

マニピュレータへの動作教示作業簡単化のための 接触情報にもとづく経路生成

○高見理瑛 笠原佑太 Francisco Jesús Arjonilla García 早川智洋 小林祐一 (静岡大学)
竹下啓嗣 和田侑也 中村陽一郎 (芝浦機械)

Manipulator Path Generation Based on Contact Information to Simplify Motion Teaching

*R. Takami Y. Kasahara F. J. Arjonilla García T. Hayakawa and Y. Kobayashi (Shizuoka University) K. Takeshita Y. Wada and Y. Nakamura (Shibaura Machine)

Abstract— There is a demand for robots that can work in place of humans with the problem of labor shortages in Japanese industrial sites. However, motion teaching to the robot is very time consuming with conventional methods and a heavy burden on the instructor. Therefore, we aim to make the motion path generation automatic. The path generation generates a path that does not contact any obstacles. In our proposed method, the instructor only needs to set the transit point and parameters, and the path is automatically corrected based on the force information obtained at the time of contact. The parameter adjustment process is difficult, so it will be necessary to consider making it automatic in the future.

Key Words: Manipulator, Path generation, Obstacle avoidance

1 緒言

生産現場では、加工や組み立てなどのように直接的に生産にかかわる正味作業とそれらの正味作業で使用する部品や材料の運搬などの付随作業がある。正味作業はすでにロボットにより自動化されているものも多くみられるが、付随作業は対象物や保管場所が多様なため自動化が正味作業に比べて難しく、ほとんどの現場で人海戦術で行われている。近年、日本の生産現場では人口減少に伴う労働力の不足が問題となっており、付随作業の自動化の需要が高まっている。ロボットは導入時に動作教示作業が必要である。主な教示方法としては、ロボット自体を人が手で動かしてその動作を記憶させるダイレクトティーチングや3DCADなどで動作環境やロボットのモデルを作成してその中で動作計画を行う方法があるが、どちらも作業時間が長く教示者の負担が大きいことが問題となっている。前者はロボットの操作性の問題から動作の微修正が難しく、障害物や対象物が近くにあり動作の細かな修正が必要な場合は何度も教示しなおす必要があるため時間がかかる。後者はモデルを作成する作業やモデルと現実との位置合わせの作業に時間がかかる。そこで、人による教示と動作計画を組み合わせた、簡単にできる最小限の教示(始点、終点、経由点のみの教示)から環境情報をロボットが自分で取得して動作生成を自動で行う方法を提案する。

Laha et al. は一度成功した教示経路と設定した初期位置と目標位置の情報をもとに、同じ動作特徴を持つタスクの動作経路を自動で生成する方法を提案した¹⁾。この研究では障害物情報を視覚センサにより取得することが検討されている。しかし、視覚センサでの環境認識では視覚センサの設置位置により死角となる部分の障害物の検知ができないため、障害物回避が不可能となる場合があると考えられる。

そこで、本研究では荒井らの研究²⁾³⁾⁴⁾⁵⁾を先行研究として、ロボット動作時の環境との接触による力覚セ

ンサ情報により環境情報を取得する方法の提案を目的とする。ロボットに実際に教示経路追従をさせて環境との接触時に得られる力覚センサ情報をもとに障害物の位置を推定、障害物に関するコスト関数を定義して最小化問題を解くことで経路修正を自動で行う。先行研究では手先位置姿勢空間における障害物回避経路生成に成功したが、環境との接触は手と把持物体のみで発生するとしており、肘などの手以外の部位で接触した場合に経路修正ができないことが課題として挙げられる。本論文では先行研究を拡張し、手先位置空間と関節角度空間の両方でコスト関数の定義を行うことで、手と把持物体だけでなく腕での接触を回避した経路自動修正方法を提案する。

2 問題設定

本研究では3次元6自由度のマニピュレータの手先姿勢を固定とした手先位置の3自由度の動作経路の生成問題を扱う。教示者の設定した経路を自動で修正して、マニピュレータ自身(腕と手)と把持物体が環境(障害物)と接触しない経路を生成することを目標とする(Fig. 1 参照)。

経路の設定は、始点、終点、経由点を設定することで行い、経路曲線はそれらの点をスプライン補間することで得る。教示者による経路の設定は初期経路の一度のみとして、経路修正は自動で行う。経路修正では環境モデルなどの事前情報は使わず、センサより得られる経路追従時の関節角度と関節トルクの情報を用いる。具体的には、関節トルクを用いて障害物との接触を判定し、接触判定時の関節角度と関節トルクの情報(接触情報)を用いる。

3 接触情報を用いた経路修正方法

本研究では一度の経路教示と経路追従時の障害物との接触により得られるセンサ情報(接触情報)を用いた経路修正により、障害物との接触を回避した経路を生成する。本節では、経路曲線の表現方法と接触情報、

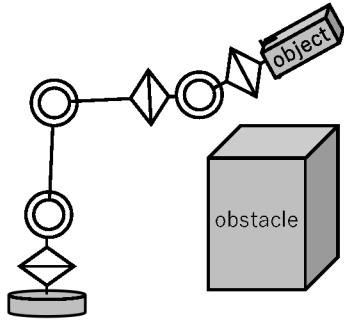


Fig. 1: Manipulator with object.

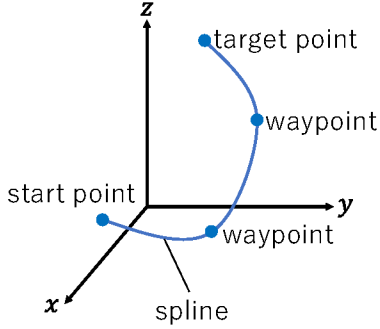


Fig. 2: Example of motion path curve.

接触情報を用いた経路修正方法について説明する。

3.1 経路曲線の表現方法と接触情報

動作経路は xyz 空間の曲線として表現する (Fig. 2 参照). 経路曲線の形は N 個の制御点 $\mathbf{x} \in \mathbb{R}^3$ の集合 $\mathbf{X} = \{\mathbf{x}_i | i = 1, \dots, N\}$ (制御点列) により設定し, 制御点列 \mathbf{X} をスプライン補間することにより曲線とする.

経路修正で使用する接触情報 \mathbf{c} は, 接触判定時の関節角度 $\boldsymbol{\theta}_c \in \mathbb{R}^6$ と接触により関節に作用した力 $\boldsymbol{\tau}_c \in \mathbb{R}^6$ ととして, 次式のように定義する.

$$\mathbf{c} = (\boldsymbol{\theta}_c, \boldsymbol{\tau}_c) \quad (1)$$

これらは1回の接触で1組得られ, i 回目の接触情報を \mathbf{c}_i と表現する. 本研究ではマニピュレータを低速で動作させることを想定して, 慣性力, 遠心力・コリオリ力は無視できるものとし, 接触により関節に作用したトルク $\boldsymbol{\tau}_c$ を関節トルクのセンサにより測定されたトルク $\boldsymbol{\tau}_s \in \mathbb{R}^6$ から自重により関節に作用するトルク $\boldsymbol{\tau}_g \in \mathbb{R}^6$ を引いた値とする. 自重によるトルク $\boldsymbol{\tau}_g$ はマニピュレータの各リンクの重心位置と質量, 関節角度から計算により求めることができる.

3.2 経路生成方法

経路生成手順を Fig. 3 に示す. 始めに教示者がロボットの動作環境を見て初期経路の制御点列 \mathbf{X}^{init} の設定を行い, それを追従経路の制御点列 $\mathbf{X}^{\text{follow}}$ に設定する. マニピュレータにこの制御点列 $\mathbf{X}^{\text{follow}}$ より得られる経路を追従させ, 障害物との接触判定を行う. 接触と判定した場合は接触情報 \mathbf{c}_1 を取得し, その情報を接触情報履歴 \mathbf{C} に記録する. 接触 N_c 回目の場合の接触情報履歴 \mathbf{C} は次のようになる.

$$\mathbf{C} = \{\mathbf{c}_i | i = 1, \dots, N_c\} \quad (2)$$

接触情報履歴 \mathbf{C} を反映した目的関数 L について最適化問題を解くことで制御点列 \mathbf{X} を変更, 経路の修正を行

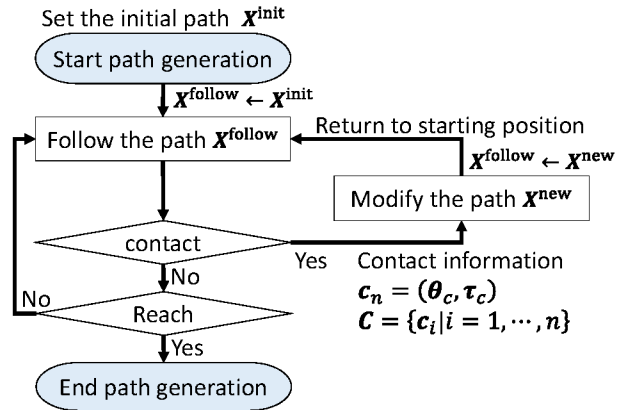


Fig. 3: Path generation process.

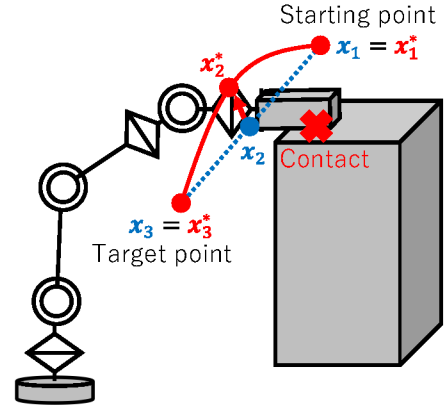


Fig. 4: Path Modification.

う. ただし, 経路の始点と終点は固定として経由点のみ変更する (Fig. 4 参照). 本研究では, 最適化問題の解法として Nelder-Mead 法⁶⁾を採用している.

$$\mathbf{X}^* = \arg \min_{\mathbf{X}} L(\mathbf{X}, \mathbf{C}) \quad (3)$$

この最適化計算により得られる制御点列 \mathbf{X}^* を追従経路の制御点列 $\mathbf{X}^{\text{follow}}$ に設定し, 経路の始点にロボットを戻してから再びこの制御点列 $\mathbf{X}^{\text{follow}}$ より得られる経路を追従させる. 接触していないと判定した場合は目標位置到達判定を行い, 目標位置 (経路の終点) に到達した時点で経路生成を終了する.

経路修正で用いる目的関数 L は, 以下の5項目に関するコスト関数 $G_i (i = 1, 2, 3, 4, 5)$ で構成されている.

- 障害物位置
- 制御点修正量
- 経路長
- 隣接制御点との距離
- リンク長さによる手先可動域

コスト関数 G_1 は接触情報履歴 \mathbf{C} を反映して一度接触した場所での接触を回避するためのコスト関数, コスト関数 $G_i (i = 2, 3, 4)$ は経路の形状を補正するためのコスト関数, コスト関数 G_5 は手先位置経路の生成範囲を手先可動域内に制限するためのコスト関数である. 各関数にそれぞれ重み $\alpha_i (i = 1, 2, 3, 4, 5)$ を設定して, コスト関数と重みの積をとり, それらの和を目的関数 L とする.

$$L(\mathbf{X}, \mathbf{C}) = \alpha_1 G_1(\mathbf{X}, \mathbf{C}) + \sum_{i=2}^5 \alpha_i G_i(\mathbf{X}) \quad (4)$$

以下では、各コスト関数の定義を説明する。

(a) 障害物位置

接触情報を用いて障害物位置で高くなるコスト g_1 を定義し、 g_1 の高い位置を通ると大きくなるようにコスト関数 G_1 を定義する。そうすることで、 g_1 の高い位置（障害物位置）を避ける経路を生成する。具体的には、経路上に評価点 $\mathbf{x}_k^q (k=1, \dots, N_q)$ をとり、各評価点におけるコスト g_1 の和をとったものをコスト関数 G_1 としている。評価点の集合 $\mathbf{X}^q = \{\mathbf{x}_k^q \in \mathbb{R}^3 \mid k=1, \dots, N_q\}$ は制御点列 \mathbf{X} をスプライン補間して得られる曲線上の点の集合であるので、制御点列 \mathbf{X} と評価点数 N_q より求められる ($\mathbf{X}^q = \text{S}_P(\mathbf{X}, N_q)$)。

$$G_1(\mathbf{X}, \mathbf{C}) = \sum_{k=1}^{N_q} \left[\max_{i=1, \dots, N_c} g_1(\mathbf{x}_k^q, \mathbf{c}_i) \right],$$

$$\mathbf{x}^q \in \mathbf{X}^q, \mathbf{X}^q = \text{S}_P(\mathbf{X}), \mathbf{c} \in \mathbf{C} \quad (5)$$

次に、コスト g_1 の定義について説明する。コスト g_1 は、手先や把持物体で接触した場合と腕で接触した場合とで異なる空間で定義する。Fig. 5 のように手先や把持物体で障害物と接触した場合、接触時の手先の位置から手先位置に作用した並進力と反対の方向に障害物があると予測できるので、コスト g_1 を手先位置空間で定義する。しかし、Fig. 6 のように腕で接触した場合は接触した部位より手先側にある関節の動きを変化させても接触を回避することはできない。そのため、腕で接触した場合は接触部位より手先側にある関節は無関係と考え、その他の接触に関係している関節角度空間でコストを定義する。Fig. 6 の場合は $\theta_1 \theta_2 \theta_3$ 空間でコストを定義する。このようにコスト g_1 の定義空間を接触部位に応じて変えるためにはどの部位で接触したかを判定する必要がある。接触部位の判定は接触情報の関節トルクの情報を用いて行っている。

手先や把持物体で接触した場合は、前述の通りに接触時の手先の位置から手先位置に作用した並進力と反対の方向に障害物があると予測できるので、手先位置姿勢空間で接触時の手先位置から並進力方向にコスト g_1 が小さくなるように定義する。また、接触位置から離れた場所では障害物との接触は発生しないと考え、並進力に対して垂直な方向へは離れるほどコスト g_1 が小さくなるように定義する。接触時の手先位置 \mathbf{x}_c は接触情報 \mathbf{c} の θ_c を用いて順運動学により求められ、接触時の手先位置での並進力 \mathbf{f}_c は接触情報 \mathbf{c} の θ_c と $\boldsymbol{\tau}_c$ を用いて仮想仕事の原理により求められる ($(\mathbf{x}_c, \mathbf{f}_c)^T = \text{S}_T(\mathbf{c})$)。 a_x, γ_x, σ_x はパラメータ（定数）である。手先または把持物体で接触した場合のコスト g_1 を Fig. 7 にカラー分布で示す。

$$g_1(\mathbf{x}^q, \mathbf{c}) = \prod_{i=1}^3 g_{1,i}(\mathbf{x}^q, \mathbf{c}) \quad (6)$$

$$g_{1,1}(\mathbf{x}^q, \mathbf{c}) = \frac{1}{1 + \exp[a_x z_1(\mathbf{x}^q | \mathbf{c})]} + \frac{1}{1 + \exp[-a_x z_1(\mathbf{x}^q | \mathbf{c}) + \gamma_x]} - 1 \quad (7)$$

$$g_{1,i}(\mathbf{x}^q, \mathbf{c}) = \exp\left[-\frac{z_i^2(\mathbf{x}^q | \mathbf{c})}{2\sigma_x^2}\right], \quad i = 2, 3 \quad (8)$$

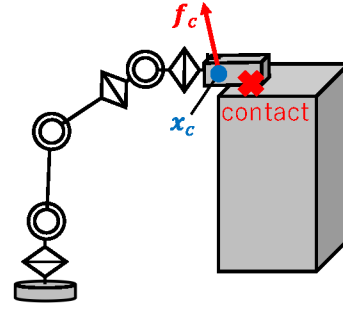


Fig. 5: Example of contact with holding object.

ただし、 $z_i(\mathbf{x}^q | \mathbf{c}) (i = 1, 2, 3)$ は次式より得る。 \perp は垂直ベクトルを表す記号である。

$$z_1(\mathbf{x}^q | \mathbf{c}) = \frac{\mathbf{f}}{\|\mathbf{f}\|} \cdot (\mathbf{x}^q - \mathbf{x}_c), \quad (\mathbf{x}_c, \mathbf{f}_c)^T = \text{S}_T(\mathbf{c}) \quad (9)$$

$$z_i(\mathbf{x}^q | \mathbf{c}) = \left(\frac{\mathbf{f}}{\|\mathbf{f}\|}\right)^{\perp i} \cdot (\mathbf{x}^q - \mathbf{x}_c),$$

$$(\mathbf{x}_c, \mathbf{f}_c)^T = \text{S}_T(\mathbf{c}), \quad i = 2, 3 \quad (10)$$

腕で接触した場合は接触時の関節角度から関節に作用したトルクの方向と反対方向に障害物があると考えられるので、接触に関係している n 次元の関節角度空間で接触時の関節角度から関節トルクの方向にコスト g_1 が小さくなるように定義する。また、接触時の姿勢と異なる姿勢であれば接触は発生しないと考え、関節トルクに垂直な方向へは離れるほどコスト g_1 が小さくなるように定義する。経路は手先位置空間で定義されているが、手先位置 \mathbf{x} は逆運動学を用いて関節角度 $\boldsymbol{\theta}$ に変換することができる ($\boldsymbol{\theta}^q = \text{I}_K(\mathbf{x}^q)$)。本研究では関口らの提案している逆運動学解法⁷⁾を用いている。 $a_\theta, \gamma_\theta, \sigma_\theta$ はパラメータ（定数）である。腕で接触した場合のコスト g_1^θ を Fig. 8 にカラー分布で示す。

$$g_1(\mathbf{x}^q, \mathbf{c}) = g_1^\theta(\boldsymbol{\theta}^q, \mathbf{c}), \quad \boldsymbol{\theta}^q = \text{I}_K(\mathbf{x}^q) \quad (11)$$

$$g_1^\theta(\boldsymbol{\theta}^q, \mathbf{c}) = \prod_{i=1}^3 g_{1,i}^\theta(\boldsymbol{\theta}^q, \mathbf{c}) \quad (12)$$

$$g_{1,1}^\theta(\boldsymbol{\theta}^q, \mathbf{c}) = \frac{1}{1 + \exp[a_x z_1(\boldsymbol{\theta}^q | \mathbf{c})]} + \frac{1}{1 + \exp[-a_x z_1(\boldsymbol{\theta}^q | \mathbf{c}) + \gamma_x]} - 1 \quad (13)$$

$$g_{1,i}^\theta(\boldsymbol{\theta}^q, \mathbf{c}) = \exp\left[-\frac{z_i^2(\boldsymbol{\theta}^q | \mathbf{c})}{2\sigma_x^2}\right], \quad i = 2, \dots, n \quad (14)$$

$$(15)$$

ただし、 $z_i(\boldsymbol{\theta}^q | \mathbf{c}) (i = 1, \dots, n)$ は次式より得る。

$$z_1(\boldsymbol{\theta}^q | \mathbf{c}) = \frac{\boldsymbol{\tau}_c}{\|\boldsymbol{\tau}_c\|} \cdot (\boldsymbol{\theta}^q - \boldsymbol{\theta}_c), \quad (\boldsymbol{\theta}_c, \boldsymbol{\tau}_c) \in \mathbf{c} \quad (16)$$

$$z_i(\boldsymbol{\theta}^q | \mathbf{c}) = \left(\frac{\boldsymbol{\tau}_c}{\|\boldsymbol{\tau}_c\|}\right)^{\perp i} \cdot (\boldsymbol{\theta}^q - \boldsymbol{\theta}_c),$$

$$(\boldsymbol{\theta}_c, \boldsymbol{\tau}_c) \in \mathbf{c}, \quad i = 2, \dots, n \quad (17)$$

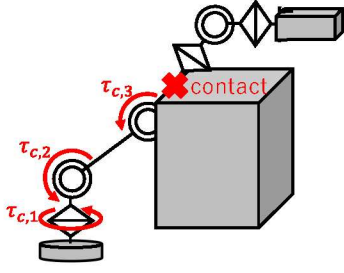


Fig. 6: Example of contact with arm.

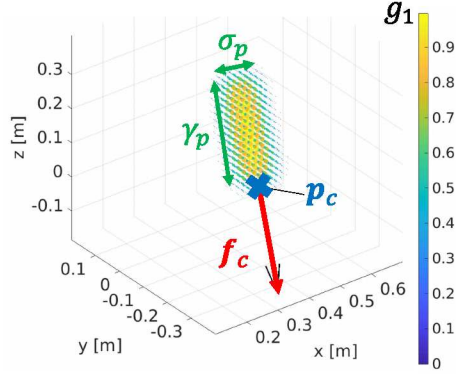


Fig. 7: Cost of contact with hand or holding object ($a_x = 50, \gamma_x = 0.30, \sigma_x = 0.05$).

(b) 制御点修正量

本研究における経路自動修正では経路の細かな修正を行うことを目的としているため、初期経路に対して経路の形状変化が大きいと経路生成の障害となる場合がある。そのため、初期経路の制御点列 \mathbf{X}^{init} と生成経路の制御点列 \mathbf{X} との差が大きいくほどコストが大きくなるようにコスト関数 G_2 を定義する。

$$G_2(\mathbf{X}) = \sum_{i=1}^N \|\mathbf{x}_i - \mathbf{x}_i^{\text{init}}\| \quad (18)$$

(c) 経路長

マニピュレータの無駄な動作は制御時に問題発生の原因となる場合がある。したがって、無駄な動作を削減するために動作経路を短くする必要がある。そこで、経路長が大きいくほどコストが大きくなるようにコスト関数 G_3 を定義する。

$$G_3(\mathbf{X}) = \sum_{k=1}^{N_q-1} \|\mathbf{x}_{k+1}^q(\mathbf{X}) - \mathbf{x}_k^q(\mathbf{X})\| \quad (19)$$

\mathbf{x}^q は経路上の評価点であり、コスト関数 G_1 で用いたものと同じものを用いる。

(d) 隣接制御点との距離

制御点位置を変更することで隣り合う制御点同士が接近もしくは重なり、経路生成の妨げとなる場合がある。そこで、隣接する制御点の間隔が均等なほどコストが小さくなるようにコスト関数 G_4 を定義する。

$$G_4(\mathbf{X}) = \sum_{i=1}^{N-1} |\Delta x_i - \Delta x_{\text{ave}}| \quad (20)$$

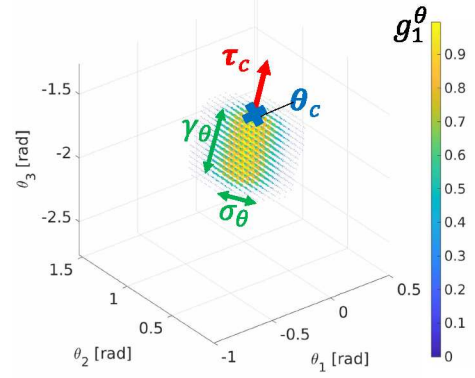


Fig. 8: Cost of contact with arm ($a_\theta = 50, \gamma_\theta = \frac{\pi}{18}, \sigma_\theta = \frac{\pi}{6}$).

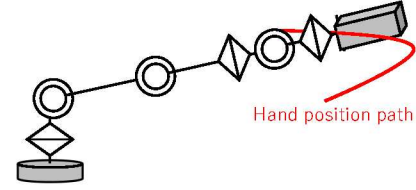


Fig. 9: Path outside the motion area of the hand.

ただし、以下の通りとする。

$$\Delta x_i = \|\mathbf{x}_{i+1} - \mathbf{x}_i\| \quad (21)$$

$$\Delta x_{\text{ave}} = \frac{1}{N-1} \sum_{j=1}^{N-1} \Delta x_j \quad (22)$$

(e) リンク長さによる手先可動域

マニピュレータの手先位置はリンクの長さにより動作可能な範囲が決まる。Fig. 9 のようにマニピュレータの手先位置の可動範囲の外まで経路が続く場合経路追従ができない。このような場合は接触しているわけではないので接触情報が得られず、経路を修正できない。そのため、手先可動域内で手先位置経路の生成できるように、手先の可動範囲をコスト関数を用いて制限する。 a_m はパラメータ（定数）である。

$$G_5(\mathbf{X}) = \sum_{k=1}^{N_q} \frac{1}{1 + \exp \left[-a_m \left(\|\mathbf{x}^q - \mathbf{x}_{l_2}\| - \sum_{i=2}^6 l_i \right) \right]} \quad (23)$$

$$\mathbf{x}_{l_2} = (0, 0, l_0 + l_1)^T \quad (24)$$

$l_i (i = 0, \dots, 6)$ は各リンクの長さである (Fig. 10 参照)。

4 シミュレーション

4.1 目的と条件

マニピュレータの腕と手、把持物体での障害物との接触を回避した経路生成方法としての提案手法の利用可能性の検証を目的として、オープンソースの物理エンジンの MuJoCo⁸⁾ を用いて 3次元 6自由度マニピュレータの PD 制御による動作経路追従シミュレーションを行った。マニピュレータの各関節に角度センサとトルクセンサを設置し、センサ情報より接触判定と接触情報の取得をした。腕での接触回避のためのコスト

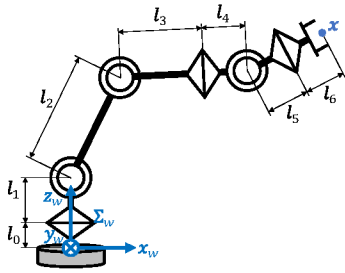


Fig. 10: Link length.

g_1^θ ありの方法となしの方法とでそれぞれ 10 パターンの動作生成を行い、成功率、経路修正回数について比較する。今回は経路修正回数の上限を 10 回として、10 回以内で障害物を回避した経路生成ができなかった場合は経路生成失敗とした。経路修正の計算で用いたパラメータの値を Table 1 に示す。

Table 1: Parameter.

Parameter	Value
α	(1, 15, 10, 5, 1)
a_x	100
γ_x	0.15
σ_x	0.05
a_θ	100
γ_θ	$\frac{\pi}{6}$
σ_θ	$\frac{\pi}{18}$
a_m	100
N_q	101

4.2 結果

経路生成の成功率を Table 2, 経路修正回数に関する箱ひげ図を Fig. 11 に示す。Table 2 より、腕での接触を含む場合はコスト g_1^θ ありの方が経路生成の成功率が高いことが確認できた。しかし、コスト g_1^θ ありでも経路生成に失敗する場合もあることが確認された。Fig. 11 より、コスト g_1^θ ありの方が経路修正回数が少ないことが確認できた。

Table 2: Success rate of path generation.

	With g_1^θ	Without g_1^θ
Rate [%]	80	50

4.3 考察

コスト g_1^θ ありで経路生成に成功し、なしで失敗した動作について生成されたコストを比較し、成功率の違いの原因について考察する。Fig. 12 のように接触したとき、コスト g_1^θ ありでは Fig. 13 のように関節角度空間でコスト g_1^θ が生成され、それを手先位置空間に変換したものを Fig. 14 に示す。コスト g_1^θ なしで手先位置空間に生成されたコスト g_1 を Fig. 15 に示す。Fig. 14 より追従経路に対してコストの分布が z 軸負の方向に寄っていることが確認でき、これにより z 軸負の方向への経路修正が制限される。一方で、Fig. 15 を見ると追従経路に対して平行にコストが分布しており回避方向に制限がない。この経路修正方向の制限の有無により経路修正回数に差が生まれ、経路修正回数の上限を設定したことで成功率に差が生じたと考えられる。そのため、コスト g_1^θ なしでも経路修正回数に制限がなければ経路生成に成功すると考えられ、実際に 23 回

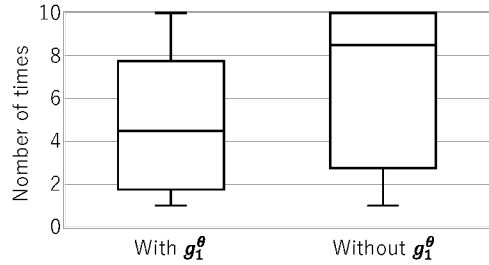


Fig. 11: Box-and-whisker plot of the number of path modifications.

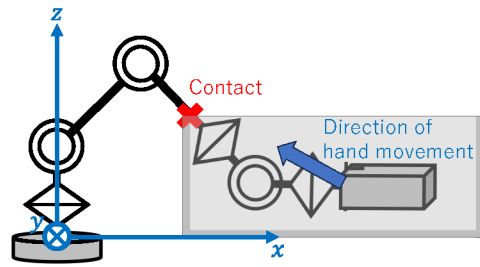


Fig. 12: Examples of contacts that succeeded with the methods with g_1^θ and failed with the methods without g_1^θ .

の経路修正で成功した。本研究ではロボットに経路を追従させて障害物と接触した際に得られるセンサ情報より経路の修正を行っているため、接触により環境やロボット自身が壊れないように低速で動作させており、経路追従には時間がかかる。コスト g_1^θ なしの方が経路修正の計算時間は短い修正回数も多く、経路追従時間が長い場合、経路生成にかかる総時間は経路修正回数が多いほうが長くなる。実際に本例では経路修正の平均時間はコスト g_1^θ なしの方が小さいが、経路生成に要した総時間はコスト g_1^θ ありに比べて長く (Table 3), 提案手法で経路修正回数を削減できる有用性が確認された。

Table 3: Average time spent per path modification and the number of times the path was modified and the total time taken to generate the path.

	With g_1^θ	Without g_1^θ
Time [ms/times]	1,448	185
Number of times [times]	2	23
Total time [ms]	137,395	367,538

次に、コスト g_1^θ ありで経路生成に失敗した原因について考察する。Fig. 16 に生成されたコストのカラー分布を示す。Fig. 16 を見ると初期経路と生成経路との間に差が見られず、生成されたコストの中心を経路が通っていることが確認できる。このことから、最適化計算の解法として Nelder-Mead 法⁶⁾を採用していることが原因であると考えられる。Nelder-Mead 法で最小化問題を解く場合は探索空間に複数の点を取り、その中で目的関数の値が最も大きい最悪点とその他の点の重心点を結ぶ直線方向に探索を進めるが、最悪点と重

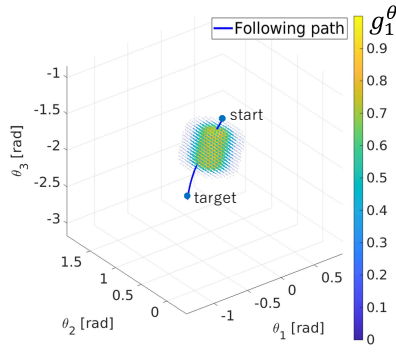


Fig. 13: Cost of joint angle space generated by the method with g_1^θ .

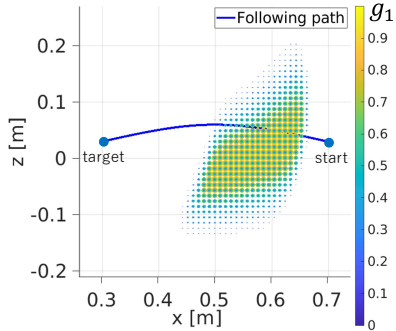


Fig. 14: Cost of joint angle space generated by the method with g_1^θ mapped to hand position space.

心点が重なる場合は探索方向が定まらず探索が進まない。本研究では初期経路の制御点列とその各要素（始点と終点は固定のため除く）を同じ距離だけ変更した要素数分の制御点列を用いて探索を行っている。初期経路がコストの中心を通ったことで初期経路の制御点列が最悪となり、その他の初期経路をもとに生成した制御点列の重心が初期経路の制御点列に一致して経路が修正できなかったと考えられる。そのため、最適化問題の解法を変更することで解決可能と考えられる。

5 結言

本研究では簡単にできる最小限の教示から環境情報をロボットが自分で取得して動作生成を自動で行う方法の提案を目的とし、ロボットが経路追従時に環境との接触により取得する力覚センサ情報を用いて定義したコスト関数の最小化による経路自動修正方法を提案した。本論文では先行研究の課題として挙げたロボットの腕での接触回避を含めた経路生成を手先位置空間と関節角度空間の両方でコスト関数の定義を行うことで実現した。検証はシミュレーション上で、姿勢を固定した3次元3自由度の動作生成について行い、腕での接触回避のためのコストありの方法となしの方法とで比較し、腕での接触回避のためのコストありの方が少ない修正回数、短時間での経路生成が可能であることが確認できた。また、腕での接触回避のためのコストありの方法でも障害物との接触を回避した経路生成ができない場合があることも確認された。コストの分布のグラフを確認したところ最適化問題の解法が原因で経路修正に失敗したと考えられ、他の解法に変更することで解決可能と考えられる。

今後の課題としては、前述した局所解に陥る問題に対して最適化計算の解法の検討、現在設計者が経験則

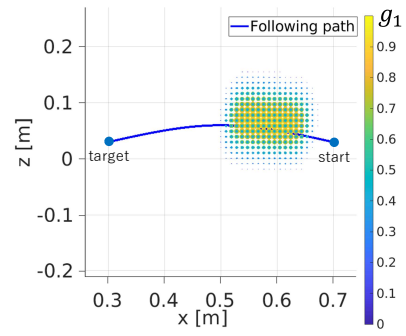


Fig. 15: Cost of hand position space generated by the method without g_1^θ .

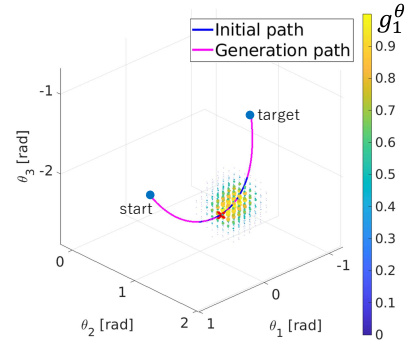


Fig. 16: Cost of failed path generation g_1^θ .

で行っているパラメータの設定の自動化またはマニュアル化が挙げられる。また、3次元6自由度の手先姿勢も含めた経路生成での検証、実機での検証も課題として挙げられる。

参考文献

- 1) R. Laha, R. Sun, W. Wu, D. Mahalingam, N. Chakraborty, L. F. C. Figueredo and S. Haddadin: Coordinate Invariant User-Guided Constrained Path Planning with Reactive Rapidly Expanding Plane-Oriented Escaping Trees, International Conference on Robotics and Automation (ICRA), 977/984 (2022)
- 2) 荒井康太, 石村芳暉, 小林祐一, 伊部公紀: マニピュレータへの作業教示時間短縮のための力覚情報を用いた経路生成およびそのメタパラメータの自動調整, 第33回自律分散システム・シンポジウム予稿集, 1A1-1 (2021)
- 3) 榊原典佑, 小林祐一, F. J. Arjonilla García: マニピュレータへの動作教示時間短縮のための経路最適化とメタパラメータの自動調整, 第49回知能システムシンポジウム講演資料, C3-3 (2022)
- 4) 高見理瑛, 小林祐一, F. J. Arjonilla García: 単純な動作教示と接触情報にもとづくマニピュレータの経路生成, ロボティクス・メカトロニクス講演会講演概要集 2023, 2A1-E14 (2023)
- 5) 笠原佑太, 高見理瑛, 小林祐一, F. J. Arjonilla García, 中村陽一郎, 竹下啓嗣: 接触情報にもとづく3次元空間における手先姿勢を考慮したマニピュレータの経路生成, ロボティクス・メカトロニクス講演会講演概要集 2024, 2A1-N05 (2024)
- 6) J. A Nelder and R. Mead: A simplex method for function minimization, The computer journal, 7 No. 4, 308/313 (1965)
- 7) 関口叡範, 武居直行: 仮想バネの弾性エネルギー最小化に基づいた特異姿勢・非可解な問題に対して安定な逆運動学の数値解法, 日本ロボット学会誌, 36 No. 9, 645/653 (2018)
- 8) E. Todorov, T. Erez and Y. Tassa: MuJoCo: A physics engine for model-based control, IEEE/RSJ International Conference on Intelligent Robots and Systems, 5026/5033 (2012)