実環境への強化学習の適用に関する実験的考察

Discussion on Applying Reinforcement Learning to A Real Environment

○土井 幹也　小野 功　小野 典彦 (徳島大)　木村 元　小林 重信 (東工大)
Mikiya Doi, Isao Ono, Norihiko Ono, University of Tokushima
Hajime Kimura, Shigenubu Kobayashi, Tokyo Institute of Technology

abstract : Reinforcement learning is expected as a learning method that is able to acquire an appropriate action policy by using reward from an unknown environment as clues. Various methods including Q-learning have been proposed so far. In most studies on reinforcement learning, reinforcement learning methods have been hardly applied to real environments where they are supposed to be useful. In this paper, we apply some representative reinforcement methods such as the Q-learning, the Stochastic Gradient Ascent (SGA) and an extended version of the SGA for the Semi-Markov Decision Processes (SMDPs) to a simple robot-navigation problem. We discuss some advantages and disadvantages of each methods based on the results.

1　はじめに

強化学習は, 未知なる環境において, 報酬を得て罰から逃れるような適切な行動を, 試行錯誤によって獲得するという基本的な適応能力を有している. 近年, 多くの研究者の注目を集めている. (図 1).

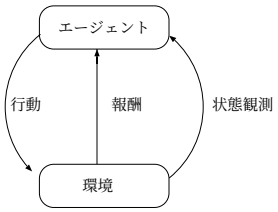

図 1: 強化学習の枠組

強化学習が注目を集める理由は, 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において不確実性の扱いが求められる. もう一つの理由には, 報酬に遅れのある段取り的な行動を獲得する点にある. 設計者がゴール状態でエージェントに報酬を与えるといった形で, 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「何をするか」をエージェントの報酬という形で指示しておくだけで「どのように実現するか」を学習によってエージェントが自動的に獲得する枠組となっている.

それゆえに, 強化学習は数多くの研究がなされてきた. 離散マルコフ決定過程 (MDPs) を対象にした先駆的研究に TD 法 [5] があり, Q-learning[6] はこの発展形として提案された. また, この Q-learning を基にし, R-learning[4],k-確実探査法 [11] など様々な改良された学習法が提案されいる.

MDPs を越えたクラスで状態観測に不確実性を付加したモデルである部分観測マルコフ決定過程 (POMDPs) が提唱されており [3], このモデルを対象に [2] はメモリーベースの学習法を提案しており, [9] は確率的政策の学習法を提案している. また, POMDPs 同様, MDPs から発展したクラスで連続時間を付加したセミマルコフ決定過程 (SMDPs) があり, [1] はこのクラスに適応可能な強化学習システムを提案している.

これら数多くの強化学習手法が研究されているが, その大部分がコンピュータシミュレーションを用いたもので, 本来活躍が期待される実環境への応用研究はほとんどなされていないのが現状である. そこで, 本論文では実ロボットの走行問題を対象に実環境に代表的な強化学習手法を適用し, それぞれの手法の利点, 欠点について実験的に考察する.

2 章では, 本論文で対象とする強化学習の環境クラスとそのクラスに対応した学習法について説明する. 3 章では, 自立走行型ロボットに 2 章で説明する学習法を実装し走行問題を対象に実験を行う. 4 章でまとめて終りとする.

2　本論文で対象とする強化学習

本論文で対象とする強化学習手法は, MDPs における Q-learning, POMDPs における確率的傾斜法, SMDPs における SMDPs に対応した確率的傾斜法である.

2.1 では各環境クラスについて, 2.2 では各手法のアルゴリズム及び利点, 欠点について述べる.

2.1　環境クラス

(1) 離散マルコフ決定過程 (MDPs:Markov Decision Processes)

従来の多くの強化学習研究はマルコフ決定過程を対象としてきた. その理由は, マルコフ決定過程が不確実性を含む実世界の多くの制御問題を精度よく近似可能だからである. 連続的な状態空間及び行動集合の上で, つまり連続的な時間の上で動作するエージェントの挙動を数学的に扱うのは煩雑な作業を必要とするが, いかなる連続的なマルコフ決定過程も有限な離散時間マルコフ決定過程によって適切に近似することができる. 時刻 t における状態 s_t で, エージェントは行動集合のいずれかを選択する. 選択された行動 a_t を実行すると状態遷移確率により確率的に決定される状態に遷移する. この時, エージェントには報酬関数により確率的に決定される報酬 r_t が与えられる.

(2) 部分観測マルコフ決定過程 (POMDPs:Partialy Observable Markov Decision Processes)

従来の多くの強化学習研究はマルコフ決定過程を対象としてるが, 実問題の多くではこのような仮定を満たさない. 実世界における強化学習では, エージェントのセンサ能力の制限等により状態観測に不完全性や不確実性を伴う隠れ状態や関数近似の問題の扱いが求められる. この隠れ状態は非マルコフ問題の一種として知られている [7]. 連続で大きな空間状態や行動の空間を扱う場合には, エージェントは何らかの汎化処理を行う必要がある. 部分観測マルコフ決定過程はマルコフ決定過程を拡張し, エージェントの状態観測に不完全性や不確実性を付加した数理モデルでり, 実問題ではこのモデルを基に学習を行う必要があると考えられる.

POMSPs に対するアプローチは, モデルを構築することで非マルコフ性を排除する立場と, モデルを構築せずに, あくまで現在の感覚入力だけから行動を決定する立場のふたつに大きく分かれる. 前者は一般にモデル構築型と呼ばれ, 後者はモデルフリー型もしくはメモリレス型と呼ばれる.

(3) セミマルコフ決定過程 (SMDPs:Semi Markov Decision Processes)

1. エージェントは環境の状態観測 s_t を受けとる.
2. エージェントは任意の行動選択法 (探索戦略) に従って行動 a_t を実行する.
3. 環境から報酬 r_t を受けとる.
4. 状態遷移後の状態観測 s_{t+1} を受けとる.
5. 以下の更新式 Q 値を更新

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

ただし α は学習率, γ は割引率 $(0\leq\gamma\leq 1)$ である.

6. 時間ステップ t を $t+1$ へ進めて手順 1 へ戻る.

図 2: Watkins の One-step Q-learning アルゴリズムの概要

セミマルコフ決定過程はマルコフ決定過程を連続な時間に拡張した数理モデルで, 何らかのイベントが発生したときに意志決定を行うイベント駆動型の意志決定モデルである [1]. MDPs では一定の時間間隔で意志決定を行っているため, 報酬に遅れのある環境では, 意志決定の間隔を短くすると報酬までの行動系列が長くなってしまい, 学習が困難になる.

また, サッカーロボットの学習では, 状態数を低減するため状態を粗くすると, 行動しても同じ状態に戻ってしまい, 学習が進まない. これに対し, 状態が変化するまで同じ行動を実行し続ける方法がとられている [8]. このように, 一定時間間隔の意志決定ではなく, イベント駆動型の意志決定が望まれる.

SMDPs はイベント駆動型であるので, 状態変化が大きい場合は短く, 状態が変化しない場合は長い間隔で意志決定を行うことができ, 無駄に多くの意志決定を行う必要が無くなる.

次節では, ここで説明した各モデルに対して本論文で対象とした強化学習手法について説明する.

2.2　強化学習手法

(1) Q-learning[6]

Q-learning は環境との試行錯誤的な相互作用の繰り返しを通じて, 最適政策をとり続けるときの利得の期待値を推定するアルゴリズムである. 図.2 にその概要を示す.

Q-learning の利点は, 環境が MDPs であれば, 最適政策の獲得が保証されることにある. そのため, 現在まで

1. 環境の観測 s_t を受け取る.
2. $\pi(a_t, W, s_t)$ の確率で行動 a_t を実行する.
3. 環境から報酬 r_t を受け取る.
4. 内部変数 W のすべて要素 w_i について以下の $e_i(t)$ と $D_i(t)$ を求める. ただし γ は割引率 $(0 \leq \gamma \leq 1)$ である.

$$\begin{aligned} e_i(t) &= \frac{\partial}{\partial w_i} \ln\bigl(\pi(a_t, W, s_t)\bigr), \\ D_i(t) &= e_i(t) + \gamma D_i(t-1), \end{aligned}$$

5. 以下の式を用いて $\Delta w_i(t)$ を求める.

$$\Delta w_i(t) = (r_t - b) D_i(t)$$

ただし b は報酬基底と呼ばれる定数である.

6. 政策の改善：以下の式で W を更新

$$\begin{aligned} \Delta W(t) &= \bigl(\Delta w_1(t), \Delta w_2(t) \cdots \Delta w_i(t) \cdots\bigr), \\ W &\leftarrow W + \alpha(1-\gamma)\Delta W(t) \end{aligned}$$

ただし α は非負の学習定数である.

7. 時間ステップ t を $t+1$ へ進めて 1 へ戻る.

図 3: 確率的傾斜法による強化学習の一般形

1. エージェントは環境において状態 s_i を観測し, 確率的政策 $\pi(a_i, W, s_i)$ により行動 a_i を実行.
2. 環境から報酬 r_i を受け取る.
3. 次のイベント発生と同時に τ_i 及び $\rho(s_i, a_i)$ が決定するので, 行動選択確率を更新する.

$$\begin{aligned} e_k(i) &= \frac{\partial}{\partial w_k} \ln\bigl(\pi(a_i, W, s_i)\bigr), \\ D_k(i) &= e_k(i) + D_k(i-1), \\ \Delta W_k(i) &= \Bigl(r_i + \frac{1 - e^{-\beta\tau_i}}{\beta}\rho(s_i, a_i)\Bigr) D_k(i), \\ W &\leftarrow W + \alpha \Delta W(i) \end{aligned}$$

ただし $e_k(i)$ は適正度, $D_k(i)$ は適正度の履歴,
$W = (w_1, w_2, \cdots, w_k \cdots)$ は政策を表す関数のパラメータ,
α は学習定数, β は減衰定数を表す.

4. 適正度の履歴を減衰：

$$D_k(i) \leftarrow e^{-\beta\tau_i} D_k(i)$$

5. $i \leftarrow i+1$ としステップ (1) から繰り返す.

図 4: SMDPs に対応した確率的傾斜法

に, 非常に多くの研究で利用されている.

一方, 欠点は解析が保証されているのはあくまでも最終結果であることと, 解析が強化学習システムの構成要素である状態認識器, 行動選択器, 学習器の 3 つのうち行動選択器を含んでいないことである. その結果, 場合によっては無駄な行動を多く含み Q 値の収束までに膨大な行動回数を必要とすることがある. また, 学習の途中段階での Q 値には近似解としての意味は無く, あくまで収束を得られなければ, そこそこの解すら得られない場合がある. さらに Q 値は環境の構造や学習率のパラメータに非常に敏感であるため, 実問題へ応用し一定の成果を得るためにはチューニングが必要となる.

(2) 確率的傾斜法 [9]

確率的傾斜法はＰＯＭＤＰの環境において, 関数近似されたメモリレスな政策, 観測から行動への写像を形成する強化学習アルゴリズムである.

エージェントの学習目標は, 報酬獲得の評価関数を最大化するように観測入力に対する行動出力を獲得することである. 確率的傾斜法では観測入力に対する行動出力の確率分布関数として定義される確率的政策を扱う.

確率的傾斜法を用いた強化学習アルゴリズムの一般形を図 3に示す. このような処理を繰り返すと, 報酬獲得に関係ない行動は打ち消され, 報酬獲得に関係する行動だけが強化される. また, 行動の履歴を強化することによって報酬に遅れのある行動も強化される. この手法は報酬を受け取った時点で今までの経験を強化するため, 経験強化型の学習アルゴリズムに分類できる.

確率的傾斜法の利点は, 各時間ステップでの状態 s_t や割引報酬の期待値 $V_N^\pi(s_t)$ 等を明示的に推定するような計算コストのかかる処理が不要であり, 実時間処理に向いた方法であることや, 確率的政策 $\pi(a, W, X)$ はエージェントの内部変数 W を用いて関数表示されているためニューラルネットやファジィなどの任意の関数近似システムを用いることが可能であることが挙げられる. また, 政策 $\pi(a, W, X)$ を行動 a を出力する確率密度関数とすれば, 連続値の行動を扱うことができる.

欠点としては, MDPs 環境では Q-learning には性能が及ばない場合があり，また山登り法に一種であるため最適性が保証されていないことなどが挙げられる.

(3) SMDPs 環境へ適用する確率的傾斜法

(2) に示す MDPs におけるアルゴリズムと SMDPs モデルのアナロジーによって機械的に拡張すると, 図 4のようになる. ここで用いられている報酬には $r_i, \rho(s_i, a_i)$ の 2 種類がある, r_i は報酬の期待値であり, $\rho(s_i, a_i)$ は状態 s_i に滞在中の単位時間あたりの報酬率の期待値である. (2) を拡張しているため利点, 欠点も (2) と同様のことが言えるだろう.

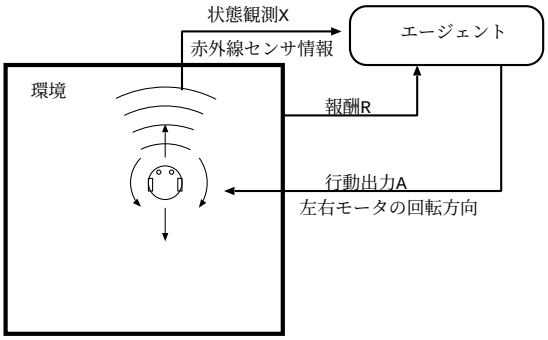

図 5: エージェント－環境

3 実験

3.1 実験環境

図 5にエージェントと環境間の概略図を示す. ロボットの目標は赤外線センサ情報による障害物回避行動かつ前進行動の獲得である. 障害物は四方を囲む壁であり, ロボットはこの壁の内側で学習を行う.

赤外線センサはロボットの左目, 右目になっており, 障害物の有無が判断できる. センサの能力が限られていることから状態は, 障害物を右前方, 左前方, 前方に感知, もしくは何も感知しなかったという 4 状態である. 出力は左右各モータの回転方向であり, 従ってロボットは前進, 後進, 右回転, 左回転の行動をとるができる. また, 前進行動かつ障害物回避行動を強化するため, 逆回転モータの個数と障害物接触の有無により負の報酬を決定する. 以上が本実験の環境である. 次節では, ここで比較実験を行う学習法の実装につて述べる.

3.2 強化学習の実装

ここでは, 2.2 で説明した各学習法の実装と, パラメータ設定について説明する.

(1) Q-learning

図 2では Q-learning の概要を示した. このアルゴリズムに前節で述べた状態, 行動, 報酬を与えロボットに実装する. パラメータは学習率 $\alpha = 0.1$, 割引率 $\gamma = 0.95$ とする. また, 行動選択法には $\varepsilon - greedy$ を用い $\varepsilon = 0.1$ とした. エージェントとロボットは以下のやりとりを行う.

a. エージェントは状態観測としてロボットのセンサ情報を受け取る.
b. エージェントは Q 値に基づき行動出力を決定し, 左右の各モータに回転方向を出力する.
c. ロボットはエージェントの指示に従い行動する.

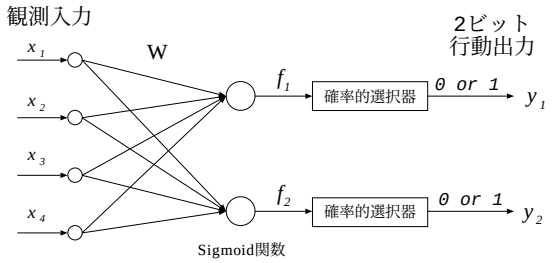

図 6: 確率的傾斜法におけるエージェントの内部構造

d. 約 0.5 秒後にロボットは停止する, この時の障害物接触の有無と逆回転したモータの数によって報酬を計算しエージェントに与える.

$$\text{報酬計算式：} r_i = -20X_1 + (-X_2)$$

ここで, X_1 は障害物に接触していれば 1 そうでないときは 0 となる変数, X_2 は, 逆回転したモータの数, すなわち行動が回転 (左または右のモータの逆回転) であれば 1 となり, 後進 (左右両方のモータが逆回転) であれば 2 となる変数.

e. Q 値を更新し, ステップ a. に戻る.

(2) 確率的傾斜法

図 6に示す内部構造のエージェントを適用し, 確率的傾斜法を実装する. 観測入力は (x_1, x_2, x_3, x_4) のベクトルとして表現される. 各要素は前節で示した観測入力 4 状態に対応しており, 要素がどれか 1 つ 1 となるとそれ以外は 0 となる. 例えば障害物が前方に感知された状態は $(x_1, x_2, x_3, x_4) = (0, 0, 1, 0)$ となる. 報酬の計算は Q-learning の場合と同じである. パラメータは学習率 $\alpha = 0.1$, 割引率 $\gamma = 0.95$ とする. 本実験に用いた確率的傾斜法の内部構造は [10] に用いられている構造と同構造である, 詳細についてはそちらを参照されたい.

(3) SMDPs 環境へ適用する確率的傾斜法

エージェントの内部構造は (2) と同じである. ここではイベントとして 1). 状態が変化したとき 2). 障害物に接触したとき 3). 同じ行動を 3.0 秒以上行ったときの 3 通りを設定した. パラメータは学習率 $\alpha = 0.07$, 割引率 $\beta = 0.8$ とする. エージェントとロボットのやりとりは次のようになる.

a. エージェントは状態観測としてロボットのセンサ情報を受け取る.
b. エージェントは内部変数より行動出力を決定し左右の各モータに回転方向を出力する.

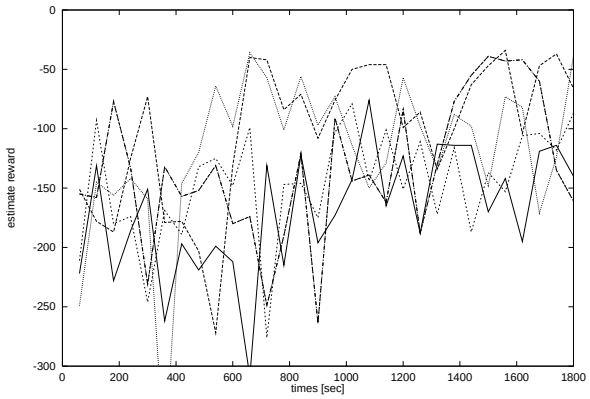

図 7: Q-learning

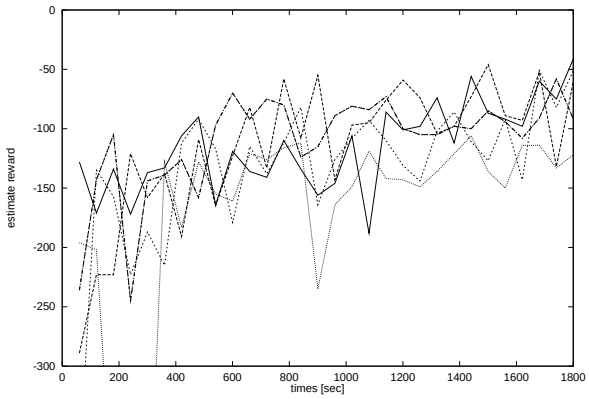

図 8: 確率的傾斜法

c. ロボットはエージェントの指示に従いイベントが発生するまで行動し続ける.

d. イベント発生直後, ロボットは停止し報酬計算を行い, エージェントに与える

$$\text{報酬計算式：} r_i = -20X_1 + (-2X_2/\tau_i)$$

ここで, X_1, X_2 は Q-learning の報酬計算式で用いた変数と同じである, τ_i はイベントが発生するまでの時間である.

e. 適正度の計算と適正度の履歴および内部変数を更新し, ステップ a. に戻る.

ここでの報酬の与え方では得られた報酬をすべて r_i としたが, 厳密なアルゴリズムでは 2.2 (3) で説明したように $r_i = -20X_1$, $\rho(s_i, a_i) = -2X_2/\tau_i$ とすべきである. この報酬の与え方は今後の課題としたい.

3.3 結果

それぞれの学習手法に対して 30 分間の学習を 5 回づつ行った. 図 7,8,9に, 各手法の 5 回試行の結果を示す. 横軸は時間を表し, 縦軸は 1 分間に得られた報酬の和を表す.

図 7より Q-learning では, 学習曲線が大きく振動していることがわかる, 図 8より, 確率的傾斜法ではある程度振動するが Q-learning に比べると安定した結果となっている. 図 9に示すように SMDPs に対応した確率的傾斜法が最も単位時間あたりの獲得報酬が高く, かつ安定した結果となった. 次節では何故, 以上のような結果が得られたのかについて考察を行う.

3.4 考察

POMDPs 環境

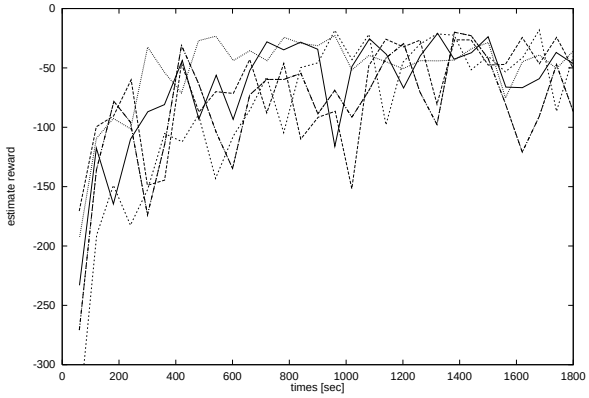

図 9: SMDPs 確率的傾斜法

今回の問題では, 環境はセンサ情報の精度より POMDPs 環境になっていると考えられる.

例えば, 壁が見えていない状態で前進行動を行った場合, 実験環境の中央付近では同じ見えていない状態に戻るだけであるが, 壁の近くであれば, 壁を感知しないままに衝突してしまう場合が見られた. 本来ならばそれぞれ別の状態として扱う事が望まれるが, センサの精度に制限され同じ状態としか判断する事が出来ない. これは状態観測が不完全な隠れ状態である.

また, 同じ位置で状態観測しても, 壁が感知される場合とされない場合が見られた, これは状態観測の不確実性である.

Q-learning と確率的傾斜法

一般に, POMDPs 環境では決定的政策よりも, 確率的政策のほうが良い場合があると知られており, Q-learning は前者にあたり確率的傾斜法は後者にあたる学習法である. また, Q-lerning は環境が MDPs でないと, 極度に性能が低下するおそれがあると言われている. 上記の考

察より, 実験環境は POMDPs 環境と言える. そのため, Q-learning は POMDPs の影響を受けてしまい, 獲得報酬の最大値は高いが学習曲線は大きく振動した結果となったと思われる. これに対し, 確率的傾斜法は Q-learning に比べると安定した結果が得られた. このことからも, POMDPs 環境では決定的政策よりも確率的政策の方が有利であると言えるだろう.

しかしながら, 確率的傾斜法では, こちらは意図していないが定位置で回転する事によって壁との接触を避ける行動を獲得しまう場合が見られた. これは, この手法が山登り法であるため, 一種の局所解に陥ってしまったと思われる.

SMDPs に対応した確率的傾斜法

上記の 2 手法に対し, SMDPs に対応した確率的傾斜法では安定した良い結果が得られている. これは, 上記の 2 手法が一定時間間隔に行動政策を決定しているのに対して, この手法はイベント駆動型であるからだと思われる. 本実験ではイベントの一つに状態遷移が起こった時を設定している. そのため実験環境が POMDPs であると思われる 1 つ原因である状態の不完全性, 壁が見えないままに衝突してしまう現象が減少されていると言えるだろう.

また, 確率的傾斜法では局所解に陥る行動が見られたが, この手法では見られなかった. これは, ある程度の時間, 同じ行動を実行し続けることによって, 内部変数の更新に良い影響を与えたためだと思われる.

本実験では, 割引率 $\beta = 0.8$ に設定してある, 確率的傾斜法で設定した行動実行時間 $\tau = 0.5$ と割引率 $\gamma = 0.95$ を考慮した場合, $e^{-\beta\tau} = \gamma$ にはならないので厳密に言うなら, 本実験結果から確率傾斜法と SMDPs に対応した確率的傾斜法を比較することはできないのだが, 同パラメータによる学習は今後の課題としたい.

SMDPs による利点

SMDPs モデルの良いところは, イベント駆動型にある. MDPs での学習法では行動実行時間を指定しなければならなかったが, その処理が不要となる. 実行時間を指定しなくてよいことは今回の実験のようなバッテリ型ロボットにとっては大変有利なことである. 何故ならば, バッテリは時間とともに消耗してくためである. バッテリがフルの時は単位時間あたりの移動距離は長くなるため, 実行時間を長くすると最初に述べた状態の不完全性が増し性能が大幅に悪化する. しかし逆に, 実行時間を短くすると頻繁に意志決定を行うので学習が遅くなる. さらに, 時間とともに単位時間あたりの移動距離は短くなっていくので, 実行時間を一定にしておくと, 性能は低下して行く.

SMDPs では実行時間の設定は不要であり, このことからも SMDPs モデルはモデル化のしやすさが伺える.

4 おわりに

本論文では実ロボットの走行問題を対象に実環境に代表的な Q-learning, 確率的傾斜法, SMDPs に拡張した確率的傾斜法を適用し, それぞれの手法の利点, 欠点について実験的に考察した. ロボットなどの実問題への適用を考えた場合，本論文における実験結果は，SMDPs に拡張した確率的傾斜法が，環境のモデル化を行いやすく，かつ良好な性能を得ることが容易であることを示唆していると考えられる.

今後の課題としては，割引率を合わせた下でのオリジナルの確率的傾斜法と SMDPs に拡張した確率的傾斜法の比較実験，SMDPs に拡張した確率的傾斜法の厳密な報酬設計での実験, ロボットの走行問題以外の問題における性能比較実験，SMDPs 環境下における探索効率の良い強化学習手法の提案などを考えている.

参考文献

[1] Bradtke, S.J.& Duff, M.O.:Reinforcement Learning Methoed for Continuous-Time Markov Decision Problems,Advances in Neural Infomation Processing Systems 7,pp.393-400,(1995)

[2] Chrisman,L.,Reinforcement learning with perceptual aliasing: The Perceptual Distinctions Approach,Proceedings of th 10th National Conference on Artificial Intelligence, pp.183-188,(1992)

[3] Lovejoy,W.S.,A Survey of Algorithmic Methoeds for Partially Observed Markov Decision Processes,Annals of Operations Research 28,pp.47-65, (1991)

[4] Schwartz, A.:A reinforcement learning method for maximizing undiscounted reward. In Proceedings of the 10th International Conference on Machine Learning,pp.298-305,(1993)

[5] Sutton,R.S.,Learning to Predict by the Methods of Temporal Differences, Machine Learning 3,pp.3-44,(1988)

[6] Watkins,C.J.C.H.,& Dayan,P.: Technical Note :Q-learning, *Machine Learning 8*,pp.55-68,(1992)

[7] Whitehead,S.D.,& Lin,L.J.:Reinforcement learning of non-Markov decision processes,*Artifical Intelligence 73*,pp.271-306,(1995)

[8] 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)

[9] 木村 元, 山村 雅幸, 小林 重信：部分観測マルコフ決定過程下での強化学習：確率的傾斜法による接近, 人工知能学会誌, Vol.11,No5.pp.761-768,(1996)

[10] 木村 元, 小林 重信, ロボットアームのほふく行動の強化学習：確率的傾斜法による接近, 人工知能学会誌, Vol.14,No1,pp.122-130,(1999)

[11] 宮崎 和光, 山村 雅幸, 小林 重信,k-確実探索法:強化学習における環境同定のための行動選択戦略, 人工知能学会誌, vol.10,No3,pp.124-133,(1995)