Emergent Computation(EC)

指導院生：渡邉 真也　奥田 環
チーフ：青井 桂子　　サブチーフ：伏見 俊彦　小池 政輝

2001 年 5 月 10 日

1　強化学習の概要

近年，宇宙空間や深海など未知環境を開拓するための自律ロボットの開発が進められている．宇宙空間においてロボットの動作に対して人が教師信号を与えることはできない．そのため，ロボットは試行錯誤を繰り返すことにより結果的に良かったか悪かったかという情報により学習する必要がある．EC ゼミでは今まで GA のように創発的な設計を行うものについて学習してきたが，強化学習はこれに対して創発的な制御論を確立するものである．

1.1　強化学習 (Reinforcement Learning) とは

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) とは異なり，状態入力に対する正しい行動出力を明示的に示す教師が存在しない．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．強化学習の枠組を Fig.1 に示す．

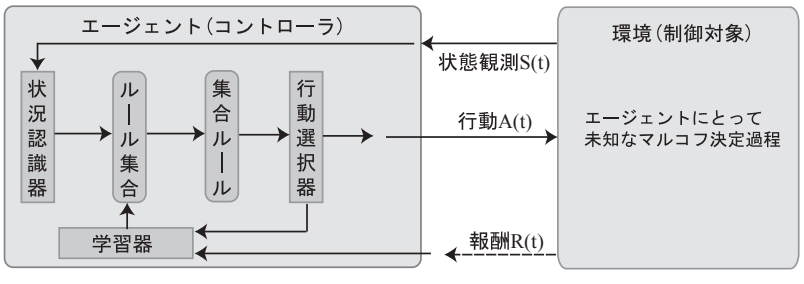

Fig. 1　強化学習の枠組みエージェントは試行錯誤を通じて適切な制御規定を獲得していく

学習主体「エージェント」[1]と制御対象「環境」[2]は以下のやりとりを行う．

強化学習の基本的アルゴリズム

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．

エージェントは各時刻において状態 (state) と行動 (action) に基づいた報酬 (reward) が与えられる．エージェントは状態，もしくは状態と行動の組に対する評価 (value) を持っており，これらの評価は報酬に基づいて更新される．行動は将来得られる報酬の期待値が最大になるように行われ，結果エージェントは最適な行動を獲得できる．

強化学習を行う際，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．

[1]学習を行うもの (実世界で言うところの生物そのもの)
[2]エージェント以外の空間を構成するもの

強化学習における環境とエージェントの一般的性質

- 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）.

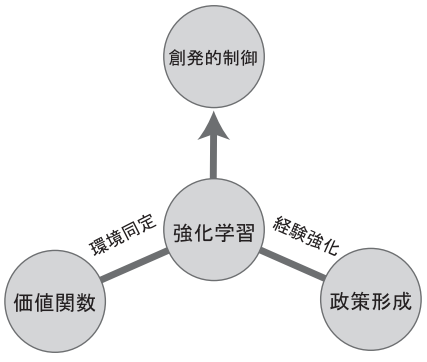

Fig. 2　強化学習の枠組み

強化学習に望まれる性能には，結果として最も大きな報酬を得るという最適性と，学習途中でもなるべく報酬を得つづけるという効率性がある.

- 環境同定アプローチ：環境を広く探索する
  (ex.Q-learning，Temporal Difference など)
- 経験強化アプローチ：効率性を重視する
  (ex.Profit Sharing)

1.2　強化学習の長所と短所

1. 長所
   - **専門家が考えた以上の解を発見する可能性がある**
   - **環境の一部が既知のとき，知識を組み込むことも可能である**
     知識ベースが不完全であってもあるいは多少の誤りが含まれてもかまわない.
   - **緩やかな環境の変化や想定外の環境変化に対応ができる**
     機械故障などの急激な変化 (環境同定型) やプラントの経年変化のような緩慢な変化 (経験強化型) 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.
   - **タスク遂行のためのプログラミング強化学習で自動化することにより，設計者の負担が軽減が期待できる**
2. 短所
   - **試行錯誤の回数**
     ある精度で政策を学習するのに必要な試行錯誤回数は少なくともパラメータ数に比例して大きくなる
   - **伝統的な人工知能の知識処理技術との相性が悪い**
   - **マルコフ性の保証がない環境などへ適用が困難である**
   - **報酬の設定を慎重にしないと最適解が求められない可能性がでる**

2　マルコフ決定過程 (Markov decision process: MDP)

強化学習理論では，環境のダイナミクスをマルコフ決定過程 (MDP) によってモデル化し，アルゴリズムの解析を行うのが一般的である.

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 $S = s_1, s_2, \ldots, s_n$，エージェントがとりうる行動の集合を $A = a_1, a_2, \ldots, a_l$ と表す．環境中のある状態 $s \in S$ 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 $s' \in S$ へ遷移する．その遷移確率を $P_r s_{t+1} = s' | s_t = s, a_t = a = P^a(s, s')$ 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 $Er_t | s_t = s, a_t = a, s_{t+1} = s' = R^a(s, s')$ により表す．エージェントの各時刻における意志決定は，政策関数 $\pi(s, a) = P_r a_t = a | s_T = s$,(ただし 全状態 s, 全行動 a において定義される) によって表される．これは単に政策 π とも呼ばれる.

- **マルコフ性**：状態 s' への遷移が，状態 s と行動 a にのみ依存し，それ以前の状態や行動には関係ないこと.
- **エルゴート性**：任意の状態 s からスタートし，無限時間経過した後の状態分布確率は 最初の状態とは無関係になること.

2.1 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 $\gamma(0 \leq \gamma < 1)$ で割引いて合計する．ある時刻 t における状態，あるいはそのとき実行した行動の利得 V_t を以下で定義する．

MDP の最適性：割引報酬による評価　　$V_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$

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．

- 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．
- 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．

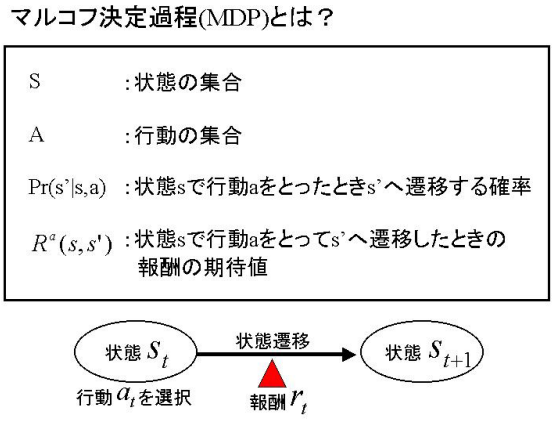

Fig. 3　マルコフ決定過程とは

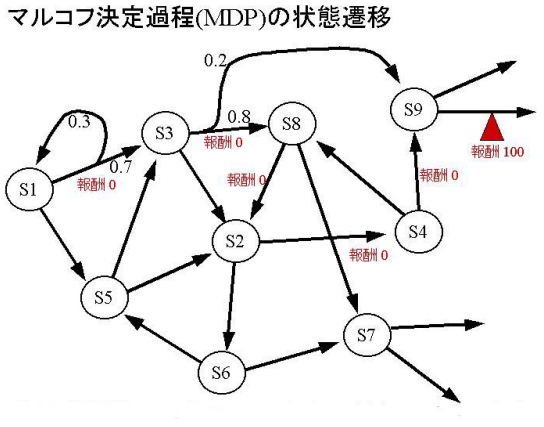

Fig. 4　マルコフ決定過程の状態遷移

3　強化学習：環境のダイナミックスが未知の場合の解法

マルコフ決定過程の環境下での強化学習問題は，以下のように定式化される．

- エージェントは環境の状態遷移確率 $Pr(s'|s,a)$ 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-learning

強化学習で最も代表的なアルゴリズムが Q-learning である．Q-learning は環境がマルコフ MDPs の時には適しているが，マルチエージェント系の環境は一般に非 MDPs であるため，Q-learning での収束性が保証されていない．Q-learning では，エージェントは状態認識器，行動選択器と学習器の 3 構成要素からなる．状態認識器は，状態と行動の対のテーブルすなわちルールベースで，各ルールは Q 値と呼ばれる重みを持っている．行動選択器には Boltzmann 選択，ϵ-greedy 選択などがある．

- Q-learning の収束定理：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 $\Sigma_{t=0}^{\infty}\alpha(t) \to \infty$ かつ $\Sigma_{t=0}^{\infty}\alpha(t)^2 < \infty$ 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 で最適な行動 value に収束する．(概収束)．ただし，環境はエルゴード製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．

- 行動選択方法 (探査戦略)：Watkins 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．学習に応じて徐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：$exp(\tilde{Q}(s,a)/T)$ 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づくなどの方法が提案されている．

学習器では次の式に従って Q 値を更新する．

$$\tilde{Q}(s_t,a_t) \leftarrow \tilde{Q}(s_t,a_t) + \alpha[r_t + \gamma max_a \tilde{Q}(s_{t+1},a) - \tilde{Q}(s_t,a_t]$$

α は学習率，γ は割引率 $(0 \leq \gamma < 1)$ 矢印 ($\leftarrow$) は左辺の変数に右辺の値を代入する操作を表す．
あるスケジュールに従って学習率 α を減少させ，多数の試行の後に Q 値が収束すると，各状態における最大の Q 値を持つルールの選択が最適な政策となる．

Q-learning のアルゴリズム

(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) 環境から報酬 r_t を受け取る
(4) 状態遷移度の状態 s_{t+1} を観測する．
(5) 以下の更新式により，Q 値を更新．
(6) 時間ステップ t を t+1 に進めて手順 (1) へ戻る．

Q-learning の欠点は解析が保証しているのはあくまで最終結果であること，場合によっては非常に無駄な試行を伴い時間のかかること，学習の途中段階での Q 値は環境の構造や学習率などのパラメータに敏感であることが指摘されている．

実問題 (生物，社会，経済 etc) では非マルコフ環境である．Q-learning ではマルコフ環境に対する収束性が証明されているが，非マルコフ環境では環境に対する最適解が存在せず，Q-learning は手法として適していない．非マルコフ環境では経験強化型の Profit Sharing などを用いると良い．

4 研究紹介

Q-learning を用いた知的照明システム

知的照明システムは知的ネットワークシステムの一つであり，複数の知的照明機器をネットワークに接続し，ネットワークに与えた「人がいるところを X[lx] の明るさにせよ」という共通の目的を満たすものである．
各知的照明の従来の制御アルゴリズムを以下に示す．

知的照明の制御アルゴリズム

(1) 各地的照明は一斉に，各々一度だけランダムに動作してみる
(2) 人の真上にいる知的照明は，(1) 後の環境 (人がいる場所の照度) をセンスし，その情報をネットワーク全体に送る．
(3) 各知的照明は (2) の行動によって，目的への達成度が上がったかどうかを判断する．上がったならば，各知的照明はもう一段階上の動作を行う下がったならば，再度 (1) の動作を行う．
(4) この手段の繰り返しによって，他の情報，自分の動作の有効性がわからなくても，知的照明全体で目的を満たすように動作することができる．

上に示した制御アルゴリズムでは判断基準を自動生成するわけではなく，予め与えている．そのため柔軟性がない．Q-leaning は目的 (報酬がもらえる場所) さえ明確であれば，判断基準を自動生成してくれるため，知的ネットワーク

システムに用いる最適化手法としてこれを行う手法として適している．次に知的照明システムに Q-learning を用いた知的照明システムのアルゴリズムを次に示す．

Q-learning を用いた知的照明システムのアルゴリズム

(1) 人の真上にいる知的照明は現在の環境 (人がいる場所の照度) 状態 S を観測し，他へ送る．

(2) 各知的照明は行動選択方法の一つである Boltzmann 選択に従って光束を強めるか弱めるかを決めて，行動 A を実行する．

(3) 各知的照明は報酬 r を受け取る．

(4) 人の真上にいる知的照明は次の環境 (人がいる場所の照度) 状態 S を観測し，他へ送る．

(5) 各知的照明はそれらの情報を基に (1) 式により Q 値を更新する

(6) この手順をくりかえす

Q-learning では判断基準が自動的に生成されるため，異なった機器を接続した場合でも柔軟に適応できると考えられる．

5 応用を指向した理論と技術

MDP による環境モデル化と強化学習法は，アルゴリズムが単純な割に最適解への収束が保障されるという意味で強力だが，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．以下にいくつかを紹介する．

5.1 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．これらの問題では，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．

5.2 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，次のようなものが提案されている．

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

5.3 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．通常の Q-learning 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，状態入力ベクトルの次元数が大きいと「次元の呪い (Curse of dimensionality)」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．

また，状態空間を適応的に分割していく方法なども提案されている．

5.4 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，あまり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．かといって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 Q-learning とその行動選択方法では，なかなか学習が進まなくなり非実用的となる．

行動空間が連続的な場合は，Q-learning よりも Actor-Critic と呼ばれる方法を用いることが多い．これは状態の Value を評価する Critic と呼ばれる部分と，状態観測に応じて確率的に行動選択を行う Actor という 2 つの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．

以下に示すとおり，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．

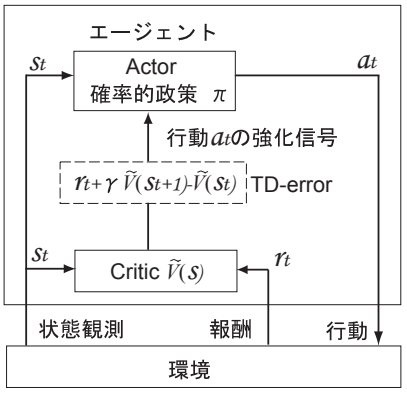

Fig. 5 Actor-Critic

一般的な actor-critic アルゴリズム

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
   $(TD-error) = [r_t + \gamma\tilde{V}(s_{t+1})] - \tilde{V}(s_t)$　　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，行動aは好ましいものと考えられる為，この選択確率を増やす．
   $(TD-error) < 0$ ならば，行動aは好ましくないと考えられる為，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．
   $\tilde{V}(s_t) \leftarrow +\alpha(TD-error)$，ただし α は学習率である．
5. 手順 (1) から繰り返す．

5.5 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．これは以下の 2 つの理由がある．

- 複雑巨大なシステム全てを集中管理することは，システム内の一部に発生した故障に対して脆く，環境等の変化に柔軟に対応し拡張することが難しく，制御ソフトウエアを作成するために膨大な労力を要するなど問題が多い．

- 稼働しているシステムへ機能単位を少しづつ付加するように別の自律的なシステムを加えることを繰り返すうちに，自然発生的に自律分散システムが構築されてしまう．

従来の分散 AI の枠組では，個々のエージェントの制御規則の獲得についてはエキスパートの知識を用いる以外に方法論が無かった．これらの解決策として，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．

5.6 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．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．

5.7 応用に必要なその他の技術

画像入力など膨大なセンサからの情報からどのようにして状態表現を生成するかについては，強化学習に限らず AI における基本的な課題である．また，様々なタスクを効率良く学習するためには，環境の状態遷移に関する知識を蓄えてタスク毎に共有／再利用することが効果的と考えられる．これはモデルベース手法と呼ばれ，多くの手法が研究されている．この他，報酬の期待値最大化だけではなくリスク最小化や複数評価規範なども研究されている．

6 強化学習の応用例

6.1 セルラー通信システムの周波数帯の動的割り当て

いわゆる PHS のような通信システムでは，サービス地域をセルと呼ばれる地域に分割し，各セル内では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，動的に変動するため，大規模になると問題が極めて複雑になる．

6.2 在庫管理・生産ライン最適化

Fig.6 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ような最適な制御を学習する問題である．

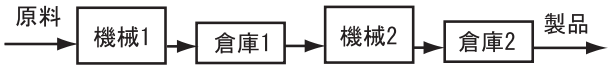

Fig. 6　生産ラインにおける在庫管理の最適化問題

各機械の下流には倉庫が設置され，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／アイドリング／メンテナンスのタイミングを制御しなければならない．

6.3 自律ロボット

未知なる空間に適応できる自律ロボットの構築に用いることも試みられている．宇宙や深海などの環境では起こりうる全ての事象を全てプログラムするのは不可能であり，強化学習が適している．しかしロボットの学習で満足な動作を得る前に壊れてしまうなどの問題も多い．
また，人間のように多自由度の身体を持ったロボットの運動獲得には従来の環境状態を定義した強化学習では困難である．これは，身体の姿勢などを表現するための状態空間が連続で状態の次元が多いため，探索に要する時間が大きくなってしまうからである．

6.4 その他の応用例

エレベータ群制御，Job-Shop スケジューリング，バックギャモンやチェスなどのゲームへの適用例がある．近年では電力網の分散学習制御やインターネットバナーのスケジューリングへの適用．

6.5 応用上期待できること

- 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．

- 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，設計者にとっては意外な新しい解を発見する場合とが考えられる．

- 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．

参考文献

1) 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 小林 (重) 研究室　http://www.fe.dis.titech.ac.jp/
2) 実世界知能技術分野生物的適応システム　http://jisp.cs.nyu.edu/RWC/rwcp/rwc-news/12/p.17.html
3) Bertsekas, D.P. & Tsitsiklis, J.N.:Neuro-Dynamic Programming, Athena Scientific
4) 冨田浩司 “ 知的ネットワークシステムへの強化学習の適用 -Q-learning による知的照明システムの構築- ”, 2000