

ロボットの直線移動のための軌道生成に関する基礎研究

日 高 良 和*

Study on Trajectory Generation for Linear Motion of Robot

Yoshikazu HITAKA

Abstract

A number of trajectory generation algorithms exist for calculating the joint positions which will drive a robotic manipulator along a given geometric path. This paper presents a rectilinear interpolation method with skip region of interpolation points for trajectory generation of robotic manipulator points of which all joints are rotational. The interpolation points are given by a segment which between start and end points of extreme robot arm, is dividing equally. A skip region make round the interpolation point. If extreme robot arm reaches in skip region, the robot arm move to the following target point. Then the speed-up of the trajectory generation can be expected. As numerical examples, the trajectories of two-degree of freedom articulated robot are simulated and the skip region effect is confirmed.

1. はじめに

産業用ロボットの制御に必要なことは、所望の運動形態と精度のもとで、作業対象物を移動・位置決めする事である。ロボットアームの手先をある点からある点まで移動させる場合の経路制御方式として、PTP(Point to Point)制御とCP(Continuous Path)制御の2つがある。

PTP 制御は、ロボットアーム移動において、目標とする位置だけに着目し、途中の動作経路を問わない制御方式である。この制御方式は、搬送、ローディング、アンローディング、スポット溶接などのように作業の重点がロボットアーム移動の途中よりも位置決めされる場所にあるような作業に用いられる。

いっぽう、CP 制御は、ロボットアーム移動に際して、

その動作経路を実現させる制御方式であり、塗装、切断シーリング、連続溶接などのように動作経路が重要となる作業の場合に必要なとなる。

ロボットが作業を行うためには、ロボットに作業の順序や作業する位置などの作業に必要なデータを記憶させなければならない。このことを教示（ティーチング）と呼ぶ。PTP 制御の場合、作業位置のティーチングは位置決めしたいポイントの座標を入力するだけでよいが、実際にロボットアームを動作させたとき、どのような軌跡を描くかがわからないため、ロボットアームがロボットの周辺に配置された装置や治具に衝突するという危険性がある。また、CP 制御の場合は、ロボットアームの動く全ての動作経路を指定するためティーチングに多くの時間とティーチング作業者の熟練が必要になる。

そこで、作業位置のティーチング回数が少なく、しかもロボットアームの経路が想定しやすい経路制御が必要となる。この経路制御方式として、本研究では、ロボッ

*宇部工業高等専門学校電気工学科

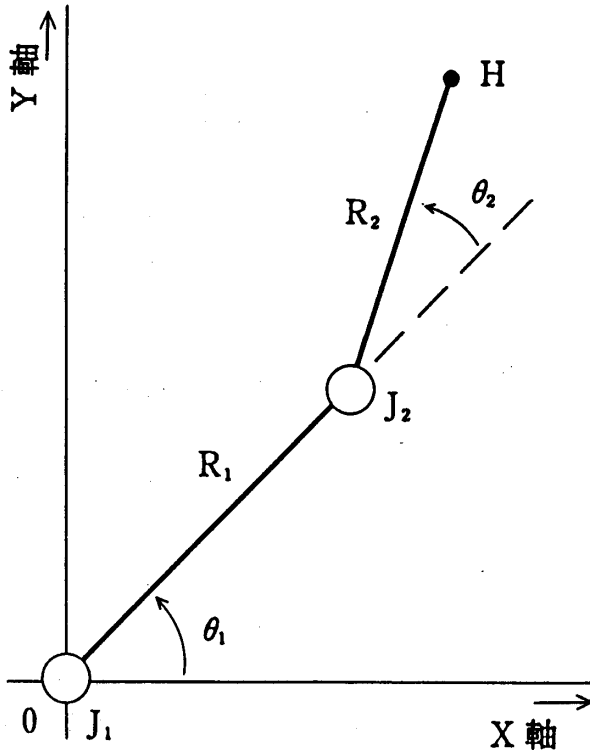


図1 2自由度水平多関節ロボットモデル

トアーム移動の始点と終点との間に仮の目標点を用意して、ロボットアームが順番に、この目標点へ向かうようにPTP制御を行う経路制御を考えた。[2] また、動作経路として種々の曲線が考えられるが、ここでは、直線の動作経路を考える。それは、経路の想定が容易であり、始点と終点の取り方によっては曲線に近い経路も実現可能であるからである。本論文では、2自由度を持つ水平多関節ロボットにこの経路制御方式を用いて、ロボットアームの直線移動のシュミレーションを行い、目標点設定の条件について検討を行ったので報告する。

2. ロボットモデル

2.1 ロボットモデルの仮定

ロボットのモデルは、ロボットアームの移動軌跡と目標経路との確認が容易であり、かつ簡単な幾何学でロボットアームの運動が表記できるように、図1に示すような2自由度を持つ水平多関節ロボットを対象とする。

ロボットアームの運動を解析する場合には、各アーム軸まわりのモーメント計算や速度、加速度の関係を考慮

した運動方程式を求める必要がある。しかし、本論文では経路生成の方法に主眼をおいているので、ロボットのダイナミクスはとりあえず無視することにする。また、ロボットの各関節を駆動するアクチュエータは、次のような理想的な制御を施されているものとする。すなわち、各アクチュエータはロボットアームの運動に必要なトルク、速度、及び加速度を瞬時に発生できるよう制御されており、その応答性もアームの運動に支障がないものである。また、アクチュエータの慣性モーメントや摩擦力などは無視できるとする。

2.2 関節変数と手先位置

図1に示すXY平面内を動く2自由度のロボットアームの各関節の変位（この場合、回転関節であるから回転変位）とアーム先端位置との関係は、関節変数ベクトルが、 $q = [\theta_1, \theta_2]^T$ 、手先位置ベクトルが、 $r = [x, y]^T$ となるから次式のようにになる。

$$\left. \begin{aligned} x &= R_1 \cos \theta_1 + R_2 \cos(\theta_1 + \theta_2) \\ y &= R_1 \sin \theta_1 + R_2 \sin(\theta_1 + \theta_2) \end{aligned} \right\} \quad (1)$$

ここで、 R_1, R_2 は第1、第2アームの長さ、 θ_1, θ_2 は関節 J_1, J_2 の回転角である。

関節変数ベクトルが与えられたときには、それに対応する手先位置ベクトルは(1)式により一意に定まる。しかし、ロボットで作業をするためにティーチングを行うとき、その入力データは関節変数やアーム長ではなく、手先位置や姿勢であることが一般的である。したがって、与えられた手先位置ベクトルに対して、(1)式を満足する関節変数ベクトルを求めなければならない。

(1)式から関節変数を求める式を導出する。このとき、手先位置が

$$\left. \begin{aligned} (R_1 - R_2)^2 \leq x^2 + y^2 \leq (R_1 + R_2)^2 \\ x^2 + y^2 \neq 0 \end{aligned} \right\} \quad (2)$$

を満足するならば関節変数が存在する。しかしながら、この条件だけではアームの姿勢（手先の方向）を指定できないので、各関節の回転角度が2組得られる。そこで、回転角度が1組だけ決定できるように、次のような動作条件を付ける。

- I) 座標系は右手系座標とする。
- II) θ_1 のとれる範囲を $-\pi \leq \theta_1 \leq \pi$ とする。
- III) θ_2 のとれる範囲を $0 \leq \theta_2 \leq \pi$ とする。

条件II)、III)は、ロボットの構造で決まる条件であり、

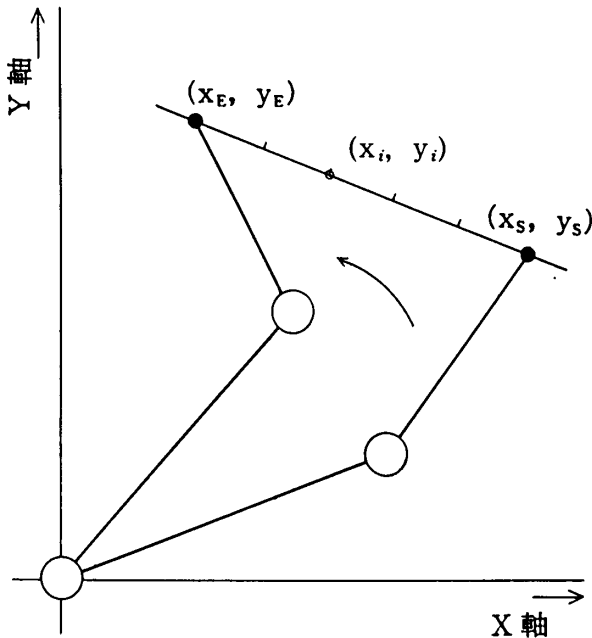


図2 ロボットの直線動作

実際のロボットではアームやアクチュエータの干渉、配線、配管の都合から上記の条件を満たすことは難しいが、ここではロボットアームの運動のみに注目するとして、この条件を用いる。

これらの条件を考慮して(1)式から回転角 θ_1, θ_2 を求めると次式ようになる。

$$\theta_1 = \tan^{-1} \frac{y \cdot p - x \cdot q}{p \cdot (x \cdot p + y \cdot q)} \quad (4)$$

$$\theta_2 = \cos^{-1} \frac{x^2 + y^2 - R_1^2 + R_2^2}{2 \cdot R_1 \cdot R_2} \quad (5)$$

ただし、 $p = R_1 + R_2 \cdot \cos \theta_2$, $q = R_2 \cdot \sin \theta_2$

3. 経路生成

3.1 中間目標点の決定

現在位置 P_s から最終目標位置 P_e まで直線移動するために、2点間を直線で結び、この区間を分割して中間目標点として新しい座標を生成する。ロボットアームは現在位置から順次つぎの中間目標点をめざして移動すれば最終目標位置 P_e まで直線移動ができる。その様子を図2に示す。

ロボットアームが現在位置 $P_s (x_s, y_s)$ から最終目標

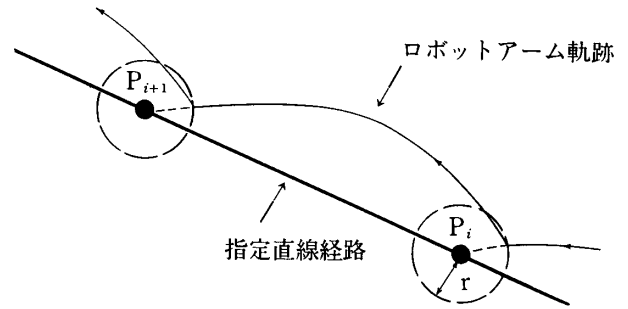


図3 パス領域

位置 $P_e (x_e, y_e)$ へ移動する場合、この区間を N 等分したときの中間目標点 $P_i (x_i, y_i)$ は、次のようになる。

$$\left. \begin{aligned} x_i &= x_s + \frac{x_e - x_s}{N} \cdot i \\ y_i &= y_s + \frac{y_e - y_s}{N} \cdot i \end{aligned} \right\} \quad (6)$$

$(i = 0, 1, 2, \dots, N)$

さらに、中間目標点間の移動に必要な各関節の回転角度を求めなければならない。中間目標点 $P_i (x_i, y_i)$ における第1、第2関節の回転角度を、それぞれ θ_{1s}, θ_{2s} 、次の中間目標点 $P_{i+1} (x_{i+1}, y_{i+1})$ での回転角度を θ_{1e}, θ_{2e} とする。この2点間の角度変位を M 等分してロボットアームの移動角 θ_{1k}, θ_{2k} を算出すると次式ようになる。

$$\left. \begin{aligned} \theta_{1k} &= \theta_{1s} + \frac{\theta_{1e} - \theta_{1s}}{M} \cdot K \\ \theta_{2k} &= \theta_{2s} + \frac{\theta_{2e} - \theta_{2s}}{M} \cdot K \end{aligned} \right\} \quad (7)$$

$(K = 0, 1, 2, \dots, M)$

3.2 経路誤差

全節で述べた経路生成法による経路生成の妥当性を検討するために、指定した直線経路と実際にロボットアームが通る経路との一致性を調べる。指定した直線経路とロボットアームの軌跡とのズレを経路誤差と呼び、次式により計算できる。

$$D = \sqrt{\frac{\begin{vmatrix} x_s - x_k & y_s - y_k \\ x_e - x_s & y_e - y_s \end{vmatrix}^2}{(x_e - x_s)^2 + (y_e - y_s)^2}} \quad (8)$$

ただし、 (x_k, y_k) はアーム軌跡上の座標値である。こ

の経路誤差が小さいほど指定した直線経路とロボットアームが通った経路との一致性がよいことになる。

3.3 中間目標点のパス領域設定

ロボットアームは中間目標点を通して移動して行くのであるが、位置決めは最終目標点 P_E で行われれば良いので中間目標点には完全に到達しなくても良いことにする。すなわち、図3に示すように、アーム先端の位置と中間目標点 P_i との距離がある許容値 r 以下になったところで、アームを次の中間目標点 P_{i+1} へ移動する。この許容値 r を半径とし、中間目標点を中心に描かれる円の内側の領域をパス領域と呼ぶことにする。このようにすることで、移動角の計算回数が少なくなり、移動時間の短縮化がはかれる。また、2点間の直線を等分にできない場合にこのパス領域を調整することにより最適な中間目標点を得られる。

4. アーム移動シミュレーション

4.1 シミュレーション条件

シミュレーションは次のような条件で行った。

a) アーム長

第1アーム： $R_1=480$ [mm]

第2アーム： $R_2=320$ [mm]

b) 移動位置 ($P_S \rightarrow P_E$)

1) $(553.6, -153.6) \rightarrow (-153.6, 553.6)$

2) $(492.4, 386.8) \rightarrow (-492.4, 213.2)$

(移動距離：1000 [mm])

c) 分割数

2点間の分割数： $N=5$

角度変位の分割数： $M=20, 30$

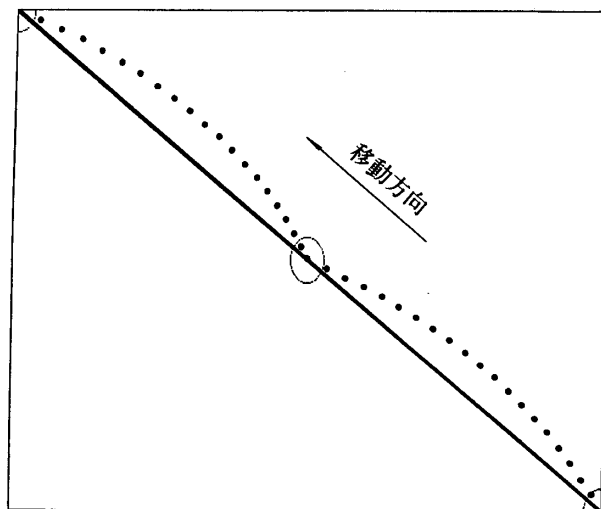
d) パス領域

許容値： $r=20$ [mm]

4.2 シミュレーション結果

ロボットアームの移動をシミュレーションした結果を図4、5に示す。図は、ロボットアームの移動状態を拡大したもので、アーム先端の軌跡をドットで示している。図から、 $M=20$ の場合、パス領域を設定したときアームの移動は20回から17回、 $M=30$ の場合、同じくアームの移動は30回から28回と少なくなっており、パス領域の設定により移動のための計算回数が減り経路生成の高速化が図れることがわかる。しかし、パス領域を設けること

によって経路誤差が大きくなるのが、図6と表1から理解できる。この図表の最大経路誤差とは、ロボットアーム移動中に発生した経路誤差の最大値のことである。2点間の分割数が40以上の場合、経路誤差の変化は0.1~0.2 [mm] となるので2点間の分割数は40前後の値を選ぶことで十分な経路制御ができると言える。

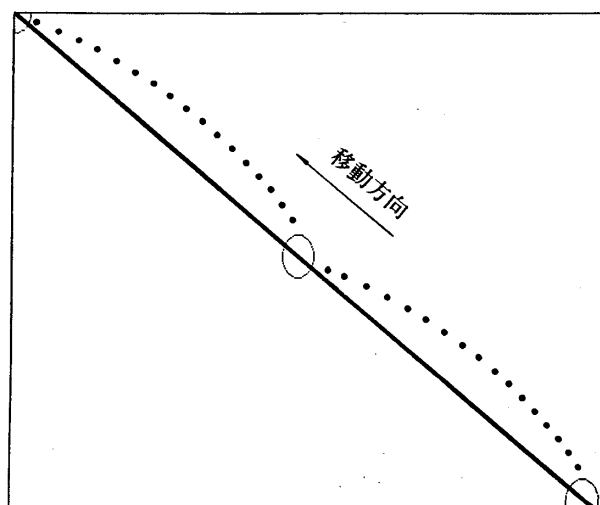


パス領域許容値 $r=0$ [mm]

移動点 $(553.6, -153.6) \rightarrow (-153.6, 553.6)$

角度変位分割数 $M=20$

図4-a ロボットアーム軌跡

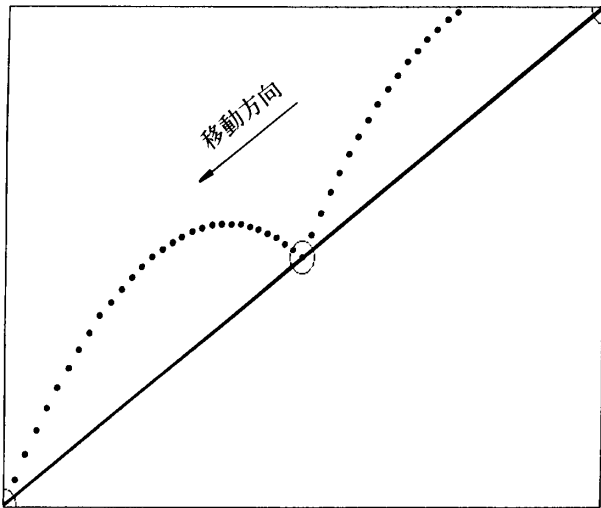


パス領域許容値 $r=20$ [mm]

移動点 $(553.6, -153.6) \rightarrow (-153.6, 553.6)$

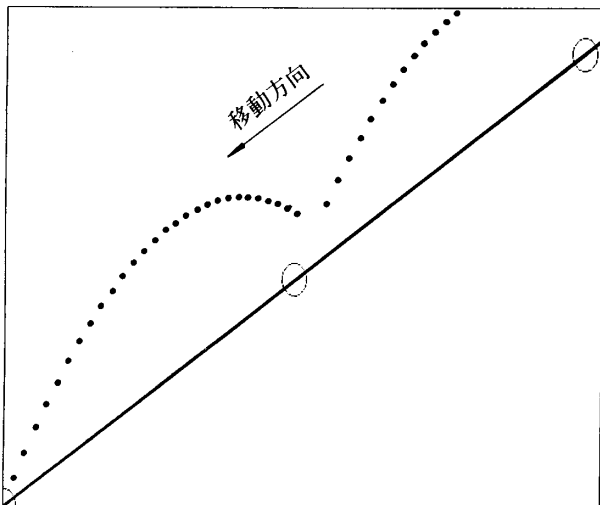
角度変位分割数 $M=20$

図4-b ロボットアーム軌跡



パス領域許容値 $r = 0$ [mm]
 移動点 (492.4, 386.8) \rightarrow (-492.4, 213.2)
 角度変位分割数 $M = 30$

図 5-a ロボットアーム軌跡



パス領域許容値 $r = 20$ [mm]
 移動点 (492.4, 386.8) \rightarrow (-492.4, 213.1)
 角度変位分割数 $M = 30$

図 5-b ロボットアーム軌跡

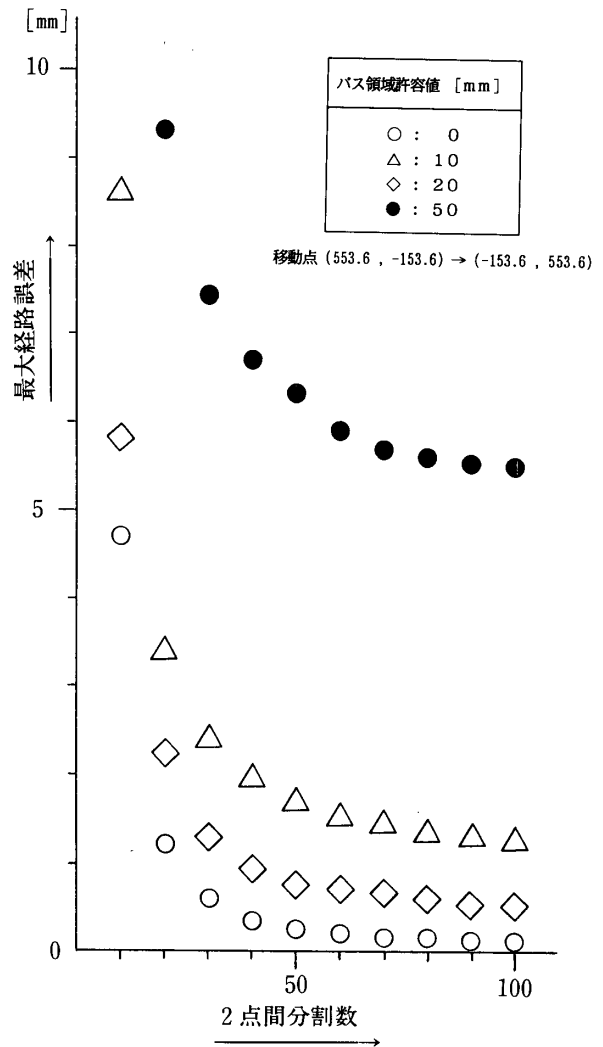


図 6 分割数と最大経路誤差との関係

表 1 各条件による最大経路誤差値

パス領域許容値 [mm]	0			50		
	5	50	100	5	50	100
2点間の分割数	5	50	100	5	50	100
移動距離に対する最大経路誤差 [%]	1.84	0.02	0.01	3.54	0.63	0.54

5. まとめ

ロボットアームを直線移動させるために、目標点間に中間目標点を設けて経路生成する方法について検討を行った。2自由度の水平多関節ロボットによるシミュレーションから、2点間の分割数が40程度であれば十分精度の良い経路制御ができることが確認された。また、パス領域を設けることでアーム移動の計算回数が減り経路生成の高速化が期待できることがわかった。

今後は、ロボットアーム、アクチュエータのダイナミクスを考慮したロボットモデルを開発し実機による経路

制御について検討を進めて行きたい。

参考文献

- 1) 中小企業事業団中小企業研究所編、柿倉正義監修：産業用ロボットの制御方式と利用技術、日刊工業新聞社
- 2) 日高：2自由度水平多関節ロボットの直線移動に関する基礎研究、宇部高専研究報告、第37号、平成3年

(平成3年9月24日受理)