強化学習によるロボットの行動獲得の効率化に関する考察 —簡単なタスクからの学習 LEM—

◯野田 彰一　浅田 稔　俵積田 健　細田 耕
大阪大学工学部

A Faster Behavior Acquisition Method in a Reinforcement Learning: LEM(Learning from Easy Missions)

◯Shoichi NODA　Minoru ASADA　Sukoya TAWARATSUMIDA
Koh HOSODA
Osaka University

1 はじめに

実世界を動き回る自律ロボットには，外界の変化に対応して敏速に行動することが要求される．近年，このような適応的，反射的な行動を，ロボット自身や環境のモデルを前提とせずに獲得する手法として，強化学習が注目されてきている[1)]．しかし，強化学習の研究のほとんどは簡単なシミュレーションのみであり，実ロボットに適用した例は少ない[2)]．

強化学習においては，学習に要する時間は状態の数に対して指数関数的に増大するため[3)]，周囲の環境が複雑な実ロボットでは，学習にかなりの時間がかかることが考えられる．このため，ロボットを実際に動かして学習させることは，学習時間の観点から非現実的と思われ，実ロボットシステムの実現には，この学習時間の問題の解決が不可欠である．

我々は，強化学習を用いた実ロボットシステムの構築を通して，自律ロボットの実現を目指している．我々はまず，実ロボットを想定したコンピュータシミュレーションによって，学習時間の短縮について検討した．さらに，実ロボットによる実験を行なったが，実ロボットを動かすための学習はコンピュータシミュレーションで行ない，その学習結果を用いることで，学習にかかる時間のコストという問題を解決した．本稿では，学習時間短縮のための手法として，簡単なタスクからの学習 LEM (Learning from Easy Missions) を提案する．この手法により，学習時間を状態数に対して，従来の指数オーダから線形オーダに減らすことが可能となる．また，この手法の有効性を調べるために，簡単なシミュレーションおよび，実ロボットを想定したシミュレーションを行なったので報告する．実ロボットシステムについては，文献 [4] を参照されたい．

2 強化学習

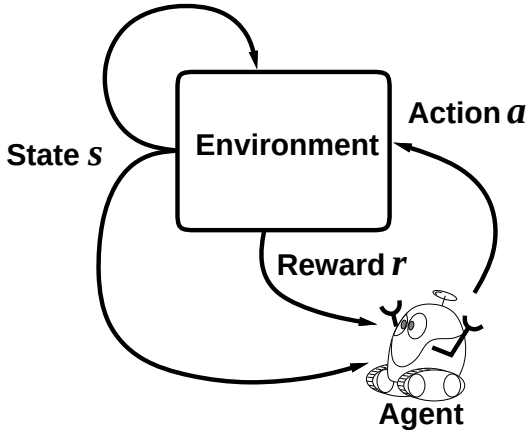


Fig.1 ロボットと環境の相互作用

強化学習は，先験的な知識をほとんど，あるいは全く与えられずに，環境からスカラー信号を受けとることのみによって行動を獲得する学習法である．**Fig.1**は，強化学習におけるロボットと環境間の相互作用を摸式的に表したものである．ロボットはセンサにより環境から現在の状態を検出し，行動を選択する．その状態と行動に基づいて，

環境は新たな状態に遷移し，ロボットへのフィードバックとしての報酬を生成する．このような相互作用を通して，ロボットは与えられた目的を達成する行動を学習する．ここでは，強化学習の中でも特に使われることの多い，Q学習についての基礎的な事柄を以下に簡単に述べる[5]．

ロボットが識別することができる状態の集合を$\mathbf{S}$とし，環境に対してとり得る行動の集合を$\mathbf{A}$とする．環境は現在の状態とロボットの行動によって確率的に遷移するマルコフ過程に従うものとする．また，現在の状態と行動の組(s,a)から次の状態s'に遷移する確率を$T(s,a,s')$とする．状態と行動の組(s,a)に対しては報酬$r(s,a)$が定義される．

一般的な強化学習の問題は，減衰する報酬の積算を最大にする政策を見つけることである．政策fは$\mathbf{S}$から$\mathbf{A}$への関数である．減衰する報酬の積算は次式で定義される．

$$\sum_{n=0}^{\infty} \gamma^n r_{t+n} \tag{1}$$

ここで，r_tはステップtにおいて，政策fが実行される時に受けとる報酬である．γは減衰係数であり，将来の報酬がどの程度政策に影響を与えるかを制御し，通常1よりやや小さい．

遷移確率と報酬の配分が定義されると，最適政策は動的計画法の手法を用いることにより求められる[6]．環境のダイナミクスを学習しながら政策を決定するものとして，WatkinsのQ学習アルゴリズム[7]は有効な手法である．

最適政策がとり続けられる時，状態sにおいて行動aをとる時の積算報酬の期待値，もしくは期待される最適行動価値関数を$Q^*(s,a)$とする．これは回帰的に次式で定義される．

$$Q^*(s,a) = r(s,a) + \gamma \sum_{s' \in S} T(s,a,s') \max_{a' \in \mathbf{A}} Q(s',a') \tag{2}$$

実システムの場合では，Tとrはあらかじめわからない場合が多いので，行動価値Q値はオンラインで見積もる必要がある．$Q(s,a)$は通常0を初期値とし，行動がとられるたびに次のように更新する．

$$Q(s,a) \Leftarrow (1-\alpha)Q(s,a) + \alpha(r(s,a) + \gamma \max_{a' \in \mathbf{A}} Q(s',a')) \tag{3}$$

ここで，αは0〜1の値をとる学習率である．Q値が与えられると，各状態sに対して$Q(s,a)$が最大となるような行動aを選ぶことによって政策が定義される．

ここで，1ステップQ学習のアルゴリズムを以下に示す．

1. $s \leftarrow$ 現在の状態
2. 通常政策fに従って行動aを選択する (ランダムな行動でも良い)
3. 行動aを実行し，次の状態s'と報酬rをそれぞれ受けとる
4. $Q(s,a)$を更新する:

   $$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha(r + \gamma \max_{a' \in \mathbf{A}} Q(s',a'))$$
5. 政策fを更新する:

   $$f(s) \leftarrow a \quad \text{ただし，} \quad Q(s,a) = \max_{b \in \mathbf{A}} Q(s,b)$$
6. 1に戻る

行動の選択については，学習の初期には各状況に対して全ての行動を試みなければならないため，ランダムにする必要がある．しかし，常にランダムな行動であると，目標とする状態になかなか到達できず，学習に時間がかかり過ぎる結果となる．これを防ぐために，経験を利用して政策に沿った行動を選択することが必要となるが，政策に沿った行動だけでは未探索な領域ができるおそれがある．このように，行動選択は経験の利用と探索のトレードオフとなるが，行動の集合$\mathbf{A}$の中から行動aを選ぶ確率を，次のボルツマン分布に従うとしたものが一般的である．

$$\frac{e^{Q(s,a)/T}}{\sum_{a \in \mathbf{A}} e^{Q(s,a)/T}} \tag{4}$$

ここで，Tは温度パラメータであり，低いほど政策に沿った行動を選ぶ確率が高くなり，高いほどランダムな行動選択になる．

3　簡単なタスクからの学習 LEM

前節で述べたQ学習などの強化学習は一種の探索問題と考えられ，状態の数が多いと計算に非常に時間がかかる．これは，実ロボットに強化学習を適用する際，大きな問題となってくる．学習時間の短縮のためのアプローチとしては，タスクを分解したり[2)]，行動をとるたびに外部から批評を与えたり (LEC:Learning with an External Critic)，いくつかのエージェントを並列に動かす (LBW:Learning By Watching) アプローチ[3)]が研究されている．しかし，タスクによっては分解することや，正確な批評を与えることが困難な場合がある．

我々はこの学習時間の短縮のために，学習の初期では初期状態をゴール状態の近くの簡単な状態に置き，徐々にゴール状態から離れた難しい状態に置くという学習のスケジューリングを行なった．これを我々は簡単なタスクからの学習 (Learning from Easy Missions:以下LEMと略記) と呼んでいる．

計算量を簡単に解析するために，Whitehead[3)]の仮定に従い，kの深さを持つ有限で等質な状態空間において，0で初期化するQ学習を考える．ここでkの深さというのは，任意の状態に到達するための最適経路の最大ステップ数であり，等質な状態空間とは，全ての行動に対して状態が可逆であることを指す．

さらに，ここでは状態遷移は決定論的で，ロボットはm種類の行動を同じ確率で選択するものとする．Q学習が収束するのに必要なステップ数を簡単に計算するために，もしQ値が初期値0から更新されるとそこで収束したものとみなす (厳密には，何ステップかのQ値の更新の後に収束することになる)．

Fig.2はそのような状態空間の例を示している．ロボットがゴールに到達した時1の報酬を与え，それ以外の時には報酬は0であるとする．上記の仮定に基づくと，初期状態s_kから次の状態s_{k-1}に移るためには，m回の試行が最悪の場合必要となる．このため，初めてゴールできるまでには最悪でm^kの試行が必要となり，そのとき状態s_1のQ値のみが更新される．次の状態s_2のQ値が更新されるまでには，m^{k-1}の試行が必要となり，全ての状態のQ値が収束するには全部で$(m^k + m^{k-1} + \cdots + m)$

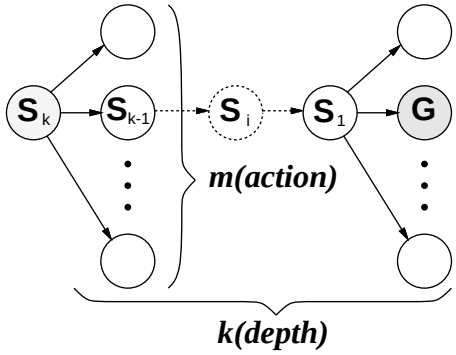


Fig.2 単純な状態空間

回の試行が必要となる．このため，普通のQ学習が収束するためにかかる時間は，kの大きさに対して指数関数のオーダになる[3)]．

一方，LEMアルゴリズムでは，最初ロボットを状態s_1に置き，ゴールに到達させる．この方法では，Q値の更新は最悪m回の試行の後である．次にロボットを状態s_2に配置し，これを繰り返す．最悪の場合でも，収束にかかる試行は$(m \times k)$回である．このためLEMでは，kの大きさに対して，収束のための時間は線形のオーダになる．

実ロボットを考える場合，状態遷移は確率的であり，状態空間も等質でない．そのために，どの状態がゴールに到達し易く，より難しい状態へ，いつシフトすれば良いかを正確に決めることは困難である．Q学習のスキーマでは最適政策に収束することが保証されているので，高い確率でゴールに到達できるおおよその状態の集合を$\mathbf{S_1}$とし，次式を満たす時にやや難しい状態の集合へシフトすることにする．

$$\Delta Q_t(\mathbf{S_1}, a) < \epsilon\ , \qquad 0 < \epsilon \ll 1 \tag{5}$$

ここで，

$$\Delta Q_t(\mathbf{S_1}, a) = \sum_{s \in \mathbf{S_1}} |\max_{a \in \mathbf{A}} Q_t(s,a) - \max_{a \in \mathbf{A}} Q_{t-1}(s,a)|$$

4　シミュレーション

4.1　二次元格子環境でのシミュレーション

LEMの効果を確認するために，**Fig.3**に示す2次元格子環境での簡単な1ステップQ学習のシミュレーションを行なった．$n \times n$の正方形の空間の

右上にゴールがあり，ロボットは左下から進んでゴールに到達した時のみ報酬1を受けとる．それ以外の時の報酬は0である．ロボットは前後左右に進む行動をとることができ，その4つの行動をランダムに選択するものとする．ただし，ロボットはフィールドから出るとリセットされる．

また，ゴールを検出するセンサがロボットの右上の範囲に (n^2-1) 個あるものとし，センサのいずれかが必ず発火するので，状態数も (n^2-1) 個となる．この場合の探索空間の深さ (ゴールまでの距離) は $2(n-1)$ である．LEMを使う場合，最初は距離1の場所 (この場合2ヶ所) にランダムに配置され，それに対応する状態の行動価値関数Qの最大値の総和が収束したと判断されると距離2の場所に配置され，以下距離 $2(n-1)$ になるまでその操作が繰り返される．LEMを使わない場合は，初期位置が左下に固定の場合とランダムの場合の2つを試みた．なお，学習率 α は0.25，減衰係数 γ は0.9としている．

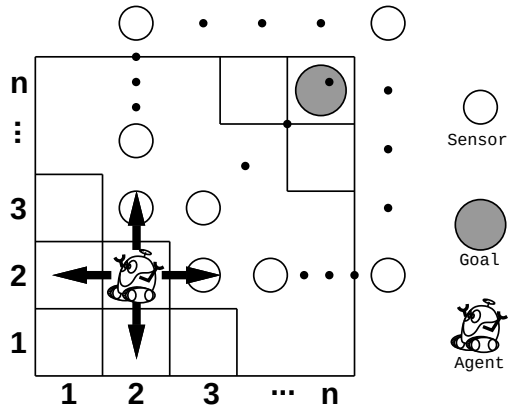


Fig.3 二次元格子環境でのシミュレーション

Fig.4はLEMを使った場合 (実線)，LEMを使わない場合で初期位置左下固定の場合 (破線)，初期位置ランダムの場合 (点線) での，探索空間の深さとQの総和が収束するまでの時間の関係である．LEMを使わない場合は深さに対して時間が指数関数的に増えていくのに対し，LEMを使うとグラフが直線的になっている．

4.2 実ロボットを想定したシミュレーション

前節でのシミュレーションは，状態空間が等質で，状態遷移も決定論的であったが，実ロボットを

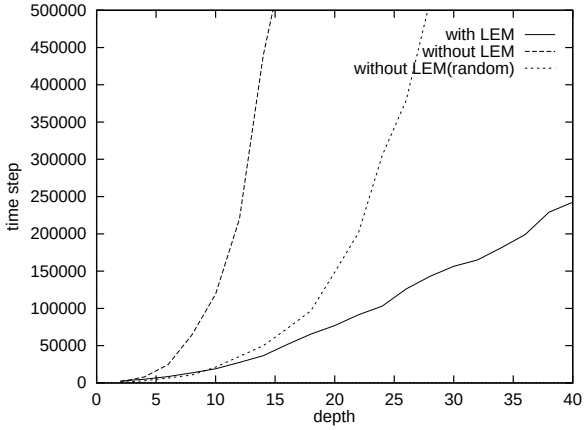


Fig.4 状態の深さと Q 値の収束時間の関係

想定した場合，状態空間を等質にすることは難しく，状態遷移も確率的になると考えられる．そのような例として，サッカーのようにボールをゴールにシュートするロボットを考える．

サッカーロボットとその環境を表したものを**Fig.5**に示す．ロボットが得られる環境からの情報は，ロボットに固定して搭載されたカメラからの画像情報のみとし，画像内のボールとゴールをトラッキングすることができる．また，ロボットは左右の車輪を独立に動かすことができるPWS(Power Wheeled Steering) システムを持っている．

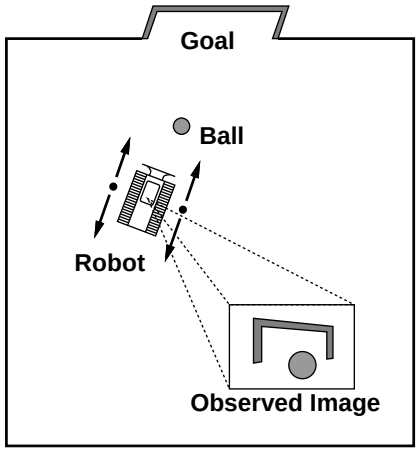


Fig.5 “サッカーロボット”

1ステップQ学習およびLEMを，このサッカーロボットへ適用するために，各パラメータを次の

ように定義した.

1. 状態集合**S**:

   ボールとゴールの画像上の見え方を分割し，その組合せによって状態集合を構成する.

   ボールについては，その位置(右，中央，左)と大きさ(大，中，小)で合計9個に分け，ゴールについても，位置(右，中央，左)，大きさ(大，中，小)，向き(右上がり，水平，左上がり)の27個に分けた．それ以外にも，ボールとゴールが見えていない状態を，前の状態から“右に消えた状態”と“左に消えた”状態にそれぞれ分けた．これらの組合せとして，全部で319個の状態がある．また，LEMを適用するために，ボールとゴールが両方見えていてゴールが大きい(近い)状態の集合を，最もゴールに到達し易い状態集合$\mathbf{S_1}$とし，以下，ゴールが中ぐらいに見える状態集合，小さく見える状態集合をそれぞれ$\mathbf{S_2}, \mathbf{S_3}$とした.

2. 行動集合**A**:

   ロボットの移動する速度を分割し，行動集合を構成する.

   ロボットはPWS車であるので，左右輪がそれぞれ前進，停止，後進でき，合計9通りの行動がとれる．この行動は，状態が変化するまでは単一の行動をとる．したがって，左右輪停止という行動は，状態の変化を生み出さないために選択せず，残りの8通りの中から一つの行動を選択する.

3. 報酬r:

   ロボットがボールをゴールに入れた状態と行動の行動価値関数値に対して1の値を与え，それ以外は0とした.

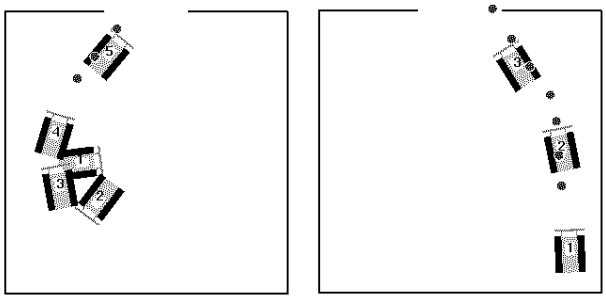
   報酬の関数をゴールやボールの距離を使って定義するほうが，学習の効率化という観点からは良いように思われるが，適切な報酬関数を設定することは難しく，行動価値関数Qが局所解に陥る危険がある.

4. 学習率αおよび減衰係数γ:

   αの値は0.25で固定とした．γは大きい程(≈ 1)，次の状態のQ値に影響され易くなり，先に続く状態を良く記憶するようになる.

   サッカーロボットの場合，γが大きいと，遠回りでも確実にシュートできるようなパスを見つけ，逆に小さいと，直線的な行動になる．ここでは，γの値は0.9とした.

シミュレーションにおける環境は**Fig.6**に示すような，300×300の正方形のフィールドで，上辺の中央に幅90，高さ18のゴールがあり，全長45，幅31のロボットが直径9のボールを蹴る．これらの単位は実世界のcm，シミュレーションの1stepは$1/30sec$と対応している．カメラはロボットの中央部についており，画角は36度である．ロボットの最大速度は3.8/step，最大回転角速度は9.1度/stepである．ロボットの質量はボールと比べて十分大きいものとし，はねかえり係数は0.5とした．また，ボールの転がり速度は床との摩擦を考えて，各ステップ毎に0.8を掛けて減衰させている.



(a) (b)

Fig.6 サッカーロボットの行動例

LEMアルゴリズムに従って，ロボットの初期位置はゴールの近くでゴールとボールが両方見える位置(状態集合$\mathbf{S_1}$)とし，ロボットやボールがフィールド外に出たり，ロボットがゴールに衝突するとリセットする．Q値が(5)式を満たして収束したと判断されれば，初期位置を$\mathbf{S_2}$にし，その次には$\mathbf{S_3}$に配置する．また，比較のためにLEMを使わない方の初期位置はロボット，ボール共にランダムな配置とした.

Fig.6は学習後にサッカーロボットが最適政策をとり続けた時の行動例である．図(a)のように，

初期位置で何も見えていない場合は回転してボールを探し，図(b)のように，ボールがゴールから離れた場所にある場合には，ドリブルしてシュートするという行動を獲得できていることがわかる．

Fig.7は状態$s \in \mathbf{S_1} + \mathbf{S_2} + \mathbf{S_3}$における$Q$値の最大値の総和の変化を表している．実線はLEMを使う場合の変化を表し，破線はLEMを使わない場合のグラフである．LEMを使う場合では，2本の矢印の位置で，それぞれ初期位置を$\mathbf{S_1}$から$\mathbf{S_2}$へ，$\mathbf{S_2}$から$\mathbf{S_3}$にシフトしている．それぞれの矢印位置で初期位置を変えず，$\mathbf{S_1}$，$\mathbf{S_2}$のままで学習を続けたものが点線で示されている．

LEMを使う場合とLEMを使わない場合とを比較すると，Q値はLEMの使用にかかわらず同じ値に収束するはずであるが，LEMを使用しない場合では学習に時間がかかり，低い値にとどまっていることがわかる．

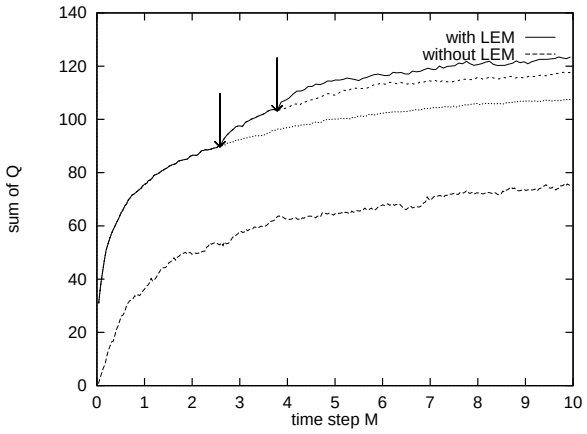


Fig.7 行動価値関数Q値の変化

5 おわりに

本稿では，強化学習の実ロボットへの適用を目的とし，実ロボットとしてサッカーロボットを想定してそのシミュレーションを行なった．実ロボットでは，このシミュレーションによる学習結果を用いている[4]．また，強化学習の学習時間短縮を目的としたLEMアルゴリズムを提案し，その有効性を確認した．

サッカーロボットでは，LEMのための初期位置としてロボットとボールの2つがあるため，配置のスケジューリングがさまざまに考えられ，何が最適かという判断は難しい．これは状態空間の分け方に密接に関係しており，強化学習を実ロボットに適用する際には，この状態空間をいかにして分けるかが，大きな問題となる．状態空間を細かく分割すると，学習にかかる時間はそれだけ増えるが，パフォーマンスは高くなると考えられる．しかし，その細分化された状態がロボットのとり得る行動に対して冗長であれば無意味である．このように状態空間は，目的とする状態と行動に基づいて分割されるべきであろう．

また，本稿ではあまり取り上げなかったが，学習時間の短縮化に対して，学習率，減衰係数，温度パラメータなど各種パラメータの設定についても考える必要がある．

参考文献

[1] J. H. Connel, S. Mahadevan, editors, *Robot Learning,* Kluwer Academic Publishers, 1993.

[2] J. H. Connel and S. Mahadevan, "Rapid task learning for real robot," In *Robot Learning,* chapter 5, Kluwer Academic Publishers, 1993.

[3] S. D. Whitehead, "A complexity analysis of cooperative mechanisms in reinforcement learning," In *Proc. AAAI-91,* pp607-613, 1991.

[4] 俵積田，浅田，野田，細田，"視覚に基づく強化学習によるサッカーロボットのシューティング行動の実現"，第4回ロボットシンポジウム予稿集, 1994.

[5] L. P. Kaelbing, "Learning to achieve goals," Proc. of the IJCAI'93, pp.1094-1098, 1993.

[6] R. Bellman, "Dynamic Programming," Princeton University Press, Princeton, NJ, 1957.

[7] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D Thesis, King's College, University of Cambridge, May 1992.