, R_{\theta_{\min}}) \quad (2.a)$$

$$\dot{x} = v_0 \cos \theta \quad (2.b)$$

$$\dot{y} = v_0 \sin \theta \quad (2.c)$$

ここで、 k_p はコントロールゲインである。式(2.a)の第2項はロボットに最も近い障害物との距離 $R_{\theta_{\min}}$ がある距離 R_0 より近くなった時に受ける反力で、

$$\text{obst}(\theta_{\min}, R_{\theta_{\min}}) = \begin{cases} \lambda [h_3(R_0 - R_{\theta_{\min}})^3 + h_2(R_0 - R_{\theta_{\min}})^2 + h_1(R_0 - R_{\theta_{\min}})] & \text{if } R_{\theta_{\min}} < R_0 \\ 0.0 & \text{otherwise} \end{cases} \quad (3)$$

$$\text{where } \lambda = \begin{cases} 1.0 & \text{if } \theta_{\min} < \theta_f \\ -1.0 & \text{otherwise} \end{cases}$$

のように表される。 $\theta_{\min}$ は、最も近い障害物がある方向である。このコントロールレベルの動作により、フォーカスポイントが変化しない限り、ロボットは近い障害物を避けながらフォーカスポイントの方向へ一定速度 v_0 で移動する。

3.2 ナビゲーションレベル

一方、上位のナビゲーションレベルは、ポテンシャル曲線に新たな極小点が出現する分岐点において、どちらの方向に進むかを決定する。Fig. 4(a)はロボットの移動経路を、Fig. 4(b)はそのロボットの移動に対応したポテンシャルの時間変化を表して、ロボット中心の座標系でポテンシャルが低い方向は濃い黒で示されている。初期状態ではレンジセンサに基づくポテンシャル曲線には極小点が1つだけ存在するので、ロボットはその極小点をフォーカスポイントとし、先述のコントロールレベルのダイナミクスに従ってフォーカスポイントをトラッキングしながら進む。それに従い、ポテンシャル曲線もフォーカスポイントである極小点がロボット正面にくるように変化している。さらに移動してロボットがAの位置に到達する

Fig. 4 Travel of robot

と、右側に新しくオープンエリアが生じ、ポテンシャル曲線に新たな極小点ができ、極小点が2つになる。この分岐点において、ナビゲーションレベルはフォーカスポイントを新しい極小点の方向に遷移させるか、または元の極小点の方向に保つかを決定する。実際のロボットの実験においてはセンシングにおけるノイズの影響を除くために新たな極小点が生じてから適当な T_{lag} の時間、その極小点が保持された場合に分岐点と判断する。Fig. 4(a)の分岐点Aにおいてはフォーカスポイントを現在の極小点に保ち、さらにロボットを進ませる。同様の分岐点Bにおいてはロボットから見て左側にできた新しい極小点にフォーカスポイントを遷移させるとロボットは左に曲がり、Cにおいては再び左側の新しい極小点にフォーカスポイントを遷移させると再びロボットは左に曲がる。こうして、Fig. 4(a)のような移動経路をロボットは辿る。このナビゲーションレベルの判断を、センサデータを入力とするニューラルネットワークを用い、適切なコマンドを学習することによって、ロボットは目的とするタスクを実行できるようになる。

本手法の特徴は、低次の制御を分離することによってナビゲーションのタスクが有限回の移動方向の判断で達成できることにある。詳細な環境地図を作らずに、作業空間をセンサ入力の特徴量に基づいた粗視化によって有限分割し、ニューラルネットワークは各分割領域での次に進む方向を表す有限個の代表ベクトルのみを学習するだけで、目的のタスクを実行するための力学構造を近似的に獲得することができる。

4. ニューラルネットワーク

本章では、ナビゲーションレベルでのコマンドをニューラルネットワークの教師付学習を通して獲得する方法を説明する。

4.1 行動学習

ロボットに与えられたタスクを学習させるのは以下のように行う。

まず学習モードでは、任意の初期位置からロボットを動かし、分岐点に達するとロボットは停止する。そして、ロボットはタスクを達成するための最適なパスになるような指示をスーパーバイザーから受けて次に進む方向を決定する。ロボットはこれを繰り返して、タスクを達成するまで移動する。指示はフォーカスポイントを遷移させない時には0を、新しい極小点に

遷移させる時には1で表し、対応する分岐点でのセンサ情報と共に学習用データとする。これを初期位置を変えて何度か行い、学習用のデータを集める。その後、分岐点でのセンサ情報を入力とし、フォーカスポイントを遷移させるかについての指示を出力の教師信号として用い、ニューラルネットワークに学習させる。

学習が終了した後の実行モードでは、ロボットは任意の初期位置からスタートし、それぞれの分岐点においてはセンサデータを入力とするニューラルネットの出力が0.5以下なら0とみなしてフォーカスポイントはそのまま、0.5以上なら1としてフォーカスポイントを遷移させる。このようにナビゲーションレベルの判断をさせ、その指示された軌道の方にロボットを進ませると、目的とするタスクを達成することができる。

4.2 ニューラルネットワークの構成

ナビゲーションレベルのコマンド生成のためには2種類のニューラルネットワークを用いる(Fig. 5)。

Fig. 5 Neural Network

まず各分岐点でのセンサデータから得られるポテンシャル曲線のデータを Kohonen[6]が提案した Topological Preserving Map(TPM)によってベクトル量子化する(Fig. 5(a)). TPM は n 次元空間に配列しているユニットからなり、各ユニットは入力信号の次元(N 次元)と等しい次元の荷重ベクトルを持っている。入力があったとき、その入力信号と最も近い荷重ベクトルを持っているユニットだけが出力する。ここでは、そのユニットの n 次元で表されるインデックスを正規化して n 次元で出力する。また同時に、選ばれたユニットとその周辺のユニットの持つ荷重ベクトルを入力ベクトルに近くなるように学習を行う。学習後は近くのユニットの荷重ベクトルは近いベクトルに学習され、良く似た入力に対しては近くのユニットが出力するようになる。すなわち、冗長な情報を含むセンサデータがその分布に対応して情報圧縮される。

TPM の学習後、その出力を多層パーセプトロンに入力してナビゲーションレベルの意思決定をする。2章で述べたセンサ情報に基づく内部空間を構成するために我々は Fig.5(b)に示したような、Feed Forward Neural Network(FFNN)に明示的に過去の履歴データを空間的に展開して入力する方法と、Fig. 5(c)に示すような出力層から入力層に Feed Back 結合を持つことで、内部的に履歴を反映することのできる Recurrent Neural Network(RNN)を用いる方法を比較する。

FFNN では、Fig. 5(b)のように、あらかじめ決められた有限回数の分岐点でのセンサ入力の回帰列を入力とする。学習は通常の Back-Propagation(BP)法[7]を用いる。

RNN は出力層である第3層の一部に状態ユニット $S(t)$ があり、それが次の時刻に第1層の一部に入力される構造である(Fig. 5(c))。学習は Back-Propagation Through Time(BPTT)法[7]を用いる。BPTT は、時間方向につながっているネットワークを空間的に展開して、多層の FFNN とみなして BP 法を行う方法で、各時刻の出力および状態ユニットにおける誤差は

$$\delta_i(t) = \begin{cases} y_i(t) - q_i(t) & \text{if } i \in \text{out} \\ \sum_j \delta_j(t+1) \frac{\partial y_j(t+1)}{\partial y_i(t)} & \text{if } i \in \text{status} \end{cases} \quad (4)$$

で表される。これらの各時刻での出力誤差が状態ユニットを通して時刻を遡って伝搬される。

5. 学 習 実 験

山彦ロボット[8](Fig. 6)を使って学習実験を行う。山彦ロボットに3台の CCD カメラと赤外線レーザープロジェクタを搭載し、ロボットの前面 160° の範囲で24方向について障害物との距離を測定する。CCD カメラの画像データをトランスミッタでホストコンピュータである PC-9801 に送り、画像処理ボードで距離計算を行う。距離データからロボットのモータコマンドを DSP ボード上で計算し、RS-232 C の無線モデムでロボットに対しコマンドを送る。測距とコマンド計算は約 100[msec]毎に行われる。ロボットはそのコマンドに基づき、毎秒 30[cm/sec]で進む。

5.1 固定点のゴールの場合

FFNN を使って、固定のゴールに回帰するタスクを学習さ

せる。TPM の入力ベクトルは 24 次元、出力ユニットは 3 次元の格子に $6 \times 6 \times 6 = 216$ 個のユニットで構成されていて、各ユニットの出力ベクトルはそのユニットの 3 次元のベクトルで表されるインデックスに等しい。また、FFNN は TPM の現在の出力と 1 つ前の出力を入力とするので多層パーセプトロンの入力層は 6 個である。隠れ層のユニットは 6 個、出力層は 2 個で、出力は分岐点における判断を出力するユニットと、ゴールについたかどうかを判定するユニットである。

Fig. 6 YAMABICO Robot

Fig. 7 Training and testing of the homing navigation

Fig. 8 Homing test with additional obstacle

Fig. 7(a)は学習モードでの軌道 20 パターンで、ロボットに LED をつけて計測したものである。黒の 3 角で示した初期位置から、ゴールへの軌道をロボットは移動し、そのときのセンサデータとモータコマンド(ナビゲーションコマンド)を学習用のデータとする。学習が終わった後、そのニューラルネットワークを使ってロボットを移動させた結果が Fig. 7(b)である。図には 10 ヶ所の適当な出発点からゴールに向かう自律移動の様子が重ねて描かれている。Fig. 7(c)は、軌道 A, B, C のセンサ入力の時系列のデータと、分岐点での TPM の出力を示している。ロボットは、未学習の場所を移動しているとき、あるいはセンサ系や機械系のさまざまなノイズを受けたときなどに、行動(分岐の仕方)が一時的に不安定になる場合もあるが、大局的にはゴールへ安定に収束しているといえる。

また、作業空間内の環境を変えた時、ロボットの行動にどのような変化が生じるか実験、観察した。Fig. 8 に示したように、新たに未知の障害物が置かれた場合でも必ずしも最適とはいえないが、ロボットは最終的にはゴールまで到達できる。

これらのことからロボットはゴールに回帰するタスクを広い引き込み領域を持つ大域的な固定点収束の力学系として学習していることが分かる。

5.2 複雑な周期軌道の場合

次に周期軌道の学習実験を RNN を用いて行った (Fig. 9)。

ロボットは図の上側の多角形の障害物を常に時計回りに、下

Fig. 9 Training of the cyclic routing navigation

Fig. 10 Testing of the cyclic routing navigation

側のコの字の障害物の回りを1回おきに時計回りと反時計回りを交互に繰り返す, Fig.9のような複雑な軌道を周回する. この軌道上で多角形の障害物を回る間はいつもコントロールレベルの動作によりほぼ同じ軌道をロボットは進む. 従って Fig. 9(a)のA点で曲がるかどうかの判断には, そこでのセンサ情報だけでは不十分で, 一回前にコの字をどちら回りで回ったかを覚えていなければならない. FFNNでは, その時間回帰分の入力ユニットを用意しなければならないので, ユニット数が

多くなり, 学習および汎化が難しくなる. そこでRNNを使えば, 必要な履歴情報がコンパクトな形で状態ユニットに蓄えられるように自己組織化されることが期待される.

学習モードの軌道はFig.9(b)のような10パターンである. TPMの構造は前と同様で, RNNは状態ユニットを2個用意し, 入力にはTPMの出力と状態ユニットで5個, 中間層は8個, 出力層は分岐点での判断の出力と状態ユニットで3個という構成になっている.

学習後, 別の初期位置からスタートした時の軌道の例をFig. 10に示す. Fig. 10(a)は黒の三角を初期位置として白丸まで, 他の図は続いてある適当な時間間隔でロボットが黒丸から白丸まで移動する軌跡を表している. 学習させた周期軌道に収束して, タスクを達成している様子がわかる. すなわち, ロボットはこのタスクをリミットサイクルのアトラクタとして学習している.

6. 考察

本論文で提案した, 力学系に基づく行動学習の手法における我々の重要な主張は, 本手法がロボットの行動に大域的な安定性を与えるという点である. 分岐点でのロボットの行動は, 学習の不足及び実空間でのノイズなどの影響により, 局所的に乱れることもある. 従って, ロボットの行動は確率的な動作となるが, 分岐点での判断が10%程度間違えても, 本論文での実験環境ではタスク自身が大域的なアトラクタに埋め込まれている限りにおいては, それは時間の経過とともに達成される.

では, タスクが大域的なアトラクタに埋め込まれているかどうかの判断はどのようにすれば良いであろうか. 今回の実験では, 作業空間内の多くの位置からロボットをスタートさせ, タスクを達成できない場合があればその位置からの動作を再学習させる方法を取っている. 本実験程度の簡単な環境であれば, このような再学習を繰り返す方法で最終的にタスクを大域的アトラクタに埋め込むことができる. つまり, タスクがロボットの初期位置に依存せず, 常に達成できる状態が得られる. しかし, 環境がより複雑な場合については今後の検討課題の一つと考えられる.

7. むすび

すでに提案した自律移動ロボットの力学系に基づく新たな行動学習の手法を実際の山彦ロボットでの実験を通してその有効性を検証した. 実空間での実験において, ロボットは特定のゴールへ回帰するタスク, 特定の軌道を周回しつづけるタスクを, それぞれ固定点収束およびリミットサイクリングのアトラクタとしてその構造を学習した.

また, モータコマンドを生成するニューラルネットワークの構造として, FFNNに時系列データの構造を事前に与える方法と学習を通して必要な内部構造を自己組織化できるRNNを用いる方法の両者について実験を行なったが, RNNの方が複雑な環境やタスクに対しても入出力データの学習を通してコンパクトな形で構造が自己組織化されていることもわかった.

本論文で提案したロボットのナビゲーションの手法は作業空間の詳細なマップを作る必要がなく, 環境の変化に対してもロ

バーストであることが特徴として挙げられる。従って現実の応用においては、作業空間のマップが作り難い環境や、環境の一部が頻繁に変わるような場合、あるいはマップを作るための精度の良いセンサがない場合などにこの手法は有効であると考えられる。

今後はロボットが実環境においても動作するために、より複雑かつ広い環境で、多量のデータから最適な構造を獲得する問題や、過去に学習して得られた構造に新しく獲得した知識を加える方法などが課題として挙げられる。また、環境が複雑になれば最適軌道を外部から教わることも難しくなるので、ロボットが自ら環境内の軌道を探索して、学習するシステムも課題である。

謝 辞

移動ロボット“山彦”を実験に使用する機会を与えて下さった、筑波大学の油田信一教授に感謝致します。

参 考 文 献

- [1] V. Lumelsky et al: "Sensor-based terrain acquisition; a seed

spreader strategy," Proc. of IEEE/RSJ Int. Workshop on Intelligent Robots and Systems, pp. 62-67, 1989.

- [2] 渡辺, 油田: "灯台とデッドレコニングを用いた移動ロボットのポジショニング", In 第7回ロボット学会学術講演, pp. 41-44, 1989.
- [3] R. A. Brooks: "A robust layered control system for a mobile robot," IEEE J. of Robotics and Automation, RA-2(1): 14-23, 1986.
- [4] M. J. Mataric: "Integration of representation into goal-driven behavior-based robot," IEEE Trans. on Robotics and Automation, 8(3): 304-312, 1992.
- [5] J. Tani and N. Fukumura: "Learning goal-directed Sensory-based navigation of a mobile robot," Neural Networks, 7(3): 553-564, 1994.
- [6] T. Kohonen: "Self-organized formation of topographically correct feature maps," Biol. Cybern., 43, 59-69, 1982.
- [7] D. E. Rumelhart, G. E. Hinton and R. J. Williams: "Learning internal representations by error propagation," In D. E. Rumelhart and J. L. McClelland, editors, Parallel Distributed Processing, chapter 8, pp. 319-361, MIT Press, Cambridge, MA, 1986.
- [8] S. Yuta and J. Iijima: "State information panel for inter-processor communication in an autonomous mobile robot controller," In the IEEE International Workshop on Intelligent Robots and Systems (IROS '90) 1990.

福村直博 (Naohiro Fukumura)

1967年1月22日生。1989年東京大学工学部計数工学科卒。1991年同修士課程修了。同年SONY入社。ニューラルネットワーク、ロボットの運動制御の研究に従事。日本神経回路学会会員。
(日本ロボット学会正会員)

谷 淳 (Jun Tani)

1958年12月16日生。1981年早稲田大学理工学部機械学科卒業。同年千代田化工建設入社。1988年ミシガン大学大学院電気および機械修士課程修了。現在SONYコンピュータサイエンス研究所勤務。ニューラルネットワークおよび複雑系の研究に従事。電子情報通信学会、日本神経回路学会各会員。