

# 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

Manipulator path generation based on contact information  
in 3D space considering the posture of the manipulator's hand

○学 笠原佑太 (静岡大) 学 高見理瑛 (静岡大)  
正 小林祐一 (静岡大) ARJONILLA GARCIA F. J. (静岡大)  
中村陽一郎 (芝浦機械) 竹下啓嗣 (芝浦機械)

Yuta KASAHARA, Shizuoka University, kasahara.yuta.19@shizuoka.ac.jp

Rie TAKAMI, Yuichi KOBAYASHI, Shizuoka University

ARJONILLA GARCIA Francisco Jesus, Shizuoka University

Yoichiro NAKAMURA, Keiji TAKESHITA, Shibaura Machine

There is a demand for robots that can work in place of humans with the problem of labor shortages in Japan's industrial sites. When robots are introduced to a production site, it is necessary to teach them how to operate. However, the conventional teaching method places an extremely heavy burden on the instructor. To solve this problem, we propose a method in which the instructor only needs to set up the motion path transit points and parameters, and the path is automatically corrected based on the force information obtained at the time of contact. The motion path is a curve that takes posture in 3D space into account. In this study, we conducted simulations of the proposed method and investigated its applicability to manipulators.

**Key Words:** Teaching, Manipulator, Path generation, Obstacle avoidance

## 1 緒言

近年、人間との協働や人間の代わりに作業を行う生産支援ロボットの需要が高まりつつある。生産支援ロボットとは、生産ラインへの部品の供給や荷物の積み下ろしといった生産作業の手助けをするようなロボットである。このようなロボットは環境や作業対象物との間に接触や拘束を伴う作業であり、高度な物体操作を求められる。また、生産現場への導入にあたり現場ではロボットのための環境が整備されていないという問題がある。そのため、人間からロボットへの動作の教示が必要になる。従来の動作教示の方法としては教示者が直接ロボットを操作して動作教示するダイレクトティーチングなどがある [1]。また、他にモーションプランニングなどの自立型がある。しかし、これらの教示方法は教示者へある程度の技術が求められ教示者への負担が大きくなるという問題や障害物や対象物の事前環境モデルの構築が必要であり、その設定にコストと時間がかかるという問題がある。以上の背景から、本研究では生産支援ロボット導入時の教示者の負担軽減のためのロボットへの作業教示時間の短縮を目的とする動作経路生成の自動化について考える。

ロボットへの動作教示に関する研究として以下のようなものがある。Losey *et al.* はロボットの自律動作中に人間が物理的に介入して動作を修正し、その情報からロボットが最適な動作経路を生成する手法を提案し、LQR 制御を勾配降下法により拡張することで、リアルタイムに軌道を更新する学習ルールを構築した [2]。しかし、この研究では人間が直接動作教示を与えることやリアルタイムで動作教示を行うことが必要であり、ロボットへの作業教示時間の短縮の問題はまだ解決されていない。先行研究として荒井らは人間が与えた 2 次元の動作経路を障害物との接触から自動修正し接触情報から修正量に関わるパラメータを自動調整する手法の提案およびシミュレーションによる検証を行った [3]。また、榊原はこの経路生成手法の 3 次元空間への拡張とシミュレーションによる検証を行った [4]。また、高見らは姿勢を考慮した 2 次元空間でのロボットの手先の経路生成手法の提案およびシミュレーションによる検証を行った [5]。しかしながら、3 次元空間でのロボットの手先姿勢を考慮した経路生成は行われていない。

そこで本研究では、生産支援ロボット導入における動作教示者

の負担削減を目的とし、先行研究の発展としてシミュレーション上で 3 次元空間でのロボットの手先の位置姿勢に関する動作経路生成を行う。本研究では、ロボットへの動作教示に関して、最初に人間が初期経路を設定するだけで済むような簡単な教示方法を目指す。

## 2 問題設定

本研究では、3 次元空間上でのマニピュレータの手先の位置姿勢が追従する 6 次元の動作経路について考える。条件は、マニピュレータの手先の位置姿勢、スタート地点、目標地点が既知であり、障害物の位置、大きさは未知とする。マニピュレータは障害物との衝突時、マニピュレータのモータにある力覚センサにより接触力を計測できる。

次に、接触情報の利用について述べる。図 1 に接触情報の概要を示す。マニピュレータの手先が障害物に衝突したとき、マニピュレータの手先の位置から接触位置が、力覚センサから接触力がわかる。この時、接触した手先の位置姿勢を失敗点  $f \in \mathbb{R}^6$ 、接触力の方向を修正ベクトル  $m \in \mathbb{R}^3$  ( $\|m\| = 1$ ) と定義する。これら、失敗点  $f$  と修正ベクトル  $m$  は、1 回の接触に対して 1 組となり、接触情報  $C = (f, m)$  と定義する。また、接触情報履歴を  $H = \{C_i \mid i = 1, \dots, N_F\}$  ( $N_F$  は接触回数) と定義する。

本研究では、人から教示された初期経路ではマニピュレータがの手先が障害物に接触するという条件を想定し、接触時に環境から受ける力覚を利用して経路を自動修正し、環境との接触の無い動作経路を生成することを目的とする。

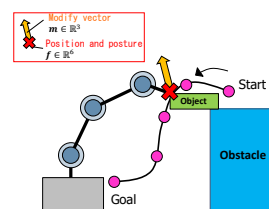


Fig.1 Contact information

### 3 方法

#### 3.1 姿勢の表現方法

姿勢の表現方法について説明する。把持する対象物は四元数を用いて表現する。初期姿勢からの回転を表す四元数を次式に示す ( $i, j, k$  は虚数)。ただし、回転軸となる単位ベクトルを  $v = (v_x, v_y, v_z)$ 、初期姿勢からの回転角を  $\theta$  とする。

$$q = \cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right)(v_x i + v_y j + v_z k) \quad (1)$$

また、以降では四元数  $q$  を  $q = (\cos(\theta/2), \sin(\theta/2)v)$  と記す。

#### 3.2 経路曲線の表現方法

動作経路の曲線の表現方法について説明する。動作経路の曲線と姿勢表現について図 2 に示す。動作経路の曲線はスプライン補間による曲線を用いる。動作経路は、手先の位置を  $x, y, z$ 、姿勢を  $\theta v_x, \theta v_y, \theta v_z$  で表現し、6次元の曲線で表現する。動作経路の経路点  $x \in \mathbb{R}^6$  を次式のように設定する。

$$x = [x \ y \ z \ \theta v_x \ \theta v_y \ \theta v_z]^T \quad (2)$$

動作経路には複数の経路点を用いる。複数の経路点の行列として、経路点列  $X = \{x_i \mid i = 1, \dots, N\}$  ( $x \in \mathbb{R}^6$  は経路点,  $N$  は経路点の数) を設定し、それらをスプライン補間した曲線で表現される。

#### 3.3 経路自動修正の手順

接触情報を用いた経路自動修正の手法について説明する。この手法は教示者が与えた初期経路を接触情報を用いて自動で修正し、障害物との接触の無い経路を生成する経路生成手法である。経路生成の手順を図 3 に示す。本手法では、複数の接触と経路の修正を繰り返して、障害物との接触の無い動作経路を生成する。始めに教示者が初期経路生成のため、初期経路の経路点列  $X^{\text{init}}$  を設定する。設定した  $X^{\text{init}}$  を追従経路の経路点列  $X^{\text{path}}$  に設定する。この経路点列により得られる動作経路をマニピュレータに追従させ、障害物との接触判定を行う。障害物と接触した場合接触情報  $C$  を取得し、その情報を接触情報履歴  $H$  に蓄積する。修正  $n$  回目の経路生成の場合の接触情報は  $C_n = (f_n, m_n)$ 、蓄積後の接触情報履歴は  $H = \{C_i \mid i = 1, \dots, n\}$  となる。接触情報履歴  $H$  を用いて経路の修正を行い、修正経路の経路点列  $X^{\text{new}}$  を追従経路の経路点列  $X^{\text{path}}$  に代入する。ロボットを初期位置に戻してから、再びこの経路点列  $X^{\text{path}}$  より得られる経路を追従させる。経路を追従し、障害物との接触がない場合は経路生成を終了する。

次に、経路の修正方法について説明する。接触情報履歴  $H$  を反映した目的関数  $L(X|H)$  を用いて、経路の最適化を行う。この目的関数  $L(X|H)$  が最小となる経路点列  $X^*$  を探索し、それを修正経路の経路点列  $X^{\text{new}}$  とする。

$$X^* = \arg \min_X L(X|H) \quad (3)$$

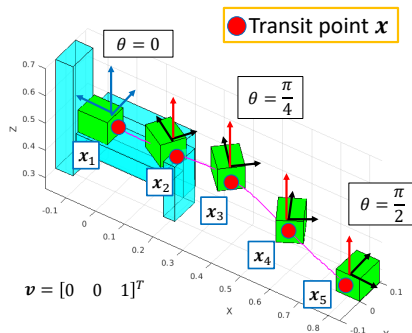


Fig.2 Example of posture

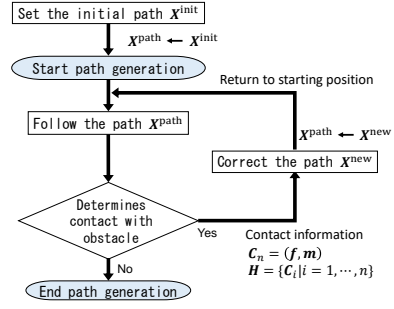


Fig.3 Process of path correction

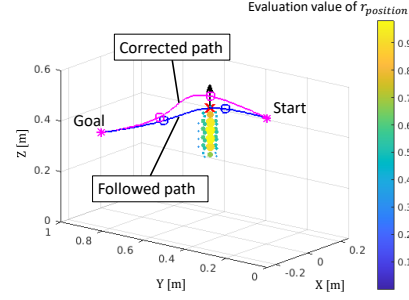


Fig.4 Example of distribution of  $r_{\text{position}}$

#### 3.4 経路の評価

目的関数 経路の評価に用いる目的関数の詳細を説明する。経路の評価基準は以下の 4 項目である。

1. 接触情報による評価
2. 初期経路の経路点と経路修正後の経路点との距離と回転量
3. 全体の経路長と回転量の総和
4. 隣接経路点間の間隔の均等さ

各評価基準から評価関数 ( $g_1, g_2, g_3, g_4$ ) を作成し、各関数にそれぞれ重み ( $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ ) をかけたものを、目的関数  $L$  とする。目的関数  $L$  を次式に示す。

$$L(X|C) = \alpha_1 g_1(X|C) + \alpha_2 g_2(X) + \alpha_3 g_3(X) + \alpha_4 g_4(X) \quad (4)$$

接触情報を反映させた経路の評価 接触情報を利用して設定した評価関数  $g_1$  について説明する。評価関数  $g_1$  は接触情報  $C = (f, m)$  を用いて、位置または姿勢を変化させることで障害物との接触の無い動作経路を生成することを目的とした評価関数である。ここで、失敗点  $f \in \mathbb{R}^6$  について、位置に関する要素について位置失敗点  $f_p \in \mathbb{R}^3$ 、姿勢に関する要素について姿勢失敗点  $f_o \in \mathbb{R}^3$  とする。また、評価点  $p \in \mathbb{R}^6$  について、位置に関する要素について位置評価点  $p_p \in \mathbb{R}^3$ 、姿勢に関する要素について姿勢評価点  $p_o \in \mathbb{R}^3$  とする。

始めに評価関数  $g_1$  の位置に関するコスト  $r_{\text{position}}$  について説明する。位置に関するコスト  $r_{\text{position}}$  の例を図 4 に示す。図 4 のように、コスト  $r_{\text{position}}$  は位置失敗点  $f_p$  を基準に修正ベクトル  $m$  と逆向き方向に大きな値をとるように設定してあり、動作経路の位置に関して障害物の無い方向へ修正するためのコストとなっている。以下で  $r_{\text{position}}$  の設定方法を説明する。まず、 $r_{\text{position}}$  の定義に必要な座標系の設定を行う。座標系は接触情報 (位置失敗点  $f_p$ 、修正ベクトル  $m$ ) を基準に設定を行う。位置失敗点  $f_p$  を原点とし、修正ベクトル  $m$  を 1 つの軸とする座標系について考える。残り 2 つの軸の方向ベクトルを  $m_{\perp 1}, m_{\perp 2}$  ( $\|m_{\perp 1}\| = 1, \|m_{\perp 2}\| = 1$ ) とする。そして、それぞれの軸変数を  $z_1, z_2, z_3$  とおく。経路上の位置評価点  $p_{p(k)}$  を用いて各軸変数の値は、次式で表される。

$$z_1(p_p, C) = m^T(p_p - f_p) \quad (f_p, m) \in C \quad (5)$$

$$z_2(p_p, C) = (m_{\perp 1})^T(p_p - f_p) \quad (f_p, m) \in C \quad (6)$$

$$z_3(p_p, C) = (m_{\perp 2})^T(p_p - f_p) \quad (f_p, m) \in C \quad (7)$$

次に、各軸変数に対する関数の設定を行う。まず、軸  $z_1$  の関数の設定を行う。軸  $z_1$  については、位置失敗点  $\mathbf{f}_p$  を中心として、修正ベクトル  $\mathbf{m}$  の方向にコストが減少する評価を表現するため、シグモイド関数を利用する。軸  $z_1$  に対する関数を次式に示す。ただし、 $a$ 、 $\gamma$  はゲイン（定数）である。

$$r_{\text{position1}}(\mathbf{p}_p, \mathbf{C}) = \frac{1}{1 + \exp\{a z_1(\mathbf{p}_p, \mathbf{C})\}} + \frac{1}{1 + \exp[-a\{z_1(\mathbf{p}_p, \mathbf{C}) + \gamma\}]} - 1 \quad (8)$$

次に、軸  $z_2$ 、 $z_3$  の関数の設定を行う。軸  $z_2$ 、 $z_3$  については、位置失敗点  $\mathbf{f}_p$  を中心として、修正ベクトル  $\mathbf{m}$  に垂直な方向にコストが減少する評価を表現するため、ガウス関数を利用する。軸  $z_2$ 、 $z_3$  に対する関数を次式に示す。ただし、 $\beta$  はゲイン（定数）である。

$$r_{\text{position2}}(\mathbf{p}_p, \mathbf{C}) = \exp\left(-\frac{1}{2} \mathbf{Z}(\mathbf{p}_p, \mathbf{C}) \mathbf{B}^{-1} \mathbf{Z}(\mathbf{p}_p, \mathbf{C})^T\right) \quad (9)$$

ここで、 $\mathbf{Z}$ 、 $\mathbf{B}$  は以下のように与えられる。

$$\mathbf{Z}(\mathbf{p}_p, \mathbf{C}) = [z_2(\mathbf{p}_p, \mathbf{C}) \quad z_3(\mathbf{p}_p, \mathbf{C})] \quad (10)$$

$$\mathbf{B} = \begin{bmatrix} \beta^2 & 0 \\ 0 & \beta^2 \end{bmatrix} \quad (11)$$

$$(12)$$

以上の2つの関数  $r_{\text{position1}}$ 、 $r_{\text{position2}}$  の積をとり、接触情報による位置に関するコスト  $r_{\text{position}}$  の分布を実現する。

$$r_{\text{position}} = r_{\text{position1}} r_{\text{position2}} \quad (13)$$

次に評価関数  $g_1$  の姿勢に関するコスト  $r_{\text{posture}}$  について説明する。コスト  $r_{\text{posture}}$  は接触したときの姿勢を基準に障害物がある方向に姿勢が回転すると大きな値をとるように設定しており、動作経路の姿勢に関して障害物との接触を回避する方向への姿勢変化（回転）を促すためのコストとなっている。以下で  $r_{\text{posture}}$  の設定方法を説明する。まず、 $r_{\text{posture}}$  の設定に必要な評価姿勢について説明する。接触姿勢したときの姿勢を  $\mathbf{q}_f = (\cos(\theta_f/2), \sin(\theta_f/2)\mathbf{v}_f)$ 、経路上の評価点の姿勢を  $\mathbf{q}_p = (\cos(\theta_p/2), \sin(\theta_p/2)\mathbf{v}_p)$  とする。そして、接触姿勢  $\mathbf{q}_f$  から経路上の姿勢  $\mathbf{q}_p$  までの回転を表す四元数を式 (14) のように計算することができ、 $\mathbf{q}_r = (\cos(\phi/2), \sin(\phi/2)\mathbf{v}_r)$  とする。この姿勢  $\mathbf{q}_r$  が評価姿勢となる。

$$\mathbf{q}_r = (\mathbf{q}_f)^{-1} \mathbf{q}_p \quad (14)$$

障害物との接触を回避する方向への姿勢変化を促すためのコスト  $r_{\text{posture}}$  を実現するために3つのコストを設定する必要がある。1つは評価姿勢の軸に関するコスト  $r_{o1}$ 、2つ目は評価姿勢の回転量に関するコスト  $r_{o2}$ 、3つ目は位置失敗点付近での姿勢の変化を促すためのコスト  $r_p$  である。以下でその詳細を説明する。

はじめに、評価姿勢の軸を評価するためのコスト  $r_{o1}$  について説明する。接触姿勢から障害物との接触を回避する方向に姿勢を回転させるためには、望ましい回転軸となるものが必要になる。 $r_{o1}$  は望ましい回転軸で姿勢を回転させるためのコストとなる。ここで、望ましい回転軸とするベクトル  $\mathbf{m}_o$  について定義する。接触情報を得たとき修正ベクトル（接触力の方向） $\mathbf{m}$  と手先位置から対象物上の基準となる点までのベクトル  $\mathbf{v}_s$  を用いて、以下のように定義する。

$$\mathbf{m}_o = \frac{1}{\|\mathbf{v}_m\|} \mathbf{v}_m \quad (15)$$

$$\mathbf{v}_m = \mathbf{m} \times \mathbf{v}_s \quad (16)$$

まず、コスト  $r_{o1}$  の軸変数の値  $z_{o1}$  を設定する。軸変数の値  $z_{o1}$  を次式のように設定することで、軸変数の値  $z_{o1}$  の値が1に近いほど評価姿勢の軸  $\mathbf{v}_r$  が望ましい回転軸  $\mathbf{m}_o$  に近いことを表現できる。

$$z_{o1}(\mathbf{q}_r) = \mathbf{m}_o \cdot \mathbf{v}_r \quad (17)$$

軸  $z_{o1}$  に対する関数  $r_{o1}$  の設定を行う。軸  $z_{o1}$  については、内積の値が1に近いほどコストが減少する評価を表現するため、シグモイド関数を利用する。軸  $z_{o1}$  に対する関数  $r_{o1}$  を次式に示す。ただし、 $b$  はゲイン（定数）、 $c$  はパラメータである。また、コストの値が逆になっている理由は後述する。

$$r_{o1}(\mathbf{v}_r) = \frac{1}{1 + \exp(-b(z_{o1} - c))} \quad (18)$$

次に、評価姿勢の回転量を評価するためのコスト  $r_{o2}$  について説明する。 $r_{o1}$  で設定した軸を回転軸として姿勢を回転させるためには、評価姿勢の回転量にもコストを設定する必要がある。まず、コスト  $r_{o2}$  の軸変数の値  $z_{o2}$  を次式のように設定する。

$$z_{o2} = \phi_p \quad (19)$$

軸  $z_{o2}$  に対する関数  $r_{o2}$  の設定を行う。軸  $z_{o2}$  については、回転量が大きいほどコストが減少する評価を表現するため、シグモイド関数を利用している。軸  $z_{o2}$  に対する関数  $r_{o2}$  を次式に示す。ただし、 $d$  はゲイン（定数）である。また、コストの値が逆になっている理由は後述する。

$$r_{o2}(\phi_p) = \frac{1}{1 + \exp(-d z_{o2})} \quad (20)$$

以上の設定した2つの関数  $r_{o1}$ 、 $r_{o2}$  を用いて、評価姿勢の軸が望ましい回転軸に近いかわ、回転量が大きいときにコストが減少する評価を表現するため、姿勢に関するコスト  $r_o$  を式 (21) のように設定する。このコストにより、望ましい軸を回転軸とした姿勢の回転が可能になる。また、 $r_o$  が式 (21) のようになっている理由について説明する。 $r_{o1}$  と  $r_{o2}$  の積だけであると、一方のコストが大きく、もう一方のコストが小さい場合に、コストが小さくなってしまふという問題がある。そのため、 $r_{o1}$ 、 $r_{o2}$  の設定の段階で、 $r_{o1}$  に関しては内積の値が1に近いほどコストが増加し、 $r_{o2}$  に関しては回転量が大きいほどコストが増加するように設定してある。こうすることにより、最後に1から  $r_{o1}$  と  $r_{o2}$  の積を引くことで、先述した問題が解決され、望ましいコストの表現が可能になる。

$$r_o = 1 - r_{o1} r_{o2} \quad (21)$$

ここで、位置失敗点付近での姿勢変化を促すためのコスト  $r_p$  を設定する。コスト  $r_p$  の軸変数の値  $z_{o3}$  を次式のように設定する。

$$z_{o3} = \|\mathbf{p}_p - \mathbf{f}_p\| \quad (22)$$

軸  $z_{o3}$  に対する関数  $r_p$  の設定を行う。軸  $z_{o3}$  については、位置失敗点から離れるほどコストが減少する評価を表現するため、シグモイド関数を利用している。軸  $z_{o3}$  に対する関数  $r_p$  を次式に示す。ただし、 $e$  はゲイン（定数）、 $f$  はパラメータである。

$$r_p(\mathbf{p}_p) = \frac{1}{1 + \exp(-e(z_{o3} - f))} \quad (23)$$

そして、 $r_p$  と  $r_o$  の積をとり、位置失敗点付近での姿勢に関するコスト  $r_{\text{posture}}$  を次式に示す。コスト  $r_{\text{posture}}$  により、位置失敗点付近での姿勢変化を促すことが可能になる。

$$r_{\text{posture}} = r_p r_o \quad (24)$$

最後に、 $r_{\text{position}}$  と  $r_{\text{posture}}$  の値を比較して大きい方を  $r$  とし、以上の方法に基づいて経路の評価に用いる評価関数  $g_1$  を定義する。 $N_F$  回目までの接触情報履歴  $\mathbf{H} = \{C_i | i = 1, \dots, N_F\}$  を用いて、経路上のすべての評価点  $\mathbf{p}$  に関して  $r$  を計算し、各点における  $r$  の最大値の総和を  $g_1$  とする。

$$g_1(\mathbf{X}, \mathbf{C}) = \sum_{k=1}^{N_T} \max_{i=1, \dots, N_F} r(\mathbf{p}_k(\mathbf{X}), C_i) \quad (25)$$



**経路を補正する評価項目** 経路の評価基準 2~4 に基づく、経路の形状を補正するための評価関数  $g_2, g_3, g_4$  について定義する。また、経由点列  $\mathbf{X}$  は  $N$  個の経由点で与えられるとする。

$$\mathbf{X} = \{\mathbf{x}_i | i = 1, \dots, N\} \quad (26)$$

ここで、経由点の姿勢を  $\mathbf{h}$ 、初期経路の経由点の姿勢を  $\mathbf{h}^{\text{path}}$  とする。また、経由点の位置に関する要素を  $\mathbf{x}_p$ 、初期経路の経由点の位置に関する要素を  $\mathbf{x}_p^{\text{path}}$  とする。また、パラメータ  $t$  は位置に関する経路の形状を厳しく評価するための重みである。パラメータ  $t$  を調節することで、位置による修正を大きくするか姿勢による修正を大きくするかを決めることができる。

評価基準 2 に基づいて評価関数  $g_2$  を式 (27) のように定義する。ここで、初期経路の経由点の姿勢  $\mathbf{h}_i^{\text{path}}$  から経由点の姿勢  $\mathbf{h}_i$  への回転を表す四元数を  $\mathbf{q}_{a(i)} = (\mathbf{h}_i^{\text{path}})^{-1} \mathbf{h}_i = (\cos(\phi_{a(i)}/2), \sin(\phi_{a(i)}/2) \mathbf{v}_{a(i)})$  とする。

$$g_2(\mathbf{X}) = \sum_{i=1}^N t \|\mathbf{x}_{p(i)} - \mathbf{x}_{p(i)}^{\text{path}}\| + \sum_{i=1}^N \phi_{a(i)} \quad (27)$$

評価基準 3 に基づいて評価関数  $g_3$  を式 (28) のように定義する。ここで、評価点の姿勢  $\mathbf{q}_{p(k+1)}$  から評価点の姿勢  $\mathbf{q}_{p(k)}$  への回転を表す四元数を  $\mathbf{q}_{b(k)} = (\mathbf{q}_{p(k)})^{-1} \mathbf{q}_{p(k+1)} = (\cos(\phi_{b(k)}/2), \sin(\phi_{b(k)}/2) \mathbf{v}_{b(k)})$  とする。

$$g_3(\mathbf{X}) = \sum_{k=1}^{N_T-1} t \|\mathbf{p}_{p(k+1)}(\mathbf{X}) - \mathbf{p}_{p(k)}(\mathbf{X})\| + \sum_{k=1}^{N_T-1} \phi_{b(k)} \quad (28)$$

評価基準 4 に基づいて評価関数  $g_4$  を式 (29) のように定義する。

$$g_4(\mathbf{X}) = \sum_{i=1}^{N-1} |d_{i(i+1)} - d_{\text{ave}}| + \sum_{i=1}^{N-1} |\phi_{c(i)} - \phi_{\text{ave}}| \quad (29)$$

ただし、以下のとおりとする。

$$d_{ij} = t \|\mathbf{x}_{p(j)} - \mathbf{x}_{p(i)}\| \quad (30)$$

$$d_{\text{ave}} = \frac{1}{N-1} \sum_{j=1}^{N-1} d_{j(j+1)} \quad (31)$$

$$\phi_{\text{ave}} = \frac{1}{N-1} \sum_{j=1}^{N-1} \phi_{c(j)} \quad (32)$$

$$\mathbf{q}_{c(i)} = (\mathbf{h}_{i+1})^{-1} \mathbf{h}_i = (\cos(\phi_{c(i)}/2), \sin(\phi_{c(i)}/2) \mathbf{v}_{c(i)}) \quad (33)$$

## 4 シミュレーション

### 4.1 目的と条件

提案手法のマニピュレータの手先の位置姿勢に関する動作経路生成方法としての利用可能性の検証を目的とし、シミュレーションを MATLAB で行った。本シミュレーションでは、パラメータの調節により 3 次元空間での姿勢変化による障害物の回避を検証した。障害物、マニピュレータの手先の把持した物体の位置や形状に関するデータを事前に用意し、手先位置姿勢に関する動作経路、把持した物体と障害物との位置関係から接触情報（接触点、修正ベクトル）を計算する。また、経路経由点の最適化はネルダーミッド法で行う。経路最適化計算で用いる定数パラメータの値を表 1 に示し、初期動作経路曲線を図 5 に示す。

### 4.2 結果

本シミュレーションで生成された経路を図 6 に示す。本シミュレーションでは、修正回数は 1 回であった。図 5 と図 6 を比較すると、位置に関する経路の変化は小さいが、姿勢による変化が大きく、それにより障害物との接触を避ける方向に修正されていることが確認できる。そのため、提案手法による 3 次元空間での手先の位置姿勢に関する動作経路の生成が可能であることを確認できた。しかしながら、調整する必要があるパラメータが多いことが問題として挙げられる。現時点では、設計者が経験則からパラメータを調節しているため、パラメータの調整に関して教示者の負担になる可能性がある。そのため、可能な限り調整が必要なパラメータを減らすこと、またはパラメータを自動調整することが必要になる。

Table 1 Parameter values

Parameter	Values	Parameter	Values
$\alpha$	(1, 15, 10, 1)	$c$	0.5
$\beta$	0.03	$d$	5
$\gamma$	0.20	$e$	5
$a$	50	$f$	0.01
$b$	10	$t$	150

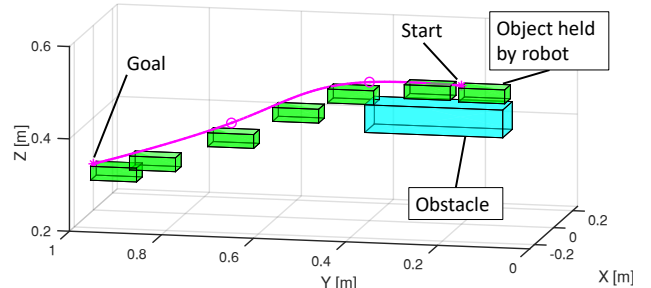


Fig.5 Initial path

## 5 結論

本研究では、生産支援ロボット導入における動作教示者の負担削減を目的とした 3 次元空間での手先姿勢を考慮した動作経路の自動修正手法の提案とシミュレーションを行った。提案手法により、3 次元空間での姿勢変化による障害物との接触が回避できることが確認された。今後の課題としては、提案した経路生成手法に関して調節する必要のあるパラメータを削減させる手法の提案、またはパラメータの自動調整が挙げられる。また、提案手法での実機での検証も行う。

## 参考文献

- [1] 堀口由貴男, 黒野見平, 中西弘明, 榎木哲夫, 長谷達也, 野田哲男, 田中健一, “産業用ロボット教示作業支援のための複合情報 GUI”, 計測自動制御学会論文集, vol.47, no.12, pp.656–665, 2011.
- [2] D. P. Losey, M. K. O’Malley: “Learning the Correct Robot Trajectory in Real- Time from Physical Human Interaction”, ACM Transaction on Human-Robot Interaction, vol.1, no.1, pp.1–8, 2019.
- [3] 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.
- [4] 榎原典佑, 小林祐一, ARJONILLA GARCIA F.J., “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自動調整”, 知能システムシンポジウム講演資料, C3-3, 2022.
- [5] 高見理瑛, 小林祐一, ARJONILLA GARCIA F.J., “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”, 日本機械学会ロボティクス・メカトロニクス講演会講演論文集, 2A1-E14, 2023.

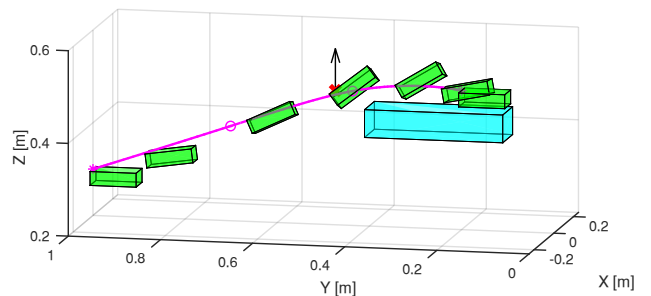


Fig.6 Generated path