

Bézier 曲線を用いたマニピュレータの軌道計画

Trajectory Planning of a Robot Manipulator with Computer Algebra using Bézier Curves for Obstacle Avoidance

筑波大学 数理物質科学研究群 畠山 諒^{*1}

RYO HATAKEYAMA
DEGREE PROGRAMS IN PURE AND APPLIED SCIENCES
UNIVERSITY OF TSUKUBA

筑波大学 数理物質系 照井 章^{*2}

AKIRA TERUI
INSTITUTE OF PURE AND APPLIED SCIENCES
UNIVERSITY OF TSUKUBA

筑波大学 図書館情報メディア系 三河 正彦^{*3}

MASAHIKO MIKAWA
FACULTY OF LIBRARY, INFORMATION AND MEDIA SCIENCE
UNIVERSITY OF TSUKUBA

Abstract

This paper discusses the trajectory planning of a robot manipulator with computer algebra. In the operation of robot manipulators, it is important to make the trajectory of the end effector so that it does not collide with obstacles. We have previously proposed a method to generate curved trajectories for this purpose, but faced the problem that the generated trajectories go out of the manipulator's feasible region. In this paper, we propose a new method of path generation by using the Bézier curve, which allows the manipulator to avoid obstacles without going out of the feasible region.

1 はじめに

本稿では、Bézier 曲線を用いたマニピュレータの軌道計画について論じる。マニピュレータとは、産業用ロボットの一つであり、人間の上腕や前腕にあたる直線的な部分をリンクと呼び、リンク同士の接続点やマニピュレータの両端にはジョイントと呼ばれる関節部分に取り付けられている。マニピュレータの一端は地面に固定され、もう一端にはエンドエフェクタに取り付けられている。ジョイントには回転や伸縮などそ

^{*1} 〒 305-8571 茨城県つくば市天久保 1-1-1 E-mail: s2320143@u.tsukuba.ac.jp

^{*2} 〒 305-8571 茨城県つくば市天久保 1-1-1 E-mail: terui@math.tsukuba.ac.jp

^{*3} 〒 305-8550 茨城県つくば市春日 1-2 E-mail: mikawa@slis.tsukuba.ac.jp

のはたらきによって種類があり、各ジョイントが稼働することでエンドエフェクタが移動する。以下では全てのジョイントが回転ジョイントであるようなマニピュレータに限定して議論を行う。

マニピュレータの逆運動学問題 [21] とは、与えられたある点に、エンドエフェクタが配置可能かどうかを検討し、可能であれば、エンドエフェクタを配置しうる各ジョイントの配置 (角度) を求める問題である。マニピュレータの軌道計画問題とは、逆運動学問題において、入力を点から経路に拡張したものをいう。すなわち、エンドエフェクタをある点 (始点) からある点 (終点) まで移動させたいとき、2 点を結ぶ連続な線を与え、その線上をエンドエフェクタが移動するような一連のジョイントの変位を求める問題である。本稿では、エンドエフェクタが通る線で、始点から終点までで連続に定義されているものをまとめて経路と呼ぶ。

計算機代数を用いたマニピュレータの逆運動学問題の解法についての研究は 30 年ほど前から取り組まれている (例えば [3, 6, 14, 15] を参照)。我々の研究グループによるマニピュレータの軌道計画についての先行研究について述べると、吉澤 [24] および Yoshizawa et al. [17] による包括的 Gröbner 基底系による限量子消去 (CGS-QE) を用いた軌道計画では、エンドエフェクタの始点と終点を結ぶ線分を経路とみなし、線分上の点列の各点に対応するジョイント配置を求めた。一方で、実世界でマニピュレータを作動させる際には、障害物の存在により、単一の線分で構成された経路が遮断され、新たに別の経路を与えなければならない状況が生ずる場合がある。白戸 [21] および Shirato et al. [11] は、エンドエフェクタの経路を曲線に拡張し、スプライン補間を用いて曲線経路を生成する手法を提案した。この手法では、エンドエフェクタが障害物を回避する経路が生成されたが、今度は生成された経路がマニピュレータの実行可能領域からはみ出る場合が生じた。このような背景に鑑み、本稿は、エンドエフェクタの経路で、実行可能領域内で障害物回避を行うようなものを生成する手法の確立および実装を目的とする。

本稿では、経路を実行可能領域に収めるうえで良い性質を持つパラメータ曲線である Bézier 曲線 [1] を用いてマニピュレータの軌道計画を行う。また、簡単のため障害物に外接する球を障害物とみなし、複数の Bézier 曲線を繋げたものをエンドエフェクタの経路とすることで障害物回避を行わせる。本稿ではまず、土橋 [22] を参考に、曲線同士が接点で滑らかに接続されるための条件の定式化を行う。次に、Bézier 曲線の制御点の配置を、実行可能領域の半径およびエンドエフェクタの始点と終点、障害物の情報から一意に定めるためのいくつかの手法を提案し、これにより得られた経路上の点列に対するジョイントの配置を求める。

本稿では、Elephant Robotics 社が開発したマニピュレータ「myCobot 280 m5 [4]」 (図 1 参照、以下「myCobot」と呼ぶ) を対象に軌道計画を行い、エンドエフェクタの大きさは考慮しないものとする。なお、エンドエフェクタの位置のみに着目して軌道計画を行うため、動かすジョイントを 3 個に限定し、3 自由度のマニピュレータとみなす。

本稿の構成は以下の通りである。第 2 章では、マニピュレータの実行可能領域の数学的な再定義など、本稿での前提条件について詳細に議論する。第 3 章では、Bézier 曲線の定義や性質、応用について述べる。第 4 章では、軌道計画における Bézier 曲線を用いた経路の生成に用いる手法を提案する。第 5 章では、軌道計画におけるジョイント配置の推定とその解法について述べる。第 6 章では実験を行い、その結果について述べる。第 7 章ではまとめと今後の研究課題について述べる。

2 マニピュレータの実行可能領域と障害物

エンドエフェクタの届きうる範囲を実行可能領域と呼ぶ。実行可能領域はマニピュレータの仕様により異なるが、本稿では一般のマニピュレータに適用できる、なるべく単純な範囲として新たに $\mathbb{R}^3$ 上の集合 F として定義する。 $\mathbb{R}^3$ において、原点にマニピュレータの基部を置くと、

$$F = \{(x, y, z) \in \mathbb{R}^3 \mid z \geq 0, x^2 + y^2 + z^2 \leq r^2\} \setminus \{(0, 0, 0)\}$$



図 1: myCobot 280 m5

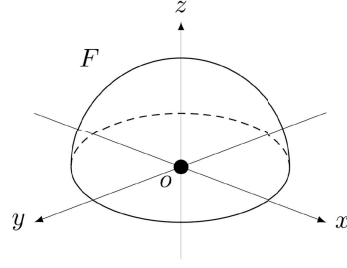


図 2: エンドエフェクタの実行可能領域 F は半球状領域の表面及び内部である

とする (図 2 参照)。 r はマニピュレータのアームの最大長であり、エンドエフェクタは F の任意の点に配置できるものとする。また、各座標における 1 目盛りを 1mm とし、myCobot のアームが最大まで伸び切ったときの基部からエンドエフェクタまでの距離が 280mm であることから、以降は特に断りがない限り $r = 280$ とする。

また、本稿では障害物がエンドエフェクタの始点と終点の間に存在して、直線的な軌道が実現できない場合の軌道を扱う。障害物としては様々な形状が想定されるが、本稿ではそれに外接する球を障害物とみなし、その中心 O_{ob} の座標 $(O_{\text{ob},x}, O_{\text{ob},y}, O_{\text{ob},z})$ および半径 r_{ob} が与えられているものとする。エンドエフェクタの大きさは考慮せず、エンドエフェクタが取り付けられるジョイント部分の点の軌跡を経路とみなす。

3 Bézier 曲線

本章では、Bézier 曲線の定義、本稿の手法において鍵となる性質である「凸包性」および Bézier 曲線の接続について述べる。

3.1 Bézier 曲線とは

Bézier 曲線は t をパラメータとする多項式で定義されるパラメータ曲線で、その形状は制御点と呼ばれる点の位置ベクトルに依存する。また、Bézier 曲線には次数があり、 $n + 1$ 個の制御点で定義される Bézier 曲線を n 次 Bézier 曲線と呼ぶ。

定義 1 (n 次 Bézier 曲線 [9])

与えられた $n + 1$ 個の制御点 $P_0, P_1, \dots, P_n \in \mathbb{R}^3$ に対し、 n 次 Bézier 曲線を

$$P(t) = \sum_{i=0}^n B_{n,i}(t) P_i \quad \left(B_{n,i}(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad 0 \leq t \leq 1 \right) \quad (1)$$

で定める。ここに $B_{n,i}(t)$ を Bernstein 関数と呼ぶ。

n 次 Bézier 曲線は P_0 を始点、 P_n を終点とする。図 3 は 3 次 Bézier 曲線の一例である。曲線は制御点に引き寄せられるような形状をとり、始点と終点以外の制御点を通るとは限らないことが分かる。以下、制御点 P_i ($i = 0, \dots, n$) の成分表示を $(P_{i,x}, P_{i,y}, P_{i,z})$ とする。

エンドエフェクタが移動する経路に Bézier 曲線を用いる利点の 1 つに、Bézier 曲線が持つ凸包性 [9] がある。この性質は本稿課題に対して、生成された軌道が実行可能領域からはみ出ないことを保証するとともに良

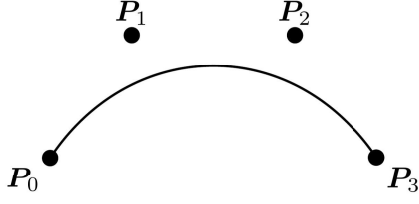


図 3: 3 次 Bézier 曲線の例

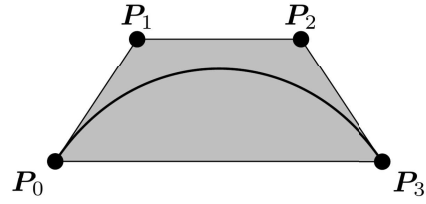


図 4: 制御点の集合 $\{P_0, P_1, P_2, P_3\}$ の凸包 A

い性質である。集合 S に対し、 S の任意の 2 点を結んだ線分上の任意の点を元を持つような集合のうち最小のものを、 S の凸包という。 n 個の制御点 $P_0, P_1, \dots, P_n$ に対して、集合 A を

$$A := \left\{ \sum_{i=0}^n a_i P_i \mid a_i \geq 0, \sum_{i=0}^n a_i = 1 \right\} \quad (2)$$

と定義すると、 A は制御点の集合 $\{P_0, P_1, \dots, P_n\}$ の凸包をなす。図 4 の着色部分は、制御点の集合 $\{P_0, P_1, P_2, P_3\}$ の凸包 A を表す。ここで、式 (1) の $B_{n,i}(t)$ は A の定義式における a_i の要件を満たすので、Bézier 曲線上の任意の点もまた A に含まれることが分かる。また、制御点がすべて実行可能領域に含まれるとき、すなわち各 i に対して $\|P_i\| \leq r$ かつ $P_{i,z} \geq 0$ のとき、

$$\begin{aligned} \left\| \sum_{i=0}^n a_i P_i \right\| &\leq \sum_{i=0}^n \|a_i P_i\| = \sum_{i=0}^n a_i \|P_i\| \leq \sum_{i=0}^n a_i r = r \sum_{i=0}^n a_i = r \cdot 1 = r, \\ (A \text{ の点の } z \text{ 座標}) &= \sum_{i=0}^n a_i P_{i,z} \geq \sum_{i=0}^n a_i \cdot 0 = 0 \end{aligned}$$

より、 A の任意の点が実行可能領域内に存在することが確かめられる。以上より「制御点が実行可能領域内にあるならば、得られる Bézier 曲線上の任意の点も実行可能領域内にある」ことが分かる。

3.2 Bézier 曲線の接続

Bézier 曲線の凸包性は、実行可能領域に収まるような軌道の生成だけでなく、障害物回避にも利用できる。式 (2) の凸包 A と、障害物が $\mathbb{R}^3$ において占める部分との共通部分が空集合ならば、Bézier 曲線は障害物との交点をもたない。一方、本稿における前提のように、エンドエフェクタの始点と終点を結ぶ線分上に障害物がある場合、軌道を 1 本の Bézier 曲線のみで構成すると、図 5 のように A は内部に必ず障害物と交わる点を持つ。

ここで、それぞれの曲線の制御点の集合の凸包が障害物上の点をもたないような複数の短い Bézier 曲線を接続して 1 本の曲線にしたものを軌道とする手法を提案する。図 6 は、3 本の 2 次 Bézier 曲線の制御点で、隣り合う Bézier 曲線の終点と始点が一致し、かつそれぞれの集合の凸包が障害物と交わらないようなものの一例である。 b 本目の Bézier 曲線の制御点を P_j^b ($j = 0, 1, 2$) とし、 $P_2^1 = P_0^2$, $P_2^2 = P_0^3$ とする。このとき、 P_0^1 を始点、 P_2^3 を終点とする軌道が生成される。

以下、2 本の Bézier 曲線が共有している制御点で、一方の Bézier 曲線の終点であり、かつもう一方の Bézier 曲線の始点であるものを接続点と呼ぶ。本稿では土橋 [22] に基づき、接続点においてなめらかに接続される 2 本の Bézier 曲線の制御点のとり方を以下の通り定める。 m, n を 2 以上の自然数とし、 $P_0, \dots, P_m$ を制御点とする m 次 Bézier 曲線 $P(t)$ と、 $Q_0, \dots, Q_n$ を制御点とする n 次 Bézier 曲線 $Q(t)$ がこの順に接続して

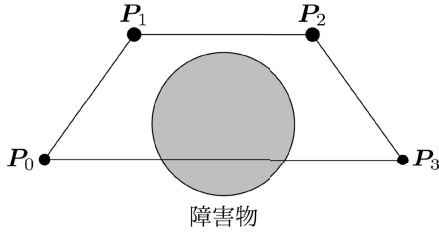


図 5: 1 本の Bézier 曲線を軌道とする場合の集合 A と障害物

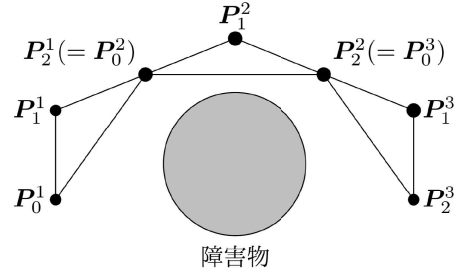


図 6: 複数の Bézier 曲線を軌道とする場合の集合 A と障害物

いるとする。このとき、 $P_m = Q_0$ が接続点となり、接続点における接線方向が 2 曲線で等しい。 $P(t)$ の接続点における一次微分係数は

$$P'(1) = m(P_m - P_{m-1}) = m(Q_0 - P_{m-1}), \quad (3)$$

$Q(t)$ の接続点における一次微分係数は

$$Q'(0) = n(Q_1 - Q_0) \quad (4)$$

であることから、 $P(t)$ の終点における接線方向は P_{m-1} から見た Q_0 の方向を示し、 $Q(t)$ の始点における接線方向は Q_0 から見た Q_1 の方向を示している。よって、2 曲線の接続点における接線方向を等しくするには、 P_{m-1} , Q_0 , Q_1 をこの順に一直線上に配置すればよい。

このとき、ある k ($0 < k < 1$) が存在して、 Q_0 が線分 $P_{m-1}Q_1$ を $k : 1 - k$ に内分するので、

$$Q_0 = (1 - k)P_{m-1} + kQ_1 \quad (5)$$

と表せる。

4 Bézier 曲線を用いた経路計画

本章ではエンドエフェクタの経路の生成方法を提案する。3 本の 2 次 Bézier 曲線 $P(t)$, $Q(t)$, $R(t)$ をこの順に接続して 1 本の経路とする。 $P(t)$, $Q(t)$, $R(t)$ の制御点をそれぞれ P_i, Q_i, R_i ($i = 0, 1, 2$) とすると、この経路は P_0 を始点、 R_2 を終点とする。また、 $P_2 = Q_0$, $Q_2 = R_0$ の 2 点が接続点である。各曲線の式は以下の通り。

$$P(t) = \sum_{i=0}^2 \binom{2}{i} t^i (1-t)^{2-i} P_i, \quad Q(t) = \sum_{i=0}^2 \binom{2}{i} t^i (1-t)^{2-i} Q_i, \quad R(t) = \sum_{i=0}^2 \binom{2}{i} t^i (1-t)^{2-i} R_i$$

本手法では全ての制御点を同一平面上に配置する。これにより、得られる曲線もその平面上に生成される。障害物の回避は 1 つの平面上に与えられた経路で十分である。

4.1 接続点の配置

本節では接続点 Q_0 , R_0 を与える。 P_0, Q_0, R_0, R_2 の 4 点を同一平面上に存在させるための条件に加え、障害物回避で生じる曲線のふくらみを最小限にするためのいくつかの制約の下で配置方法を提案する。

Q_0, R_0 を配置する平面を, 始点 P_0 と終点 R_2 , 障害物球の中心 O_{ob} の 3 点を通る平面とする。まず, 線分 Q_0R_0 が線分 P_0R_2 と平行であると仮定し, P_0, Q_0, R_0, R_2 は同一平面上に存在することを保証する。また, 線分 Q_0R_0 の長さは障害物球の直径と等しく $2r_{ob}$ とする。次に, O_{ob} から線分 P_0R_2 に下ろした垂線の足を K , 垂線の延長と障害物球の球面との交点を L とする (図 7 参照)。 L は線分 Q_0R_0 の中点と一致すると仮定する。また, K について,

$$(K - P_0) \cdot (K - O_{ob}) = 0 \quad (6)$$

が成り立つ。また, ある c ($0 < c < 1$) が存在して, $K = (1 - c)P_0 + cR_2 = P_0 + c(R_2 - P_0)$ と表せる。これを式 (6) に代入し, c について解くと,

$$c = \frac{(R_2 - P_0) \cdot (O_{ob} - P_0)}{\|R_2 - P_0\|^2} \quad (7)$$

より, K が求まる。また, L の定義から,

$$L = \frac{r_{ob}(K - O_{ob})}{\|K - O_{ob}\|} + O_{ob} \quad (8)$$

が得られる。 L は線分 Q_0R_0 の中点にあたるので, $L = (Q_0 + R_0)/2$ 。線分 Q_0R_0 は線分 P_0R_2 に平行かつ長さが障害物球の直径に等しいので, $R_0 - Q_0 = 2r_{ob}(R_2 - P_0)/\|R_2 - P_0\|$ と表せる。以上により, 連立方程式

$$\begin{cases} R_0 + Q_0 = 2L \\ R_0 - Q_0 = \frac{2r_{ob}(R_2 - P_0)}{\|R_2 - P_0\|} \end{cases} \quad (9)$$

を得る。2 式とも右辺が既知の値および位置ベクトルのみから成るので, これを解くことにより Q_0, R_0 が定まる。実際, 式 (9) を Q_0, R_0 について解くと, 式 (8) より,

$$\begin{aligned} Q_0 &= \frac{r_{ob}(K - O_{ob})}{\|K - O_{ob}\|} + O_{ob} - \frac{r_{ob}(R_2 - P_0)}{\|R_2 - P_0\|}, \\ R_0 &= \frac{r_{ob}(K - O_{ob})}{\|K - O_{ob}\|} + O_{ob} + \frac{r_{ob}(R_2 - P_0)}{\|R_2 - P_0\|} \end{aligned}$$

を得る。ここで, 式 (7) より,

$$K = P_0 + \frac{\{(R_2 - P_0) \cdot (O_{ob} - P_0)\}(R_2 - P_0)}{\|R_2 - P_0\|^2}$$

であることから, Q_0, R_0 の位置を P_0, R_0, O_{ob}, r_{ob} のみを用いて表せた。

この配置方法により, 線分 Q_0R_0 が線分 P_0R_2 に平行で長さが $2r_{ob}$, かつ中点で障害物球に接するような Q_0, R_0 の中で, $\|R_2 - R_0\| + \|Q_0 - P_0\|$ が最小になるものを選ぶことができる。

4.2 その他の制御点の配置

前節に述べた手法で, 接続点の位置が定まったら, 残りの制御点の位置も決定する。本節では第 3.2 節の議論をもとに, 残りの制御点を決めるための手法を提案する。

接続点における接線方向が隣接する Bézier 曲線同士で等しいとき, ある k, l ($0 < k, l < 1$) が存在して,

$$Q_0 = (1 - k)P_1 + kQ_1, \quad R_0 = (1 - l)R_1 + lQ_1$$

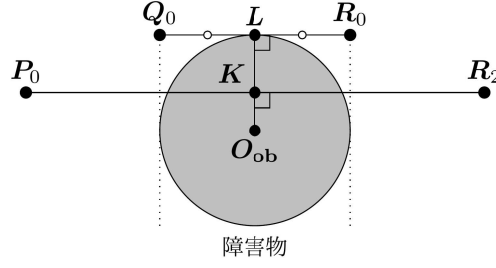


図 7: 手法 2 における K, L (線分 P_0R_2 と線分 Q_0R_0 は平行)

が成り立つ。これらを変形すると,

$$P_1 = \frac{Q_0 - kQ_1}{1 - k}, \quad R_1 = \frac{R_0 - lQ_1}{1 - l} \quad (10)$$

となる。 $P_1, Q_1, R_1 \in F$ とするため, 以下の不等式が成り立つ。

$$\|P_1\|^2, \|Q_1\|^2, \|R_1\|^2 \leq r^2, \quad P_{1,z}, Q_{1,z}, R_{1,z} \geq 0 \quad (11)$$

式 (11) を満たす Q_1 の各座標と k, l の組のうち $\|R_2 - R_1\|^2 + \|R_1 - R_0\|^2 + \|R_0 - Q_1\|^2 + \|Q_1 - Q_0\|^2 + \|Q_0 - P_1\|^2 + \|P_1 - P_0\|^2$ が最小になるものを選ぶことにすると, Q_1, k, l は次の制約付き最小化問題の解として得られ, P_1, R_1 はこの Q_1, k, l を式 (10) に代入することで得られる。

$$\begin{aligned} \text{minimize} \quad & \left\| \frac{l(Q_1 - R_2) - R_0 + R_2}{1 - l} \right\|^2 + \left\| \frac{-l(Q_1 - R_0)}{1 - l} \right\|^2 + \| -Q_1 + R_0 \|^2 + \| -Q_0 + Q_1 \|^2 \\ & + \left\| \frac{-k(Q_0 - Q_1)}{1 - k} \right\|^2 + \left\| \frac{k(P_0 - Q_1) - P_0 + Q_0}{1 - k} \right\|^2 \\ \text{subject to} \quad & 0 < k < 1 \wedge 0 < l < 1 \wedge \|Q_0 - kQ_1\|^2 \leq r^2 \wedge \|Q_1\|^2 \leq r^2 \wedge \|R_0 - lQ_1\|^2 \leq r^2 \\ & \wedge Q_{0,z} - kQ_{1,z} \geq 0 \wedge Q_{1,z} \geq 0 \wedge R_{0,z} - lQ_{1,z} \geq 0 \end{aligned}$$

4.3 曲線上の点列の配置

第 4.1 節および第 4.2 節で提案した手法ですべての制御点が求まり, エンドエフェクタが移動する経路が定まったら, 経路上にエンドエフェクタが通過する点列を配置する。本稿では, この点列における各点に対して逆運動学問題を解き, マニピュレータのジョイント配置を求める。以下, 軌道上の点列の配置方法を述べる。 $N \in \mathbb{N}$ に対し, t_i ($i = 0, 1, \dots, N$) を

$$t_i := \frac{i}{N}$$

で定める。Bézier 曲線 $P(t), Q(t), R(t)$ に $t = t_i$ を代入すると, 以下の $3(N + 1)$ 個の点を得られる。

$$P(t_0), \dots, P(t_N), Q(t_0), \dots, Q(t_N), R(t_0), \dots, R(t_N)$$

ここで, $P(t_N) = P(N/N) = P(1) = Q(0) = Q(0/N) = Q(t_0)$, $Q(t_N) = Q(N/N) = Q(1) = R(0) = R(0/N) = R(t_0)$ となり重複する 2 点が存在するため, これらを取り除いた $3(N + 1) - 2 (= 3N + 1)$ 個の点を以下のように並べる。

$$(P(t_0), \dots, P(t_{N-1}), Q(t_0), \dots, Q(t_{N-1}), R(t_0), \dots, R(t_N))$$

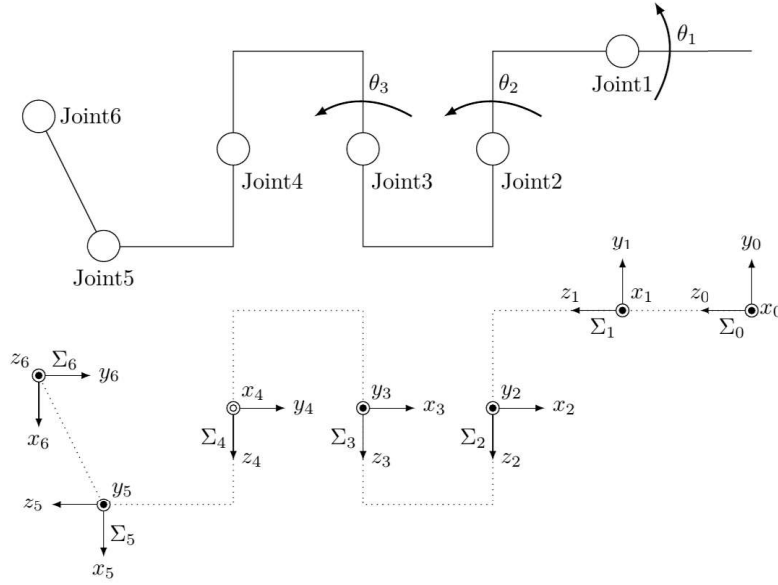


図 8: myCobot のモデル図と各ジョイントで定義される座標系

この点列のとり方は単純である一方で, 用いる Bézier 曲線の長さに応じて隣接する点同士の間隔が異なるため, 生成された 3 本の Bézier 曲線の長さが大きく異なる場合には軌道上の点に疎密が生じる。このような場合, Bézier 曲線のパラメータ t を時刻とみなしたとき, 区間によってエンドエフェクタの移動速度が極端に異なる現象が生ずる恐れがある。これを改善する点列の与え方として, 弧長パラメータを用いるもの, 曲線の長さの比を用いるものなどが挙げられるが, 具体的な点列の与え方の手法の確立は今後の課題である。

5 ジョイント配置の推定

経路上の点列が求まったら, 各点にエンドエフェクタを配置するためのジョイント配置を求める。本稿では堀込 [23] および Horigome et al. [5], 大瀧 [18] および Otaki et al. [8], 吉澤 [24] および Yoshizawa et al. [17], 白戸 [21] および Shirato et al. [11] と同様, 「Denavit-Hartenberg の表記法」[12] に基づいて myCobot を構成する各ジョイントにおける 3 次元の座標系を定める。各ジョイントの名称および変数の定義については白戸 [21] および Shirato et al. [11] のものを用いる。myCobot のモデルと, 各 $i = 1, \dots, 6$ に対して基部から i 番目のジョイント (Joint i と表す。) で定義される座標系 Σ_i (基部で定義される座標系を Σ_0 とする。) は図 8 に表される。各座標系の軸の方向は岡崎 [20] を参考に設定した。

myCobot の座標系どうしの関係を示すため, 隣接する座標系間の変換行列を与える。まず, 各 i に対して, DH パラメータと呼ばれる 4 つのパラメータ $a_i, \alpha_i, d_i, \theta_i$ と, 岡崎 [20] によるパラメータ δ_i を以下のように定義する。

- a_i : z_{i-1} 軸と z_i 軸の距離
- α_i : x_{i-1} 軸を中心とした z_{i-1} 軸から z_i 軸への角度 (rad)
- d_i : x_{i-1} 軸と x_i 軸の距離
- θ_i : z_i 軸を中心とした x_{i-1} 軸から x_i 軸への角度 (rad)

表 1: myCobot の DH パラメータ [20]

Joint	a_i	α_i	d_i	θ_i	δ_i
1	0	$\pi/2$	131.56	θ_1	$-\pi/2$
2	-110.4	0	0	θ_2	0
3	-96	0	0	θ_3	$-\pi/2$
4	0	$\pi/2$	64.62	θ_4	$\pi/2$
5	0	$-\pi/2$	73.18	θ_5	0
6	0	0	48.6	θ_6	0

- δ_i : z_{i-1} 軸を中心とした x_{i-1} 軸から x_i 軸への角度 (rad)

myCobot における各パラメータの値を表 1 に示す。 θ_i は Joint i の回転角度にあたる。 Σ_i から Σ_{i-1} への変換行列 ${}^{i-1}T_i$ は以下のように表される。

$${}^{i-1}T_i = \begin{pmatrix} \cos \delta_i \cos \theta_i - \cos \alpha_i \sin \delta_i \sin \theta_i & -\cos \alpha_i \cos \delta_i \sin \theta_i - \sin \delta_i \cos \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \cos \alpha_i \sin \delta_i \cos \theta_i + \cos \delta_i \sin \theta_i & \cos \alpha_i \cos \delta_i \cos \theta_i - \sin \delta_i \sin \theta_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ \sin \alpha_i \sin \delta_i & \sin \alpha_i \cos \delta_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Σ_0 座標系におけるエンドエフェクタの座標を (x, y, z) , Σ_6 座標系から Σ_0 座標系への変換行列 T を

$$T := {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6$$

と定義すると、エンドエフェクタは Σ_6 座標系の原点に存在することから以下の等式が成り立つ。

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = T \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (12)$$

本稿では動かすジョイントを Joint 1, Joint 2, Joint 3 に限定するため、それ以外のジョイントの角度については $\theta_4 = \theta_5 = \theta_6 = 0$ と固定する。 $i = 1, 2, 3$ に対して、 $c_i = \cos \theta_i$, $s_i = \sin \theta_i$ とおくと、式 (12) の変形および三角関数の性質から以下の連立代数方程式を得る。(次式の第 1, 2, 3 式は係数を定数倍することにより整数化している。)

$$\begin{cases} 6462s_1 + c_1\{c_2(4860c_3 - 16918s_3) + s_2(-11040 - 16918c_3 - 4860s_3)\} - 100x = 0 \\ -6462c_1 + s_1\{c_2(4860c_3 - 16918s_3) + s_2(-11040 - 16918c_3 - 4860s_3)\} - 100y = 0 \\ 13156 + 4860c_3s_2 - 16918s_2s_3 + c_2(11040 + 16918c_3 + 4860s_3) - 100z = 0 \\ c_1^2 + s_1^2 - 1 = 0 \\ c_2^2 + s_2^2 - 1 = 0 \\ c_3^2 + s_3^2 - 1 = 0 \end{cases} \quad (13)$$

式 (13) の (x, y, z) に点列上の各点の座標を代入してこれを解くことで、対応するジョイント配置を得る。本稿では、吉澤 [24] および Yoshizawa et al. [17] の逆運動学問題の解法アルゴリズム (アルゴリズム 1) に式 (13) を適用してジョイント配置を求める。アルゴリズム 1 における入力である多項式集合 U とは、式 (13)

アルゴリズム 1 マニピュレータの逆運動学問題の解法 [17]

入力: 式 (13) の多項式集合 U , エンドエフェクタの座標 (x, y, z) の値

出力: 式 (13) を満たす変数 $\theta_1, \theta_2, \theta_3$ の値

```
1:  $U$  の包括的グレブナー基底系  $\mathcal{G} = \{(S_1, T_1, G_1), \dots, (S_k, T_k, G_k)\}$  を求める;  
2: for  $i \in [1, \dots, k]$  do  
3:   if (分割部  $S_i$  が実数体上に存在しない) then  
4:      $\mathcal{G} := \mathcal{G} \setminus \{(S_i, T_i, G_i)\}$ ;  
5:   end if  
6: end for  
7:  $\mathcal{G}$  において,  $(x, y, z) \in S_i$  を満たす  $i$  を決定する;  
8: グレブナー基底  $G_i$  に  $(x, y, z)$  の値を代入する;  
9: if (式 (13) を満たす実根が存在する) then  
10:  return (式 (13) を満たす変数  $\theta_1, \theta_2, \theta_3$  の値);  
11: else  
12:  return  $\emptyset$ ;  
13: end if
```

の各方程式の左辺の集合である。アルゴリズム 1 では, まず U の包括的グレブナー基底系 [13] を求める。次に経路上の点列の各点の座標を, その点を元を持つ分割部における U のグレブナー基底に代入し, それが実根を持つならば, その実根から定まる $\theta_1, \theta_2, \theta_3$ を返す。アルゴリズム 1 における包括的グレブナー基底系に関する記法および記号の定義は白戸 [21] のものを用いた。

6 実験結果

本章では, 第 4 章, 第 5 章で述べた経路生成の手法の実験結果について述べる。

6.1 Bézier 曲線を用いた経路計画の実験結果

まず, 第 4 章の議論をもとに Bézier 曲線の制御点および経路上の点列を求めた結果について述べる。本稿では, 以下の Case 1, 2, 3, 4 の, エンドエフェクタの始点 $\mathbf{S}$, 終点 $\mathbf{G}$, $\mathbf{O}_{\text{ob}}$ の座標および r_{ob} が異なる 4 つの入力に対して, Bézier 曲線の制御点を配置し, 生成された経路上の点列を配置するまでの計算時間を測定した。経路上の点列の配置については, 第 4.3 節において $N = 10$ とし, $\mathbf{S}$ と $\mathbf{G}$ を含む 31 個の点を算出した。

Case 1 $\mathbf{S} = (-200, 50, 50)$, $\mathbf{G} = (200, 50, 50)$, $\mathbf{O}_{\text{ob}} = (0, 50, 40)$, $r_{\text{ob}} = 40$

Case 2 $\mathbf{S} = (-200, 50, 50)$, $\mathbf{G} = (200, 50, 50)$, $\mathbf{O}_{\text{ob}} = (0, 70, 40)$, $r_{\text{ob}} = 40$

Case 3 $\mathbf{S} = (-200, 0, 50)$, $\mathbf{G} = (200, 100, 50)$, $\mathbf{O}_{\text{ob}} = (0, 50, 40)$, $r_{\text{ob}} = 40$

Case 4 $\mathbf{S} = (-200, 0, 50)$, $\mathbf{G} = (200, 100, 50)$, $\mathbf{O}_{\text{ob}} = (0, 70, 40)$, $r_{\text{ob}} = 40$

実験結果を表 2 に示す。ここで記されている計算時間は, それぞれの Case において同様の計算を 10 回繰り返した合計時間を 10 で割った平均である。計算時間を測定した実行環境には Python 3.12.8 を用いており, 各手法における最適化問題の求解には Python の SciPy [16] ライブラリに含まれるモジュール `scipy.optimize` の関数 `minimize` を用いた。なお, 数理最適化の手法には SLSQP (逐次 2 次計画法) [10] が使われている。

表 2: 点列配置にかかる計算時間

Case	1	2	3	4
計算時間 (秒)	0.022	0.029	0.027	0.032

6.2 ジョイント配置の推定の実験結果

次に、第 5 章の議論をもとにジョイント配置を求めた結果について述べる。本稿では、Case 1 において得られた 31 個の点列の各点にエンドエフェクタを配置できるジョイント配置を求めた。アルゴリズム 1 の実行には、数式処理システム Risa/Asir [7] を用いた。

実験結果について述べる。得られた点列上の点のうち 20 個の点に対しては、エンドエフェクタを配置できるジョイント配置が得られたが、残りの 11 個の点に対しては対応するジョイント配置が得られなかった。この現象の原因の 1 つとして、myCobot の動かすジョイントを 3 つに限定し、それ以外のジョイントの角度を $\theta_4 = \theta_5 = \theta_6 = 0$ と固定したことが影響していると考えられる。

7 まとめと今後の課題

本稿では、マニピュレータの軌道計画において、エンドエフェクタが実行可能領域からはみ出さずに障害物回避を行う経路の生成を行った。Bézier 曲線の凸包性に着目し、経路には接続された複数個の Bézier 曲線を用いた。Bézier 曲線が滑らかに接続されるための条件の定式化を行い、始点と終点、障害物の情報が与えられているときの制御点の配置の方法を、接続点とその他の制御点に分けてそれぞれ提案した。その他の制御点については、先述の曲線の滑らかな接続の条件を用いた。また、制御点の配置の過程には最適化問題の求解も含む。求まった制御点に対して、Bézier 曲線で構成された経路および経路上の点列を与え、逆運動学問題アルゴリズムを用いてエンドエフェクタの座標に対応するジョイント配置の列を求めた。今後の課題としては以下が挙げられる。

1. エンドエフェクタの大きさを考慮した経路の生成

本稿ではエンドエフェクタの大きさを考慮せず、エンドエフェクタが取り付けられるジョイント部分の点の軌跡を経路として扱ったが、本稿で提案した手法ではエンドエフェクタが障害物に衝突する可能性もあり得る。実際にはエンドエフェクタは幅や高さ、厚みを持つため、実際のエンドエフェクタが障害物と接触しないような制御点を与える必要がある。このための手法の一つとして、エンドエフェクタを球とみなし、球が障害物に触れないための制約条件を与えた 1 階述語論理式に対して限量子消去を行い、制御点配置を得ることが考えられる。

2. エンドエフェクタの速度・加速度を考慮した点列の与え方の確立

本稿で提案したエンドエフェクタの経路上の点列のとり方では、隣接する点同士の間隔に疎密が見られる場合がある。このような場合、パラメータを時刻とみなした際のエンドエフェクタの移動速度に極端な変化が起こりうる。この課題の解決には、経路上のある点に対して、エンドエフェクタの始点からある点までの経路に沿った長さから定まるパラメータを与えることが考えられる。しかし、パラメータの区間により定義式が異なる曲線では弧長パラメータ表示が与えられないため、別の定式化によるパラメータを与える必要がある。

3. ジョイント配置の推定に関する諸問題の解決

白戸 [21] および Shirato et al. [11] によれば, ジョイント配置を求める過程において, 逆運動学問題における最適解の選択や求解アルゴリズムの効率化, マニピュレータの自由度の拡張などの重要な課題が多く残されている。本稿ではこれらの課題は深くは取り扱わなかったが, これらを解決してより効率のよい軌道計画を行うことが本稿の今後の課題である。これらの解決には岡 [19] および Oka et al. [11], 岡崎 [20] による取り組みがある。また, 本稿では, エンドエフェクタの経路上の点列において, ジョイント配置が得られない点が存在することが確認された。動かすジョイントを限定する場合には, F 上であってもエンドエフェクタの配置が不可能な点が存在するため, エンドエフェクタの位置のみに着目して軌道計画を行うときのマニピュレータの実行可能領域を調べ, 再定義することも課題として挙げられる。

謝 辞

本稿は、京都大学の国際共同利用・共同研究拠点である数理科学研究所の支援を受けています。
また、本稿は競輪の補助を受けています。

参 考 文 献

- [1] P. Bézier, The Mathematical Basis of the UNIURF CAD System, Butterworth-Heinemann, 72p, 1986. <https://doi.org/10.1016/C2013-0-01005-5>
- [2] CHUWI INNOVATION LIMITED, LarkBox. <https://www.chuwi.com/jp/product/items/Chuwi-LarkBox.html>
- [3] R. X. da Silva, L. Schnitman, and V. Cesca Filho, A Solution of the Inverse Kinematics Problem for a 7-Degrees-of-Freedom Serial Redundant Manipulator Using Grbner Bases Theory, Mathematical Problems in Engineering, 14p, 2021. <https://doi.org/10.1155/2021/6680687>
- [4] Elephant Robotics, myCobot 280. <https://www.elephantrobotics.com/en/mycobot-en>
- [5] N. Horigome, A. Terui, and M. Mikawa, A Design and an Implementation of an Inverse Kinematics Computation in Robotics Using Gröbner Bases, Mathematical Software – ICMS 2020, pp. 3–13, 2020. https://doi.org/10.1007/978-3-030-52200-1_1
- [6] C. M. Kalker-Kalkman, An implementation of Buchbergers’ algorithm with applications to robotics, Mechanism and Machine Theory, vol. 28, no. 4, pp. 523–537, 1993. [https://doi.org/10.1016/0094-114X\(93\)90033-R](https://doi.org/10.1016/0094-114X(93)90033-R)
- [7] M. Noro and T. Takeshima, Risa/Asir—a computer algebra system, Papers from the International Symposium on Symbolic and Algebraic Computation, pp. 387–396, 1992. <https://doi.org/10.1145/143242.143362>
- [8] S. Otaki and A. Terui and M. Mikawa, A Design and an Implementation of an Inverse Kinematics Computation in Robotics Using Real Quantifier Elimination based on Comprehensive Gröbner Systems, Preprint, 26p, 2021. <https://doi.org/10.48550/arXiv.2111.00384>
- [9] D. Salomon, Curves and Surfaces for Computer Graphics, Springer, 460p, 2005. <https://doi.org/10.1007/0-387-28452-4>

- [10] K. Schittkowski, The nonlinear programming method of Wilson, Han, and Powell with an augmented Lagrangian type line search function, *Numerische Mathematik*, vol. 38, pp.83–114, 1982. <https://doi.org/10.1007/BF01395810>
- [11] Y. Shirato and N. Oka and A. Terui and M. Mikawa, An Optimized Path Planning of Manipulator with Spline Curves Using Real Quantifier Elimination Based on Comprehensive Gröbner Systems, *SCSS 2024 Work in Progress Proceedings*, pp. 105–112, 2024.
- [12] B. Siciliano, et al. *Robotics: Modelling, Planning and Control*, Springer, 632p, 2008. <https://doi.org/10.1007/978-1-84628-642-1>
- [13] A. Suzuki and Y. Sato, A simple algorithm to compute comprehensive Gröbner bases using Gröbner bases, *Proc. ISSAC 2006*, pp. 326–331, 2006. <https://doi.org/10.1145/1145768.1145821>
- [14] T. Uchida and J. McPhee, Triangularizing kinematic constraint equations using Grbner bases for real-time dynamic simulation, *Multibody System Dynamics*, vol. 25, pp. 335–356, 2011. <https://doi.org/10.1007/s11044-010-9241-8>
- [15] T. Uchida and J. McPhee, Using Grbner bases to generate efficient kinematic solutions for the dynamic simulation of multi-loop mechanisms, *Mechanism and Machine Theory*, vol. 52, pp. 144–157, 2012. <https://doi.org/10.1016/j.mechmachtheory.2012.01.015>
- [16] P. Virtanen, R. Gommers, Oliphant, T.E. et al, SciPy 1.0: fundamental algorithms for scientific computing in Python, *Nature Methods*, vol. 17, pp. 261–272, 2020. <https://doi.org/10.1038/s41592-019-0686-2>
- [17] M. Yoshizawa and A. Terui and M. Mikawa, Inverse kinematics and path planning of manipulator using real quantifier elimination based on Comprehensive Gröbner Systems, *Computer Algebra in Scientific Computing (CASC 2023)*, Springer, 2023, pp. 393–419. https://doi.org/10.1007/978-3-031-41724-5_21
- [18] 大瀧脩人, 包括的グレブナー基底系に基づいた限量子消去のアルゴリズムとその応用, 筑波大学数理物質科学研究科修士論文, 31p, 2021.
- [19] 岡夏未, 数式処理によるマニピュレータの制御と最適経路計画, 筑波大学理工情報生命学術院数理物質科学研究群修士論文, 18p, 2024.
- [20] 岡崎拓夢, 数式処理による 6 自由度ロボットマニピュレータの逆運動学問題の解法とその応用, 筑波大学理工情報生命学術院数理物質科学研究群修士論文, 69p, 2025.
- [21] 白戸祐輔, 包括的グレブナー基底系とスプライン補間を用いたロボットアームの軌道計画, 筑波大学理工情報生命学術院数理物質科学研究群修士論文, 16p, 2024.
- [22] 土橋敏生, Bezier 曲線を用いたマニプレータの軌道生成とその曲線制御, *日本機械学会論文集 C 編*, vol. 55, no. 509, pp. 124–128, 1989. <https://doi.org/10.1299/kikaic.55.124>
- [23] 堀込紀行, 数式処理によるロボットの逆運動学問題の解法と実装, 19p, 筑波大学数理物質科学研究科修士論文, 19p, 2020.
- [24] 吉澤瑞輝, 包括的グレブナー基底系を用いたロボットアームの逆運動学問題の解法とその応用, 筑波大学理工情報生命学術院数理物質科学研究群修士論文, 34p, 2023.