進化的recruitment戦略を用いた強化学習による 自律移動ロボットの制御器設計[†]

近 藤 敏 之*・伊 藤 宏 司*

A Reinforcement Learning with Evolutionary State Recruitment Strategy
for Autonomous Mobile Robots Control[†]

Toshiyuki KONDO* and Koji ITO*

In recent robotics fields, much attention has been focused on utilizing reinforcement learning for designing robot controllers, since environments the robots will be situated could be unpredictable for human designers in advance. However there exist some difficulties, one of them is well known as “curse of dimensionality problem” in which as a state space for a learning system (e.g. robot) becomes continuous and high dimensional, the learning process results in time-consuming. Therefore, so as to adopt reinforcement learning for such complicated systems, not only *adaptability* but also *computational efficiencies* should be taken into account. In this paper, an evolutionary state recruitment strategy for an actor-critic reinforcement learning system based on NGnet is proposed, which enables the learning system to divide/rearrange its state space gradually according to the task complexity and the progress of learning. Simulation results and real robot implementations of a peg pushing robot control task show the validity of the method.

Key Words: reinforcement learning, evolutionary state recruitment strategy, peg pushing robot control

1. はじめに

近年、多種・多様な入力情報の統合による認識機能、多自由度で複雑な運動機能を有するロボットが次々と開発され注目を集めている。これらの高度な機能を実現する制御システムの大半は、一般に熟練技術者の豊富な経験・知識に基づいて設計されている。しかしながら、概して専門家の知識は断片的、定性的なものであり、また想定されるロボットの動作環境は、適切に管理された工場などとは異なり多くの不確実性を含むため、その設計過程には依然として多くの試行錯誤を必要とすることが知られている。したがって、できる限り専門家が介入することなく自律的に実環境に適応し、ときには専門家が設計するよりも高度な知能を獲得するロボットの実現に対する期待は大きい。

このような背景から、強化学習¹²⁾や進化ロボティクス³⁾に代表される、環境との相互作用を通して構成論的に制御器を設計するアプローチの有効性が認められつつある。しかし

ながら、一般に強化学習では入出力の次元が増加するにつれて状態の組み合わせ数が指数的に増加するため、実用時間での学習は困難となる。タスクをより次元の低いサブタスクに分割・階層化することで学習を高速化する手法⁹⁾の有効性が報告されているが、本質を失うことなくタスクをうまく分割・階層化するためには、報酬系を適切に設計する必要があるなど、設計者が事前に制御対象の性質を熟知していることが求められる。

また、ロボットに実環境との高い親和性を持たせることを考慮すれば、状態・行動空間は連続系で表現されることが望ましいが、特に実ロボットに実装することを意識した強化学習研究の多くは、計算量的な制約から離散状態・行動を対象として適用されてきた。これに対し、多層パーセプトロン (Multi-Layered Perceptron) や動径基底関数 (Radial Basis Function, RBF) ネットワークなどの関数近似器を用いて制御器を実装することで連続状態を扱えるように拡張した強化学習法の有用性が報告されている。特に、関数近似器として正規化ガウシアンネットワーク (Normalized Gaussian network, NGnet)⁸⁾を用いた場合、その基底関数の局所性のために高速な学習が期待でき、かつ正規化処理によって基底関数のカバー範囲外についても外挿汎化が行なわれるため、より少ない基底関数を用いて関数近似することができることが知られている^{4), 11), 13)}。しかしながら、学習の精度

[†] SICE システム・情報部門学術講演会 (SICE-SSI2002) で発表 (2002・11)

* 東京工業大学大学院総合理工学研究科
横浜市緑区長津田町 4259

* Tokyo Institute of Technology, 4259 Nagatsuta, Midori-ku, Yokohama

(Received January 15, 2003)

(Revised May 29, 2003)

Fig. 1 Reinforcement learning with evolutionary state recruitment strategy

や収束時間に大きく影響する基底関数のパラメータ（RBFの中心ベクトル，分散共分散行列）および基底の数は，通常，設計者が天下りの与えるか，あるいは学習データの統計量からオフラインで決定される必要がある．強化学習など，データが逐次的に得られるオンライン学習では，事前に学習器の最適な構造を設定することは難しい．これまでに，これらの基底関数パラメータをオンラインで推定するさまざまなモデル選択アルゴリズム^{2), 6), 7), 10), 11), 13)}が提案されているが，逐次的に RBF の共分散行列の固有値・固有ベクトルを求める必要があるなど計算コストが高く，高次元入出力を持つロボットの学習に適用することは困難である．

一方，生物の行動知識の獲得過程に目を向けると，数回から数十回の試行錯誤の後にいくつかの可能性の中から適切な行動パターンを選択しているように見える．すなわち，生物はただ闇雲に感覚・行動写像を学習し続けているのではなく，環境のとらえかた（状態の拘束条件）をみずから仮説的に生成し，その拘束のもとで行動学習を行なうことによって擬似的に次元を落とし，その結果として学習を高速化していると考えられる．本研究では学習主体（ロボット）が，状態空間を自律的に分割して形成する外界の内部表現（拘束条件）を，環境から得られる評価に基づいて漸進的に改善していく過程を環境共創と呼び，自律的に拘束条件を生成するための必要条件としてロボットの移動に着目する．ロボットはみずから移動することによって周囲環境（外部ダイナミクス）を変化させるとともに，能動的に働きかけた結果として経験・評価を得ることができる．したがって本研究では，未知環境を能動的かつ探索的に行動するロボットが，移動の結果として逐次的に得られる入出力データと外部評価に基づいて拘束条件の更新と行動学習を同時並行に行なう，環境共創型の強化学習法（Fig. 1）を提案する．

本手法では actor-critic 法^{1), 12)}に基づくアーキテクチャを採用し，ロボットの制御器（Actor）および状態価値推定器（Critic）は NGnet を用いて関数近似される．ここで，NGnet の結合重みパラメータの更新には TD 学習¹²⁾を用いる．また，内的な状態表現の仮説的生成・評価ループを実装するた

めに，分類システム（Classifier System）^{5), 14)}の発展戦略を参考にした進化的 recruitment 戦略（Evolutionary State Recruitment Strategy）を提案する．同手法では，ちょうど生物の体細胞が新陳代謝によって生成・消滅を繰り返しながらもその巨視的な機能は恒常的に維持されること（ホメオスタシス）に倣い，NGnet の各基底関数を分類システムにおける個体とみなし，各個体の評価値を参照頻度，TD 誤差などから算出する．このようにして得られる評価に基づいて新規個体の追加，淘汰ならびにパラメータの突然変異操作を繰り返す行なうことで学習器の構造最適化を試みる．本稿では，提案手法を移動ロボットの Peg 押し課題に適用し，シミュレーションおよび実機による実験の結果からその有効性を示す．

2. 提案手法

2.1 基本構造

本研究ではロボットの制御器（Actor と呼ぶ）をセンサ入力 $\mathbf{x}(t) \in \mathbb{R}^M$ からアクチュエータ出力 $\mathbf{m}(\mathbf{x}(t)) \in \mathbb{R}^N$ までの連続関数として表現するため，Fig. 2 に示す NGnet⁸⁾を用いる．同様に， $\mathbf{x}(t)$ から状態価値 $V(\mathbf{x}(t))$ への写像を関数近似する Critic は，本研究では同図のように Actor と同一の NGnet で表現される．

NGnet は状態空間を複数の RBF で分割し，各基底関数の出力を正規化した後に線形和をとる関数近似器であり，RBF のパラメータ（中心ベクトル μ ，分散共分散行列 Σ ）と結合重み $\mathbf{w}$ を適切に調節することにより，任意の連続写像を近似表現することができる．ここで基底関数としてシグモイド関数のような大域的な関数系を用いた場合，追加学習時にすべての結合重みパラメータを更新しなければならないのに対して，RBF を基底として用いることで学習が局所的に進行するため高速な学習が期待できる．また，基底関数の追加・削除の影響が関数全体に影響しないため，追加学習の実装が容易であり，オンライン学習に適した構造的特徴を有している．さらに NGnet は正規化処理によって基底関数のカバー範囲外についても外挿して出力することができるため，

Fig. 2 Structure of NGnet

より少ない基底数で高い近似性能を得ることが可能である。具体的に Actor, Critic の出力 $m(x(t))$ および $V(x(t))$ はつぎのように求められる。

$$a_i(t) = \exp \left[-\frac{1}{2} (x(t) - \mu_i)^T \Sigma_i^{-1} (x(t) - \mu_i) \right] \quad (1)$$

$$\Sigma_i = \text{diag}(\sigma_{i1}^2, \sigma_{i2}^2, \dots, \sigma_{ij}^2, \dots, \sigma_{iM}^2) \quad (2)$$

$$b_i(t) = \frac{a_i(t)}{\sum_{j=1}^{N_B} a_j(t)} \quad (3)$$

$$m_k(x(t)) = f \left[\sum_{j=1}^{N_B} w_{kj} b_j(t) + \beta n_k(t) \right] \quad (4)$$

$$(k = 1, \dots, N)$$

$$V(x(t)) = \sum_{j=1}^{N_B} v_j b_j(t) \quad (5)$$

ここで a_i は RBF ユニット i の活性度, T は転置, Σ_i は RBF の分散共分散行列 (本研究では簡単のため対角成分 (分散) のみを考慮する), b_i は a_i を正規化した活性度, N_B は NGnet を構成する RBF の数, w_{kj} ならびに v_j は結合重み, $n_k(t)$ は行動空間を探索するために印加する正規ノイズである。また $f[\cdot]$ は Actor 出力を制御入力の値域に合わせるために用いるシグモイド関数である。

2.2 TD 学習による結合重みパラメータの更新

先にも述べたように任意の連続関数を近似するためには、使用する基底関数の数 (N_B), 各基底関数のパラメータ (μ , Σ) および出力の結合重み (w , v) を適切に定める必要がある。本研究では w と v の更新に TD 学習¹²⁾ を用いる。TD 学習ではつぎのように TD 誤差 $\delta(t)$ を定義する。

$$\delta(t) = r(t+1) + \gamma V(t+1) - V(t) \quad (6)$$

ここで $r(t)$ は環境から得られる報酬, γ は将来の報酬に対する割引率 (discount rate) である。Actor は状態価値 $V(x(t))$ をより高める行動を出力するように学習し, Critic は状態の評価を正しく見積もる (すなわち TD 誤差が 0 になる) ように学習する。このため, 結合重みパラメータ w_{kj} , v_j は TD 誤差 $\delta(t)$ を用いて (7), (8) 式のように更新される。

$$\Delta w_{kj}(t) = \eta_A b_j(t) \delta(t) n_k(t) \quad (7)$$

$$\Delta v_j(t) = \eta_C b_j(t) \delta(t) \quad (8)$$

ここで η_A および η_C はおのおの Actor, Critic ネットワークの学習率である。

2.3 進化的 recruitment 戦略による基底関数パラメータの学習

一方, 学習の精度や収束時間に大きく影響する基底関数の各パラメータ (中心・分散) および使用される基底の数は, 通常, 設計者が天下一的に与えるか, あるいは学習データの統計量からオフラインで決定される。したがって, 強化学習のようにデータが逐次的に得られるオンライン学習では, 制御器自体が未成熟である学習初期に獲得されたデータに基づいて行なわれた状態分割は学習の進展とともに逐次修正される必要があると考えられる。これまでに, これらのパラメータをオンラインで修正するモデル選択アルゴリズムが種々提案されているが, 逐次的に RBF の共分散行列の固有値・固有ベクトルを求める必要があるなど, 状態分割のための付加的な計算負荷が大きく, 高次元入出力を持つロボットに実装することは難しい^{11), 13)}。そこで本研究では, 各 RBF ユニットごとに近似性能への貢献度を個体評価として算出し, 状態空間が類似するユニット同士がその評価値に基づいて競合することで局所的に近似性能を改善していく進化的 recruitment 戦略を導入し, 学習器構造の最適化を試みる。

ここで提案する手法は, 新規 RBF ユニットの生成, 各ユニット評価値の更新, 世代交代 (淘汰・複製・突然変異) の過程からなる。本手法では RBF の属性パラメータ自体を逐次的に決定していくため, 初期状態では基底関数が存在しない。このため, ここではロボットのセンサ入力ベクトルが未経験の状態 (すなわち現存するすべての RBF ユニットの活性度 $a(x(t))$ が閾値 a_{min} 以下となる $x(t)$) であると判定された場合に, $x(t)$ を中心とした RBF を逐次追加する方策をとる。ただし, TD 誤差 $\delta(t)$ の大きさが許容誤差 δ_{min} 以内であればユニットの追加は行なわれない。

提案する適応戦略において個体評価・世代交代を適切に行なうため, 本研究では各 RBF 個体が以下に示す評価パラメータ L_i および C_i を有するものとする。ここで L_i は新規に投入された RBF ユニットが十分に評価される前に淘汰・突然変異の対象になることを防ぐために導入する評価時間パラメータである。 L_i はユニットが NGnet に新規に投入される際に 1.0 にセットされ, 次式にしたがって各時間ステップごとに減衰する。

$$\Delta L_i(t) = -\tau b_i(t) \quad (9)$$

ここで, τ は減衰係数である。更新は $L_i(t) > 0.0$ の間続けられ, $L_i(t)$ が負となった時点でこのユニットは十分評価されたと判定する。

一方 C_i は各 RBF ユニットの評価値であり、次式に示すように、TD 誤差 $\delta(t)$ の分散への寄与率を意味している。

$$C_i(t) = \lambda_C C_i(t-1) + (1 - \lambda_C) b_i(t) \delta(t)^2 \quad (10)$$

$$\lambda_C = 1 - b_i(t) \lambda \quad (11)$$

ここで λ は評価値 C_i を時間平均するための減衰係数である。ただし、アクティブでないユニット（すなわち $b_i(t)$ が小さいユニット）の評価値 C_i が減衰することを防ぐため、実際には (11) 式に示すように $b_i(t)$ による重み付けを行なった λ_C を用いている¹¹⁾。

一般に学習が進行するにつれて TD 誤差の分散値は減少するため、より適切な状態分割をしている基底関数ユニットほど C_i が小さくなる。したがって、ほぼ同時にアクティブとなる、空間的に隣接した（あるいは状態空間が一部重なっている）ユニット同士は C_i に基づいて比較することができる。

つぎに、評価値に基づいて選択・淘汰・突然変異する手順を説明する。本手法では各ステップにおいて、 $L_i(t) < 0.0$ かつ $b_i(t) > b_{act}$ （すなわち「評価済み」かつ「アクティブ」）である個体を競合ユニット (Competing units) と呼ぶ。ここで競合ユニット数が 1 であれば、局所的に競合する基底関数が存在しないため、この状態領域の関数近似はそのユニット単独で担当すれば良いことになる。この場合は評価期間パラメータ L_i をリセット ($L_i = 1.0$) し、再度ユニットの評価を行なう。一方、競合ユニット数が 2 以上である場合には、競合ユニット群から 2 つの個体（たとえば P_1 と P_2 、ただし $C_{P_1} > C_{P_2}$ とする）をランダムに選択する。先に述べたように評価値 C が小さいユニット（すなわち P_2 ）は他方 (P_1) と比べて（局所的に）良い関数近似を実現していると判断されるため、同じ属性パラメータを持つ複製体 (P'_2) が生成され、その複製体 (P'_2) の分散パラメータに (12) 式に示す突然変異操作が加えられる。これはモデル構造の局所的探索に相当する。逆に C_i が大きいユニット (P_1) は淘汰される。

$$\sigma_i \leftarrow \sigma_i \cdot [1.0 + \eta_\sigma N(0, C_{P_1} - C_{P_2})] \quad (12)$$

ここで η_σ は学習率、 $N(0, c)$ は平均 0、分散 c の正規乱数である。以上の進化的 recruitment 戦略に基づいて基底関数パラメータを逐次更新することにより、タスク達成に必要な十分な状態分割を実現する。

3. 自律ロボットの Peg 押し動作学習

3.1 シミュレーション設定

提案手法を自律移動ロボットの行動学習に適用する。実験環境を Fig. 3 に示す。ここでロボット (Robot) に課せられるタスクは、環境中にある円筒型の物体（以降 Peg と呼ぶ）をできる限り早く光源 (Light Source) まで押し進めることである。ロボットのセンサ入力 $\mathbf{x}(t)$ は 4 次元であり、おのの Peg/Light とロボット間の距離/角度である（すなわち

Fig. 3 Configuration of peg pushing task

$\mathbf{x}(t) = [\theta_{PR}, d_{PR}, \theta_{LR}, d_{LR}]^T$, Fig. 3 参照)。一方、出力は 2 次元（左右のモータ速度 $\mathbf{m}(\mathbf{x}(t)) = [v_r, v_l]^T$ ）とした。

本シミュレーションでは簡単のため運動学のみを考慮し、また、ロボット、Peg、光源の初期位置は固定とした。しかしながら、獲得される制御器の頑健性を高めるため、ロボットの初期方向 (θ_R) は各試行ごとに $[-\pi, \pi]$ [rad] からランダムに選択した。このため、試行を重ねるにつれて、任意の姿勢からスタートしても Peg を光源まで押し進めることが可能な制御器を獲得することが期待される。

また、ロボットには次式で定められる報酬 $r(t)$ を与えるものとする。

$$r(t) = f_1 \cdot f_2 \quad (13)$$

$$f_1 = \exp \left[-\kappa \cdot \frac{d_{LR}^2(t)}{d_{init}^2} \right] \quad (14)$$

$$f_2 = \begin{cases} 1.0 \cdots & \text{if } d_{PR}(t) \leq R_{Robot} + R_{Peg} \\ 0.0 \cdots & \text{otherwise} \end{cases} \quad (15)$$

ここで d_{init} はロボットと光源の初期位置での距離（つまり $d_{LR}(0)$ ）、 R_{Robot}/R_{Peg} はおののロボット/Peg の半径、 κ は正の定数である。また本実験では、Table 1 に示すパラメータを用いた。

Table 1 Parameters used in the peg pushing task

β :	noise ratio	1.0
γ :	discount factor	0.9
η_A :	learning rate for Actor	0.1
η_C :	learning rate for Critic	0.01
η_σ :	learning rate for radius of each RBF unit	0.05
a_{min} :	threshold for RBF addition	0.4
δ_{min} :	threshold for allowable TD error	0.01
$L: \tau$:	dissipation factor for lifetime	0.01
λ :	decay rate for fitness time-averaging	0.9
b_{act} :	threshold for being competing unit	0.1
κ :	decay rate for reward landscape	0.5

3.2 シミュレーション結果

3.2.1 固定幅 RBF 追加法との比較

提案手法の有効性を示すため、固定幅の RBF (分散共分散行列 $\Sigma = \sigma^2 \mathbf{I}$, ここで $\sigma = 0.2, 0.3$ および 0.4 , $\mathbf{I}$ は単位行列) を逐次追加しながら関数近似を行なった実験結果との比較を行なった. Fig. 4 および Fig. 5 に, 固定半径の場合の獲得報酬と TD 誤差の 10 回試行の平均値, 標準偏差の推移を示す. 同図より, 半径が小さい (すなわち $\sigma = 0.2$) ほど偏差も小さく, 安定して高い学習性能を示すことが確認できる. しかしながら, Fig. 7 に示すように, 半径が小さくなるにつれて多くの RBF ユニットの必要とするため計算量は増加する.

一方, Fig. 6 に示す提案手法 (進化的 recruitment 戦略) を用いた場合の推移をみると, 固定半径の最良の場合 (つまり $\sigma = 0.2$) と同等の獲得報酬を実現し, なおかつその分散が小さいことがわかる. また, Fig. 7 から, 必要とした RBF の数も $\sigma = 0.2$ の場合と比較して少ないことが見てとれる. このことから, 基底関数の形状・位置を提案手法によって適応的に修正することで学習過程が安定化し, さらに基底数の減少によって計算量を軽減できることが実験的に示された. 加えて, 提案手法では各 RBF の分散パラメータを明示的に与える必要がないため, 一種のパラメータフリーな適応手法になっているともいえる.

一試行時間 (30 秒間) 中のロボットの移動軌跡と, Critic/Actor 関数の 2 次元投影図 ($V(\theta_{PR}, \theta_{LR})$, $V(\theta_{PR}, d_{LR})$, $v_r(\theta_{PR}, \theta_{LR})$ および $v_l(\theta_{PR}, \theta_{LR})$) を Fig. 8 に示す. 同図では色が薄い部分ほど等高線の高い領域を表わしており, たとえば価値関数 $V(\theta_{PR}, \theta_{LR})$ の形状からは, ロボットが光源と Peg を正面にとらえた状態の価値がより高いこと, 価値関数 $V(\theta_{PR}, d_{LR})$ からは, 光源に近いほど状態価値が高いことが確認できる. 本実験で仮定した移動ロボットは非ホロノミックな特性を持つことから, Peg がロボットの真横 ($|\theta_{PR}| = \pi/2 [\text{rad}]$) に位置する場合には特異姿勢となるため, Peg を全く押すことができない. ここでは, 陽には Peg-光源間の距離に対する報酬しか与えていないが, この特異姿勢状態への遷移を避けるように状態価値関数を獲得したと考えられる.

3.2.2 移動ロボットの初期姿勢に対する頑健性の検証

また, 先に仮定したように, 本シミュレーションでは獲得する制御器の頑健性を高めるため, ロボットの初期姿勢 (θ_R) を各試行ごとに $[-\pi, \pi] [\text{rad}]$ からランダムに選択した. したがって, 学習後の Actor は, Peg を任意の初期方向から光源まで押し進めることが可能な感覚-行動写像 (制御器) を実現していると考えられる.

獲得された制御器を用いて, 8 種類の初期姿勢から Peg 押し動作を試みた結果を Fig. 9(a)-(h) に示す. 同図より, すべての方向から Peg を光源に押し進められることが確認できる. たとえば同図 (e) のように初期姿勢が光源, Peg と逆方向である場合 (すなわち $|\theta_{LR}| \simeq \pi$, $|\theta_{PR}| \simeq \pi$) には,

Fig. 4 Transitions of the averaged rewards (with fixed size of RBF)

Fig. 5 Transitions of the averaged TD errors (with fixed size of RBF)

Fig. 6 Transitions of the averaged reward and TD error (with evolutionary state recruitment strategy)

いったん右後方へ後退することで相対的に Peg をロボットの左前方へ押しやり, 同図 (b) と同じ姿勢 (状態) に遷移する. したがって, その後の移動軌跡は同じように推移する. また, ここで示した 8 方向の初期姿勢に限らず, 任意の方向から Peg 押し動作が可能であることも確認した.

Fig. 8 Trajectory of the robot

Fig. 7 Comparison of the number of RBF units

4. 実機による検証実験

実環境における本手法の有効性を示すため、先に示した Peg 押し課題について実機による検証実験を行なった。

Fig. 10 に示すように本実験では、Pioneer II DXe (Active Media Robotics 社製) を用いた。Pioneer II は周囲 16 方向に超音波距離センサを搭載しており、この各センサ値を $[0,1]$ に正規化して状態ベクトル $\mathbf{x}(t)$ とした。また天井に CCD カメラを設置し、その画像情報を用いて、光源方向と Peg の移動方向（ここでは報酬計算のために利用）を算出した。先のシミュレーションではロボットの初期方向をランダムに決定したが、本実験では簡単のため、常に Fig. 10(a) に示す初期状態とした。1 時間弱の学習の後に、Peg 押し動作を実現する制御器を獲得した。学習後の Peg 押し動作の一例を Fig. 10(b), (c) に示す。

5. おわりに

一般に、行動政策が不確実な状態から探索的にオンライン学習する場合、学習初期の状態分割は不確実で偏りのある観測データに基づいて行なわれるため、その状態分割は学習の伸展とともに適宜修正される必要がある。本研究では、学習主体が仮想的に状態を分割し、その状態拘束のもとで行動学習を行なうことにより擬似的に状態空間を低次元化し、学習を高速化するとともに、逐次的に得られる評価に基づいて漸進的に状態分割を改善する過程を環境共創と呼び、その重要性を指摘した。

本研究では、未知環境を能動的かつ探索的に行動するロボットが、逐次的に得られる入出力データと評価（報酬）に基づいて、環境の内部表現（RBF の形状・位置）の更新と行動学習を同時並行に行なう環境共創型の強化学習法を提案した。ここでは NGnet を用いてロボットの状態評価器ならびに制御器を関数近似し、その結合荷重パラメータの更新には TD 学習を用いた。また、内的な状態表現の形成をリソース割当て問題としてとらえ、その仮想的生成—評価ループを実装するために、分類子システムの発展戦略を参考にした進化的 recruitment 戦略を提案した。

本稿では提案手法を移動ロボットの Peg 押し課題に適用し、シミュレーションによる実験結果から、固定サイズの RBF を逐次追加する手法と比較して、より少ない基底数で状態評価器および制御器を近似可能であることを示した。また、獲得した制御器がロボットの初期姿勢に対して頑健性を有することもあわせて示した。さらに実機による Peg 押し動作学習実験を行ない、提案手法によって実時間でロボットの制御器を獲得することが可能であることも示した。

提案手法は、初期段階に形成された不確実な状態分割を

Fig. 9 Robustness with respect to eight initial heading directions

Fig. 10 Resultant trajectories of real robot implementation

局所的な競合によって段階的に解消することができるため、従来のモデル選択アルゴリズムと比べ、環境変化に対して頑健な手法であると考えられる。今後は動的環境下の学習課題に適用し、上記の検証を試みる。

謝辞 本研究の一部は文科省科研費(課題番号:14350227および14750362)によるものである。ここに謝意を表す。

参考文献

- 1) A. G. Barto, R. S. Sutton and C. W. Anderson: Neuronlike adaptive elements that can solve difficult learning control problems, IEEE Trans. Systems, Man, and Cybernetics, **13**, 834/846 (1983)
- 2) B. Fritzke: Incremental learning of locally linear mappings, Proc. of the International Conference on Artificial Neural Networks (1995)
- 3) S. Nolfi and D. Floreano: Evolutionary Robotics — The Biology, Intelligence, and Technology of Self-Organizing Machines —, A Bradford Book, MIT Press, Cambridge, MA (2000)
- 4) K. Doya: Reinforcement Learning in continuous time and space, Neural Computation, **12**, 243/269 (1999)
- 5) J. Holland: Adaptation in Natural and Artificial Systems, University of Michigan Press, MIT Press (1975)
- 6) S. Jockusch and H. Ritter: Self Organizing Maps: Local Competition and Evolutionary Optimization Neural Networks, **7-8**, 1229/1239 (1994)
- 7) 小林, 湯浅, 新井: 自律分散系の適応アルゴリズムによる強化学習のための関数近似, 計測自動制御学会論文集, **38-2**, 219/226 (2002)
- 8) J. Moody and C.J. Darken: Fast learning in networks of locally-tuned processing units, Neural Computation, **1**, 281/294 (1989)
- 9) J. Morimoto and K. Doya: Acquisition of stand-up behavior by a real robot using hierarchical reinforcement learning. Robotics and Autonomous Systems, **36**, 37/51 (2001)
- 10) J. Platt: A Resource-Allocating Network for Function Interpolation Neural Networks, **3-2**, 213/225 (1991)
- 11) K. Samejima and T. Omori: Adaptive internal state space formation by reinforcement learning for real-world agent, Neural Networks, **12-7-8**, 1143/1156 (1999)
- 12) R. S. Sutton and A. G. Barto: Reinforcement Learning, A Bradford Book, The MIT Press, Cambridge, MA (1998)
- 13) S. Sato and S. Ishii: On-line EM algorithm for mixture of local experts, Proc. of the Fifth International Conference on Neural Information Processing, 1397/1401 (1998)
- 14) S.W. Wilson: Classifier that Approximate Functions ILLIGAL Report No.2001027, Illinois Genetic Algorithms Laboratory (2001)

[著者紹介]

近藤 敏之 (正会員)

1999年名古屋大学大学院工学研究科計算理工学専攻博士課程後期課程修了。博士(工学)。同年、カリフォルニア大学サンディエゴ校客員研究員(日本学術振興会特別研究員)。2000年東京工業大学大学院総合理工学研究科知能システム科学専攻助手、現在に至る。複雑適応システムのロボット工学への適用に関する研究に従事。計測自動制御学会システム・情報部門学術講演会2002優秀論文賞受賞。日本ロボット学会、電子情報通信学会、IEEEなどの会員。

伊藤 宏司 (正会員)

1969年名古屋大学大学院工学研究科修士課程応用物理学専攻修了。70年同大学工学部自動制御研究施設助手。79年広島大学工学部第2類(電気系)助教授、92年豊橋技術科学大学情報工学系教授を経て、96年東京工業大学大学院総合理工学研究科知能システム科学専攻教授、現在に至る。運動・行動の学習と制御、自律ロボット、人間共存ロボット、義肢制御に関する研究に従事。日本ロボット学会、電子情報通信学会、IEEEなどの会員。工学博士