

冗長な構造を持つロボットに対する累積サンプルに基づく制御

A control method for a redundant robot based on weighted samples

岡留 有哉*
Yuya Okadome

中村 泰*
Yutaka Nakamura

石黒 浩*
Hiroshi Ishiguro

概要

実環境でロボットを動作させるためには、外乱やダイナミクスの変化に対応することのできる柔軟性と、そのような柔軟なロボットを制御するための制御手法とが必要となる。本論文では、そのような柔軟性を持つロボットの制御手法として、累積サンプルに基づく制御法を提案する。この手法は過去に取得した制御対象の状態・制御入力サンプルを、制御対象の現在の状態と直接比較することでサンプルに重みを付けて選択し、制御入力を選ばれたサンプルから直接決定するものである。冗長な構造を持つ多自由度ロボットである Spring Binary Manipulators (SBM) を用いたシミュレーション実験により、本制御手法でリーチングタスクが実行出来ることを示す。さらに、人間上肢型ロボティックアームのリーチングタスクに対して本制御手法を適用し、その結果を示す。

1 はじめに

実世界は複雑で、その振る舞いをすべてモデル化することは困難である。モデル化できない多様な外乱や環境の変化の存在する実世界においては、様々な状況に対処できる柔軟性を持ったシステムが必要である。物理的には系の自由度を大きくすることで様々な状況に応じた振る舞いができるため、多自由度のロボットは実環境で動かすには優れていると考えられる。一方で、自由度の大きさから状態変数が多くなり、また、その振る舞いも非線形となる事が多いため、一般に制御を行うことは困難になる。そのような線形近似では高次元の非線形系システムを十分に近似することはできない³⁾ ため、状態空間を微小な領域ごとに分割・線形化、すなわち制御対象をモデル化することで、問題を解決する手法が提案されている⁴⁾⁹⁾。このようなシステム同定などの制御対象のモデル化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。さらに、外乱の多い環境下や、環境が常に変化するような場合は、パラメータの推定が困難になったり、パラメータを再

推定する必要が生じるため適用は困難になる。

また、不確実性のある環境において制御を行う手法として、学習に基づく制御が挙げられる。例えば、強化学習の一種である Actor-Critic アルゴリズム²⁾¹¹⁾ や Q-Learning¹⁾¹²⁾ を用いることで制御則の学習を行うことができる。このような手法は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、パラメータ数が多くなった場合、すなわち状態変数が多くなってしまうと、指数関数的に計算量が増加してしまうため、現実的な時間での学習が不可能となってしまう。

そこで本研究では、過去に取得し、累積したサンプルを直接利用する制御手法を提案する。本制御手法では、過去に取得した累積サンプルに含まれる状態サンプルから制御対象の現在の状態に近いサンプルをいくつか選択することで制御を行う。提案手法では制御対象のモデル化や学習器を利用することなく制御を行う。過去のサンプルの中には制御対象のもつ入出力特性、センサの読み取り誤差等によるノイズなど、制御対象の持つシステムの挙動を示す情報が含まれている。よって、学習やシステム同定等によりシステムをモデル化しなくても、サンプルをそのまま用いて制御ができれば、同じデータを用いてモデル推定を行った制御と同様にシステムの振る舞いを考慮した制御ができると期待できる。これはパラメトリックモデルとノンパラメトリックモデルの関係と同様のものである⁶⁾。本論文ではこれらの考えに基づく単純な制御手法を提案し、その性質を議論する。

本稿ではまず、累積サンプルと現在の状態から制御入力を決定する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。次に、本制御手法を人間上肢型ロボティックアームに適用し、その実験結果を示す。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 Stored Sample-Based Control

累積サンプルに基づく制御では、はじめに制御対象を適当に動作させて動作サンプルを取得する。制御を行うときには、過去に取得した累積サンプルから、現在の状態に近いサンプルを K 個選択する。このサンプルの選択法は K 近傍法¹⁰⁾ の考えに基づいている。そして、目的に合致

*大阪大学大学院基礎工学研究科システム創成専攻, Department of Systems Innovation, Graduate School of Engineering Science, Osaka University email: okadome.yuya@irl.sys.es.osaka-u.ac.jp

した動作を行ったサンプルに対して大きな重みを与え、その重みに基づいて確率的にサンプルを選択する。選ばれたサンプルと同じ制御信号を出力する。このように累積サンプルに基づく制御手法では、制御対象のモデル化を行う代わりに、累積サンプルに対して重み付けを行うことで制御信号を決定する。初めにサンプルの取得について説明し、次に重み付けについて説明をする。

2.1 サンプル取得

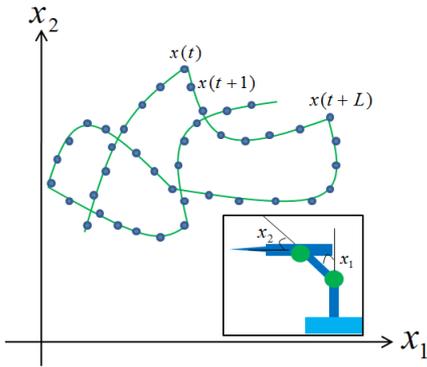


図 1: データベース中のサンプル

一般的に、ロボットなどの物理システムのダイナミクスは時刻 t でのロボットの状態 $x(t)$ と制御入力 $u(t)$ を用いて

$$x(t+1) = f(x(t), u(t)) \quad (1)$$

と表現される。(1)式において関数 f は線形か非線形かは一般には与えられていない。しかし、何らかの制御入力を与えればロボットの状態が変化し、その変化は観測することができる。真の状態は観測可能であるとは限らず、隠れた状態変数を持つ可能性があるが、本稿では観測可能な変数が制御を行うにあたって十分な情報を持つものと仮定する。そこで、ロボットに対し、ある入力を与えたとき状態を状態サンプルとして保存し、その時のを制御入力サンプルとして保存する。したがって、動作の系列 $\{x(t), u(t) | t = 1, 2, 3, \dots\}$ が得られる。そして、保存されたデータを用いて、 $d_i = \{x(t=i), u(t=i)\}$ としてデータベース D を構築する。

つまり、データベース D はそこに保存されている各サンプル d_i を用いて以下のように書くことができる

$$D = \{d_i | i = 1, 2, \dots, N\} \quad (2)$$

N はサンプル数とする。今後は d_i の状態成分と制御入力成分をそれぞれ x_i, u_i と書く。ここで d_{i+1} は i 番目のサンプル d_i の次系列のサンプルである。図 1 にサンプルとして得られた軌跡の例を示す。

2.2 重み選択による制御

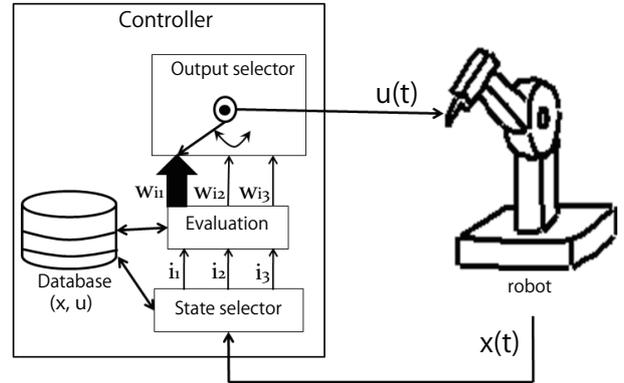


図 2: 提案手法ブロック図

図 2 は提案する制御法のブロック図である。コントローラはロボットの現在の状態 $x(t)$ を取得したのち、state-selector でデータベースにある K 個の近傍サンプルを選択する。この時、 K 個の近傍サンプルを選択する基準には、制御対象の現在の状態 $x(t)$ と状態サンプルに含まれる x_i とのユークリッド距離:

$$k_i = \|x(t) - x_i\|_2 \quad (3)$$

を用いる。選択された K 個のサンプルのインデックスを集合 KNN に加える。

選択された i 番目 ($i \in KNN$) のサンプル点から始まる運動の系列に対して重み付けを行う。重みは以下の式で定義する

$$w_i = -\min_{j < p} \{\|G - x_{i+j}\|_2\} \quad (4)$$

ただし、 G は目標の状態、 p は評価する系列の最大長とする。

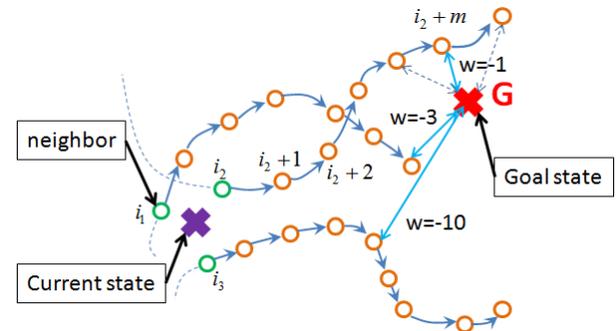


図 3: Evaluation of samples

以上のように計算した重み w_i を用い、図 2 の Output selector で使用する制御入力サンプルを選択する。 i 番目

の重みに対する選択確率 $P(i)$ を次の Soft-Max 関数⁵⁾:

$$P(i) = \begin{cases} \frac{\exp(w_i T)}{\sum_{j=1}^K \exp(w_j T)} & i \in KNN \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

で定義し、この確率に従ったルーレット選択により、サンプルを選択する。Soft-Max 関数は、逆温度 T の値を小さくしたときは $P(i)$ ($\forall i$) が $\frac{1}{K}$ に近づくため、確率的にサンプルが選択され、逆に逆温度 T の値を大きくしたときは $P(i)$ ($i = \operatorname{argmax}_i w_i$) が 1 に近づくため、最も重みの大きいサンプルが常に選択されるようになる。 i 番目のサンプル $d_i = (x_i, u_i)$ が選択されると、制御入力 u_i が出力される。表 1 にアルゴリズムの擬似コードを示す。擬似

表 1: Pseudocode of the proposed method

1:	start
2:	$(X, U) \leftarrow \text{file}(\text{dataset})$
3:	$T \leftarrow \text{const}$
4:	while(TRUE)
5:	$\mathbf{x}(t) \leftarrow \text{getposition}$
6:	for($i = 1; i < N; i++$)
7:	if $\ x_i - \mathbf{x}(t)\ _2 < \text{the smallest distance}$
8:	is added to KNN
9:	endfor
10:	for($\forall i \in KNN$)
11:	$w_i \leftarrow \left\{ \min_{j < p} (\ G - \mathbf{x}_{i+j}\) \right\}$
12:	endfor
13:	$\text{RouletteSelection}(\mathbf{W}, \text{out})$
14:	<input(<math>u_i)</input(<math>
15:	endwhile
16:	end

コード中の `getposition` でロボットの状態を取得する。また、`RouletteSelection(W, out)` では式 (5) に従って計算された確率に基づき、サンプルを KNN から一つ選択する。制御入力 $u(t)$ はルーレット選択で選ばれたサンプルの入力 u_i となる。

3 Spring Binary Manipulators の制御

累積サンプルに基づく制御法を冗長な構造をもつ制御対象の制御に適用し、提案手法の評価を行った。冗長な構造を持つシミュレータとして、二次元空間内で動作する Spring Binary Manipulators (SBM) を作成し、リーチングタスクを行った。この SBM は Binary Manipulators⁷⁾⁸⁾¹³⁾ の構造を参考としたものである。

3.1 Spring Binary Manipulators

Binary Manipulators (BM) は多数のリンクがトラス状に組み合わされたロボットで、リンクの長短を切り替え

て制御することにより姿勢を変化させる。今回作成した SBM(図 4) は、BM のリンクをバネダンパで置き換え、バネダンパが接続されている点を質点としたもので、黒点が質点、線がバネダンパである。SBM はバネダンパの自然長の長短を変化させることが可能で、一つのリンクの自然長を変化させるとそのバネダンパに接続されている質点だけでなく、系全体に影響が生じるため、通常の BM より複雑に振る舞う。また、このシミュレータは冗長な構造をもっており、一部のリンクが制御不可能となっても他のリンクの動作で壊れたリンクの動作を補うことができる。

SBM のダイナミクスはそれぞれの質点に加わる力によって決まり、質点 i に掛かる力は、

$$\mathbf{F}_i = \sum_{j \in Cnt} \left\{ -\kappa_{ij}(l_{ij} - \bar{l}_{ij}) \frac{\mathbf{s}_i - \mathbf{s}_j}{\|\mathbf{s}_i - \mathbf{s}_j\|} - D\mathbf{v}_i \right\} \quad (6)$$

となるものとした。ここで、 Cnt は質点 i に接続する質点の集合である。 κ_{ij} は i と j の間のバネリンクのバネ定数、 l_{ij} は i と j の間のバネリンクの現在の長さ、 \bar{l}_{ij} は i と j の間のバネリンクの自然長である。 \mathbf{s}_i は i の座標、 \mathbf{s}_j は i に接続する j の座標である。また、 $-D\mathbf{v}_i$ はダンパによる減衰項で、 D はダンパ係数、 \mathbf{v}_i は質点 i の速度である。式 (6) に基づいて各質点の運動を、4 次のルンゲ=クッタ法を用いて計算した。

シミュレーションでは SBM は水平面上に置かれている、すなわち重力は働かないものとする。また、各質点の質量は 1[kg]、バネ定数はすべて $\kappa = 500$ とし、ダンパ係数は $D = 0.05$ とする。

SBM ではバネダンパの基準の自然長と基準の 1.5 倍の自然長の 2 通りの自然長を切り替えることで制御を行うものとした。図 4 の SBM は 38 本のリンクで構成されている。なお、バネリンクが交差した部分は接続されていない。

図 4 の青点を手先位置とし、リーチングタスクの目標は手先位置を目標座標に移動させることである。全リンクのうち制御可能なリンクは縦方向の点線のリンク 12 本とした。

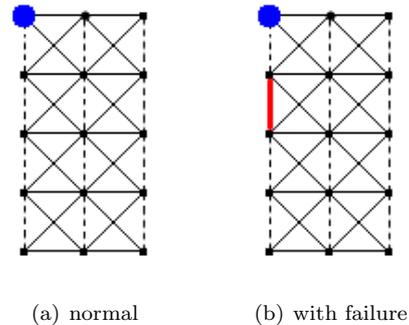


図 4: Structure of the SBM

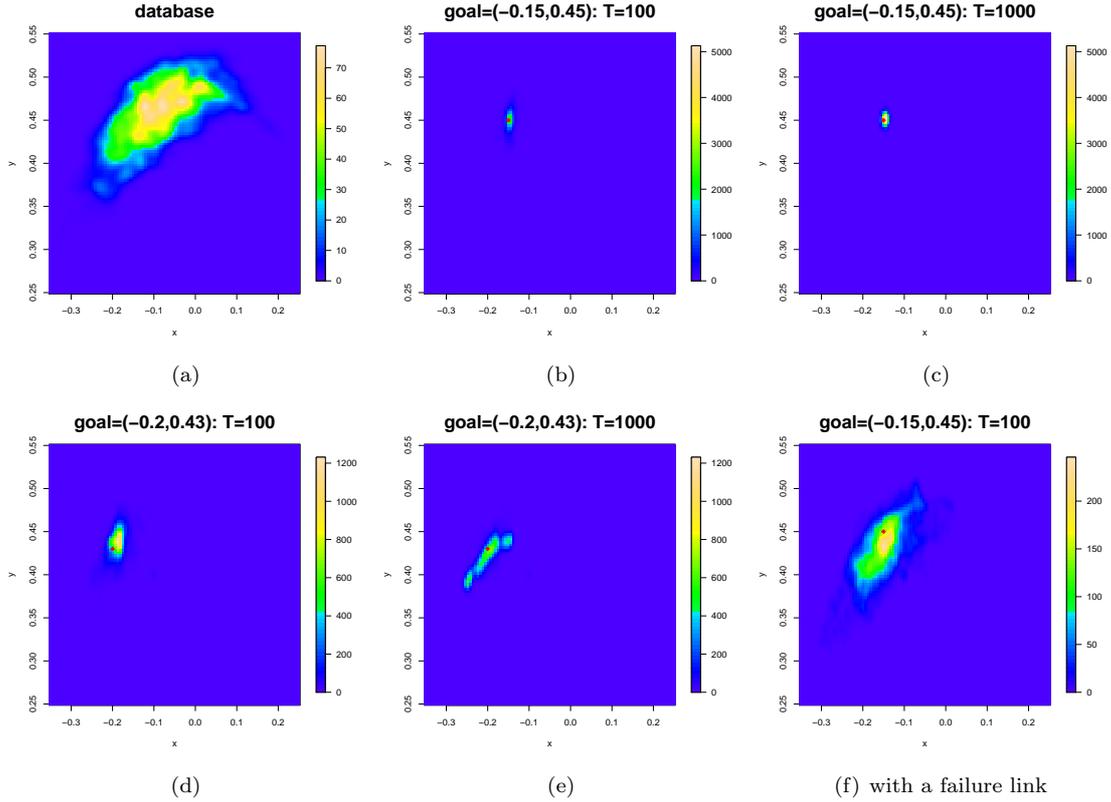


図 5: Result of reaching task: SBM

3.2 リーチングタスク

シミュレーションで行うリーチングタスクについての説明を行う。本制御手法では先に述べたように確率的に動作を行うため、SBM の手先位置が目標座標の近傍 ($\|G - x(t)\| \leq \epsilon$, ϵ は正の定数) へ到達しても動作は止まらない。そのため、規定回数の制御入力を与え、目標座標の近傍での手先軌道の密度が高くなっているかどうかを調べた。目標座標の近傍での手先軌道の密度が高くなっているとき、リーチングタスクが達成できたと考える。

制御に用いたデータベースのサンプルは 500,000 ステップ分動作させて得たエンドエフェクタの軌道から作成した。入力 $u(t)$ は各リンクの現在の自然長とする。この実験では 6 本のリンクをランダムに選択し、選択されたリンクの自然長の長短を変更することによって入力信号を生成する。また、この入力の値は 1000 ステップに 1 度更新するものとする。

3.3 SBM の制御

近傍数 K は 5 とした。実験は二つの異なる目標 ($G = (-0.15, 0.45)$ と $(-0.2, 0.43)$) と二つの異なる逆温度 ($T = 100$ と 1000) で行った。なお、逆温度 T はハンドチューニングで決定した。図 5(a) はデータベース中の状態サンプルの密度をカーネル密度推定を用いて表示したものであ

る。図 5(b), 図 5(c), 図 5(d) と図 5(e) はリーチングタスクで得られた軌道の密度を示している。なお、各図中の赤点はリーチングタスクの目標座標である。

確率的にサンプルを選択する $T = 100$ の時 (図 5(b)) に比べ、決定論的にサンプルを選択する $T = 1000$ の時 (図 5(e)) の方がより目標座標付近の動作の密度が高くなっている事が分かる。しかし、図 5(e) のように、 $T = 1000$ の場合には、目標付近を通り過ぎる軌道を持つようなサンプルが数ヶ所存在する場合には、そのようなサンプルが局所最適解となってしまう、その局所最適解を抜け出すことが難しくなる。このような局所最適解から抜け出すためには、確率的なサンプルの選択が有効であると考えられる。

3.4 一部のリンクが故障した SBM の制御

図 4(b) に示すように一部のリンクが故障した SBM の制御課題に取り組んだ。ここでは、図中の赤いリンクの自然長を変更できないものとなっている。これは提案手法が制御対象のダイナミクスの変化を扱うことができるか調べるために行った。タスクはリーチングタスクを行った。なお、データベースは故障の無い状態の SBM を動作させた時に収集したしたものを利用した。図 5(f) は故障した SBM (図 4(b)) の制御結果を示している。密度の分布は広がっているが、目標位置の付近の密度は高くなっている。すなわち、提案手法では少しのダイナミクスの変化

には対応できることがわかった。

4 人間上肢型ロボティックアームの制御

累積サンプルに基づく制御を人間上肢型ロボティックアーム (図 7) の制御に適用し、リーチングタスクを行った。この人間上肢型ロボティックアームとは人間の腕と同じように筋肉を配置した、冗長な構造を持つロボティックアームである。

4.1 人間上肢型ロボティックアーム

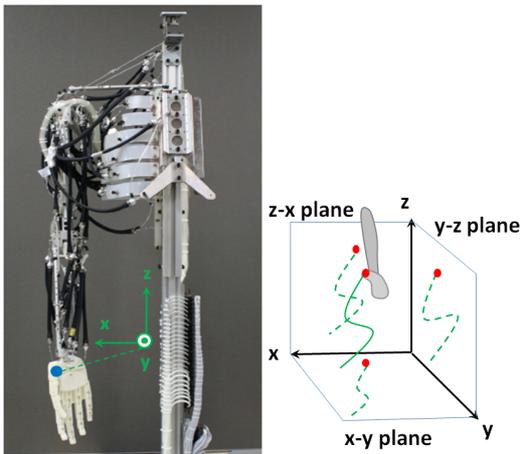


図 6: Human-Like Robotic Arm

人間上肢型ロボティックアームは 6 本のリンクとそれらに接続されている 30 本以上の McKibben 型空気圧アクチュエータで構成されている。それぞれのアクチュエータは人間の持つ筋肉の代表的なものと同様の配置で接続されている。このロボティックアームのもつ自由度は 9 と高く、また、一関節筋だけでなく多関節筋も存在する。それらが相互に複雑な影響を与えるため、どのアクチュエータがどのような動作に影響を与えるかがわからない。すなわち、モデル化が困難で、従来の制御法を適用することは困難である。

このロボティックアームはアクチュエータの伸縮を流量制御弁を用いた空気の出し入れにより制御する。流量制御弁を用いた空気の出し入れは、D/A コンバータを持つ計算機からの出力により調節することができる。また、D/A コンバータの出力は 26 次元のアナログ値の出力となっている。このロボットを制御するためには、状況に応じて 26 次元の連続値を適切に出力する必要がある。

4.2 リーチングタスク

リーチングタスクの目的はロボティックアームの手先 (図 7 の青点) を目標位置に持って行くことである。

サンプルを取得する際の入力は、26 個の流量制御弁のうち、13 個ランダムに選択し、それぞれに乱数を入力する。

流量制御弁への入力は 6 が最大値で空気の流量最大、0 が最小値で空気の流量が 0 となるため、この入力の値域は 0-6 である。入力は 1[ms] を 1[step] として制御周期 1000[step] で 1000 回与え、50Hz で動作させているモーションキャプチャーシステムで手先軌道を取得する。つまり、制御周期ごとに 50 点軌道を取得する。また、各制御周期毎に 26 次元の制御入力を取得する。得られた累積サンプルを用いて、制御周期 500[step] で制御入力を 300 回与え、リーチングタスクを行った。また、近傍数は $K = 10$ とした。実験は 2ヶ所の目標 $G = (0, 150, 140)$ と $(-40, 100, 80)$ を設定し、それぞれの目標で式 (5) の逆温度 T の値として $\frac{2}{15}$ を設定し、リーチングタスクを行った。逆温度 T の値はハンドチューニングで決定したものである。

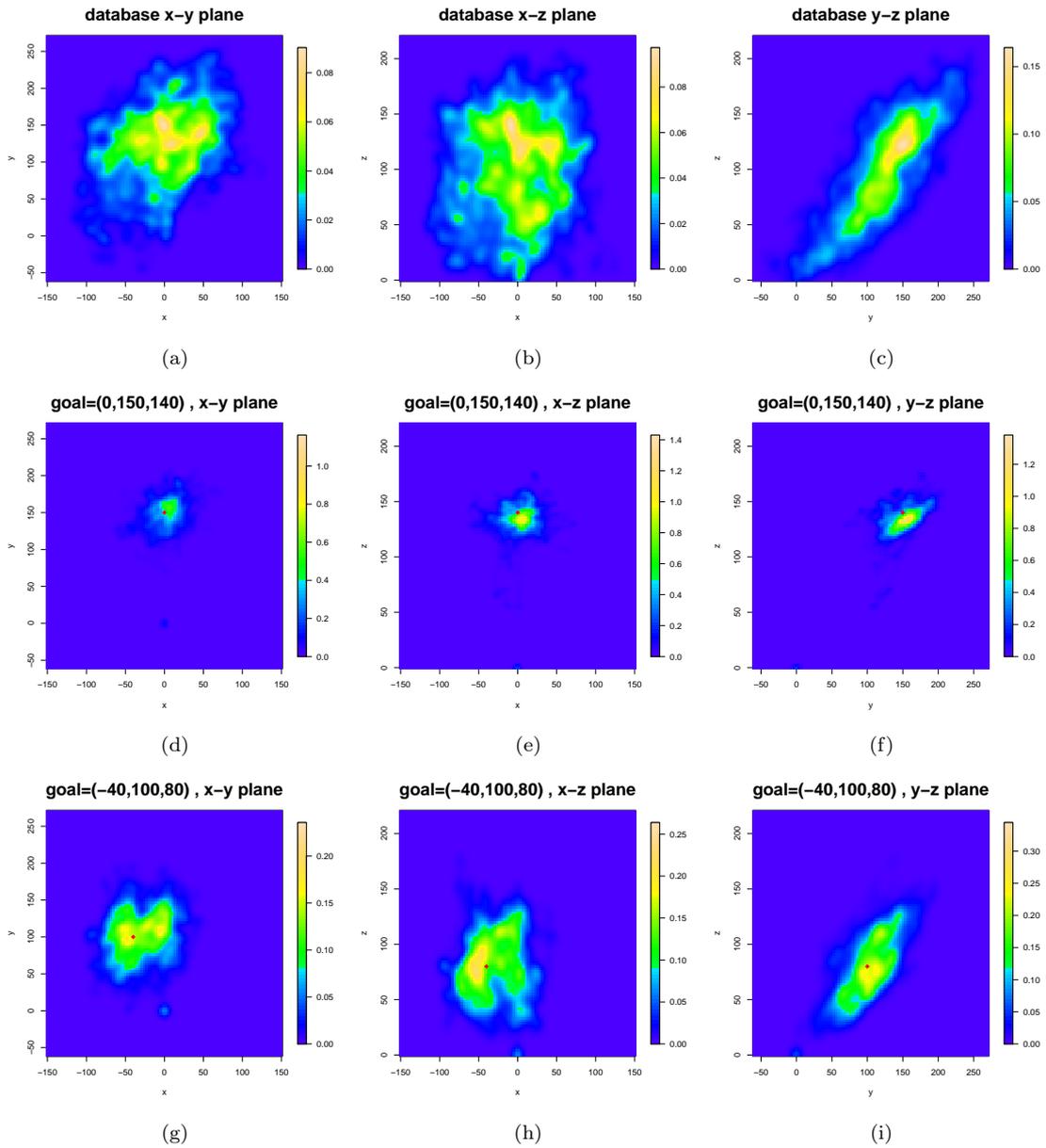
図 7(a) - 図 7(c) はデータベースの状態サンプルの密度を現している。図 7(d) - 図 7(i) はリーチングタスクで得られた手先軌道の密度を表している。それぞれの場合において目標位置近傍の密度が高くなっており、リーチングタスクを達成できたといえる。

目標位置 $G = (0, 150, 140)$ の場合では、目標位置 $G = (-40, 100, 80)$ に比べて目標付近により狭く密度が集中している。これは、利用した軌道サンプルに密度の高い位置が存在するからだと考えられる。軌道サンプルの密度の高い位置から離れた位置を目標としたときに、その密度が最も高い位置へ向かうようなサンプルが多く、結果的にそれらが選択される確率が高くなると考えられる。そのため密度の高い位置へ向かう確率が上がり、目標座標の近傍へ集中しなくなる。この軌道サンプルの密度の高い位置の存在については、確率的に動作させているときに特に影響を受ける事が分かる。選択されたサンプルのうち、いくつかは軌道サンプルの密度の高い位置に向かうようなサンプルであった場合、密度の高い位置へ向かうサンプルが選択される確率が上がるからであると考えられる。

5 結論

本論文ではモデル化を行わない制御として累積サンプルに基づく制御法を提案した。そして、累積サンプルに基づく制御法を冗長な構造を持つ Spring Binary Manipulator の制御に適用し、シミュレーションによりリーチングタスクを達成できることを示した。また、実ロボットである人間上肢型ロボティックアームに累積サンプルに基づく制御を適用し、リーチングタスクを達成した。すなわち、累積サンプルに基づく制御法は冗長な構造を持つロボットに対し実時間で制御信号の決定が可能であるということを示している。

シミュレーションと実機のいずれの場合にもリーチングタスクは達成できたが、いくつかの問題点も存在する。まず一つ目に、軌道サンプルに密度の高い位置が存在する場合、その密度の高い位置に向かって動作をするようなサンプルが選択されやすくなり、動作の精度が落ちる。この問題に対しては、現在の状態に近いサンプルを K 個取得する際に、同様の振る舞いをしているサンプルは選択しない



7: Result of reaching task: Human-Like Robotic Arm

という処理を加えることで解決できると考えられる。しかし、この処理を加えた場合にはより多くの累積サンプルが必要になると考えられる。

次に、制御対象のダイナミクスが大きく変化した場合に、目的とする動作を行わせる事ができなくなると考えられる。これは、サンプルとして保存されている系列データと現在の系列との性質が一致していない場合、精度よく制御ができない事が考えられるからである。この問題に対しては、インクリメンタルにサンプルを追加する処理を追加することで解決できると考えられる。

参考文献

- [1] 浅田稔, 野田彰一, 俵積田健, 細田耕: “視覚に基づく強化学習によるロボットの行動獲得” 日本ロボット学会誌 Vol.13 No. 1, pp.68-74, 1995
- [2] 木村元, 小林重信: “ロボットの強化学習における状態-行動空間の汎化” 日本ロボット学会誌 Vol. 22 No. 2, pp.161 ~ 164, 2004
- [3] 杉江俊治, 藤本健治: “近似線形化を用いた倒立振子の制御” 計測自動制御学会論文集 31(10), 1643-1649, 1995-10
- [4] 田崎勇一, 井上順一: “平面 2 足歩行ロボットのグラフ型モデル予測制御” 日本ロボット学会誌 Vol. 24 No.5, pp.663 ~ 671, 2006
- [5] Bridle, J.: “Probabilistic interpretation of feed-forward classification network outputs with relationships to statistical pattern recognition” In F. Fogelman-Soulie and J. Hertz (Eds.), *Neuro-computing: Algorithms, Architectures*. Springer-Verlag., 1989
- [6] Christopher M. Bishop: “Pattern Recognition and Machine Learning” Springer, August, 2006
- [7] Jackrit Suthakorn, Gregory S. Chirikjian: “A new inverse kinematics algorithm for binary manipulators with many actuators” *Advanced Robotics*, Volume 15, pp.225-244(20), Number 2, 2001
- [8] Lees, D.S. and Chirikjian, G.S.: “A Combinatorial Approach to Trajectory Planning for Binary Manipulators” *Robotics and Automation*, 1996. Proceedings., IEEE International Conference on, 1996
- [9] M. Haruno, D.M. Wolpert, and M. Kawato: “Mosaic: Module selection and identification for control,” *Neural Computation*, vol.13, no.10, pp.2201-2220, 2001
- [10] T.M. Cover, P.E. Hart: “Nearest neighbor pattern classification” *IEEE Trans Inform Theory* IT-13, pp.21-27, 1967
- [11] VIJAYMOHAN R. KONDA AND VIVEK S. BORKAR: “ACTOR-CRITIC-TYPE LEARNING ALGORITHMS FOR MARKOV DECISION PROCESSES” *SIAM J. CONTROL OPTIM.* Vol. 38, No. 1, pp.94-123, 1999
- [12] Watkins, C. J. C. H. and Dayan, P.: “Technical Note: Q-Learning” *Machine Learning* 8, pp. 279-292, 1992
- [13] Yoon Young Kim, Gang-Won Jang, and Sang Jun Nam: “Inverse Kinematics of Binary Manipulators by Using the Continuous-Variable-Based Optimization Method” *IEEE Transactions on Robotics*, Vol.22, No.1, pp.33-42, February, 2006