

Title	ポテンシャル場を用いた関節ロボットの位置決め制御に関する研究
Author(s)	田中, 宏和
Citation	
Issue Date	1999-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1226
Rights	
Description	Supervisor:浅野 哲夫, 情報科学研究科, 修士

修 士 論 文

**ポテンシャル場を用いた関節ロボットの
位置決め制御に関する研究**

指導教官 浅野哲夫 教授

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

田中 宏和

1999 年 2 月 15 日

要 旨

本稿では、関節ロボットの位置決め制御において、冗長自由度を有効に利用するために、ポテンシャル場の考えを用いた新しい動作計画作成法を提案する．そして、シミュレーションを行なうことにより、その有効性を検証する．

目次

1	序論	1
1.1	はじめに	1
1.2	背景となる知識	2
1.2.1	コンフィギュレーション空間	2
1.2.2	ポテンシャル場	4
1.2.3	ドローネ網	5
1.3	研究の概要	6
2	前処理	8
2.1	幾何学的モデルによる解析	8
2.1.1	2関節ロボット	8
2.1.2	3関節ロボット	9
2.1.3	その他の関節ロボット	12
2.2	自由空間を求める	13
2.2.1	関節角座標空間	13
2.2.2	リンクどうしが交差する姿勢の判定	14
2.2.3	障害物と衝突する姿勢の判定	16
3	経路探索	19
3.1	目標姿勢の集合について	19
3.1.1	通過姿勢の場合	20
3.1.2	保持姿勢の場合	20
3.2	ポテンシャル場の作成と探索	21
3.2.1	ポテンシャル場の構造	21
3.2.2	ポテンシャル場の生成アルゴリズムの説明	23

4	シミュレーション	25
4.1	シミュレーション1：目標姿勢の形状について	25
4.1.1	結果	25
4.1.2	考察	30
4.2	シミュレーション2：動作計画	31
4.2.1	準備	31
4.2.2	結果	32
4.2.3	アルゴリズムの解析	36
5	まとめ	37

目 次

1.1	2 関節ロボットにおける作業空間 (a) と C 空間 (b) の対応関係	3
1.2	2 関節ロボットの関節座標空間における障害物 (点)	3
1.3	ポテンシャル場の等高線による表現	4
1.4	ボロノイ図とドロネ網	5
1.5	平面上を動作する 3 関節ロボット	6
1.6	従来の C 空間における動作計画	7
1.7	本稿が提案する C 空間における動作計画	7
2.1	2 関節ロボットの幾何学的モデル	8
2.2	3 関節ロボットの幾何学的モデル	9
2.3	$L_0 = 2$ における解の集合 (C 空間)	11
2.4	$L_0 = 2$ における解の集合 (作業空間)	12
2.5	リンクどうしが交差する姿勢	14
2.6	リンクが交差する姿勢を除いた解の集合	15
2.7	関節角座標空間の描画方法	15
2.8	障害物と接する姿勢	16
2.9	3 関節ロボットと 3 つの障害物	17
2.10	障害物 A と接する姿勢	17
2.11	障害物 B と接する姿勢	18
2.12	障害物 C と接する姿勢	18
3.1	通過姿勢の探索経路	20
3.2	保持姿勢の探索経路	21
3.3	目標点が 1 点のときのポテンシャル場の構造	22
3.4	目標点が複数のときのポテンシャル場の構造	22
3.5	アルゴリズムの説明図	23

4.1	条件 1 における目標姿勢（関節角座標空間）	26
4.2	条件 1 における目標姿勢（作業空間）	26
4.3	条件 2 における目標姿勢（関節角座標空間）	27
4.4	条件 2 における目標姿勢（作業空間）	27
4.5	条件 3 における目標姿勢（関節角座標空間）	28
4.6	条件 3 における目標姿勢（作業空間）	28
4.7	条件 4 における目標姿勢（関節角座標空間）	29
4.8	条件 4 における目標姿勢（作業空間）	29
4.9	関節角座標空間の構造	32
4.10	作業空間における初期状態	33
4.11	$10^3(1000)$ 節点による実験結果（関節角座標空間）	34
4.12	$10^3(1000)$ 節点による実験結果（作業空間）	34
4.13	$16^3(4096)$ 節点による実験結果（関節角座標空間）	35
4.14	$16^3(4096)$ 節点による実験結果（作業空間）	35

表 目 次

2.1	2 関節の符号の組合せ	9
2.2	3 関節の符号の組合せ	10
4.1	シミュレーション 1 における条件	25
4.2	障害物の座標	33
4.3	各プロセスの平均実行時間	33

第 1 章

序論

1.1 はじめに

近年，工場の生産ラインや人が直接作業できない環境において，人のかわりにロボットを用いて作業をおこなう機会が増えてきた．しかし，ロボットの教示には大変に時間がかかる．またロボットを1台設置する費用で何人もの人間を雇うことができる，などの理由により，それらの作業は人手により行なわれているのが現状である．工場においては，短いサイクルで次々に新しい製品を作ることが要求されるので，特に教示時間の短縮が求められている．

工場で使われるロボットのうち，おもに回転と旋回の自由度から構成されるものは関節ロボット¹ [1] と呼ばれ，3次元空間内で任意の位置・姿勢をとることができる．そのため，様々な作業に対する適応性に優れているが，かわりに座標計算や制御は複雑である．この関節ロボットの制御法の一つに位置決め制御（P T P制御）と呼ばれるものがある．位置決め制御はロボットのエフェクター部が，経路に関係なく与えられた目標位置に達することのみを要求する制御である．スポット溶接や組み立て作業などは，経路上の有限個の通過点（通過姿勢）と作業点（作業姿勢）を教示し，実際の作業においては，次々に与えられた目標点（目標姿勢）に対して位置決め制御を繰り返す．また，アーク溶接や塗装作業は，位置決め制御における通過点をさらに細かくとることにより経路を憶えさせ，位置決め制御による疑似的な経路決め制御（C P制御）により行なうこともできる．

従来の位置決め制御における教示は目標姿勢を憶えさせるが，本論文では目標点を満たす目標姿勢の決定も動作計画に組み込むことにより，教示作業の簡略化と高速化をはかる．

¹日本ロボット学会「ロボット学術用語集」に基づいた，慣用的に使われている語句である．

1.2 背景となる知識

関節ロボットの位置決め制御における動作計画は、初期姿勢と目標姿勢を前もって与えることにより、その2点間の経路を求めるものであった。しかし、コンピュータの計算能力の向上とともに、作業環境のモデルを数値データとして蓄え、シミュレーションにより動作計画を立てる方法が主流になりつつある。実際に関節ロボットを動作させながらの教示では、目標姿勢まで動かすことができれば経路もかならず存在したが、シミュレーションでは目標姿勢をとることができても、障害物などによりその姿勢までの経路が存在しない場合や無駄に長くなってしまうことがある。関節ロボットに冗長自由度が存在する場合には、手先が同じ目標点を満たす姿勢は複数存在し、この複数存在する目標姿勢の選び方[2][3]によっては、初期姿勢からの経路の長さがかかわってしまう。また、他の目標姿勢を選べば経路が存在するにもかかわらず、経路の存在しない目標姿勢を選んでしまう可能性もでてくる。経路の有無や長さも考慮に入れた目標姿勢の選択が必要であるが、現在そのような研究はほとんどなされていない。経路が存在する目標姿勢を必ず選ぶことができると、教示において手先の目標位置を決定するだけで、姿勢の指定を行なう必要がなくなる。また、マスタスレーブ方式などの遠隔操作時においても同様に、手先位置の座標の入力だけが必要となり操作が非常に楽になる。

初期姿勢から求めた目標姿勢までに経路が生成できるか探索を行なう経路探索において、よく用いられる手法にコンフィギュレーション空間² [4] とポテンシャル場 [5] がある。ヤコビ行列の疑似逆行列を生成して計算により求める方法 [6][7][8] もあるが、本稿においてはC空間とポテンシャル場を用いて、与えられた手先位置を満たす目標姿勢の決定および経路探索を行なった。以下にC空間とポテンシャル場の大きな説明を述べる。また、幾何学的構造であるドロネー網も利用したので簡単に説明する。

1.2.1 コンフィギュレーション空間

関節ロボットの姿勢は、作業空間とC空間の両方で記述することができる。ここで用いる作業空間とは、関節ロボットのベース座標系をもとに実際に作業を行なう空間を直交座標系と考えたものである。この空間においては図 1.1-a に示すように、関節ロボットは実際の姿勢を見たままの状態であらわすことができる。それに対してC空間は、リンク座標系どうしの偏角や変位を軸としたパラメータ空間である [9]。この空間においては図 1.1-b に示すように、関節ロボットのどんな姿勢も1点であらわすことができる。

²以下C空間と表記がある場合には、コンフィギュレーション空間を指すものとする。

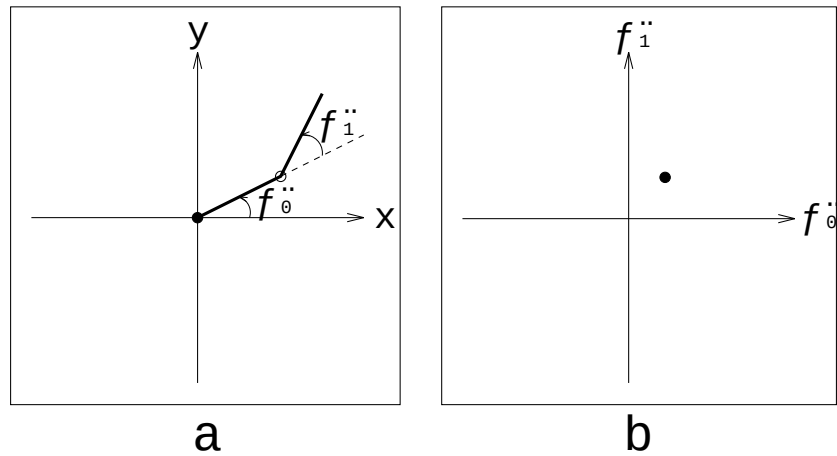


図 1.1: 2 関節ロボットにおける作業空間 (a) と C 空間 (b) の対応関係

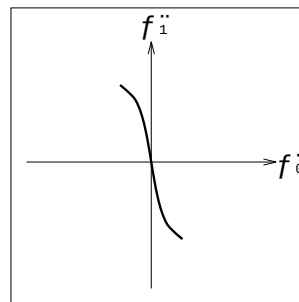


図 1.2: 2 関節ロボットの関節座標空間における障害物（点）

そのかわりに，C空間において障害物の形は複雑になってしまう．例えば，図 1.1-a の 2 関節ロボットにおいて x 軸上正の方向の 2 リンク目のとどく範囲に点の障害物がある場合，作業空間における障害物（点）が関節座標空間では図 1.2 に示すように S 字形の曲線になってしまう．C空間において障害物の形状をすばやく求める研究には [10] [11][12][13] などがある．しかし，そのほとんどは 2 リンクから構成されるロボットに関してだけしか記しておらず，3 リンク以上の関節ロボットに対する障害物をすばやく記述する研究はなされていない．

C空間を用いた経路探索は，まず初期姿勢に対応する初期点と目標姿勢に対応する目標点を与え，その 2 点の間をつなぐ経路を探索する方法が一般的である．このような 2 点間の経路探索に関しては様々な研究がなされている．C空間をセル（小領域）に分割し，ロ

ボットの経路として隣接するセルの系列を求める方法 [14], その分割を効率的かつ効果的に行なうために, 形状の複雑さに応じて階層的にセルの大きさを変化させる四分木を用いた方法 [15][16], 格子状のグラフで表し, 複雑な空間は四分木のように更に細かく分割する方法 [17], ランダムに決めた姿勢どうしを結んでグラフであらわす方法 [18][19][20][21], 計算幾何学における可視グラフの問題としてグラフ理論を利用する方法, 幾何学的構造を利用した方法, ポテンシャル場を利用する方法, またはそれらを組み合わせた方法などがある [22]. また, C空間の次元数はロボットの関節数と等しいので, もともと自由度が多いロボットに対しては, いくつかのC空間をまとめて表現する方法 [23] や部分的に仮想リンクに置き換える方法 [24] などがある. その他にも, 大局的な観点から経路が乗るであろう自由曲面をC空間内に生成し, その曲面上で経路探索する方法 [25] などがある.

1.2.2 ポテンシャル場

ポテンシャル場を用いた経路探索の方法は, ロボットの周囲に仮想的なポテンシャルを作るところからはじまる. 通常は目標点にマイナスのポテンシャル (引力) を与え, それぞれの障害物に対してはプラスのポテンシャル (斥力) を与える. あとは, ポテンシャルの勾配にそってロボットを運動させ, その軌跡を記録することにより経路が求まる. 図 1.3 にポテンシャル場を等高線であらわしたものを示す. この図においてポテンシャルは, 目標点から順々に増加しながら波紋のように広がってゆき, 障害物の外周においてもっとも高くなっている. 初期点の位置にロボットをおくと, 障害物を左手方向に見ながら回り込むように目標点にちかづく経路が生成される. ポテンシャル場はC空間で用いられるだけでなく, 作業空間において用いられることもある [26].

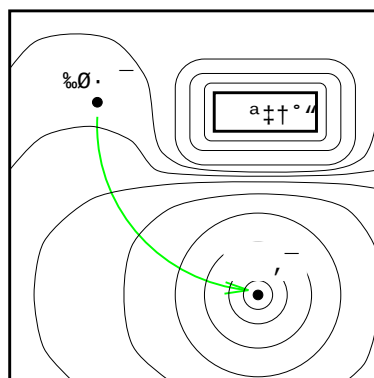


図 1.3: ポテンシャル場の等高線による表現

ポテンシャルを発生させるための関数には、目標点に対するベクトルを計算するものや障害物の表面から単調に減少する関数などが用いられる。しかし、いくつかのポテンシャル関数を別々に計算するために、障害物の配置によっては極小点が発生してしまい、目標点まで到達できない場合が生じる。この問題に対しては、色々な手法が提案されている。まず、障害物により発生するポテンシャルを改良することにより極小点をなくするという手法 [27][28][29][30]。つぎに、障害物を含めた空間全体を目標点から波紋が広がるように等ポテンシャル面を順次形成してゆくことにより、極小点のないポテンシャル場を生成する手法として [31][32][33] やラプラスの微分方程式の解を用いる方法 [34]、電界をポテンシャルに見立てる方法 [35] などがある。かわったものとしてはニューラルネットワークを利用したポテンシャル場の自動生成法 [36] などもある。他には、空間全体にポテンシャルを考えずに障害物のポテンシャルを工夫することにより、まえもって与えられた経路が障害物をよけるように修正する方法 [37][38][39] やポテンシャルを用いて動作時間が最適化される方向に修正する方法 [40][41][42]、空間を量子化するなどして領域分割を行ない、その領域ごとに一つのポテンシャルがわりふられるようにする手法 [43] などが提案されている。

1.2.3 ドローネ網

ドローネ網はドローネ三角形分割ともいわれ、平面上に与えられた点集合に対し、それらの点を頂点とする三角形メッシュ³を形成する。更に、3次元においては4面体分割となり、それ以上の次元に対しても定義されている。

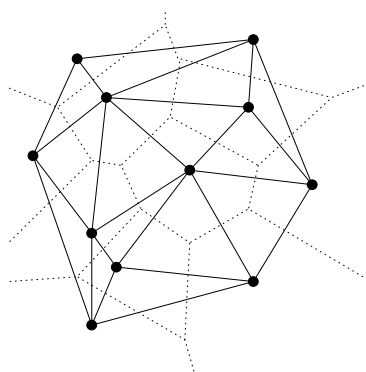


図 1.4: ボロノイ図とドローネ網

³分割された三角形において最小の角が最大となる。

ドロネ網の作成法には、フリッピング法、逐次追加法、分割統治法など [45] のほかにボロノイ図を利用する方法もある。ボロノイ図はドロネ網の双対グラフであり、平面上に与えられた点集合に対し、それらの点の勢力範囲図を形成する。この勢力範囲図において、個々の点はそれぞれ一つの領域に対応しており、その領域内のどの位置に対しても、対応する点が一番近いという性質 [44] がある。図 1.4 に 11 点を与えられた時のボロノイ図とドロネ網を示した。図において点線で書かれているのがボロノイ図であり、実線で書かれているのがドロネ網である。ドロネ網においてつながっている点どうしの関係は、ボロノイ図において隣あった領域の点どうしを線分でむすび、ボロノイ図自身を消したものである。最短木を作る際には、ドロネ網の辺は最短木の候補となるので、その性質を利用することにより全ての点の組合せを調べないですむ。また、ドロネ網の外周は凸包に等しくなるなど、隣あった領域の点どうしが幾何学的に距離が近いという性質を利用したアルゴリズムは数多く存在する。

1.3 研究の概要

本稿では、平面上で動作する 3 関節ロボットを例に、冗長自由度を利用した目標姿勢の決定と経路生成を同時に行なう手法を説明する。図 1.5-左に本稿で用いる 3 関節ロボットの作業空間における姿勢を示した。図において黒丸が根元の関節、白丸がそれ以外の関節をあらわしたもので、それぞれの関節における角度⁴ は図のように θ_0 , $-\theta_1$, θ_2 となる。

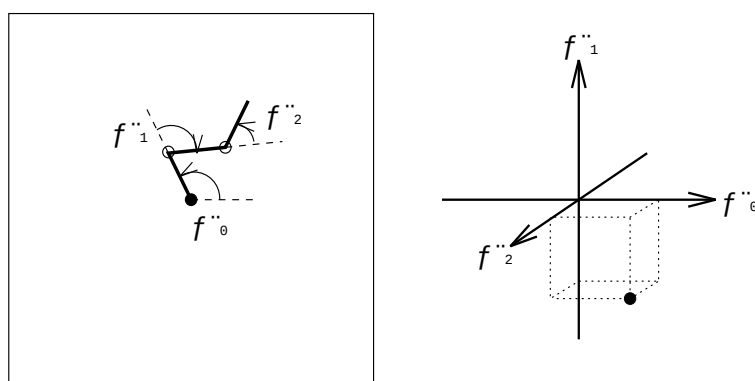


図 1.5: 平面上を動作する 3 関節ロボット

⁴最初の関節の角度は根元を通る水平線からの偏角、それ以外の関節の角度は一つ手前（根元に近い方）のリンクの延長線上からの偏角で、すべての角度は半時計方向を正とする。

図 1.5-右は，それらの角度をもとに 3 関節ロボットの姿勢を C 空間で表現したもので， $\theta_0, \theta_1, \theta_2$ の 3 軸から構成され，右下にある点 $(+\theta_0, -\theta_1, +\theta_2)$ が対応する姿勢をあらわす．

関節ロボットの従来の動作計画は，図 1.6 に示すように初期姿勢 (A) と目標姿勢 (B) を与え，その 2 点間の経路を求めるものであった．しかし，目標点をシミュレーションによって教示すると，障害物の配置などによって初期点と目標点の間に経路が存在しない場合や経路が無駄に長くなる場合が生じてしまう．関節ロボットに冗長自由度が存在するとき，目標手先位置をみたす姿勢は複数存在しえるので，本稿においては目標点を前もって 1 点に限定せずに複数個あたえる．図 1.6 の姿勢 B と同じ手先位置 (灰色の丸) をみたす姿勢は図 1.7 に示すように複数ありえるが，最終的にどの目標点に達するかは，ポテンシャル場によって決まる．

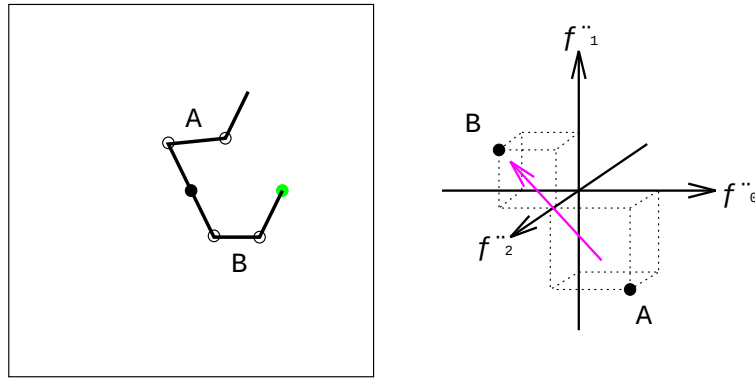


図 1.6: 従来の C 空間における動作計画

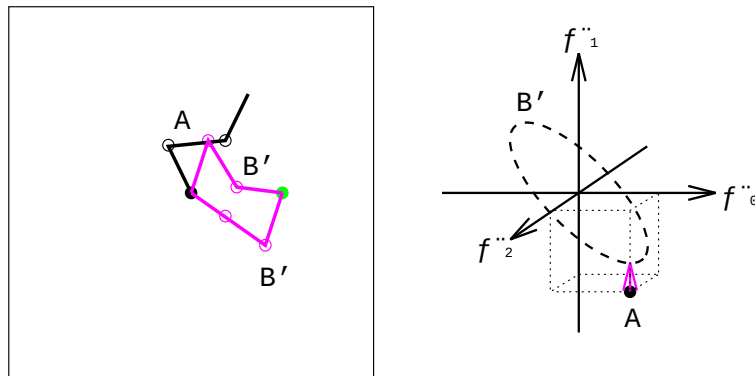


図 1.7: 本稿が提案する C 空間における動作計画

第 2 章

前処理

2.1 幾何学的モデルによる解析

同じ手先位置をみたす姿勢の集合をえるためには，関節ロボットの構造を解析し定式化する必要がある．3 関節ロボットについて解析する前に，まずは簡単な 2 関節ロボットについて説明する．

2.1.1 2 関節ロボット

平面上を動作する 2 関節ロボットは，図 2.1 のように 2 つのリンク l_0 , l_1 とその間の関節（白丸）と根元の関節（黒丸）からなる．ロボットの根元と手先を通る直線を引いたとき，根元と手先の距離を L とすれば，この図は L , l_0 , l_1 を 3 辺とする三角形と考えることができる．

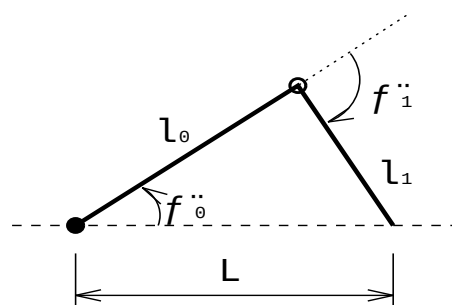


図 2.1: 2 関節ロボットの幾何学的モデル

このときの関節の角度を根元から順に θ_0 , θ_1 とすれば, L の範囲が $|\ell_0 - \ell_1| \leq L \leq \ell_0 + \ell_1$ のときに三角形が成り立ち, 余弦定理より θ_0 , θ_1 は次のように定式化できる.

$$\theta_0 = \cos^{-1} \left(\frac{L^2 + \ell_0^2 - \ell_1^2}{2L\ell_0} \right), \quad (2.1)$$

$$\theta_1 = \cos^{-1} \left(\frac{L^2 - \ell_0^2 - \ell_1^2}{2\ell_0\ell_1} \right). \quad (2.2)$$

また, θ_0 と θ_1 の符号の組合せは, 表 2.1 にしめすように, θ_0 が正のとき θ_1 は負になり, θ_0 が負のとき θ_1 は正になる.

表 2.1: 2 関節の符号の組合せ

θ_0	θ_1
$+\theta_0 \rightarrow -\theta_1$	
$-\theta_0 \rightarrow +\theta_1$	

2.1.2 3 関節ロボット

平面上で動作する 3 関節ロボットは, 手先の位置を固定しても冗長自由度のために複数の姿勢をとることができ, 定式化が難しい. そこで図 2.2 にしめたように 2 つの三角形と考えることにより, 定式化を可能にした.

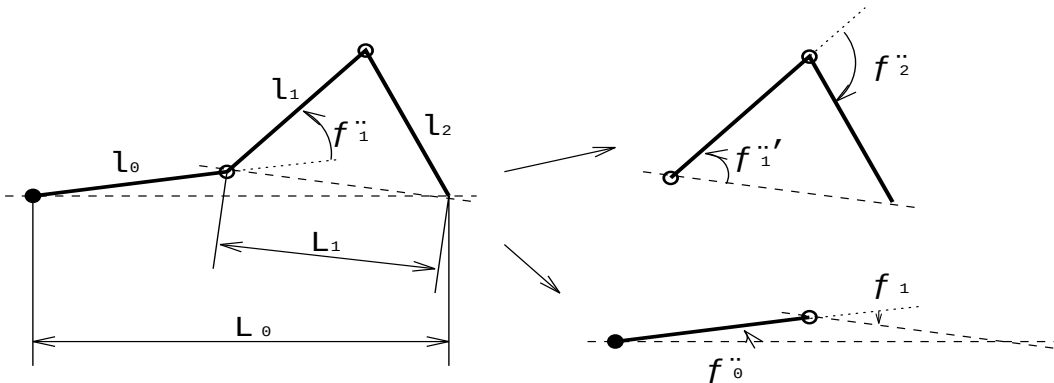


図 2.2: 3 関節ロボットの幾何学的モデル

リンクは根元（黒丸）に近い方から順に ℓ_0, ℓ_1, ℓ_2 とした。また、ロボットの根元と手先を通る直線を引いたとき、根元と手先の距離を L_0 とし、第1関節から手先までの距離を L_1 とした。 L_0, ℓ_0, L_1 を3辺とする三角形が成り立つのは、 L_0 の範囲が $\min(|\ell_0 - L_1|) \leq L_0 \leq \ell_0 + \ell_1 + \ell_2$ のときである。また、 L_1, ℓ_1, ℓ_2 を3辺とする三角形が成り立つのは、 L_1 の範囲が $\max(|L_0 - \ell_0|, |\ell_1 - \ell_2|) \leq L_1 \leq \min(L_0 + \ell_0, \ell_1 + \ell_2)$ のときである。

このとき、関節の角度を根元から順に $\theta_0, \theta_1, \theta_2$ とし、 L_1 と ℓ_1 の間の角度を θ'_1 とすれば、余弦定理より以下の式が導かれる。

$$\theta_0 = \cos^{-1} \left(\frac{L_0^2 + \ell_0^2 - L_1^2}{2L_0\ell_0} \right), \quad (2.3)$$

$$\phi_1 = \cos^{-1} \left(\frac{L_0^2 - \ell_0^2 - L_1^2}{2\ell_0 L_1} \right), \quad (2.4)$$

$$\theta'_1 = \cos^{-1} \left(\frac{L_1^2 + \ell_1^2 - \ell_2^2}{2\ell_1 L_1} \right), \quad (2.5)$$

$$\theta_2 = \cos^{-1} \left(\frac{L_1^2 - \ell_1^2 - \ell_2^2}{2\ell_1 \ell_2} \right). \quad (2.6)$$

このとき、 θ_1 は以下の式であらわせられる。

$$\theta_1 = \phi_1 \pm \theta'_1. \quad (2.7)$$

また、 $\theta_0, \theta_1, \theta_2$ を構成する組合せは、表 2.2 に示した4通りであった。 θ_0 が正のとき ϕ_1 の符号は負になり、 θ_0 が負のとき ϕ_1 の符号は正になる。同様に、 θ'_1 が正のとき θ_2 は負になり、 θ'_1 が負のとき θ_2 は正になる。目標手先位置を満たす姿勢は、 θ_0 と ϕ_1 と θ'_1 と θ_2 を計算で求め、組合せにならない4回計算すればすべて求めることができる。

表 2.2: 3関節の符号の組合せ

θ_0	θ_1	θ_2
$+\theta_0 \rightarrow -\phi_1$	$+\theta'_1 \rightarrow -\theta_2$	
	$-\theta'_1 \rightarrow +\theta_2$	
$-\theta_0 \rightarrow +\phi_1$	$+\theta'_1 \rightarrow -\theta_2$	
	$-\theta'_1 \rightarrow +\theta_2$	

以上の結果をもとに、3関節ロボットの3つのリンクの長さをすべて1とした場合の $L_0 = 2$ における手先位置の等しい姿勢の集合をC空間で示したのが図 2.3である。この図中の原点を通る3本の直線は右手系の順で右向きの水平な矢印が θ_0 を表す軸で、上向きの垂直な矢印が θ_1 を表す軸で、左下に向かう矢印が θ_2 を表す軸である。図の左端と下に単位を表す軸が加えてあり、示された数値の単位はラジアンである。また、図 2.3のA, B, C, D, Eは、それぞれ図 2.4の同じアルファベットの姿勢に対応している作業空間における3関節ロボットの姿勢である。図 2.4において、5つの姿勢がAは実線で、Bは一点鎖線で、Cは細かい点線で、Dは破線で、Eは点線で示されている。関節ロボットの根元の関節は原点の位置（左端）にあり、手先の関節の位置は座標 $(2, 0)$ （右端）である。Aから順番に見てゆくと、まず $\theta_0 = 0$ で $\theta_1 > 0$ かつ $\theta_2 < 0$ の姿勢から徐々に θ_0 の値が大きくなってゆき、Bを経由してCの姿勢に至る。Cの姿勢は θ_0 が最大値をとっており、このとき L_1 も最大値をとっており $\theta_2 = 0$ である。そのあとは θ_0 の値が徐々に小さくなってゆき、 $\theta_1 < 0$ かつ $\theta_2 > 0$ となりDの姿勢を経由してEに至る。

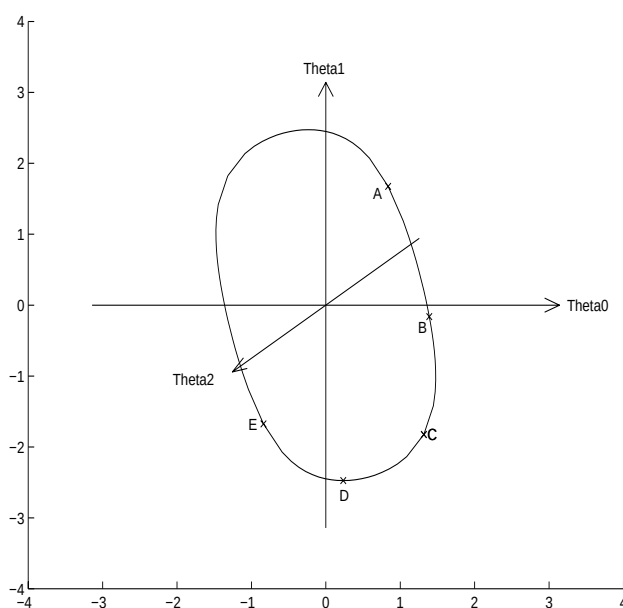


図 2.3: $L_0 = 2$ における解の集合（C空間）

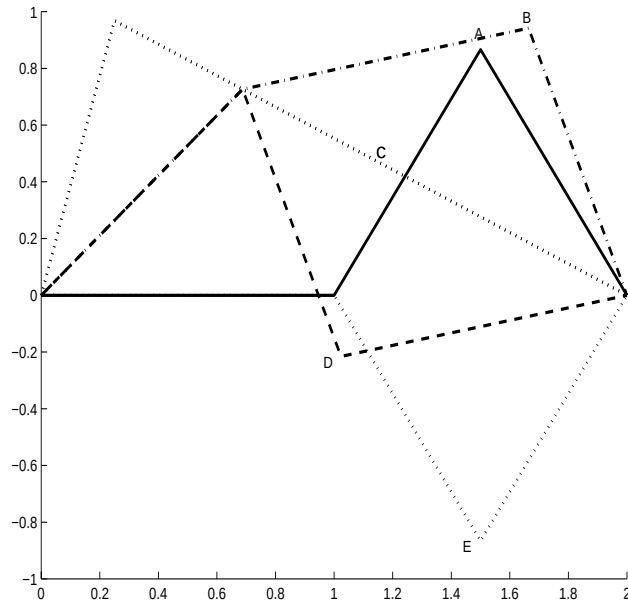


図 2.4: $L_0 = 2$ における解の集合（作業空間）

2.1.3 その他の関節ロボット

本研究において3関節ロボットの関節といえば、旋回または回転するものとし扱っている。C空間のそれぞれの軸は偏角をあらわしている。平面上を動作する関節ロボットにおいて、直動する関節を持つロボットを扱いたければ、C空間においてその関節に対応する軸を偏角から変位にかえるだけでよい。例えばC空間において θ_2 の軸の変わりに L_1 を軸として用いれば、先端のリンクの長さが変化する2関節ロボットをあらわすことになるし、 l_1 を軸として用いても同じ2関節ロボットをあらわすことができる。

また、多関節ロボットに対しては関節の数が n 個だとすれば、根元のリンクから順番に $L_0, L_1, \dots, L_n$ と計算してゆけば、旋回と回転を行なう関節を有するロボットを同様の手法で定式化でき、同じ目標手先位置に対する姿勢の集合を求めることができる。直動関節が含まれる場合には、C空間においてその関節をあらわす軸を θ から l に変更すれば、同様に同じ手先位置に対する姿勢の集合を求めることができる。

2.2 自由空間を求める

C空間内の経路探索において自由空間とは、関節ロボットが障害物と接することなく動き回れる空間を表す。本論文では前もって障害物の位置がわかっているものとし、関節角座標空間¹における自由空間を求める。自由空間をえるためには、まず関節角座標空間をどのようなデータ構造で構成するかが問題となる。

2.2.1 関節角座標空間

関節角座標空間の表現方法として空間内のすべての座標を考えると、データ量が非常に多くなってしまうので量子化を行なう。本論文では L.Kavraki と J.C.Latombe の方法 [18] にならい、空間全体をグラフで表すことによりデータ量を減らした。つまり、関節角座標空間内でランダムに姿勢をあたえていき、それらの姿勢を結ぶことによりグラフを作成した。ただし、彼らの方法とは違い関節角座標空間内の姿勢どうしをグラフで結ぶ際には、ドローネ網 [45][46] を用いた。ドローネ網を用いることにより、C空間をボロノイ領域に分割した際にその領域内にはいる姿勢をすべて、対応する1点であらわすことができる。また、ドローネ網の作成方法には色々あるが、本論文では逐次追加法を用いた。逐次追加法は、最初からすべての点に対してのドローネ網を作成せずに、点を適当な順番に一つずつ追加し、そのたびにドローネ網を順次更新してゆく方法である。この方法を用いて前処理として自由空間を求めておけば、目標手先位置が与えられてから目標姿勢の集合をその空間に追加し経路探索をおこなえるので、動作計画時間を短縮することができる。

関節角座標空間において一つの姿勢は、3つの角度の組合せによって表されているが、組合せによってはリンクどうしが交差したり障害物と衝突してしまうので、それらの姿勢は除く必要がある。前処理としては、まず関節角座標空間内にランダムに姿勢をあたえていくが、任意の角度の組合せがすべて実現できる姿勢ではないので、姿勢を与えるたびに実現可能か判定を行う。そして、最後に初期姿勢を加えれば自由空間が求まる。また、逐次追加法によりドローネ網が更新されてゆくが、新しく現れた辺において障害物と衝突する姿勢が含まれる可能性も生じるので、その判定も逐次行なう。

¹本研究で用いるC空間は3個の関節の偏角を軸とした空間で、本論文においてはこの空間を関節角座標空間と呼ぶことにした。

2.2.2 リンクどうしが交差する姿勢の判定

本論文で扱う関節ロボットは3つのリンクから成り立っており，同じ関節においてつながっているリンクどうしの交差判定と，連続していないリンクどうしの交差判定²が必要である．同じ関節においてつながっているリンクどうしの交差判定は，各関節の可動範囲を以下の様に決定することにより必要なくなる． $\theta_0, \theta_1, \theta_2$ を3軸とする関節角座標空間において，まず， θ_0 に関しては時計の針のように一方向に回転し続けることができるが，ちょうど1回転すれば元の姿勢に戻るので，可動範囲は $(-\pi, \pi]$ として，可動範囲を越えたものに関しては $\pm 2\pi$ することにより可動範囲内の値にもどす．つぎに， θ_1 と θ_2 に関しては，その関節においてつながっている2つのリンクどうしが重ならないように $(-\pi, \pi)$ を可動範囲とする．連続していないリンクどうしが交差する姿勢は図 2.5 のようにあらわすことができる．この図において， L_0 と L_1 と l_0 からなる三角形と L_1 と l_1 と l_2 からなる三角形に着目すると，三角形の外角の性質より以下の2式が導かれる．よってある姿勢においてリンクどうしが交差しているかどうかの判別は，この2式を同時に満たすか調べれば分かる．別の判別法としては，各リンクの両端の関節の位置は計算により求めることができるので，2線分の交差を調べれば判別できる．

$$\pi \leq \phi_1 + \theta'_1 \leq \pi + \phi_1, \quad (2.8)$$

$$\theta_2 - \theta'_1 \leq \phi_1 - \theta_0. \quad (2.9)$$

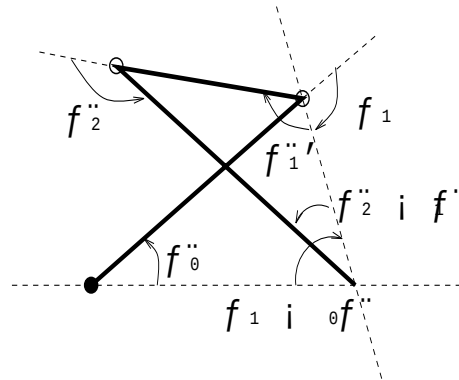


図 2.5: リンクどうしが交差する姿勢

²本論文では3関節ロボットを扱うため一つおきを意味するが，関節数が更に多い場合は二つおきや，三つおきの判定も含む．

3関節ロボットにおいてリンクの長さをすべて1とした時に、目標手先位置を根元から水平方向に0から3まで0.1おきに置いた場合の手先位置を満たす姿勢の集合を調べたものが図2.6である。この図において、原点の位置がちょうどリンクを伸ばしきった姿勢にあたり、外側の輪になるほど目標手先位置が根元に近付くことを表す。図の中央上部と下部が欠けているのは、その姿勢をとった場合にリンクどうしが交差すると判定されて除かれた結果である。ただし、図2.6は見やすいように少々工夫しており、図2.7のAの位置にあるものをBの位置に描き、同様にA'の位置にあるものをB'の位置に描いてある。³

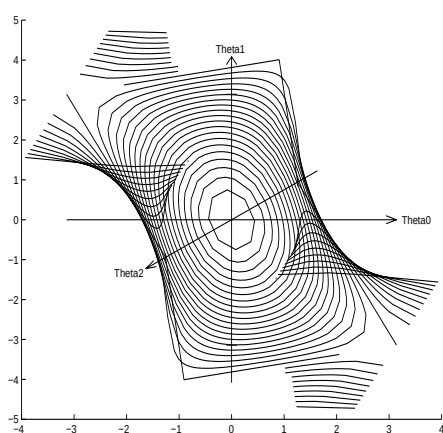


図 2.6: リンクが交差する姿勢を除いた解の集合

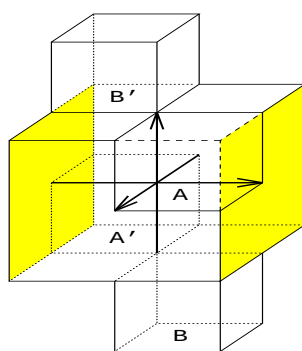


図 2.7: 関節角座標空間の描画方法

³本稿で扱う関節角座標空間はすべてこの様に描かれている。また、色付部分どうしは連続しており、空間としてはつながっている。

2.2.3 障害物と衝突する姿勢の判定

関節角座標空間における障害物領域は M.S.Branicky と W.S.Newman による方法 [10][11] を参考にした。彼らは平面上を動作する 2 関節ロボットについて、それぞれのリンクが障害物に接する姿勢をロボットの幾何学的構造から明らかにすることにより定式化し、C 空間における障害物を素早く記述する方法を見つけた。本稿では、この章の最初の節で関節ロボットの幾何学的構造を明らかにしてあるので、それらの式を障害物を求めるために使用した。図 2.8 には、リンクの長さがすべて 1 の 3 関節ロボットについて、根元から水平方向に 2.4 の距離に点の障害物が存在するとして、最初の関節の角度を 0 度に固定した場合の様子を示した。ロボットの先端が障害物の位置にある姿勢から、障害物に接したまま徐々に関節の角度を小さくしてゆき伸び切った姿勢になるまでを表している。これは、先端のリンクの障害物よりも先にいった部分が無視すれば、リンクの長さがすべて 1 の時に手先位置が根元から水平方向に 2.4 の位置にある姿勢から、目標手先位置を満たしたまま先端のリンクの長さを徐々に短くして 0.4 になるまでの姿勢と同じである。つまり、先端のリンクの長さが 1 から 0.4 の間であり、かつ手先位置が根元から 2.4 を満たす姿勢はすべて障害物と衝突する姿勢だとわかる。

図 2.9 はリンクの長さがそれぞれ 1 である 3 関節ロボットと根元から水平方向に 2.5 の位置に A、1.5 の位置に B、0.5 の位置に C の 3 つの障害物が置いてある。このそれぞれの障害物に対する障害物と接する姿勢の集合を以下に示した。ただし、データを離散的に計算したために本来は面であるべきものが線の集合であらわされていることを考慮して頂きたい。

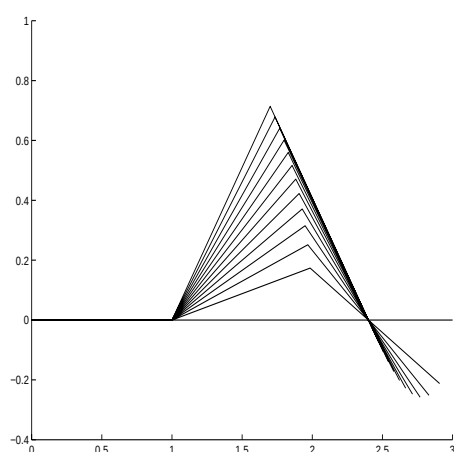


図 2.8: 障害物と接する姿勢

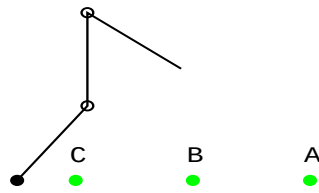


図 2.9: 3 関節ロボットと 3 つの障害物

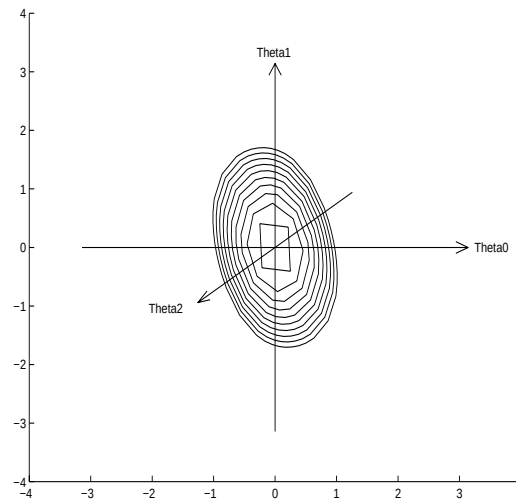


図 2.10: 障害物 A と接する姿勢

図 2.10では先端のリンクのみが障害物と接するので、円盤型をした領域が障害物と接する姿勢をあらわす．図 2.11では先端のリンクが障害物と接するとき，中央の空いた円盤上の領域が障害物と接する姿勢をあらわし，2 番目のリンクが接するときは， θ_0 と θ_1 のみによって障害物と接するか決まるので，その θ_0 と θ_1 の組合せを含む θ_2 がすべて障害物と接する姿勢をあらわす．図では曲面として描かれている．図 2.12で左上と右下に描かれているのが先端のリンクが障害物と接する領域であり，中央に描かれている曲面が2 番目のリンクと障害物が接する姿勢をあらわす領域である．根元のリンクと接するのは，障害物が水平方向にあるので $\theta_0 = 0$ の面がすべてそうである．

関節座標空間にランダムに与えられた姿勢の集合をグラフにつなぐ際に，障害物領域と交差する枝を除くことにより自由空間をあらわすグラフは求めることができる．

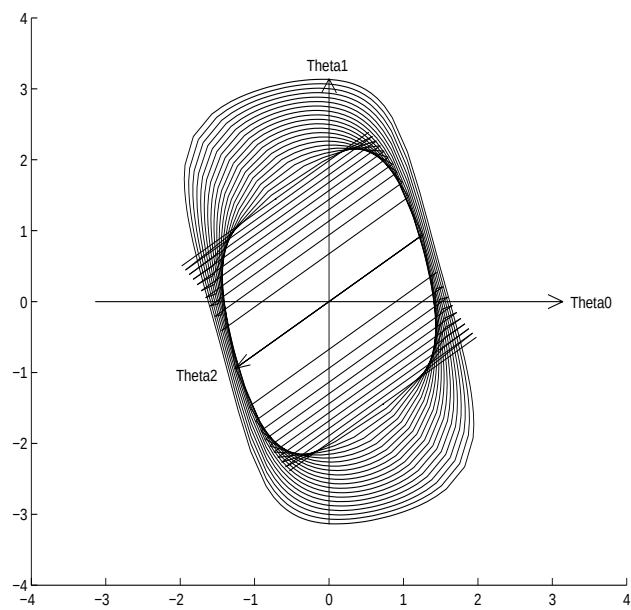


図 2.11: 障害物 B と接する姿勢

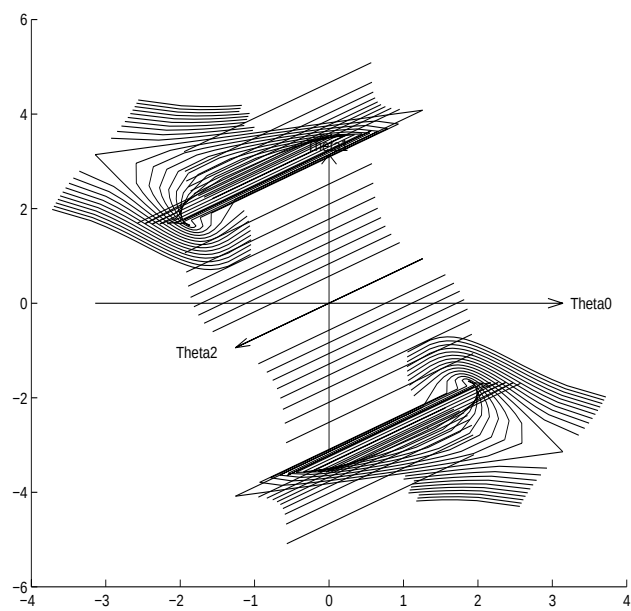


図 2.12: 障害物 C と接する姿勢

第 3 章

経路探索

動作計画のおもな流れは、はじめに前処理を行ない、まず目標手先位置を満たす姿勢の集合を求める。次に、求まった目標姿勢の集合から波紋が広がるようにポテンシャル場を形成してゆく。しかし実際には、関節角座標空間内にランダムに与えられた節点に対して通路を形成してゆき、この段階で同時に探索も行なわれる。最後に、求まった通路にそって関節ロボットを動作させれば動作計画は終了である。ただし、本論文によって求まった目標姿勢とそこまでの経路は関節角座標空間内の節点から節点への直線的な動きの連続であるために少々粗いので、実際に適応するときには他のアルゴリズムを用いて滑らかに動くように修正を行なうべきである。以下に、手順を更に細かく説明した。

3.1 目標姿勢の集合について

前の章において関節ロボットの同じ手先位置を満たす姿勢の集合を定式化したが、その姿勢の集合すべてを目標姿勢にするわけではなく、実際には作業目的に合わせて2種類の方法を使い分けて目標姿勢の集合を決定した。どちらの方法も関節ロボットの手先を目標位置まで素早く移動することを目的としているが、一つ目の方法は通過姿勢のように目標姿勢に移動後にまたすぐ次の目標姿勢が与えられるような、ロボットの姿勢を長時間保持する必要がない場合であり、二つ目の方法は目標姿勢に移動後しばらく同じ手先位置を保持しなければならないような場合である。以下にそれぞれの場合における目標姿勢の与え方の違いを述べた。

3.1.1 通過姿勢の場合

通過姿勢の場合には，定式化により求めた姿勢の集合がすべてがそのまま目標姿勢として与えられる．ただし，障害物領域に重なってしまったものやリンクどうしが交差してしまう姿勢はこの限りでない．実際には定式化により求めた式を用いて，障害物領域やリンクどうしが交差しないような目標手先位置を満たす姿勢を関節角座標空間内にランダムに何点か計算して与え，それらを目標姿勢の集合とする．あとは関節角座標空間内にポテンシャル場を形成し探索を行えば，目標姿勢のうちで初期姿勢からもっとも素早く移行できる姿勢までの経路を得ることができる．図 3.1では，障害物領域により2つに分かれた目標姿勢の集合のうち，近い目標姿勢までの経路が求めた様子を表している．

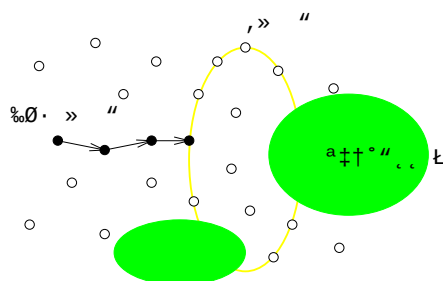


図 3.1: 通過姿勢の探索経路

3.1.2 保持姿勢の場合

関節ロボットが与えられた目標手先位置を長時間保持しなければならない場合は，与えられた手先位置を満たす姿勢の中で最適な姿勢¹をとった方が姿勢を維持するために必要なエネルギーが少なくてすむ．素早く目標手先位置を満たし，かつ最適な姿勢をとるためには目標姿勢に工夫が必要である．この場合は，最適姿勢からたどってゆき障害物領域に初めて接するまでを目標姿勢の集合とする．こうすることにより目標姿勢のどれかにとりあえずたどり着き，手先位置を満たしたまま最適姿勢まで移行できる．あとは通過姿勢と同様に目標姿勢の集合に対するポテンシャルを与えるだけでよい．ただし，目標姿勢の集合だけに対しても最適な姿勢に対するポテンシャルを与える必要がある．

¹ ロボットの関節の位置が指定されている場合にはそれを満たす姿勢であり，それ以外にも関節の位置をなるべく低くして姿勢維持に必要なエネルギーを減らす姿勢などが当てはまる．

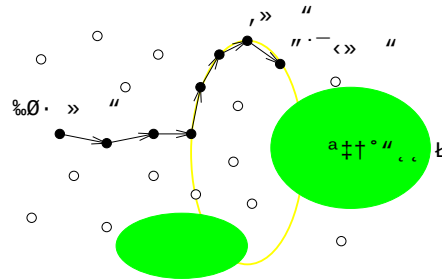


図 3.2: 保持姿勢の探索経路

図 3.2では目標姿勢は障害物領域によって2つに分断されているが、最適姿勢が含まれる側の同じ手先位置を満たす姿勢の集合だけを目標姿勢としている．この方法により目標姿勢のうちで初期姿勢からもっとも素早く移行できる姿勢にまず移行し、そのあとで手先位置を満たしたまま最適な姿勢へと移行できるようになる．

3.2 ポテンシャル場の作成と探索

一般的なポテンシャル場は、障害物に対しては斥力、目標姿勢に対しては引力を生じるようなポテンシャルを与えることにより形成される．しかし、本論文ではすでに関節角座標空間から障害物領域を除いた自由空間がグラフとして求まっているので、目標姿勢の集合に対するポテンシャルと最適な姿勢に対するポテンシャルだけを考える．最適な姿勢に対するポテンシャルは、目標姿勢の集合を式から求める段階で同時に目標姿勢の集合のみでグラフを作成し、最適な姿勢から順番に手先位置を満たしたまままたどれる範囲を最終的な目標姿勢とすることにより求まる．

3.2.1 ポテンシャル場の構造

グラフで表された自由空間における、目標姿勢の集合にたいするポテンシャルを説明する前に、まず目標姿勢が1点である場合について説明する．ポテンシャル場の性質としては、まず目標点までたどり着くことが保証されるべきであり、あらゆる姿勢においてポテンシャルが簡単に計算でき、また目標姿勢に向かって単調に減少していなければならない．つまりグラフで表したときに、かならず目標姿勢の節点までつながっている必要があり、目標姿勢に向かって単調に減少させるためには、ポテンシャル場は閉路や循環路を含

まな木であらわすべきである。しかも目標姿勢の節点を根にすることにより，ある姿勢におけるポテンシャルはその節点の深さで簡単に説明でき，また目標姿勢に向かうと親節点では確実に深さは減少してゆく。以上の議論をまとめると，グラフで表された自由空間において目標点が1点である場合のポテンシャル場は，図 3.3に描いたように目標姿勢の節点を根とする最短木²によりうまく表現することができる。目標姿勢が複数の場合には図 3.4に描いたように，各目標姿勢にたいして目標姿勢を根とした木が生成される。木を生成する手順は，幅優先探索を並列に行なうようにすべての木において同じ深さの節点³を決定してゆき，同じ深さの節点が終了したあとに次の深さの節点を決定してゆく方法で，同時にすべて目標姿勢にたいして木を形成してゆく。ポテンシャル場の形成は初期姿勢の節点が現れるか，またはつなぐ節点なくなるまで続けられる。この方法を用いれば，初期姿勢の節点が見つかった時点で目標節点とそこまでの通路が同時に決定される。つまりポテンシャル場の形成後に再び探索を行なわないですむ。

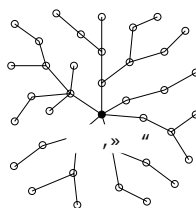


図 3.3: 目標点が1点のときのポテンシャル場の構造

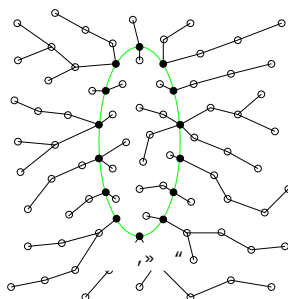


図 3.4: 目標点が複数のときのポテンシャル場の構造

² 本論文においてグラフ作成のみにドローネ網を用いたのはこのためである。

³ 同じ深さの節点どうしをつないでいくと等ポテンシャル面が形成できる。

3.2.2 ポテンシャル場の生成アルゴリズムの説明

ポテンシャル場の生成と探索を行なうために考えたアルゴリズムMAKEPOTENTIALを説明する。MAKEPOTENTIALは、引数としてドローネ網から障害物と交わる姿勢を除いたグラフ Del と初期姿勢 P_{init} と目標姿勢の集合 S_{goal} をとる。出力としてポテンシャル場が形成され、探索が成功した場合には目標姿勢と通路が求まる。アルゴリズム内で一時的にデータを蓄えるスタックとして $OPEN$ 、 $CLOSED$ 、 $DONE$ を用意した。このアルゴリズムは、まず目標姿勢の集合を $OPEN$ に蓄えるところからはじまって、 $OPEN$ から毎回一つの節点を取り出してはつながる枝を調べて次の等ポテンシャル面となる節点を探す。そして見つけるたびにその節点を $CLOSED$ に蓄えてゆき、 $OPEN$ 内の節点をすべて調べ終わると、 $CLOSED$ 内にある節点をすべて空になった $OPEN$ に移し、再び次の値の等ポテンシャル面を形成しはじめる。等ポテンシャル面が形成されるたびに、その等ポテンシャル面を形成している節点は $DONE$ に蓄えられてゆき、 P_{init} が等ポテンシャル面として見つかるか調べる節点が無くなるまでアルゴリズムは続けられる。 P_{init} が見つかったのであれば、 P_{init} から枝をたどってゆけば最終的な目標姿勢とその姿勢までの通路が見つかる。アルゴリズムの動作過程を図3.5に示した。

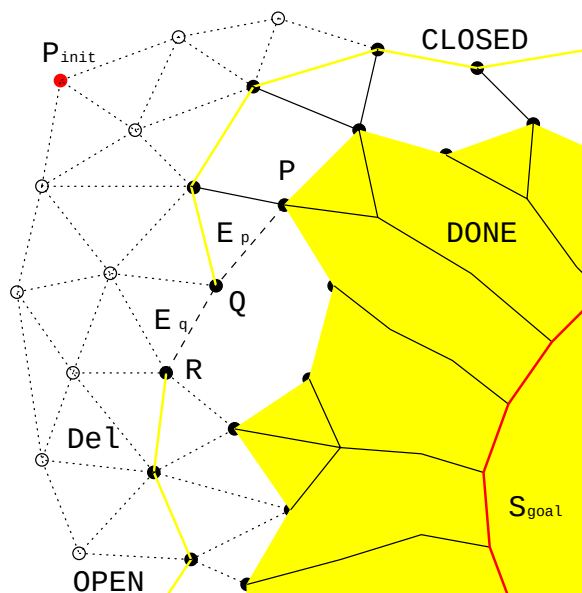


図 3.5: アルゴリズムの説明図

Algorithm MAKEPOTENTIAL(Del, P_{init}, S_{goal})

Input. グラフ Del と初期節点 P_{init} と目標節点の集合 S_{goal} .

Output. ポテンシャル場の形成され, 探索結果がわかる.

1. S_{goal} の各点を $OPEN$ に入れる.
2. **while** ($OPEN$ が空でなくかつ節点 P_{init} が含まれていない)
3. **while** ($OPEN$ が空でない)
4. $OPEN$ から節点 P を取り出して $DONE$ に入れる.
5. **for** 節点 P につながる各枝 E_p について,
6. **if** (E_p の先の節点 Q が $CLOSED$ じゃない)
7. Q を $CLOSED$ に入れる.
8. **for** 節点 Q につながる各枝 E_q について,
9. **if** (E_q の先の節点 R が $OPEN$ か $DONE$ にある)
10. E_p と E_q でユークリッド距離の長い方を消す.
11. **else if** (R が $CLOSED$ にある)
12. E_q を消す.
13. $CLOSED$ の各節点を $OPEN$ に入れ $CLOSED$ を空にする.
14. **if** ($OPEN$ に P_{init} がある)
15. 探索は成功で求まった通路を表示する.
16. **else** 探索は失敗.

第 4 章

シミュレーション

目標姿勢の形状について調べるためにシミュレーションを行なった。また、前処理と経路探索による動作計画のシミュレーションを行なった。

4.1 シミュレーション 1： 目標姿勢の形状について

第 2.1.2 小節において式 2.3, 式 2.4, 式 2.5, 式 2.6, 式 2.7 で表される同じ手先位置を満たす姿勢の集合が, 与える条件をかえることによりどのように変化するかをシミュレーションにより調べた。ワークステーション上に MATLAB(R) Version 5.2.0.3084¹ により実装し, 表 4.1 に示した条件のもとで実験を行なった。

表 4.1: シミュレーション 1 における条件

条件	ℓ_0	ℓ_1	ℓ_2	L_0
条件 1	1	1	1	2
条件 2	1	1	1	0.5
条件 3	1	2	1	1.5
条件 4	1	2	3	3

4.1.1 結果

それぞれの条件における実験結果を図 4.1 から図 4.8 に示した。

¹(c) Copyright 1984-98 The MathWorks, Inc. All Rights Reserved

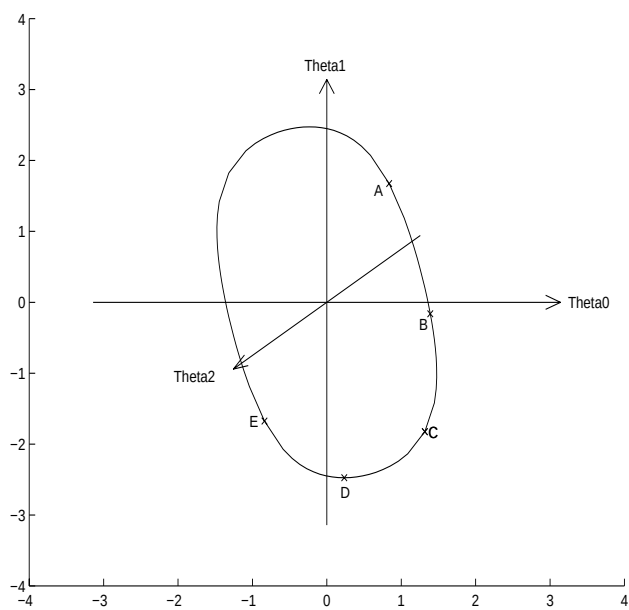


図 4.1: 条件 1 における目標姿勢（関節角座標空間）

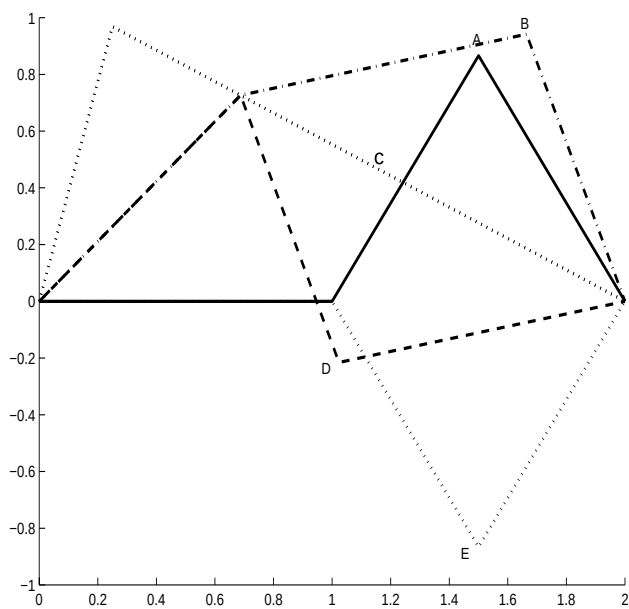


図 4.2: 条件 1 における目標姿勢（作業空間）

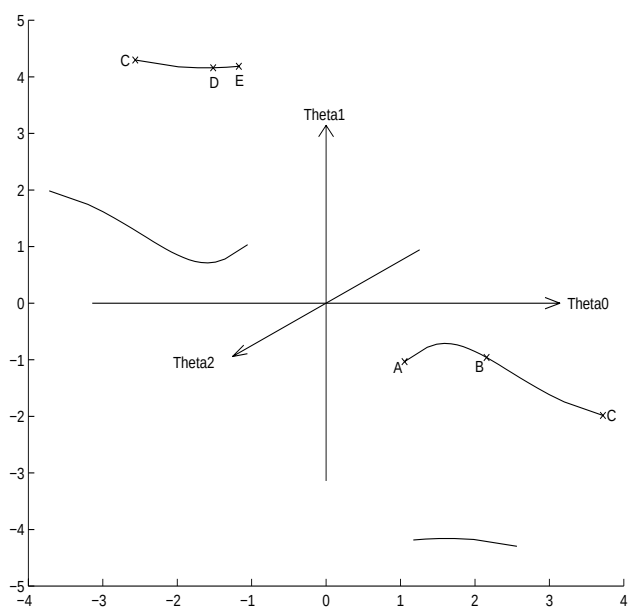


図 4.3: 条件 2 における目標姿勢（関節角座標空間）

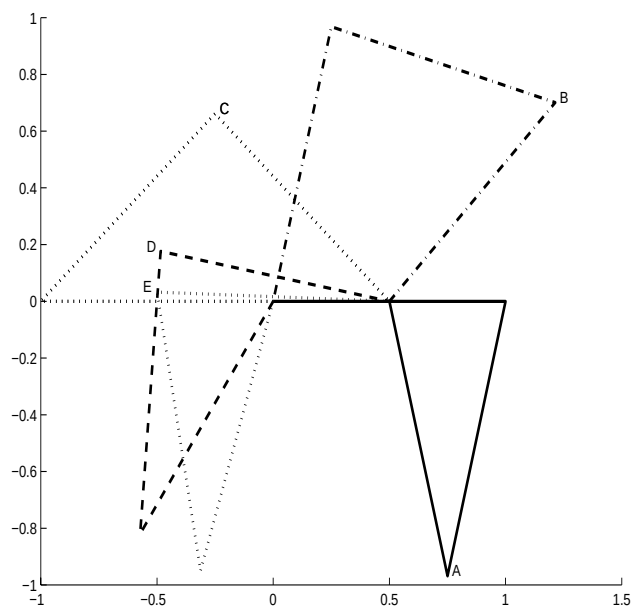


図 4.4: 条件 2 における目標姿勢（作業空間）

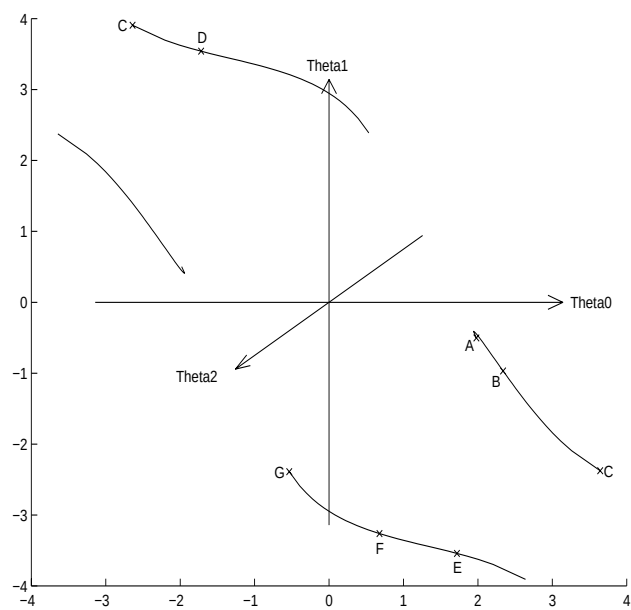


図 4.5: 条件 3 における目標姿勢（関節角座標空間）

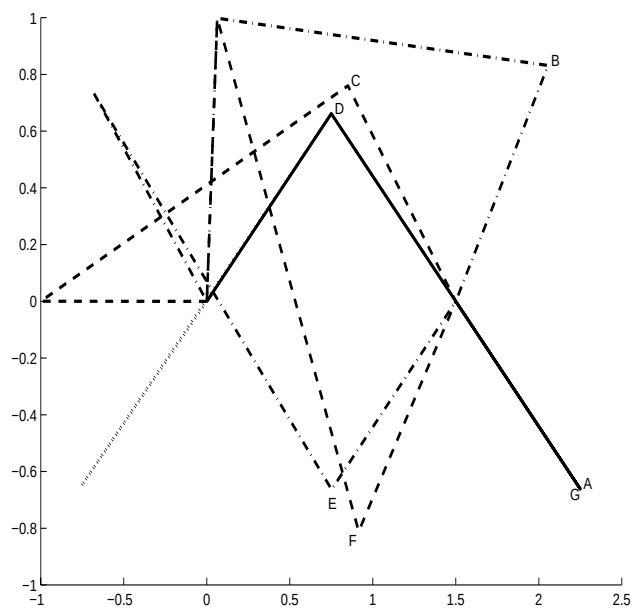


図 4.6: 条件 3 における目標姿勢（作業空間）

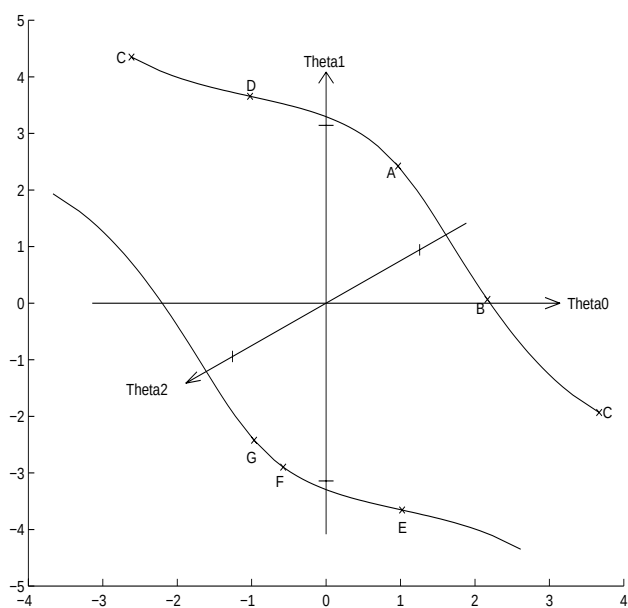


図 4.7: 条件 4 における目標姿勢（関節角座標空間）

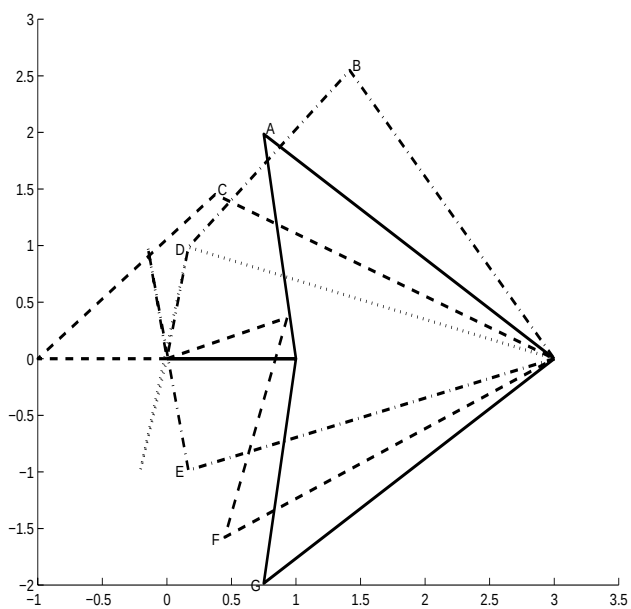


図 4.8: 条件 4 における目標姿勢（作業空間）

関節角座標空間についての説明や図中のアルファベットと作業空間に描かれた関節ロボットとの対応関係および条件 1 の結果については page11 にすでに詳しく説明済みなので、以下には条件 2～4 の結果に対する説明のみ記した。

まず条件 2 の A から順番に見てゆくと、まず徐々に θ_0 の値が大きくなってゆき、B を経由して C の姿勢に至る。C の姿勢では θ_0 が最大値をとっており、このとき L_1 も最大値をとっているが、さらに θ_0 を半時計方向に回すことができる。 θ_0 を最小値と考えると値が徐々に大きくなってゆき、 $\theta_1 < 0$ かつ $\theta_2 < 0$ の D の姿勢を経由して E に至る。このときの同じ手先位置を満たす姿勢は条件 1 と違い、連続した 2 つの集合に分かれてしまう。

つぎに条件 3 では、A から D までは条件 2 とおなじように動くが、ちょうど D において θ_1 が不連続になる。図 4.6 において D の姿勢と上下対称の姿勢が E にあたり、E から F を経由して G の姿勢に至る。ここで、姿勢 A と姿勢 G は重なって見えるが θ_2 が不連続である。つまり条件 3 では、同じ手先位置を満たす姿勢は連続した 4 つの集合に分かれてしまう。

最後に条件 4 でも、A から D まではおなじように動く。ここでまた、図 4.8 において D の姿勢と上下対称の姿勢が E にあたり、E から F を経由して G の姿勢に至ることができる。しかし、条件 3 とは異なり姿勢 G は姿勢 A と上下が対称な姿勢である。つまり図 4.7 において同じ手先位置を満たす姿勢は 2 つに分かれているが、姿勢 D と姿勢 E においてのみ θ_1 が不連続となる。

4.1.2 考察

以上の結果をもとに考えると、条件 1 とその他の条件の違いから L_1 が最大値に至ってからも θ_0 がおなじ方向に動ける条件のときに、同じ手先位置に対する姿勢は 2 つに分かれることが判る。また条件 2 と条件 4 より、 L_1 が最小値に至ってから θ_0 がおなじ方向に動けないときに、同じ手先位置に対する姿勢は 2 つに分かれることも判る。つまり、 $L_0 + \ell_0 \leq \ell_1 + \ell_2$ の時と $|L_0 - \ell_0| \leq |\ell_1 - \ell_2|$ の時にそれぞれ 2 つに分かれる。 L_1 の範囲は $\max(|L_0 - \ell_0|, |\ell_1 - \ell_2|) \leq L_1 \leq \min(L_0 + \ell_0, \ell_1 + \ell_2)$ と表すことができるので、 L_1 の最小値が $|\ell_1 - \ell_2|$ の時と L_1 の最大値が $L_0 + \ell_0$ の時に分けられると書くこともできる。このように同じ手先位置を満たす姿勢が分かれる場合には、保持姿勢におけるポテンシャルを求めるときに最適な姿勢がどの姿勢の集合に含まれるか考慮しなければならない事がわかった。

4.2 シミュレーション2：動作計画

第 3.2.2 小節で説明したアルゴリズム MAKEPOTENTIAL を実際に実装し、動作計画のシミュレーションを行なった。アルゴリズムをワークステーションに実装するにあたり言語は C++ を使用し、ライブラリとして LEDA Version 3.5.1² を利用した。また、ドローネ網の作成には逐次追加法である O.Devillers の配布しているプログラム³ に手を加えたものを用いた。本論文で紹介した動作計画法を実装しシミュレーションを行なうにあたり、実際のプログラムの大まかな流れや用いたデータの構造などをまず説明する。

4.2.1 準備

プログラムは大きく分けて、ドローネ網の作成、前処理によるグラフの作成、経路探索、の 3 つのプロセスからなり立っている。まずはじめに行なわれるのがドローネ網の作成であるが、その際に用いた O.Devillers のプログラムについて説明する。このプログラムは 3 次元の点の座標を入力とし、逐次追加法によりドローネ網を作成する。しかし、帰り値としてグラフを返さずにドローネ網を形成するすべての 4 面体について、4 面体を構成する 4 点の組でかえす。そのために、初期姿勢、同じ手先位置を満たす複数の目標姿勢、ランダムに加える姿勢、から前もってドローネ網を構成するすべての 4 面体の点の組を求めておく。そして前処理として、4 面体の点の組から障害物と交差する辺を除きながらドローネ網をあらわす 2 分木を作成する。この 2 分木には節点に、3 次元の点の座標のほかに、その点と辺でつながった点をあらわす節点へのポインタをリストのかたちで蓄えた。最後に経路探索は、アルゴリズム MAKEPOTENTIAL で行なった。アルゴリズム内でスタックとして用いた *OPEN*, *CLOSED*, *DONE* は、ドローネ網を表す 2 分木の節点へのポインタを 2 分木のかたちで蓄えた。

あと、説明しなければならないのが姿勢をあらわす点の数である。初期姿勢は 1 点であり、同じ手先位置を満たす目標姿勢は 40 点与えた。この 40 という数は経験的に求めたが、目標姿勢の特徴をあらわすために最低限度必要な数である。シミュレーションにおいてランダムに加える姿勢の数は、それぞれ $10^3(1000)$ 個、 $12^3(1728)$ 個、 $14^3(2744)$ 個、 $16^3(4096)$ 個とした。

²©LEDA Software GmbH, Postfach 151101, 66041 Saarbrücken.

³<http://www.inria.fr/prisme/logiciel/del-tree.html>

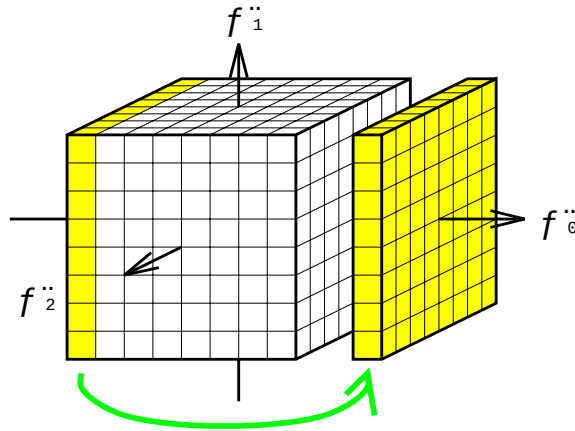


図 4.9: 関節角座標空間の構造

ランダムに与えられた姿勢が関節角座標空間内に均一に散らばっていると考えたとき、関節角座標空間は図 4.9 に示すように量子化されていると考えることができ、このとき図のように一番左の面に分布している姿勢をそのまま右にコピーしドローネ網を形成すれば θ_0 の循環を表現できる。例えば関節角座標空間内にランダムな節点が 10^3 個与えられたときは、 θ_0 の小さい方から 10^2 節点をコピーすることになる。

4.2.2 結果

障害物が存在する空間において、すべての関節の長さが 2 である関節ロボットを用いて初期姿勢 $(\theta_0, \theta_1, \theta_2) = (100, 75, -100)$ ⁴ から、目標手先位置 $(\theta_{goal}, L_{goal}) = (-60, 4)$ に対する動作計画をシミュレートした。 θ_{goal} は関節ロボットの根本を通る水平線からの反時計方向を正とする角度で、 L_{goal} は関節ロボットの根本からの距離をリンクの長さとの比率で表したものである。このとき作業空間に与えられた障害物（点）の位置を表 4.2 に示した。以上の条件にもとづく作業空間の初期状態は図 4.10 の様になっている。

それぞれの節点数における各プロセスの平均実行時間（秒）⁵ を表 4.3 にまとめた。また、ランダムな節点を $10^3(1000)$ 個与えた場合と $16^3(4096)$ 個与えた場合の関節角座標空間と作業空間の様子をそれぞれ図 4.11 と図 4.12、図 4.13 と図 4.14 に示した。

関節角座標空間に描かれた矢印の連続が見つかった通路である。その通路にそって関節角を変化させた軌跡が作業空間には描かれている。

⁴ これらの数値は度で表されているが、実際にはラジアンで計算した。

⁵ 実験をそれぞれ 10 回ずつ行なった平均時間である。

表 4.2: 障害物の座標

障害物	A	B	C	D	E	F	G
角度	-160	-110	-45	10	50	130	170
距離	4	3	2	4	3	5	3

表 4.3: 各プロセスの平均実行時間

節点数	ドローネ網	前処理	経路探索
1000	4.1	26.6	4.2
1728	10.2	46.8	11.9
2744	21.9	81.3	31.1
4096	122.5	154.4	71.5

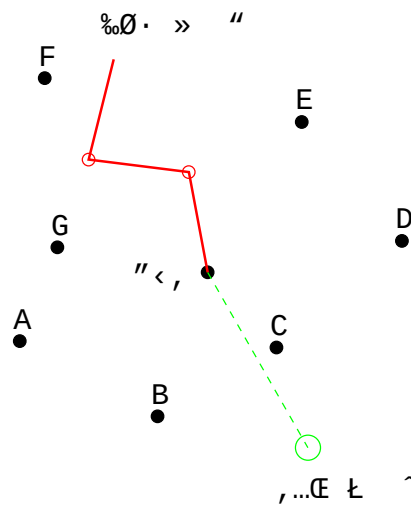


図 4.10: 作業空間における初期状態

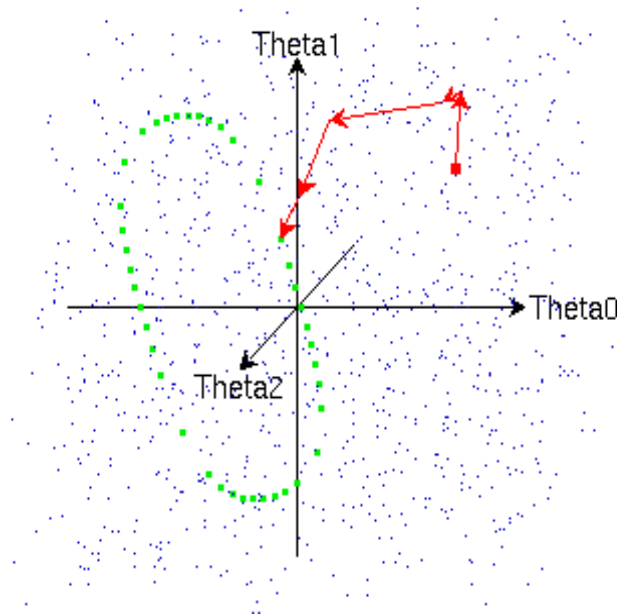


図 4.11: $10^3(1000)$ 節点による実験結果（関節角座標空間）

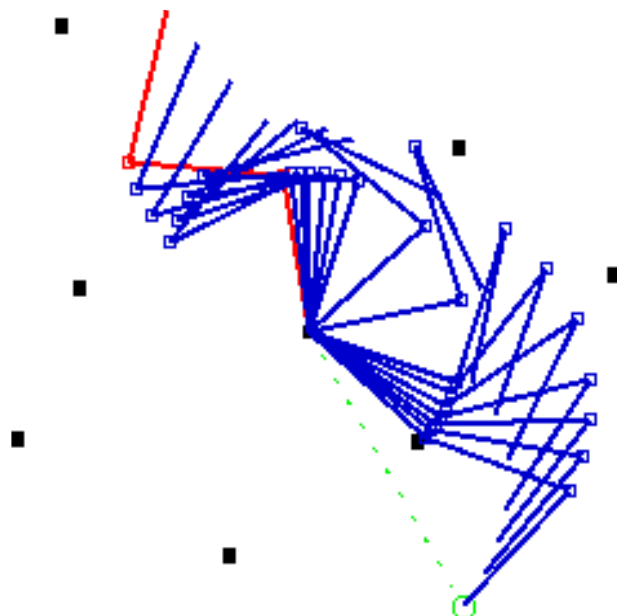


図 4.12: $10^3(1000)$ 節点による実験結果（作業空間）

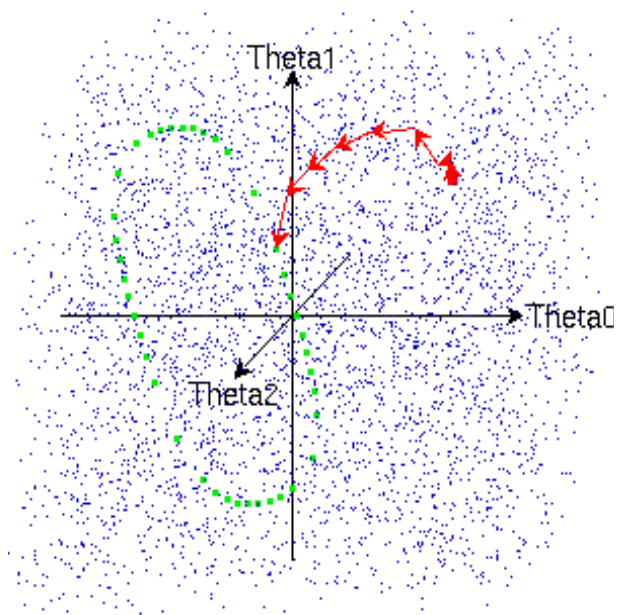


図 4.13: $16^3(4096)$ 節点による実験結果（関節角座標空間）

