# カルマンTD誤差を使用した多重モデル学習
# Refining Multiple Models using Kalman Temporal Differences (KTD) framework

北尾　健大†　　白井　匡人††　　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
†† 島根大学大学院 総合理工学研究科 情報システム学領域　〒 690-8504 島根県松江市西川津町 1060
E-mail: †takehiro.kitao.7j@stu.hosei.ac.jp, ††shirai@cis.shimane-u.ac.jp, †††miurat@hosei.ac.jp

**あらまし**　本研究は多重モデル学習による学習制度の向上を目的とし，このためにカルマン TD 構造を用いる．カルマン TD(KTD) とはカルマンフィルタと TD 誤差 (時間的差分) 学習を組み合わせた学習方法である．しかし, カルマンフィルタと同様に自明でない初期パラメタ依存性問題がある．この問題に対して, 複数の初期パラメタを使用して同時に学習を行い, 逐次的にモデル選択を行う手法を提案する．

**キーワード**　強化学習, カルマンフィルタ, カルマン TD 誤差

## 1. 前　書　き

近年，強化学習はシステム制御，ゲーム，人工知能など幅広い分野での応用に向けて研究が行われている [1]．強化学習では，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．その代わりに，動作主 (エージェント) は環境の相互作用から学習を行う．具体的にはエージェントが，環境に対して行動を取り状態が遷移することで，報酬と呼ばれる評価値を得る．エージェントはこの報酬が最大となるように学習する．強化学習の代表的な手法として，Q 学習や Sarsa などが挙げられる [6]．しかし，強化学習を現実問題に適用する場合，状態が速度や角度，位置といった連続値であることが多い．上記の手法は，状態が離散値である場合を想定しており，連続値に対し直接的に適用することができない．そこで，システム制御の枠組で発展した，カルマンフィルタ [3] と強化学習を組み合わせたカルマン TD(KTD) が提案されている [2]．この手法を用いることで，状態が連続である問題に対して効率的に学習が可能である．反面、カルマンフィルタに由来する自明ではない初期パラメタ依存問題が存在する．このため，適切な初期パラメタを設定しなければ，学習がうまく機能しない．KTD を使用して，効率的に学習するためにはどの様に適切なパラメタを決定するかを考慮する必要がある．しかし，パラメタの推定は容易でない。本研究では，複数のパラメタを多重に学習することでこの問題の解決を図る．

本稿では，複数の初期パラメタを同時に学習し，モデル選択を行う手法を提案する．提案手法の主な利点は，カルマンフィルタの特徴である学習過程を追うことが可能な点にある．本研究の貢献は以下の 2 つである．

(1)KTD の初期パラメタの探索コストを削減する．
(2) 適切なモデル選択によって学習精度が向上する．

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 KTD の特徴と問題点を提示し，第 5 章で提案手法について述べる．第 6 章で提案手法の有用性を論じ、第 7 章で結論とする．

## 2. 多重強化学習

### 2.1 強 化 学 習

強化学習とは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．

一般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 $\pi$ (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価値 Q(行動価値関数) を利用する [6]．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」「$\varepsilon$-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 [2], [8]．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，$\phi_s$ 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．

$$\hat{V}_{\theta(s)} = \sum_{j=0}^{p} w_j \phi_s = \phi_s^T \theta$$

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

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 [9]，フーリエ級数 [10] 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 TD 学 習

TD 学習とはオンラインで学習可能な，TD 誤差を使用した学習方法である．一般的には下記の式で表現できる．

$$\theta_i = \theta_{i-1} + K_i \delta_i \quad (2)$$

$\theta$ は価値関数の推定値，$K_i$ はステップサイズパラメータ，$\delta_i$ は TD 誤差である．TD 誤差とは 1 ステップ後の価値と現在の推定値の差である．TD 学習の代表的な手法として，2.1 章で説明した Q 学習と Sarsa が挙げられる．下記に Sarsa の更新式を示す．

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma Q(s',a') - Q(s,a)] \quad (3)$$

Q 学習と Sarsa の違いは，更新式の次状態の Q 値である．Q 学習では実際に選択した行動に関係なく，次状態で最大の Q 値を選択して更新する．一方で Sarsa は実際に取った行動に合わせて更新を行う．前者を政策オフ型，後者を政策オン型と呼ぶ．Q 学習と Sarsa のどちらを使用するべきかはタスクによって決まる．

### 2.3 多重モデル学習

提案手法の一部である多重モデル学習について述べる．多重学習とは，1 エージェントで強化学習モデルを複数同時に学習することを意味する．この手法を使う理由として，強化学習は全てパラメタが異なっており，状況に応じて適切なモデルを利用するためである．

多重学習で実行する強化学習は，政策オフ型でなければならない．政策オン型である場合，更新式と選択する行動が連動しているため学習中のモデルから一つ選択して行動を選択しなければならない．この場合，選択されたモデルの政策は改善されるが，その他のモデルは適切に改善されない．政策オフ型であれば，実際に選択した行動に関係なく，次状態で最大の Q 値を利用して更新されることから，全てのモデルの改善が行われる．以下に多重学習のイメージ図を示す．

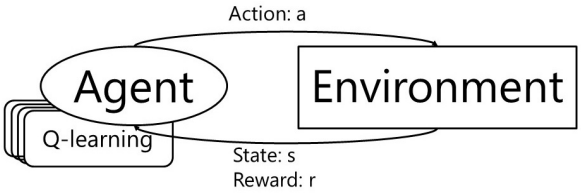


図 1　Multiple Learning

提案手法は以下の手順でモデル学習を行う．

(1) 初期パラメタを複数選択して多重学習
(2) モデルの分散から傾きを計算
(3) 傾きを用いてモデルを選択

## 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 [3], [4].

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．

$$x_i = Ax_{i-1} + w_{i-1} \tag{4}$$

$$z_i = Hx_i + v_i \tag{5}$$

ここで，式 (4) は状態方程式と呼ばれ，行列 A 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 (5) は観測方程式と呼ばれ，行列 H 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 (4) と式 (5) 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．ノイズは $p(w)$〜$N(0,Q)$, $p(v)$〜$N(0,R)$ に従うホワイトノイズである．

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

$$\hat{x}_{i|i-1} = A\hat{x}_{i-1|i-1} \tag{6}$$

$$P_{i|i-1} = AP_{i-1|i-1}A^T + Q \tag{7}$$

更新ステップ

$$K_i = P_{i|i-1}H^T(HP_{i|i-1}H^T + R)^{-1} \tag{8}$$

$$\hat{x}_{i|i} = \hat{x}_{i|i-1} + K_i(z_i - H\hat{x}_{i|i-1}) \tag{9}$$

$$P_{i|i} = (I - K_iH)P_{i|i-1} \tag{10}$$

ここで $\hat{x}_{i|i-1}$ を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 $P_{i|i-1}$ も $\hat{x}_{i|i-1}$ 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 (8) 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．カルマンフィルタのイメージ図を以下に示す．

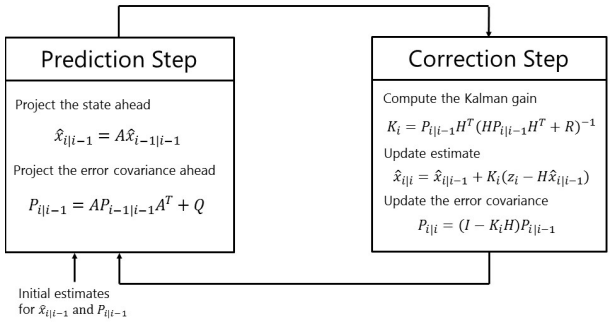


図 2　Kalman filter

## 4. カルマン TD

### 4.1 カルマン TD の基本要素と特徴

カルマンフィルタを使用して，連続値強化学習の重み $\theta$ を推定するフレームワークをカルマン TD(KTD) と呼ぶ [2]．TD 誤差を用いるアルゴリズムであれば，KTD を使用して学習が可能である．KTD には以下の 6 つの特徴がある．

(1) オンラインで学習
(2) 膨大な状態数や連続状態が扱える
(3) 可能な限り少ない学習回数で政策を獲得
(4) 非定常状態が扱える
(5) 探索と知識利用のジレンマにおける情報を利用可能
(6) Q 学習の Max 演算子のような非線形が扱える

KTD を使用するには，” プロセスノイズ” と” 観測ノイズ” の 2 つのパラメタを選択しなければならない．もし学習を行うタスクに対して，事前知識を保持しているのあれば，それに従ってパラメタを選択すればよい．しかし，実際には事前知識を保持している事はほとんどなく，パラメタの選択方法は自明ではない．そのような場合，幾つかのパラメタで実行してみて，精度が良いものを選択する．Geist [2] らはプロセスノイズにおいて，$P_{v_i} = \eta P_{\theta_{i-1|i-1}}(\eta << 1)$ を使用し，試行錯誤により全てのパラメタを決定している．プロセスノイズにおいて，本研究でも同様の方法を使用する．

### 4.2 カルマン TD-Q

ここでは TD 誤差を用いるアルゴリズムとして，Q 学習を使用した，カルマン TD-Q(KTD-Q) について述べる．KTD-Q は状態空間モデルを下記のように表現する．

$$\theta_i = \theta_{i-1} + v_{i-1} \tag{11}$$

$$r_i = \hat{Q}_{\theta_i}(s_i, a_i) - \gamma max_{b\in A}\hat{Q}_{\theta_i}(s_{i+1}, b) + n_i \tag{12}$$

ここで，$r_i$ は時刻 $i$ で獲得した報酬を表す．

カルマンフィルタと違い，カルマン TD 誤差では推定値を解析的に計算することが不可能なため，下記の線形推定器を仮定する．

$$\hat{\theta}_{i|i} = \hat{\theta}_{i|i-1} + K_i(r_i - \hat{r}_{i|i-1})$$

この線形推定器に最も適した $\theta_i$ を計算する．また，式 (12) には max 演算子がついているので，3 章で扱ったカルマンフィルタでは計算できない．そのため，非線形カルマンフィルタの一種である UKF(Unscented Kalman Filter) を使用して計算する [5]．UKF の考え方として，シグマポイントと呼ばれる少数の代表点を平均値の周りで使用し．計算を行う．

KTD-Q のアルゴリズムを下記に示す．”シグマポイントの計算” で UKF のシグマポイントを計算しており，具体的な計算方法は以下の式で求まる．

$$\hat{\theta}^{(0)}_{i|i-1} = \hat{\theta}_{i|i-1}(j = 0)$$
$$\hat{\theta}^{(j)}_{i|i-1} = \hat{\theta}_{i|i-1} + (\sqrt{(n+\kappa)P_{i|i-1}})_j(1 \leqq j \leqq n)$$
$$\hat{\theta}^{(j)}_{i|i-1} = \hat{\theta}_{i|i-1} - (\sqrt{(n+\kappa)P_{i|i-1}})_j(n+1 \leqq j \leqq 2n)$$

n は $\theta$ の次元数を表し，$\kappa$ はサンプリングの幅を調整するスカラー値である．$(\sqrt{(n+\kappa)P_{i|i-1}})_j$ は行列 $(n+\kappa)P_{i|i-1}$ をコレスキー分解をした $j$ 列目である．また，W は以下のように計算

**Algorithm 1** KTD-Q

初期値: $\hat{\theta}_{0|0}$ and $P_{0|0}$
**for** $i \leftarrow 1, 2, ...do$ **do**
観測遷移 $t_i = (s_i, a_i, s_{s+1})$
予測ステップ;
$\hat{\theta}_{i|i-1} = \hat{\theta}_{i-1|i-1}$;
$P_{i|i-1} = P_{i-1|i-1} + P_{v_i}$;
シグマポイントの計算;
$\Theta_{i|i-1} = \{\hat{\theta}^{(j)}_{i|i-1}, 0 \leqq j \leqq 2p\}$(from $\hat{\theta}_{i|i-1}$ and $P_{i|i-1}$);
$W = \{w_j, 0 \leqq j \leqq 2p\}$;
$R_{i|i-1}$
$= \{\hat{r}^{(j)}_{i|i-1} = \hat{Q}_{\theta_i}(s_i, a_i) - \gamma max_{b \in A}\hat{Q}_{\theta_i}(s_{i+1}, b), 0 \leqq j \leqq 2p\}$;
指標の計算;
$\hat{r}_{i|i-1} = \sum_{j=0}^{2p} w_j \hat{r}^{(j)}_{i|i-1}$;
$P_{\theta r_i} = \sum_{j=0}^{2p} w_j(\hat{\theta}^{(j)}_{i|i-1} - \hat{\theta}_{i|i-1})(\hat{r}^{(j)}_{i|i-1} - \hat{r}_{i|i-1})$;
$P_{r_i} = \sum_{j=0}^{2p} w_j(\hat{r}^{(j)}_{i|i-1} - \hat{r}_{i|i-1})^2 + P_{n_i}$;
更新ステップ;
$K_i = P_{\theta r_i} P_{r_i}^{-1}$;
$\hat{\theta}_{i|i} = \hat{\theta}_{i|i-1} + K_i(r_i - \hat{r}_{i|i-1})$;
$P_{r_i} = P_{i|i-1} - K_i P_{r_i} K_i^t$;
**end for**

される．

$$w_0 = \frac{\kappa}{n+\kappa}(j=0)$$
$$w_j = \frac{1}{2(n+\kappa)}(\forall j > 0)$$

## 5. 提案手法

提案手法の扱う問題として，4.1 章の KTD の初期パラメタ依存である．既存の KTD では，自明でない 2 つの初期パラメタ，プロセスノイズと観測ノイズを選択しなければならない．適切にパラメタの選択をしなければ，学習精度の悪化が考えられる．KTD では試行錯誤を重ねることで，適したパラメタを選択してきた．しかし，この選択方法では試行錯誤を行っているため，時間的コストの高さが問題点である．本稿では，2.3 章の多重モデル学習を用いて，複数の初期パラメタを同時に学習し，探索コストを削減する手法を提案する．また，学習過程に合わせて，同時に学習したパラメタを適切に選択する方法も示す．具体的なモデルの選択方法として共分散のフロベニウスノルムの変化に注目する．

モデル選択は 2 ステップに分けて行う．最初のステップでは，提案手法はモデル学習の打ち切り判定を行う．カルマンフィルタでは，常に分散の小さくなるようにカルマンゲインを計算し，フィルタリングする．常に分散は減少しているため，ある間隔での傾きは負になる．一方で，傾きが負でなければカルマンフィルタが機能しておらず，適切なパラメタでないと判断できる．そのようなモデルは学習を打ち切る．したがって，$j$ 番目のモデルの傾きを $a^j$ とすると，以下の条件になる．

$$a^j = \begin{cases} \text{学習継続} & (a^j \leqq 0) \\ \text{学習打ち切り} & (otherwise) \end{cases}$$

時間間隔を $T$ とすると，$a^j$ は以下で計算される．

$$a^j = \frac{\|P^j_{i-T|i-T}\|_F - \|P^j_{i|i}\|_F}{T} \tag{13}$$

2 つめのステップは，傾き最大となるモデルの選択である．全てのモデル対して．式 (13) に従い，傾きが求まっているので，その中から負の傾きが最大のものを選択する．ここで傾きの大きさは，間隔内での選択したパラメタの適合度を表していると仮定する．提案手法は，以上の 2 ステップを任意の間隔で行い，モデルを選択をする．

以下の分散の傾きの例を使用して，モデル選択方法の具体例を示す．例では，3 つのモデルが同時に学習している．最初の

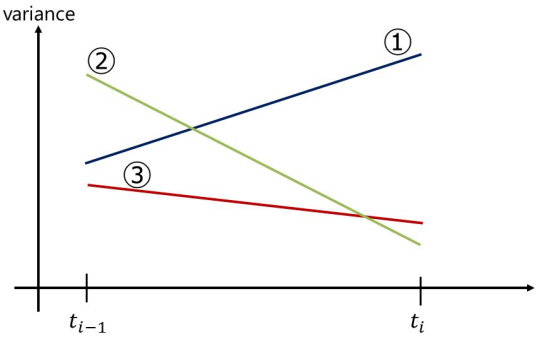


図 3　分散の傾き

ステップでは傾きの計算を行い，傾きが正であるモデルの学習を打ち切る．②と③は傾きが負であるため，学習を継続させる．一方，①は傾きが正であるため，学習を打ち切る．第 2 ステップは，モデルの選択である．ここでは，負の傾きが最大のモデルを選択する．したがって，選択されるモデルは③である．選択されなかった②も学習は継続して行う．

## 6. 実　　験

### 6.1 実験準備

本章では，提案手法の有効性を実験で確認する．学習の経過に合わせ，適切にモデル選択が行われているか，一定期間ごとに学習を停止し，選択されたモデルの評価する．既存の KTD と比較することで，モデルの選択の有効性を確かめる．実験では倒立振子問題を扱う [2], [7]. 倒立振子問題とは，長さと質量が未知の振り子に取り付けられているカートに，力を加えて直立位置でバランスをとる問題である．この問題では right force (+50), no force (0), left force (-50) の 3 つの行動が選択できる．全ての行動には [-10, 10] の範囲で一様にノイズが含まれる．状態空間は連続であり，角度 $\varphi$ と角速度 $\dot{\varphi}$ で表現される．状態遷移は選択した行動と力学に基づき以下の式に従う．

$$\ddot{\varphi} = \frac{gsin(\varphi) - \beta ml\dot{\varphi}^2 sin(2\varphi)/2 - \beta cos(\varphi)a}{4l/3 - \beta mlcos^2(\varphi)}$$

ここで $g$ は重力加速度 ($g = 9.8m/s^2$), $m$ は振り子の重量 ($m = 2.0kg$), $M$ はカートの質量 ($M = 8.0kg$), $l$ は振り子の長さ ($l = 0.5m$), $\beta = \frac{1}{m+M}$ である．角度が $[-\frac{\pi}{2}, \frac{\pi}{2}]$ の範囲内であれば報酬は 0, それ以外は-1 を与える．倒立振子の図を下記

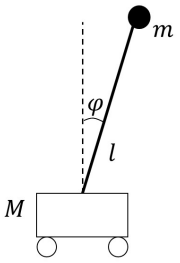


図 4　Inverted pendulum

に示す．

特徴ベクトルには 1 つの定数と 9 つのガウスカーネルを各行動ごとに使用する．ガウスカーネルの中心は $\{-\frac{\pi}{4}, 0, \frac{\pi}{4}\} \times \{-1, 0, 1\}$, 標準偏差を 1 とする．よって，特徴ベクトルの次元は $(1+9) \times 3 = 30$ になる．また割引率 $\gamma$ は 0.95 に設定する．

提案手法と比較手法は共に KTD-Q を使用し，共通の初期パラメタとして,$P_{0|0} = 10I$, $\theta_{0|0} = 0$ に設定する．提案手法では同時に学習するモデルの数を 9，初期パラメタの組み合わせをそれぞれプロセスノイズ $\eta$ と観測ノイズ $P_{n_i}$ とすると $\{0.0, 0.01, 0.1\} \times \{1.0, 0.5, 0.1\}$ である．比較手法では 3 パターンのパラメタを使用する．第 1 に Geist&Pietquin [2] らが使用していた初期位置パラメタ，プロセスノイズの $\eta$=0, 観測ノイズ $P_{n_i}$ =1.0 である．第 2 にプロセスノイズ $\eta$=0.01, 観測ノイズ $P_{n_i}$ =1.0，第 3 にプロセスノイズ $\eta$=0.1, 観測ノイズ $P_{n_i}$ =1.0 である．

振り子の初期位置として $(\varphi, \dot{\varphi})$=(0, 0) を基準にして，微小な値のノイズを角度，角速度に加えた位置とする．初期位置から報酬-1 を獲得するまでを 1 エピソードと呼び，それを 300 回繰り返す．これを 1 セットと呼び，100 セット実行して平均を取る．行動はランダムで選択し，学習を行う．25 回ごとに学習を停止し，現在まで政策の精度を評価する．その際にエピソードが終了するまで，行動選択はグリーディ手法を使用する．1 エピソードの最大数は 3000 とする．これは 5 分間失敗せずバランスを保つことができたことを意味する．また提案手法と比較手法では，学習中は同じエピソードを学習する．

### 6.2 実験結果

25 回ごとの精度を表 1 に示す．Geist&Pietquin らが使用していたプロセスノイズ $\eta$=0, 観測ノイズ $P_{n_i}$ =1.0 の場合と比較して，25 エピソード時点で最大 158.6%学習精度が向上しているのが分かる．また最悪の場合でも 225 回の時点で 3.4%の精度悪化で抑えられている．$\eta$=0.01, $P_{n_i}$ =1.0 の場合，50 エピソードまで順調に学習しており，50 エピソード時点で比較手法に比べ，24.6%精度が向上してる．しかし 50 エピソード以降，学習が進むに連れ，学習精度が向上しなくなっている．$\eta$=0.1, $P_{n_i}$ =1.0 の場合，全体を通して学習がうまくできていない．

### 6.3 考　　察

比較手法 3 つの分散の変化を表 2 に示す．プロセスノイズが大きいほど，学習が進むに連れ，分散が大きくなっている．表

表 1　学 習 精 度

| | $\eta$ of process noise/observation noise | | | |
|---|---|---|---|---|
| episode | 0.0/1.0 | 0.01/1.0 | 0.1/ 1.0 | 提案手法 |
| 25 | 587.2 | 1416.8 | 531.3 | 1518.3 |
| 50 | 1316.0 | 2043.1 | 527.4 | 1639.9 |
| 75 | 1708.3 | 1845.1 | 472.4 | 1895.6 |
| 100 | 1928.6 | 1845.4 | 584.7 | 1926.4 |
| 125 | 1940.5 | 1845.8 | 599.8 | 1888.1 |
| 150 | 2043.8 | 1600.0 | 376.0 | 2164.3 |
| 175 | 2252.3 | 1574.0 | 556.4 | 2240.9 |
| 200 | 2396.2 | 1852.9 | 432.9 | 2340.2 |
| 225 | 2338.0 | 1851.0 | 516.4 | 2258.9 |
| 250 | 2400.9 | 1814.8 | 331.0 | 2387.4 |
| 275 | 2329.9 | 1834.1 | 842.3 | 2347.3 |
| 300 | 2282.3 | 1835.0 | 775.2 | 2302.8 |

1 の学習結果と照らし合わせると，分散の値の大きくなる幅が大きいほど学習精度が低下するのが分かる．一方で，学習が進むに連れ分散が減っているものは，精度が向上している．このように，プロセスノイズにより学習精度に差が出る理由として，式 (11) の状態方程式で倒立振子問題の状態遷移を正しく表現できていないからだと考えられる．

表 2　分　　散

| | $\eta$ of process noise/observation noise | | |
|---|---|---|---|
| episode | 0.0/1.0 | 0.01/1.0 | 0.1/ 1.0 |
| 25 | 30.2 | 354.7 | $122.0 \times 10^{9}$ |
| 50 | 26.9 | 2004.9 | $79.0 \times 10^{15}$ |
| 75 | 25.0 | 4270.4 | $7.7 \times 10^{21}$ |
| 100 | 23.4 | 4659.6 | $526.2 \times 10^{30}$ |
| 125 | 22.2 | 4758.8 | $38.4 \times 10^{42}$ |
| 150 | 21.1 | 4502.8 | $1.4 \times 10^{54}$ |
| 175 | 20.1 | 4920.4 | $617.5 \times 10^{63}$ |
| 200 | 19.3 | 5074.5 | $154.2 \times 10^{75}$ |
| 225 | 18.6 | 4896.1 | $51.1 \times 10^{87}$ |
| 250 | 17.9 | 4735.4 | $3.9 \times 10^{99}$ |
| 275 | 17.3 | 4806.4 | $2.0 \times 10^{111}$ |
| 300 | 16.7 | 4550.7 | $331.6 \times 10^{120}$ |

提案手法によって選択されたモデルを表 3 に示す．表から 3 種類のモデルしか選択されていないことが分かる．その他のモデルは分散が大きくなって打ち切られている．選択されたモデルを注目すると，3 つのモデルで選択されやすい範囲がある．$\eta$=0.0, 観測ノイズ $P_{n_i}$ =1.0 の場合，学習初期に選択され，$\eta$=0.0, $P_{n_i}$ =0.5 が学習中期，$\eta$=0.0, $P_{n_i}$ =1.0 は学習後期に多くなっている．このことから観測ノイズは学習精度に大きく関わっており，学習が進むに連れ，値を大きくすると学習精度が向上することが分かる．

## 7. 結　　論

本研究では，カルマン TD を使用した，複数のパラメタを同時に学習する多重モデル学習手法を提案した．多重モデルで学習することで，カルマン TD で問題であった自明でない初期

パラメタを探すコストが削減された．実験では，比較手法と比べ，学習初期に最大 158.6%精度が向上することを示した．また，プロセスノイズを大きく設定すると，分散は学習が進むにつれ精度が悪化する．加えて，観測ノイズは学習の進行に合わせて，次第に大きくすることで学習精度が向上する．

## 文　　献


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

[2] Geist, Matthieu, and Olivier Pietquin, “Kalman temporal differences.” Journal of artificial intelligence research 39 (2010): 483-532.

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

[4] 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

[5] Julier, Simon, Jeffrey Uhlmann, and Hugh F. Durrant-Whyte. ”A new method for the nonlinear transformation of means and covariances in filters and estimators.” IEEE Transactions on automatic control 45.3 (2000): 477-482.

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

[7] Lagoudakis, Michail G., and Ronald Parr, ”Least-squares policy iteration.” Journal of Machine Learning Research 4.Dec (2003): 1107-1149.

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

[9] 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.

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


表 3　モデル選択

| | $\eta$ of process noise/observation noise | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| episode | 0.0/1.0 | 0.01/1.0 | 0.1/1.0 | 0.0/0.5 | 0.01/0.5 | 0.1/0.5 | 0.0/0.1 | 0.01/0.1 | 0.1/0.1 |
| 25 | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0 | 0 |
| 50 | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0 | 0 |
| 75 | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0 | 0 |
| 100 | 0 | 0 | 0 | 8 | 0 | 0 | 92 | 0 | 0 |
| 125 | 0 | 0 | 0 | 60 | 0 | 0 | 40 | 0 | 0 |
| 150 | 0 | 0 | 0 | 94 | 0 | 0 | 6 | 0 | 0 |
| 175 | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 0 |
| 200 | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 0 |
| 225 | 1 | 0 | 0 | 99 | 0 | 0 | 0 | 0 | 0 |
| 250 | 7 | 0 | 0 | 93 | 0 | 0 | 0 | 0 | 0 |
| 275 | 15 | 0 | 0 | 85 | 0 | 0 | 0 | 0 | 0 |
| 300 | 44 | 0 | 0 | 56 | 0 | 0 | 0 | 0 | 0 |