経験に基づく学習型ファジィ推論モデル
Fuzzy inference model for learning based on experiences

○ 郷古 学　　菅谷 至寛　　阿曽 弘具
○ Manabu GOUKO, Yoshihiro SUGAYA, Hirotomo ASO
東北大学大学院工学研究科　電気・通信工学専攻
Dept. of Electrical and Communication Eng., Graduate School of Eng., Tohoku University

***Abstract*: Fuzzy inference models can conduct advanced inference using knowledge which is easily understood by humans. In this paper, we propose a fuzzy inference model for learning based on experiences. The proposed model executes learning with input/output (I/O) data of the model and their evaluations obtained by trial-and-error of a task. Such a learning is executed after each end of a trial. Hence, it is expected that the achievement rate increases with repetition of trials, and that the model adapts to change of environment. We confirm performance of the model using a mobile robot navigation task simulation. It confirmed that the model acqires the knowledge that was effective in the task achievement.**

***Key Words*: Fuzzy inference model, Learning, Experience, Mobile robot navigation task**

1. は　じ　め　に

ロボット等に代表される知識処理システムの実現を目的とした様々な研究がなされている．このような知識処理システムでは，多様な環境に対して高い適応能力が要求される．組み込み型の知識により情報処理を行うシステムを構築する場合，設計者はシステムが適応する環境に加え，システム自身の物理的制約を充分に考慮した上で，知識を作成している．しかし，環境及びシステムの複雑化に従い，予想外の事態が生じ，組み込み型知識には自ずと限界がある．そこで近年，学習によりシステム自身に知識を獲得させる様々な研究が行われている．

学習による知識獲得において，システムが扱う知識が人間にとって親和性が高いということは，次のようなメリットをもたらすと考えられる．例えば，環境との相互作用によりシステムが学習した知識が，設計者である人間にとって容易に理解可能であるならば，その情報をシステム設計にフィードバックすることにより，より適応性の高いシステムの構築が期待できる．逆に，システムに対して，少しでも人間の持つ知識を与えることが可能であれば，システムはその情報を事前知識 (バイアス) として効率的な学習が実現できる[(1)]．このように，環境-システム間の相互作用に加えて人間をも含めたインタラクションを実現することで，多くの工学的なメリットがあると考えられる．

人間との親和性が高い知識表現法としてファジィルールによる知識表現がある．ファジィルールは自然言語により表現される知識を if-then 形式で記述したものであり，知識の表現・理解が容易である．ファジィルールを用いて推論を行うファジィ推論モデルは，これまでに制御分野をはじめとして様々な分野で応用されている．そこで本研究では，学習型のファジィ推論モデルを扱う．

これまでに提案された学習型のファジィ推論モデルに関する研究は，学習法によりいくつかに分類することが出来る．教師あり学習[(2) (3)]，遺伝的アルゴリズムなど進化的計算手法による学習[(4) (5)]，強化学習[(6)〜(9)]などがある．教師あり学習に関しては，学習に必要なデータとして，入力情報 (状態) と，それに対する目標出力 (教師信号) を設定する必要があり，環境が変化するような場合には，常に新たな学習データを設定し，再学習する必要がある．そのため，環境の変化への追従性は低いと考えられる．また，例えば将棋やチェスのように，タスク達成 (勝負が決まる) までに遷移する各状態に対して，明示的に教師を与えることが困難な問題には適応することが難しい．

このような問題に対し，進化的計算手法による学習は，各個体ごとにタスク試行結果に対する評価値を与え，進化を繰り返すことで，タスク達成に優れたモデルを獲得することが可能である．しかし，一般に多くの計算時間を要するため，やはり環境の変化への追従性は低いと考えられる．

強化学習はエージェント (学習者) が与えられた問題環境下で試行錯誤を繰り返し，よりよい解を自律的に獲得する方法である．設計者はゴールの状態に対して報酬を設定するだけで学習が可能である．文献 (6)(7) では，強化学習の一つである Q-Learning 法に基づく学習を行っている．しかし，Q-Learning 法のような，環境を幅広く探索し最適解を発見する方法 (環境同定型) は膨大な計算量を必要とすることが問題とされており，環境の変化に対する追従性が低いと考えられる．文献 (8)(9) では，タスクを行った際に得られるデータ (経験) を用いて学習を行う，経験強化型の学習法を用いている．経験強化型の学習法は，最適解を発見する保証はないが，環境同定型に比べ学習速度が速いという特徴を持つ．高濱らは離散値制御のための経験強化型のファジィ制御規則の学習法を提案している[(8)]．また，堀内らは学習に用いる価値関数をファジィ推論モデルで表現し，連続的な状態-行動空間における効率的な学習を実現した．しかし，学習により得られた知識については議論されておらず，また，いずれの研究においても，問題として扱っているのは，静的な環境における学習であり，環境の変化については扱っていない．

以上の考察から，本研究では環境との相互作用により経験強化型の学習を行うファジィ推論モデルを提案し，学習により獲得された知識について確認する．提案モデルは

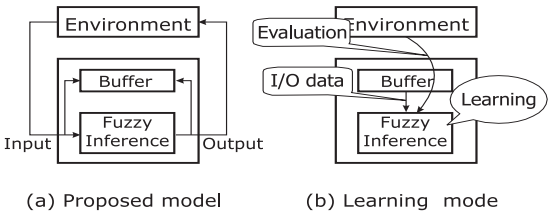

Fig. 1 (a) Proposed model. (b) Learning mode.

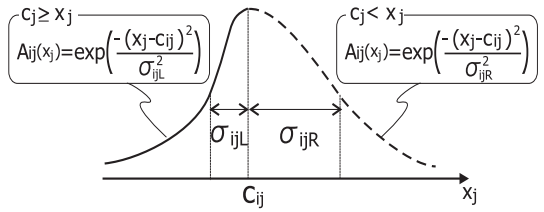

Fig. 2 Membership function.

自らの知識を用いて実際にタスクを実行し，経験強化型の学習を行う．従来の経験強化型学習法の多くが，成功もしくは失敗のいずれかの経験に対してのみ学習を行うのに対し，提案モデルは両方の経験を基に学習することが可能であり，より効率的な学習が期待できる．学習はタスク試行毎に逐次的に行われるため，モデルのタスク達成率はタスク試行毎に上昇し，環境が変化するような場合でも，変化に追従することが期待される．モデルが扱う知識はファジィルールで表現されるため可読性が高く，人間にとっても容易に理解可能である．本研究では，提案モデルを用いたロボットによるナビゲーションタスクのシミュレーションを行い，モデルの学習性能を確認すると共に，学習により得られたファジィルールのメンバーシップ関数の配置や形状の変化から，環境の変化に応じて，どのような知識が獲得されるのかを調査する．以下 2 で提案モデルの説明，3 で実験および考察，4 でまとめを述べる．

2. 提 案 モ デ ル

本章では，提案モデルの構造と学習法について述べる．提案モデルの構造を図 1(a) に示す．モデルは入力情報を基にファジィ推論 (fuzzy inference) を行う推論機構と，データを保存するバッファ(buffer) から構成される．提案モデルは，自身の知識を基にタスクの達成を試行錯誤的に試みる．タスク開始から終了までを，タスクの 1 試行とし，タスク試行中の推論機構への入出力データはバッファに蓄えられる．モデルは 1 試行あたり複数回 (1 回以上)，推論を行うと考える．試行終了後，モデルは試行結果に対して与えられる評価値と，バッファに蓄えられている複数の入出力データ対を基に学習を行う (学習モード．図 1(b))．モデルは，学習により自らの知識を変化させて，環境に適した知識を獲得する．学習はタスクの成功・失敗に関わらず，タスク試行の度に逐次的に行われる．そのため，環境の変化に対しても，試行錯誤を繰り返しながら学習を行うことで，追従することが期待される．

2.1 ファジィ推論機構

推論機構では if-then 形式のルールを基に推論を行う．i 番目のファジィルール $Rule^i$ は以下のように定義される．

$$Rule^i: \text{if } x_1 \text{ is } A_{i1} \cdots \text{ and } x_j \text{ is } A_{ij} \text{ and} \cdots \cdots \text{and } x_m \text{ is } A_{im} \text{ then } y = b_i \quad (i = 1, 2, \cdots, n) \quad (1)$$

ここで，x_j は入力変数，A_{ij} はファジィ集合，y は出力変数，b_i は後件部定数である．モデルへは設計者が n 個のルールを事前知識として与える．推論は次に示す簡略推論法を用いる．

$$\mu_i = \prod_{j=1}^{m} A_{ij}(x_j) \quad (i = 1, 2, \cdots, n) \quad (2)$$

$$y^* = \sum_{i=1}^{n} \mu_i b_i / \sum_{i=1}^{n} \mu_i \quad (3)$$

ここで，μ_i は入力に対する $Rule^i$ の前件部適合度，y^* は推論結果 (出力) である．$A_{ij}()$ は，以下の式で定義されるファジィ集合 A_{ij} のメンバーシップ関数である．

$$A_{ij}(x_j) = \exp\left(-(x_j - c_{ij})^2 / \sigma_{ijk}^2\right) \quad (i = 1, 2, \cdots, n, j = 1, 2, \cdots, m) \quad (4)$$

ここで，k は分散を指定するパラメータであり，次式で定める．

$$k = \begin{cases} L & \text{if } c_{ij} \geq x_j \\ R & \text{otherwise} \end{cases} \quad (5)$$

提案モデルでは図 2 に示すような，中心値 c_{ij} と二つの分散 $\sigma_{ijL}, \sigma_{ijR}$ で定義される非対称な釣鐘型メンバーシップ関数を用いる．

バッファには，タスク試行中の推論機構への入力 $(x_1, x_2, \cdots, x_m)$ とそれに対する出力 (y^*) が蓄えられる．そのため，1 試行あたり，推論を行った回数分の入出力データ対が得られる．バッファのデータはタスク試行直後に行われる学習に用いた後にリセット (消去) される．

2.2 学習モード

提案モデルの学習法について説明する．提案モデルはタスク試行終了毎に逐次的に学習を行う (学習モード)．学習モードでは，バッファに蓄えられている，直前のタスク試行時に得られた入出力データ対 (以下，学習データと呼ぶ) と，評価値を用いて各ルールのメンバーシップ関数の中心値及び分散，後件部定数の更新を行う．

評価値 E $(-1 \leq E \leq 1)$ はタスク試行終了後にモデルに対して外部から与えられるものであり，結果が良好なほど大きい値を持つとする．学習モードでは，評価値が高い $(E \geq 0)$ 場合は，学習データの入出力関係を近似する方向に学習が行われ，評価値が低い $(E < 0)$ 場合には，逆の方向に学習が行われる．前者を成功学習，後者を失敗学習と呼ぶ．このように，提案モデルでは評価値が正負どちらの場合からでも，学習を進めることが可能である．

評価値と学習データに基づく学習法の問題点について考える．一般にタスクの達成方法 (成功パターン) は複数あり，各成功パターンから得られる学習データの入出力関係も，それぞれ異なることが予想される．モデルが持つルールでは，得られるすべての学習データの入出力関係を充分に表すことが出来ない場合，ある成功パターンに基づく成功学習を行っているルールが，異なる成功パターンに基づく学習による影響を受け，学習がうまく進まなくなる場合や，学習した知識が不安定になることが考えられる．また，タ

スクを失敗するケースとして，試行の途中までは，望ましい推論が出来ていたにも関わらず，失敗に至るようなケースが考えられる．この場合，学習データの中には望ましい推論を行った際のデータ対も含まれる．しかし，これらのデータに対して，他のデータと同様に失敗学習を行うことにより，望ましい入出力関係を表現していた知識が不安定になってしまうという問題が考えられる．

一般に，成功パターンによる学習データのばらつきによる知識の不安定化に関しては，学習係数を小さくすることで対処することが出来るが，学習速度の低下を招くことが考えられ，環境の変化への追従性の低下につながる．タスクが失敗した場合の学習による知識の不安定化に関して，畝見[(10)]は，失敗するまで試行を行い，得られた入出力データに対し，試行が失敗した時刻から遠い過去のデータほど，望ましい入出力データとみなして，それらのデータの入出力関係を学習するという方法を提案している．また，高濱ら[(8)]も同様にデータを獲得した時間情報を用いる学習法を提案している．これらの方法は，失敗に至った原因は失敗に対して近い過去における推論の影響によるものである，という仮定に基づく方法である．しかし，タスクによっては失敗の直前ではなく，より過去に行われた推論の影響で失敗に至る場合も考えられる．そのため，このような時間情報を用いる従来の方法はタスクに特化した方法と考えられる．

学習により知識が不安定になるという問題について，ファジィルールが表す入出力関係の変化という観点から考える．各ルールは，前件部の各メンバーシップ関数により表される入力空間の領域 (入力領域) から，後件部定数が示す出力空間の点 (出力点) への写像を記述している．学習によるルールの更新は，入力領域と出力点が各空間内で変化 (移動・変形) することに対応する．このとき，出力点の変化が入力領域の変化に比べて充分小さいと考え，入力領域の変化に着目する (このような，出力点と入力領域の変化の関係は，学習係数の設定により実現できる)．成功学習により，入力領域は学習データの分布を近似するように変化する．成功学習における知識の不安定化は，各成功パターンにより得られる学習データのばらつきに対し，入力領域が敏感に反応し，変化してしまうことで起こると考えられる．また，失敗学習では入力領域が学習データから離れる方向に学習がなされる．失敗学習における知識の不安定化は，学習データ中に，成功学習で近似すべきデータが含まれていることにより起こると考えられる．

以上の考察から，成功学習においては，入力領域の中心に近い学習データに対しては，学習に与える影響を大きくし，逆に入力領域から遠いデータに関しては影響を小さくすることで，知識の不安定化を防ぐことが期待できる．また，失敗学習時においては，入力領域の中心に近いデータに対しては学習に与える影響を小さくし，遠いデータに関しては，逆に，影響を大きくすることで同様の効果が期待できる．本研究では，このように，学習データと各ルールの前件部が定義する領域の中心との距離を考慮した学習則を提案する．以下に学習モードにおける学習手順を示す．

(1) バッファから学習データ $(\boldsymbol{x}^l, y^l) = (x_1^l, x_2^l, \cdots, x_m^l, y^l)$ を一つ選択する．

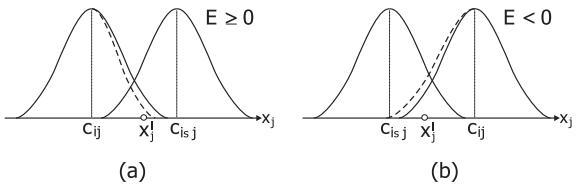

Fig. 3　Update on σ_{ijk} ((a)$E \geq 0$, (b)$E < 0$).

(2) 式 (2) により，$\boldsymbol{x}^l$ とルール i との適合度 μ_i^l を求め，μ_i^l が最大となるルール i_s を求める．

(3) 以下の式により，ルール i_s の中心値と後件部定数の更新を行う．

$$c_{i_sj}^{new} = c_{i_sj}^{old} + \alpha E A_{i_sj}(x_j^l)(x_j^l - c_{i_sj}^{old}) \quad \text{if } E \geq 0 \quad (j = 1, 2, \cdots, m) \tag{6}$$

$$b_{i_s}^{new} = b_{i_s}^{old} + \beta \mu_{i_s}^l E(y^l - b_{i_s}^{old}) \quad \text{if } E \geq 0 \tag{7}$$

ここで，α, β はある定数で，学習係数である．提案する学習法では，中心値と後件部定数の更新は $E \geq 0$ の時にのみ行われる．式 (6) において，$A_{i_sj}(x_j^l)$ は学習データ x_j^l に対するルール i_s のメンバーシップ関数の値である．これは，学習データと入力領域の中心との距離を学習に反映するためのものである．これにより，学習データ x_j^l がメンバーシップ関数の中心に近いほど，学習に与える影響が大きくなる．

(4) 以下に示す式によりメンバーシップ関数の分散の更新を行う．

$$\sigma_{ijp}^{new} = \begin{cases} \sigma_{ijp}^{old} - \gamma_1 E A_{i_sj}(x_j^l)\mu_i^l & \text{if } E \geq 0 \\ \sigma_{ijp}^{old} - \gamma_2 E(1 - A_{i_sj}(x_j^l))(1 - A_{ij}(x_j^l)) & \text{if } E < 0 \end{cases} \quad (i = 1, 2, \ldots, n,\ j = 1, 2, \ldots, m) \tag{8}$$

ここで，γ_1, γ_2 はある定数で，学習係数である．また，パラメータ p は，メンバーシップ関数の持つ 2 つの分散のどちらを更新するかを決めるもので，次式で定める．

$$p = \begin{cases} L & \text{if } c_{i_sj} < x_j^l \leq c_{ij} \ \&\ i \neq i_s \\ & \quad \text{または } x_j^l \leq c_{ij} \leq c_{i_sj} \ \&\ i \neq i_s \\ R & \text{if } c_{ij} \leq x_j^l \leq c_{i_sj} \ \&\ i \neq i_s \\ & \quad \text{または } c_{i_sj} \leq c_{ij} \leq x_j^l \ \&\ i \neq i_s \\ \phi & \text{otherwise} \end{cases} \tag{9}$$

$p = \phi$ の場合は，分散の更新は行わない．分散の更新の様子を図 3 に示す．図中の破線は，分散の更新により変化したメンバーシップ関数を表す．評価値が正の場合 (図 3(a)) は，ルール i_s 以外の各ルールの持つメンバーシップ関数の分散を，それぞれのルールと学習データの適合度 μ_i^l に応じて小さくする．これにより，推論全体としては，ルール i_s の適合度を上昇させるように学習がなされる．式 (8) において，$E \geq 0$ の場合の更新式に含まれる $A_{i_sj}(x_j^l)$ は，学習データと入力領域の中心との距離を学習に反映するためのものである．また，評価値が負の場合 (図 3(b)) は，ルール i_s 以外の各ルールの持つメンバーシップ関数の分散を大きくする．

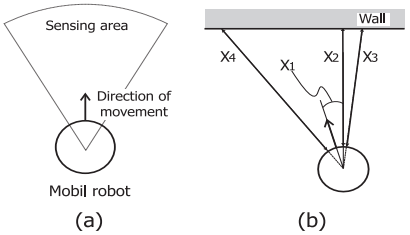

Fig. 4　(a) Mobile robot. (b) Input data.

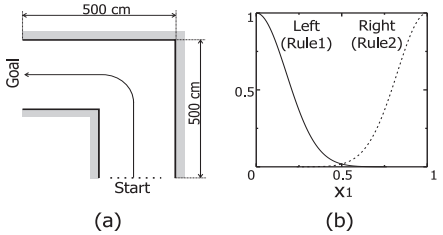

Fig. 5　(a) Course of robot navigation. (b) Membership function.

これにより，推論全体としてはルール i_s の適合度を減少させるように学習がなされる．式 (8) において，$E < 0$ の場合の更新則に含まれる $(1-A_{i_sj}(x_j^l))(1-A_{ij}(x_j^l))$ により，ルール i_s 及び分散が更新されるルール i のそれぞれの入力領域の中心に対し，学習データが近い場合には，学習の影響が小さくなる．これにより，失敗学習における知識の不安定化を防ぐ．

以上の操作が完了したならば，学習に用いたデータ対 $(\boldsymbol{x}^l, y^l)$ をバッファから消去し，手順 (1) から (4) までを繰り返す．バッファにデータがなくなった時点で学習モードが終了する．

3. 実験及び考察

提案モデルの学習能力の検証及び学習により獲得した知識について調査するため，計算機実験として，提案モデルを用いた移動ロボットによるナビゲーションタスク[(5)]を行った．

ロボットナビゲーションタスクは，移動ロボットが壁に衝突することなく，スタート地点からゴールまで移動するタスクである．実験で用いる移動ロボットを上から見た図を図 4(a) に示す．移動ロボットは直径 20cm の円形であり，進行方向に対し左右 30 度，200cm 先までの物体を検出できるセンサがついている．ロボットはセンシングエリアに壁が存在しない場合は推論を行わずに直進し，壁が存在する場合は推論により移動方向を決定し，移動する．ロボットへの入力は，図 4(b) に示す最短の壁面と進行方向の相対角度 $x_1(-30 \leq x_1 \leq 30[\text{degree}])$(角度は，進行方向に対し時計回りの方向を正としている)，壁との最短の距離 $x_2(0 \leq x_2 \leq 200[\text{cm}])$ 及びセンシングエリア右端の壁との距離 $x_3(0 \leq x_3 \leq 200[\text{cm}])$，左端の壁との距離 $x_4(0 \leq x_4 \leq 200[\text{cm}])$ の 4 つとした．推論によりロボットのステアリング角度 (y^*) が出力される．出力が正の場合は右へ，負の場合は左へステアリングを切ることを意味している．最大操舵角は進行方向に対して $\pm 10[\text{degree}]$ である．推論機構においては，各入力変数及び出力は値域 [0,1] に収まるように正規化された値を用いる．

ロボットは図 5(a) に示す，幅 250cm の左曲がりの通路をスタートからゴールへと向かうタスク (以下，学習タスクと呼ぶ) を繰り返し行う[(5)]．スタート位置は図 5(a) の点線で示す，両側の壁から 20cm 以上離れた，ランダムな位置を選択する．また，スタート時の向きに関しても，一定範囲内でランダムな角度を与える．移動ロボットはセンサにより観測される環境からの情報を基に移動を行う．この観測から移動までを 1 ステップとする．1 ステップあたりロボットの移動距離は Lcm で一定とする (L を速度と呼ぶ)．スタート後にロボットが壁に衝突するか，衝突せずにゴールに到達するまでを，学習タスクの 1 試行 (trial) とする．評価値 E はゴールへ到達した場合は 1，壁に衝突した場合は -1 とした．モデルはタスク試行後に学習を行い，学習終了後，再びスタート地点から試行を開始する．

実験では，学習完了後に学習モードを行わずに同様のタスク (テストタスク) を行い，達成率を調査した．つまり，実験はロボットが学習タスクを試行し，学習完了後に，テストタスクを 100 回 (スタート時の位置と向きが異なる) 行い達成率を求めるという操作を繰り返す．学習が有効に働いていれば，学習タスクの試行回数が増えるに従い，テストタスクの達成率が上昇することが期待される．

3.1 環境が変化しない場合の学習能力と知識の獲得

本実験では，環境が変化しない場合における，提案モデルの学習能力及び獲得した知識についての調査を行う．モデルが獲得した知識の調査は，学習により得られた各ルールのメンバーシップ関数の形状を確認することで行う．提案する学習法は，成功した経験だけでなく，失敗した経験からも学習が可能である．そこで，失敗したデータからの学習の有効性を確認するため，失敗学習によるタスク達成率の変化についても調べた．実験で使用したパラメータは $\alpha = 0.004, \beta = 0.0001, \gamma_1 = 110, \gamma_2 = 0.004,$ である．ロボットの速度は $L = 10$ で一定とした．なお，本実験では入力変数 x_1, x_2 のみを用いた．

モデルへの事前知識として，$x_1 < 0.5$ のときは，壁がロボットに対して左側にあるとして，右に回避するルール (ルール 1)，$x_1 > 0.5$ のときは，壁が右側にあるとして左に回避するルール (ルール 2) という二つの単純なルールを設定した (正規化により，壁がロボットの正面にある場合，$x_1 = 0.5$ となる)．各ルールの後件部定数は $b_1 = 1, b_2 = 0$ で，それぞれステアリング角度が $10, -10[\text{degree}]$ に対応する．図 5(b) に事前知識として与えた各ルールの x_1 に対応するメンバーシップ関数の形状を示す．なお，x_2 に関してはどちらのルールも同じ形状のメンバーシップ関数を設定した．事前知識により，図 4(b) の場合は，$x_1 > 0.5$ のためルール 2 の適合度の方が高くなり，結果として，ロボットは左へステアリングを切る．

学習タスクの試行回数と，テストタスクの達成率の変化を図 6(a) に示す．実線は提案した，入力領域と学習データの距離を考慮した学習法による結果であり，破線は考慮しないで学習を行った場合の結果である．考慮しない場合の学習則は，式 (6)，(8) において，$E \geq 0$ の場合は $A_{i_sj}(x_j^l)$ を，$E < 0$ の場合は $(1 - A_{i_sj}(x_j^l))(1 - A_{ij}(x_j^l))$ をそれぞれ 1 としたものを用いた．どちらの結果も 10 回の実験結果の平均であり，分散をエラーバーで表している．達成率を見ると，タスク開始直後は，どちらも達成率が上昇しているが，入力領域と学習データの距離を考慮しない学習

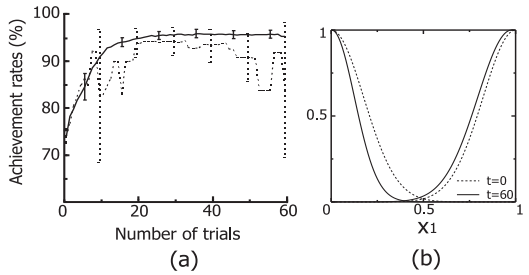

Fig. 6　(a)Achievement rates. (b)Membership function (x_1).

法は，ときおり達成率が急激に下がる不安定な挙動を示している．一方，提案した学習法を用いた場合は，学習前は 74%であった達成率が 30 試行終了後で 95.5%まで安定して上昇し，その後も安定して高い達成率が維持される様子が分かる．入力領域と学習データの距離を考慮しない学習法を用いた場合は，達成率が高い状態においても，急激な達成率の低下が見られ，やはり不安定な状態であるといえる．また，分散からも提案する学習法の方がばらつきも小さく安定していることが分かる．

提案学習法により，モデルが獲得した知識に関して調査するため，学習タスク 60 試行目における各ルールのメンバーシップ関数について確認した．特に，学習による変化が大きいメンバーシップ関数として，各ルールの x_1 に対応するメンバーシップ関数の変化に着目した (図 6(b))．これを見ると，学習前に事前知識を与えた段階 ($t = 0$) では，各メンバーシップ関数の交点は $x_1 = 0.5$ であったが，学習により関数の交点は，$x_1 = 0.39$ に変化している ($t = 60$)．この変化により，ロボットは最短の壁が，正面もしくは多少左にあったとしても，ルール 2 の適合度の方が大きくなり，左にステアリングを切るようになる．つまり，ロボットは左に曲がるタスクを繰り返して学習を行うことにより，より左へステアリングを切りやすい知識を獲得したことを意味している．

提案する学習法は，成功だけではなく，失敗した際のデータからも学習を行うことで効率的な学習の実現を目指す．そこで，同実験における全試行数 600 回のうち，失敗学習を行った全 50 回の，学習後の達成率の変化 (達成率増加，減少，変化なし) の割合を確認した．失敗学習を行った場合，達成率の増加，減少，変化なしの割合は 48%，4%，48%であり，約半分で達成率の上昇が見られた．また，失敗学習 1 回につき，達成率は平均で 1.76%上昇した．これより，提案モデルでは，成功のみならず，失敗したデータからの学習も有効に働いていることが分かる．

3.2 環境が変化する場合の学習能力と知識の獲得　　本実験では，環境が変化する場合における，学習能力及び獲得した知識の調査を行う．実験内容は 3.1 と同様であるが，30 試行を境にロボットの速度を変更した．30 試行目までは $L = 10$ で一定とし，31 試行目以降は $L = 25$ とした．速度は学習・テストタスク共に変化させる．速度の変化により，一時的な達成率の低下が予想されるが，速度変化後の環境で学習を繰り返すことにより，変化に追従して達成率が徐々に増加していくことが期待される．

本実験では入力変数として，x_1, x_2 だけを用いた場合 (2 入力モデル) と，4 つの入力変数すべてを用いた場合 (4 入力モデル) で実験を行った．事前知識として，2 入力モデルへは 3.1 と同じルールを与え，4 入力モデルに関しては，各ルールの x_3, x_4 のメンバーシップ関数をすべて，中心値 0.5，分散を $\sigma_{ijL} = \sigma_{ijR} = 0.5$ とした．このような入力変数の追加は，センサの追加に対応している．本実験では追加分のセンサの情報に関して，各ルールに共通のメンバーシップ関数を与えておき，学習により，自律的にこれら追加分のセンサ情報を用いた知識が獲得されるかどうかを確認する．なお，実験に用いたパラメータは，すべて 3.1 と同じである．

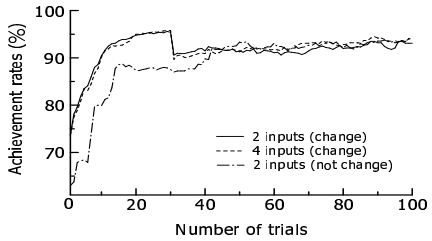

Fig. 7　Achievement rates.

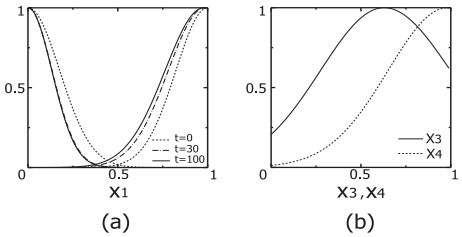

Fig. 8　(a) Membership function (x_1). (b) Membership function ($Rule^2, x_3, x_4$).

図 7 は，環境が変化する場合における 2 入力モデル (実線) 及び 4 入力モデル (破線) と，速度を $L = 25$ で一定として学習させた 2 入力モデル (一点鎖線) のテストタスクの達成率の変化である．いずれの結果も 10 回の実験結果の平均である．環境が変化する場合の，2 入力モデル及び 4 入力モデルの達成率を見ると，30 試行目までに 2 入力で 95.4%，4 入力で 95.8%までタスク達成率が上昇している．速度変化直後の 31 試行目での達成率は 2 入力で 90.6%，4 入力で 89.7%まで低下している．100 試行目での達成率は 2 入力で 93.1%，4 入力で 94.1%であった．環境変化後における達成率の増加は，速度が変化した環境下での学習の効果によるものである．速度変化により達成率が一時的に低下しているが，事前知識のみで $L = 25$ として同タスクを行った場合の達成率を調べたところ，2 入力，4 入力共に 62.9%であった．そのため，30 試行目までに獲得した知識は，事前知識に比べて，速度の増加に対し，ある程度対応できていると考えられる．また，速度変化後の達成率の増加速度 (学習速度) が速度変化前に比べて小さくなっているが，速度を $L = 25$ で一定として学習を行った 2 入力モデルの学習速度と同程度であることから，学習速度の変化は，環境の変化によるものではなく，そもそも，ロボットの速度が高い場合にはタスクの難易度も高いことが影響していると考えられる．

次に，モデルの持つ知識の変化を確認するために，2 入力モデルの各ルールの持つ x_1 に対応するメンバーシップ関数の形状について確認した (図 8(a))．図は，学習前 ($t = 0$)，30，100 試行時 ($t = 30, 100$) におけるものである．これ

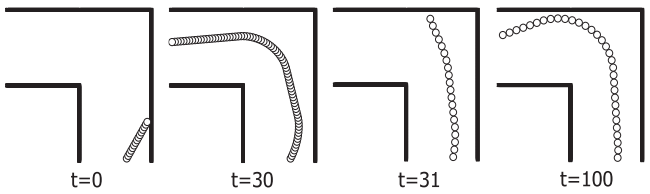

Fig. 9 Trajectory.

を見ると，速度の変化により，両ルールのメンバーシップ関数の交点の位置が移動している．各時点における交点の x_1 の値は，0.5，0.39，0.37 であった．これは，速度の増加に伴い，曲がりきれずに壁へ衝突することを回避するために，より左へステアリングを切りやすい知識を獲得したことを意味する．

4 入力モデルの持つ入力変数 x_3, x_4 に対応するメンバーシップ関数についても着目した．図 8(b) は 100 試行時におけるルール 2 の x_3, x_4 の持つメンバーシップ関数である．ルール 2 は左にステアリングを切るルールであり，図の各関数の関係から，$x_3 < x_4$ のときに左に向きを変えるという知識が獲得されている．$x_3 < x_4$ という状態は図 4(b) のように，ロボットに対して右側に壁がある場合に観測される状態である．つまり，ロボットは学習により，左へステアリングを切る場合の，妥当な入力変数 x_3, x_4 のメンバーシップ関数を獲得していると考えられる．

さらに，入力変数 x_3, x_4 による知識が，学習により各環境下で有効な知識へと変化する様子を確認する．確認は，同実験で学習を行った 4 入力モデルに対し，入力変数 x_3, x_4 のみを推論に用いてテストタスクを行い，ロボットの軌道を調査することで行う．図 9 は，学習前 $(t = 0)$，30，31，100 試行 $(t = 30, 31, 100)$ 後の，テストタスクのロボットの軌道であり，すべて同じ初期位置と方向からスタートしている．図中の○ はロボットを表す．これを見ると，学習前の段階では，知識が学習されておらず，壁に向かって直進し衝突しているが，$t = 30$ を見ると，学習により知識を獲得したことで，ゴールへと到達している様子が分かる．$t = 31$ では速度の増加に伴い，それまでに獲得した知識では曲がりきれずに壁に衝突しているが，その後，速度増加後の環境における学習により，ゴールへと到達可能な知識が獲得されたことが分かる $(t = 100)$．この結果より，ロボットは環境の変化に応じて，追加された入力変数 x_3, x_4 を利用した，タスク達成に有効な知識を獲得していることが分かる．

実験において，ロボットが試行錯誤を繰り返して獲得した知識は，自らの物理的制約を考慮したものであると考えられる．例えば，速度の変化による x_1 のメンバーシップ関数の変化は，ロボットの持つ最大操舵角や，センシング可能な距離という制約により，曲がりきれずに壁に衝突してしまうために，ステアリングを左へ切りやすい知識を獲得したと考えられる．提案モデルにおいて，知識はファジィルールにより記述されているため，学習によりシステムが獲得した知識は容易に理解できる．こういった情報を，システムのパラメータ設定 (この場合，最大操舵角やセンシング距離) 等，設計段階へとフィードバックすることで，より環境への適応性が高いシステムの実現が期待できる．また，新たに追加した入力変数の情報により，どのような知識が得られるのかを確認することが出来る．これにより，その情報 (入力変数) が推論にどういった影響を及ぼしているのかが，容易に理解できる．本実験では，入力変数 x_3, x_4 の情報を用いたどのような知識が獲得されたのかを確認し，さらに，学習によりその知識が，各環境において有効な知識へと変わっていく様子を示した．

4. お わ り に

本研究では，人間との親和性の高い知識を扱うファジィ推論モデルに着目し，学習型のファジィ推論モデルの提案及び学習により獲得した知識の調査を行った．提案モデルは，タスク試行時に得られるデータと，タスク試行結果に対して与えられる評価値に基づいて学習を行う．本研究では，このような学習を行う際に，モデルの持つ知識が不安定になるという問題に対し，学習データとメンバーシップ関数により表現される入力空間の領域との距離を考慮した，新しい学習法を提案し，計算機実験により，学習性能について確認した．さらに，学習により獲得した知識を調査し，環境が変化する場合においても，各環境下で有効な知識を獲得していることを確認した．

今後は，より複雑なタスクに対して適応出来るように，ルールの追加・削除機能について検討する予定である．また評価値の与え方を変化させた場合の学習への影響に関する調査も今後の課題とする．

謝辞 本研究の一部は東北大学 21 世紀 COE プログラム「新世代情報エレクトロニクスシステムの構築」の援助により行われた．

参 考 文 献

(1) 片上大輔, 山田誠二: 対話的進化ロボティクスの観測に基づく教示の設計, システム制御情報学会論文誌, Vol. 16, No. 6, pp. 279–286 (2003).
(2) Er, M., Tan, T. P. and Loh, S. Y.: Control of a mobile robot using generalized dynamic fuzzy neural networks, *Microprocessors and Microsystems*, Vol. 28, pp. 491–498 (2004).
(3) Nishina, T. and Hagiwara, M.: Fuzzy inference neural network, *Neurocomputing*, Vol. 14, pp. 223–239 (1997).
(4) 古橋武, 中岡謙, 森川孝治, 前田宏, 内川嘉樹: ファジィクラシファイアシステムによる知識発見に関する一考察, 日本ファジィ学会誌, Vol. 7, No. 4, pp. 839–848 (1995).
(5) 井谷久博, 古橋武: 自律移動ロボットによる教示情報の理解, 計測自動制御学会論文集, Vol. 38, No. 11, pp. 966–973 (2002).
(6) Jouffe, L.: Fuzzy Inference System Learning by Reinforcement Method, *IEEE Trans. Syst., Man, Cybern., part C*, Vol. 28, No. 3, pp. 338–355 (1998).
(7) 堀内匡, 藤野昭典, 片井修, 椹木哲夫: 連続値入出力を扱うファジィ内挿型 Q-Learning の提案, 計測自動制御学会, Vol. 35, No. 2, pp. 271–279 (1999).
(8) 高濱徹行, 阪井節子, 小倉久和, 中村正郎: 強化学習法による離散値制御のためのファジィ規則の学習, 日本ファジィ学会誌, Vol. 8, No. 1, pp. 115–122 (1996).
(9) 堀内匡, 藤野昭典, 片井修, 椹木哲夫: 経験強化を考慮した Q-Learning の提案とその応用, 計測自動制御学会, Vol. 35, No. 5, pp. 645–653 (1999).
(10) 畝見達夫: 実例に基づく強化学習法による失敗しない制御方法の学習, 人工知能学会誌, Vol. 7, No. 6, pp. 1001–1008 (1992).