特集　展望

計　測　と　制　御
Vol.38, No.10, 1/6(1999)

強化学習システムの設計指針

木　村　　元*・宮　崎　和　光*・小　林　重　信*

Key Words: 強化学習 (reinforcement learning)，マルコフ決定過程 (Markov decision process)，セミマルコフ決定過程 (Semi-Markov decision process)，Q 学習 (Q-learning)，actor-critic

1. はじめに

強化学習とは，試行錯誤を通じて環境に適応する学習制御の枠組である．教師付き学習 (Supervised learning) と異なり，状態入力に対する正しい行動出力を明示する教師が存在しない．かわりに報酬というスカラーの情報を手がかりに学習するが，報酬にはノイズや遅れがある．そのため，行動を実行した直後の報酬をみるだけでは，学習主体はその行動が正しかったかどうかを判断できないという困難を伴う．学習主体「エージェント」と制御対象「環境」は以下のやりとりを行う (**Fig. 1** 参照)．

（1）エージェントは時刻 t において環境の状態観測 s_t に応じて意志決定を行い，行動 a_t を出力

（2）エージェントの行動により，環境は s_{t+1} へ状態遷移し，その遷移に応じた報酬 r_t をエージェントへ与える．

（3）時刻 t を $t+1$ に進めてステップ 1 へ戻る．

エージェントは利得 (return: 最も単純な場合，報酬の総計) の最大化を目的として，状態観測から行動出力へのマッピング (政策 (policy) と呼ばれる) を獲得する．環境とエージェントには一般に下記の性質が想定される．

- エージェントは予め環境に関する知識を持たない．
- 環境の状態遷移は確率的．
- 報酬の与えられ方は確率的．
- 状態遷移を繰返した後，やっと報酬にたどり着くような，段取り的な行動を必要とする環境 (報酬の遅れ)．

本稿では，強化学習の利用価値およびいくつかの理論的知見を紹介し，システム構成方法の指針を示す．

1.1　制御の視点から見た強化学習の特徴

強化学習が注目を集める理由の一つは，不確実性のある環境を扱っている点にある．多くの実世界の制御問題では，不確実性の扱いは厄介である．もう一つの理由は，報酬に遅れが存在し，離散的な状態遷移も含んだ段取り的な制御規則の獲得を行う点にある．設計者がゴール状態で報酬を与えるという形で，させたいタスクをエージェントに指示しておけば，ゴールへの到達方法はエージェントの試行錯誤学習によって自動的に獲得される．つまり設計者が「何をすべきか」をエージェントに報酬という形で指示しておけば「どのように実現するか」をエージェントが学習によって自動的に獲得する枠組となっている．

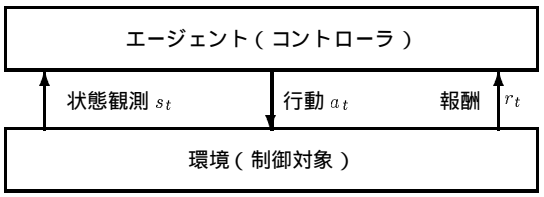


Fig. 1　強化学習の枠組．エージェントは試行錯誤を通じて適切な制御規則を獲得していく．

1.2　応用上期待できること

1.2.1　制御プログラミングの自動化・省力化

環境に不確実性や計測不能な未知のパラメータが存在すると，タスクの達成方法やゴールへの到達方法は設計者にとって自明ではない．よってロボットへタスクを遂行するための制御規則をプログラムすることは設計者にとって重労働である．ところが，達成すべき目標を報酬によって指示することは前記に比べれば遥かに簡単である．そのため，タスク遂行のためのプログラミングを強化学習で自動化することにより，設計者の負担軽減が期待できる．十分に優れた性能を持つ強化学習エージェントをコントローラとして 1 つだけ開発しておけば，あとはロボットの目的に応じて報酬の与え方だけを設計者が設定するだけで，あらゆる種類のロボット制御方法を同一のコントローラによって自動的に獲得できる．

1.2.2　ハンドコーディングよりも優れた解

試行錯誤を通じて学習するため，人間のエキスパートが得た解よりも優れた解を発見する可能性がある．特に不確実性 (摩擦やガタ，振動，誤差など) や計測が困難な未知パラメータが多い場合，人間の常識では対処し切れないことが予想され，強化学習の効果が期待できる．エキスパートの制御規則を学習初期状態に設定して，それを改善する場合と，全くのゼロから学習を開始し，設計者にとっては

* 東京工業大学 大学院総合理工学研究科
* Interdisciplinary Graduate School of Sci. and Eng., Tokyo Institute of Technology

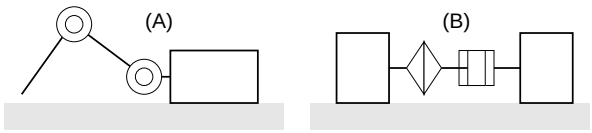


Fig. 2 学習対象としたロボット機構のその模式図．A はボディから 2 節リンクアームが張り出す構造を持ち，B はボディにねじりと曲げを行う構造を持つ．A と B はメカニズム的に全く異なるが，完全に同じ学習アルゴリズムを適用可能．

意外な新しい解を発見する場合とが考えられる．

1.2.3 自律性と想定外の環境変化への対応

機械故障などの急激な変化やプラントの経年変化のような緩慢な変化など，予め事態を想定してプログラミングしておくことが困難な環境の変化に対しても自動的に追従することが期待できる．特に宇宙や海底など，通信が物理的に困難な場合や，通信ネットワークの制御のように現象のダイナミクスが人間にとって速すぎる場合において，強化学習の自律的な適応能力が特に威力を発揮する．

2. 強化学習の適用例：ロボットの歩行動作獲得

前章で説明した強化学習の利点について，具体例を挙げて説明する．**Fig. 2** に示すように，モータを 2 個搭載した 2 自由度の機構を持つロボット A および B に対し，完全に同一の強化学習アルゴリズムを適用し，効率よく前進する動作を獲得する[10]．エージェントすなわちロボットのコントローラが獲得すべき制御規則は，現在の関節の角度を状態入力として与えられたとき，前進するような動きとなるようにモータの目標値とすべき関節の角度を出力することである．ロボットの学習目標は，効率よく前進することなので，各時刻におけるボディの前進速度をエージェントが報酬として受け取るよう設定する．エージェントとロボットは以下のやりとりを行う．(1) エージェントは状態観測としてロボットの関節の角度θ_1, θ_2を受け取る．(2) エージェントは行動出力として関節モータの角度の目標値a_1, a_2を出力．(3) ロボットは目標角度の方向へ各モータを動かす．(4) 約 0.2 秒後，ロボットはボディが移動した距離を計測し，その値を報酬としてエージェントに与える．(5) ステップ 1 に戻って繰り返す．

上記のように設定することにより，ロボットを効率よく前進させる学習問題は，エージェントが利得 (報酬の総計) を最大化するよう政策を探索する最適化問題へ帰着される．

ここで注目すべき点は，ロボット A と B がメカニズム的に全く異なるにもかかわらず，強化学習問題として見ると同じになる点と，求めるべき制御規則が比較的複雑である割に，報酬の設定が極めて簡単な点である．よって，ロボット A へ適用可能な強化学習アルゴリズムが，何も変更することなくロボット B にも適用できるという意味と，設計者が極めて簡単な報酬設定をするだけで複雑な制御規則を自動的に得られるという 2 つの意味において，強化学習による制御規則プログラミングの自動化・省力化を実現している．

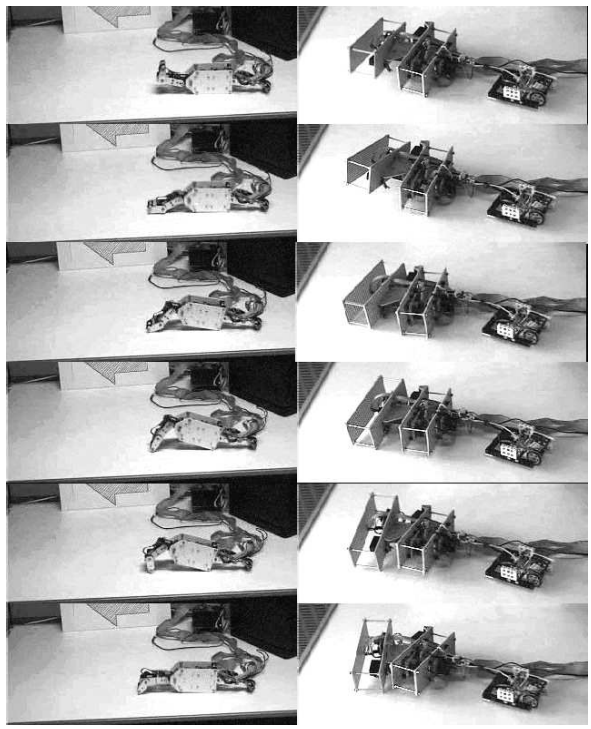


Fig. 3 強化学習による試行錯誤を通じて得た動作例

状態観測である関節の角度θ_1, θ_2および行動出力である関節モータの角度の目標値 a_1, a_2は，それぞれ 0 から 255 までの整数値をとる．報酬の値は$-128 \sim 127$ の範囲の整数値をとり，ボディが移動しない場合は 0 である．移動距離を計測するために 1 回転 200 パルスのロータリーエンコーダに直径 3cm の車輪を付け，パルスの個数を報酬の絶対値，回転方向を報酬の符号として計測する．

実時間でおよそ $5 \sim 6$ 分後の学習中の動作例を **Fig. 3** に示す．ロボット B について観察されたのは Fig. 3 に示した動作に類似する動作のみだった．ところがロボット A は Fig. 3 に示した動作以外にも学習途中においてさまざまなパターンが見られ，常に変化が観測された．特に，アーム先端を地面に触れたまま，アームを激しく上下に動かすと同時にアーム自身も曲げたり伸ばしたりして，尺取虫のように移動する様子が観測された．これはアームを下に動かすときはアーム先端と地面との摩擦力が増すため，このときアームを曲げると前進しやすく，逆にアームを上に動かすときは摩擦力が減るため，このときアームを伸ばすと前方へ滑べりやすいことを利用している．このような動作は実際に試行錯誤しない限り，獲得するのは困難である．

3. 強化学習の基礎理論

3.1 マルコフ決定過程 (MDP) とは?

環境のダイナミクスを以下のようにモデル化したのが MDP である．環境のとりうる状態の集合を $\mathcal{S} = \{s_1, s_2, \cdots, s_n\}$，エージェントがとりうる行動の集合を $\mathcal{A} = \{a_1, a_2, \cdots, a_\ell\}$ と表す．環境中のある状態 $s \in \mathcal{S}$において，エージェントがある行動 a を実行すると，環境は確率的に状態 $s' \in \mathcal{S}$へ遷移する．その遷移確率を $P^a(s, s')$ により表す．このとき環境からエージェントへ報酬 rが確率的に与えられるが，その期待値を $R^a(s, s')$ により表す．エージェントは状態集合から行動集合への写像関数 (確率分布関数でも良い) を保持する．これを政策と呼び πと表す．

3.2 MDP の最適性：割引報酬による評価

ある時間ステップで実行した行動が，その後の報酬獲得にどの程度貢献したのかを評価するため，その後得られる報酬の時系列を考える．報酬の時系列評価は利得 (return) と呼ばれる．エージェントの学習目標は，利得を最大化すること，あるいはそのような政策を求めることである．強化学習では，割引報酬合計による評価を利得として用いる場合が多い．これは，時間の経過とともに報酬を割引率γ $(0 \leq \gamma < 1)$ で割引いて合計する．ある時刻 t における状態 (あるいは行動) の利得 V_tを以下で定義する．

$$V_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \quad (1)$$

ただし r_tは時刻 t における報酬である．この V_tの期待値は，1 ステップあたり $(1-\gamma)$ の確率で停止するエージェントによって得られる報酬合計の期待値と等価である．未来の報酬を割引く理由は以下による．

(1) 実環境では，時間の経過とともに環境が変化したり，エージェントが故障等で停止する可能性があるため，時系列上の全ての報酬を同じ重みで考慮するのは妥当ではない．いわばリスクを考慮する必要がある．

(2) 無限期間時系列の利得を有限の値として扱うため．

マルコフ決定過程においてエージェントが定常政策π(時不変な政策) をとるとき，利得の期待値は，時間に関係なく状態 s だけに依存する性質を持つ．よって value は状態 s の関数になるので State-Value 関数と呼び $V^\pi(s)$ と表す．

- 最適な State-Value 関数: 全ての状態 s において $V^\pi(s) \geq V^{\pi'}(s)$ となるとき，政策πはπ'より優れているという．マルコフ決定過程では，他のどんな政策よりも優れた，あるいは同等な政策が少なくとも 1 つ存在する．これを最適政策π^*という．最適政策は複数存在することもあるが，全ての最適政策は唯一の State-Value 関数を共有する．これは最適な State-Value 関数 V^*と呼ばれ，以下に定義される．

$$V^*(s) = \max_\pi V^\pi(s) \qquad \text{for all } s \in S.$$

- 最適な Action-Value 関数：最適な政策はまた，以下に示す唯一の Action-Value 関数を共有する．

$$Q^*(s, a) = \max_\pi Q^\pi(s, a) \qquad \text{for all } s \in S \text{ and } a \in A.$$

$Q^*(s, a)$ は Q 値と呼ばれ，状態 s で行動 a を選択後，ずっと最適政策をとりつづけるときの利得の期待値を表す．$Q^*(s, a)$ が与えられた場合，状態 s において最大の Q 値を持つ行動 a が最適な行動である．

3.3 マルコフ決定過程の環境における強化学習

以下に MDP 環境下の強化学習問題の定式化を示す．

- エージェントは環境の状態遷移確率 $P^a(s, s')$ や報酬の与えられ方 $R^a(s, s')$ についての知識を予め持たない
- エージェントは環境との試行錯誤的な相互作用を繰り返して，最適な政策を学習する．

$Q^*(s, a)$ が得られれば，最適な政策は簡単に得られる．Q-learning [22] は環境との試行錯誤的な相互作用の繰り返しを通じて $Q^*(s, a)$ を推定する代表的な強化学習アルゴリズムである．**Fig. 4** にその概要を示す．

(1) エージェントは環境の状態 s を観測する．
(2) エージェントは任意の行動選択方法 (探査戦略) に従って行動 a を実行する．
(3) 環境から報酬 rを受け取る．
(4) 状態遷移後の状態 s'を観測する．
(5) 以下の更新式により Q 値を更新：

$$Q(s, a) \leftarrow (1-\alpha)\, Q(s, a) + \alpha \left[r + \gamma \max_{a'} Q(s', a') \right]$$

ただしαは学習率 $(0 < \alpha \leq 1)$，γは割引率 $(0 \leq \gamma < 1)$．
(6) 時間ステップ t を $t+1$ へ進めて手順 1 へ戻る．

Fig. 4 Q-learning アルゴリズム

- Q-learning の収束定理 [22]：エージェントの行動選択において，全ての行動を十分な回数選択し，かつ学習率αが $\sum_{t=0}^{\infty} \alpha(t) \to \infty$ かつ $\sum_{t=0}^{\infty} \alpha(t)^2 < \infty$ を満たす時間 t の関数となっているとき，Q-learning のアルゴリズムで得る Q 値は確率 1 で最適な Q 値に収束する (概収束)．ただし環境はエルゴート性を有する離散有限マルコフ決定過程であることを仮定する．その他，解析については文献 [3] を参照．
- 行動選択方法 (探査戦略)：上記の収束定理は，全ての行動を十分な回数選択しさえすれば行動選択方法 (探査戦略) には依存せずに成り立つ．よって行動選択はランダムでもよい．しかし，強化学習ではまだ Q 値が収束していない学習の途中においてもなるべく多くの報酬を得るような行動選択を求められることが多い．学習に応じて序々に挙動を改善していくような行動選択方法として，

1) ϵ-greedy 選択: ϵの確率でランダム，それ以外は最大の Q 値を持つ行動を選択．2) ボルツマン選択: $exp(Q(s, a)/T)$ に比例した割合で行動選択，ただし Tは時間とともにゼロに近付く，などの方法が提案されている [19]．

4. 応用を指向した理論と技術

前章の MDP による環境モデル化と強化学習法は，単純で強力だが，そのまま応用するには問題が多い．実用化するには，適用する問題の性質に応じて環境のモデル化やアルゴリズムを工夫する必要がある．以下に紹介する．

4.1 セミマルコフ決定過程 (SMDP)

ネットワークのルーティングやサービス，在庫管理問題など，待ち行列を扱う応用問題では，意志決定の時間間隔が一定ではなく，ランダムになる．サッカーロボットのように地面を自走するロボットでは，一定時間間隔で頻繁に意志決定すると，学習中同じ場所を行ったり来たりを繰り返すばかりで学習が進まないため，ある行動を選択したら状態観測に変化がみられるまで新たな意志決定をしないなどの方法がとられる[2]．これらの問題では，イベントドリブンな意志決定，つまり意志決定の時間間隔が任意な場合に対応した強化学習が求められる．そのような環境の数理モデルとしてセミマルコフ決定過程 (SMDP) がある．**Fig. 5** に SMDP 環境へ対応した Q-learning アルゴリズム[4) 16)]を示す．本アルゴリズムは Fig. 4 と同様の理論的

(1) エージェントは環境の状態 s_tを観測する．
(2) エージェントは任意の行動選択方法 (探査戦略) に従って行動 a_tを実行する．
(3) N時間ステップ経過後 ($N > 0$) にイベント (状態遷移) が発生するまで，環境から報酬 $r_t, r_{t+1}, \cdots, r_{t+N-1}$を受け取り続け，以下の割引報酬合計 R_{sum}を計算する：

$$R_{sum} = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^{N-1} r_{t+N-1}$$

(4) イベント (状態遷移) 発生後の状態 s_{t+N}を観測する．
(5) 以下の更新式により Q 値を更新：

$$Q(s_t, a_t) \leftarrow (1-\alpha)\, Q(s_t, a_t) + \alpha \left[R_{sum} + \gamma^N \max_{a'} Q(s_{t+N}, a') \right]$$

ただしαは学習率，γは割引率 ($0 \leq \gamma < 1$) である．
(6) 時間ステップ t を $t+N$へ進めて手順 1 へ戻る．

Fig. 5 SMDP の環境に対応した Q-learning アルゴリズム

性質を持つ．行動選択方法 (探査戦略) も同様．

4.2 部分観測マルコフ決定過程 (POMDP)

MDP の環境では，エージェントによる環境の状態観測は完全であることが仮定されている．しかし実問題では，ノイズやセンサの能力が不十分なため，状態観測に不確実性や不完全性が存在することが多い．部分観測マルコフ決定過程 (POMDP) は，MDP のモデルを拡張し，エージェントの状態観測に不確実性を付加した数理モデルであり，上記のような実問題をモデル化して解析する上で有用な知見を与える．POMDP の環境に対応した強化学習法は，いくつかのアプローチに分類できる[8)]：1) エージェント内部で，環境の状態遷移を推定 / 予測する方法 (モデルベースによる内部状態表現)，2) 有限長の過去の状態や行動の履歴を用いた内部状態表現，3) 確率的な政策を用いる方法，などが提案されている．

4.3 連続な状態空間への対応

実問題ではコントローラの状態入力が連続値のベクトルで与えられる場合が少なくない．Fig. 4 のアルゴリズムの形式に合わせて，連続値の状態入力を適宜離散化するのが普通だが，状態入力の次元数が大きいと「次元の呪い」と呼ばれる状態空間の爆発を招く．

連続な状態空間では，各状態間に位相構造 (つまり状態間の距離を定義できる) を持つ．距離的に近い状態では Q 値も近い値を持ち，2 つの状態の中間あたりに存在する状態の Q 値はそれら 2 つの Q 値の中間くらいの値を持つことが多い．そこで，連続な状態空間を持つ強化学習問題では，Q-learning における Q 値や Value の表現に関数近似を用いることが多い．関数近似を用いると，学習が高速になったり，今まで経験したことのない状態に遭遇しても，似た状態での経験を生かして適切な行動選択ができるなどのメリットがある．代表的な関数近似法として，tile coding(CMAC)，ニューラルネット，ファジィ，基底関数を固定した radial-basis-function network，nearest neighbor，locally weighted linear regression などが提案されている[19)]．上記の関数近似は多層ニューラルネットを除いて線形アーキテクチャと呼ばれる．これは，ある状態入力 s が与えられたとき，Value を近似するためにまず s を K次元特徴ベクトル$\phi(s) \in \mathcal{R}^K$にマッピングし，次に K次元のパラメータベクトル Wとの線形和により $V(s) = \phi(s) \cdot W$ のように表すものである (Q 値も同様)．線形アーキテクチャを用いた場合，ある条件下で最適値への収束が保証される[20)]．

この他，状態空間を適応的に分割していく方法[2) 15)]なども提案されている．

4.4 連続な行動空間への対応

実問題では連続値の状態入力と同様，連続値の行動出力を求められることも多い．行動空間を離散化するのが普通だが，粗く離散化すると細やかな制御ができないという問題が生じる．逆に離散化が細かすぎると探索空間が増大し，通常の離散 MDP における学習方法では，なかなか学習が進まなくなり非実用的となる．

Fuzzy 内挿型 Q-learning[6)]は，ファジィを用いた関数近似によって連続値の行動に関する Q 値を表現し，行動選択時には，行動空間を等間隔に区切ったいくつかの点について Q 値を計算する．これらの離散的なポイントにおける Q 値を用いて行動を決定するが，連続的な値の行動を選ぶような拡張ルーレット選択を提案している．

行動空間が連続的な場合は，Q-learning よりも actor-critic[19) 9)] (**Fig. 6**) を用いることが多い．これは状態の

（ 1 ） エージェントは環境において状態 s_t を観測する．
Actor は，確率的政策 π に従って行動 a_t を実行する．
（ 2 ） Critic は報酬 r_t を受け取り，次の状態 s_{t+1} を観測し，actor への強化信号として以下の TD-error を計算する．

$$(\text{TD-error}) = [r_t + \gamma V(s_{t+1})] - V(s_t) \ ,$$

γ は割引率，$V(s)$ は critic が推定した割引報酬の期待値．
（ 3 ） TD-error を用いて actor の行動選択確率を更新する．
(TD-error) > 0 ならば，実行した行動 a は比較的好ましいものと考えられるので，この選択確率を増やす．
逆に (TD-error) < 0 ならば，実行した行動 a は比較的好ましくないものと考えられるので，この選択確率を減らす．
（ 4 ） TD 法を用いて critic の value の推定値を更新する．
例えば TD(0) ならば以下のように計算する．
$V(s_t) \leftarrow V(s_t) + \alpha$ (TD-error), ただし α は学習率である．
（ 5 ） 手順 (1) から繰り返す．

Fig. 6　一般的な actor-critic アルゴリズム

Value を評価する critic と，状態観測に応じて確率的に行動選択を行う actor の 2 要素より構成される．ここで actor は行動選択の確率を調整できる物であればよい．連続値の行動であれば，actor の確率的政策は，状態入力に応じて中心値と分散が変化する正規分布とする方法がある．Fig. 6 に示すとおり，行動を選択した結果，よい状態へ遷移したなら選択した行動を強化する．正規分布の actor の場合において行動を強化するには，実行した行動へ分布の中心値を近付け，実行した行動が標準偏差の内側なら，正規分布の広がりを狭め，外側なら広げるよう調節すればよいので，実装は極めて簡単である．

4.5　マルチエージェント環境下での強化学習

高度に複雑，巨大化したシステムでは，ある程度の機能単位ごとに自律的な知的判断部を持たせ，それらを互いに協調させる自律分散システムによる管理が求められている．個々のエージェントの制御規則の獲得について，エキスパートの知識だけに頼っていた従来手法に代わり，マルチエージェント環境下での強化学習が注目されている[13]．マルコフゲームという数理モデルを用いて，ミニマックス点[12]やナッシュ均衡[7]を学習するマルチエージェント強化学習システムの解析を行った研究があるが，多くの場合，理論的解析や最適性を示すことが難しい．そこで，最適性という要求を緩和し，解の合理性を保証するというアプローチでマルチエージェント系の強化学習に適したアルゴリズムと解析を示した研究がある[14]．

4.6　強化学習アルゴリズムの階層化

階層的強化学習 (Hierarchical RL) は，大規模な問題を分割して解くという意味においてマルチエージェントと類似しており，様々な方法が提案されている[16) 17) 21)]．マルチエージェントと異なるのは，上位階層が下位階層 (サブタスク) の知識を再利用または共有する点と，下位階層での部分観測性を上位階層でカバーできる点である．

5.　応　用　例

5.1　セルラー通信システムの周波数帯の動的割りあて

いわゆる PHS のような通信システムでは，サービス地域はセルと呼ばれる地域に分割され，セル内の各通話者はそれぞれ異なる周波数帯を使うが，近接するセルでは同一の周波数帯を使えないという制約がある．限られたチャンネルで可能な通話数が最大となるように周波数を割当てることが要求される．通話サービス要求や切断の発生は確率的で，それらの頻度はセル毎に異なる上，動的に変動する．Singh らは，SMDP の強化学習に基づく方法を提案し，学習時間をさほどかけることなく既存のヒューリスティクスを上回る性能を達成した[18]．

5.2　在庫管理・生産ライン最適化

Fig. 7 に示すように，複数の加工機械を直列に連結して構成された生産ラインにおいて，在庫を最小化しつつ製品の需要を満たす最適な制御を学習する問題である．各機械

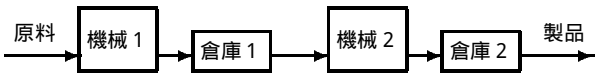


Fig. 7　製造ラインにおける在庫管理の最適化問題

の下流には倉庫 (buffer) が設置され，機械の故障中あるいはメンテナンス中の製品需要に対応することで全体の流れに与える影響を少なくする．各機械は運用時間の増加とともに故障が発生しやすくなり，故障すると修理が必要である．コストのかかる修理を回避し，在庫不足によるライン停止を避け，かつ在庫もコストがかかるのでなるべく最小限の在庫となるように，運用時間や在庫の量に応じて機械の稼働 / 待機 / メンテナンスのタイミングを制御しなければならない．この問題は SMDP としてモデル化できるが，各機械毎にエージェントを割り当てるマルチエージェントシステムが用いられている[21]．トヨタのカンバン方式等と比較し，優れた制御規則を獲得したとの報告がある．

5.3　倒立振子の振り上げ安定化

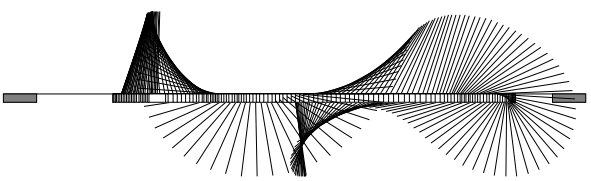


Fig. 8　倒立振子の振り上げ安定化の動作例

階層化と actor-critic に基づく連続値行動を組み合わせることで，**Fig. 8** に示す倒立振子の振り上げ安定化をゼロから学習した例が報告されている[11]．政策の初期値を教師が与え，強化学習によって政策の改善を行うことにより速やかに学習する方法も提案されている[5]．

5.4 その他の応用例

エレベータ群制御[19]，電力網の分散学習制御[17]，インターネットバナーの最適化[1]などが報告されている．

6. おわりに

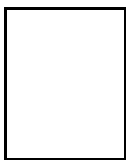
本稿では強化学習を既存の問題へ適用することに重点を置き，問題に合わせたアルゴリズムを紹介した．しかし，実問題では「試行錯誤」が許されない場合が多く，ロボットでは満足な動作を獲得する前に壊れてしまうなど問題も多い．そのため，強化学習に対して批判的な意見があるのも事実だが，教師付き学習との組み合わせなどによって解決されていくものと期待される．さらに今後，強化学習の使用を前提としたハードウエア設計がなされれば，強化学習のポテンシャルを十分に生かした新しい製品やサービスが出現する可能性がある．

参考文献

1) Abe, N. & Nakamura, A.: Learning to Optimally Schedule Internet Banner Advertisements, Proc. of 16th Int. Conf. on Machine Learning, pp.12–21 (1999).
2) 浅田 稔：強化学習の実ロボットへの応用とその課題，人工知能学会誌, Vol.12, No.6, pp.831–836 (1997).
3) Bertsekas, D.P. & Tsitsiklis, J.N.: Neuro-Dynamic Programming, Athena Scientific (1996).
4) Bradtke,S.J. & Duff,M.O.: Reinforcement Learning Method for Continuous-Time Markov Decision Problems, Advances in Neural Information Processing Systems 7, pp.393–400 (1994).
5) Doya, K.: Efficient Nonlinear Control with Actor-Tutor Architecture, Advances in Neural Information Processing Systems 9, pp.1012–1018 (1996).
6) 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，計測自動制御学会論文集, Vol.35, No.2, pp.271–279 (1999).
7) Hu, J. & Wellman, M.P.: Multiagent Reinforcement Learning: Teoretical Framework and an Algorithm, Proceedings of the 15th International Conference on Machine Learning, pp.242–250 (1998).
8) 木村 元，Kaelbling,L.P.: 部分観測マルコフ決定過程下での強化学習，人工知能学会誌, Vol.12, No.6, pp.822–830 (1997).
9) Kimura, H. & Kobayashi, S.: An analysis of actor/critic algorithms using eligibility traces: reinforcement learning with imperfect value function, Proc. of 15th Int. Conf. on Machine Learning, pp.278–286 (1998).
10) 木村 元，小林 重信：確率的傾斜法を用いた強化学習とロボットへの適用，電気学会，電子・情報システム部門誌, Vol.119, No.8 (1999).
11) Kimura, H. & Kobayashi, S.: Efficient Non-Linear Control by Combining Q-learning with Local Linear Controllers, Proceedings of the 16th International Conference on Machine Learning, pp.210–219 (1999).
12) Littman, M.: Markov games as a framework for multi-agent reinforcement learning, Proc. of 11th Int. Conf. on Machine Learning, pp.157–163 (1994).
13) 三上 貞芳：強化学習のマルチエージェント系への応用，人工知能学会誌, Vol.12, No.6, pp.845–849 (1997).
14) 宮崎和光，木村 元，小林重信：Profit Sharingに基づく強化学習の理論と応用，人工知能学会誌, Vol.14, No.5 (1999掲載予定).
15) Moore A.W. & Atkeson, C.G.: The Parti-game Algorithm for Variable Resolution Reinforcement Learning in Multidimensional State-spaces, Machine Learning 21, pp.199–233 (1995).
16) Parr, R. & Russell, S.: Reinforcement Learning with Hierarchies of Machines, Advances in Neural Information Processing Systems 10, pp.1043–1049 (1998).
17) Schneider,J., Wong,W., Moore,A. & Riedmiller,M.: Distributed Value Functions, Proc. of 16th International Conference on Machine Learning, pp.371–378 (1999).
18) Singh, S., & Bertsekas, D.: Reinforcement Learning for Dynamic Channel Allocation in Cellular Telephone Systems, Advances in Neural Information Processing Systems 9, pp. 974–980 (1997).
19) Sutton, R.S. & Barto, A.: Reinforcement Learning: An Introduction, A Bradford Book, The MIT Press (1998).
20) Tsitsiklis, J.N. & Roy, B.V.: An Analysis of Temporal-Difference Learning with Function Approximation, IEEE Transactions on Automatic Control, Vol.42, No.5, pp.674–690 (1997).
21) Wang, G. & Mahadevan, S.: Hierarchical Optimization of Policy-Coupled Semi-Markov Decision Processes, Proceedings of the 16th International Conference on Machine Learning, pp.464–473 (1999).
22) Watkins, C.J.C.H. & Dayan, P.: Technical Note: Q-Learning, Machine Learning 8, pp.279–292 (1992).

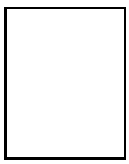
[著者紹介]

木村 元（正会員）



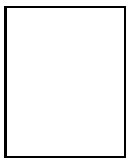
1992年東京工業大学工学部制御工学科卒業．1997年同大学大学院総合理工学研究科知能科学専攻博士後期課程修了．1998年4月，同大学大学院総合理工学研究科助手，現在に至る．人工知能，特に強化学習に関する研究に従事．人工知能学会，日本ロボット学会各会員．

宮崎 和光（正会員）



1991年明治大学工学部精密工学科卒業．1996年東京工業大学大学院総合理工学研究科知能科学専攻博士後期課程修了．同年4月，同研究科助手．1998年4月，同大学大学院総合理工学研究科リサーチアソシエイト，現在に至る．人工知能，特に強化学習に関する研究に従事．人工知能学会，日本機械学会各会員．

小林 重信（正会員）



1974年東京工業大学大学院博士課程経営工学専攻修了．同年4月，同大学工学部制御工学科助手．1981年8月，同大学大学院総合理工学研究科助教授．1990年8月，教授，現在に至る．問題解決と推論制御，知識獲得と学習などの研究に従事．人工知能学会，情報処理学会各会員．