マルチエージェント環境における政策推定

北尾　健大†　　三浦　孝夫†

† 法政大学大学院 理工学研究科 システム理工学専攻　〒 184–8584 東京都小金井市梶野町 3-7-2
E-mail: †takehiro.kitao.7j@stu.hosei.ac.jp, ††miurat@hosei.ac.jp

あらまし　マルチエージェント環境において，学習の進行状況や他エージェントの政策や環境の変化により，エージェントの政策は変化することが考えられる．本研究では，マルチエージェント環境での政策の推定を目的とする．具体的には，エージェントの政策を観測情報からカルマンフィルタを用いて動的に学習し，推定を行う手法を提案する．提案手法を強化学習タスクに適用し，有効性の検証を行う．

キーワード　強化学習，Q 学習，カルマンフィルタ

1. 前　書　き

近年エージェント技術の利用が一般的になっている．エージェントとは，動作主を表す．例えば，環境にエージェントが 1 体存在する，シングルエージェントシステムはゲームやシステム制御などに使用されている．また．複数のエージェントが環境を共有するマルチエージェントシステムの場合，分散処理，ロボットサッカー，サプライチェーンなどに使用されている．しかし，昨今のハードウェアやソフトウェアの性能の向上により，扱うタスクが複雑になってきている．そのため事前知識として，タスクのルールなどをエージェントへ付与することが困難である．また，事前にルールを付与できたとしても，タスクに変化が生じたとき，エージェントは対応することができない．

強化学習 [1] は教師あり学習と異なり，明示的正解を示す教師情報が存在しない．教師情報がない代わりに，エージェントは事前知識を用いることなく，環境の相互作用から学習をオンラインで行う．エージェントシステムにおいて，強化学習を用いることは，タスクに対して柔軟に適応する．

強化学習を使用する際，マルチエージェント環境はシングルエージェント環境より，エージェント間の協調を考えなければいけない．エージェントの選択した行動は，環境にのみ影響を与えるだけではなく，複数のエージェントにも影響を与えている．このように，環境とエージェント，エージェントとエージェントは密に影響を及ぼし合っている．各エージェントが目的を達成するために，複数のエージェントとの影響を考慮した協調動作をとる必要がある．各エージェントが協調することで，より良い解が求まると考えられる．しかし，エージェントが協調するには，複数エージェントの行動の選択基準を理解しなければならない．ここで，行動の選択基準を政策と呼ぶ．

マルチエージェント強化学習 [10], [11] において，エージェント間の通信の有無により，協調を達成する方法が変わってくる．通信が可能であれば，自分の情報を相手に教え，相手の情報を得ることで協調を達成しようとする．その代表的な学習方法として，ゲーム理論に基づく強化学習である [12], [13]．例えば，ナッシュQ 学習 [12] では，ナッシュ均衡を基に学習と行動選択を行う手法である．学習を繰り返すことで，ナッシュ均衡の意味でエージェント間の協調を獲得できる．しかし，各エージェントは全エージェントの報酬と行動を観測できると仮定しているが，現実エージェントへ適用するには現実的でない．通信を行わないマルチエージェント強化学習として，Nagayuki らの研究 [8] がある．Nagayuki らは，政策を推定するエージェントに対し，内部モデルを仮定し，モデルの更新を観測情報を用いて行う．内部モデルは強化学習を基としているため，状態数が莫大な場合や状態が連続値の場合を考慮できない．

本研究では変化する政策を推定するために，2 つの問題を論じる．第 1 の問題は政策の変化をどう捉えるか，第 2 の問題は推定した政策をどのように学習に反映させるかである．上記の問題に対し，本研究ではカルマンフィルタを用いた政策推定を行い，マルチエージェントに拡張した Q 学習を用いて学習する手法を提案する．

本研究の貢献は以下の通りである．

(1) カルマンフィルタのパラメータ推定を使用した政策推定の提案
(2) 推定した政策を使用した学習方法の提案

第 2 章で強化学習について述べ，第 3 章ではカルマンフィルタについて述べる．第 4 章で提案手法について述べる．第 5 章で提案手法の有用性を論じ、第 6 章で結論とする．

2. 強 化 学 習

2.1 シングルエージェント強化学習

シングルエージェント強化学習とは 1 体のエージェントが現在の環境を知覚し，それを元に行動を選択・実行しながら状態を遷移することで，目的を達成する学習手法である [1]．教師あり学習とは異なり，強化学習では目的に対して具体的な解法は与えず，取った行動に対して「報酬」という形で評価値を与えることで適切な行動を学習する．

代表的な強化学習手法として，TD 学習，Q 学習， sarsa などがある．これらの手法は状態と行動を離散値として扱う．現在の時刻 t における状態と行動は s_t，a_t であり，次の時刻 t+1 での状態は s_{t+1} となる．状態間の遷移確率を $p^{a_t}_{s_t,\, s_{t+1}}$ と表すと，これは条件確率として記述できる．

$$p^{a_t}_{s_t,s_{t+1}} = P\{s_{t+1}|s_t,\, a_t,\, s_{t-1},\, a_{t-1}\cdots\, s_0,\, a_0\}$$

このように n 個前の時刻まで考慮している場合を n 重マルコフ過程と呼ぶ．また，1 つ前の時刻のみ考慮することを (単純) マルコフ過程と呼び下記の式で表す．

$$p^{a_t}_{s_t,s_{t+1}} = P\{s_{t+1}|s_t, a_t\}$$

エージェントが何らかの行動 a を決定することで状態が遷移し，それに対応して報酬 (正負を含める) を得る．このような離散マルコフ決定過程を基にした逐次的な決定過程のことをマルコフ決定過程と呼ぶ．

状態 s において実行可能な行動 A(s) がいくつも存在するとき，実行すべき行動の選択基準を政策という．政策は，状態 s において行動 a を行う確率 π (s,a) で表現される．エージェントは学習することで問題を効率的に解く政策や最終的に受け取る報酬が最大となる政策を獲得できる．

強化学習では，解決すべき問題に対して，目標 (解) を設定する．強化学習は，目標に合わせて報酬を与え，選択した行動に基づき報酬を決定する．ここで，エージェントは得られる報酬を知覚している．この報酬の総和を最終的に最大化することが強化学習の目的である．しかし，報酬は現在の行動によって与えられるが，選択・実行した行動が後続の報酬に影響することが考えられる．このため，遅延報酬を考慮して報酬を最大化する必要がある．

強化学習では報酬を使いルールの価値 (状態行動対) を決定する．特に，本研究で使用する Q 学習では推定値ベースの行動価値関数である Q 関数を利用する [4]．すなわち，ある状態においてある行動を実行すれば，どれだけの報酬を得ることができるかの期待値を Q 値は意味する．ここで状態 s, 行動 a の時の価値を Q(s,a) とすると，Q 値は適切に更新されれば最適な期待報酬値に近づいていく．適切な評価値に近づけるためには，報酬 (直接的な報酬と遅延報酬を含めた) を手掛かりに試行錯誤しながら更新を行い，値を保持する．状態 s の次状態を s' とすると，下記に Q 学習の更新式を示す．

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

エージェントはルールの価値を政策に変換して，その政策を基に行動を決定する．しかし，行動選択手法を用いれば，直接的にルールの価値から行動を選択することができる．主な行動選択手法として「グリーディ手法」「ε -グリーディ手法」「ソフトマックス手法」などがある

強化学習の現実問題への適用では，状態数が膨大であることや状態の表現が連続値であることが問題になる [5]．例えば，式 (1) の状態が離散値であることを仮定する Q 学習では，保持する Q 値は状態と行動の組合わせの数だけ存在するため，状態数の増加に比例して必要なメモリ量が膨大になる．また，Q 値は状態と行動の組によって決定するため，状態が無数に存在する連続値には対応できない．

価値関数 Q を関数近似するため，ϕ_s を状態を特徴ベクトルに変換した w を重みベクトルとする．下記に価値関数を線形近似した式を示す．

$$\hat{Q}_{\theta(s,a)} = \sum_{j=0}^{p} w_j \phi_{s,a} = \phi^T_{s,a} W$$

状態を特徴ベクトルに変換する方法として，ガウスカーネルやタイルコーディング [6]，フーリエ級数 [7] を使用する方法が提案されている．しかし，上記で示した線形の形では価値関数を表現するのには限界がある．そこで，ニューラルネットワークを用いた非線形近似する方法が提案されている．しかし，ニューラルネットワークを用いたものは，学習が発散する例が示されており正しく学習できるとは限らない．

2.2 マルチエージェント強化学習

マルチエージェント強化学習とは，複数のエージェントが環境を共有している強化学習である．シングルエージェント環境と違い，マルチエージェント環境においての状態遷移は，全エージェントの行動によって状態遷移確率が決まる．そのため，各エージェントの報酬は自分だけの政策によって決まらず，全エージェントの政策を考慮しなければならない．例として，環境内にエージェントが i, j の 2 体を配置する．現在の状態を s, 次状態を s' とし，エージェント i の状態 s での行動を a_i, 状態 s' のでの行動を a'_i, エージェント j の状態 s' での行動を a'_j とすると，エージェント i の Q 学習の更新式は以下の式である．

$$Q_i(s,a_i,a_j) \leftarrow Q_i(s,a_i,a_j) + \alpha[r + \gamma \max_{a'_i,a'_j \in A(s')} Q_i(s,a'_i,a'_j) - Q_i(s,a_i,a_j)] \quad (2)$$

上記の式では $\max_{a'_i,a'_j \in A(s')} Q(s,a'_i,a'_j)$ を計算しなければならない．しかし，エージェント i が エージェント j の次状態 s' でとる行動 a'_j を推定するのは容易ではない．

マルチエージェント強化学習の利点は，ロバスト性（頑強性），並列性，拡張性などがあげられる．ロバスト性（頑強性）では，1 体以上のエージェントがタスクに失敗しても複数のエージェントにそのタスクを引き継ぐことができる．並列性では，中央処理に比べ，各エージェントが並列でタスクを処理することができるため処理の効率が向上する．拡張性では，エージェントの追加のなどが容易に行うことができ，規模の大きな問題に対しても対応可能である．

3. カルマンフィルタ

カルマンフィルタとは，システムの観測値から観測不可能なシステムの状態を推定するフィルタリング手法である [2], [3]．カルマンフィルタは，状態と観測値の関係を，下記の状態空間モデルを用いて以下の 2 つの式で表す．

$$x_i = A_{i-1}x_{i-1} + w_{i-1} \quad (3)$$

$$z_i = H_i x_i + v_i \quad (4)$$

ここで，式 (3) は状態方程式と呼ばれ，行列 A_{i-1} は時刻 i での状態 x_i と時刻 $i-1$ での状態 x_{i-1} を関連付ける行列である．式 (4) は観測方程式と呼ばれ，行列 H_i が時刻 i での観測値と

時刻 i での状態の関係を表している．式 (3) と式 (4) において，w_{i-1} と v_i はそれぞれプロセスノイズ，観測ノイズを表している．各ノイズは $p(w) \sim N(0, Q)$, $p(v) \sim N(0, R)$ に従う．

カルマンフィルタは，予測と更新の 2 つのステップを繰り返し実行し，状態を推定していく．

予測ステップ

$$\hat{x}_{i|i-1} = A\hat{x}_{i-1|i-1} \tag{5}$$

$$P_{i|i-1} = AP_{i-1|i-1}A^T + Q \tag{6}$$

更新ステップ

$$K_i = P_{i|i-1}H^T(HP_{i|i-1}H^T + R)^{-1} \tag{7}$$

$$\hat{x}_{i|i} = \hat{x}_{i|i-1} + K_i(z_i - H\hat{x}_{i|i-1}) \tag{8}$$

$$P_{i|i} = (I - K_iH)P_{i|i-1} \tag{9}$$

ここで $\hat{x}_{i|i-1}$ は事前状態推定値と呼ばれ，時刻 $i-1$ までのデータに基づいた，時刻 i での状態の予測推定値であり，$\hat{x}_{i|i}$ は事後推定値と呼ばれ，時刻 i までのデータに基づいた，時刻 i での推定値である．また $P_{i|i-1}$ は $\hat{x}_{i|i-1}$ と同様に，時刻 $i-1$ までのデータに基づいた，時刻 i での予測共分散である．同じく $P_{i|i}$ は時刻 i までのデータに基づいた，時刻 i の共分散である．K_i をカルマンゲインと呼び，共分散が最小になるように式 (7) で計算する．共分散が減少することに伴い，カルマンゲインも減少する．カルマンフィルタを使用するには初期値として，状態の初期値 $\hat{x}_{i|i-1}$ と予測共分散の初期値 $P_{i|i-1}$，ノイズの共分散 Q, R を設定しなければならない．カルマンフィルタのイメージ図を以下に示す．

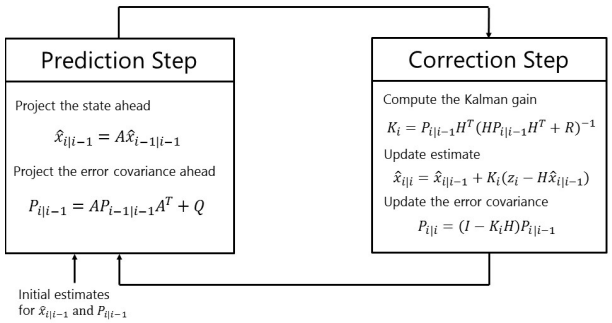

図 1　フィールド

カルマンフィルタの状態空間モデルを利用することで，パラメータ推定が可能である．推定したいパラメータを θ とすると，状態空間モデルは以下の式で表現される．

$$\theta_i = \theta_{i-1} + w_{i-1} \tag{10}$$

$$z_i = H_i\theta_i + v_i \tag{11}$$

パラメータは教師データを使用し，カルマンフィルタを更新することで逐次的に推定ができる．式 (10) のプロセスノイズを $\mathbf{0}$ に設定すると，不時変のパラメータ推定ができる．一方で，式 (10) のプロセスノイズに値を設定することで，時変のパラメータ推定ができる．

4. 提 案 手 法

4.1 政 策 推 定

学習やエージェント間の影響，環境とエージェント間の影響により，エージェントの選択する行動，つまり政策は変化すると考えられる．本研究では 3 章で示した，カルマンフィルタによるパラメータ推定を使用し，エージェントの政策を推定する．カルマンフィルタの枠組みを使用することで，変化する政策を捉える．

具体的には，エージェントの状態を特徴量，選択した行動を正解ラベルとして扱い，逐次的にカルマンフィルタを使用してパラメータ更新し，政策を推定する．カルマンフィルタは連続値を扱うことを前提としているため，エージェントの行動が離散値の場合，そのまま利用できない．そのため，離散値の行動を 1-of-K 表現へ変換し，カルマンフィルタの更新を行う．1-of-K 表現とは，ベクトルのある要素のみが 1 で，それ以外の要素は 0 のベクトルのことである．例えば，エージェントの選択可能な行動数を 5, 選択した行動を 3 とすると，$(0\,0\,1\,0\,0)^T$ が 1-of-K 表現である．また，エージェントが知覚した状態を，直接カルマンフィルタのパラメータ更新へ適用することはできない．本研究では，ガウスカーネルを使用し，エージェントの状態を特徴ベクトルへ変換する．ガウスカーネルを用いる理由は，正規分布とパラメータの線形結合で，任意の関数を近似 [9] するためである．パラメータを更新すると関数の形が変わるため，政策の変化に対応できる．

状態が連続値，行動が離散値であるタスクを例に政策の学習方法を説明する．行動が離散値の場合を扱うが，カルマンフィルタは連続値を扱うことが前提のため，行動が連続値であっても提案手法は適用可能である．エージェントの選択可能な行動数を $|A|$ とする．加えて，$\phi(s)$ は状態 s を n 次元の特徴ベクトルへ変換する関数とする．カルマンフィルタのパラメータ推定の式 (10), (11) において，θ は政策を表現する $n|A|$ 次元のパラメータのベクトル，z が離散値の行動を 1-of-K 表現へ変換した $|A|$ 次元のベクトルである．ここで，H はパラメータ θ と行動 z の関係を表す行列である．本研究では，エージェントは現在の状態から行動を選択してると仮定し，状態 s の添字をつけ，行列 H_s は以下の行列で表現する．

$$H_s = \begin{pmatrix} \phi(s)^T & \dots & 0 & \dots & 0 \\ 0 & \dots & \phi(s)^T & \dots & 0 \\ \vdots & \dots & \vdots & \dots & \vdots \\ 0 & \dots & 0 & \dots & \phi(s)^T \end{pmatrix}$$

上記の z と H_s をエージェントが行動を実行するごとに求め，式 (5)〜(9) を使用し，パラメータを更新する．

4.2 政策を利用した Q 学習

推定した政策を利用して強化学習を行うため，マルチエージェントに拡張した Q 学習を使用する．環境を共有するエージェント数を N とする．i 番目のエージェントが使用する Q 学習の式を以下に示す．

$$Q_i(s_i, a_i, a_{-i}) \leftarrow Q_i(s_i, a_i, a_{-i})$$

$$+\alpha[r+\gamma \max_{a'_i, a'_{-i} \in A(s')} Q_i(s'_i, a'_i, a'_{-i}) - Q_i(s_i, a_i, a_{-i})]$$

ここで a_{-i} は i 番目のエージェントを除いた，$(N-1)$ 体のエージェントが選択した行動の組み合わせである．エージェント数を i, j の 2 体とすると，上記の Q 学習の式は以下になる．

$$Q_i(s_i, a_i, a_j) \leftarrow Q_i(s_i, a_i, a_j)$$
$$+\alpha[r+\gamma \max_{a'_i, a'_j \in A(s')} Q_i(s'_i, a'_i, a'_j) - Q_i(s_i, a_i, a_j)] \quad (12)$$

これ以降，式 (12) を使用して説明を行う．式 (12) で Q 学習を行うには，状態 s'_i において，エージェント j の行動を推定する必要がある．以下に，エージェント j の行動の推定方法を示す．

(1) $A = H_{s'}\theta$ を計算
(2) ベクトル A の要素の中から最大値 a を計算
(3) 最大値 a に対応するベクトル A の次元数 d を計算
(4) 次元数 d を返す

行動は 1-of-K 表現を使用しているため，ベクトル A の各次元がエージェントの行動と対応している．そのため，ベクトルの各要素の最大値を計算し，その最大値に対応する次元数を返すことが行動の推定に対応する．

エージェント i が行動選択するとき，ある状態において最大の Q 値の行動を選択する．状態 s_i の最大の Q 値 : $Q_i(s_i, a_i, a_j)$ を求めるには a_j が必要である．a_j を求めるには，上記のエージェント j の行動の推定方法を使用する．

5. 実　　験

5.1 追 跡 問 題

実験で扱うタスクは追跡問題である．追跡問題とは，複数のハンタエージェントが獲物エージェントを捕獲するタスクである．追跡問題はマルチエージェント強化学習の標準的なタスクである．以下，ハンタエージェントをハンタ，獲物エージェントを獲物と呼ぶ．通常の追跡問題は状態と行動を離散値で表現するが，本研究では状態を連続値，行動を離散値とする．使用するフィールド は 2 次元座標空間の $(x, y) \in [0, 1]^2$ を使用し，各エージェントは点で表現される．ハンタは自分を中心とした相対座標で状態を知覚する．エージェントは上下左右停滞の 5 つの行動から選択を行う．上下左右の 4 つの行動は 0.1 にノイズを含んだ大きさで移動する．行動に含まれるノイズは $[-0.02, 0.02]$ の範囲で一様である．獲物は行動をランダムに選択する．獲物の捕獲条件は，獲物を中心とした半径 0.1 の円の中に，2 体のハンタが存在するときである．以下にハンタ 2 体，獲物 1 体のフィールドの図 2 とハンタ 1 が知覚する状態の図 3 を示す．

5.2 実 験 準 備

実験では，2 体のハンタ，1 体の獲物を使用し，追跡問題を行う．また，2 体のハンタをそれぞれ”ハンタ 1”, ”ハンタ 2”とする．各エージェントの初期位置はランダムとする．初期位置から各エージェントは同時に，状態の知覚→行動→学習 (ハンタのみ) を繰り返す．エージェントが 1 回の行動を実行したとき，

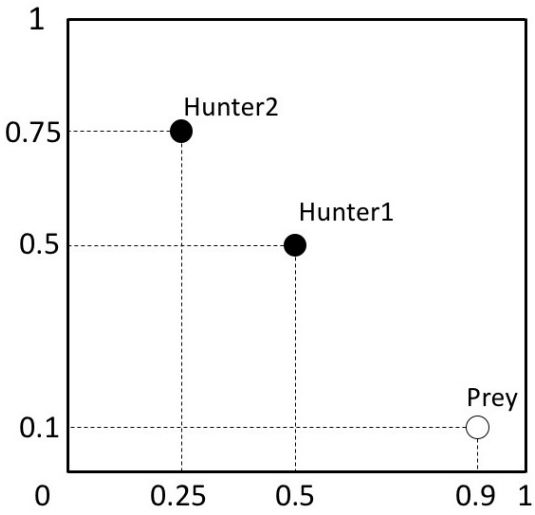

図 2　フィールド

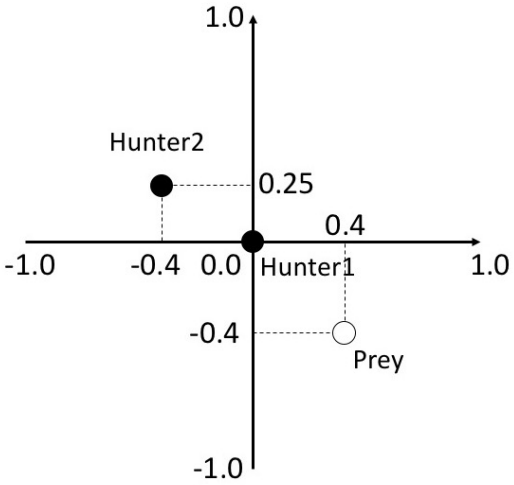

図 3　ハンタ 1 の相対位置

これを 1 ステップと呼ぶ．初期位置からハンタが獲物を捕獲するまでを 1 エピソードと呼び，10000 エピソード繰り返す．もし，ハンタが 1000 ステップで獲物を捕獲できなければ，次のエピソードに移る．10000 エピソード繰り返しを 1 セットと呼び，10 セット実行して平均を取る．学習率は 0.001, 割引率は 0.9 に設定する．報酬は獲物を捕獲したとき 100, それ以外は-0.1 である．行動選択手法として ε -greed 法を用いる． ε の値は学習の経過によって小さくするため， $\varepsilon = 0.3 *$ (総エピソード − 現在のエピソード)/総エピソード を使用する．このように ε を設定することで，エピソード初期は探索の行動を多く行い，後期はグリーディな行動を多く実行する．特徴ベクトルは，1 つの定数と状態の次元ごとに 3 つのガウスカーネルを使用する．ガウスカーネルの中心は $\{-1, 0, 1\}$, 標準偏差を 0.5 とする．したがって，特徴ベクトルの次元は $(1 + 3 \times 4) \times 5 \times 5 = 625$ である．評価方法は，500 エピソードごとの捕獲までのステップ数（捕獲ステップ）の区間平均を使用する．提案手法のカルマンフィルタの初期値は，予測共分散を $10^{10}I$, プロセスノイズの共分散を $0.01I$, 観測ノイズの共分散を $0.999309463^{\text{現在のエピソード}}I$

とする．ここで，$0.999309463^{10000} \approx 0.0001$ である．

比較手法は，政策推定にパーセプトロン [14] を使用するハンタである．パーセプトロンを用いる理由は，線形識別関数を使用しており，カルマンフィルタの線形性の仮定と一致しているからである．政策の推定は，エージェントの行動を状態から分類する，分類問題と考えることができる．比較手法では，政策をパーセプトロンを用いて，オンライン学習を行い，政策の推定 (分類) を行う．パーセプトロンの学習に用いる特徴量，正解ラベルは，提案手法と同じである．比較手法の学習方法は提案手法と同じである．

5.3 実験結果

捕獲ステップを表 1 に示す．提案手法は比較手法と比べ，2000 エピソードで最大 21.7%，捕獲ステップが減少し，平均で 11.7%，捕獲ステップが減少している．また，最悪の場合でも 500 エピソードで 1.4%の悪化に抑えられている．

表 1　捕獲ステップ

	政策の推定方法	
エピソード	提案手法	パーセプトロン
500	602.058	593.886
1000	421.237	470.640
1500	286.772	359.682
2000	213.868	273.007
2500	169.404	214.232
3000	151.292	176.371
3500	133.585	152.671
4000	121.561	136.994
4500	112.431	123.700
5000	104.311	117.133
5500	98.504	107.731
6000	91.683	101.751
6500	84.882	94.613
7000	80.134	90.651
7500	76.978	84.653
8000	72.976	82.296
8500	69.928	78.428
9000	67.142	75.841
9500	63.582	71.819
10000	61.469	69.185

5.4 考　　察

ハンタ 1 からハンタ 2 を推定した行動の正解率を表 2, ハンタ 2 からハンタ 1 を推定した行動の正解率を表 3 に示す．加えて，提案手法のカルマンフィルタにおける，予測共分散のフロベニウスノルムを表 4 に示す．

表 4 から，政策推定の過程が分かる．エピソード初期は，推定するハンタの政策の変化が大きいため，それに伴いノルムは大きくなっていると考えられる．すなわち，提案手法はカルマンフィルタのパラメータ推定の枠組みを利用しているので，パラメータを大きく変化させて政策を推定しようとしている．一方，エピソード後期はノルムがほぼ一定で変化していない．つまり，ハンタの政策の大きく変化しなくなり，パラメータをエピソード前期に比べ，変更する必要がないからだと考えられる．

表 2　ハンタ 1 からハンタ 2 を推定した行動の正解率

	政策の推定方法	
エピソード	提案手法	パーセプトロン
500	0.606	0.356
1000	0.561	0.359
1500	0.559	0.365
2000	0.553	0.380
2500	0.547	0.392
3000	0.535	0.405
3500	0.528	0.415
4000	0.520	0.425
4500	0.520	0.438
5000	0.519	0.444
5500	0.528	0.457
6000	0.530	0.474
6500	0.549	0.490
7000	0.558	0.503
7500	0.582	0.516
8000	0.600	0.533
8500	0.620	0.548
9000	0.647	0.565
9500	0.678	0.588
10000	0.709	0.609

表 3　ハンタ 2 からハンタ 1 を推定した行動の正解率

	政策の推定方法	
エピソード	提案手法	パーセプトロン
500	0.607	0.354
1000	0.561	0.361
1500	0.559	0.364
2000	0.553	0.377
2500	0.545	0.390
3000	0.532	0.406
3500	0.529	0.413
4000	0.521	0.424
4500	0.522	0.440
5000	0.521	0.447
5500	0.527	0.457
6000	0.528	0.473
6500	0.549	0.488
7000	0.560	0.503
7500	0.583	0.516
8000	0.599	0.534
8500	0.622	0.547
9000	0.648	0.565
9500	0.678	0.587
10000	0.707	0.609

したがって，エピソードが進むに連れ，ハンタの政策を推定できていると考えられる．また，ノルムがこれ以上小さくならない理由として，プロセスノイズを $0.01I$ に設定しているからである．

表 2, 3 の正解率を表 1 の結果に対応させると 500 エピソードを除き，エピソード初期に提案手法が効果的である．エピソード初期のハンタは探索の行動を多く行うため，同じ状態でも学

表 4　予測共分散のフロベニウスノルム

	ハンタの名前	
エピソード	ハンタ 1	ハンタ 2
500	63.990	64.004
1000	45.431	45.235
1500	37.280	37.153
2000	34.390	34.519
2500	31.773	31.838
3000	29.600	29.597
3500	27.328	27.349
4000	25.574	25.646
4500	24.248	24.261
5000	22.985	23.019
5500	21.943	22.033
6000	20.769	20.752
6500	20.459	20.385
7000	19.643	19.679
7500	19.547	19.638
8000	19.118	19.158
8500	19.114	19.022
9000	19.133	19.094
9500	19.482	19.425
10000	19.561	19.624

習状況に応じて，異なる行動を実行する．そのため，比較手法のエピソード初期の正確率は低いと考えられる．一方，提案手法ではそのような状況であっても，高い正解率である．提案手法は比較手法に比べ政策を推定できているため，学習初期に捕獲ステップが減少していると考えられる．

表 1 において，提案手法が比較手法に比べ，500 エピソードで 1.4%の悪化をしている．表 2, 3 の 500 エピソードでの提案手法の正解率は 0.606, 0.607 と比較手法に比べて高いが，表 4 の 500 エピソードでのノルム 63.990, 64.004 と大きい．つまり，500 エピソードまでに推定した政策は，タスクを解く意味で正しくない政策のため，Q 学習に活かされず捕獲ステップが悪化したと考えられる．

6. 結　　論

本研究では，マルチエージェント強化学習において，カルマンフィルタを使用した，政策推定の方法を提案した．カルマンフィルタのパラメータ推定の枠組みを応用することで，変化する政策の推定を可能とした．実験では，政策が変化するエージェントに対し，比較手法と比べ．最大 21.7%, 平均で 11.7%の捕獲ステップを削減した．これにより，マルチエージェント環境において，政策を推定し，強化学習を行えることを示した．

文　　献

[1] Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. Vol. 1. No. 1. Cambridge: MIT press, 1998.

[2] Kalman, Rudolph Emil. "A new approach to linear filtering and prediction problems." Journal of basic Engineering 82.1 (1960): 35-45.

[3] Greg Welch and Gary Bishop, "An Introduction to the Kalman Filter", TR-95-041, Department of Computer Science, University of North Carolina at Chapell Hill, updated in 2006

[4] Watkins, Christopher JCH, and Peter Dayan. "Q-learning." Machine learning 8.3-4 (1992): 279-292.

[5] Busoniu, Lucian, et al. "Approximate reinforcement learning: An overview." Adaptive Dynamic Programming And Reinforcement Learning (ADPRL), 2011 IEEE Symposium on. IEEE, 2011. APA

[6] Albus, James S. "A new approach to manipulator control: The cerebellar model articulation controller (CMAC)." Journal of dynamic systems, measurement and control 97.3 (1975): 220-227.

[7] Konidaris, George, and Sarah Osentoski. "Value function approximation in reinforcement learning using the Fourier basis." (2008).

[8] Nagayuki, Yasuo, Shin Ishii, and Kenji Doya. "Multi-agent reinforcement learning: An approach based on the other agent's internal model." MultiAgent Systems, 2000. Proceedings. Fourth International Conference on. IEEE, 2000.

[9] Powell, Michael JD. "Radial basis function for multivariable interpolation: a review." IMA Conference on Algorithms for the Approximation of Functions ans Data, 1985. RMCS, 1985.

[10] Panait, Liviu, and Sean Luke. "Cooperative multi-agent learning: The state of the art." Autonomous agents and multi-agent systems 11.3 (2005): 387-434.

[11] Busoniu, Lucian, Robert Babuska, and Bart De Schutter. "Multi-agent reinforcement learning: An overview." Innovations in multi-agent systems and applications-1 310 (2010): 183-221.

[12] Hu, Junling, and Michael P. Wellman. "Nash Q-learning for general-sum stochastic games." Journal of machine learning research 4.Nov (2003): 1039-1069.

[13] Greenwald, Amy, Keith Hall, and Roberto Serrano. "Correlated Q-learning." ICML. Vol. 3. 2003.

[14] Rosenblatt, Frank. "The perceptron: A probabilistic model for information storage and organization in the brain." Psychological review 65.6 (1958): 386.