k -NN を用いた予測に基づく複雑な構造をもつロボットに対する制御法Control method for redundant robot using k -nearest neighbors based prediction

○ 岡留 有哉 (大阪大) 中村 泰 (大阪大)
石黒 浩 (大阪大)

Yuya OKADOME, Osaka University
Yutaka NAKAMURA, Osaka University
Hiroshi ISHIGURO, Osaka University

A robot with a large number of degree of freedoms (DOFs) may have the flexibility to cope with disturbances and irregular factors exist in a real environment. In this paper, we propose a control method for such a robot where the prediction using k -nearest neighbors from past experiences (stored instances) of a similar situation is used to determine the control signal. We apply our method to the control task of a robot with complicated structure, driven by several elastic actuators. Experimental results show that the control of a robot with a large number of DOFs can be achieved by the proposed method.

Key words: adaptive control, k -nearest neighbors, redundant robot

1. はじめに

実環境には多様な外乱や変化が存在するため、そこで活動するロボットは様々な状況に直面し、その中で適切な行動を選択する必要がある。多数の関節を持つロボットなど、物理的に自由度の大きいシステムは多様な振る舞いができるため、実環境で起こる様々な状況に対処できる可能性を持つ。一方で、モデルの複雑さ自体から生じる制御則の設計の困難さだけでなく、実環境では外乱等の外部からの影響を受けるため、前もってモデルを得ることができない。

このようなダイナミクスが与えられていないシステムに対する制御法としては、システム同定などを用いた制御対象のモデル化を行う手法が良く用いられる。例えば、状態空間を微小な領域ごとに分割し、それぞれの領域を線形モデルで近似することで、問題を解決する手法が提案されている [4]。システム同定などの制御対象のモデル化を行う手法では、制御対象のシステムが複雑になるとモデル化誤差やモデルの複雑さによるパラメータ推定の困難さが問題となる。制御対象のモデルを利用せずに制御則を学習する手法として、例えば、Actor-Critic アルゴリズム [3] や Q-Learning [8] などの強化学習がある。このような手法は試行錯誤に基づき適切な制御器の学習を行うモデルであり、低次元の状態空間を持つ比較的単純な問題については比較的容易に制御則を学習できる。しかし、多くの自由度を持つロボットの制御などで状態変数が多くなってしまうと、探索するべき組み合わせが爆発的に増加するため、現実的な時間での学習が困難となる。

本稿ではノンパラメトリック法を利用した予測に基づく制御法を提案する。すなわち、過去に観測・実行したロボットの状態と行動をペアとして保存し、それらを

直接用いて行動を決定する。ロボットは現在の状態を観測すると、保存したデータを利用し、複数の行動を選択した場合の将来の状態を予測し、それぞれの状態系列に対する評価を決める。そして、予測した系列の推定した評価に基づく行動の良さ(効用)に従い確率的に行動を決定する。この手法は制御対象のモデルを明に利用することなく、蓄積したサンプルに基づいた予測を行うことにより、適切な行動を探索・選択するものとなっており、モデル化誤差等の影響を受けない。

本稿ではまず、提案する制御手法の概要について説明し、シミュレーションでの実験結果を示す。そして最後に本制御手法の改善すべき点と今後について述べる。

2. 予測制御アルゴリズム

本研究ではシステムがマルコフ性を持つと仮定し、Markov decision process (MDP) [7]、すなわち、観測した状態に応じた行動選択の課題を扱う。制御対象を離散時間システムであると仮定するとロボットなどの物理システムのダイナミクスは状態遷移確率: $p(s'|s, a)$ で表現できる。ここで、 s , a , s' はそれぞれ現在の状態、選択した行動と次時刻での状態を示す。すなわち、MDPでは現在の状態 $s(t)$ を観測すると行動選択確率 $(p(a|s))$ に基づき行動 a を選択し、次の状態 $s(t+1)$ に遷移する。その時、状態と行動に応じた報酬 $r(s(t), a(t))$ を得る。このプロセスを繰り返す間に得られる報酬和(評価)を最大化する行動選択確率 $p(a|s)$ を求めることが課題となる。このような制御課題に対して、予測を用いた制御アルゴリズムが提案されており、例えば Model Predictive Control (MPC) [2] では制御対象のモデルを用いてシステムの将来の挙動を予測し、行動を決定する。しかし、対象のモデルが必要であるため、システム同定などによるモデル化が困難な場合には適用できない。そこで提案

手法では、制御を行う前にロボットを動作させて蓄えたデータを直接用いるノンパラメトリック法に基づき将来の状態遷移を予測する。予測した系列に対して評価を与えることで行動を決定する。この方法ではシステム同定など制御対象の明示的なモデル化を行うことなく制御を行うため、モデル化における誤差などの困難を回避することができる。

2.1 データセットの生成

ロボットのダイナミクスは前もって与えられていないが、ある行動をロボットに入力するとその状態は変化し、その変化を観測することができる。よって、ロボットに対し適当な行動系列を入力することにより、動作の系列 $\{s(1), a(1), s(2), \dots\}$ を得ることができる。この系列は、状態遷移確率に依存した系列となっており、隣り合う時刻の状態は実際のシステムから得た状態遷移の実現値となっている。

そこで、時刻 t における状態、行動および次時刻 $t+1$ における状態の組 $\{s(t), a(t), s(t+1)\}$ をインスタンス $d = \{s, a, s'\}$ とし、データセットに格納することで、インスタンスを多数集めたデータセット $D = \{d_i | i = 1, 2, 3, \dots, N\}$ を構成する。ここで、今後データセット内のインスタンスのインデックスと制御時における時刻を区別するためにインスタンスのインデックスは i で表記し、 i 番目のインスタンスを d_i 、その各成分をそれぞれ s_i, a_i, s'_i と書く。また、複数の系列からなるデータセットを用いても構わないため、データセット内のデータ数は今後 N と表記する。

2.2 動作の予測と行動の決定

状態遷移確率 $p(s'|s, a)$ と各時刻で選択する行動 $p(a|s)$ を用いると、 $s = s(t)$ において $a = a(t)$ を選択した場合にある系列 $\phi = \{s(t+1), a(t+1), \dots, s(t+\tau)\}$ が生起する確率は

$$p_\phi \propto \prod_{\tau=1}^{T-1} p(s(t+\tau+1)|s(t+\tau), a(t+\tau)) p(a(t+\tau)|s(t+\tau)), \quad (1)$$

と計算できる。この計算を行うためには、状態遷移確率と行動選択確率が必要となる。

Fig. 1 k -nearest neighbors

■状態遷移確率 本研究では状態遷移確率を

$$p(s'|s, a) = \frac{1}{K} \sum_{i \in KNN} \delta(s'_i), \quad (2)$$

のように定義する。 $\delta(\cdot)$ はディラックのデルタ関数である。 K は取得する k 近傍 (Fig. 1) の数で、 KNN は取得した k 近傍のインデックス集合である。すなわち、データセット D 内から近傍データを抽出し、状態遷移を近似する。

■行動選択確率 本研究では、行動選択確率をその行動の良さを表す効用 w を用いて以下のように定義する。

$$p(a_j|s) = \frac{1}{Z} \exp(\beta w(a_j, s)), \quad (3)$$

ただし、 $Z = \sum_n \exp(\beta w(a_n|s))$ は正規化項である。 a_j は j 番目のインスタンス d_j に含まれる行動である。ここで選択された行動 a_j を行動 $a(t)$ として出力する。

本研究では効用は一定期間内に得られる報酬の和とする [1]。すなわち、系列 ϕ の評価値を、その時得られた報酬の和:

$$\omega(\phi) = \sum_{\tau=0}^T r(s(t), a(t)) \quad (4)$$

とする。よって、行動 a_j の効用は、

$$w(a_j, s) = E[\omega(\phi^j)]_{\phi^j} = \int d\phi^j P_{\phi^j} \omega(\phi^j) \quad (5)$$

のように計算できる。ただし、 ϕ^j は時刻 t において行動 a_j を選択したすべての系列で、 P_{ϕ^j} はその系列が生成される確率である。

起こり得るすべての系列に対してこの計算を行うことで、現在の状態に応じた適切な行動を決定することができる。しかし、予測系列は無限に存在し、解析的な計算も容易ではないため、本研究では予測木を用いて計算を行う。

以下に手続きの詳細を述べる。

2.3 予測木の生成

取得したインスタンスを用いて予測木を生成する。ロボットの現在の状態 $s(t)$ を観測し、 $\tau = 0$ として Table. 1 に示す子ノード生成アルゴリズムに入力する。

このとき、生成された系列は Fig. 2 に示すように木構造で表現することができるため、この予測の系列のことを予測木と呼ぶ。

2.4 効用の計算

式 (5) の計算を前節で得た予測木を用いて計算する。 a_j を選択した場合の系列の集合は a_j を根とする部分木となる。その集合を $\{\phi_h^j | h = 1, 2, \dots, H\}$ と書くと、式 (5) は

$$w(a_j, s) \approx \sum_{h=1}^H P_{\phi_h^j} \omega(\phi_h^j), \quad (6)$$

*1 インスタンス同士の距離は $k_j = \alpha \|a_j - a_i\|_2 + \|s_j - s(t)\|_2$ で定義した。 α はスケールリングのための定数である。

Table 1 Algorithm of creating child node

1:	start
2:	入力された状態 $s(\tau)$ を用いて状態ノード作成
3:	$s(\tau)$ の近傍にあるインスタンスに 含まれる行動を重複を含まず J_a 個抽出. 距離: $\ s(\tau) - s_i\ _2$
4:	時刻 τ における行動の候補として $a(\tau) = a_j$, $j = 1, 2, \dots, J_a$ の行動ノードを作成
5:	各状態ノードに対して, $\{s(\tau), a_j\}$ の近傍*1のインスタンスを K 個抽出
6:	抽出したインスタンスから等確率で J_s 個 のインスタンスを選択
7:	選択されたインスタンス ($d_{j_s}, j_s = 1, 2, \dots, J_s$) を子ノードとして生成
8:	$\tau++$
9:	if($\tau < T$)
10:	作成された状態子ノードを 子ノード生成アルゴリズムに入力
11:	else
12:	終了
13:	endif
14:	end

Fig. 2 Predictive tree

と計算できる. 報酬の和は $w(\phi_h^j) = r(a_j, s) + \sum_{\tau=1}^T r(a_h^j(t) + \tau, s_h^j(t + \tau))$ と分解できるので,

$$w(a_j, s) \approx \sum_{h=1}^H P_{\phi^j} w(\phi_h^j) \quad (7)$$

$$= \sum_{j_s} p(s_{j_s} | a_j, s) \left[r(a_j, s) + \sum_{j'} p(a_{j'} | s_{j_s}) w(a_{j'}, s_{j_s}) \right]$$

と計算できる. ここで, $p(s_{j_s} | a_j, s) = 1/K$ かつ, 時刻 $t+T$ において $w(a, s) = r(a, s)$ なので, 予測木を再帰的にたどることにより現時刻の各行動に対する効用が計算できる.

3. Elastic Binary Manipulators の制御

本制御手法を複雑な構造を持つロボットのリーチングタスクの制御に適用した. 本研究で扱う Elastic Binary Manipulators (EBM) [6] (Fig. 3(a)) は, Binary Manipulators [5] のリンクをすべてバネダンパに置き換えたものである. EBM は多数のバネダンパのリンクがトラス状に組み合わされたロボットで, バネの自然長の長短を切り替えることで姿勢を変化させることができる.

3.1 Elastic Binary Manipulators

Fig. 3(a) に示す EBM を用いて実験を行った. 破線と実線のリンクは質量のないバネダンパ, 関節である黒点は, 質点としてそれぞれモデル化されている. また, 青点がエンドエフェクタである. 破線のリンクはバネの自然長の長短を制御器からの入力で切り替えることができる. このロボットの i 番目の質点にかかる力は,

$$\mathbf{F}_i = \sum_{j \in L_i} \left\{ -k(l_{ij} - \bar{l}_{ij}) \frac{\mathbf{s}_i - \mathbf{s}_j}{\|\mathbf{s}_i - \mathbf{s}_j\|_2} - D(\mathbf{v}_i - \mathbf{v}_j) \right\}, \quad (8)$$

となる. L_i は i 番目の関節に接続されている関節のインデックス集合である. また, $\mathbf{s}_i$ と $\mathbf{v}_i$ はそれぞれ i 番目の関節の位置と速度, l_{ij} と $\bar{l}_{ij}$ はそれぞれ i 番目の関節と j 番目の関節に接続しているバネの現在の長さや自然長である. k と D はそれぞれバネ定数とダンパ係数である.

3.2 リーチングタスク

リーチングタスクの目標はエンドエフェクタを目標位置に移動させることである. インスタンス取得時には, 広い範囲のインスタンスを取得するために, 1000 ステップに一度, 50% の確率でランダムに選択された2つのバネリンクの自然長の長短を切り替える. そして, 状態と行動を取得する操作を2000回実行した.

ゴールを $G = (0.03, 0.52)$ に設定して実験を行った. K と β はそれぞれ4と15とした. また, 4ステップ先まで予測を行うものと1ステップ先まで予測を行うもので実験を行った. これらの値はハンドチューニングで決定した. また, 本タスクでは即時的な報酬 r は

$$r = -\frac{1}{\|\mathbf{s} - G\|_2 + \epsilon}, \quad (9)$$

Fig. 3 Result of reaching task: Elastic Binary Manipulators

とした. ϵ は正数である.

Fig. 3(b) はカーネル密度推定法で可視化した, データセットに保存されている s_i の密度分布を表している. Fig. 3(c)とFig. 3(d)はリーチングタスクで得られた目標位置とエンドエフェクタの軌道との誤差のヒストグラムを示している. 予測ステップの短い場合(Fig. 3(d))目標位置のエンドエフェクタを到達させることが出来なかった. これは, 1ステップではある領域以上に目標へ近づいていくようなインスタンスを探索できないためだと考えられる. 一方, 4ステップの予測を行う場合(Fig. 3(c))この局所最適解から抜け出し, エンドエフェクタを目標位置付近に到達させることができたことがわかる.

提案手法がEBMの冗長性を利用してきているか調べるため, 追加実験として故障を含むEBMの制御を行った(Fig. 3(e)). あるリンクが破損した場合(バネリンクの自然長を固定した場合)においても, 目標位置へエンドエフェクタを到達させることができた.

4. 結論

本稿では, 制御対象の厳密なモデル化を行う必要のない, 過去の経験を利用したノンパラメトリック法を用いた予測に基づく制御手法を提案した. 本制御手法を複雑な構造を持つロボットに適用し, リーチングタスクが達成できることを示した.

リーチングのようなタスクの場合ではあまり長期的な予測は必要ではなかったが, 非線形性の強い問題を解く場合においては長期の予測が必要であると考えられる. しかし, 予測のステップ数を大きくすると計算コストが大きくなるという問題が生じる. この問題に対処するためには, 実行中のタスクに必要なデータと不必要なデータを選別するなどといった枠組みが必要になるため, データの選別法の開発は今後の課題である.

参考文献

- [1] J. Bellingham, A. Richards, and J.P. How. Receding horizon control of autonomous aerial vehicles. In American Control Conference, 2002. Proceedings of the 2002, volume 5, pages 3741 – 3746 vol.5, 2002.
- [2] Eduardo F Camacho and Carlos Bordons. Model predictive control. Springer, London, 1999.
- [3] KIMURA H. An analysis of actor/critic algorithms using eligibility traces: reinforcement learning with imperfect value function. 15th International Conference on Machine Learning, pages 278–286, 1998.
- [4] Masahiko Haruno, Daniel M. Wolpert, and Mitsuo Kawato. Mosaic model for sensorimotor learning and control. Neural Computation, 13(10):2201–2220, 2001.
- [5] D.S. Lees and G.S. Chirikjian. A combinatorial approach to trajectory planning for binary manipulators. In Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on, volume 3, pages 2749 –2754 vol.3, apr 1996.
- [6] Yuya Okadome, Yutaka Nakamura, and Hiroshi Ishiguro. Control method for redundant robot using stored instances. In International symposium on artificial life and robotics, pages 1123–1126, Jan 2012.
- [7] Martin L. Puterman. Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994.
- [8] Christopher J. C. H. Watkins and Peter Dayan. Technical note: Q-learning. Machine Learning, 8:279–292, 1992. 10.1007/BF00992698.