モデル予測経路積分制御と深層経路コスト予測器による高次元観測モデルベース強化学習

Model-based RL with High Dimensional Observations using MPPI and Deep Path-cost Predictor

○学　権　裕煥 (関西大院)　　学　鶴峯義久（奈良先端大）
正　本仲君子　三好誠司（関西大）　　正　松原崇充（奈良先端大）

Yuhwan KWON, Kansai University, k921940@kansai-u.ac.jp
Yoshihisa TSURUMINE, Nara Institute of Science and Technology, tsurumine.yoshihisa.tm6@is.naist.jp
Kimiko MOTONAKA, Kansai University, motonaka@kansai-u.ac.jp
Seiji MIYOSHI, Kansai University, miyoshi@kansai-u.ac.jp
Takamitsu MATSUBARA, Nara Institute of Science and Technology, takam-m@is.naist.jp

In this paper, we propose a model-based reinforcement learning framework combining Model Predictive Path Integral (MPPI) with a Deep Path-cost Predictor that outputs a state-trajectory cost given an image sequence and a control input sequence as input. We validate the effectiveness of the proposed method by carrying out 2DOF robot arm reaching tasks with multiple targets in simulation.

Key Words: Model Predictive Path Integral, Reinforcement Learning, Deep Learning

1　はじめに

強化学習の一つのアプローチとして，初めにモデルを学習しそのモデルを用いて方策を更新するモデルベース強化学習が存在する．モデルベース強化学習はサンプル効率の良い学習が可能となるが，モデルを陽に学習するために状態空間が低次元な変数の場合に限定される [1]．画像のような高次元の状態を扱うアプローチとしては，高次元なデータから潜在的なモデルを抽出し，そのモデル上で制御を行うアプローチ [2] も検討されている．

本研究では，高次元状態をもつ強化学習問題に対して適用可能なモデルベース強化学習手法を検討する．具体的には，モデル予測制御 (MPC) の一手法である，モデル予測経路積分制御 (MPPI)[3] に注目する．この手法は，生成した多数の経路を評価し最適な制御入力を計算する．MPPI において状態遷移モデルは経路の評価のために利用されるものの，初期状態から経路のコストを直接予測するモデルが構築できれば，状態遷移モデルを陽に構築する必要はない．本研究ではこの特徴に着目し，モデル予測経路積分制御と深層経路コスト予測器を用いたモデルベース強化学習手法を提案する．また，複数の目標が存在する 2 リンクアームのリーチングタスクに提案手法を適用することで，強化学習問題における提案手法の有効性を検証する．

関連研究 [4] では，Convolutional Neural Network (CNN) を用いて車載カメラ画像からコストマップを生成することで，MPPI による画像入力も用いた制御を実現しようとしているが，CNN によるコストマップの生成と制御タスクは完全に分離されており，動力学モデルについても別の手段で学習済みとしている．よって，高次元状態をもつ対象には不向きである．

2　モデル予測経路積分制御

モデル予測経路積分制御 (MPPI) は経路積分制御を反復的に実行可能にした手法で，制御周期ごとに多数の経路を生成，評価することによって最適な制御入力を求める．時刻 t における状態を $\mathbf{x}_t$，制御入力を $\mathbf{u}_t$ とすると，MPPI では時不変かつ非線形なモデル $\mathbf{F}$ を用いて

$$\mathbf{x}_{t+1} = \mathbf{F}(\mathbf{x}_t, \mathbf{v}_t) \tag{1}$$

と状態遷移が決定的に与えられると仮定する．ここで，$\mathbf{v}_t$ は $\mathbf{u}_t$ に $\mathcal{N}(0, \Sigma)$ に従うようなノイズが加わったものである．

MPPI は，並列に生成された $\mathbf{v}$ を反復的に式 (1) に適用することで求まる多数の状態軌道を評価するので，コスト関数は

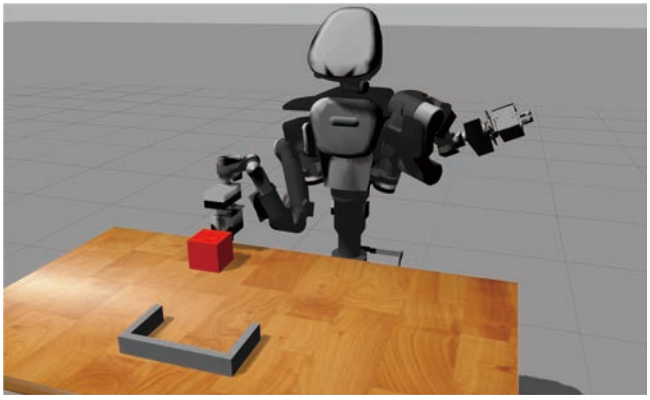


Fig.1 Task to push the red box into the gray goal on the Gazebo simulator

$$C(\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_T) = \sum_{t=1}^{T-1} q(\mathbf{x}_t) + \phi(\mathbf{x}_T), \tag{2}$$

のように軌道のコストとして考える．ここで，q は状態依存のコスト関数であり，ϕ は終端コスト関数である．また，T は生成する軌道の長さを表しており，これは加えられた制御入力系列 $U = (\mathbf{u}_0, \mathbf{u}_1, \cdots, \mathbf{u}_{T-1})$ の長さでもある．

MPPI では，制御入力系列 U を開ループ系列とした $V = (\mathbf{v}_0, \mathbf{v}_1, \cdots, \mathbf{v}_{T-1})$ の分布 $\mathbb{Q}$ と，式 (2) で表されるコストが最小となるような最適な分布 $\mathbb{Q}^*$ の KL 距離を最小化することで，最適な制御入力系列が求まると考える．ゆえに，$\mathbb{Q}$ の確率密度関数 $\mathbf{q}(V)$ の平均である U を更新して，KL 距離を最小化する．

最適な制御入力 $\mathbf{u}_t^*$ は，$\mathbb{Q}^*$ の確率密度関数を $\mathbf{q}^*(V)$ とすると

$$\mathbf{u}_t^* = \int \mathbf{q}^*(V)\mathbf{v}_t \mathrm{d}V. \tag{3}$$

のように最適な分布の期待値として得られるものの，$\mathbf{q}^*(V)$ の正規化定数を計算することができないため，重点サンプリングを用いて $\mathbf{q}(V)$ から計算される．

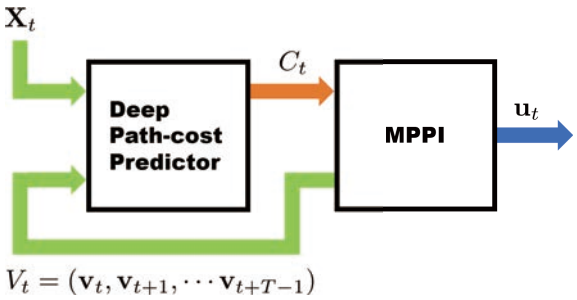


Fig.2 Input/output diagram of MPPI and Deep Path-cost Predictor. The color of the arrow represents the time relationship, after the C_t is output from Deep Path-cost Predictor, the $\mathbf{u_t}$ is calculated by MPPI.

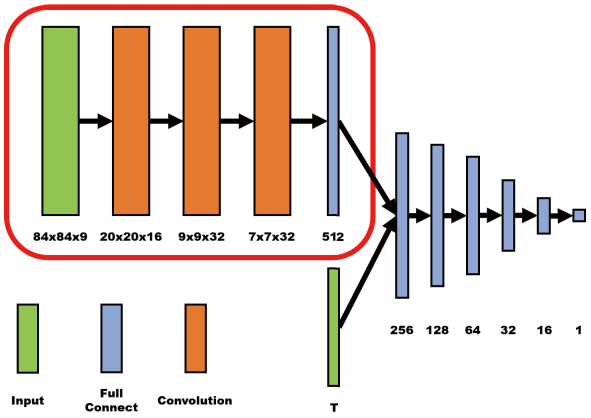


Fig.3 Network structure of Deep Path-cost Predictor

重点サンプリングの重みを w，経路の生成に用いたノイズ系列を $\mathcal{E}$ とすると，U の更新則は

$$U^{(i+1)} = U^{(i)} + \sum_{k=1}^{K} w_k \mathcal{E}_k \tag{4}$$

となる．ここで，右肩の (i) は更新回数を表し，添え字の k は生成した各経路を，K は経路の総数を表している．

3　提案手法

本研究では，MPPI で必要となる式 (1) のモデルと式 (2) のコスト関数を統合した深層経路コスト予測器と MPPI を組み合わせた，高次元観測モデルベース強化学習手法を提案する．時刻 $t-2$ から t までの観測画像系列を　$\mathbf{X}_t = (\mathbf{x}_{t-2}, \mathbf{x}_{t-1}, \mathbf{x}_t)$，制御入力系列 $U_t = (\mathbf{u}_t, \mathbf{u}_{t+1}, \cdots, \mathbf{u}_{t+T-1})$ にノイズを加えたものを $V_t = (\mathbf{v}_t, \mathbf{v}_{t+1}, \cdots, \mathbf{v}_{t+T-1})$，軌道のコストを C_t とすると，深層経路コスト予測器と MPPI の入出力関係は図 2 となる．ここで，状態を $\mathbf{X}_t$ と複数枚の観測画像で表しているのは，画像から速度情報を取得するためである．図 2 では，まず $\mathbf{X}_t$ と V_t が深層経路コスト予測器に入力されることで C_t が返される．次に，その C_t を用いて MPPI が U_t を更新し，時刻 t における制御入力 $\mathbf{u}_t$ を環境へ出力する．

提案手法の処理手順は，以下のようになる．

1. 制御入力系列を $\mathcal{N}(0, \Sigma)$ として，初期データセットを収集
2. データセットを使って，深層経路コスト予測器をトレーニング
3. 学習した深層経路コスト予測器を使って MPPI を実行し，U を更新
4. 更新した U を用いて収集したデータを，データセットに追加
5. 2. から 4. を繰り返す

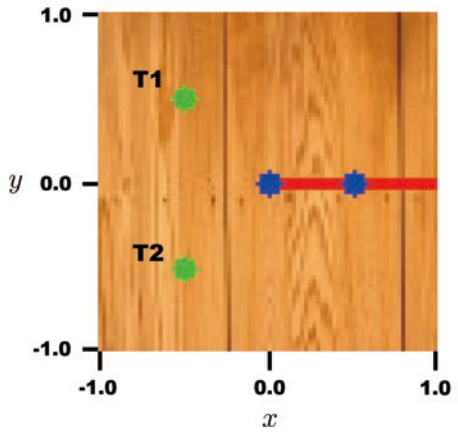


Fig.4 2DOF robot's arm used for the leaching task. The green dots **T1** and **T2** represent the target position of the hand.

ここで，3. においては MPPI を複数回実行することで，U の更新を行っている．2. で学習した予測器を用いて MPPI を 1 回実行することをトライアル，トライアルを複数回実行して U を更新することをテスト，2. の予測器の学習と 3. のテストを合わせたものをイテレーションと便宜上呼ぶこととし，テストごとに U を初期化，イテレーションごとに深層経路コスト予測器の重み以外を初期化するものとする．なお，学習用データセットの集め方については (目標の数)×(トライアルの長さ) となるよう，テストの初めのトライアルのデータだけを用いることとする．

深層経路コスト予測器のネットワーク構造については，図 3 のように二つの入力層を持っており，カラー画像系列 $\mathbf{X}$ はサイズ 84×84×9 の入力層から，V はサイズ T の入力層から入力される．ここで，MPPI の経路生成においては図中の赤枠の部分を同じ $\mathbf{X}$ が経路数 K だけ流れるため，処理時間が大幅に増加する恐れがある．それゆえ，生成時には赤枠の部分のバッチ数を 1 として処理した後に K だけ重ねて，後方の全結合層に渡す処理を行う．

4　シミュレーションによる検証

4.1　シミュレーションの設定

提案手法の有効性を検証するために，図 1 に示したタスクの前準備として図 4 に示すような 2 リンクアームのリーチングタスクを行った．ここで，2 リンクアームは青い点と赤の棒で構成されており，手先の目標位置を緑の点で表した．使用する画像は提案手法で述べたように大きさが 84×84 のカラー画像で，制御入力は各関節の角度としている．また，長さ T の状態軌道のコスト C はアームの手先座標を (x_t, y_t)，目標位置の座標を (x_d, y_d) とすると

$$C = 1000 \sum_{t=1}^{T} (|x_t - x_d| + |y_t - y_d|) \tag{5}$$

と定義した．本実験では，目標座標を $\mathbf{T1} = (-0.5, 0.5)$, $\mathbf{T2} = (-0.5, -0.5)$ の二通り準備し，テストごとに目標座標が交互に切り替わるものとしている．なお，今後予定している Gazebo シミュレータ上でのシミュレーションや実機実験も考慮して，背景を木目調の複雑なものとしている．

4.2　シミュレーションの結果

まず，深層経路コスト予測器の学習を行うために初期データセットを集めた．データセットは 1 トライアルあたり 200 ステップとし，30 トライアル分のデータを用意した．データセットの各データは，3 ステップ分の観測画像系列と長さ T の制御入力系列および式 (5) で示した状態軌道のコストの組となっている．ここで，初期データセットの制御入力については，各関節に $\mathcal{N}(0, 0.01)$ に従うようなノイズを加えたものを用いた．なお，学習時のオプ

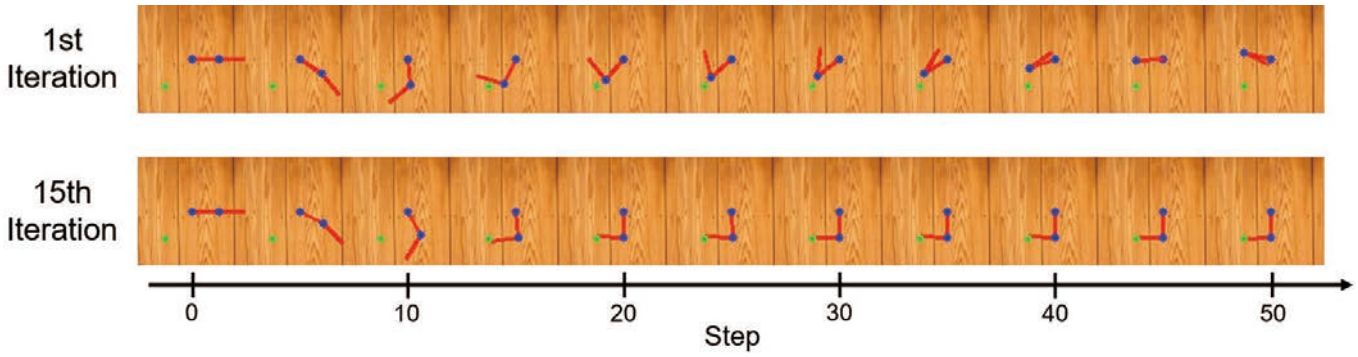


Fig.5 Controlled behaviors of 2DOF robot arm at the 1st and 15th iterations

ティマイザには Adam を用いており，学習率は 1×10^{-4}，エポック数は 80 とした．

次に，学習した深層経路コスト予測器と MPPI を組み合わせてリーチングタスクを実行した．MPPI のパラメータは，$T = 20$ ステップ，$K = 2000$，探索ノイズの分散については両関節ともに 5×10^{-5} とした．また，トライアルの回数は 5 回，テストの回数は目標位置ごとに 3 回，イテレーションの回数は 15 回にそれぞれ設定している．

図 5 は，目標座標を **T2** としたときのイテレーション 1 回目と 15 回目のアームの挙動を表した画像系列で，50 ステップまで 5 ステップごとに並べている．図 5 から，イテレーション 1 回目では手先を目標座標で保持するような動きは獲得できていないものの，15 回目では 15 から 20 ステップほどで手先を目標座標にて保持できていることがわかる．この結果は，イテレーションごとの平均コストの推移を表した図 6 において，平均コストがイテレーションを重ねるごとに減少していることからも確認することができる．ここで，図 6 の赤の破線は，モデルとコスト関数が既知であるとして MPPI を実行した際の平均コストの値である．また，図上の緑色の実線はリーチングタスクを 3 回実行した際の平均で，エラーバーは標準偏差を表している．

5 まとめ

本研究では，深層経路コスト予測器とモデル予測経路積分制御を組み合わせることで，高次元な画像入力から最適な制御入力を求めるモデルベース強化学習手法を提案し，複数の目標がある 2 リンクアームのリーチングタスクを実行することで，その性能を検証した．結果として，提案手法は目標が複数存在するような場合であってもリーチングタスクを実行できることが確認された．今後の予定としては，図 1 に示した複雑なタスクで性能を検証したのち，実ロボットへ提案手法を適用することを考えている．

謝辞

本研究は JSPS 科研費 JP17K06449, JP18K13782 の助成を受けたものです．

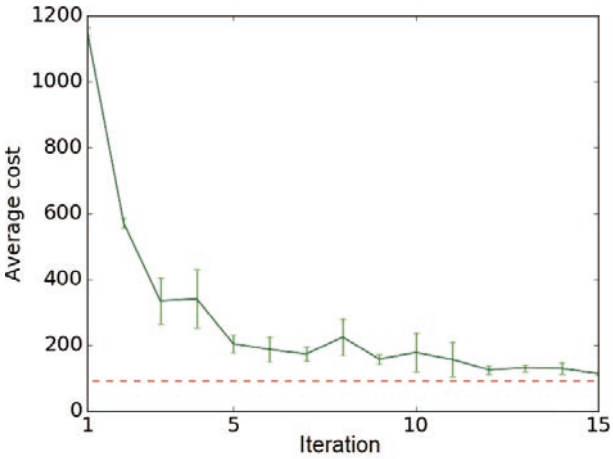
参考文献

[1] M. P. Deisenroth, D. Fox, and C. E. Rasmussen, “Gaussian processes for data-efficient learning in robotics and control”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 37, No. 2, pp. 408–423, 2015.

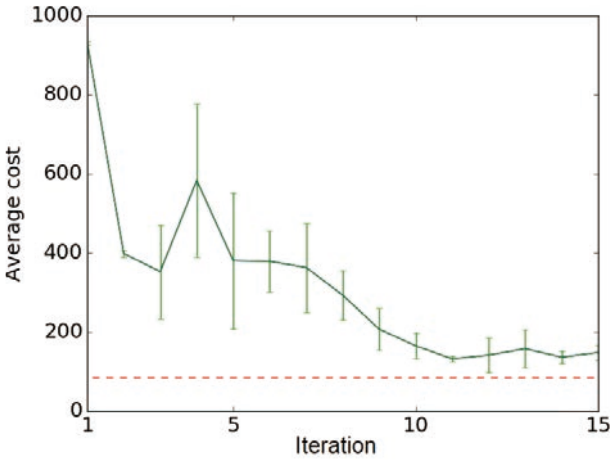
[2] 権 裕煥, 金子 拓光, 鶴峯 義久, 佐々木 光, 本仲 君子, 三好 誠司, 松原 崇充, “変分オートエンコーデッドモデル予測経路積分制御と画像入力に基づくロボット制御への応用 ”, 第 36 回日本ロボット学会学術講演会, 2E2–04, 2018.

[3] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou, “Information theoretic mpc for model-based reinforcement learning ”, In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1714–1721, 2017.

[4] P. Drews, B. Williams, G. Goldfain, E. A. Theodorou, and J. M. Rehg, “Aggressive deep driving: Combining convolutional neural networks and model predictive control ”, In *Proceedings of the 1st Annual Conference on Robot Learning (CoRL)*, pp. 133–142, 2017.



(a) The target position is $\mathbf{T_1}$



(b) The target position is $\mathbf{T_2}$

Fig.6 Transition of average cost per iteration