

ノード評価値をもつ自己組織化関係ネットワークによる
知識獲得に関する研究

坂口俊輔

本論文で使用する記号の一覧表

\tilde{A}	ファジィ <i>if – then</i> ルールの前件部メンバーシップ関数
a	マイクロフォン間距離
a_i	ノードの活性度
B	ファジィ <i>if – then</i> ルールの後件部実数値
b_i	進化型自己組織化関係ネットワークにおける近傍係数
c	勝者ノードの番号
d_e	学習ベクトルと望ましい入出力間の距離
E, E_{thr}	学習ベクトルの評価値（入出力評価値）および閾値
ε	ノード評価値
h	近傍係数
I	学習ベクトル（入出力ベクトルペア）
i	ノードの番号
j	ノードの番号
k	出力参照ベクトルの要素番号
L, l	ベクトル集合の個数と番号
m	入力参照ベクトルの要素番号
n	試行信号
N	ノードの数（ファジィ <i>if – then</i> ルールの数）
$N_{c,0}$	適応型自己組織化関係ネットワークの近傍範囲に関するパラメータ
$N_{c,SOR}, N_{c,max}$	適応型自己組織化関係ネットワークにおける近傍範囲および近傍範囲が定義される最大値
N_{node}	ノード数
$N_{noise}, N_{noise,0}, N_{noise,max}$	試行信号の振幅，振幅に関するパラメータおよび振幅が定義される最大値
N_x	入力参照ベクトルの要素数
N_y	出力参照ベクトルの要素数
nec, nec_{max}	学習必要性尺度および学習必要性尺度が定義される最大値
S, S^{th}	満足率および満足率の閾値
s	リンク結合強度
T	転置記号
T_p, t	任意の学習ステップ数および学習ステップ
t_l	満足率を算出するステップ数
r_c, r_i	競合層上における勝者ノードおよび競合層ノードの座標
u_i	出力参照ベクトル
V, v_i	参照ベクトル集合および参照ベクトル (入力参照ベクトルおよび出力参照ベクトルの結合ベクトル)

v_s	音速
\tilde{w}, \tilde{w}	ファジィ <i>if - then</i> ルールの前件部メンバーシップ関数
w_i	入力参照ベクトル
x_A	両耳間時間差
$x_V, x_{V,rot}$	画像上の音源のピクセル値, 回転後の画像上の音源のピクセル値
$x_{V,center}$	画像の中心のピクセル値
$x_{V,border}$	中心窩の最大受容範囲を決定する画像上のピクセル値の幅
x_l, x^*	入力ベクトルおよびテスト用の入力ベクトル
y_l, y^*	出力ベクトルおよび出力ベクトルの推定値
z_i	類似度もしくは活性度
α, α_{SOR}	引力学習の学習係数
α_{max}	適応自己組織化関係ネットワークにおける引力学習係数の最大値
α_u	動的自己組織化関係ネットワークにおける出力重みベクトルの学習率
α_w	動的自己組織化関係ネットワークにおける入力重みベクトルの学習率
α_0	学習必要性尺度に関するパラメータ
β, β_{SOR}	斥力学習の学習係数
β_{max}	適応型自己組織化関係ネットワークの斥力学習の学習係数が定義される最大値
β_0	適応型自己組織化関係ネットワークにおける斥力学習に関するパラメータ
γ	類似度関数の幅を決定するパラメータ
δ	参照ベクトルの活性度に関する忘却係数
η	学習必要性尺度に関するパラメータ
η_+, η_-	学習必要性の増加および減少に関するパラメータ
θ	カメラの正面からの音源の相対角度
θ_{rot}	ロボットの回転角度
$\theta_V, \theta_{V,border}$	視野角度および中心窩の最大受容角度
μ_i	メンバーシップ関数との適合度
π	円周率
ρ	正規化活性度
σ	近傍関数の幅を決定するパラメータ
σ_r	斥力学習が及ぶ範囲を規定するパラメータ
σ_{noise}	試行信号の振幅を決定するパラメータ
σ_{trial}	ガウスノイズの評価値依存性を決定するパラメータ
σ_{α_u}	学習係数 α_u の関数の傾きを決めるパラメータ
$\tau_\alpha, \tau_\beta, \tau_\sigma$	学習回数および近傍の幅を決定するパラメータの減衰率
ϕ	受容野半径
Ω	近傍集合
ω	リンク結合強度に関する忘却係数
0	零ベクトル

目次

本論文で使用する記号の一覧表	i
第1章 序論	1
第2章 自己組織化関係ネットワークと適応型自己組織化関係ネットワーク	9
2.1 はじめに	9
2.2 自己組織化関係ネットワーク	10
2.2.1 自己組織化関係ネットワークの学習モード	12
2.2.2 自己組織化関係ネットワークの実行モード	17
2.3 二つの適応型自己組織化関係ネットワーク	21
2.3.1 進化型自己組織化関係ネットワーク	22
2.3.2 適応自己組織化関係ネットワーク	25
2.4 おわりに	28
第3章 ノード評価値をもつ自己組織化関係ネットワーク	31
3.1 はじめに	31
3.2 従来の斥力学習の問題点とその解決策	31
3.3 動的自己組織化関係ネットワーク	34
3.3.1 知識獲得ブロックの処理手順	36
3.3.2 推論ブロックの処理手順	38
3.4 おわりに	39
第4章 動的自己組織化関係ネットワークの音源定位学習への応用	41
4.1 はじめに	41
4.2 実験の枠組み	43
4.3 音源定位能の獲得	48
4.4 適応学習	57
4.5 考察	63

4.6 おわりに	64
第 5 章 結論	67
謝辞	71
参考文献	73

第1章 序論

本論文は、自律ロボットの柔軟かつ効率的な行動学習の実現を目的とし、新たな学習手法の提案およびその自律ロボットへの応用に関する研究についてまとめたものである。

近年、生物の脳の持つ優れた情報処理機能を工学的に応用する「脳情報工学」という学問分野が確立されつつある [1]。中でも、自律ロボットの行動学習は、環境の認識、推論、意思決定、身体制御など、脳の持つ優れた情報処理機能を総合的に扱う重要な問題であり、盛んに研究が行われている [2]。人間の高い運動能力は、学習によって後天的に獲得される [3]。人間は行動学習において、自分のおかれている状態と、その状態における最適な行動の間の関係を学習する。ここで、最適な行動は、試行錯誤によって自律的に獲得される。このような学習過程を経て、身体の大きさや感覚器の特性など、身体的な条件に対して最適な制御能力を獲得する。そして、身体の成長や老化など、物理的特性の変化に対応するために、生まれてから死ぬまでの間、絶えず学習を続ける。

ロボットの行動学習においても生物の場合と同様に、感覚入力とそれに対する運動出力との間の入出力関係を学習する必要がある。望ましい入出力関係を学習するための脳のモデルは数多く提案されているが、主に、教師あり学習と評価値付き学習に分けられる。教師あり学習手法においては、入力に対するネットワークの出力が教師信号に近づくように、ネットワークの状態を調節することによって望ましい入出力関係を獲得する。ここで、教師信号とは、入力に対する望ましい出力を指す。教師あり学習手法の代表的なものには、階層型ニューラルネットワーク [4; 5; 6; 7; 8; 9] やファジィニューラルネットワーク [10; 11] などがある。これらの手法は、予め教師信号が得られるような場合に有用である。しかし、未知環境における試行錯誤的な行動学習において、望ましい行動を予め知ることは難しい。一方、明示的な教師信号を必要としない学習手法として、評価値付き学習がある [12; 13; 14; 15; 16; 17]。評価値付き学習は、入力に対する望ましい出力は得られ

なくとも，出力に対する何らかの評価を得ることができれば，適用することが可能である．評価値付き学習において設計者は「何をすべきか」という目的を示すだけで「どのように実現するか」という規則は学習によって自動的に獲得される．したがって，評価値付き学習手法は，未知環境において学習を行う自律ロボットの行動学習に適していると考えられる．評価値付き学習の代表的な手法としては，強化学習 [12; 13]，および自己組織化関係ネットワーク (SOR: Self-Organizing Relationship Network) が挙げられる [14; 15; 16; 17]．

強化学習において，入力および出力は，それぞれ『状態』および『行動』と呼ばれる．強化学習では，試行錯誤を通じて得られる報酬と呼ばれる情報を手掛かりに，状態と行動の間の望ましい関係を学習する．報酬とは，各状態に対する行動の望ましさに応じて環境から与えられるスカラの情報である．一般に，各状態に対して，高い報酬が得られる行動を学習する．しかし，実環境で動作するロボットが取りうる状態の数は事実上無限にあるので，強化学習では状態空間が膨大になる，いわゆる「次元の呪い」と呼ばれる問題が生じる．これによって，必要な計算量が指数関数的に増加してしまう．

SOR は，望ましい入出力関係をファジィ *if-then* ルールの集合として獲得する学習機能と，獲得したファジィ *if-then* ルールの集合を用いてファジィ推論を行う機能を併せ持つ．SOR は，試行錯誤の結果得られたいくつかの事例から本質的な *if-then* ルールの集合を抽出し，それらを用いて望ましい入出力関係を表現できる．したがって，次元の呪いの影響がなく，記憶容量や処理速度の面からみて非常に優れている．SOR は，自己組織化マップ (SOM: Self-Organizing Map) [18; 19] を拡張した学習手法であり，競合層上のノード*およびノードに関連づけられた参照ベクトルを持ち，自己組織化学習を行う．各ノードは，それぞれひとつのファジィ *if-then* ルールを表現する．SOR の学習においては，SOM と同様に，学習係数や近傍範囲などの学習パラメータを時間的に減衰させることによって学習を収束させる．SOR において学習用のデータは，入力ベクトルと出力ベクトルのペアで表され，これを『学習ベクトル』と呼ぶ．設計者は，試行錯誤によって得られた学習ベクトルに対して，その良否を評価する．ここで，評価は -1 から 1 までの範囲をもつスカラで表され，『評価値』と呼ばれる．評価値の正と負の値は，それぞれ望ましい入出力関係および望ましくない入出力関係を表す．SOR は，与えられた学習ベクトルとその評価値をもとに学習を行う．このように，SOR は，学習の際に設計者の主観的な評価を積極的に利用する

*本論文で用いる『ノード』は，自己組織化マップや従来の SOR における『ユニット』と同義である．

ことによって、効率的に学習を行うことができる。さらに SOR では、評価関数を用いて客観的に評価を行うことも可能である。このように SOR は、主観的評価および客観的評価のいずれにも対応できるので、様々な問題へ適用することが可能になる。

SOR の最大の特長は、望ましくない入出力関係も学習に利用できることである。これによって、(1) 学習データ取得のためのコスト削減が可能、(2) 制御不能状態・危険領域の回避 [20] が可能、などの効果が期待できる。SOR では、負の評価値をもつ学習ベクトルが与えられた場合、参照ベクトルが学習ベクトルから遠ざかるように更新される。これを『斥力学習』と呼ぶ。従来の SOR における斥力学習では、負の評価値を持つ学習ベクトルから参照ベクトルが単純に遠ざかるように更新される。この方法は、バッチ学習の際には有効に機能する。しかし、逐次学習においては、参照ベクトルは必ずしも望ましい入出力関係を表現するように更新されるとは限らない。この問題は、入出力空間における評価値の分布が未知であることに起因する。逐次学習において SOR は、学習ベクトルとその評価値しか与えられない。これでは、評価値の勾配が分からないので、その評価値をより高くするための更新方向が分からない。この問題を解決するために、本研究では、学習ベクトルの評価値に加えて、各参照ベクトルの評価値を表す新たなパラメータを導入する。学習ベクトルの評価値と各参照ベクトルの評価値を比較し、評価値が高くなる方向へ参照ベクトルを更新することによって、より望ましい入出力関係を表現できるようになると考えられる。具体的には、SOR の各ノードが表現する *if - then* ルールの望ましさを表すパラメータとして『ノード評価値*』を新たに導入する。これを用いると、斥力学習において常に望ましい更新が可能となる。これに加えて、以下の二つの機能も導入する。

- (1) 時変システムに対する適応学習機能
- (2) 試行錯誤による学習データ生成機能

従来の SOR は、環境の変化に対する適応性を持たない。これは、主に二つの特徴に起因する。一つ目は、*if - then* ルールを表現するノードの数を予め決定しておく必要があること、二つ目は、時間と共に減衰する学習係数が用いられていることである。自律ロボットを未知の環境へ適用する場合、適切なルール数を事前に知ることは難しい。あらゆる環境に適用できるようにするためには、ルール数が柔軟に変更できる事が望ましい。また、学習を一端完了した後も、環境の変化

*先に述べたように、各ノードは、それぞれ一つの参照ベクトルに関連づけられている。したがって、ノード評価値は、参照ベクトルの評価値を意味する。

や、ロボットの劣化または修理などによってロボットの特性が変化する可能性が考えられる。これらの変化に適応するためには、ロボットが動作し続ける限り、必要に応じて再学習できなければならない。このSORに対して提案手法は、(1)に示すように、時変システムに対する適応学習機能を導入する。具体的には、(a) ノードの追加・削除、(b) 環境に対する *if-then* ルールの集合の望ましさに応じた学習パラメータの調節、という二つの機能を導入する。これら二つの機能は、進化型自己組織化関係ネットワーク (ESOR: Evolving SOR) [21] の進化的学習アルゴリズムを導入することによって実現する。提案手法では、ノードは学習ベクトルの分布およびその入力順序に応じて、追加および削除される。これによって、適切な数の *if-then* ルールを得ることができる。さらに提案手法では、学習係数や試行の大きさを決める係数などのパラメータを、ノード評価値に応じて自動的に調節することが可能である。これによって、適応学習を行うことが可能になる。

さらに従来のSORでは、学習に際して、学習ベクトルおよびその評価値を予め用意する必要があった。しかし、自律ロボットは、学習ベクトルの獲得、学習ベクトルに対する評価、学習（参照ベクトルの更新）を、全て自律的に行わなければならないので、従来のSORをそのまま適用することは難しい。そこで、提案手法では、(2)に示すように、試行錯誤による学習データ生成機能を導入する。具体的には、適応自己組織化関係ネットワーク (ASOR: Adaptive SOR) [17; 22] のもつ、学習用データの自己生成機能を導入する。これによって提案手法では、自律的に生成した信号を用いて、望ましい入出力関係を試行錯誤的に獲得することが可能になる。

以上をまとめると、提案手法は、

1. 望ましくない入出力関係をも学習に利用する柔軟性
2. ロボット自身が生成および獲得した情報のみに基づいて学習できる自律性
3. 環境およびロボット自身の変化に対する適応性

という三つの特長を兼備した学習手法であると言える。これらの特長から、提案手法を、動的自己組織化関係ネットワーク (DSOR: Dynamical Self-Organizing Relationship Network) と名付けた。DSORは、上記のような特長によって、環境やロボットシステム自体に関する知識をまったく持たない状態においても、自律的かつ効率的な試行錯誤によって、ゼロから行動を獲得することができる。このような状況は、生まれたばかりの赤ちゃんが基本的な動作を獲得する過程に

似ている．そこで，ロボットが基本的な動作を自律的に獲得する問題へ DSOR を適用し，その有効性を検証する．

上記のような問題の代表的な例として，物体の認識が挙げられる．人間は，視覚の動物と呼ばれるほど視覚優位性が高く，対象物を見ることによって物体の認識や識別を行う．特に網膜上の中心窩と呼ばれる部分は，網膜上の他の部分に比べて空間分解能が高いので，人間が視覚的に物体を認識しようとする場合には，注目する物体を中心窩でとらえるように首や眼球を動かす．さらに，このような行動は，視野外の対象物体に対しても，音を手掛かりに行うことができる．これによって，身体の周りの全ての領域に渡って瞬時に注目することが可能となる．ここでは，音を発する対象物体を視覚的に認識するためのこのような行動を音源定位と呼ぶ．ロボットが実環境において動作する場合も人間と同様に，注目する対象物体を視覚的に認識する事は重要な能力の一つである．ロボットに音源定位機能を実装することによる効果として，視覚情報を用いた対象物体の認識率の向上，人間とロボットのより自然なインタラクションの実現などが期待できる．そこで本論文では，ロボットの音源定位能獲得について考える．

ロボットのための音源定位機能の実現方法には，マイクロフォンなどのセンサから得られた信号をどのように処理するかによって大きく分けて二つのアプローチがある．一つは，解析的に処理する方法，もう一つは，生物の神経回路モデルなどを用いて学習によって音源定位を実現する方法である．解析的な手法を用いる場合，マイクロフォンアレイなど，多数のセンサから得られた信号を用いて処理が行われることが多い．多数のセンサ信号を用いる利点としては，空間分解能が高くなることや複数の音源信号を定位できるようになることなどが挙げられる．しかし，マイクロフォンアレイを用いたロボットシステムは，センサ数が多くなる，センサ形状が固定される，センサ数に比例して計算コストが増加し専用ハードウェアが必要になる，センサ数や専用ハードウェアの使用によって製作コストがかかる，などハードウェアの制約が大きい．さらに，ロボットの形状やセンサの特性に合わせた数理モデルを考える必要があり，実用化に対する問題が多い．一方，学習によって音源定位能を獲得する手法は，一般的に二つのマイクロフォンのみで実現可能，ロボットの形状やセンサの特性に左右されない，などの特長をもつ．これらの特長によって，ハードウェアコストの削減や，ロボット形状の設計に対する自由度の向上が期待できる．自律ロボットの音源定位学習では，主に，以下に示す三つの問題点を考慮する必要がある．

1. 外部からの教師信号が得られない

ロボットは、自律学習を行うために、センサから得られた入力信号とそれに対するモータの回転角度との間の関係を自ら評価する必要がある。

2. 単一の感覚情報のみでは学習できない

聴覚情報のみでは、音源の位置を知ることができないので、視覚情報によって音源との位置関係を知る必要がある。しかし、視覚は聴覚と違って認識できる領域が限られているので、視野外の音源を定位するためには聴覚情報に基づく定位能力が必要になる。これを解決するためには、視覚と聴覚の両方の感覚情報が必要である。

3. 環境やロボット自体の特性が経時変化する

自律ロボットが行動する事によって、その動作環境は時々刻々と変化する。また、ロボットが行動することによって、センサや効果器などは劣化し、特性が変化する可能性がある。これらの変化に対する適応性が必要とされる。

ロボットの自律的な学習によって音源定位能を獲得する手法はいくつか提案されている [23; 24; 25; 26; 27]。特に文献 [27] で提案された手法は、音源が視野外に存在する場合でも、視覚と聴覚の情報を効果的に用いて音源定位能を獲得することができる。しかし、そこでは、視聴覚情報とモータの回転角度の間の入出力関係を学習するために、古典的な SOM の学習則を用いている。これに起因して、教師信号が必要、適応学習ができない、などの問題点が残る。また、聴覚情報と運動との間の入出力関係を学習するためには、視覚情報と運動との間の入出力関係を予め学習しなければならない、学習効率が悪いという問題もある。そこで本研究では、上で述べた自律ロボットの音源定位学習に関する三つの問題点を解決するために DSOR を適用し、その有効性を検証する。先に述べたように、DSOR は、学習データの獲得およびそれに対する評価をロボットが自律的に行うことができる。これによってロボットは、センサ入力とモータの制御量間の望ましい入出力関係を試行錯誤的に獲得できるようになる。また、環境や音の特徴量などについて未知の状態から学習を開始するので、学習初期においては、望ましい行動を行うことは難しい。これに対して DSOR は、望ましくない入出力関係を有効に利用した学習が可能なので、望ましい入出力関係を得ることができない学習初期段階から、効率的な学習を行う事が可能である。このような効率的な学習は、ロボットの物理的な消耗を抑えるために有効に働くと考えられる。DSOR を用い

ることによるこれらの有効性によって、効果的かつ効率的な音源定位能の獲得を実現する。

本論文は、五つの章からなる。

第1章は序論である。

第2章では、提案手法の基礎となるSORおよび、二つの適応型SORについて、それぞれの特徴および基本的な動作について説明する。

第3章では、第2章で述べた適応型SORを自律ロボットの学習に用いる際の問題点についてまとめた後、その問題を解決するための手法として、ノード評価値を導入した提案手法について述べ、その特長と基本的な動作についてまとめる。

第4章では、提案手法を音源定位能の自律獲得問題へ応用し、その有効性について検討する。

第5章は結論である。

第2章 自己組織化関係ネットワークと適応型自己組織化関係ネットワーク

2.1 はじめに

自己組織化関係ネットワーク (SOR: Self-Organizing Relationship Network) は, 自己組織化マップ [18; 19] のもつ自己組織化学習の機能と, カウンタープロパゲーション [28] のもつ入出力関係の学習機能を併せ持つ手法として提案された [14; 15; 16; 17]. SOR の最も大きな特長は, 『評価値付き学習』と呼ばれる学習手法である. SOR の学習用データは, 設計対象とするシステムから得られた入出力ベクトルペアおよび, 入出力ベクトルペアに対する評価値によって構成される. 評価値は, それぞれの入出力ベクトルペアが表現する入出力関係の望ましさを数値化したものであり, 一般に $[-1, 1]$ の範囲をとり, 望ましい場合は正の値, 望ましくない場合は負の値をとる. SOR は, 評価値を用いることによって, 望ましい入出力ベクトルペアだけでなく, 望ましくない入出力ベクトルペアも有効に利用して学習することが可能である. この特長によって SOR は, 明示的な教師信号なしに, 設計対象とするシステムの望ましい入出力関係を獲得できる. 学習後, SOR は簡略型ファジィ推論エンジンとして動作し, 入力ベクトルに対する望ましい出力ベクトルを生成する. これまでに SOR は, 主観的評価に基づく画像強調 [29] や客観的評価に基づく制御設計 [16; 22] などへ応用され, その有効性が示されている.

上記の通り, SOR は非常に優れた特性を持つが, 自律ロボットの行動学習への適用に対しては以下のような問題点がある.

- (1) ルール数を予め規定する必要がある.
- (2) 適応学習能力を持たない.
- (3) 学習用データ集合を予め獲得しておく必要がある.

これらの問題点に対して，二つの適応型 SOR が提案されている．進化型自己組織化関係ネットワーク (ESOR: Evolving SOR) [21] は，SOR に進化学習アルゴリズム [30; 31] を導入した学習手法である．ESOR は，競合層を持たず，ノードを自由に追加および削除することが可能になった．各ノードは SOR と同様に，一つの *if-then* ルールを表現している．つまり，ESOR は，(1) の問題に対して，学習する環境に応じた適切なルール数で入出力関係を表現することが可能になった．さらにこの特長は，入力データを一度提示しただけで学習が完了するワンパス学習を可能にした．さらに ESOR は，学習係数や近傍係数などのパラメータのスケジューリングが不要であり，継続的な学習が可能である．ESOR はノードの追加・削除機能およびパラメータのスケジューリングが不要という二つの特長によって，高い適応学習を実現することが可能であり，(2) の問題に対して十分な能力を有すると言える．問題 (3) に対しては，適応自己組織化関係ネットワーク (ASOR: Adaptive SOR) [22] が提案されている．ASOR は，SOR の実行モードと学習モードを交互に繰り返す事によって逐次的に学習用データを取得できる．また，出力信号にノイズを加え，試行錯誤的に望ましい入出力関係を生成することができるので，ASOR は自律学習が可能である．

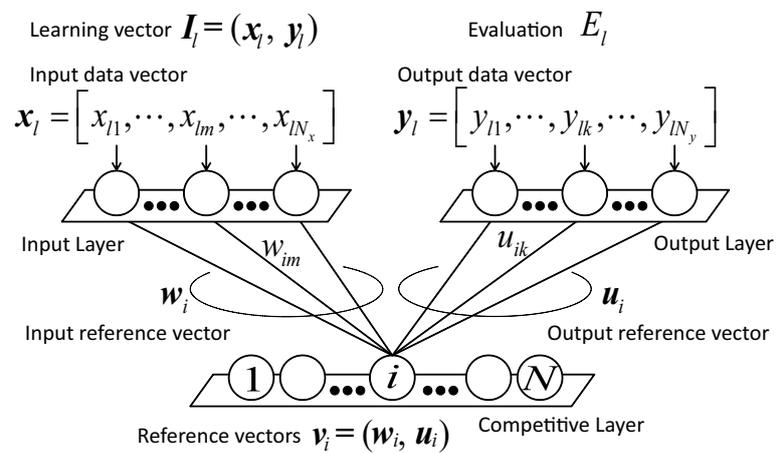
本章ではまず，2.2 において，SOR の基本的な特徴およびその動作について述べる．その後，2.3 において，ESOR および ASOR についてまとめる．

2.2 自己組織化関係ネットワーク

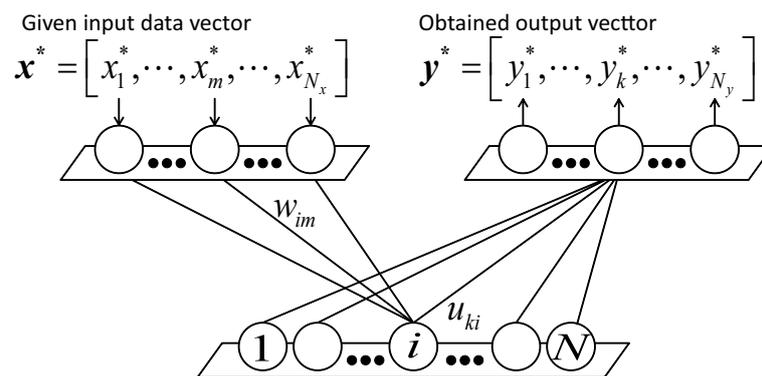
SOR の構成を図 2.1 に示す．SOR は，入力層 (Input layer)，出力層 (Output layer)，競合層 (Competitive layer) によって構成される (図 2.1(a))．競合層上の各ノードは，入力層および出力層との間に関連づけられた参照ベクトル集合 v_i を持つ．参照ベクトル v_i は，入力ベクトル空間における参照ベクトル w_i と出力ベクトル空間における参照ベクトル u_i を連結したベクトルであり，

$$v_i = (w_i, u_i), \quad i = 1, \dots, N \quad (2.1)$$

で表される． w_i, u_i は，それぞれ N_x 個， N_y 個の要素を有し，その添え字をそれぞれ m, k とする．学習に用いる入出力ベクトルペア I_l は，試行錯誤によって得られる入力ベクトル x_l と出力ベ



(a)



(b)

図 2.1: 自己組織化関係ネットワークの概念図 . (a) 自己組織化関係ネットワークの学習モード . (b) 自己組織化関係ネットワークの実行モード .

クトル y_l のペアであり，次式で表される．

$$I_l = (x_l, y_l), \quad l = 1, \dots, L \quad (2.2)$$

x_l と y_l は，各参照ベクトルと同様に，それぞれ N_x 個， N_y 個の要素を有し，その添え字をそれぞれ m, k とする．以後，入出力ベクトルペア I_l を学習ベクトルと呼び， l は省略して記述する．各学習ベクトルは，評価値 E をもつ．評価値 E は，設計者の主観や定量的な評価規範に基づいて，学習前に与えられる．通常，評価値 E は $[-1, 1]$ のスカラ値をとる．正の評価値を持つ学習ベクトルがネットワークに提示されると，各参照ベクトルは，その学習ベクトルに近づくように更新される．これを『引力学習』と呼ぶ．一方，負の評価値を持つ学習ベクトルが提示された場合，各参照ベクトルは学習ベクトルから遠ざかるように更新される．これを『斥力学習』と呼ぶ．SOR は斥力学習によって，望ましい事例からだけでなく，望ましくない事例からも学習を行うことができる点が大きな特長である．

SOR の動作は，大きく分けて二つの動作モードから構成される．与えられた学習ベクトルを用いて望ましい入出力関係を獲得する学習モード（図 2.1(a)）と，学習後，実際の入力に対して望ましい出力を生成する実行モード（図 2.1(b)）である．SOR は，学習モードにおいて，入力と出力間の望ましい写像関係 $y^* = f(x^*)$ を，学習によって獲得する．学習後，SOR は，実行モードにおいて，ファジィ推論エンジンとして働き，実際の入力に対して望ましい出力を生成することができる．

2.2.1 および 2.2.2 では，SOR の学習モードおよび実行モードについてそれぞれ説明する．

2.2.1 自己組織化関係ネットワークの学習モード

学習モードの処理は，学習ベクトルの取得，学習ベクトルの評価，そして評価値に基づく自己組織化学習の三つに分けられる．設計者はまず，SOR を用いて設計を行う対象となる関数近似器，制御器，フィルタなどを試行錯誤的に動作させることによって学習ベクトルを獲得する．次に，学習に使用する L 個の学習ベクトルと評価値のセット

$$\{x_l, y_l; E_l \mid l = 1, 2, \dots, L\} \quad (2.3)$$

を用意するために、学習ベクトルの評価を行う。ここで、評価値 E_l は、入力ベクトル x_l と出力ベクトル y_l の関係に対する評価である。評価値 E_l は、設計しようとする対象の性質に応じて、定性的もしくは定量的に与えられる [17]。学習ベクトルに対する評価値の与え方を図 2.2 に示す。図 2.2(a) は、SOR によって関数近似器を構築しようとする場合を表す。この場合は、試行錯誤によって得られた入力ベクトルと出力ベクトルの関係を直接評価し、評価値を与える。一方、制御器やフィルタ等を試行錯誤によって構築する場合には、入力ベクトルと出力ベクトルの関係を直接評価することはできない。そこで、図 2.2(b) に示すように、制御における離散サンプリング時刻 t における Plant の状態に対して、SOR からの出力によって $t+1$ 時刻における Plant の状態が望ましい状態に遷移したか否かによって評価を行う。ここで得られた評価値は、入力ベクトル x_l と出力ベクトル y_l の関係に対する評価として与えられる。このような手順によって得られた学習ベクトルと評価値を SOR に提示すると、学習によって望ましい入出力関係を表すクリスプ *if-then* ルールが競合層ノードに獲得される。以下に、望ましい入出力関係を獲得するための具体的な学習の手順を示す。図 2.1(a) に学習モードの処理の概念図を示す。

手順 0 参照ベクトルの初期化を行う。通常、学習ベクトル集合に関する事前情報がない場合には、乱数を用いて全ての参照ベクトルの値を決定する。

手順 1 学習ベクトル集合の中から一つの学習ベクトル I_l を選択し、すべてのノードに提示する。

手順 2 学習ベクトルとの距離（通常、ユークリッド距離を用いる）が最小となる参照ベクトルを持つ競合層のノード c を次式で決定し、勝者ノード c とする。

$$c = \arg \min_i \sqrt{\sum_{m=1}^{N_x} (w_{im} - x_{lm})^2 + \sum_{k=1}^{N_y} (u_{ik} - y_{lk})^2} \quad (2.4)$$

競合層上において、勝者ノードの近傍に位置するノードを近傍ノードとする。ここで、近傍ノードの参照ベクトルの更新度合いを決定する近傍係数 $h_{c,i}(t)$ を次式に従って算出する。

$$h_{c,i}(t) = \exp\left(-\frac{\|r_c - r_i\|^2}{2\sigma^2(t)}\right) \quad (2.5)$$

ここで, t は学習ステップを表し, r_c と r_i は競合層上における勝者ノードおよび競合層ノードの座標を表す. $\sigma(t)$ は近傍の幅を決定するパラメータである.

手順 3 勝者ノードおよび近傍ノードの参照ベクトルを次式に従って更新する.

$$\mathbf{v}_i(t+1) = \begin{cases} \mathbf{v}_i(t) + \alpha(t)h_{c,i}(t)E_l(\mathbf{I}_l - \mathbf{v}_i(t)) & \text{for } 0 \leq E \\ \mathbf{v}_i(t) + \beta(t)h_{c,i}(t)E_l & \\ \exp\left(-\frac{\|\mathbf{I}_l - \mathbf{v}_i(t)\|}{\sigma_r}\right) \frac{\mathbf{I}_l - \mathbf{v}_i(t)}{\|\mathbf{I}_l - \mathbf{v}_i(t)\|} & \text{for } E < 0. \end{cases} \quad (2.6)$$

ここで, $\mathbf{v}_i(t+1)$, と $\mathbf{v}_i(t)$ はそれぞれ更新後と更新前の参照ベクトルを表す. $\alpha(t)$ および $\beta(t)$ はそれぞれ, 学習ステップ t における引力学習および斥力学習の学習係数である. σ_r は斥力学習が及ぶ範囲を規定するパラメータである.

手順 4 学習ベクトル集合の中に一度も選択されていない学習ベクトルがある場合は, 手順 1 に戻り, これまでに選択されていない学習ベクトルを提示する. 学習ベクトル集合内の全ての学習ベクトルを選択した場合は, 手順 5 に進む.

手順 5 手順 1~手順 4 の操作を規定回数繰り返す. 繰り返す過程で, 学習係数 $\alpha(t)$, $\beta(t)$ および $\sigma(t)$ を小さくしていく.

式 (2.6) の参照ベクトル \mathbf{v}_i の更新式は, 提示された学習ベクトル \mathbf{I}_l の評価値 E_l が正の場合, 参照ベクトル \mathbf{v}_i を学習ベクトル \mathbf{I}_l に近づく方向に更新し, 評価値 E_l が負の場合, 学習ベクトルから遠ざかる方向に更新する. 両方の更新において, 参照ベクトルの更新量は, 評価値の絶対値に比例する. 学習ベクトルと参照ベクトル間の距離に関しては, 評価値が正の場合は距離が大きいほど更新量を大きくし, 評価値が負の場合は距離が大きいほど更新量を小さくする. 学習ベクトル - 参照ベクトル間の距離と更新量の関係を図 2.3 に示す. 学習後の参照ベクトルは, 評価の悪い学習ベクトルから遠ざかり, 評価の良い学習ベクトルの存在する領域に分布される.

例として,

$$y = 0.5 \sin(\pi x_1) + 0.5 \cos(0.5\pi x_2) \quad (2.7)$$

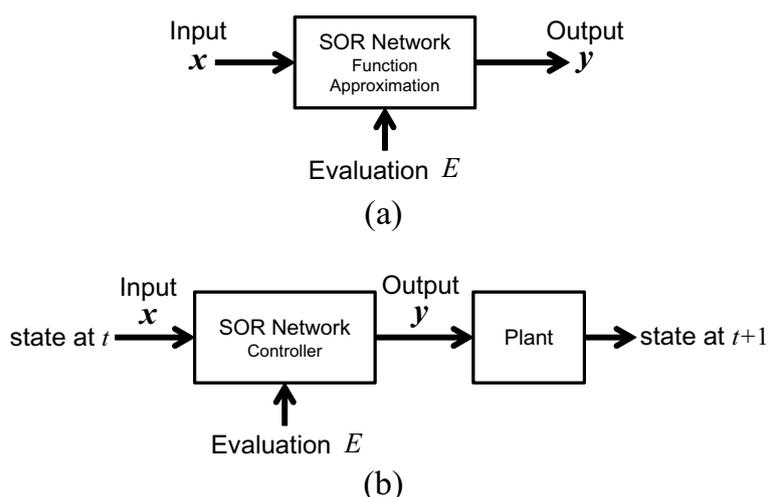


図 2.2: SOR ネットワークの学習ベクトルの評価方法．(a) 関数近似器を構築する場合の例．(b) 制御器を構築する場合の例．

で表される望ましい入出力関係を学習によって獲得した結果を示す．望ましい入出力間の写像関係を図 2.4(a) に示す．学習に用いるデータとして，図 2.4(b) に示す 500 個の学習ベクトルを考える．入力ベクトルは，一様分布 $U_{x_1}[-1:1]$ ， $U_{x_2}[-1:1]$ から獲得した．各入力ベクトルに対する出力は一様分布 $U_y[-1:1]$ から得られた値で決定した．入出力ベクトルペアは，以下の評価関数で評価した．

$$E = 1 - 10d_e \quad (2.8)$$

ここで， d_e は学習ベクトルから望ましい入出力間の写像関係を示す曲面 (図 2.4(a)) までの距離である．評価値は，定性的または定量的に決定できるが，ここでは，SOR の学習の理解を助けるために，理想的な入出力間の写像関係を仮定して定量的に決定した．図 2.4(b) 中， \times と \circ は，それぞれ評価値が正および負の評価値をもつ学習ベクトルを表し，各 250 個ずつである．これらの学習ベクトルと評価値を用いて SOR の学習を行なった．競合層ノードは 100 (10×10) 個で 2 次元に配置した．学習回数は，500 個の入力ベクトル全てを 1 度ずつ提示することを学習の単位とし，1000 回繰り返した．学習係数と近傍範囲は，次式に従い，指数関数的に単調減少させた．

$$\alpha(t) = (\alpha(0) - \alpha(final)) \exp\left(-\frac{t}{\tau_\alpha}\right) + \alpha(final) \quad (2.9)$$

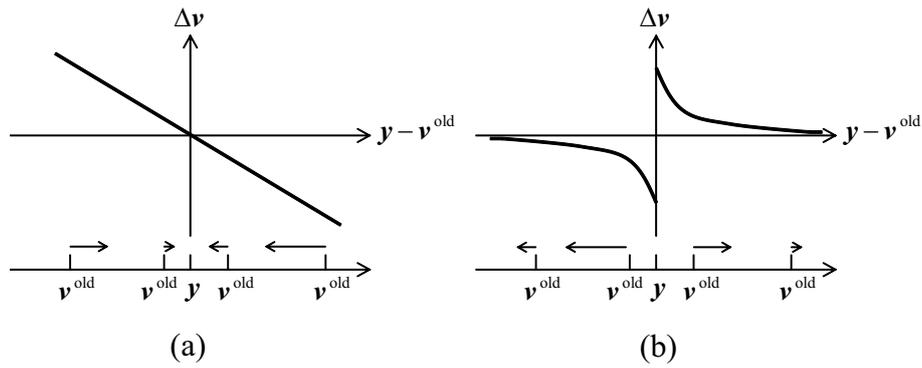


図 2.3: 自己組織化関係ネットワークにおける参照ベクトルの更新量 . (a) 与えられた学習ベクトルの評価値が正の場合 (引力学習) . (b) 与えられた学習ベクトルの評価値が負の場合 (斥力学習) . $y - v^{old}$ は学習ベクトルと参照ベクトルとの差 , Δv は参照ベクトルの更新量を表す .

$$\beta(t) = (\beta(0) - \beta(final)) \exp\left(-\frac{t}{\tau_\beta}\right) + \beta(final) \quad (2.10)$$

$$\sigma(t) = (\sigma(0) - \sigma(final)) \exp\left(-\frac{t}{\tau_\sigma}\right) + \sigma(final) \quad (2.11)$$

ここで , τ_α , τ_β および τ_σ は減衰率を表す . 学習における各パラメータを , $\alpha(0) = 0.5$, $\alpha(final) = 0.05$, $\beta(0) = 0.5$, $\beta(final) = 0.005$, $\sigma(0) = 10$, $\sigma(final) = 0.1$, $\tau_\alpha = 100$, $\tau_\beta = 100$, $\tau_\sigma = 100$, $\sigma_r = 0.1$ として学習を行った .

参照ベクトル v_i の初期値は , 一様分布 $U[-0.05 : 0.05]$ から得られる値を用いて決定した . 学習前および学習後の参照ベクトルの分布をそれぞれ図 2.4(c) , (d) に示す . 図中 , * (アスタリスク) は参照ベクトルを表し , 競合層上において隣り合うノードの参照ベクトルは直線で結ばれている . 図 2.4(d) から , 学習後の参照ベクトルは , 評価の良い学習ベクトルが集中して存在する領域を中心に分布していることがわかる .

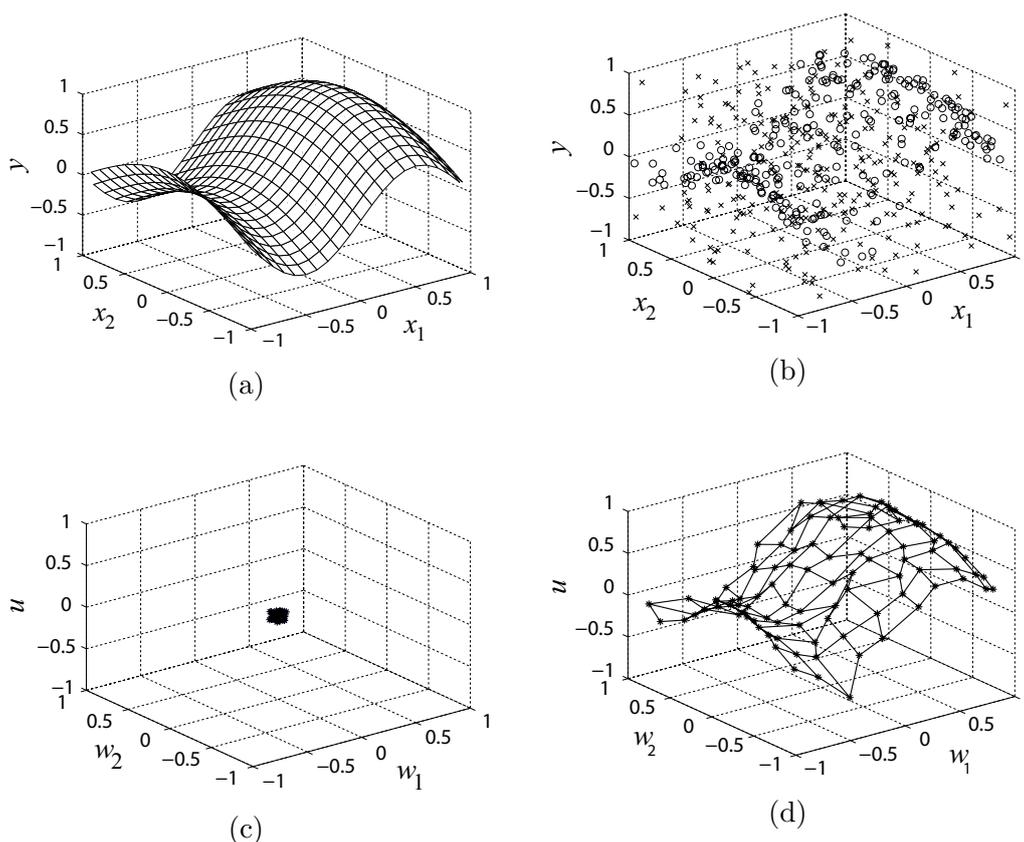


図 2.4: SOR の学習例 . (a) 望ましい入出力間の写像関係 . (b) 学習に用いた入出力ベクトルペア . \times は、それぞれ正および負の評価値をもつ学習ベクトルを表す . (c) 参照ベクトル v_i の初期値 . (d) 学習後の参照ベクトル v_i . * (アスタリスク) は参照ベクトルを表し、競合層上で隣り合うノードの参照ベクトルは直線で結ばれている .

2.2.2 自己組織化関係ネットワークの実行モード

図 2.1(b) に、実行モードの処理の概念図を示す . 学習後の SOR における各ノードは、それぞれが一つの *if - then* ルールを表現している . 実行モードでは、学習モードで獲得した *if - then* ルール集合を用いて、実際の入力に対する望ましい出力を生成する . 実行モードでは、以下の手順に従って、学習後の参照ベクトルから出力ベクトルが生成される . テスト用の入力ベクトル x^* が与えられた際、入力ベクトル空間における参照ベクトル w_i と入力ベクトル x^* との類似度が次式で算出される .

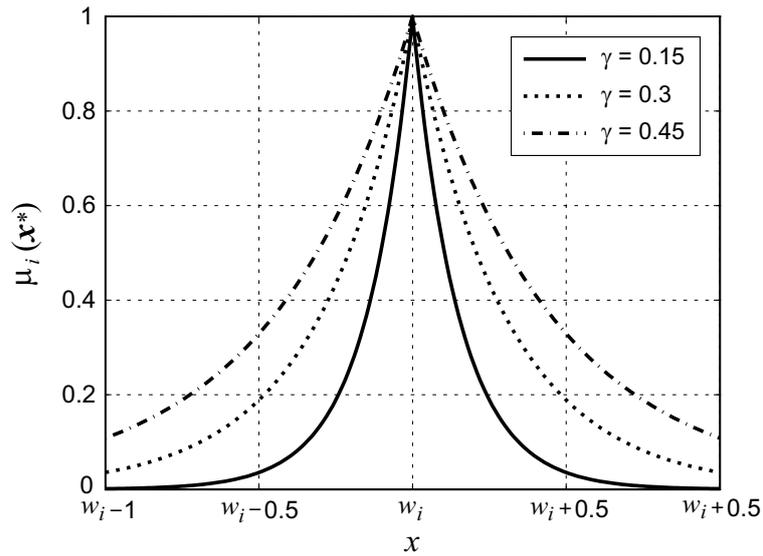


図 2.5: ファジィ *if-then* ルールの前件部メンバーシップ関数 .

$$z_i = \exp\left(-\frac{\|\mathbf{x}^* - \mathbf{w}_i\|}{\gamma}\right) \quad (2.12)$$

ここで, γ は類似度に関するパラメータである . 類似度を表す関数は , 入力空間における参照ベクトルを中心とした放射状関数であれば , 関数の形状は任意である . ここでは , 文献 [16] で使用されている指数関数状の関数を用いて説明を行う .

出力ベクトル \mathbf{y}^* の k 番目の要素は次式により算出される .

$$y_k^* = \frac{\sum_{i=1}^N z_i u_{ik}}{\sum_{i=1}^N z_i} \quad (2.13)$$

算出された出力ベクトル \mathbf{y}^* は , 出力ベクトル空間における参照ベクトル \mathbf{u}_i の , 類似度 z_i に関する重み付き平均となる .

SOR の実行モードの処理は , 以下に示す N 個のファジィ *if-then* ルールを用いた簡略型ファ

ジィ推論 [32; 33] と類似している .

$$\begin{aligned}
 & \text{If } \mathbf{x}^* \text{ is } \tilde{A}_1, \text{ then } \mathbf{y}^* \text{ is } B_1. \\
 & \quad \vdots \\
 & \text{If } \mathbf{x}^* \text{ is } \tilde{A}_i, \text{ then } \mathbf{y}^* \text{ is } B_i. \\
 & \quad \vdots \\
 & \text{If } \mathbf{x}^* \text{ is } \tilde{A}_N, \text{ then } \mathbf{y}^* \text{ is } B_N.
 \end{aligned} \tag{2.14}$$

ここで, \tilde{A}_i, B_i は, 前件部メンバーシップ関数および後件部実数値である . 前件部メンバーシップ関数は, 図 2.5 に示されるように, 中心座標が w_i であり, その広がりは γ によって定義される . ファジィ推論において入力ベクトルの前件部メンバーシップ関数との適合度 $\mu_i(\mathbf{x}^*)$ を算出することは, SOR の実行モードにおいて, 式 (2.12) を用いて類似度 z_i を算出する操作に対応する . 推論結果を算出する非ファジィ化は, 次式で行なわれる .

$$\mathbf{y}^* = \frac{\sum_{i=1}^N \mu_i B_i}{\sum_{i=1}^N \mu_i} \tag{2.15}$$

この操作によって推論値を算出する方法をファジィ推論では重心法と呼ぶ . 非ファジィ化は, 式 (2.13) の重み付き平均を行なう操作に対応する .

実行モードの例として, 前項の学習例で得られた参照ベクトル (図 2.4(d)) を用いた際に, 実行モードが示す入出力間の写像関係を図 2.6(a) に示す . 実行モードにおけるパラメータ γ は 0.15 とした . 図 2.6(b) は, 入力ベクトルに対して SOR の実行モードによって得られた出力と理想的な出力との絶対誤差を示している . SOR で生成した出力ベクトルと理想的な出力ベクトルの平均絶対誤差は 0.059 であった .

次に, 学習ベクトルとして評価値 1.0 をもつ望ましい入出力ベクトルペアだけを用いた場合のシミュレーション結果を示す . 学習条件は, 前項の SOR の学習と同じである . 実行モードにおける

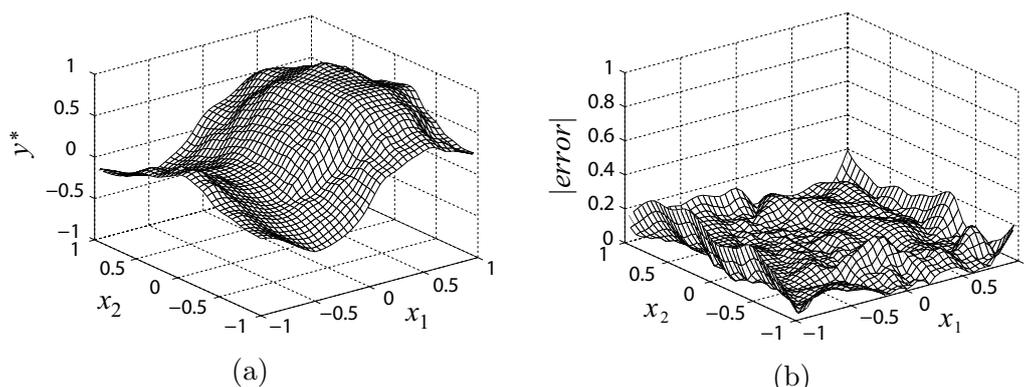


図 2.6: 自己組織化関係ネットワークの実行モードのシミュレーション結果。(a) 自己組織化関係ネットワークの実行モードが示す入出力ベクトル間の写像関係。(b) 自己組織化関係ネットワークで得られた出力と理想的な出力との絶対誤差(平均絶対誤差=0.059)。

パラメータ γ は 0.15 とした。学習に用いた学習ベクトルの分布を図 2.7(a) に示す。学習後の参照ベクトルの分布を図 2.7(b) に示す。自己組織化関係ネットワークの実行モードにおける入出力間の写像関係を図 2.7(c) に示す。図 2.7(d) は、入力ベクトルに対して SOR によって得られた出力と理想的な出力との絶対誤差を示している。SOR で生成した出力ベクトルと理想的な出力ベクトルの平均絶対誤差は 0.04 であった。以上の結果から、SOR は教師信号を用いずに入出力ベクトルペアとその評価値という情報のみを用いて学習しているにもかかわらず、その近似誤差は教師信号を用いた場合とほとんど差がないことがわかる。

試行錯誤から望ましい入出力間の写像関係を抽出する場合、良好な試行結果のみを教師信号として学習を行なうことは可能であるが、このようなデータの獲得には膨大な時間と労力が必要となる。これに対し、SOR は試行結果とその評価値を用いて学習を行なうので、データの獲得に要する時間と労力の削減という観点から、非常に有効である。

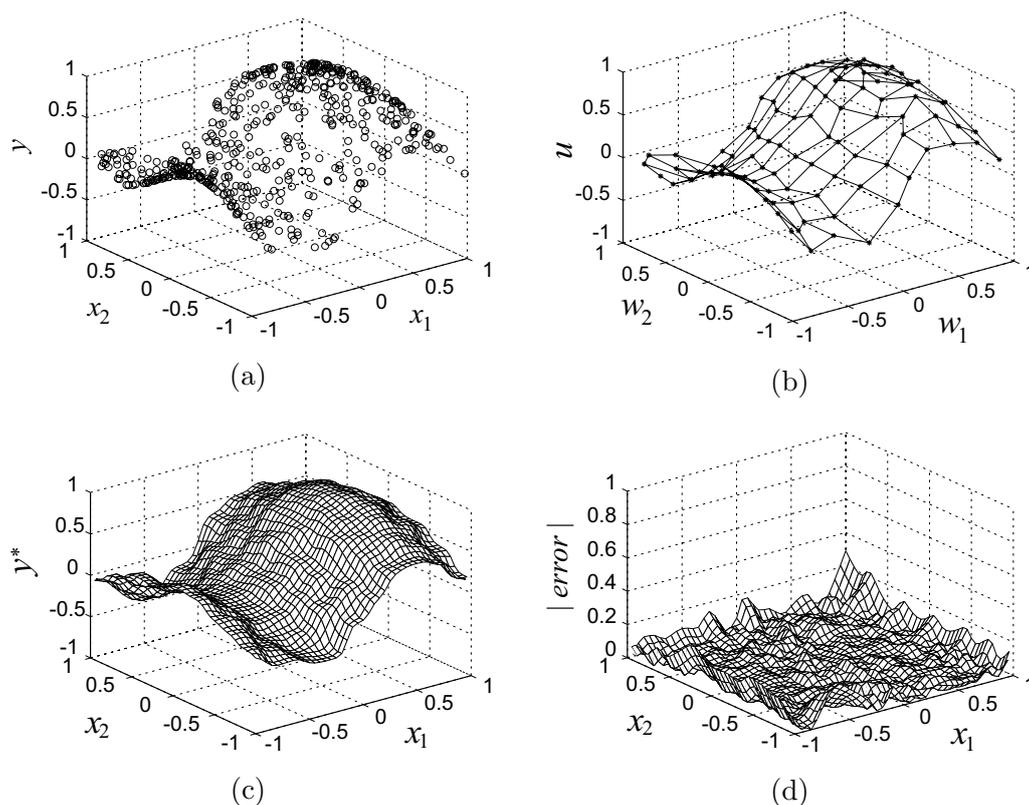


図 2.7: 学習ベクトルとして理想的な入出力ベクトルペアのみを用いた際の、自己組織化関係ネットワークの実行モードのシミュレーション結果。(a) 学習ベクトル I の分布。(b) 学習後の参照ベクトル v_j の分布。(c) 自己組織化関係ネットワークの実行モードが示す入出力ベクトル間の写像関係。(d) 自己組織化関係ネットワークで得られた出力と理想的な出力との絶対誤差 (平均絶対誤差=0.040)。

2.3 二つの適応型自己組織化関係ネットワーク

2.2 で述べたように、SOR は評価値付き学習によってシステムの望ましい入出力関係を抽出し、簡易型ファジィ推論によって出力を生成するネットワークである。しかしながら、SOR の学習では、学習係数および近傍範囲を小さくして学習を収束させる必要性や、学習ベクトルを事前に獲得する必要があり、時間とともに望ましい入出力関係が変化するシステムへの適用は不可能であった。この問題点を解決するために、ESOR[21] および ASOR[22] という二つの適応型 SOR が提案されている。本節では、これら二つの適応型 SOR について、それぞれの特長およびアルゴリズム

について述べる．

2.3.1 進化型自己組織化関係ネットワーク

SORの学習は、自己組織化マップと同様に、競合層の次元と位相構造を予め定め、その構造に基づいて競合学習が行われる．しかし、自律ロボットの行動学習においては、事前に学習データ集合の構造は分からない．競合層の構造が適切に設計されない場合、設計者の意図する学習結果を得ることができない．このような、競合層の構造と学習ベクトル集合の位相構造のミスマッチによって生じる学習に対する悪影響 [34] を抑える手法として、ESOR が提案されている．

位相構造の不一致がもたらす悪影響を回避する自己組織化アルゴリズムとしては主に、学習データ集合の分布近似だけを行うノード数を学習データ集合にあわせて生成的に決定する手法 [35; 36]、学習データ集合の分布近似だけを行う手法 [37]、位相構造を学習前に規定せず、学習の過程で学習データ集合の本質的な構造を抽出する手法 [30; 31; 38] がある．中でも、Evolving SOM [30; 31] の進化的学習アルゴリズムは、構造抽出能力、パラメータスケジューリングの必要性がない、処理速度などの観点から優れており、SORへ導入された．

ESORは、2章で述べたSORと同様に、 N 個の参照ベクトル集合、

$$V = v_1, v_2, \dots, v_N \quad (2.16)$$

によって構成される．ここで、参照ベクトル v_i は、入力参照ベクトル w_i と出力参照ベクトル u_i を連結したベクトルである．各参照ベクトルは、受容野半径 ϕ と活性度 a_i をもつ．ESORは、競合層を持たないので、学習以前に参照ベクトルの個数や位相構造を定める必要はない．さらに、学習係数や近傍係数などを、学習の経過に合わせてスケジューリングする必要もない．その代わりに、設計者は、受容野半径 ϕ というパラメータを学習前に定数として決定する．学習ベクトルも従来手法と同様に、入力ベクトルと出力ベクトルを連結したベクトル $I_l = (x_l, y_l)$ とする（以降、特に必要がない場合を除いて、添字を省略する）．ESORの学習は次の手順に従って進められる．

手順 0 参照ベクトルが存在しない状態から学習を始める．

手順 1 学習ベクトル I をネットワークに提示する．もし、参照ベクトルが一つも存在せず、現在の学習ベクトルの評価値 E が事前に定めた閾値 E_{thr} を上回る場合には、学習ベクトル I の位置に最初の参照ベクトルを生成する．

手順 2 学習ベクトル I と既存の参照ベクトル間の類似度を計算する．通常はユークリッド距離を用いる．学習ベクトル I から最近傍の参照ベクトルまでの距離が受容野半径 ϕ よりも大きい場合には、現在の学習ベクトル I の位置に新しい参照ベクトルを生成するために、手順 3 に進む．

$$\|v_i - I\| \geq \phi, \forall v_i \in V \quad (2.17)$$

学習ベクトルから最近傍の参照ベクトルまでの距離が受容野半径 ϕ よりも小さい場合には、既存の参照ベクトル集合を更新するために、手順 4 に進む．

$$\|v_i - I\| < \phi, \forall v_i \in V \quad (2.18)$$

手順 3 (生成) 新しい参照ベクトルを現在の入力ベクトルの位置に配置する．

$$v_{N+1} = I \quad (2.19)$$

新しい参照ベクトルから、最近傍となる二つの参照ベクトルに対して、リンクを生成する．手順 5 に進む．

手順 4 (更新) 学習ベクトル I に対する勝者ノード c とその近傍ノードの更新を行う．勝者ノード c とその最近傍ノードの間にリンク結合が存在しない場合は、リンク結合を生成する．ここでは、勝者ノード c との間にリンク結合をもつ全てのノードの集合を近傍ノード集合 $\Omega(c)$ と定義する．

$$\Omega(c) = \{i | s_{c,i}(t) > 0\}, \forall i \in [1, N] \quad (2.20)$$

ここで、 $s_{c,i}(t)$ は、更新前の状態において、勝者ノード c と i 番目のノードとの間のリンク結合強度を表す． i 番目のノードの近傍係数は、以下の式によって定義される．

$$b_i = \exp\left(-\frac{\|\mathbf{I} - \mathbf{v}_i\|^2}{2\phi^2}\right) \quad (2.21)$$

次式によって、現在のデータに対する参照ベクトルの更新量を算出する。

$$\Delta \mathbf{v}_i = \begin{cases} \alpha \frac{b_i E}{\sum_{i'} b_{i'} |E|} (\mathbf{I} - \mathbf{v}_i) & \text{if } i \in \Omega(c), E > 0 \\ \beta \frac{b_i E}{\sum_{i'} b_{i'} |E|} \exp\left(-\frac{\|\mathbf{I} - \mathbf{v}_i\|}{\sigma_r}\right) \frac{\mathbf{I} - \mathbf{v}_i}{\|\mathbf{I} - \mathbf{v}_i\|} & \text{if } i \in \Omega(c), E < 0 \\ 0 & \text{otherwise} \end{cases} \quad (2.22)$$

ここで、 α 、 β はそれぞれ、引力学習および斥力学習の学習係数である。参照ベクトルの更新後、勝者ノード c に接続しているリンクの結合強度を次式によって更新する。

$$s_{c,i}(t+1) = \omega s_{c,i}(t) + (1 - \omega) b_c b_i E \quad (2.23)$$

ここで、 ω は、リンクの結合強度に関する忘却係数である。結合強度が0以下の場合、リンクを削除する。さらに、勝者ノードと近傍ノードの活性度を次式によって更新する。

$$a_i(t+1) = \delta a_i(t) + (1 - \delta) a_i(t) E \quad (2.24)$$

ここで、 δ は、ノードの活性度に関する忘却係数である。活性度が0以下の場合、ノードおよび接続しているリンクを削除する。

手順5 結合強度が最も弱いリンクを T_p ステップごとに削除する。以後、学習ベクトルが提示される間、上記の手順を繰り返す。

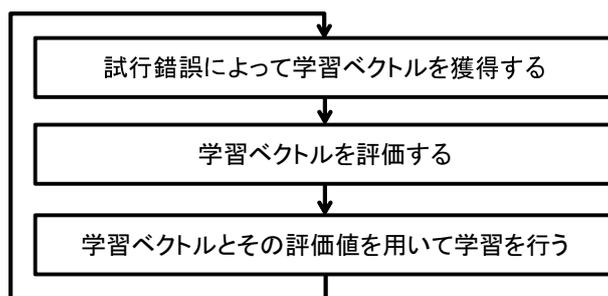


図 2.8: 適応自己組織化関係ネットワークの処理手順 .

2.3.2 適応自己組織化関係ネットワーク

SOR は、対象の望ましい入出力関係の変化に対する適応性を持たない。これは、SOR における以下の二つの点に起因する。

- (1) 学習ベクトルおよびその評価値を事前に獲得する必要がある。
- (2) 学習係数と近傍範囲を時間と共に減少させて学習を収束させる必要がある。

これらの問題点を解決する手法として ASOR が提案された [22]。

ASOR では、(1) の問題点を解決するために、図 2.8 に示すように、実行モードで入出力ベクトルペアを生成し、それに対する評価を用いて学習を行う。これによって、事前に学習ベクトルを用意する必要がなくなっただけでなく、自律学習も可能となった。(2) の問題点については、SOR の現在の状態の良否を判断するために、学習必要性尺度が導入された。学習必要性尺度は、現在の SOR の実行モードで生成された入出力ベクトルペアが、評価規範を満たしているかいないかという情報に基づいて次式によって算出される。

$$nec(t_l) = nec(t_l - 1) + \eta(S^{th} - S(t_l - 1)), \quad 0 \leq nec(t_l) \leq nec_{max} \quad (2.25)$$

η はパラメータであり、次式によって表される。

$$\eta = \begin{cases} \eta_- & \text{for } S^{th} \leq S(t_l - 1) \\ \eta_+ & \text{for } S(t_l - 1) < S^{th} \end{cases} \quad (2.26)$$

ここで、 η_- と η_+ は学習必要性の減少および増加に関するパラメータで正の定数である。 t_l は評価関数に対する満足率 S を算出するステップである。 $S(t_l - 1)$ と S^{th} は、それぞれステップ $t_l - 1$ における満足率と閾値である。 nec_{max} は学習必要性尺度が定義される最大値である。満足率 S とは、SOR の出力が正の評価値を得る度合いを表し、次式を用いて算出される。

$$S = \frac{\int E(t) \|_{0 \leq E(t)} dt}{\int \|E(t)\| dt} \quad (2.27)$$

満足率は、全ての出力に対して正の評価を得ることができれば1になる。

式(2.25)は、ステップ $t_l - 1$ における満足率 $S(t_l - 1)$ が閾値 S^{th} より大きい場合、SOR の状態は良好であると判断し学習必要性尺度を小さくし、満足率 $S(t_l - 1)$ が閾値 S^{th} より小さい場合、SOR の状態は良好でないと判断し学習必要性尺度を大きくすることを意味する。ステップ t_l における引力の学習係数 $\alpha_{SOR}(t_l)$ は、学習必要性尺度に基づいて次式で決定される。

$$\alpha_{SOR}(t_l) = \alpha_0 nec(t_l), \quad 0 \leq \alpha_{SOR}(t_l) \leq \alpha_{max} \quad (2.28)$$

ここで、 α_0 はパラメータで正の定数である。 α_{max} は学習係数が定義される最大値である。ステップ t_l における近傍範囲 $N_{c,SOR}(t_l)$ は、学習必要性尺度に基づいて次式によって決定される。

$$N_{c,SOR}(t_l) = N_{c,0} nec(t_l), \quad 0 \leq N_{c,SOR}(t_l) \leq N_{c,max} \quad (2.29)$$

ここで、 $N_{c,0}$ はパラメータで正の定数である。 $N_{c,max}$ は近傍範囲が定義される最大値である。学習必要性尺度が大きい場合、引力学習の学習係数および近傍範囲も大きくなり、SOR の学習が進行する。一方、学習必要性尺度が小さい場合、引力の学習係数および近傍範囲も小さくなり、SOR の学習は収束する。ステップ t_l における斥力の学習係数 $\beta_{SOR}(t_l)$ は次式で決定される。

$$\beta_{SOR}(t_l) = \beta_0 (nec(t_l - 1) - nec(t_l)) \quad 0 \leq \beta_{SOR}(t_l) \leq \beta_{max} \quad (2.30)$$

ここで、 β_0 はパラメータで正の定数である。 β_{max} は学習係数が定義される最大値である。学習必要性尺度が小さくなる際に、斥力の学習係数は大きくなる。このような係数の調節は、学習必要性尺度が大きくなる際は、学習の安定のために引力学習のみを行ない、学習必要性が小さくなる際には、学習の調整のために引力学習および斥力学習を行なうことを目的としている。従来のSORにおいて、学習ベクトルを獲得する際には、出力にノイズを加えて種々の入出力ベクトルペアを獲得していた。ASORでもこの考え方をを用いる。学習必要性尺度が大きいとき、言い換えるとSORの実行モードで生成された出力ベクトルが望ましくないときは、ネットワークの出力にノイズを加えることによって、望ましい出力ベクトルを探し出す機能を加える。ノイズは一様乱数から生成されるとし、その範囲は $[-N_{noise}(t_l) : N_{noise}(t_l)]$ とする。 $N_{noise}(t_l)$ は次式によって決定される。

$$N_{noise}(t_l) = N_{noise,0nec}(t_l), \quad 0 \leq N_{noise}(t_l) \leq N_{noise,max} \quad (2.31)$$

ここで、 $N_{noise,0}$ はパラメータで正の定数である。 $N_{noise,max}$ はノイズの範囲が定義される最大値である。引力の学習係数や近傍範囲と同様に、ノイズの範囲も学習必要性尺度に比例する。

学習に用いるデータとして、 L 個の入力ベクトル

$$\{x_l \mid l = 1, \dots, L\} \quad (2.32)$$

を考える。これらの入力ベクトルから以下のアルゴリズムに従い、時間とともに変化する望ましい入出力ベクトル間の写像関係を抽出する。

手順 0. 競合層ノードの全ての結合重みベクトルの初期化を行なう。その初期化は乱数を用いて決定してもよいし、ユーザが決定してもよい。

手順 1. 実行モードと学習モードを繰り返す。

手順 1.1. (実行モード) 入力ベクトル集合の中から一つのベクトル x_l を選択し、入力層に提示する。自己組織化関係ネットワークの実行モードによって出力ベクトル y_l を生成する。生成した入出力ベクトルペアの評価値 E_l を算出する。

手順 1.2. (学習モード) 手順 1.1 で生成した入出力ベクトルペアを自己組織化関係ネットワークに提示して、その評価に基づき結合重みベクトルを更新する。

手順 1.3. 入力ベクトル集合の中に一度も選択していないベクトルがある場合は、手順 1.1 に戻り、これまで選択していない入力ベクトルを新たな入力ベクトルとする。入力ベクトル集合内の全ての入力ベクトルを選択した場合は手順 2 に進む。

手順 2. 手順 1 で生成した入出力ベクトルペアの評価関数に対する満足度を算出する。

手順 3. 手順 2 で算出した満足度から式 (2.25) に従って学習必要性尺度を算出する。学習必要性尺度に基づき、引力および斥力の学習係数、近傍範囲、出力に加えるノイズの範囲を決定する。

手順 4. 手順 1 ~ 手順 3 を繰り返す。

以上のように、引力学習および斥力学習の学習係数、近傍範囲、出力に加えるノイズの範囲を学習必要性尺度に基づいて決定することで、学習の収束および再学習を制御することができ、対象システムの変化に適応することが可能になる。

2.4 おわりに

本章では、従来の SOR および二つの適応型 SOR について、それらの構成、学習方法、および特長についてまとめた。SOR は、SOM のベクトル量子化能力および位相保持能力といった特長を継承し、入出力関係を表現できるように改良されたアルゴリズムである。SOR は学習モードにおいて、学習ベクトルの評価値に応じた更新を行うことによって、望ましい入出力関係を記述する *if-then* ルールを獲得する。そして実行モードにおいて、簡略型ファジィ推論エンジンとして動作し、実際の入力に対する望ましい出力を生成する。SOR は学習モードにおいて、引力学習と斥力学習を評価値に応じて使い分けることによって、教師あり学習では利用できなかった望ましくない入出力関係を表す学習ベクトルを利用できるようになった。斥力学習を行うことによって、学習の効率化だけでなく、危険領域の回避も可能になる。したがって、未知の環境で動作する自律ロボットへの応用については、教師あり学習手法よりも効果的な学習が可能であると考えられる。

SORの学習は、競合層の次元と位相構造を予め定める必要があった。しかし、自律ロボットの自律学習において、予め適切な競合層の構造を予測することは難しい。このような問題を解決する手法として、ESORが提案されている。ESORは、Evolving SOMの進化的学習アルゴリズムを導入することによって、構造抽出能力、パラメータスケジューリングの必要性がない、計算コストが低いなどの特長を持つ。

SORは、対象の望ましい入出力関係の変化に対する適応性を持たない。これは、SORにおける以下の二つの点に起因する。

- 学習ベクトルおよびその評価値を事前に獲得する必要がある。
- 学習係数と近傍範囲を時間と共に減少させて学習を収束させる必要がある。

これらの問題点を解決する手法としてASORが提案された [22]。ASORでは、実行モードと学習モードを交互に繰り返し行う機能と、実行モードで生成した出力に対して試行信号を付加する機能を導入した。これによって、予め学習ベクトルを獲得しておくことなく、自律的な試行錯誤によって学習することが可能になった。

第3章以降では、本章でまとめたSORについて、自律ロボットの行動学習へ適用する際の問題点を明らかにした上で、その問題点を解決するための改良を加えた新規の学習手法を提案する。

第3章 ノード評価値をもつ自己組織化関係ネットワーク

3.1 はじめに

1章で述べたように、自律ロボットの行動学習には、自律性と適応性が必要とされる。これに対して、2章で示したように、SOR を拡張した二つの適応型 SOR は、それぞれ優れた能力を有する。具体的には、ESOR は、未知環境から得られる入力データ分布の時間変化に対しても対応できる高い適応能力を有する。また、ASOR は、学習用データを試行錯誤的に自己生成することが可能であり、自律学習を可能にする。しかし、実機への適用を考える場合、これらのいずれか一つでは不十分であり、これら二つの機能は兼備されている必要がある。したがって、二つの機能を兼備した手法を考えなければならない。しかし、二つの機能を統合する際、従来の SOR における斥力学習の更新方法に関する問題が生じる。本章ではまず、3.2 において、二つの機能を統合する際の斥力学習の問題点とその解決策について述べる。そして、3.3 において、これら二つの機能を兼備した手法を提案する。

3.2 従来の斥力学習の問題点とその解決策

ここでは、従来の SOR の斥力学習における問題点とその解決策について述べる。図 3.1 に問題点の一例を示す。図中の \mathbf{o} および \mathbf{b} はそれぞれ、入出力空間におけるノードの参照ベクトルおよび学習ベクトルを表す。図では、説明のために学習ベクトル I の評価の分布を示しているが、実際には未知である。図 3.1(a) に示すように、SOR および二つの適応型 SOR における斥力学習で

は、次式にしたがって参照ベクトル v_i は学習ベクトル I から単純に離れるように更新される。

$$\begin{aligned}\Delta v_i &= v_i^{new} - v_i^{old} \\ &= \beta E(I - v_i^{old}),\end{aligned}\tag{3.1}$$

$$I = (x, y),$$

$$v_i = (w_i, u_i).$$

ここで、 β は学習係数を表す。このような更新方法では、図 3.1(a) に示すように、参照ベクトル v_i が高い評価を得られる領域から遠ざかるように更新される場合がある。これは、学習ベクトルの評価分布の勾配が未知であることに起因する。参照ベクトル v_i の正しい更新方向は、参照ベクトル v_i と学習ベクトル I の比較によってのみ得られる。したがって、各ノードの位置における参照ベクトルの評価値を表す新たなパラメータを導入する必要がある。そこで本研究では、新たなパラメータとして「ノード評価値」を定義し、学習ベクトルの評価値とは独立に扱う。二つの評価値を以下のように定義する。

E : 入出力評価値,

\mathcal{E} : ノード評価値.

E および \mathcal{E} は -1 から 1 までの範囲で表され、お互いに比較することが可能である。また、参照ベクトル v_i の位置は学習によって更新されるので、同時に \mathcal{E}_i の値も更新する必要がある。ここで、 i はノードの番号を表し、 \mathcal{E}_i は i 番目のノードの評価値を表す。 \mathcal{E}_i は、 E と \mathcal{E}_i との差が最少になるように更新される。ノード評価値 \mathcal{E} の導入によって、参照ベクトルは適切な方向へ更新される。これによって、斥力学習による学習効率が飛躍的に向上する。この学習効率の向上は、自律ロボットへの適用の際、ロボットの物理的な消耗を抑制するために大きな効果を生むと考えられる。

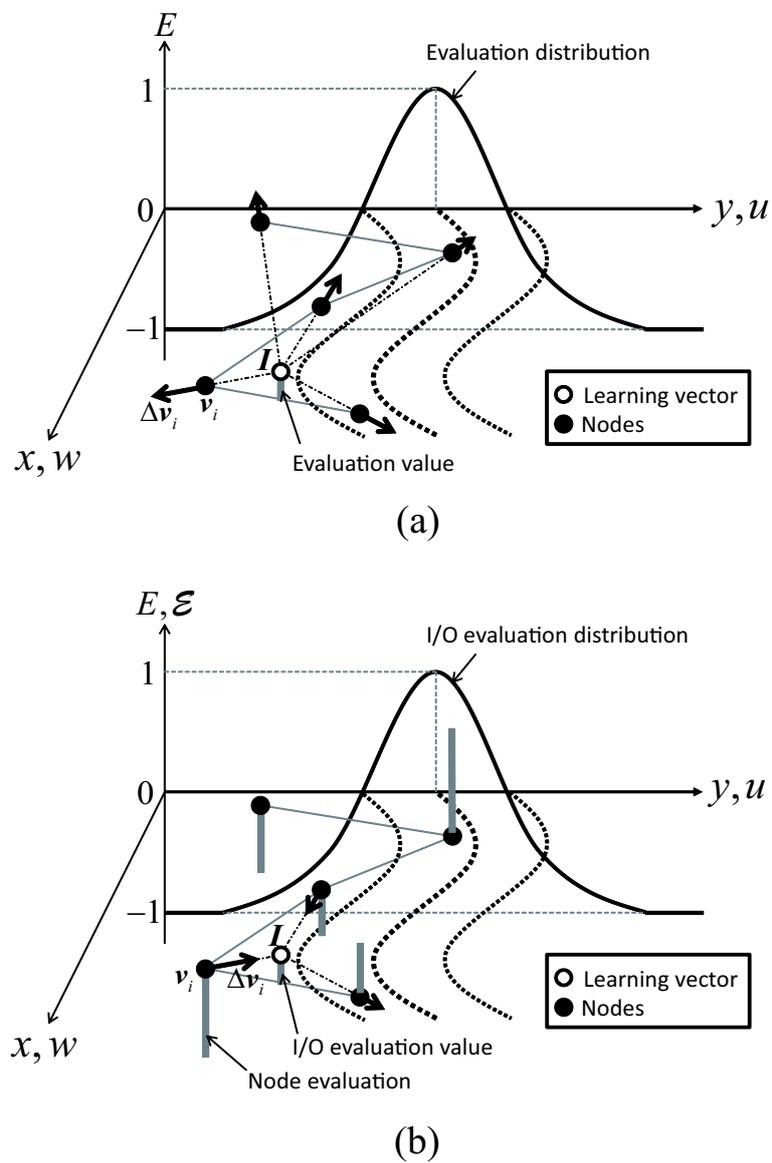


図 3.1: 参照ベクトルの更新時における問題点と解決策 . (a) 従来の自己組織化関係ネットワークにおける斥力学習 . (b) 提案手法 (動的自己組織化関係ネットワーク) における学習 .

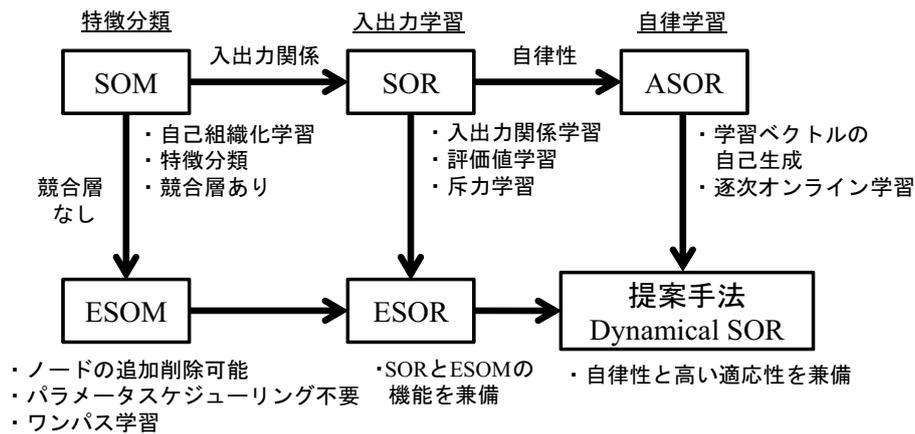


図 3.2: 既存手法と提案手法の関係 . SOM: Self-Organizing Map, ESOM: Evolving SOM, SOR: Self-Organizing Relationship Network, ESOR: Evolving SOR, ASOR: Adaptive SOR.

3.3 動的自己組織化関係ネットワーク

3.2 で述べたように、ノード評価値 ε を導入することによって、参照ベクトルを常に望ましい方向へ更新できるようになった。この改良によって、二つの適応型 SOR のもつ優れた機能を統合することが可能になった。本節では、二つの適応型 SOR の機能を兼備した手法として、動的自己組織化関係ネットワーク (DSOR: Dynamical Self-Organizing Relationship Network) を提案する。図 3.2 に、2 章で述べた従来の SOR と DSOR との関係性を示す。

DSOR は図 3.2 に示すように、ESOR および ASOR の機能を組みあわせることによって実現される。DSOR の概念図を図 3.3 に示す。図 3.3 中の、知識獲得ブロック (Knowledge Acquisition) および推論ブロック (Inference) はそれぞれ、SOR における学習モードおよび実行モードに相当する。ここではまず、本節において DSOR の学習手順について述べたあと、3.3.1 および 3.3.2 において知識獲得ブロックおよび推論ブロックの処理手順について説明する。

DSOR の学習手順を以下に示す。

手順 1 センサ信号を用いて入力ベクトル $\mathbf{x} = [x_1, \dots, x_m, \dots, x_{N_x}]^T$ を生成し、DSOR へ入力する。ここで、 m と N_x はそれぞれ、ベクトルの要素番号およびセンサ信号数 (= 入力参照ベ

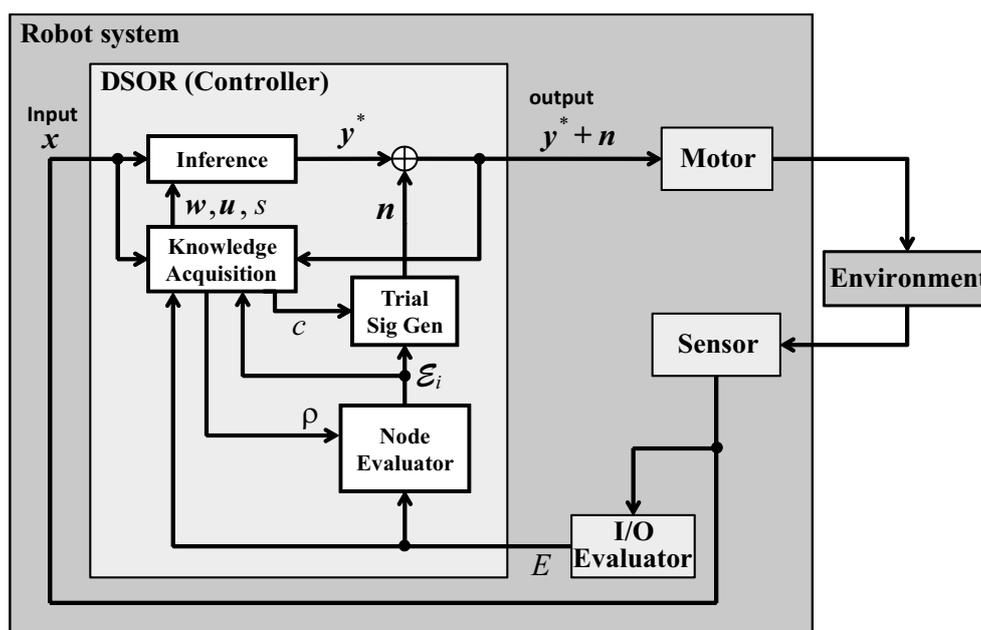


図 3.3: 動的自己組織化関係ネットワークの概念図 .

クトルの要素数) を表す .

手順 2 推論ブロックによって出力ベクトル $y = [y_1, \dots, y_k, \dots, y_{N_y}]^T$ を生成する . ここで , k と N_y はそれぞれ , ベクトルの要素番号およびコントロールする装置の数 (= 出力参照ベクトルの要素数) を表す . もしノードの数がゼロなら , $y^* = 0$ とする .

手順 3 推論ブロックにおける勝者ノード c のノード評価値 ε_c に応じた大きさの試行信号 n を y^* へ加算する . ここで , 試行信号 n はガウスノイズ $N(0, \sigma_{noise}^2)$ によって決定される . σ_{noise} は勝者ノード c のノード評価値 ε_c に応じて決まる . もし , ノード数がゼロなら , n はゼロとする .

手順 4 $y^* + n$ をロボットシステムへ入力し , これによってロボットが試行動作する .

手順 5 ロボットの試行動作後 , センサ信号を取得する .

手順 6 入出力評価器において , 手順 5 で得られたセンサ信号を用いて E を生成する . E は , 知識獲得ブロックおよびノード評価器へ与えられる .

手順 7 $x, y^* + n, E, \varepsilon_i$ によって , ノード集合の更新を行う .

手順 8 次式によって, ノード評価値 \mathcal{E}_i を更新する

$$\mathcal{E}_i(t+1) = \mathcal{E}_i(t) + \rho_i(t)(E(t) - \mathcal{E}_i(t)). \quad (3.2)$$

ここで, ρ_i は, 正規化された活性度を表す (これらのパラメータについては, 次節において詳しく説明する.)

手順 9 手順 1 へ.

3.3.1 知識獲得ブロックの処理手順

手順 0 ノードが存在しない状態から学習を開始する.

手順 1 学習ベクトル \mathbf{x} , $\mathbf{y}^* + \mathbf{n}$ および E をネットワークへ提示する. もし, ノード数がゼロなら, 新しいノードを生成し, 入力参照ベクトル $\mathbf{w}_i = [w_{i1}, \dots, w_{im}, \dots, w_{iN_x}]^T$ を \mathbf{x} の位置に, 出力参照ベクトル $\mathbf{u}_i = [u_{i1}, \dots, u_{ik}, \dots, u_{iN_x}]^T$ を $\mathbf{y}^* + \mathbf{n}$ の位置に配置する.

手順 2 時刻 t におけるノードの活性度を次式によって導出する.

$$z_i = \exp \left\{ - \frac{\| \mathbf{x}(t) - \mathbf{w}_i(t) \|^2}{2\phi_i^2} \right\}, \quad (3.3)$$

ここで, ϕ_i は i 番目のノードの受容野半径を表す. ノードの挿入は ϵ -completeness[39] に基づいて決定する. もし存在するノードの活性度の最大値が 0.1354 以下であれば, 入力ベクトル \mathbf{x} の位置に新しいノードを挿入する. 0.1354 という値は, $[\mathbf{w}_{im} - 2\phi_i, \mathbf{w}_{im} + 2\phi_i]$ における活性度を表す. これは, 入力データの分布を正規分布と仮定したとき, 95% の入力データがこのメンバシップ関数の $[\mathbf{w}_{im} - 2\phi_i, \mathbf{w}_{im} + 2\phi_i]$ の範囲内に含まれることを意味する [40; 41]. これによって, 全ての入力データは存在するノードのうちのいずれかのメンバシップ関数に含まれる. もし, 活性度が 0.1354 よりも小さければ手順 3 へ進む. もし, 活性度が 0.1354 よりも大きければ手順 4 へ進む.

手順 3 (生成) 新たなノードを生成し, その参照ベクトル $\mathbf{v}_i = (\mathbf{w}_i, \mathbf{u}_i)$ を $(\mathbf{x}, \mathbf{y}^* + \mathbf{n})$ の位置に配置する. ノードの生成後, 新たに生成されたノードとその最近傍ノードとの間にリンクを生成し手順 5 へ進む.

手順 4 (更新) 次式によって勝者ノード c を決定する.

$$c(t) = \arg \max_i \{z_i(t)\}, \quad (3.4)$$

勝者ノード c および勝者ノード c との間にリンク結合をもつノードのみを更新する. もし勝者ノード c およびその最近傍ノード i との間にリンク結合がない場合, リンク結合を生成する. リンク結合強度は次式で計算される.

$$s_{c,i}(t) = (1 - \omega)z_c(t)z_i(t), \quad (3.5)$$

ここで, ω は, リンク結合強度の忘却係数を表す. また, 勝者ノード c の近傍ノードの集合は以下のように定義される.

$$\Omega(c(t)) = \{i | s_{c,i}(t) > 0, \forall i \in [1, N_{node}]\}, \quad (3.6)$$

ここで, N_{node} はノード数を表す. 入力参照ベクトルは次式によって更新される.

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \alpha z_i(t) \{\mathbf{x}(t) - \mathbf{w}_i(t)\}, \quad (3.7)$$

ここで α は学習係数を表す. 出力参照ベクトルは, 次式を用いて更新される.

$$\mathbf{u}_i(t+1) = \mathbf{u}_i(t) + \alpha_u \rho_i(t) \{E(t) - \mathcal{E}_i\} [\{\mathbf{y}^*(t) + \mathbf{n}\} - \mathbf{u}_i(t)], \quad (3.8)$$

ここで, α_u および ρ_i はそれぞれ, 学習係数と正規化活性度を表す. 正規化活性度は, 次式によって求められる.

$$\rho_i(t) = \frac{z_i(t)}{\sum_i z_i(t)}. \quad (3.9)$$

参照ベクトルの更新後，結合強度の更新を次式によって行う．

$$s_{c,i}(t+1) = \omega s_{c,i}(t) + (1+\omega)z_c(t)z_i(t), \quad (3.10)$$

$$s(t+1) = \omega s_{c,i}(t). \quad (3.11)$$

更新後，手順 5 へ進む．

手順 5 T_p ステップ毎に，最弱のリンク結合および結合を持たないノードの削除を行う．また，隣り合うノード i と j との距離が $2\phi_i$ よりも小さい場合，これらのノードを次式によって結合する．

$$\mathbf{w}_i(t) = \frac{1}{2} \{\mathbf{w}_i(t) + \mathbf{w}_j(t)\}, \quad (3.12)$$

$$\mathbf{u}_i(t) = \frac{1}{2} \{\mathbf{u}_i(t) + \mathbf{u}_j(t)\}, \quad (3.13)$$

$$\mathcal{E}_i(t) = \frac{1}{2} \{\mathcal{E}_i(t) + \mathcal{E}_j(t)\}. \quad (3.14)$$

3.3.2 推論ブロックの処理手順

DSOR の推論ブロックの処理は，SOR と同様に簡略化ファジィ推論 [32] によって行われる．出力 \mathbf{y}^* は以下の手順で計算される．

手順 1 入力ベクトル \mathbf{x} をネットワークへ提示し，活性度 z_i を式 (3.3) によって導出する．勝者ノード c を式 (3.4) によって計算する．もしノード数がゼロなら，ネットワークの出力は $\mathbf{y}^* = \mathbf{0}$ とする．

手順 2 もし勝者ノード c がリンク結合を持たない場合，2 番目に近いノードとの間に一時的なリンク結合を生成する．

手順 3 出力 \mathbf{y}^* は次式によって計算される．

$$y_k^* = \sum_{i \in \Omega(c)} \rho_i u_{ik}. \quad (3.15)$$

3.4 おわりに

本章では、DSOR を提案し、基本動作と特長についてまとめた。従来の SOR およびその適応学習手法の斥力学習においては、参照ベクトルが望ましくない方向へ更新されてしまうという問題点があった。これは、入出力空間における評価値の分布が未知であることと、学習ベクトルの評価値のみを用いて更新の向きを決定していたことが原因であった。そこで DSOR ではノード評価値を導入し、更新の際に、学習ベクトルの評価値と各ノードの評価値を比較することによって、評価値の勾配を用いた更新が行えるようにした。ノード評価値の導入によって、参照ベクトルは、学習ベクトルの評価値の正負によらず、より望ましい向きに更新されるようになった。

さらに DSOR は、二つの適応型 SOR の特長を導入することによって、自律的かつ適応的な学習が可能になった。従来の SOR では、SOM と同様に競合層上における競合学習を行うので、ノード数および競合層の次元数を予め規定する必要がある。SOR の各ノードは一つの *if - then* ルールに相当するので、ノード数を決定するためには、入出力関係を表現するために適切なルール数を予め予測することが可能である必要がある。しかし、未知環境において学習を行う際に、予め適切なルール数を予測することは難しい。ESOR は、ESOM の進化的な学習則を導入することによって、環境に応じてノードの追加および削除を行うことができるという特長がある。さらに ESOR の、パラメータのスケジューリングが不要であるという特長を導入することによって、学習の必要性に応じて適応的に学習することが可能になった。また、パラメータスケジューリング不要の特長は、入力空間におけるワンパス学習を可能にした。

従来の SOR では、事前に学習ベクトルを獲得し、それらについて評価を行っておく必要があった。しかし、自律ロボットの行動学習において、ロボットは自律的かつオンラインで学習を行うので、自律的な学習ベクトルの取得方法が必要となる。ASOR は、実行モードと学習モードを交互に繰り返す事によって、逐次オンラインの学習を可能にした。また、試行錯誤によって望ましい入出力関係を獲得するために、実行モードの出力に試行信号を付加することによって、学習ベクトルを自己生成することが可能である。DSOR では、ASOR の持つこれらの特長とノード評価値を組みあわせることによって、より効果的な学習ベクトルの取得を実現した。

DSOR の類似手法としては、Fuzzy Actor-Critic Reinforcement Learning Network (FACRLN)[40] や Actor-Critic Reinforcement Learning with NGnet[42] などが挙げられる。これらの手法は、

DSOR と同様に，明示的な教師信号なしに入出力関係を学習することができる．また，類似のルールの結合や基底関数の幅の調節も可能である．しかしこれらの手法では，入力データ分布の時間変化に伴って不要になったルールを削除する機能を持たない．したがって，自律ロボットの行動学習のように，望ましい入出力関係が時変である場合への適応性は不十分であると考えられる．一方，DSOR は，以下のような特長を持つ．

1. 新しい環境において不要になったノードを削除することが可能であり，最小限のノード数によって十分な推論精度を実現できる．
2. 更新プロセスにおいて，勝者ノードとのリンク結合を持つノードのみを更新する．

これらの特長は，計算コストの削減を可能とし，自律ロボットへの適用に対する優位性を高める．

第4章では，本章で提案した DSOR を，音源定位能の自律獲得問題へ応用することによって，その学習能力の実証を行う．

第4章 動的自己組織化関係ネットワークの音源定位学習への応用

4.1 はじめに

人間は、学習によって様々な動きを行うことができるようになる。中でも、注目する物体を見るという行動は、感覚運動協調による最もシンプルかつ本質的な行動である [43]。ヒトの眼球の網膜には、中心窩と呼ばれる部分があり、網膜上の他の部位に比べて空間分解能が高い [34]。中心窩で対象物をとらえることによって、物体をより正確に認識することが可能になる。中心窩に対象物体が投射されるように頭と眼球を動かす定位行動は、視野外の対象物体であっても聴覚を利用することによって高い精度で行うことができる [44]。本論文では、音を出す対象物体に対する定位について考える。視覚と聴覚を用いて対象音源を視野の中心でとらえる行動を音源定位と呼ぶ。ヒトの音源定位能力は学習によって獲得され、成人であっても、視覚刺激の変化に対して数日で適応できる事が確認されている [45]。

このような、ヒトの優れた音源定位機能をロボットへ実装する研究は、ヒトとロボットのより自然なインタラクションや視覚情報を用いた対象物体の認識の向上などを目的として、盛んに行われている。特に今後、生活環境へロボットが多用されるようになると考えられるので、より実用的な音源定位能力の実現が不可欠であると考えられる。これら、ロボットのための音源定位機能の実現方法には、大きく分けて二つのアプローチがある。一つ目は、マイクロフォンなどのセンサから得られた信号を解析的に処理する方法。二つ目は、生物の神経回路モデルなどを用いて、学習によって音源定位を実現する方法である。解析的な手法を用いる場合、マイクロフォンアレイなど、多数のセンサによって得られた信号を用いて処理が行われることが多い。多数のセンサ信号を用いる利点としては、空間分解能が高くなることや複数の音源信号を定位することが可能になることなどが挙げられる。しかし、マイクロフォンアレイを用いたロボットシステムは、センサ数が多くなる、センサ形状が固定される、製作コストがかかる、センサ数に比例して計算コ

ストが増加し専用ハードウェアが必要になるなど、ハードウェア面の制約が大きいだけでなく、ロボットの形状やセンサの特性に合わせた数理モデルを考える必要があり、実用化に対する問題が多い。一方、学習によって音源定位能力を獲得する手法は、一般的に二つのマイクロフォンを用いて実現可能、ロボットの形状やセンサの特性に左右されないなどの特長をもつ。学習による音源定位手法はこのような特長によって、ハードウェアコストがかからず、ロボット形状の設計に対する制約も少ない。したがって本研究では、学習によって音源定位能力を獲得する手法を採用する。自律ロボットの音源定位学習では、主に以下の三つの点について考慮する必要がある。

- 答えを教えてもらえない
- 単一の感覚のみでは全領域にわたる定位は難しい
- 環境の変化（残響時間，温度，障害物など）やロボット自体の特性の変化（物理的構造，センサや効果器など）に対する適応性が必要

これらの点を考慮した音源定位学習手法は、大きく分けて二つのモデルがある。一つは生理学的に得られた知見を実証するための忠実な神経回路網モデル [46; 23] であり、もう一つは、脳機能を簡略化して作られたニューラルネットワークなどを用いたモデル [24; 25; 27] である。自律学習によって音源定位能力を獲得するロボットに関する研究は、実際の神経回路を忠実にモデル化したものと、脳機能を簡略化したニューラルネットワークモデルなどで行うものに分けられる。忠実な神経回路モデルとしては Jeffress モデル [46] が広く用いられている。また、フクロウの定位行動に關係する神経回路のモデルをロボットに実装した研究 [23] では、視覚刺激と聴覚刺激間の相関關係を利用し、ヘブ則によって学習が行われる。これによって、感覚刺激間の経時的な変化に対する適応学習が可能である。一方、ニューラルネットワークを用いた学習では、視覚と運動、聴覚と運動の二つのネットワークを形成したものがある。この手法は、予め学習された視覚情報を用いて、聴覚刺激と運動間の入出力關係を学習する。しかし、これらの研究は、視野外の音源については考慮されていない。この問題を解決するために、視野外の音源に対しても学習することが可能なアルゴリズムが提案されている [26; 27]。[26; 27] では、センサ・モーター間の入出力關係を学習するために SOM をカスタマイズ（以降、customized-SOM）している。この customized-SOM の基本構造は SOR と同じであるが、評価値に基づく学習手法ではない。これに起因して customized-SOM は、出力参照ベクトルを学習するために誤差信号を必要とする。先に述べたように、自律ロボッ

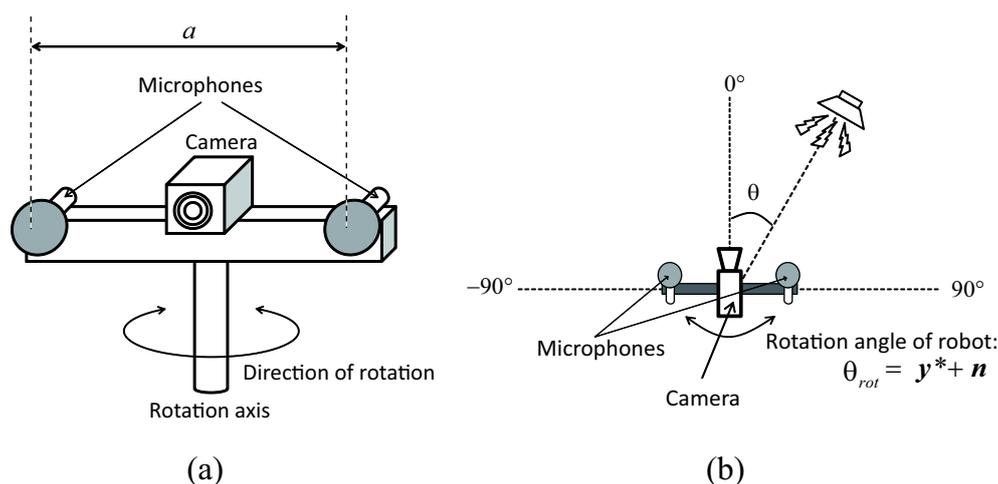


図 4.1: ロボットシステム . (a) ロボットシステムの概念図 , (b) ロボットシステムと音源との関係 .

トの学習においては、答えを得ることは事は難しい . この問題を解決するために [26; 27] では、教師となる誤差信号を取得するための処理手順を学習プロセスに追加した (図 4.2(b)) . しかし、教師を取得するための処理手順を追加したことによって、1 回の試行 (音源提示) に対してロボットは 3 回も回転しなければならない . 実応用において、音源は学習が完了するまで静止しているとは限らないので、これは重大な問題である . したがって、教師信号を必要としない学習手法を適用することが望まれる . さらに、customized-SOM は、古典的な SOM の学習則を用いているので、学習係数のスケジューリングが必要である . これによって、適応学習ができない . 実環境下で動作する自律ロボットにとって、環境やロボット自身の変化に対する適応性は必要不可欠である .

そこで本章では、上で述べた自律ロボットの音源定位学習に関する三つの要点を解決するために DSOR を適用し、その有効性の検証を行う .

4.2 実験の枠組み

ここでは、音源定位行動獲得のシミュレーション実験について述べる . シミュレーションに用いるロボットシステムを図 4.1 に示す . ロボットシステム的前提条件は以下のように設定した .

- ロボットシステムは回転軸上に設置された 1 個のカメラと、カメラの両脇に 20cm の間隔で

設置された二つのマイクから構成される

- ロボットは垂直軸を中心にして図 4.1(a) の回転方向に回転可能
- 音源は単一かつ静止しているものとする
- 音源定位を行う範囲は、 -90 度から 90 度の範囲とする。

実験の流れ図を図 4.2(a) に示し、以下、その流れを説明する。実験開始時のロボットの状態を初期状態と呼び、学習における試行は必ず初期状態から開始するものとする。まず、カメラの正面からの相対角度 θ から音を提示する（図 4.1(b)）。角度 θ はランダムに決定される。定位手掛かりとして、両耳間時間差 $x_A[\text{sec}]$ およびカメラ画像上の音源の位置情報 x_V を用いる。 x_V は画像上の音源のピクセル値とした。視覚的な音源検出能力は既得とする。

x_A と x_V はそれぞれ図 4.3 に示すような関数であり、以下の式によって導出する。

$$x_A = \frac{a \sin \theta}{v_s}, \quad (4.1)$$

$$x_V = \begin{cases} \lceil x_{V,center} + \left(\frac{\theta}{|\theta_V|} x_{V,center} \right) \rceil, & |\theta| \leq |\theta_V|, \\ -1, & otherwise, \end{cases} \quad (4.2)$$

ここで、 a はマイクロフォン間の距離、 v_s は音速、 $x_{V,center}$ は画像の中心のピクセル値を表す。本実験では、 $a = 0.2[m]$ 、 $v_s = 340[m/s]$ 、 $x_{V,center} = 320$ とした。 x_A と x_V を結合し、入力ベクトル $\mathbf{x} = (x_A, x_V)$ を生成する。 \mathbf{x} は DSOR へ入力され、 $\mathbf{y}^* + \mathbf{n}$ が生成される。ここで、試行信号の分散値は、次式によって計算される。

$$\sigma_{noise}^2 = \exp \left\{ -\frac{(\mathcal{E}_c + 1)}{\sigma_0} \right\} - \exp \left(-\frac{2}{\sigma_0} \right) \quad (4.3)$$

ここで、 σ_0 は関数の減衰率を表す。図 4.4 にこの関数を示す。 $\mathbf{y}^* + \mathbf{n}$ をロボットの制御信号としてモータを回転する。ここで、ロボットの回転角 θ_{rot} は、 $\mathbf{y}^* + \mathbf{n}$ と線形の入出力関係を仮定した。

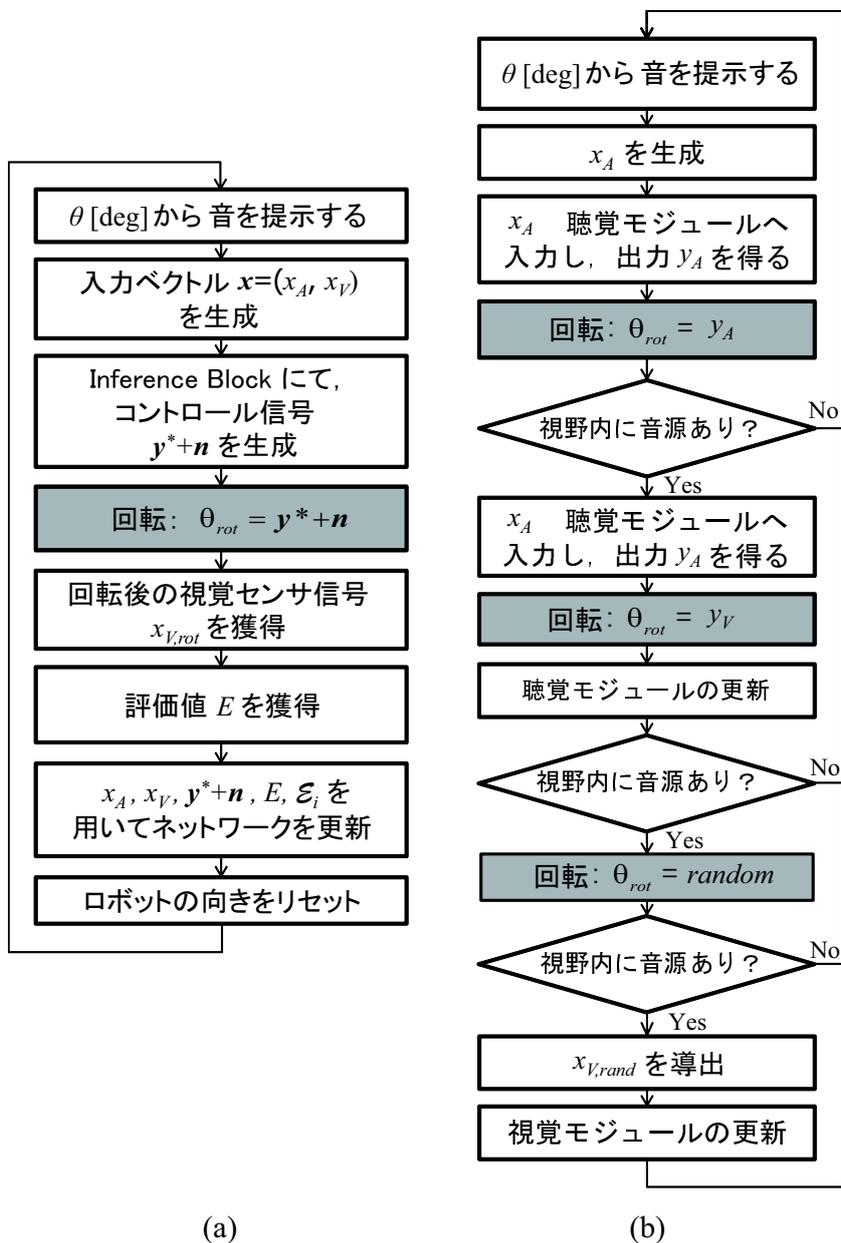


図 4.2: 音源定位能獲得実験の流れ図 . (a) 提案手法を応用した場合の流れ図 . (b) 先行研究の流れ図 [27] .

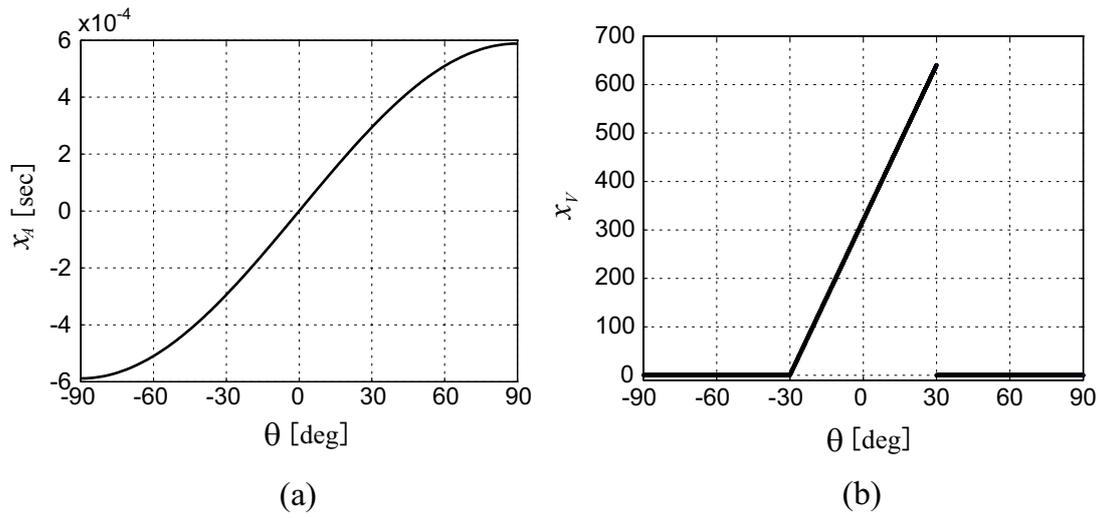


図 4.3: 音源定位の手掛かり . (a) 両耳間時間差 (聴覚情報) : x_A . (b) 画像上における音源のピクセル値 (視覚情報) : x_V .

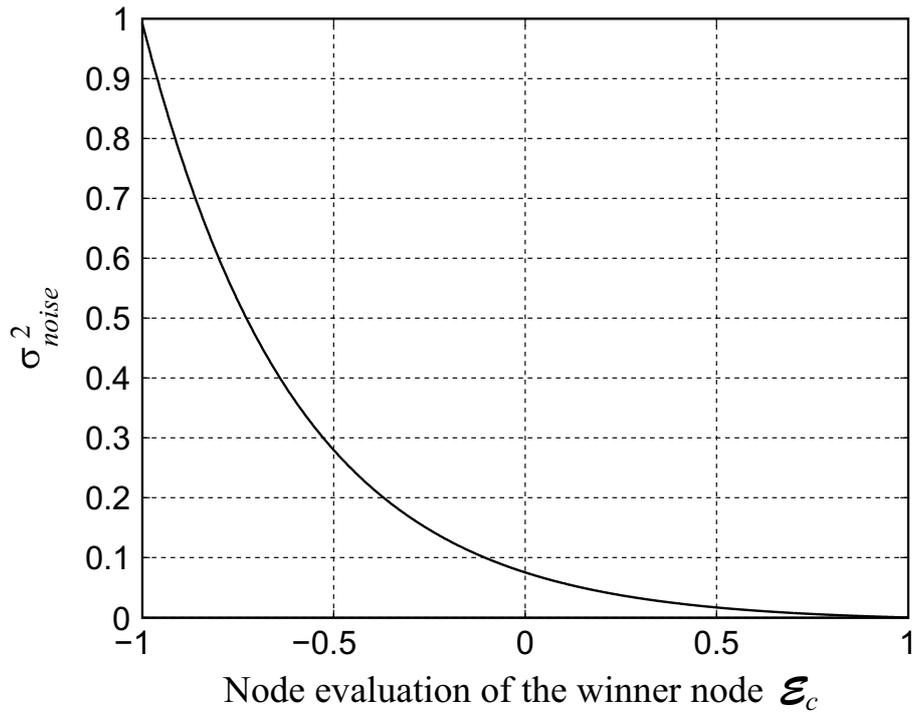


図 4.4: ノード評価値と試行信号の分散値の関係 .

ロボットの回転後，カメラ画像を取得し，画像上における音源の位置 $x_{V,rot}$ を得る． $x_{V,rot}$ を入出力評価器で評価し，評価値 E を得る．本実験では，評価値 E は次式によって定量的に計算した．

$$E = \begin{cases} \left(\frac{x_{V,rot}}{x_{V,center} - x_{V,border}} \right) - 1, & \text{for } x_{V,rot} \leq x_{V,center} - x_{V,border} \\ \left(\frac{x_{V,rot} - x_{V,center}}{x_{V,border}} \right) + 1, & \text{for } x_{V,center} - x_{V,border} < x_{V,rot} \leq x_{V,center} \\ 1 - \left(\frac{x_{V,rot} - x_{V,center}}{x_{V,border}} \right), & \text{for } x_{V,center} < x_{V,rot} \leq x_{V,center} + x_{V,border} \\ \frac{x_{V,rot} - x_{V,center} - x_{V,border}}{x_{V,center} - x_{V,border}}, & \text{for } x_{V,center} + x_{V,border} \leq x_{V,rot} \\ -1, & \text{others} \end{cases} \quad (4.4)$$

ここで， $x_{V,border}$ は中心窩の最大受容範囲を表し，次式によって算出される．

$$x_{V,border} = \frac{\theta_{V,border}}{|\theta_V|} x_{V,center}, \quad (4.5)$$

ここで， $\theta_{V,border}$ は中心窩の最大受容角度を表す．式 (4.4) および式 (4.5) は図 4.5 に示すように，もし音源を $x_{V,center} \pm x_{V,border}$ の範囲に音源をとらえることができれば， E には正の値が割り当てられることを意味する．式 (4.4) の計算結果が $E < -1$ である場合およびロボットの回転後の視野内に音源が存在しない場合は， $E = -1$ とした． $x, y^* + n$ および E によってネットワークの更新を行う．ネットワークの更新後， ε の更新を行う．更新後，ロボットの状態を初期位置に戻し，次の試行を行う．

本実験では，音源の角度が視野角 θ_V よりも大きい場合，視覚情報 x_V は得られない．そこで， x_V の有無によらず学習ベクトルを扱えるように，聴覚ノード (1D ノード) と視聴覚ノード (2D ノード) の二つを定義して学習を行った．1D ノードは w_{i1} の値のみを持ち，2D ノードは w_{i1}, w_{i2} の両方の値を持つノードとした．もし入力ベクトルが x_A の値しか持たない場合，新規ノードの追加およびノードの更新は w_{i1} 空間上のみで行う．逆に入力ベクトルが x_A, x_V の両方の値を持つ場合は， $w_{i1} - w_{i2}$ 空間上で更新処理を行う．全ての実験は，MTALAB を用いたシミュレーションによって行った．

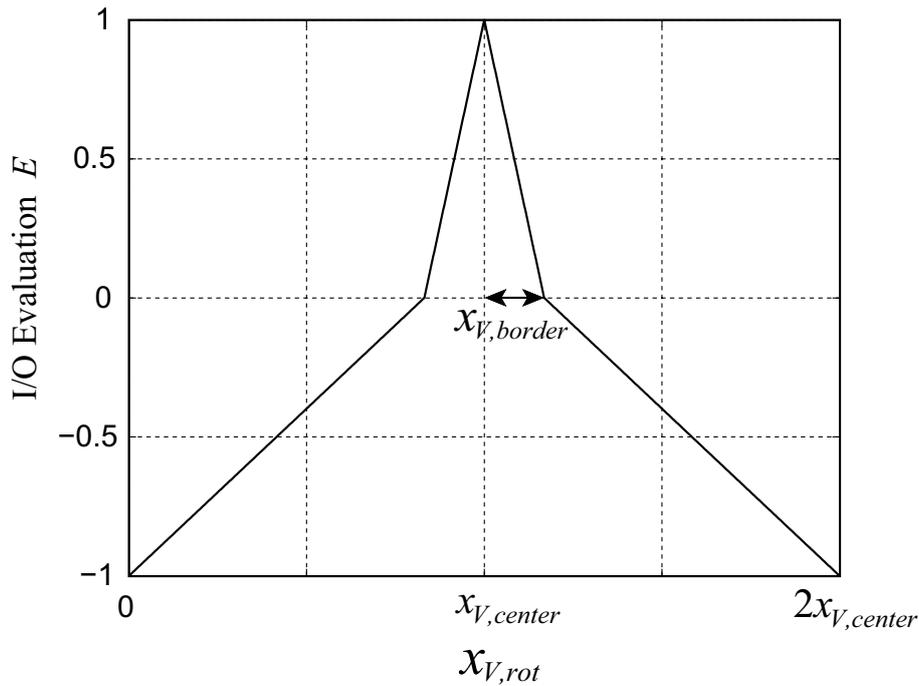


図 4.5: 評価関数 .

4.3 音源定位能の獲得

本節では、音源定位能力の獲得実験を行う。ここでは、ノード数がゼロの状態から音源定位能力を獲得する過程について述べる。本実験の目的は、DSORの音源定位学習問題に対する有用性を実証することである。

DSORの各パラメータは、試行回数5000回、初期ノード数0、 $\phi = 0.01$ 、 $\theta_V = \pm 30^\circ$ 、 $\omega = 0.8$ 、 $T_p = 100$ 、 $\theta_{V,border} = 5^\circ$ 、 $\alpha_w = 0.001$ 、 $\alpha_u = 1$ とした。シミュレーションにおいて、評価値は式(4.4)を用いて定量的に導出した。本実験では、ロボットの定位誤差が 5° 以内の場合のみ正の評価が得られる。

図4.6–4.9に実験結果を示す。図4.6–4.9(a)は、入出力空間、(b)および(c)の上段はそれぞれ、 $w_{i1} - u$ 空間および $w_{i2} - u$ 空間を表す。各図中の空間上のおよびはそれぞれ、1Dノードおよび2Dノードを表し、各ノードを結ぶ直線はリンク結合を表す。図4.6–4.9(a)において、各ノードと点線でつながれた $w_{i1} - w_{i2}$ 空間上の円は、 $2\phi_i$ の等高線を示す。もし入力ベクトルがこの円のいずれの範囲内にも含まれない場合、ネットワークに新規ノードを追加する。 $w_{i1} - u$ 空間の横軸

の0と1はそれぞれ可聴領域 -90° と 90° に対応し, $w_{i2}-u$ 空間の横軸の0と $1/3$ はそれぞれ視野角 -30° と 30° に対応する. 図4.6–4.9(b)および(c)の中段に示されたガウス関数は,各ノードのメンバシップ関数を表し,直線と点線はそれぞれ,1Dノードと2Dノードを表す. 図4.6–4.9(b)および(c)の下段は各ノードのノード評価値を表す.

図4.6–4.9(a)より,入力空間は生成されたノードによって分割されていることがわかる.各ノードは受容野領域に相当する $[w_{im}-2\phi_i, w_{im}+2\phi_i]$ の範囲に対する入力を担当する. 図4.8–4.9(a)から分かるように, $w_{i1}=0.8$ 付近において1Dノードと2Dノード間にリンク結合が生成されている.これは,画像情報を持たない入力信号に対する更新アルゴリズムによるものである.視覚情報 x_V が得られない場合,ネットワークは w_{i1} 空間上のみで更新が行われる.

図4.10に獲得された知識を示す.この図は,図4.9(a)の $w_{i1}-w_{i2}$ 平面を上から見た図である.各ノードによって分割された領域は,それぞれのノードの出力参照ベクトルの値 u_i にしたがって色づけされている.以下に例として,2Dノード i および1Dノード j の表現するファジィif-thenルールを示す.

$$2D \text{ node } i: \text{ IF } x \text{ is } \tilde{w}_i, \text{ THEN } y \text{ is } u_i, \quad (4.6)$$

$$1D \text{ node } j: \text{ IF } x_A \text{ is } \tilde{w}_{j1}, \text{ THEN } y \text{ is } u_j. \quad (4.7)$$

ここで, \tilde{w}_i および \tilde{w}_{j1} はそれぞれ,前件部メンバシップ関数を表す. 図4.11に,ノード数の時間変化を示す.ノード数は,学習初期段階において望ましい入出力関係を表現できるノード数まで一気に増加し,その後安定する.ノードの増加傾向は,入力空間における入力データの分布や,提示の順序などに関係して変化すると考えられる.ここでは,入力空間上に一様に分布した入力データをランダムに提示した結果であり,ノード数の増加速度は最も早い場合であると考えられる.

図4.12に,定位誤差の時間変化を示す.図より,定位誤差は,学習の初期段階において急速に減衰し,遅くとも $t=200$ までには,全ての提示角度 θ において 30° 未満に収束している事がわかる.これは,ロボットが,わずか200回の音の提示によって,いかなる方向から提示された音源であっても視野内にとらえられるようになった事を意味する. 図4.13に, $t=5000$ における定位誤差とその標準偏差の100回平均を示す.音源角度 θ 毎に定位精度が異なるのは,図4.3(a)に示した聴覚手掛かり x_A の非線形性から生じる. 図4.3(a)から分かるように,音源角度 θ の絶対値が

大きくなるに従って、マイク間時間差の変化量が小さくなる。従って、音源角度が大きくなるに従って特徴量 x_v の違いを判別することが困難になり、定位精度が低下する。この現象は、ヒトやその他の生物の定位においても同様である。図 4.13 には、比較のために、先行研究 [27] の定位誤差とその標準偏差の 100 回平均も示している。ここで、先行研究の SOM のノード数は、提案手法におけるノード数 (図 4.11) と同程度にするために 40 個とした。図 4.13 より、提案手法の定位誤差は、 $\pm 60^\circ$ の範囲において 2.5° 未満であり、 $\pm 85^\circ$ の範囲内において 5° 未満である。 90° における定位誤差が比較的大きくなっているのは、エッジ効果による推定誤差の影響である。提案手法の定位誤差は、 $\pm 40^\circ$ 以内の範囲においては先行研究よりも小さい。一方、それよりも角度が大きくなると、先行研究の定位誤差の方が小さくなる。しかし、ほぼ全ての領域において、両者の定位誤差は 5° 未満に収まっている。ヒトの定位誤差は、 $\theta = 0$ で $\pm 3.6^\circ$ 、 $\theta = |90|^\circ$ では $\pm 10^\circ$ である [44] ことを考慮すると、提案手法の定位精度は実問題に対しても十分な精度を実現できていると考えられる。ここで、提案手法と先行研究の定位精度は同程度であったが、図 4.2 において比較したように、提案手法における学習の流れは、先行研究における学習の流れに比べて、非常に単純である。したがって、提案手法は、先行研究に比べて、音源定位の自律獲得問題に対してより効果的な学習手法であると言える。

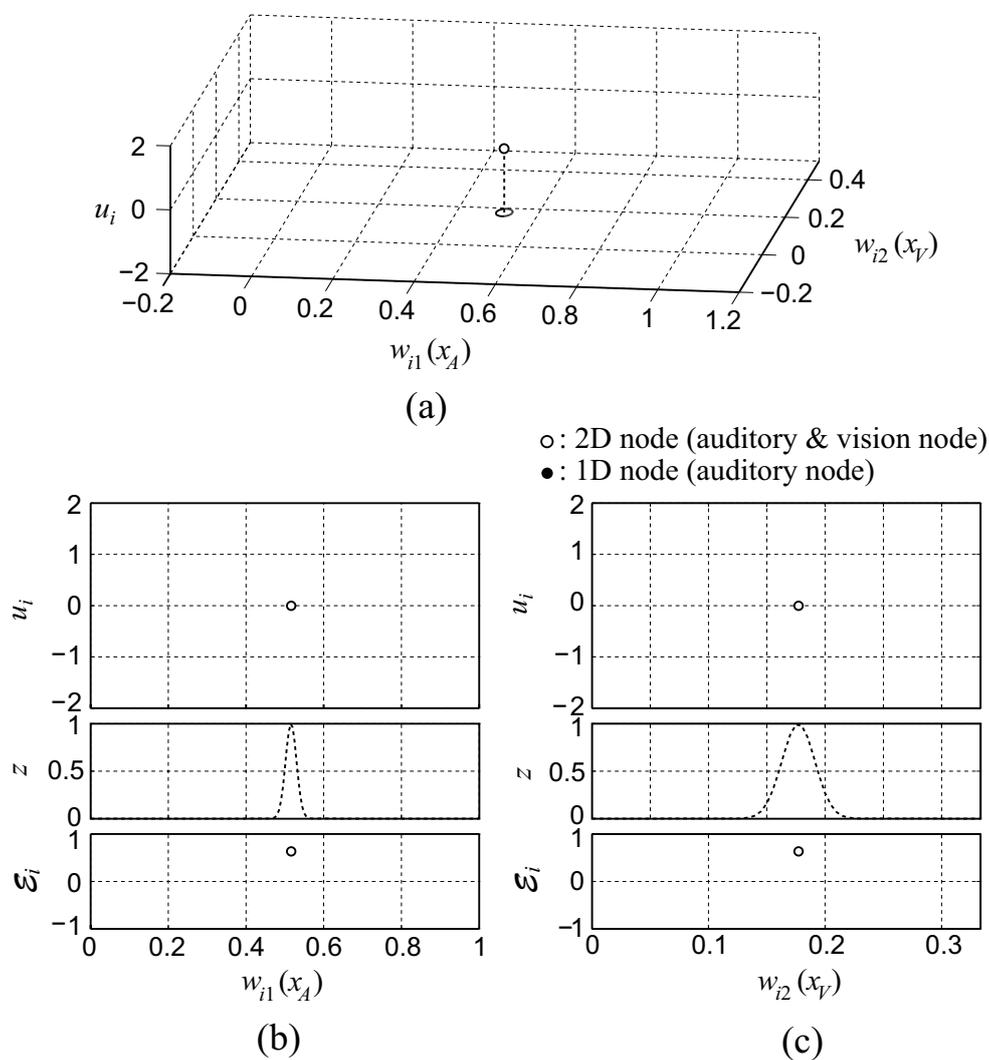


図 4.6: 実験結果 ($t = 1$) . (a) 入出力空間 , (b) $w_{i1} - u$ 空間 . (c) $w_{i2} - u$ 空間 . (a) および (b)(c) の上段の ○ は 2D ノードを表す . (a) のノードと点線でつながれた $w_{i1} - w_{i2}$ 空間上の円は , $2\phi_i$ の等高線を表す . (b) の横軸は可聴領域 -90° から 90° , (c) の横軸は視野角 -30° から 30° の範囲に対応する . (b)(c) 中段のガウス関数は , ノードのメンバシップ関数を表す . (b)(c) 下段はノード評価値を表す .

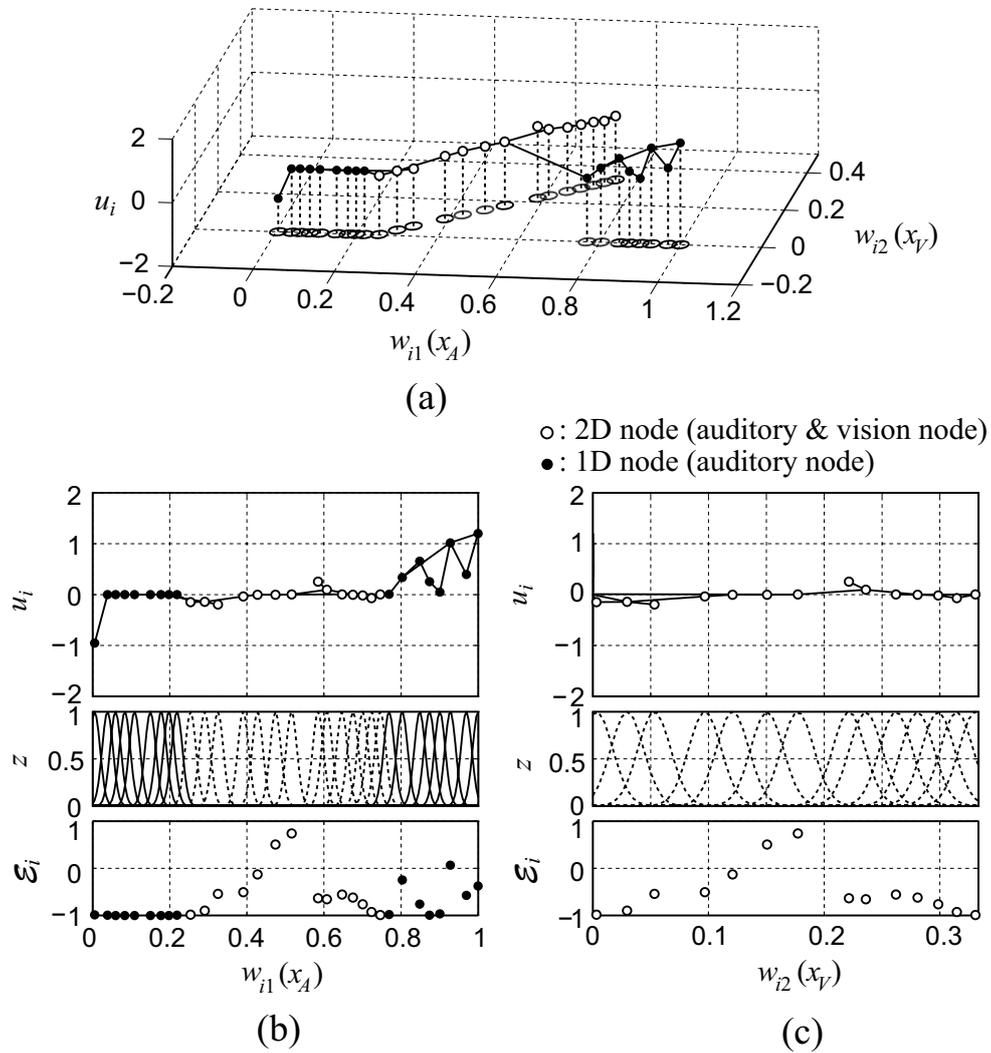


図 4.7: 実験結果 ($t = 100$) . (a) および (b)(c) の上段の ○ は 1D ノードを表す . 各ノードを結ぶ直線はリンク結合を表す . (b)(c) 中段のメンバシップ関数の直線と点線はそれぞれ , 1D ノードと 2D ノードを表す .

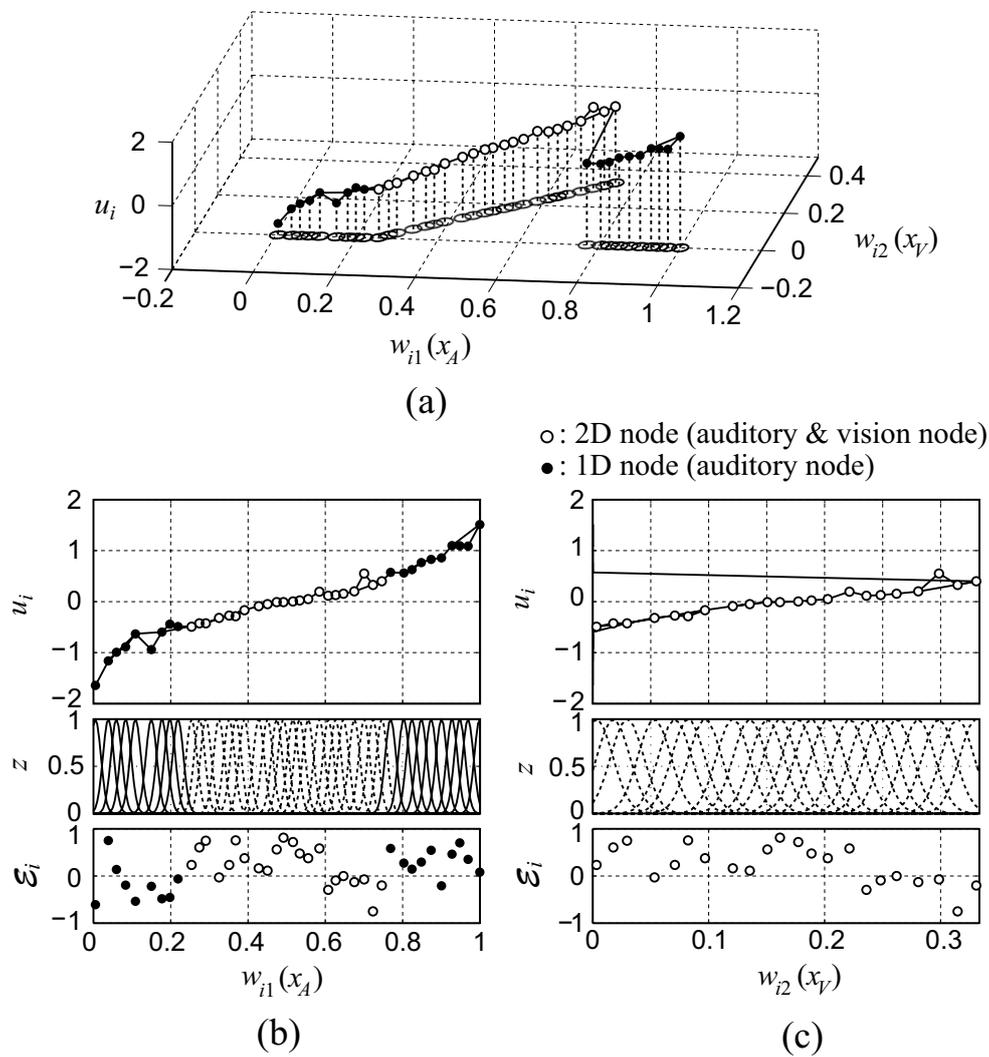


図 4.8: 実験結果 ($t = 1000$) .

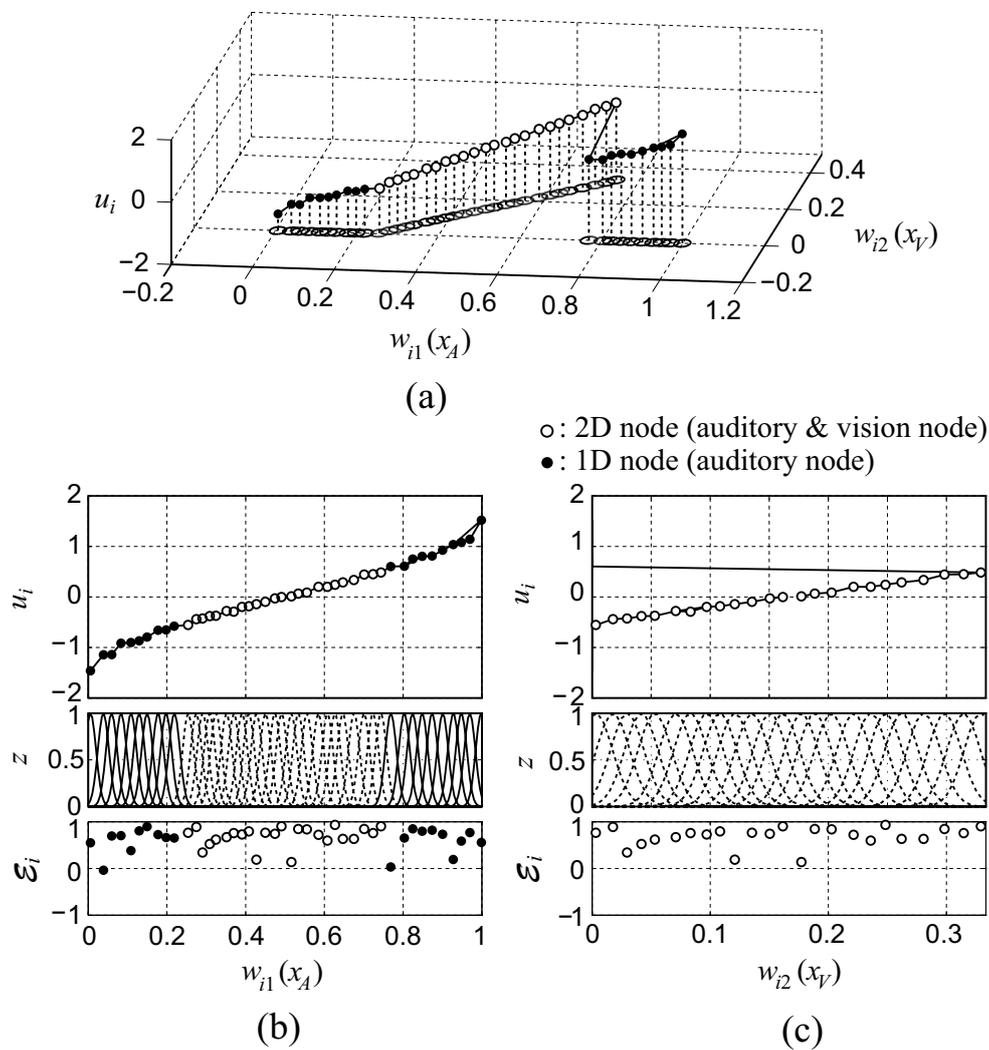


図 4.9: 実験結果 ($t = 5000$) .

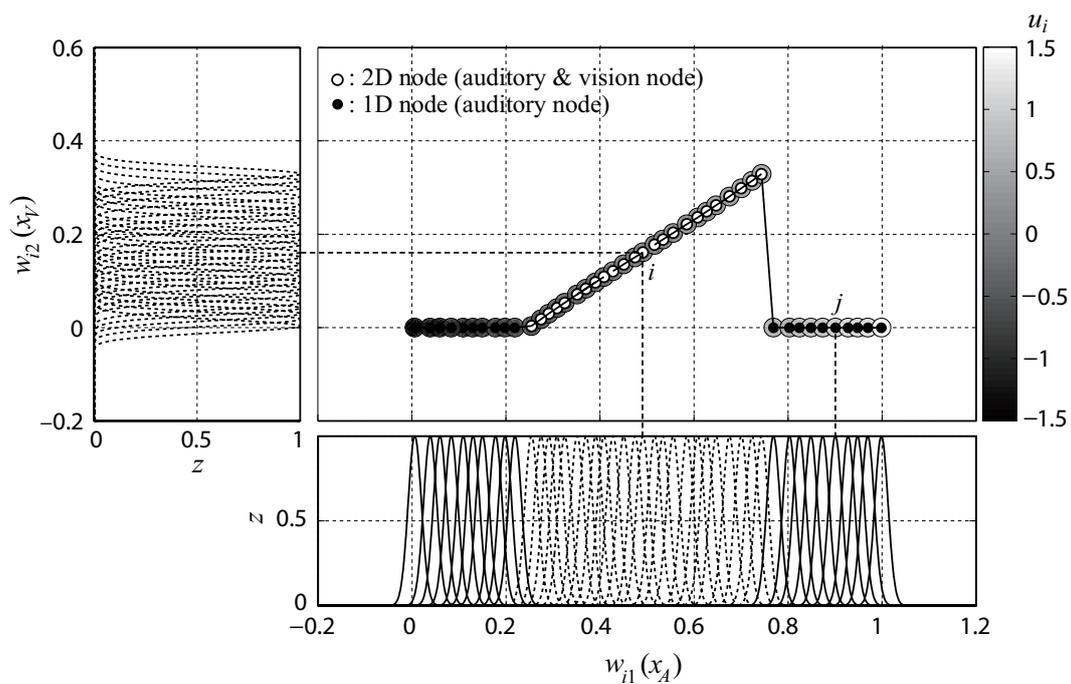


図 4.10: 獲得された知識 ($\phi = 0.01$) .

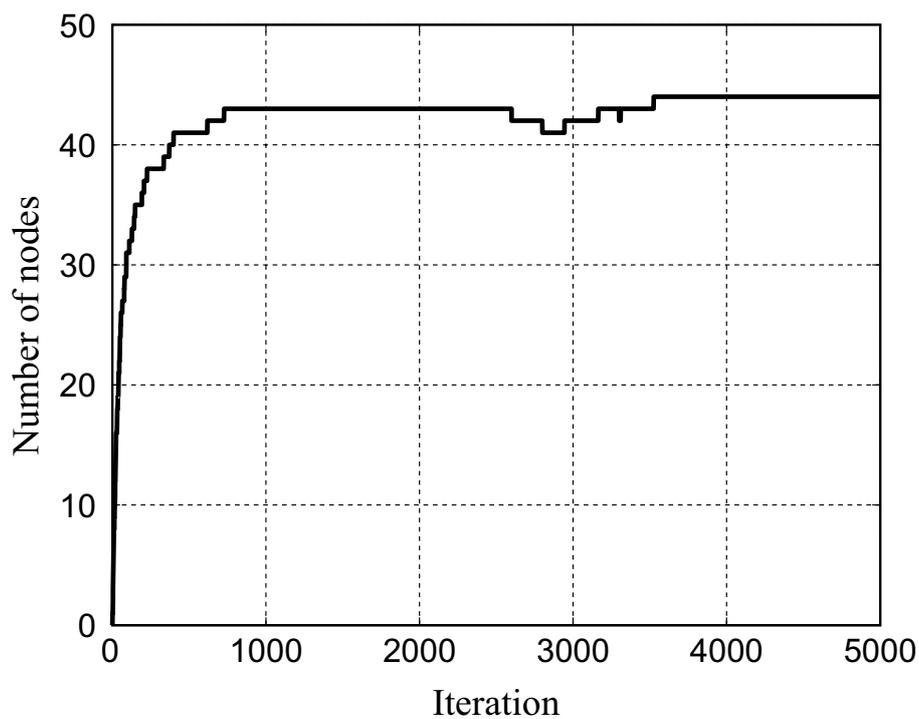
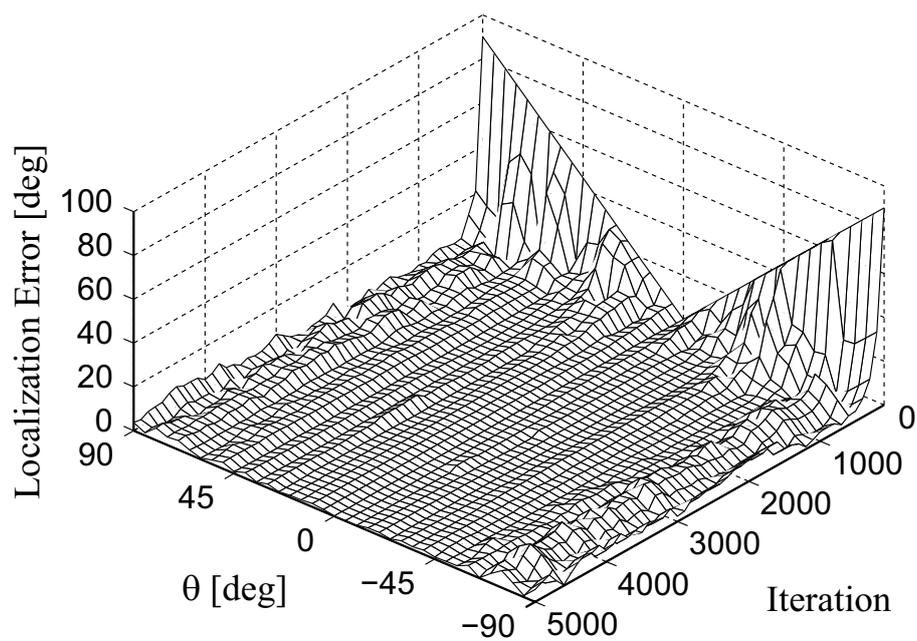
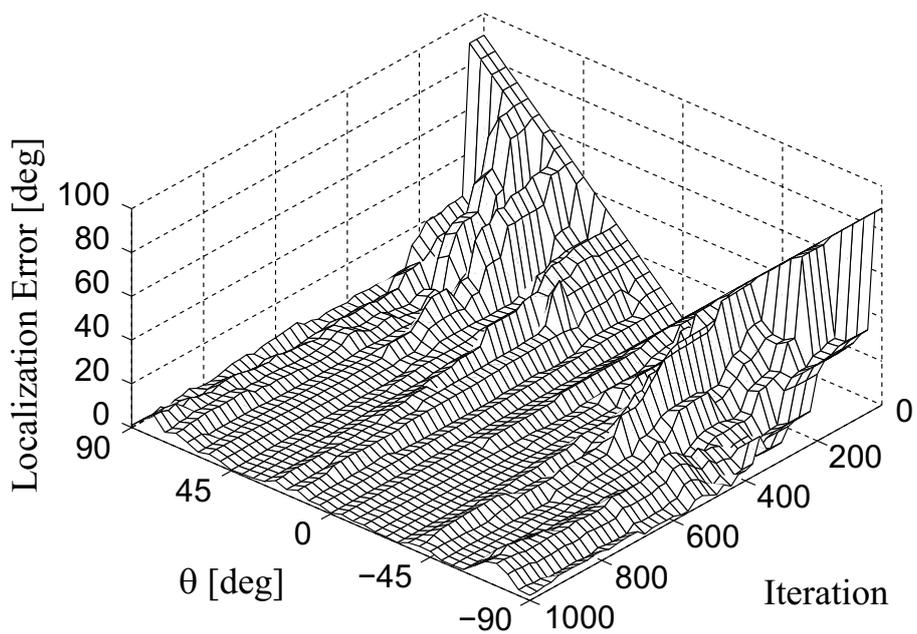


図 4.11: ノード数の変化 .



(a)



(b)

図 4.12: 定位誤差の時間変化 ($\phi = 0.01$) . (a) 全体 . (b) 学習初期段階 .

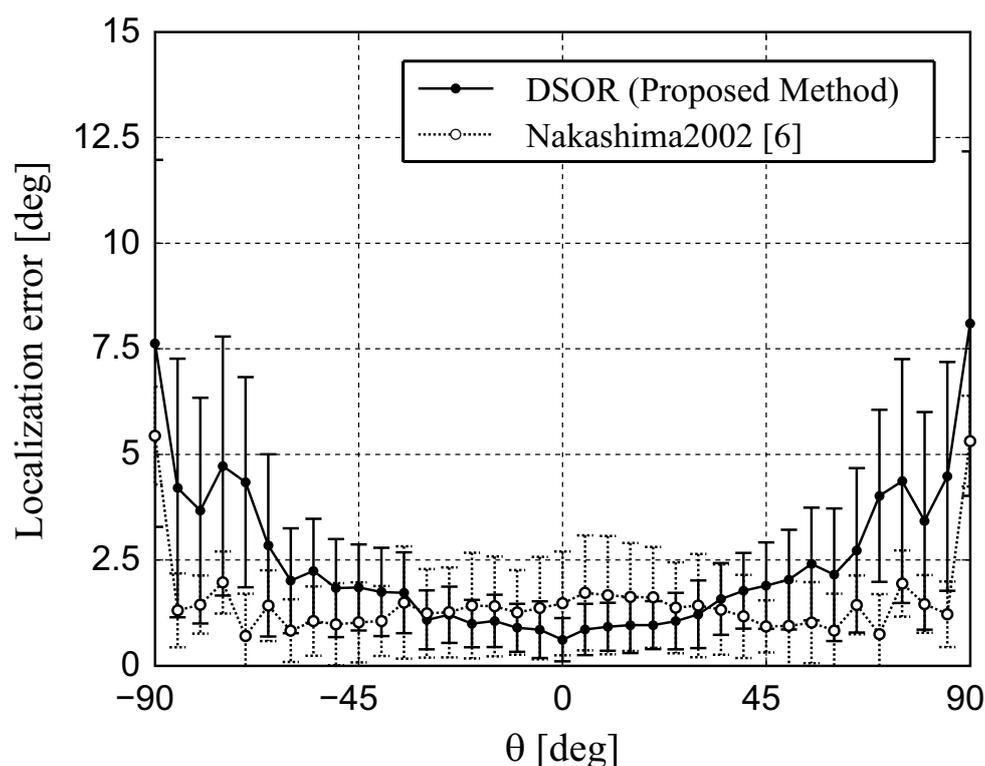


図 4.13: 最終的な定位誤差 (100 回の試行の平均値)

4.4 適応学習

システムの変化に対する DSOR の適応性を実証するために、図 4.14 に示すように、カメラの角度を θ_e 回転させて実験を行った。視覚情報のシフトや反転などの大きな変化に対する音源定位の優れた適応性はヒトにおいても明らかになっている [45]。ロボットのセンサ信号は、環境やロボットの構造などに大きく影響されるので、音源定位ロボットには高い適応能力が必要不可欠である。視覚情報を左右反転させる実験は先行研究 [27] でも行われているが、視野を反転させるタイミングは学習途中である。先行研究 [27] で提案された学習手法では、時間的に学習係数を減少させる必要があるので、学習が完了した状態で新たな入出力関係を学習することはできない。

これに対して本実験では、5000 回の音源提示後のネットワーク状態から学習を開始する。シミュレーションのパラメータは 4.3 に等しい。適応学習の試行回数は 5000 回、反転された視覚情報は次式によって計算した。

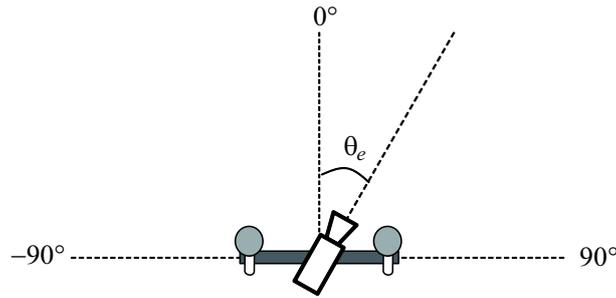


図 4.14: カメラの角度が θ_e ずれたロボットシステム.

$$x_V = \begin{cases} \lceil x_{V,center} + \left(\frac{(\theta + \theta_e)}{|\theta_V|} x_{V,center} \right) \rceil, & |\theta - \theta_e| \leq |\theta_V|, \\ -1, & otherwise, \end{cases} \quad (4.8)$$

本実験では, $\theta_e = 30^\circ$ とした.

図 4.15-4.17 に適応学習の結果を示す. 図の見方は 4.3 と同じである. カメラを回転させる前の状態 (図 4.9) から, カメラを回転させた後, 音を 100 回提示した状態を図 4.15 に示す. 図 4.15 より, 環境が変化したことによって, ノード数が増加し, ノード評価値が低下していることが分かる. ノード評価値が低下すると, 式 4.3 に従って試行信号の振幅が大きくなり, 再学習が始まる. 4.16 に示すように, 再学習が終わると, 新しい入出力関係を表現するために必要なノードのみが残り, それ以外は削除される. 最終的なノード数は, 図 4.18 に示すように, 適応前と同程度に収束する. 図 4.19 に, 適応学習における定位誤差の時間変化を示す. 図中, $t = 1$ から $t = 5000$ までは, 4.3 の結果である. $t = 5001$ から $t = 10000$ が適応実験の結果を表す. 図 4.19 より, 定位誤差は, 環境が変化した後一時的に大きくなる. しかし, その後すぐに, 環境が変化する前の水準まで減衰する. 図 4.20 に, $t = 10000$ における定位誤差とその標準偏差の 100 回平均を示す. この図より, 定位誤差は, 図 4.13 に示した, 適応前と同程度に収束していることがわかる. これらの結果より, 提案手法は, 環境の変化に対して十分な適応能力を有する事を実証した.

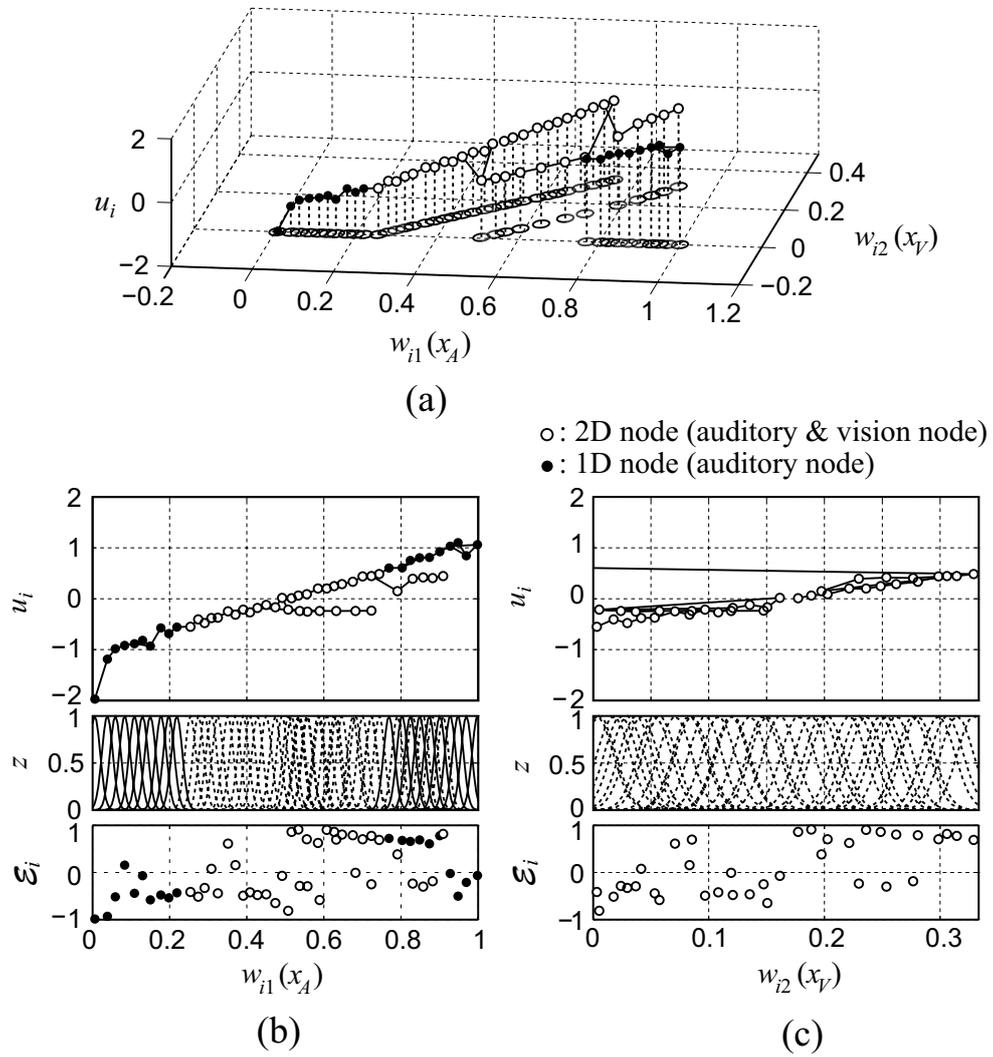
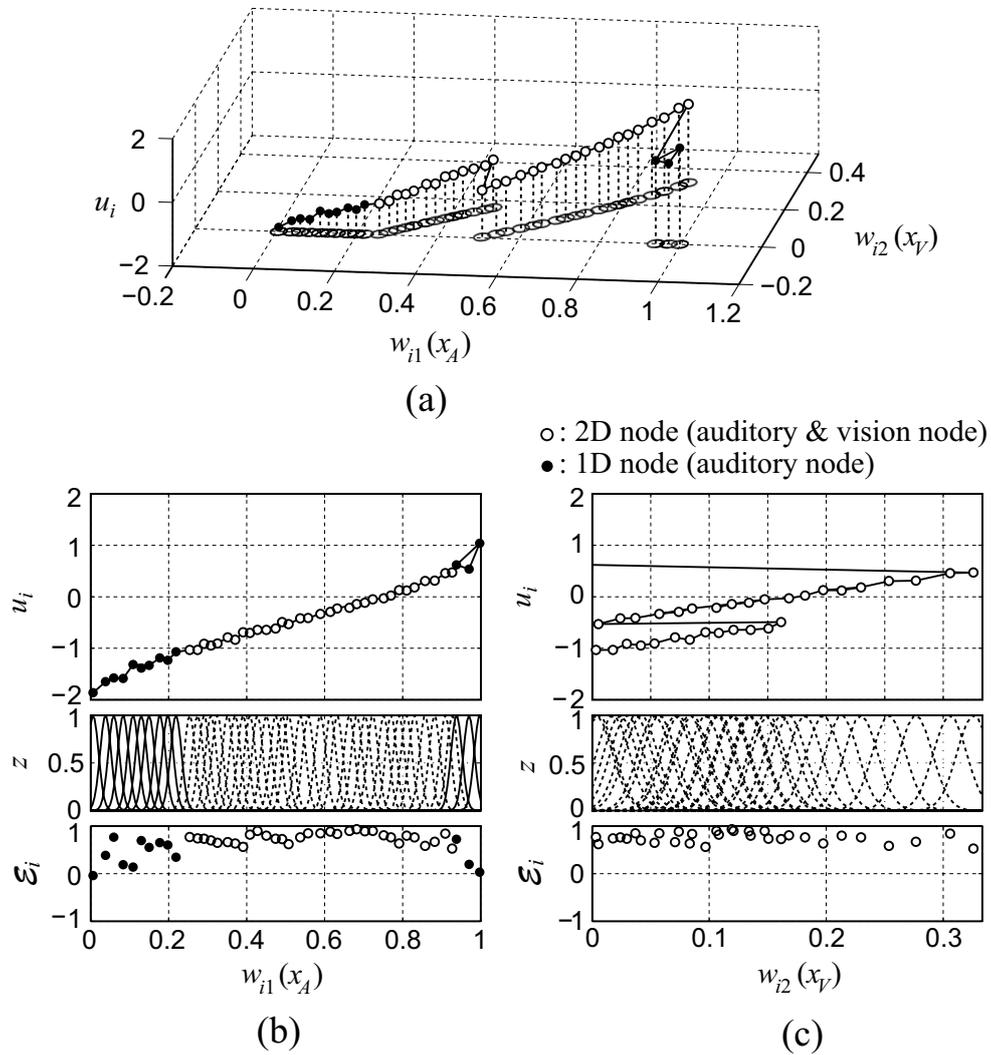


图 4.15: 適応実験結果 ($t = 5100$) .

図 4.16: 適応実験結果 ($t = 10000$) .

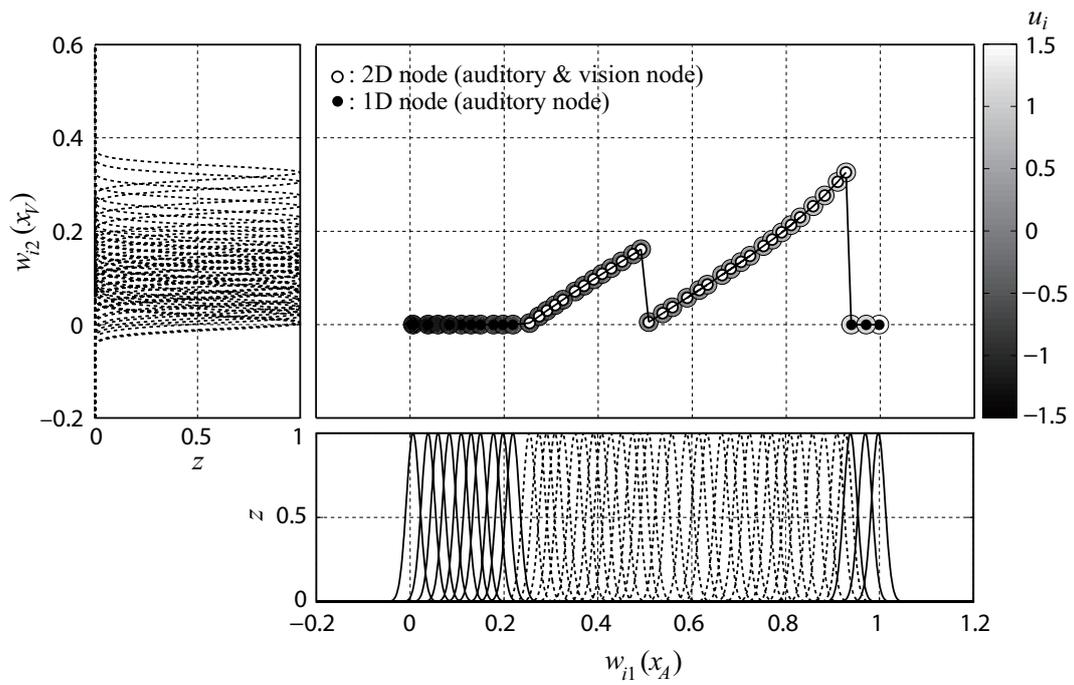


図 4.17: 適応学習によって獲得された知識 .

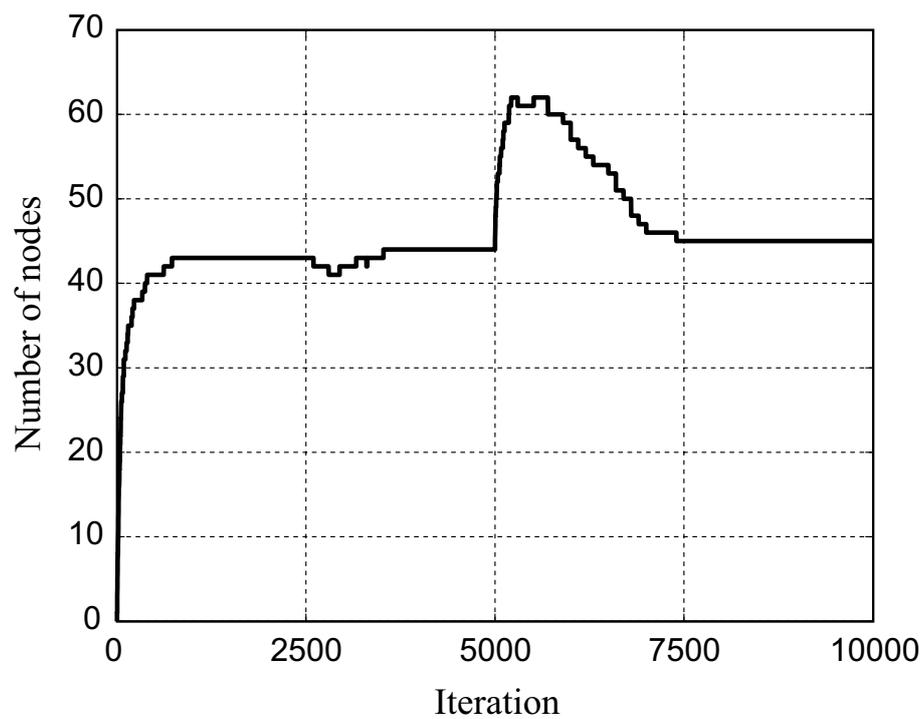


図 4.18: ノード数の時間変化 .

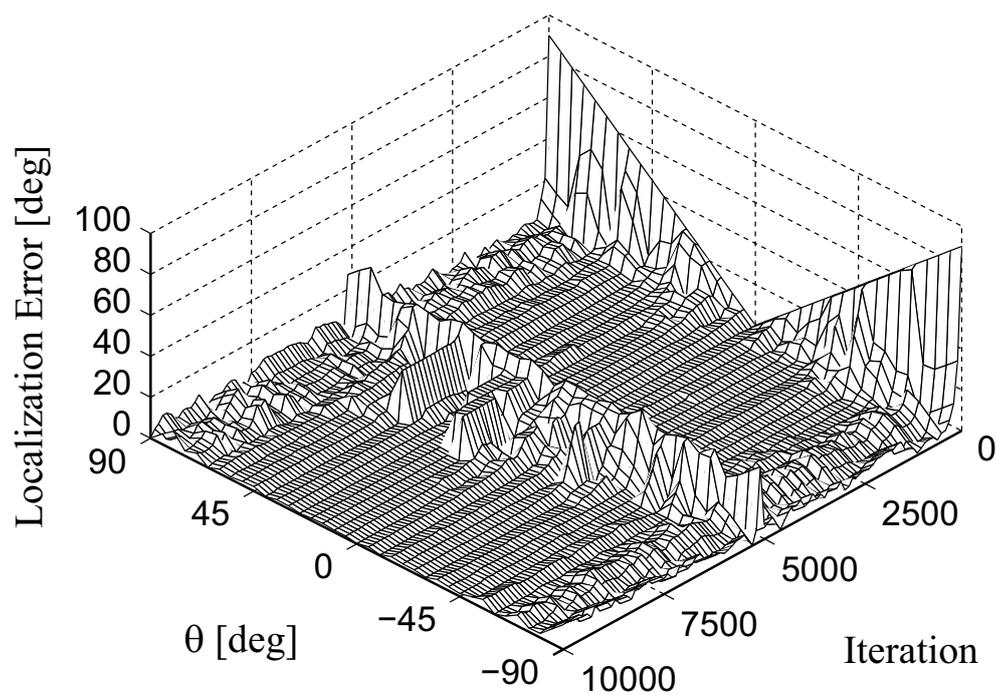


図 4.19: 定位誤差の時間変化 .

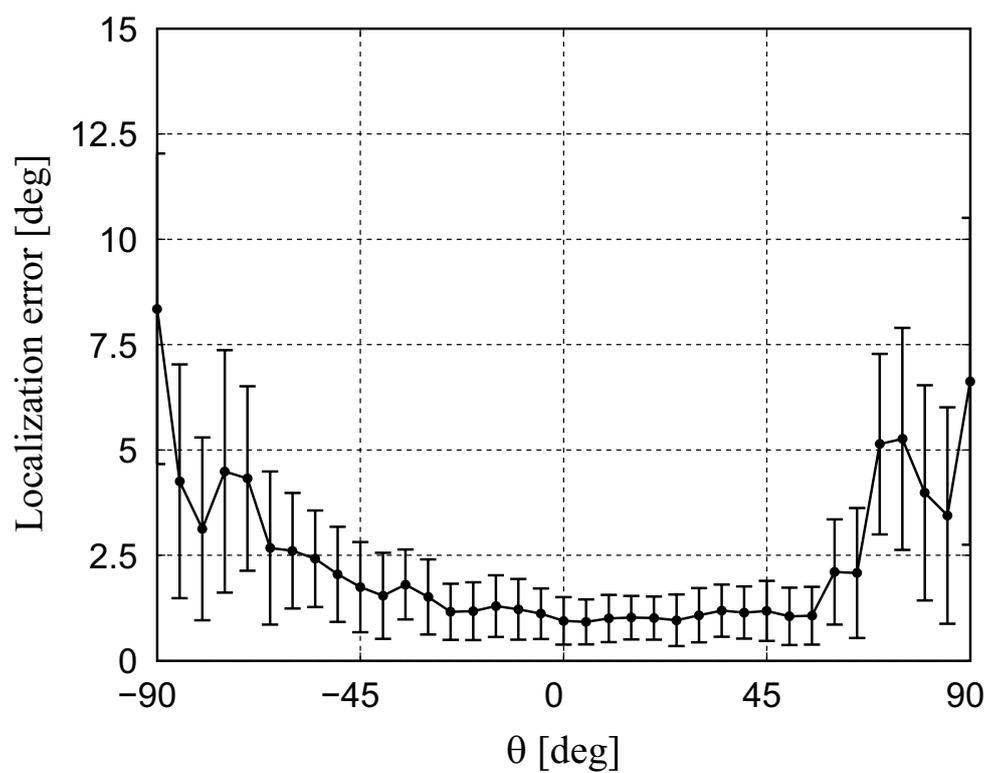


図 4.20: 最終的な定位誤差 (100 回の試行の平均値)

4.5 考察

ここでは、4.1 で述べた、音源定位学習に必要な三つの条件について考察する。

先行研究 [27] においては、センサ入力とモータ出力間の入出力関係を学習するために customized-SOM が用いられていた。しかし、この手法は、定位誤差の情報を教師として学習していた。この誤差情報を獲得するために、学習の手順は非常に複雑になってしまっていた。具体的には、図 4.2(b) に示すように、1 回の音源の提示に対してロボットが 3 回回転しなければならなかった。この複雑な学習手順は、話者認識のような実環境における音源定位の実応用において重大な問題となる。実環境におけるこれらの実応用において、音源はロボットが学習する間、同じ場所に留まり続けているとは限らない。さらに、ロボットの回転後に音源が視野内にとらえられなかった場合、試行が無駄になっていた。一方、DSOR は評価値付き学習手法なので、明示的な教師信号を必要としない。これによって、DSOR を用いた学習手順は、先行研究 [27] よりも単純化された。具体的には、1 回の音の提示に対して、ロボットが 1 回回転するだけでネットワークを更新することが可能になり (図 4.2(a))、ロボットの実用性が増した。さらに、4.3 で述べたように、提案手法の学習手順は、先行研究に比べて単純化されたにもかかわらず、学習の速度と定位精度は同程度であった。音源定位学習における提案手法のこれらの優位性から、DSOR は自律ロボットの音源定位学習問題において、先行研究 [24; 25; 27] よりも有効な学習手法であると言える。

先行研究 [24; 25; 27] において、聴覚-運動間の関係と視覚-運動間の関係は、それぞれ異なるネットワークによって学習が行われていた。一方、本実験では、1D ノードと 2D ノードという 2 種類のノードを使用した。これらのノードは、単一のネットワークのみで、聴覚-運動間の関係および視覚-運動間の関係の両方を学習することを可能にした。これによって、定位精度は保持したまま、ネットワーク全体における実質的なノード数は半分になった。

上記の優位性に加えて、DSOR は、適応学習能力を有する。環境の変化やロボットシステム自体の変化に対する頑健性は、自律ロボットの学習において非常に重要である。[27] では、customized-SOM の適応性について議論されていた。しかしこの手法は、SOM と同じ学習則に基づいているので、学習係数のスケジューリングを行う必要があり、本質的に適応能力を持たない。これに対して 4.4 では、DSOR がシステムの入出力関係の変化に対する高い適応能力を有することを示した。さらに DSOR の適応学習では、新しい入出力関係を表現するために不必要なノードは時間と

共に削除された。そして、適応後のネットワークは最小限のノードかつ新たな環境に対して最適な構造に再構築された。結果的に DSOR は、環境が劇的に変化した場合においても、高い定位精度を維持することができた。

4.6 おわりに

本章では、DSOR を音源定位能の自律獲得問題へ適用し、その有効性について実証を行った。

人間は、視野外の音源に対しても正確な定位を行うことが可能である。ロボットの音源定位を実現する方法としては、解析的な手法と、学習を通して定位能を獲得するものが挙げられる。ここでは、ハードウェアコスト、計算コスト、環境やロボットシステムの変化に対する適応性などの観点から、学習によって音源定位能を獲得する方法を採用した。

4.1 で述べたように、自律ロボットの音源定位能獲得のためには、以下のような要件をクリアする学習方法を考える必要がある。

1. 教師信号なしに学習が可能であること
2. 複数の異なる感覚信号を利用して学習が可能であること
3. 環境の変化やロボット自体の特性の変化に対する適応学習が可能であること

これらの点を考慮した音源定位学習手法は多数提案されている。中でも、[27] において提案された手法は、視野外の音源に対する定位を行うことができるので、ロボットが周りの全ての環境に対して高い音源定位学習が可能である。しかし、[27] においては、センサ入力とモータ出力間の入出力関係を学習するために customized-SOM を用いていた。この手法では、古典的な SOM の学習則を用いているので、教師信号が必要、適応学習ができないなどの問題点が生じる。

そこで本研究では、上で述べた自律ロボットの音源定位学習に関する三つの要点を解決するために DSOR を適用し、その有効性の検証を行った。まず 4.3 において、音源定位能を学習によって獲得できることを実証した。自律ロボットはシミュレーション実験において、提示された音に対して試行錯誤的に行動した。そして、行動によって変化した環境を視覚情報を用いて評価し、この評価に基づいて自らの行動則を更新した。最終的に自律ロボットは、望ましい行動則を獲得した。学習の結果、DSOR を用いた学習手法は、先行研究 [27] に比べて学習の流れが簡単化された

にもかかわらず，学習速度および定位精度共に同等の性能を実現した．この結果より，DSOR は，自律的音源定位能の獲得に対して，有効な学習手法であると言える．

さらに，4.4 の適応実験において，DSOR 適応性が実証された．実験では，音源定位能を獲得済みのネットワークを用いて，ロボット自身の状態を変化させた場合の適応学習をテストした．実験の結果，環境が変化すると，学習済みの入出力関係を用いて高い評価値を得られなくなるのでノード評価値が低下した．それに伴って試行信号の振幅が大きくなり，ネットワークの再学習が行われ，定位誤差は速やかに減少した．また，再学習の過程において，新しい入出力関係を表現するために必要なノード以外は時間と共に削除された．最終的には，再学習前と同程度のノード数で，同程度の定位精度に収束することを実証した．この結果より，DSOR は，各時点におけるセンサ - モータ間の入出力関係を，常に最小限のノード数で表現できることを示した．

また本実験では，視野内の音源と視野外の音源を同時に扱えるように，2 種類のノードを用いた．聴覚情報のみを持つ 1D ノードは，視野外の音源を定位する際に用いられた．これに対して 2D ノードは，視聴覚両方の情報を持ち，視野内の音源の推定に用いられた．これら 2 種類のノードを使い分ける事によって，視覚と聴覚という複数の異なる様式の情報を同時に扱うことが可能になった．また本節では，獲得した知識を図によって示した．このように，獲得した知識を，設計者が容易に理解できる形で表現することによって，入出力関係の直感的な理解が可能になり，さらに効果的な学習手法の提案や，ロボットシステム開発の効率化に繋がると考えられる．

第5章 結論

本論文では、自律ロボットの柔軟かつ効率的な行動学習の実現を目指して、新たな学習手法の提案およびその自律ロボットへの適用を行った。以下、各章で得られた成果について述べる。

2章ではまず、提案手法の基礎となる SOR および二つの適応型 SOR について、それらの構成、学習方法、および特長についてまとめた。SOR は、SOM のベクトル量子化能力および位相保持能力といった特長を継承し、入出力関係を学習できるように改良されたアルゴリズムであり、望ましい入出力関係を記述するファジィ *if - then* ルールの集合を獲得する。さらに、獲得したファジィ *if - then* ルールの集合を用いてファジィ推論を行い、実際の入力に対する望ましい出力を生成する。SOR は学習の際、引力学習と斥力学習を評価値に応じて使い分けることによって、望ましい入出力関係のみならず、望ましくない入出力関係を表す学習ベクトルも利用できる。これによって、学習の効率化および制御不能状態・危険領域の回避が可能になる。SOR はこのような優れた学習能力を持つが、適応性を持たない。この主な原因は、*if - then* ルールを表現するノードの数を予め決定しておく必要があることや、時間と共に減衰する学習係数が用いられていることであった。自律ロボットの自律学習において、予め適切なルール数を知ることは難しい。また、環境やロボット自身の特性はいつ変化するかわからないので、学習係数を予めスケジューリングすることはできない。これらの問題を解決する手法として、ESOR が提案されている。ESOR (Evolving SOR) は、Evolving SOM の進化的学習アルゴリズムを導入することによって、ノードの追加・削除および不変の学習係数による学習が可能になった。また SOR は、学習に際して、学習ベクトルおよびその評価値を事前に獲得しておく必要があった。しかし、自律ロボットの学習において、学習ベクトルおよびその評価値を予め用意することはできない。この問題点を解決する手法として ASOR (Adaptive SOR) が提案されている。ASOR では、実行モードと学習モードを交互に繰り返し行う機能と、実行モードで生成した出力に対して試行信号を付加する機能によって、予め学習ベクトルを獲得しておくことなく、自律的な試行錯誤によって学習することが可能である。

3章では、従来のSORの斥力学習における問題点とその解決策について述べた上で、2章において述べたESORおよびASORのもつ優れた機能を兼備した動的自己組織化関係ネットワーク(DSOR: Dynamical SOR)を提案した。DSORでは、参照ベクトルが必ず望ましい方向へ更新できるよう、ノード評価値という新たなパラメータを導入した。これによって参照ベクトルは、学習ベクトルの評価値の正負によらず、より望ましい向きに更新されるようになった。DSORは、二つの適応型SORの機能を兼備しており、以下の特長を持つ。

- ノードの追加・削除が可能であり、学習ベクトルの分布に応じた適切なルール数で入出力関係の表現が可能である。
- パラメータのスケジューリングが不要であり、適応学習が可能である。
- 入力空間のワンパス学習が可能である。
- 更新プロセスにおいて、勝者ノードおよび勝者ノードとの間にリンク結合を持つノードのみを更新するので、従来のSORに比べて計算コストが安価である。
- 推論と学習を交互に繰り返し行う事によって、自律学習が可能である。
- 推論ブロックの出力に試行信号を付加する事によって、試行錯誤による学習ベクトルの自己生成が可能である。

4章では、DSORを音源定位能の自律獲得問題へ適用し、シミュレーション実験によってその有効性について実証を行った。人間のように、視野外の音源に対しても正確な定位を行うことができるような優れた音源定位機能をロボットに実装することによって、ヒトとロボットのより自然なインタラクションや視覚情報を用いた対象物体の認識の向上が期待できる。自律ロボットの音源定位能獲得のためには、以下のような要件を満たす必要がある。

- 教師信号なしに学習が可能であること
- 複数の異なる感覚信号を利用して学習が可能であること
- 環境の変化やロボット自体の特性の変化に対する適応学習が可能であること

4.3 において、音源定位能の獲得実験を行った結果、ロボットは提示された音に対して自律的な試行錯誤を繰り返しながら、音源定位能を獲得した。実験によって、DSOR を用いた学習手法は先行研究と比べて、学習速度および定位精度を落とすことなく、学習機序を簡単化できることを示した。これらの結果によって、DSOR を用いた学習手法は、音源定位学習を行う自律ロボットの実応用に対して、既存の手法より効果的であることを実証した。さらに 4.4 の適応実験において、環境の変化に対する適応性を検証した。その結果、DSOR は、環境が新しくなると同時に再学習を開始し、新たな環境に対して速やかに適応した。また、適応過程において、新しい入出力関係を表現するために不必要なノードは時間と共に削除された。この結果、DSOR は、各時点におけるセンサ入力とモータ出力の間の入出力関係を、常に最小限のノード数で表現できることを示した。また本実験では、視野内の音源と視野外の音源を同時に扱えるように、聴覚ノード（1D ノード）および視聴覚ノード（2D ノード）の 2 種類のノードを用いた。これら 2 種類のノードを使い分ける事によって、単一のネットワークのみで入出力関係を学習することが可能になり、先行研究に比べて、ネットワーク上の実質的なノード数を半減させることができた。さらに本研究では、獲得した知識を図示した。獲得した知識を設計者が容易に理解可能な形で表現することによって、入出力関係の直感的な理解が可能になり、さらに効果的な学習手法の提案や、ロボットシステム開発の効率化に繋がると考えられる。

謝辞

本研究を通して、懇切丁寧な御指導ならびに様々なアドバイスを賜りました九州工業大学大学院生命体工学研究科脳情報専攻 堀尾 恵一 准教授に心から感謝致します。

本論文をまとめるにあたり、有意義な御助言と御討論を頂いた九州工業大学大学院生命体工学研究科脳情報専攻 永松 正博 教授，同 松岡 清利 教授，同大学大学院生体機能専攻 横井 博一 教授に心から謝意を申し上げます。

博士前期課程におけるマルチタレント英才教育から博士号取得までの長きに渡り、昼夜を問わず終始懇切丁寧な御指導を賜りました九州工業大学大学院生命体工学研究科脳情報専攻 山川 烈 名誉教授に心から感謝致します。

研究に対する意見や激励だけでなく、公私にわたって御世話になった、山川研究室の皆様に厚く感謝致します。

最後に、小学校入学から 22 年間という長きにわたる学生生活を通じ、時に厳しくも暖かく育ててくれた両親と家族，そして優しく見守ってくれた祖父母に心から感謝します。

参考文献

- [1] 山川烈, “21 世紀 COE プログラム - 「生物とロボットが織りなす脳情報工学の世界」”, 知能と情報 (日本知能情報ファジィ学会誌), Vol.17, No.1, pp.68-72, 2005.
- [2] 松原仁, 野田五十樹, 松野文俊, 稲見昌彦, 大須賀公一 編, “ロボット情報学ハンドブック”, ナノオプトニクス・エナジー, 2010.
- [3] Carlson, C.R. 著, 泰羅雅登・中村克樹 翻訳, “第 2 版 カールソン神経科学テキスト 脳と行動”, 丸善, 2008.
- [4] 坂和正敏・田中雅博, “ニューロコンピューティング入門”, 森北出版, 1997.
- [5] Bishop, C.M., “Neural networks for pattern recognition”, OXFORD University Press, 1995.
- [6] Bishop, C.M., “Pattern recognition and machine learning”, Springer-Verlag New York, 2006.
- [7] 渡辺澄夫, 萩原克幸, 赤穂昭太郎, 木村陽一, 福水健次, 岡田真人, 青柳美輝 著, “学習システムの理論と実現”, 森北出版, 2005.
- [8] Haykin, S.S., “Neural Networks: A Comprehensive Foundation second edition”, Prentice Hall, 1998.
- [9] Duda R.O., Hart, P.E. and Stork, D.G., “Pattern Classification second edition”, WILEY-INTERSCIENCE, 2001.
- [10] 日本ファジィ学会編, “ファジィ・ニューラルシステム”, 日刊工業新聞社, 1995.
- [11] Kasabov, N., “Evolving connectionist systems - Methods and applications in bioinformatics, Brain study and intelligent machines”, Springer, 2003.

- [12] Barto, A.G., Sutton, R.S., Brouwer, P.S., “Associative search network: A reinforcement learning associative memory”, *Biological Cybernetics*, Vol.40, pp.201-211, 1981.
- [13] Sutton, R.S., Barto, A.G., “Reinforcement learning”, MIT Press, 1998.
- [14] Yamakawa, T., Horio, K., Japanese Patent, NO.4267726 (Filing Date: October 15), 1998.
- [15] Yamakawa, T., Horio, K., “New Design Method of Fuzzy Logic Controller Using Self-Organizing Relationship”, *Proceedings of IIZUKA '98*, Iizuka, Japan, October 16-20, pp.155-158 1998.
- [16] Yamakawa, T. and Horio, K., “Self-Organizing Relationship (SOR) Network”, *IEICE Trans. on Fundamentals*, Vol.E82-A, No.8, pp.1674-1678, 1999.
- [17] 堀尾恵一, “評価値を利用して入出力関係を自己組織的に抽出するネットワークとその適応化に関する研究”, 九州工業大学 博士論文, 2001.
- [18] Kohonen, T., “Self-organized formation of topologically correct feature maps”, *Biological Cybernetics*, Vol.43, pp.59-69, 1982.
- [19] Kohonen, T., “Self-organizing maps (3rd ed.)”, Springer, 2001.
- [20] Koga, T., Horio, K., Yamakawa, T., “The Self-Organizing Relationship (SOR) network employing fuzzy inference based heuristic evaluation”, *Neural Networks*, Vol.19, pp.799-811, 2006.
- [21] 古賀崇了, “自己組織化学習による因果関係の獲得とその応用に関する研究”, 九州工業大学 博士論文, 2006.
- [22] Horio, K., Yamakawa, T., “Adaptive self-organizing relationship network and its application to adaptive control”, *Proc. of the 6th Int. Conf. on Soft-Computing and Information/Intelligent Systems (IIZUKA 2000)*, pp.299-304, 2000.
- [23] Rucci, M., Edelman, G.M., Wray, J., “Adaptation of orienting behavior: From the barn owl

- to a robotic system”, IEEE Trans. on Robotics and Automation, Vol.15, No.1, pp.96-110, 1999.
- [24] Irie, R.E., “Robust Sound Localization: An Application of an Auditory Perception System for a Humanoid Robot”, Master’s thesis, MIT, Dept. of Electrical Engineering and Computer Science, 1995.
- [25] Asai, Y., Nakashima, H., Yamamura, T., Huang, J., Ohnishi, N., “Acquiring the ability of object localization by vision and audition through motion”, Electronics and Communications in Japan, Part 2, Vol.85, No.4, pp.58-67, 2002.
- [26] 中島弘道, 大西昇, “運動と感覚のインタラクションによる音源定位の学習 - 非線形な制御対象の場合 - ”, 信学技報, NC98-33, 1998.
- [27] 中島弘道, 松本哲也, 大西昇, “非線形な制御対象での音源定位能力の自己組織的獲得”, 電気学会論文誌 C, Vol.122, NO.4, pp.669-676, 2002.
- [28] Hecht-Nielsen, R., “Counter-propagation network”, Proc. of IEEE First Int. Conf. on Neural Networks, Vol.2, pp.19-32, 1987.
- [29] Horio, K., Haraguchi, T. and Yamakawa, T., “An Intuitive Contrast Enhancement of an Image Data Employing the Self-Organizing Relationship (SOR)”, Proc. of Int. Joint Conf. on Neural Networks (IJCNN’99), pp.10-16, 1999.
- [30] Deng, D., Kasabov, N., “ESOM: An algorithm to evolve self-organizing maps from on-line data streams”, Proc. of IJCNN2000, Como, Italy, Vol. VI, pp.3-8, 2000.
- [31] Deng, D., Kasabov, N., “On-line pattern analysis by evolving self-organizing maps”, Neurocomputation, Vol.51, pp.87-103, 2003.
- [32] 前田幹夫, 村上讓司, 村上周太, “Fuzzy logic コントローラの設計”, 九州工業大学研究報告(工学), No.49, 1984.

- [33] Mizumoto, M., “Fuzzy controls by product-sum-gravity method”, *Advancement of Fuzzy Theory and Systems in China and Japan*, International Academic Publishers, pp.c1.1-c.1.4., 1990.
- [34] 甘利俊一, 外山敬介編, “脳科学大辞典”, 朝倉書店, 2000.
- [35] Fritzke, B., “Growing Cell Structures - a self-organizing network for unsupervised and supervised learning”, *Neural Networks*, Vol.7, No.9, pp.1441-1460, 1994.
- [36] Bruske, J., Sommer, G., “Dynamic cell structure learns perfectly topology preserving map”, *Neural Computation*, Vol.7, pp.845-865, 1995.
- [37] Martinetz, T.M., Berkovich, S.G., Schulten, K.J., ““Neural-gas” network for vector quantization and its application to time-series prediction”, *IEEE Trans on Neural Networks*, Vol.4, No.4, pp.558-569, 1993.
- [38] Martinetz, T.M., Schulten, K.J., “Topological representing networks”, *Neural Networks*, Vol.7, No.3, pp.507-522, 1994.
- [39] Lee, C.C., “Fuzzy logic in control systems: fuzzy logic controller - Part I and Part II”, *IEEE Trans. Systems, Man and Cybernetics*, Vol.20, No.2, pp.404-435, 1990.
- [40] Wang, X-S., Cheng, Y-H., Yi, J-Q., “A fuzzy actor-critic reinforcement learning network”, *Information Sciences*, Vol.177, pp.3764-3781, 2007.
- [41] Leng, G., Prasad, G., McGinnity, T.M., “An on-line algorithm for creating self-organizing fuzzy neural networks”, *Neural Networks*, Vol.17, pp. 1477-1493, 2004.
- [42] Kondo, T., Ito, K., “A reinforcement learning with evolutionary state recruitment strategy for autonomous mobile robots control”, *Robotics and Autonomous Systems*, Vol.46, pp.111-124, 2004.
- [43] Pfeifer, R., Bongard, J., “How the body shapes the way we think - A new view of intelligence”, MIT Press Cambridge, 2007.

- [44] Blauert, J., "Spatial hearing - The psychophysics of Human Sound Localization (Revised ed.)", MIT Press, 1997.
- [45] 牧野達郎編, "知覚の可塑性と行動適応", プレーン出版, 1998.
- [46] Jeffress, L.A., "A place theory of sound localization", J. Comp. Physiol. Psychol., Vol.41, pp.35-39, 1948.