学術・技術論文

端点の多峰性最適化による 複数の解が導出可能な軌道計画法

長 隆之[*1*2] 佐藤雅也[*3] 森木和也[*3]
杉山聡[*3] 杉田直彦[*4] 中尾政之[*4]

Manipulation Planning with Multimodal End Point Optimization for Obtaining Multiple Solutions

Takayuki Osa[*1*2], Masaya Sato[*3], Kazuya Moriki[*3], Satoshi Sugiyama[*3], Naohiko Sugita[*4] *and* Masayuki Nakao[*4]

Motion planning for robotics manipulation is an essential component for automating various tasks. In this study we discuss optimization-based motion planning methods for robotic manipulation. The optimization-based method can compute smooth and collision-free trajectories with relatively short computational cost. Although existing methods are often designed to output a single solution, the objective function is often multimodal and there exist multiple solutions to achieve a given task. On such a task, obtaining multiple solutions gives a user an opportunity to choose one of the solutions based on factors which are not encoded in the objective function. To address this issue, we propose a motion planning framework that finds multiple solutions. The proposed method is validated in simulated environments with a four-link manipulator in 2D space and a 6 DoFs manipualtor in 3D space.

Key Words: Motion Planning, Multimodal Optimization, Importance Sampling

1. はじめに

マニピュレータの動作計画はロボットによる様々なタスクを自動化するためには欠かせない要素であり，長年研究されてきた問題である．これまで，ロボットにおける軌道計画を実現する方法として，最適化に基づく方法 [1] [2] やサンプリングに基づく方法 [3]〜[5] が開発されてきた．サンプリングに基づいた方法は，計算時間がかかるものの，複雑な環境下において干渉のない動作を計画することが可能であるため，迷路などのような環境下でのモバイル・ロボット等の動作計画に用いられる．一方で，最適化ベースの手法は，干渉のない滑らかな軌道を短時間で計算することができ，マニピュレータの動作計画などに適している．本研究では，目的関数を定義し，それを最適化することによって軌道を計画する最適化ベースの手法について論じる．最適化ベースのアプローチは干渉のない滑らかな軌道を短時間で計算することができるが，多くの場合単一の局所解を見つけ出すことを目的としている．しかし，現実の問題においては，目的関数は多峰性を示し，複数の解が存在することが多い．軌道計画において，複数の解を見つけ出し，ユーザに提示することができれば，ユーザには目的関数に含まれていない情報も加味して，適切な軌道を選ぶ余地が与えられる．また，最適化に基づく手法は，しばしば最適化の段階において数値的に不安定な挙動を示したり，干渉がない軌道を見つけることに失敗することがある．これは，目的関数が非凸であり，複数のモードをもつことに由来することが多い，複数のモードがあることを明示的に学習し，それぞれについて最適化を行うことができれば，干渉がない軌道を見つける割合も向上させることができると考えられる．加えて，既存の手法では軌道の始点および終点にあたるマニピュレータの姿勢を指定することが前提となっている．しかし実際には，タスクを実行するための最適な始点および終点のマニピュレータの姿勢は自明ではないことが多い，例えば，シリンダ状の物体を把持する際には，シリンダの軸を回転軸として，エンドエフェクタの姿勢をある程度変更しても把持を実行することができる（**Fig. 1**）．このような場合，マニピュレータがどのような角度で物体を把持をするように指定すれば最短かつ滑らかな軌道を得ることができるのかは自明ではなく，ユーザによる調整がしばしば必要になる．

本論文においては，本論文においては，目的関数の多峰性を考慮し，複数の軌道を得ることができる軌道の最適化法を提案する．軌道計画において，複数の解を見つけ出し，ユーザに提示することができれば，ユーザには目的関数に含まれていない情報も加味して，適切な軌道を選ぶ余地が与えられる．提案手法では，まず，筆者らが開発した階層型強化学習のアルゴリズム [6] を最適化のためのアルゴリズムとして捉えなおし，軌道

原稿受付 2018 年 12 月 26 日
*1九州工業大学生命体工学研究科人間知能システム工学専攻
*2理化学研究所革新知能統合研究センター
*3古河電気工業株式会社
*4東京大学大学院工学系研究科機械工学専攻
*1Department of Human Intelligence Systems, Graduate School of Life Science and Engineering, Kyushu Institute of Technology
*2RIKEN, Center for Advanced Intelligence Project
*3Furukawa Electric Co., Ltd.
*4Department of Mechanical Engineering, Graduate School of Engineering, the University of Tokyo
■ 本論文は新規性（要素分野）で評価されました．

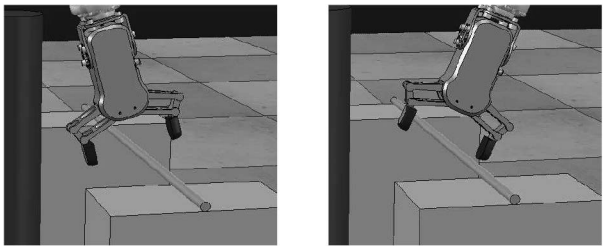


Fig. 1 When grasping a grasping a cylindrical object, there exist infinite number of postures for grasping, and it is not trivial to determine the optimal posture. It is also the case when grasping a planar object

の最適化へと拡張する．提案手法においては，コスト重み付きサンプリングを導入することで，目的関数を最適化する問題を，密度推定の問題として定式化する．本研究では，このアプローチを用いて，軌道の終点にあたるマニピュレータの姿勢を多峰性を考慮して最適化し，複数の終点姿勢を得る．加えて，本論文では軌道の始点および終点を目的関数に関して最適化し，局所解を見つけるアルゴリズムも提案する．この二つのアルゴリズムを組み合わせることにより，始点および終点の姿勢を自動で最適化し，複数の軌道を与えることができ，始点および終点をユーザが調整する負担を軽減することができる．

2. 関 連 研 究

軌道計画の手法には，大きく分けて 1）最適化ベース，2）サンプリングベース，3）模倣学習ベースの三つのタイプがある．最適化ベースの手法として有名なものとしては，CHOMP [2]，STOMP [7]，TrajOpt [1] などが挙げられる．これらの手法は，目的関数を定義し，軌道を目的関数に基づいて最適化することで，比較的短時間で滑らかで干渉のない軌道を求めることができる．しかし，目的関数が非凸であることから，適切な軌道を得ることができない場合もある．サンプリングベースの手法としては，Probabilistic RoadMap（PRM）[8] [9]，Rapidly-exploring Random Trees（RRT）[4] [5]，RRT* [3] などが挙げられる．これらのサンプリングベースの手法は非常に複雑な環境下においても干渉のない軌道を見つけ出すことができる．一方で，計算時間は比較的長くなり，得られる軌道は滑らかでないことが多いため，得られた軌道の平滑化がしばしば必要になる．三つめの模倣学習ベースの手法としては，Dynamic movement primitives（DMP）[10]，Probabilistic Movement Primitives（ProMP）[11] などがこの分類に属する．模倣学習ベースの手法では，ユーザが何らかの方法で行った軌道を計測し，そのモデルを学習することで，新たな環境に合わせて軌道を計画する．模倣学習ベースの手法は，点から点への単純な軌道でなく，糸の結紮 [12] など，軌道の形状そのものがタスクの成否にかかわるような場合に適している．しかし，模倣学習ベースの手法は障害物などが存在し干渉回避が必要になるような条件下では効率がよくないと一般的に言われている．模倣学習の詳細については，文献 [13] を参照されたい．

ここで紹介した三つのアプローチは完全に分かれているわけではなく，近年の研究はそれぞれのアプローチの利点を組み合わせた手法が開発されている．模倣学習ベースである DMP と最適化ベースである CHOMP の相似的な関係については，近年文献 [14] で論じられ，最適化ベースと模倣学習ベースの手法を融合させた手法も多く提案されている [15] [16].

同様に，文献 [17] の研究では，サンプリングベースの手法と模倣学習ベースの手法を組み合わせた手法が提案されている．また，STOMP [7] は最適化ベースの手法と考えられるが，その最適化のプロセスには，確率的に軌道をサンプリングするというサンプリングベースの手法の考え方が用いられている．さらに，文献 [5] で提案されている RRT*と PRM*は，サンプリングベースの手法に基づいているものの，目的関数に関して最適な経路を計画するという点で最適化ベースの手法とも関連がある．

上記のように，最適化ベース，サンプリングベース，および模倣学習ベースの手法はそれぞれ密接に関係している．これらの分類の中でも，本研究では最適化ベースの手法について論じる．提案する手法は CHOMP および STOMP と密接な関係があるが，我々の手法の新規性は，目的関数の多峰性を考慮し，複数の軌道を解として得られる点である．先行研究 [18] においては，与えられた終点の姿勢が障害物と干渉するようなものが与えられた場合には干渉がないような終点の姿勢に収束するような滑らかな軌道を生成するアルゴリズムを提案している．しかし，複数の実行可能な軌道を生成するようなことはできない．また，目的関数の最適化という観点は，サンプリングベースや模倣学習ベースの手法にも重要なものであり，多峰性最適化という観点について考慮した本研究の知見は，最適化ベース以外の手法にも有用であると考える．

3. 提 案 手 法

3.1 提案手法の概要

提案手法の概要を Algorithm 1 に示す．本研究では，最適な終点の姿勢の候補を複数抽出するアルゴリズム，および始点と終点の姿勢を含めて軌道全体を最適化するアルゴリズムの二つを提案する．提案する軌道計画法では，ユーザは暫定的な始点と終点におけるマニピュレータの姿勢に加えて，始点および終点においてマニピュレータが自由に角度を変えてよい回転軸を入力することとする．開発したシステムでは，まず，障害物とマニピュレータの位置関係に基づき，最適な終点の姿勢の候補を複数抽出する．本研究では，抽出する姿勢の候補の数は自動で決定されるようにアルゴリズムを実装した．次に，抽出された終点と，ユーザが指定した暫定的な始点の姿勢に基づき，軌道全体を最適化を行う．このとき，既存の多くの軌道計画法と異なり，提案する手法では，始点および終点の姿勢を含めて，軌道全体の局所最適化を行う．提案する軌道計画の概要を Algorithm 1 に示す．以下では，本研究で提案する，最適な終点の姿勢の候補を複数抽出するアルゴリズム，および始点と終点の姿勢を含めて軌道全体を最適化するアルゴリズムの詳細を述べる．

Algorithm 1 Overview of the proposed trajectory planning framework

1: **Input:** Initial starting and goal configurations $\boldsymbol{\xi}_0^0$ and $\boldsymbol{\xi}_N^0$, free rotation axes for start and goal configurations, $\boldsymbol{r}_s$, $\boldsymbol{r}_g$
2: Perform the multimodal optimization of the goal configuration with Algorithm 2 and obtain multiple goal configurations
3: **for each** goal configuration **do**
4: Perform the trajectory optimization with Algorithm 3
5: **end for**

3.2 最適化による軌道計画の問題設定

マニピュレータのコンフィギュレーションを $\boldsymbol{q}$ とし，マニピュレータの軌道をコンフィギュレーションの連続として

$\boldsymbol{\xi} = [\boldsymbol{q}_0, \ldots, \boldsymbol{q}_N]$ のように表現することとする．ここで $N \in \mathbb{N}$ は軌道の総ステップ数である．軌道 $\boldsymbol{\xi}$ のコストを与える関数を $\mathcal{C}(\boldsymbol{\xi})$ とすると，最適化による軌道計画は以下のような問題として定式化される．

$$\boldsymbol{\xi}^* = \arg\min_{\boldsymbol{\xi}} \mathcal{C}(\boldsymbol{\xi}) \tag{1}$$

本研究では，CHOMP で提案されたものと同様のコスト関数を用いる．すなわち，コスト関数は以下で与えられる．

$$\mathcal{C}(\boldsymbol{\xi}) = \mathcal{C}_{\mathrm{obs}}(\boldsymbol{\xi}) + \mathcal{C}_{\mathrm{smooth}}(\boldsymbol{\xi}) \tag{2}$$

ここで，$\mathcal{C}_{\mathrm{obs}}(\boldsymbol{\xi})$ は障害物との干渉のコストであり，$\mathcal{C}_{\mathrm{smooth}}(\boldsymbol{\xi})$ は軌道の滑らかさに関するコストである．滑らかさに関するコストは以下で与えられる．

$$\mathcal{C}_{\mathrm{smooth}}(\boldsymbol{\xi}) = \sum_{i=1}^{N} \|K_1\boldsymbol{\xi} + \boldsymbol{e}_1\|^2 \tag{3}$$

ここで，$K_1, \boldsymbol{e}_1$ は

$$K_1 = \begin{bmatrix} 1 & 0 & 0 & \ldots & & 0 \\ -1 & 1 & 0 & & & \\ 0 & -1 & 1 & & & \\ 0 & 0 & & & 0 & 0 \\ 0 & 0 & & & 1 & 0 \\ & & & & -1 & 1 \\ 0 & 0 & & & 0 & -1 \end{bmatrix}, \tag{4}$$

$$\boldsymbol{e}_1 = [\boldsymbol{q}_0, 0, \ldots, 0, \boldsymbol{q}_N]^T \tag{5}$$

で与えられる．干渉のコスト $\mathcal{C}_{\mathrm{obs}}(\boldsymbol{\xi})$ は以下で与えられる．

$$\mathcal{C}_{\mathrm{obs}}(\boldsymbol{\xi}) = \frac{1}{2}\sum_{t=0}^{N}\sum_{u\in\mathcal{B}} c\left(\boldsymbol{x}_u(t)\right)\left\|\frac{d}{dt}\boldsymbol{x}_u(t)\right\| \tag{6}$$

$\mathcal{B}$ はマニピュレータのボディを表す点群であり，u は $\mathcal{B}$ の中の点を指す．$\boldsymbol{x}_u(t)$ は点 u のステップ t におけるタスク空間上での位置を表す．ここで，障害物の表面と $\boldsymbol{x}_u(t)$ との位置関係を表す関数 $d(\boldsymbol{x}_u)$ を導入する．障害物の表面と $\boldsymbol{x}_u(t)$ との最短距離を $d_u^{\min}$ とすると，点 u が障害物の内部にあるとき，$d(\boldsymbol{x}_u) = -d_u^{\min}$ であり，点 u が障害物の外部にあるとき，$d(\boldsymbol{x}_u) = d_u^{\min}$ であるとする．この関数 $d(\boldsymbol{x}_u)$ を用いて，マニピュレータのボディの各点のコスト $c(\boldsymbol{x}_u)$ は以下で与えられる．

$$c(\boldsymbol{x}_u) = \begin{cases} 0, & \text{if} \quad d(\boldsymbol{x}_u) > \epsilon, \\ \frac{1}{2\epsilon}(d(\boldsymbol{x}_u) - \epsilon)^2, & \text{if} \quad 0 < d(\boldsymbol{x}_u) < \epsilon, \\ -\ d(\boldsymbol{x}_u) + \frac{1}{2}\epsilon, & \text{if} \quad d(\boldsymbol{x}_u) < 0 \end{cases} \tag{7}$$

ここでは，このコスト関数に関する詳細な考察は述べないが，詳細は文献 [2] を参照されたい．

逐次的に軌道を最適化するプロセスについて論じるため，k 回目の更新で得られた軌道を $\boldsymbol{\xi}^k = [\boldsymbol{q}_0^k, \ldots, \boldsymbol{q}_N^k]$ とし，ユーザが最初に指定した軌道を $\boldsymbol{\xi}^0 = [\boldsymbol{q}_0^0, \ldots, \boldsymbol{q}_N^0]$ とする．ここで，軌道 $\boldsymbol{\xi}^k$ 近傍においてコスト関数をテーラー近似すると以下の式が得られる．

$$\mathcal{C}(\boldsymbol{\xi}) \approx \mathcal{C}(\boldsymbol{\xi}^k) + \boldsymbol{g}_k^\top(\boldsymbol{\xi} - \boldsymbol{\xi}_k) \tag{8}$$

ここで $\boldsymbol{g}_k = \nabla\mathcal{C}(\boldsymbol{\xi}^k)$ である．CHOMP においては，以下のように正規化した軌道の更新式を用いる．

$$\boldsymbol{\xi}^{k+1} = \arg\min_{\xi}\left\{\mathcal{C}(\boldsymbol{\xi}^k) + \boldsymbol{g}_k^\top(\boldsymbol{\xi} - \boldsymbol{\xi}^k) + \frac{\lambda}{2}\left\|\boldsymbol{\xi} - \boldsymbol{\xi}^k\right\|_M^2\right\} \tag{9}$$

ここで M は軌道の変化を評価する行列であり，$\left\|\boldsymbol{\xi} - \boldsymbol{\xi}^k\right\|_M = (\boldsymbol{\xi} - \boldsymbol{\xi}^k)^T M(\boldsymbol{\xi} - \boldsymbol{\xi}^k)$ である．右辺の中身を 0 とすると，更新式は以下のように簡潔な形で得ることができる．

$$\boldsymbol{\xi}^{k+1} = \boldsymbol{\xi}^k - \frac{1}{\lambda}M^{-1}\boldsymbol{g}_k \tag{10}$$

式 (9) の右辺第 3 項は，滑らかな軌道を得るための正規化のための項とみなすことができ，$M = K_1^\top K_1$ とすると，コスト関数の軌道に関する勾配を滑らかに軌道全体に波及させる効果がある．

3.3 提案手法：コスト重み付きサンプリングを用いた多峰最適化による端点姿勢の計画

軌道計画における最適化の目標は，目的関数 $\mathcal{C}(\boldsymbol{\xi})$ を最小化する解を見つけることである．本研究では，筆者らが開発した報酬重み付き密度推定による階層型強化学習アルゴリズム [6] を終点の姿勢の最適化に用いることができるように拡張する．この最適化問題を解くにあたって，以下のようなマニピュレータのコンフィギュレーションの分布を便宜上考える．

$$d^{\mathcal{C}}(\boldsymbol{q}) = \frac{f(-\mathcal{C}(\boldsymbol{q}))}{Z} \tag{11}$$

ここで，$f(\cdot)$ は入力に対して単調増加する関数であり，かつ，常に $f > 0$ である．Z はパーティション関数とよばれるもので，分布を正規化するための項である．このような分布に従うとき，コストのより小さいコンフィギュレーションがより高い確率で生成される．よって，$d^{\mathcal{C}}(\boldsymbol{q})$ が作り出すコンフィギュレーションの密度のモード（峰）は，コスト関数の極値に一致する．つまり，$d^{\mathcal{C}}(\boldsymbol{q})$ が作り出す密度のモードを与えるコンフィギュレーションを求めるということは，目的関数を最小化するコンフィギュレーションを求めることに等しい．したがって，目的関数を最小化するという問題を，式 (11) が作り出すコンフィギュレーションの密度を推定するという問題に置き換えることができる．

提案する方法では，終点姿勢 $\boldsymbol{q}_N$ を計画するという問題を，ベクトル $\boldsymbol{\theta}$ によってパラメタライズされた分布 $d_{\boldsymbol{\theta}}(\boldsymbol{q})$ を学習する問題として捉える．以下のような KL 情報量を最小化する $d_{\boldsymbol{\theta}}(\boldsymbol{q})$ を求める問題として定式化する．

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} D_{\mathrm{KL}}\left(d^{\mathcal{C}}(\boldsymbol{q})||d_{\boldsymbol{\theta}}(\boldsymbol{q})\right) \tag{12}$$

ここで，KL 情報量は以下のように定義される．

$$D_{\mathrm{KL}}\left(d^{\mathcal{C}}(\boldsymbol{q})||d_{\boldsymbol{\theta}}(\boldsymbol{q}_N)\right) \quad = \int d^{\mathcal{C}}(\boldsymbol{q}_N)\log\frac{d^{\mathcal{C}}(\boldsymbol{q})}{d_{\boldsymbol{\theta}}(\boldsymbol{q})}\mathrm{d}\boldsymbol{q} \tag{13}$$

しかし，実際には $d^{\mathcal{C}}(\boldsymbol{q})$ からコンフィギュレーションをサンプリングすることはできない．この密度推定問題を解くために，重点サンプリングのアプローチをとる．すなわち，L 個のコンフィギュレーションのセット $\{\boldsymbol{q}_l\}_{l=1}^L$ を提案分布 $d^{\mathrm{p}}(\boldsymbol{q})$ に従ってサンプリングし，それぞれの軌道のコスト $\mathcal{C}(\boldsymbol{q}_l)$ を評価したうえで，$d^{\mathcal{C}}(\boldsymbol{q})$ を重みづけを使って推定するということを行う．重要度重みは以下のように与えられる．

$$W(\boldsymbol{q}) = \frac{d^{\mathcal{C}}(\boldsymbol{q})}{d^{\mathrm{p}}(\boldsymbol{q})} = \frac{f(-\mathcal{C}(\boldsymbol{q}))}{Zd^{\mathrm{p}}(\boldsymbol{q})}. \tag{14}$$

実用のうえでは，重要度重みは正規化して用いられる．正規化された重要度重みは

$$\tilde{W}(\boldsymbol{q}) = \frac{W(\boldsymbol{q})}{\sum_{l=1}^{L} W(\boldsymbol{q}_l)} = \frac{\frac{f(-\mathcal{C}(\boldsymbol{q}))}{Zd^{\mathrm{P}}(\boldsymbol{q})}}{\sum_{l=1}^{L} \frac{f(-\mathcal{C}(\boldsymbol{q}_l))}{Zd^{\mathrm{P}}(\boldsymbol{q}_l)}} \quad (15)$$

$$= \frac{\frac{f(-\mathcal{C}(\boldsymbol{q}))}{d^{\mathrm{P}}(\boldsymbol{q})}}{\sum_{l=1}^{L} \frac{f(-\mathcal{C}(\boldsymbol{q}_l))}{d^{\mathrm{P}}(\boldsymbol{q}_l)}}. \quad (16)$$

で与えられる．パーティション関数 Z は正規化においてキャンセルされてしまうため，実際には Z を計算する必要はない．このようにして得られた重要度重みを用いた密度推定を，コスト重み付きサンプリングとよぶこととする．筆者らの先行研究 [6] において，このようなコスト重み付きサンプリングを用いて式 (12) の密度推定の問題を解くことは，コストの最小化あるいは報酬の最大化を行うことと同等であることが示されている．

提案する手法の実装においては，ガウス混合モデル（GMMs）を用いて多峰性の分布を表現することとした．O 個のクラスタを持つガウス混合分布は以下のように表される．

$$d_{\boldsymbol{\theta}}(\boldsymbol{q}) = \sum_{o=1}^{O} p(o)p(\boldsymbol{q}|o) \quad (17)$$

このガウス混合モデルをコスト重み付きサンプリングによって学習するための手法として，ベイズ EM アルゴリズムを用いることとした．最尤法 EM アルゴリズムによっても GMM を学習することはできるが，ベイズ EM を用いることで，必要なクラスタの数を自動で推定することができる．ベイズ EM を用いて GMM をフィッティングすると，まったく要素を含まないクラスタと要素を含むクラスタが現れ，要素を含むクラスタの数が実際にサンプルの密度を表現するために必要なガウス分布の数であるといえる．提案手法では，要素を含むクラスタの数を解の個数とみなすことができ，すなわち解の個数を自動的に学習することができるといえる．

学習したガウス混合モデルを利用すると，サンプリングされたコンフィギュレーションをそれぞれのクラスタに以下のように割り振ることができる．

$$o(\boldsymbol{q}) = \arg\max_{o'} p(o'|\boldsymbol{q}). \quad (18)$$

学習されたガウス混合モデルの各クラスタの平均に対応するコンフィギュレーションが，コスト関数のモードに対応する．すなわち，j 番めのモードに対応するコンフィギュレーションは，

$$\boldsymbol{q}_j = \frac{\sum_{l=1}^{L} \left(\delta(o(\boldsymbol{q}_l) - j)\tilde{W}(\boldsymbol{q}_l)\boldsymbol{q}_l\right)}{\sum_{l=1}^{L} \left(\delta(o(\boldsymbol{q}_l) - j)\tilde{W}(\boldsymbol{q}_l)\right)}, \quad (19)$$

で与えられる．ここで，$\delta(\cdot)$ はデルタ関数であり，$o = j$ ならば $\delta(o - j) = 1$ であり，そのほかの場合は $\delta(o - j) = 0$ となる．

終点姿勢の最適化に当たっては，エンドエフェクタのタスク空間上の並進位置は変えず，姿勢にはある回転軸に沿って自由度があると仮定する．実装したシステムにおいては，姿勢をサンプリングする提案分布 d^{P} として一様分布を用いた．終点姿勢を最適化するアルゴリズムの概要を Algorithm 2 に示す．

Algorithm 2 Mutimodal End-point Optimization Algorithm (MultiEndOpt)

1: **Input:** an initial goal configuration $\boldsymbol{q}_N^0$
2: **for** $l = 1, \ldots, L$ **do**
3: sample a goal configuration with a fixed end-effector position and a different posture
4: Compute the cost of the sampled configuration
5: **end for**
6: Perform density estimation with importance sampling using the weight in Eq. (16)
7: **return** goal configurations that correspond to the modes of the objective function

4. 始点姿勢および終点姿勢の局所最適化

前章で述べたアルゴリズムにより抽出される複数の終点の姿勢は，終点の姿勢のみを考慮した，あくまで大まかな推定に過ぎない．このため，最適な軌道を得るためには，終点の姿勢も含めて，軌道全体を最適化を行う必要がある．本章では，始点姿勢および終点姿勢の局所最適化を行う軌道の更新を行うべく CHOMP を拡張する．

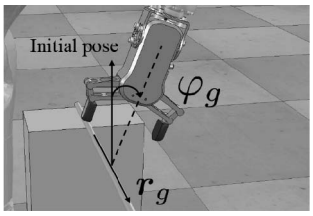


Fig. 2 We denote by φ_g the rotation angle around the rotation axis $\boldsymbol{r}_g$ specified by a user

ここでは簡単のため，終点姿勢の最適化について考える．終点姿勢の最適化に当たっては，エンドエフェクタのタスク空間上の並進位置は変えず，姿勢にはある回転軸に沿って自由度があると仮定する．ここで，自由度のある回転軸が $\boldsymbol{r}_N = [r_x^N, r_y^N, r_z^N]$ というベクトルで与えられるとする．ユーザに与えられた終点における姿勢に対する回転軸 $\boldsymbol{r}$ 周りの回転の角度を φ_g とする（**Fig. 2**）．CHOMP と同様逐次的に軌道を最適化するプロセスについて論じるため，k 回めの更新で得られた軌道を $\boldsymbol{\xi}^k = [\boldsymbol{q}_0^k, \ldots, \boldsymbol{q}_N^k]$ とし，ユーザが最初に指定した軌道を $\boldsymbol{\xi}^0 = [\boldsymbol{q}_0^0, \ldots, \boldsymbol{q}_N^0]$ とする．また，k 回めの更新で得られた軌道の終点の姿勢とユーザが最初に指定した軌道の終点の姿勢のなす角度を φ_g^k とする．

φ_g^k を $\Delta\varphi_g$ だけ変化させるとき，軌道の終点のコンフィギュレーション $\boldsymbol{q}_N$ は，$\boldsymbol{q}_N^k$ の近傍で

$$\boldsymbol{q}_N \approx \boldsymbol{q}_N^k + \Delta\varphi_g J^{-1}(\boldsymbol{q}_N^k)[0, 0, 0, r_x^N, r_y^N, r_z^N]^\top \quad (20)$$

で与えられる．ここで，J はマニピュレータのヤコビ行列である．終点での姿勢の変化 $\Delta\boldsymbol{q}_N = \Delta\varphi_g J^{-1}[0, 0, 0, r_x^N, r_y^N, r_z^N]^\top$ を CHOMP と同様に軌道全体に滑らかに伝播させることで，軌道全体の滑らかさを保つ．すなわち，軌道全体の微小変化を

$$\Delta\boldsymbol{\xi}_N = M_{\mathrm{end}}^{\dagger}[0, \ldots, 0, \Delta\boldsymbol{q}_N]^\top \quad (21)$$

とする．ここで $M_{\mathrm{end}}^{\dagger}$ は

$$M_{\mathrm{end}} = \begin{bmatrix} 0 & 0 & 0 & \ldots & & 0 \\ 0 & 2 & -1 & & & \\ 0 & -1 & 2 & & & \\ 0 & 0 & -1 & & 0 & 0 \\ 0 & 0 & & & -1 & 0 \\ & & & & 2 & -1 \\ 0 & 0 & & & -1 & 2 \end{bmatrix}, \quad (22)$$

で与えられる $N \times N$ の行列 M_{end} の疑似逆行列である．軌道の局所的な変化を軌道全体に伝播させる行列の形式については，文献 [14] に考察が詳述されているので参照されたい．

以上を考慮し，終点姿勢を回転軸 $\boldsymbol{r}_N$ 周りに微小角 $\Delta\varphi_g$ だけ変化させて軌道全体を更新する問題は，以下のような最適化問題として定式化する．

$$\Delta\varphi_g^* = \arg\min_{\Delta\varphi} \mathcal{C}(\boldsymbol{\xi} + \Delta\boldsymbol{\xi}_N) \tag{23}$$

$$\text{s.t. } \Delta\boldsymbol{q}_N = \Delta\varphi_g J^{-1}(\boldsymbol{q}_N^k)[0,0,0,r_x^N, r_y^N, r_z^N]^\top \tag{24}$$

$$\Delta\boldsymbol{\xi}_N = M_{\mathrm{end}}^\dagger [0,\ldots,0,\Delta\boldsymbol{q}_N]^\top \tag{25}$$

上記の問題を解くため，終点姿勢を回転軸 $\boldsymbol{r}_N$ 周りに微小角 $\Delta\varphi_g$ だけ回転した際の軌道のコストの変化を以下のように評価する．

$$\Delta\mathcal{C}_{\mathrm{end}}^k = \mathcal{C}(\boldsymbol{\xi}^k + \Delta\boldsymbol{\xi}_N) - \mathcal{C}(\boldsymbol{\xi}^k) \tag{26}$$

コストを最小化するように，終点姿勢を回転軸 $\boldsymbol{r}_N$ 周りの回転角 φ_g は以下のように更新される．

$$\varphi_g^{k+1} = \varphi_g^k - \alpha \frac{\Delta\mathcal{C}_{\mathrm{end}}^k}{\Delta\varphi_g} \tag{27}$$

ここで，α は学習率となる定数である．軌道全体は，式（22），（23）に従うように更新すればよい．すなわち，

$$\boldsymbol{\xi}^{k+1} = \boldsymbol{\xi}^k - \Delta\boldsymbol{\xi}'_N, \tag{28}$$

$$\Delta\boldsymbol{q}'_N = (\varphi_g^{k+1} - \varphi_g^k) J^{-1}(\boldsymbol{q}_N^k)[0,0,0,r_x^N, r_y^N, r_z^N]^\top, \tag{29}$$

$$\Delta\boldsymbol{\xi}'_N = M_{\mathrm{end}}^\dagger [0,\ldots,0,\Delta\boldsymbol{q}'_N]^\top \tag{30}$$

のように軌道を更新できる．

始点の姿勢の最適化についても，同様に更新式を得ることができる．k 回めの更新で得られた軌道の始点の姿勢とユーザが最初に指定した軌道の始点の姿勢のなす回転軸 $\boldsymbol{r}_0 = [r_x^0, r_y^0, r_z^0]$ 周りの角度を φ_s^k とする．コストを最小化するように，φ_s^k を更新するための式は以下のように得られる．

$$\varphi_s^{k+1} = \varphi_s^k - \alpha \frac{\Delta\mathcal{C}_{\mathrm{st}}^k}{\Delta\varphi_s} \tag{31}$$

ここで

$$\Delta\mathcal{C}_{\mathrm{st}}^k = \mathcal{C}(\boldsymbol{\xi}^k + \Delta\boldsymbol{\xi}_0) - \mathcal{C}(\boldsymbol{\xi}^k), \tag{32}$$

$$\Delta\boldsymbol{q}_0 = \Delta\varphi_s J^{-1}[0,0,0,r_x^0, r_y^0, r_z^0], \tag{33}$$

$$\Delta\boldsymbol{\xi}_0 = M_{\mathrm{st}}^\dagger [\Delta\boldsymbol{q}_0, 0,\ldots,0]^\top, \tag{34}$$

$$M_{\mathrm{st}} = \begin{bmatrix} 2 & -1 & & & & 0 \\ -1 & 2 & & & & 0 \\ 0 & -1 & & 0 & 0 & 0 \\ 0 & & & -1 & 0 & 0 \\ & & & 2 & -1 & 0 \\ 0 & & & -1 & 2 & 0 \\ 0 & 0 & \cdots & & 0 & 0 \end{bmatrix}, \tag{35}$$

である．軌道全体の更新については，

$$\boldsymbol{\xi}^{k+1} = \boldsymbol{\xi}^k - \Delta\boldsymbol{\xi}'_0, \tag{36}$$

$$\Delta\boldsymbol{q}'_0 = (\varphi_s^{k+1} - \varphi_s^k) J^{-1}(\boldsymbol{q}_0^k)[0,0,0,r_x^0, r_y^0, r_z^0]^\top, \tag{37}$$

$$\Delta\boldsymbol{\xi}'_0 = M_{\mathrm{st}}^\dagger [\Delta\boldsymbol{q}'_0, 0,\ldots,0]^\top. \tag{38}$$

のように与えられる．

始点姿勢および終点姿勢の局所最適化を行う，CHOMP を拡張したアルゴリズムを Algorithm 3 に示す．次章で示すシミュレーションによる評価では，$\boldsymbol{q}_0^0$ および $\boldsymbol{q}_N^0$ を指定し，線形補完することで経路の中間点を計算し，初期経路を与えた．このほかの初期経路の与え方としては，STOMP のなどで用いられているように，確率的な方法でランダムな動きを組み込んで経路を初期化することも可能である [7]．始点および終点に設定する自由回転軸の与え方はそれぞれの問題設定によって異なるが，2 指のグリッパーを用いている場合には，グリッパーの指が動作する平面に対して垂直な方向を与えると多くの場合において機能すると考えられる．例えば，長軸状の物体を把持し移動させるようなタスクの軌道計画であれば，その長軸状の物体の長軸方向を始点および終点の自由回転軸として与えるのが自然であると考えられる．このとき，物体の長軸方向は，グリッパーの指が動作する平面に対して垂直な方向と一致する．

Algorithm 3 CHOMP with Start and Goal Configuration Optimization

1: **Input:** Initial trajectory $\boldsymbol{\xi}^1$, free rotation axis $\boldsymbol{r}_0$, $\boldsymbol{r}_N$
2: **for** k = 1,..., K **do**
3: Compute the cost gradient
4: Update the trajectory with Eq. (10)
5: Perturb the start configuration:
$\boldsymbol{q}_0 \to \boldsymbol{q}_0 + \Delta\boldsymbol{q}_0$
where $\Delta\boldsymbol{q}_0 = \Delta\varphi_s J^{-1}[0,0,0,r_x^0, r_y^0, r_z^0]^\top$
6: Compute the change of the cost function:
$\Delta\mathcal{C}_{\mathrm{st}} = \mathcal{C}(\boldsymbol{\xi}^k + \Delta\boldsymbol{\xi}_0) - \mathcal{C}(\boldsymbol{\xi}^k)$
where $\Delta\boldsymbol{\xi}_0 = M_{\mathrm{st}}^\dagger[\Delta\boldsymbol{q}_0, 0,\ldots,0]^\top$
7: Update the trajectory using Eq. (36)
8: Perturb the goal configuration:
$\boldsymbol{q}_N \to \boldsymbol{q}_N + \Delta\boldsymbol{q}_N$
where $\Delta\boldsymbol{q}_N = \Delta\varphi_g J^{-1}(\boldsymbol{q}_N^k)[0,0,0,r_x^N, r_y^N, r_z^N]^\top$
9: Compute the change of the cost function:
$\Delta\mathcal{C}_{\mathrm{end}} = \mathcal{C}(\boldsymbol{\xi}^k + \Delta\boldsymbol{\xi}_N) - \mathcal{C}(\boldsymbol{\xi}^k)$
where $\Delta\boldsymbol{\xi}_N = M_{\mathrm{end}}^\dagger[0,\ldots,0,\Delta\boldsymbol{q}_N]^\top$
10: Update the trajectory using Eq. (28)
11: **end for**
12: **return** the optimized trajectory

5. シミュレーションによる検証

提案手法の有効性を確認するため，シミュレーション上で提案手法の評価を行った．まず，可視化が容易な二次元平面上での 4 リンク機構の軌道計画を行い，評価を行った．次に，三次元空間上で 6 自由度を持つマニピュレータに対し，提案手法による軌道計画を行った．

5.1 二次元平面上での軌道計画

ここでは，二次元平面上で動作する 4 リンクマニピュレータの軌道計画に対して，提案手法の評価を行った．ここで示す軌道計画では，マニピュレータの始点と終点の並進位置のみ固定であり，紙面に対して垂直な方向に対して，自由度があるという問題を考える．

提案手法による軌道の最適化の例を **Fig. 3** に示す．Fig. 3 (a) に指定された始点および終点の姿勢が示されている．図のとおり，与えられた終点の姿勢では障害物と干渉する設定となっている．この終点姿勢と障害物の配置が与えられた条件下で Algorithm 2 に示した提案手法による軌道計画を実施した．提案手

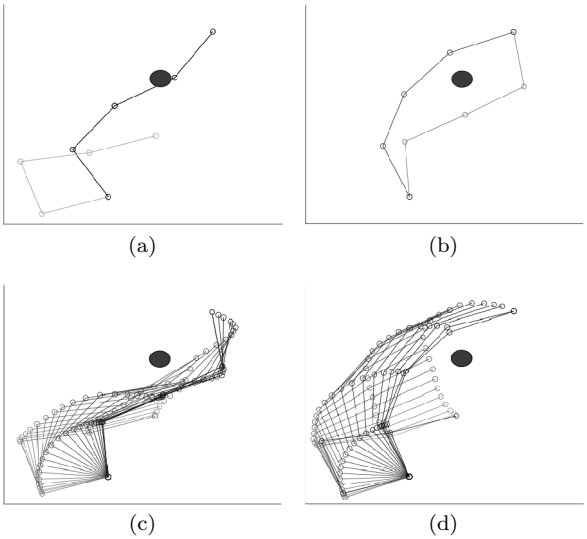


Fig. 3 An example of trajectory optimization using the proposed method. In (a), a solid circle represents an obstacle, and given configurations at the start and goal points are shown in gray and black, respectively. The given goal configuration has a collision with the obstacle. (b) shows two collision-free goal configurations obtained by the proposed method. (c) and (d) shows two trajectories obtained by applying the proposed method summarized in Algorithm 3

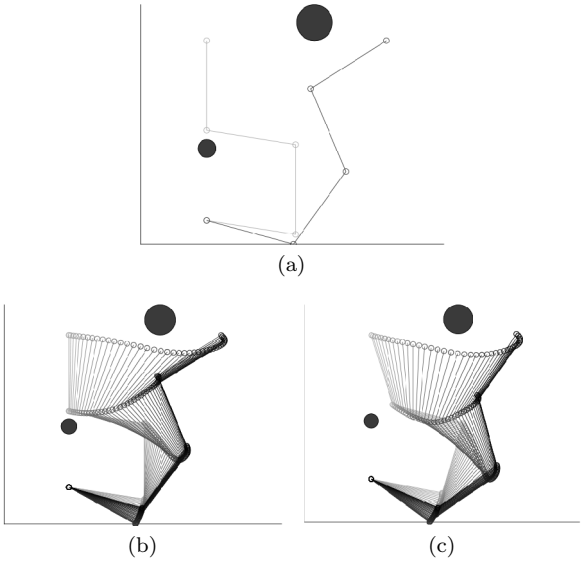


Fig. 4 An example of trajectory optimization for a four-link manipulator in 2D space. In (a), a solid circle represents an obstacle, and given configurations at the start and goal points are shown in gray and black, respectively. (b) shows the result of CHOMP, which does not optimize the start and goal configurations. (c) shows the result of the proposed method summarized in Algorithm 3. One can see that the start and goal configurations are updated from given ones. Especially, the start configuration is updated so that the manipulator is away from the obstacle

法では確率的な最適化を行うため，提案手法によって得られる解が実施するごとに異なる．このため，10 回軌道計画を実施し，その性能を評価した．提案手法では，10 回の軌道計画のうち，8 回で 2 個の干渉のない軌道が得られ，2 回では干渉のない軌道が 1 個得られた．Fig. 3 (b) には，2 個の干渉のない終点姿勢を得られた際の結果を示す．また，得られた 2 とおりの終点姿勢と与えられた始点姿勢に対し，それぞれ Algorithm 3 を適用したところ，Fig. 3 (c) および (d) のような軌道を得ることができた．本シミュレーションは, Matlab を用いて実装し，CPU は core i7-8659U を備えたコンピュータにおいて実施した．Fig. 3 に示した軌道計画において，Algorithm 2 によって終点姿勢を最適化する処理にかかった時間は 10 回の平均で 0.09 秒ほどであった．

5.2 始点および終点を最適化する提案手法の評価

始点および終点の姿勢を最適化する提案手法である Algortihm 3 を評価するため **Fig. 4** に示すような軌道計画を行った．ここでは Fig. 4 (a) に示される始点および終点の姿勢を与え，中実の円に示すような障害物を設定した．Fig. 4 (b) は CHOMP による軌道最適化の結果であり，始点と終点の最適化は行われない．Fig. 4 (c) が Algortihm 3 による軌道最適化の結果を示しており，Fig. 4 (a) と比較すると，始点および終点の姿勢が変化していることが分かる．特に，始点の姿勢が障害物から遠ざかる方向に更新されている．Fig. 4 (a) と (b) の軌道を定量的に比較すると，滑らかさに関するコストが 18%，干渉のコストが 95% 低減されている．干渉コストの低減の割合は，障害物の配置等で大きく変わるものであるため，改善の割合の大きさは重要な意味を持たないが，この結果は，軌道の始点および終点の姿勢を提案手法によって更新することによって，滑らかさや干渉のリスクを低減することができることを示している．本シミュレーションは, CPU はクロック周波数が 1.9〜2.1 [GHz] の core i7-8659U を備えたコンピュータにおいて実施し，Fig. 4 に示された軌道最適化を 10 回行った際の計算時間の平均値は，CHOMP が 0.20 秒，提案手法である Algortihm 3 は 0.21 秒とほぼ同等であった．また，筆者らの実装したプログラムでは，CHOMP は平均 38 回，提案手法は平均 15 回で解が収束した．提案手法は CHOMP よりも 1 回の軌道の更新についての計算量が多いことは明らかであるが，提案手法はより少ない更新回数で軌道が解に収束するため，全体としての計算時間がほぼ同等となったと考えられる．

5.3 三次元空間における 6 軸マニピュレータにおける軌道計画への適用

三次元空間上での軌道計画における提案手法の性能を確認するためのシミュレーションを行った．ここでは，軌道計画法を適用する 6 軸マニピュレータをデンソーウェーブ社の VS060 とし，**Fig. 5** に示すように，柱状の障害物が存在する中で，棒状の目標物を把持するための位置へと移動するための軌道を計画する問題へと，提案手法を適用した．ここでは，指定した目標姿勢は，あえて障害物と干渉するような姿勢を与えた．本検証では，提案手法において導き出せる解の最大の数を 10 に指定した．軌道を可視化するためのシミュレーションの構築には V-REP を用いた [19]．本シミュレーションも，前節で述べた実験と同様，CPU はクロック周波数が 1.9〜2.1 [GHz] の core i7-8659U を備えたコンピュータを用いて実施した．

提案手法では，確率的な方法で終端姿勢を最適化しており，生成される解の個数は最適化を実行するたびに異なる．10 回の軌道生成のうち，2 個の解を得られたのが 2 回，3 個の解を得られたのが 7 回，4 個の解を得られたのが 1 回だった．2 個の解が得られた場合の結果を，**Fig. 6** に示す．3 個以上の解が得られる場合には，Fig. 6 の (a) か (b) のどちらかに似た軌道を 2 個生成していた．計算時間としては，Algortihm 2 によって複数の終点姿勢を得るための最適化に要した時間は 10 回の平均が 1.33 秒

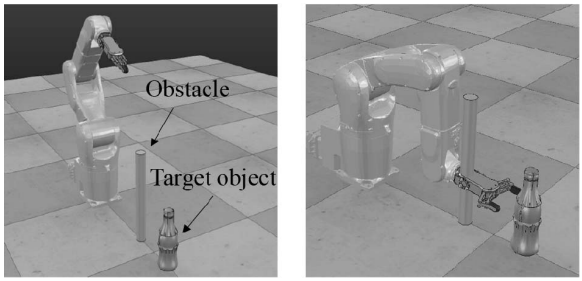
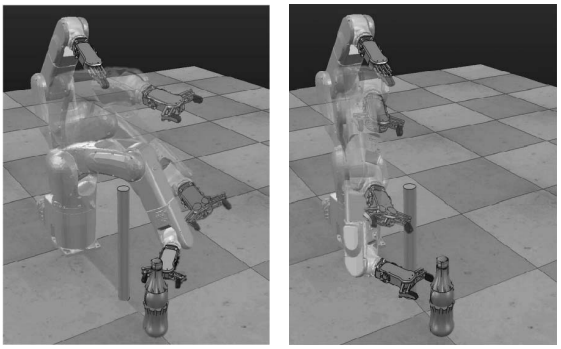


Fig. 5 The first problem setting for 6 DOF manipulator. A cylindrical obstacle needs to be avoided to reach a cylindrical target object. The left figure shows a given start configuration, and the right figure shows a given goal configuration. The given goal configuration has a collision with the obstacle



(a) (b)

Fig. 6 Two trajectories obtained by the proposed method for the first problem setting. Each trajectory reaches the target object with different postures

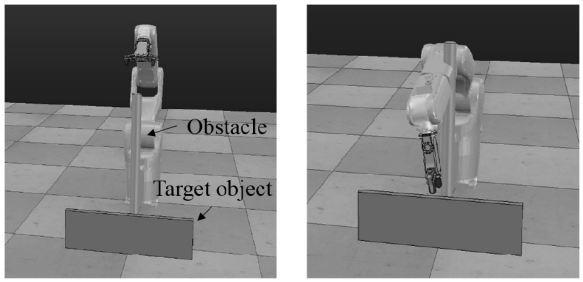
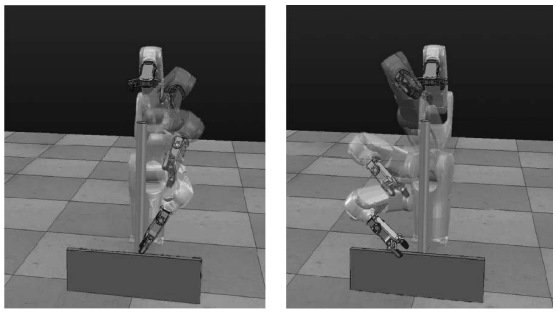


Fig. 7 The second problem setting for 6 DOF manipulator. A cylindrical obstacle needs to be avoided to reach a planar target object. The left figure shows a given start configuration, and the right figure shows a given goal configuration. The given goal configuration has a collision with the obstacle

であった．また，Algortihm 3 によって始点および終点を含めた軌道の最適化に要した時間は 10 回の平均が 121.3 秒であった．

次に，**Fig. 7** に示すような，板状の物体を把持することを想定した状況での軌道計画に対して，同様に提案する軌道計画法を適用した．10 回の軌道生成のうち，2 個の解を得られたのが 4 回，3 個の解を得られたのが 6 回だった．**Fig. 8** の (a), (b) には，2 個の軌道が得られた際の結果を示した．3 個以上の解が得られる場合には，Fig. 8 の (a) あるいは (b) のどちらかに似た軌道を 2 個生成していた．Fig. 8 において左右対称な解が現れないのは，提案手法の最適化が確率的な方法で行われているためと考えられる．軌道生成を複数回行う中で，左右対称に見える解が生成される場合も観察された．

ここで得られた複数の軌道はそれぞれ，定義されたコスト関



(a) (b)

Fig. 8 Two trajectories obtained by the proposed method for the second problem setting. Each trajectory reaches the target object with different postures

数の値という観点からは，同等の値を示す．単一の軌道のみを解として出力する既存の手法では，異なるアームの動きが望ましい場合には，ユーザがコスト関数をチューニングしたり，異なる軌道の初期化を試すなどの作業が必要になる．しかし，提案手法においては，複数の軌道の候補を出力することができ，ユーザに好ましい軌道を選ぶ余地を与えることができるということが実験からも示唆された．

6. 考 察

本研究において想定した最適な姿勢とは，与えられた目的関数を最小化すると考えられる姿勢であり，目的関数がどのような中身であれ，計算可能なものであれば提案手法と組み合わせて用いることができると考えられる．今回の論文で示したシミュレーションによる評価では把持などのタスクに基づく終点姿勢の評価は組み込まれていないが，それぞれの終点のコンフィギュレーションについて把持の安定性などを評価することが可能であれば，それらを式（11）の評価関数に組み込むことは可能である．例えば，把持計画ソフトウェアである GraspIt!でも用いられているような把持の安定性を評価する手法を適用すれば [20] [21]，与えられた終点姿勢を用いた際の把持の安定性を評価することができ，これを目的関数に用いることができると考えられる．把持タスクへの適用という点では，初期に与える終端姿勢を把持計画問題などから算出し，そのうえで提案手法での自由回転軸を与えて軌道を最適化するという方法も考えられる．

一方で，本論文に示したシミュレーションによる評価において，式（11）のコスト関数では平滑化の評価はしていない．終点のコンフィギュレーションの評価に軌道の平滑度を組み込むには，それぞれの終点位置について軌道を最適化し，その軌道の平滑度を評価するのが自然と考えられるが，そのような処理は計算時間が膨大になると考えられる．このため，軌道の平滑度等，軌道全体の評価を終点の最適化に組み込むには，提案手法とは異なる工夫が必要になると考えられ，改善の余地がある点である．

提案手法は，終点における自由回転軸が与えられるような軌道生成の問題であれば，様々な適用可能であると考えられる．長軸の物体に限らず，平板を把持する問題にも適用可能である．また長軸用の物体を把持した状態で，その物体を移動させる，差し込む等の動作の軌道計画にも利用できると考えられる．

7. 結 論

本研究では，多峰性を考慮した最適化を用いることで，複数の終点姿勢を導き出す手法を提案した．また，軌道の始点および

終点の姿勢を局所的に最適化するアルゴリズムを提案した．提案手法を二次元平面上で動作する 4 リンクマニピュレータ，および三次元空間上で動作する 6 軸マニピュレータの軌道計画に適用しその性能を評価した．シミュレーションにおける評価から，提案手法によって干渉のない複数の軌道を得ることができることが確認された．従来の軌道計画法では，単一の解を見つけることのみを対象としているものがほとんどであり，本研究の成果は，軌道計画において複数の解を求めるという点で，新規性があるものであると考えられる．

謝　辞　この研究は古河電気工業との社会連携講座における研究の一環として行われました．また，査読者による建設的なコメントに感謝いたします．

参　考　文　献

[1] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg and P. Abbeel: “Motion planning with sequential convex optimization and convex collision checking,” Int. J. Robotics Res., vol.33, no.9, pp.1251–1270, 2014.
[2] M. Zucker, N. Ratliff, A. Dragan, M. Pivtoraiko, M. Klingensmith, C. Dellin, J.A. Bagnell and S. Srinivasa: “CHOMP: Covariant hamiltonian optimization for motion planning,” Int. J. Robotics Res., vol.32, pp.1164–1193, 2013.
[3] S. Karaman and E. Frazzoli: “Sampling-based algorithms for optimal motion planning,” Int. J. Robotics Res., vol.30, pp.846–849, 2011.
[4] S.M. LaValle: Planning Algorithms. Cambridge University Press, New York, NY, USA, 2006.
[5] S.M. LaValle and J.J. Kuffner: “Randomized kinodynamic planning,” Int. J. Robotics Res., 2001.
[6] T. Osa and M. Sugiyama: “Hierarchical policy search via return-weighted density estimation,” Proc. AAAI conf. Artificial Intelligence (AAAI), 2018.
[7] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor and S. Schaal: “STOMP: Stochastic trajectory optimization for motion planning,” Proc. IEEE Int. Conf. Robot. Autom. (ICRA), pp.4569–4574, 2011.
[8] L.E. Kavraki, P. Svestka, J.C. Latombe and M.H. Overmars: “Probabilistic roadmaps for path planning in high-dimensional conguration spaces,” IEEE Transactions on Robotics and Automation, vol.12, no.4, pp.566–580, 1996.
[9] L.E. Kavraki, M.N. Kolountzakis and J.C. Latombe: “Analysis of probabilistic roadmaps for path planning,” IEEE Transactions on Roborics and Automation, vol.14, pp.166–171, 1998.
[10] A.J. Ijspeert, J. Nakanishi and S. Schaal: “Learning attractor landscapes for learning motor primitives,” Advances in Neural Information Processing Systems (NIPS), 2002.
[11] A. Paraschos, C. Daniel, J. Peters and G. Neumann: “Probabilistic movement primitives,” Advances in Neural Information Processing Systems (NIPS), 2013.
[12] T. Osa, N. Sugita and M. Mitsuishi: “Online trajectory planning and force control for automation of surgical tasks,” IEEE Transactions on Automation Science and Engineering, vol.15, no.2, pp.675–691, 2018.
[13] T. Osa, J. Pajarinen, G. Neumann, J.A. Bagnell, P. Abbeel and J. Peters: “An algorithmic perspective on imitation learning,” Foundations and Trends in Robotics, vol.7, no.1–2, pp.1–179, 2018.
[14] A.D. Dragan, K. Muelling, J. Andrew Bagnell and S.S. Srinivasa: “Movement primitives via optimization,” Proc. IEEE Int. Conf. Robot. Autom. (ICRA), pp.2339–2346, 2015.
[15] D. Koert, G.J. Maeda, R. Lioutikov, G. Neumann and J. Peters: “Demonstration based trajectory optimization for generalizable robot motions,” Proc. Int. Conf. Humanoid Robots (Humanoids), pp.515–522, 2016.
[16] T. Osa, E.A.M. Ghalamzan, R. Stolkin, R. Lioutikov, J. Peters and G. Neumann: “Guiding trajectory optimization by demonstrated distributions,” IEEE Robotics and Automation Letters, pp.819–826, 2017.
[17] G. Ye and R. Alterovitz: “Demonstration-guided motion planning,” Proc. Int. Symp. Robotics Res. (ISRR), 2011.
[18] A. Dragan, N. Ratli and S. Srinivasa: “Manipulation planning with goal sets using constrained trajectory optimization,” Proc. of the IEEE International Conference on Robotics and Automation (ICRA), pp.4582–4588, 2011.
[19] E. Rohmer, S.P.N. Singh and M. Freese: “V-rep: a versatile and scalable robot simulation framework,” Proc. Int. Conf. Intelligent Robots and Systems (IROS), pp.1321–1326, 2013.
[20] A.T. Miller and P.K. Allen: “Graspit! a versatile simulator for robotic grasping,” IEEE Robotics & Automation Magazine, vol.11, no.4, pp.110–122, 2004.
[21] C. Goldfeder and P.K. Allen: “Data-driven grasping,” Autonomous Robots, vol.31, pp.1–20, 2011.

長　隆之（Takayuki Osa）

東京大学大学院工学系研究科機械工学専攻特任講師．2015 年東京大学大学院にて博士号取得．2015 年 4 月より 2017 年 3 月まで，ダルムシュタット工科大学にてポスドク研究員．2017 年 4 月より東京大学大学院新領域創成科学研究科特任助教．2018 年 4 月より東京大学工学系研究科機械工学専攻特任講師．2019 年 3 月より九州工業大学生命体工学研究科人間知能システム工学専攻准教授．　（日本ロボット学会正会員）

森木和也（Kazuya Moriki）

古河電気工業株式会社ものづくり改革本部所属．2017 年 3 月名古屋大学大学院博士前期課程修了．修士（理学）．2017 年 4 月より現職．データ解析を通じた生産プロセスの制御・予測に従事．機械学習を用いた最適化制御などに関心．

杉田直彦（Naohiko Sugita）

東京大学大学院工学系研究科機械工学専攻教授．1996 年 3 月東京大学大学院工学研究科産業機械工学専攻修士課程修了．同年 4 月日本電気（株）入社，2003 年 9 月東京大学助手，2007 年 3 月同准教授となり，2014 年より現職．生産工学，医用工学の研究に従事．博士（工学）．日本機械学会などの会員．

佐藤雅也（Masaya Sato）

古河電気工業株式会社ものづくり改革本部所属．1994 年 3 月中央大学理工学部精密機械工学科卒業．2008 年 4 月より現職．ロボットを中核とした工程改善，自動機開発に従事．

杉山　聡（Satoshi Sugiyama）

古河電気工業株式会社ものづくり改革本部所属．1996 年 3 月東京大学大学院修士課程修了．修士（工学）．光ファイバ製造プロセスの開発，光通信用レーザーダイオードモジュールの開発等を経て 2017 年 4 月より現職．ロボットインテグレートによる自動生産技術の開発に従事．

中尾政之（Masayuki Nakao）

1983 年東京大学大学院工学系研究科産業機械工学専攻修士課程修了．同年日立金属株式会社勤務，1989 年 HMT Technology Corp. に出向，1992 年東京大学大学院工学系研究科産業機械工学専攻助教授，2001 年東京大学工学部附属総合試験所教授を経て，2006 年より東京大学大学院工学系研究科機械工学専攻教授．専門は生産技術，ナノ・マイクロ加工，加工の知能化と情報化，創造設計と脳科学，失敗学．