

多自由度の移動をする物体の
障害物回避動作計画とその応用

近藤 浩一

多自由度の移動をする物体の
障害物回避動作計画とその応用

近藤 浩一

— 概 要 —

本論文では、一般の多自由度の移動をする物体の障害物回避動作計画問題について、コンフィギュレーション空間において探索の際に参照する範囲を限定する効率の高い方法を提案する。障害物回避動作計画の問題は、一般的には、移動物体の自由度を規定するパラメータによって張られるコンフィギュレーション空間において、初期点から最終点まで点が障害物を回避して移動する問題に帰着される。本論文では、個々の移動経路を探索する際に参照されるのは広大なコンフィギュレーション空間の極く一部である点に着目し、経路探索に関連するコンフィギュレーション空間を限定する方法を用いる。すなわち、初期点ないしは最終点との距離に依存するヒューリスティック関数を用いて、N次元配列状にセル分割されたコンフィギュレーション空間において探索を行なう。この際、ヒューリスティック関数の係数を変えることで定義される異なる方向性を持つ複数の探索戦略を用いて探索を行ない、ヒューリスティックスの平均化を図るとともに、探索の際にすでに参照された領域の情報を用いる評価関数の値により個々の戦略に対する適切な探索努力配分を見積る。また、初期点、最終点付近の空間の状況の差を探索に反映させるために、初期点、最終点両方から出発する両方向探索を用いる。これにより、これまで適用が困難であった多次元コンフィギュレーション空間を用いた障害物回避動作計画を一般的に実行できるようになる。また、本手法は実際に計算機上にインプリメントされており、宇宙用マニピュレータ、原子炉内点検ロボット、プラント設計CADなどの現実の問題に適用し、本論文の提案する手法が幅広い応用範囲を持ち、工業的にも極めて重要であることを確認した。

— 目 次 —

第1章 序論	
1. 1 本研究の目的	… 2
1. 2 高度生産システム実現のための動作計画	… 5
1. 3 極限作業ロボットにおける動作計画	… 11
1. 4 機構設計における動作計画	… 15
1. 5 ブラント設計における動作計画	… 17
1. 6 障害物回避動作計画問題と本論文の提案する方法	… 19
1. 7 本論文の構成	… 22
第2章 障害物回避動作計画	
2. 1 Piano Mover's Problem	… 25
2. 2 コンフィギュレーション空間と自由空間	… 28
2. 3 種々の障害物回避手法	… 31
2. 3. 1 仮説検定法	… 33
2. 3. 2 ポテンシャル法	… 34
2. 3. 3 近似的な自由空間の表現	… 37
2. 3. 4 代数的な自由空間の表現	… 39
2. 3. 5 その他の方法	… 43
2. 3. 6 本論文の提案する方法	… 46
2. 4 形状表現と干渉条件	… 50
2. 4. 1 形状表現手法	… 51
2. 4. 2 干渉条件	… 56
2. 5 干渉以外の拘束条件の表現	… 63
2. 6 第2章のまとめ	… 66
第3章 自由空間算出のための探索戦略	
3. 1 探索による自由空間算出	… 68
3. 2 ヒューリスティックなグラフ探索アルゴリズム	… 71
3. 3 ヒューリスティック関数	… 74

3.4	複数探索戦略の同時実行	… 80
3.5	探索努力の最適配分問題	… 84
3.6	探索努力配分のための評価関数	… 90
3.7	両方向探索に基づく自由空間の算出	… 94
3.8	探索戦略の入れ替え	… 100
3.9	第3章のまとめ	… 101
第4章	自由空間の表現と経路の算出	
4.1	システム構築に必要な技術	… 103
4.2	経路の有無をチェックするための探索戦略	… 104
4.3	自由空間の階層表現	… 115
4.4	階層的な自由空間の算出	… 122
4.5	緩やかな制約条件を考慮した経路探索	… 123
4.6	第4章のまとめ	… 133
第5章	実験による効率の評価	
5.1	例題	… 135
5.2	実験結果	… 149
5.3	考察	… 175
5.3.1	係数 $\alpha(i)$ の影響	… 175
5.3.2	複数探索戦略を用いる効果	… 177
5.3.3	両方向探索の効果	… 181
5.3.4	自由度の影響	… 183
5.3.5	計算時間	… 185
5.4	第5章のまとめ	… 189
第6章	極限作業ロボットにおける障害物回避動作計画	
6.1	宇宙用マニピュレータJEMRMS	… 191
6.1.1	JEMRMSの概要	… 191
6.1.2	JEMRMSにおける障害物回避動作計画	… 200

6.1.3 例題	… 201
6.2 原子炉内下部構造物点検ロボット	… 205
6.2.1 BWR原子炉の概要	… 205
6.2.2 原子炉内下部構造物点検ロボットにおける障害物回避動作計画	… 213
6.2.3 例題	… 216
6.3 第6章のまとめ	… 220
第7章 プラント設計における障害物回避動作計画	
7.1 プラント設計CADにおける物体の移動の表現と推論	… 222
7.2 機器搬出入計画における障害物回避動作計画	… 226
7.3 例題	… 229
7.4 第7章のまとめ	… 239
第8章 結論と展望	
8.1 結論	… 241
8.2 展望	… 244
参考文献	… 246

第1章

序 論

1. 1 本研究の目的

機械系のCAD/CAM/CAE、ロボティクスといった分野では、物体の3次元的な形状や変形、運動などが本質的な問題であり、これらの問題を計算機の上に表現したり、効率的に処理する技術は、広く応用が可能な基礎技術である。

最近では、半導体などの分野でも、複雑な素子の構造を3次元的に表現し、より高精度のシミュレーションが行われるなど、形状処理の応用分野が広がっている。このように、3次元の形状処理が広く実際に使われるようになってきた背景には、計算機の処理能力の向上にともなって、かなりの計算能力を必要とする3次元形状処理も、比較的小規模の計算機システムでも実行可能となってきたことや、コンピュータグラフィックスとこれを実行するグラフィックスハードウェアの進歩によって、手軽に3次元形状を表示し、操作することが可能になったことなどがある。流体解析などの数値シミュレーションの分野でも、計算機能力の向上とともに、3次元解析を行って、その結果をコンピュータグラフィックスを用いて可視化することが比較的一般的に行われるようになってきている。

計算機を用いたシミュレーションが非常に広い分野で頻繁に実行されるようになってきたもう一方の背景には、技術の急激な進歩に伴って、簡単には実験で確認することの困難な現象を利用した高度な機能を実現する必要に迫られたり、非常に短い時間で複雑なシステムを間違えなく最適に設計しなければならなくなっている現実がある。例えば、宇宙用機器のように無重力下で確実に機能する必要があるシステムでは、地球上で機能を確認することが困難であり、計算機上でのシミュレーションなしで設計を行うのは不可能である。また、情報関連機器や自動車のように複雑な機能を限られたコストで実現しなければならない場合には、設計コストをできるだけ圧縮し、設計時間をできるだけ短縮するために、計算機シミュレーションで十分に検討を重ねた上で試作と実験を行うようにしなければ、技術の進歩や価

格競争に対応できない。このように、計算機を用いて設計や生産、システムの運用を効率的に実行する手法は、社会的、技術的な面から近年非常に重要性が増している。

しかし、現状のシミュレーションや形状処理の技術では、このような幅広い分野での高度な計算機の応用の要求に十分に対応できていないのが現状である。一つの理由としては、現在の計算機の処理能力が、現実の大規模な問題を扱うには依然不十分であることが挙げられる。ただ、最近の半導体技術の進歩はめざましく、今後急速に計算機の処理能力が向上することが期待されるので、演算速度の問題はそれほど本質的ではないと思われる。それよりはむしろ、幅広い分野の多様な応用に普遍的に適用できる一般的なソフトウェア的な手法が確立されていないために、個別の問題ごとにソフトウェアを開発しなければならない点に問題がある。3次元形状の表現という意味からは、ソリッドモデルの技術が一般的になりつつあり、共通の手法として普及してきている。しかし、組立品の表現をどうするか、寸法や公差をソリッドモデルとの関連でどのように表現するか、物体の移動や運動をどう扱うかなど、現実の製品やシステムを扱う上で解決しなければならない問題が多く残っている。特に物体の移動や運動の扱いは、運動をともなうシステムの表現やシミュレーションを実行する目的で、静的な形状の表現という意味でのソリッドモデルの技術とは別の観点から非常に重要性が高い問題である。

特に、ロボティクス、機構設計、プラントCADなどの分野では、マニピュレータ、機構部品、機器などの移動方法を計画したり、移動可能性のチェックを行ったりすることが非常に重要になる。比較的単純な場合には、与えられた移動経路にそって物体を移動させたときの周囲との干渉の有無を調べれば十分であるが、より一般的には、干渉があった場合に、干渉しない移動経路を新たに生成することが要求される。しかし、自動的に干渉しない経路を見つけるためには、周囲の障害物との干渉をチェックをしながら移動経路を探索する必要があり、

そのためには、形状表現や干渉チェック、計算機内部での運動の表現法、経路の探索手法などの幅広い技術を統合的に扱わなければならない。さらに、アプリケーションそれぞれの分野で扱う移動物体は多種多様であり、あらゆる対象に普遍的に広く適用できるような手法を開発することは極めて困難である。そのため、実用的には、移動経路の自動生成や障害物を自動的に回避する問題は、ほとんど計算機による自動化がなされていないという現状がある。

本研究では、以上で述べたような観点から、様々な対象やアプリケーションにおいて問題となる動作計画の問題を、普遍的に広く適用できるような形で実行する方法について考える。特に、極限作業ロボットやプラント設計における複雑な動作計画問題を、計算機を用いて解決することを目的とする。そのために、本論文では、物体が周囲の障害物を避けて移動する経路を計算機に自動的に生成させる障害物回避問題を一般的かつ効率よく解くアルゴリズムを提案する。

まず、本章では、動作計画にかかわる技術の必要性について、障害物回避を中心に、高度生産システム、極限作業ロボット、機構設計、プラントCADといった異なる観点、応用分野からさらに詳しく考察する。また、これらの議論の中で、関連する研究分野についても必要に応じて言及し、本論文で扱う障害物回避動作計画問題の位置づけを考える。その上で、本論文が対象とする問題を明確にし、本論文で提案する方法の特徴を述べて、本研究の背景と目的を明確にする。また、最後に本論文の構成について触れる。

1. 2 高度生産システム実現のための動作計画

顧客ニーズが多様化し、製品のライフサイクルが短くなってきている現在では、従来の少品種大量生産から多品種小量生産へ生産体制を移行する必要性に迫られている。その結果、同一の生産ラインで同時に複数製品を生産したり、製品が変わるたびにたびたび生産ラインを組み直すなど、従来の固定的な生産システムでは対応できない生産体制を実現することが必要不可欠となっている。

このような状況に対応するために、多くの生産現場においてFMS化が進められている。さらに、設計段階におけるCADやCAEの進展とともに、設計から生産まで全体を通じた情報管理を行うCIMが推進され、計算機を用いて効率的で柔軟な製品の管理体制が確立されつつある。

このような背景のもと、プログラムの変更によって様々な作業に対応することができる産業用ロボットが生産システムに多数導入され、複雑な作業に適用されるようになってきている。産業用ロボットの柔軟性を十分に引き出し、高い生産効率を実現するためには、生産すべき製品の変更に伴うロボットの作業変更時間はできるだけ短縮されなければならない。しかし、従来の産業用ロボットの動作教示方法はティーチングプレイバック方式であり、個々のロボットごとに実際に動作を実行させて作業を教示する必要があり、ロボットの柔軟性を十分に生かすことが難しかった。このような問題点を解決する方法として期待されているのがオフラインプログラミングである。オフラインプログラミングでは、ロボット言語によるプログラミングやグラフィックシミュレーションによるデバッグによって、実際のロボットを動作させることなく作業を教示することが可能で、生産ラインを止めることなく短時間で作業教示ができる。また、CADとのリンクによって設計段階での製品情報を共有すれば、情報の欠落がなく、またモデル入力を大幅に簡略化することができる。

このようなオフラインプログラミングを実現する際に、最も重要なのが、ロボットの移動経路を自動生成する障害物回避動作計画の機能である。障害物回避動作計画の機能が備わっていれば、個々の作業を実行する途中の移動経路が自動的に算出され、極めて効率的にプログラミングできる。

以下では、ロボット言語という立場から広くオフラインプログラミングに関するサーベイを行い、次に生産システム特有の問題も考慮しながら、障害物回避動作計画の位置づけを考察する。

ロボット言語システムは、作業をどのレベルで指定するかによっていくつかのレベルに分類することができる。例えば[井上84]では以下の5つのレベルに分類されている。

(1) コマンドレベル

ロボットを動かすための基本的な命令で直接動作を記述する単純なもの。アセンブラに相当する。

(2) 原始的動作レベル

ロボットハンドの動きに注目して動作を記述する。プログラマはハンドの位置・姿勢を具体的に指定する。

(3) 構造的動作レベル

Pascalのような高水準言語にロボットの動作記述向きの構文を追加し、言語仕様を拡張したもの。複雑な動作を構造的に記述できる。

(4) 対象物動作レベル

作業を対象物間の状態変化という観点から記述するもの。部品の状態や部品どうしの組み合わせ状態を指定するのでロボット言語処理系には、実際のロボットの動作を生成する問題解決能力が必要となる。

(5) 作業目標レベル

対象物動作レベルをさらに一般化し、作業の最終目標のみを与え、それを達成するための詳細な手順とデータをすべて自動的に生成させようとするもの。

このように、ロボット言語のレベルが高くなればなるほど、計算機に高い問題解決能力が要求される。特に、対象物動作レベルのロボット言語はタスクレベルのロボットプログラミングという呼び方をすることもある。どのようにして問題解決能力を実現するかという観点から、タスクレベルのロボットプログラミングは、人工知能の一研究分野としても活発に研究されてきた [Lozano85]。AUTOPASS [Lieber77] や SHARP [Laugier86] などは、タスクレベルロボットプログラミングを目指したプロトタイプシステムの代表例である。こういったタスクレベルのロボットプログラミングの研究では、

Put Object-A on Object-B

Insert Pin-C into Hole-D

といった対象物の状態に関する記述からロボットの動作を自動的に生成することを目標としている。こういった機能を実現するためには、ロボットがハンドを使って対象物をどのようにつかむか (Grasp planning)、対象物を目標位置まで障害物を避けてどのように移動するか (Free motion planning または Global motion planning)、他の対象物との接触を保ちながらどのように微小な誤差を吸収して組み付け動作を実行するか (Fine motion planning) の大きく3つの問題を解決しなければならない。

対象物の把握をどうするかに関しては、ハンドと対象物の間に干渉がないか、最終的な把握状態に至るまでにハンドが他と干渉しないかなどを考慮した上で、ハンドが安定して対象物を把握できる姿勢を計算する必要がある。ハンドと対象物の干渉に係わる条件は、周囲の障害物との干渉を避ける経路を計画する問題に帰着可能であり、本

論文で議論する手法によって解くことができる。ハンドが安定して対象物を把握できる姿勢を選ぶ問題は、把握に特有の問題であり、対象物の重心とハンドの位置関係を見るだけの簡単な方法から多指ハンドの安定性の判定に至るまで様々な研究がされている [Hanafusa77][Barber86][Holzmann86][Jameson86]。把握の問題に関する研究についてのサーベイとしては、例えば [Lozano89] を参照するのがよい。

組み付け動作計画においては、対象物どうしの接触関係をどのように扱うか、モデルと実物との誤差をどう評価し、それをいかに吸収させるかが主な研究の課題となっている。末広ら [末広88] は、すべりなどの接触運動の表現と制御について、組立への応用という観点から考察している。Lozano-Pérezら [Lozano84] は、制御やモデルの誤差をプランニングで考慮するために、最終組み付け状態から、誤差があっても目標状態に到達できる状態を逆戻りしながら再帰的に求める back projection と呼ばれる方法を提案している。すべりのようなコンプライアント動作によって誤差を吸収できる部分も含めて back projection を実行させることで、力制御を含めた動作の計画を可能にしている。ただし、この手法は穴への軸の挿入など限られた問題に適用されているのみで、自由度が高い場合などにどう適用するかなどについては明らかではない。しかし、誤差を考慮したプランニングの研究は、近年活発になってきており [Laugier89][Jennings89]、今後の発展が期待される。

また、Popplestoneらは、RAPT[Ambler75][Popple78][Popple80] と呼ばれるシステムにおいて、主に組立品の記述の観点からロボット言語の研究を行った。ここでは、物体の面、軸、稜線、頂点といった形状特徴間の関係から物体間の位置関係を推論し、ロボット動作に関連させようとするものである。このシステムでは、物体の形状情報は持っておらず、干渉チェックなどの機能は実現されていないが、物体や形状特徴などの間の関係を 4×4 の変換マトリックスで表現し、物体間の位置関係をこのマトリックスに関連させて一般的に導出しようとする

もので、CADの観点からも興味深い研究である。

現実の生産システムにおいても、ロボット動作をできるだけ高いレベルで記述して動作を教示できれば、教示時間の短縮になるとともに、人間のミスの防止にもなる。ただ、生産ラインの場合にはハンドリングの対象となる部品は限られており、効率を向上させることが第一であるので、人工知能の分野で研究されているタスクレベルロボット言語のすべての機能は必ずしも必要ではない。例えば、生産ラインでは人間の手のように非常に汎用性の高いハンドよりは、それぞれの対象部品を確実に効率よくつかむために専用工具的なハンドを個別に設計して利用することが多い。そのため、部品の把握方法を一般的に計画するモジュールは産業用ロボットの教示には余り必要ではない。

また、部品の組み付け動作についても、複雑な動作を行わせることは少なく、組み付け動作計画も産業用ロボットの教示ではそれほど用いられない機能である。一つの理由としては、部品を滑らすといった他の部品との接触状態を保ったままでの動作を実行するには、力制御と位置制御を適切に組み合わせる必要があり、このような動作が可能なロボットの導入はかなりのコスト高になってしまうことがある。さらに、複雑な動作が必要となる組み付け動作が必要な場合には、その工程がボトルネックとなって生産ラインとしてのスループットが低下するので、より組み立て易い構造に製品自体を設計変更するのが普通である。穴を皿もみしたり、軸にテーパを付ける、位置決め用の突起を付けるといった個別の処置から、部品の組み付け方向を一方方向に揃えるといった対策まで、作り易く設計することは設計から生産まで全体を考えた場合に非常に重要である。また、難しい作業をどうしても行わなければならないときには、専用工具を用いることが多い。穴入れなどによく用いられるRCC(Remote Center Compliance)[Whitney79]はこういった工具の代表例である。

これに対し、周囲の障害物を避けて移動する経路を求める機能は、他の2つに比べて生産

システムにおいても必要性の高い機能である。例えば、ロボットを用いてシートを車のボデーに組み付ける、エンジンを搭載する、といった動作では、ボデーにぶつからずに目標位置までシートやエンジンを移動させる複雑な動作を計画しなければならない。移動はロボットの最も基本的な機能であり、この機能は専用工具や治具で対処できないので、ロボット化するためには多くの手間をかけてでも、このような動作の作成と教示を行っているのが現状である。このような動作が計算機によって自動的に生成できれば、大幅な省力化と信頼性の向上が期待できる。

このような観点から、生産システムにおけるロボットのためのオフラインプログラミングシステムでは、上位のCADシステムとの情報の共有、実際のロボットへの動作データの転送の容易さなどの点で、上流から下流への情報伝達の流れに確実に組み込まれていて、基本的な障害物回避動作計画機能が備わっていれば、ほとんどの要求を満たすことができると考えられる。ただ、上位のCADシステムとの情報の共有という面から考えると、特定の形状表現方法を必要とする障害物回避計画のアルゴリズムでは適用範囲が極めて狭くなるので、どのようなCAD情報であってもそれを利用できる汎用性の高い障害物回避動作計画システムを開発する必要がある。

以上に述べたように、障害物回避動作計画は、研究の歴史を考えると、オフラインプログラミングの分野において非常に重要であった。本論文では、障害物回避の技術を、以下で述べる極限作業ロボット、プラント設計などのさらに広い応用分野に適用して行くことを目的とする。

1. 3 極限作業ロボットにおける動作計画

近年のロボット技術の進展に伴って、ロボットが産業用以外の様々な分野で利用されるようになってきた。極限作業ロボットとして総称される原子力プラントにおける保守点検ロボットや宇宙用マニピュレータなどが典型的な例である。こういったロボットの場合には、産業用ロボットと異なり、作業が多岐にわたり、しかもそれらのほとんどが非定型作業であるので、その場の状況に応じた柔軟な対応をする能力が必要となる。このような意味で、極限作業ロボットにおいては、産業用ロボット以上に高度な問題解決能力が要求される。

このような意味で、極限作業ロボットにおいては、基本的なプランニングの機能としての障害物回避動作計画の機能は必要不可欠である。ここでは、高度な問題解決能力を目指した知能ロボットに関する研究のサーベイを行って、本論文が対象としようとする障害物回避動作計画の意味を考察する。

これまで、極限作業ロボットへの応用を目指して、様々な知能ロボットの研究が行われている[白井87]。より高度な知能ロボットにおいては、視覚、触覚などのセンサを用いて未知の環境を認識して周囲の環境モデルを自動的に生成する必要がある。さらに、この環境モデルの情報に基づいて行動プランを計画し、移動やマニピュレーションを自律的に行って外界に働きかけるという機能が要求される。そのためには、障害物回避動作計画の機能のほか、センシングに関するプランニング、センサ情報の処理、センサ情報にもとづく環境モデルの管理と更新など、環境認識に係わる多くの研究課題を解決しなければならない。

センサに関しては、視覚センサ、近接センサ、触覚センサの扱いから、これら様々なセンサ情報を統合するセンサフュージョン[石川88]の考え方まで、幅広い研究が行われている。また、センサ情報の扱いは、ロボットが持つ環境モデルの生成、更新、管理の問題と密接に関連しており、視覚センサと環境モデルに関する研究(例えば[Iba88][坪内89][浅田90]など)や、触覚

センサと環境モデルに関する研究(例えば[Grimson85]など)を含めて非常に多くの研究がある。ただ、このような問題は、これ自体で非常に困難な研究課題であり、現在多くの研究が様々な方面から行われている分野なので、本論文では扱わない。この方面の研究のサーベイとしては、例えば[Faugeras89]や[Dario89]を参照すればよい。

一方、知能ロボットにおいても、オフラインプログラミングに関連して述べた、対象物の把握方法の計画問題、障害物回避動作計画の問題、組み付け動作のようなコンプライアント動作の計画問題などはセンサ情報処理と同様に極めて重要である。特に、目的位置への移動経路の生成や、マニピュレーションの際のアームの動きの計画など、行動プランの計画段階で障害物回避動作計画の機能は不可欠である。しかし、視覚、触覚などのセンサを用いて未知の環境を認識しながら計画を立てる場合には、あらかじめすべての障害物の情報は得られておらず、また得られた情報にも誤差が含まれていることを考慮しなければならない。このような意味で、周囲の障害物の情報に未知の部分があっても、分っている範囲の情報を用いて、一応移動経路を生成することが可能で、後から障害物の情報が加わった場合には、それに対応して再度移動経路を生成できるようになっていることが望ましい。そのためには、障害物回避動作計画のための経路探索手法を、環境モデル内部でのデータ表現形式などできるだけ独立させ、環境モデルで管理されている範囲の情報のみで柔軟に経路を生成することができるようにしなければならない。経路探索手法が特定の環境データを要求せず、環境モデルと経路探索手法が独立していれば、環境モデルが詳細化されるに伴って、同一の探索手法を用いて必要に応じて経路を生成しなおすことが可能で、全体として柔軟性の高いシステムを実現できる。

また、実際のプラントの保守を行うような、より現実での応用に近い極限作業ロボットにおいても、障害物回避動作計画の機能は非常に重要である。現状のレベルでは、高度な環境

認識能力や問題解決能力を、現実の問題に信頼性高く適用できるように実現することは困難であるので、人間が介在して適切な情報を与えながら仕事をさせる場合が多い。このようなときに、障害物回避動作計画のような基本的なプランニングの機能が、人間のミスの防止と運用の効率向上に大きく貢献する。

例えば、プラントの保守点検ロボットにおいては、決められた時間内に所定の点検作業や保守作業を効率良く実行し、保守点検のためにプラントの運転を停止する時間を少しでも短くする必要がある。そのためには、保守点検作業をあらかじめ計画しておき、実際の点検作業に入ってからの変更やオペレータの介在を最少限にとどめる努力が不可欠である。プラントの保守点検においては、その対象に関する情報がCADデータとして保存されている場合が多く、このデータを利用してオフラインプログラミングを実行するのが最も適切であると考えられる。特に点検箇所や補修箇所間の移動経路は、点検の結果や補修内容に左右されることがほとんどないため、あらかじめプログラミングしておくことが可能であり、それによって効率向上を図ることができる。また、実際の点検作業中にその場の状況に応じた動作を実行する必要が生じた場合でも、障害物回避動作計画の機能を利用することにより、オペレータがロボットが実行すべき動作を簡潔に指定するのみで、効率的に作業を進めることができる。このときに、障害物回避動作計画のための探索戦略が特定のCADデータなどに依存せず、一般的に適用できるものであれば、現場でのオペレータの判断を反映した柔軟な動作計画が実行できる。

宇宙環境においては、難しい作業を自動化したり、クルーの活動を支援する目的で、宇宙用ロボットシステムが期待されている [狼89]。こういった宇宙用ロボットでは、無重力、真空といった厳しい環境下で動作するメカニズムを、打上げ時の効率の面からできるだけ軽量に実現することが要求される。これと同時に、宇宙空間という特殊環境下でのオペレータの負

担を少しでも軽減し、効率向上と安全性の確保を同時に実現することが不可欠となる。例えば、宇宙空間における実験など個々の作業については、オペレータがマスタースレーブモードでアームを操作する場合が多い。このような場合には、安全性確保の立場から、周囲の環境との干渉をチェックするモニタ機能が重要になる。一方、ペイロードの移動など大局的な移動については、オペレータの負担を軽減する目的で、あらかじめ計画された移動経路を自動的に移動させるプログラムモードが用いられる。この移動経路を計画する際には、障害物回避動作計画の機能が不可欠である。また、マスタースレーブモードで作業を実行する場合でも、作業を実行する位置までの移動経路は、安全性確保のためにも、障害物回避動作計画によって自動的に生成されることが望ましい。このように、作業内容や実行方法に対応して、干渉チェック機能のみが必要な場合や、障害物回避の機能まで必要な場合があり、これらに柔軟に対応できるシステムを実現しなければならない。そのためには、干渉チェックの機能と障害物回避動作計画のための探索手法が独立しており、必要なレベルの機能を柔軟に利用できるようなシステムを構築しなければならない。

以上で考えたように、極限作業ロボットのような高い柔軟性が要求されるシステムにおいては、様々なシステムの統合化や多様なレベルでの人間とのインタラクションが必要となる。そのため、障害物回避動作計画の機能にも、あらゆる状況に対応できる柔軟性が要求される。本論文では、このような要求に対応できる障害物回避動作計画手法として、経路探索手法が環境モデルや干渉チェックアルゴリズムに左右されない、一般的な方法の実現について考える。さらに第6章では、宇宙用マニピュレータ、原子炉内点検ロボットへの適用例について述べる。

1.4 機構設計における動作計画

機械設計の中でも、機構の設計においては、各部品間の相対運動や相互におよぼしあう力の影響など、物体の移動とそれに付随する問題を数多く扱わなければならない。

機構の設計では、機構系や制御系の多量のパラメータを効率よく決定する必要があり、そのためには、運動学的特性、静特性、動特性や制御系まで含めた特性の予測技術が重要である[黒岩88]。機構のシミュレーションでは、ジョイントの変位、速度、加速度を与えて、リンクの位置、速度、加速度やジョイントに働く力、モーメントを求める運動解析、逆にジョイントに働く力、モーメントを与えて、それらの変位、速度、加速度等を求める時間応答解析、また、固有振動数、振動モード、伝達関数を求める振動解析、外力等が作用した場合の静的平衡位置を求める静解析等を行う必要がある。

このような解析を実行するための、汎用の3次元機構解析システムとしては、IMP (Integrated Mechanism Program) [Sheth72] や ADAMS (Automatic Dynamic Analysis of Mechanical Systems) [Orlandea77] があげられる。さらに、ADAMSには、一般の機構解析の他に、自動車の走行性や耐久性のシミュレーションを対象としたサブシステム等も用意されている。ただ、このような問題は機構学、振動学、制御などの広い範囲の研究と関連する複雑な問題であるので、本論文ではこれ以上扱わない。

幾何学的な問題で、機構設計に関連する問題としては、干渉チェックがある。干渉チェックは、比較的単純ではあるが、設計者にとって負担が大きく、ミスを起こしがちな設計案のチェック作業を大幅に効率化するために非常に重要な問題である。特に最近では、複雑な機構をできるだけ小さなスペースにおさめることが重要になってきており、設計した機構が他部品との干渉なく所定の運動をすることができるかどうかのチェックや、組み立て可能性のチェックも含めて干渉チェック機能は不可欠である。振動特性などの条件が特別に厳しく、こ

の条件を満足させるために必然的に部品形状が決まってしまう場合などを除けば、一般的に機構設計においては、まず干渉せずに動く機構を設計することが必要である。そのためには、回転やスライドなどの部品間の対偶関係からその機構としての自由度を算出し、機構としての運動をシミュレートして、他部品との干渉をチェックする必要がある。また、組み立て可能性を調べるためには、部品の組み付け位置を最終位置として、組み立て順序に従って障害物回避動作計画を実行すれば、周囲の部品とぶつかって組み立てられない場合を事前チェックできる。

さらに、ロボット設計などにおいては、障害物回避動作計画が重要な設計ツールとして利用される。特に、原子力プラントの点検ロボットのように、対象となる施設がすでに存在し、施設側の設計変更が困難な場合には、所定の機能や運動が行なえるように、リンクの長さなどの形状にかかわる設計パラメータを慎重に決定しなければならない。この際に、個々の設計案について障害物回避動作計画を実行させ、干渉による不具合の有無を調べたり、いくつかの候補の中から作業効率の高いものを選定することが可能となる。また、多機能でありながら使用頻度が低く大量生産できないという観点からも、試作回数の削減という意味で、障害物回避動作計画を設計ツールとして使うことが重要である。特に原子力プラントの点検ロボットのような場合は、現場での動作確認が困難であり、設計段階でのコンピュータシミュレーションによる検討の役割が大きい。原子力プラントにおける点検ロボットのための機構設計ツールとしての有用性については、第6章で事例を通して言及する。

本論文では、このような機構設計においても応用可能な障害物回避動作計画システムとなるように、干渉チェックアルゴリズムや対象製品に独立した汎用性の高い手法を考える。

1. 5 プラント設計における動作計画

原子力発電プラントに代表されるようなプラント設計においては、決められた立地条件のもとで、必要な建屋を配置し、また、それぞれの建屋のなかに、所定の機器を配置し、これらの機器間の配管の引き回しを決定する必要がある。

プラントの建屋の設計や建屋内の機器配置設計などにおいては、機器の搬出入の可能性のチェック、定期点検時などに分解した機器の移動方法や、点検中の部品の一時保管場所（レイダウンスペース）などについて詳細に検討しておく必要がある。複雑なプラントになると、建屋内に設置すべき機器の数も膨大になり、これらの機器の搬入経路や搬入の順序、また故障の際の搬出入可能性のチェックなどの項目を設計者が見落としなく考慮するには、多大な労力が必要とされる。また、定期点検時の機器の移動や一時保管などの点からも、機器の分解手順と外した機器の移動方法や、メンテナンススペースを考慮した一時保管位置などを、プラント設計の段階で詳細に決定する必要がある（レイダウン計画）。また、機器の搬出入が極めて困難な場合には、その機器を分解可能な形に設計変更することが必要であり、プラント設計において搬出入の問題は非常に重要な問題である。このように機器の搬出入計画や、定期点検時の運用法などの観点から、プラント設計においては、機器の移動の問題を頻繁に扱う必要がある。

以上のようなプラント設計における機器の移動問題は、そのまま障害物回避動作計画問題として考えることができる。例えば、機器搬出入の問題を障害物回避動作計画の問題としてとらえ、搬入経路を、機器の設置位置と搬入位置との間の移動経路として自動生成させることができる。設計者はこの結果に基づき、プラントの設計案の妥当性の確認をすることが可能になり、搬入の手順や必要な時間などを容易に見積もることが可能になる。

ただし、プラントにおける機器搬出入においては、クレーン、モノレールなどそれぞれの

場所や対象に応じた移動用の設備が使用される。そのため、機器の移動はこれらの移動用の設備の使用を前提としなければならず、移動設備それぞれに特有な移動方法の制約や能力の問題を柔軟に反映した上で、移動経路が算出可能なシステムでないに対応が困難となる。

また、プラント設計においては、様々な種類の配管を合理的に引き回す作業も重要かつ複雑な仕事である。このような配管の配置や引き回しを決定する作業を配管ルーティングと呼ぶ。原子力プラントの配管ルーティングでは、放射線レベルによる区分、常用系や非常用系などの配管の系統、バルブなどへのアクセス性、万が一の事故の際の安全性などの複雑な条件を考慮して配管の引き回しを決定しなければならない[成川88]。

流体的な問題や熱変形の問題など最終的に考慮しなければならない問題を除けば、配管ルーティングの問題も経路探索問題として考えることができる。すなわち、配管を移動する点ないしは球が掃引してできる移動空間として考え、配管の引き回しを、2点間の点ないしは球の移動経路探索問題として解くことができる。配管ルーティングでは、基本的に3次元空間における点の移動問題に帰着されるので、多自由度マニピュレータの障害物回避動作計画などに比べて自由度が低くなり、自由度の上での問題は少ない。しかし、単に移動経路を求めるのではなく、放射線レベルのような設計条件を考慮した経路を算出しなければならず、こういった付加的な条件が容易に考慮可能な経路探索手法でなければ、配管ルーティングの様な問題まで応用可能な障害物回避動作計画のシステムとはならない。

本論文では、様々な条件を柔軟に組み込むことができるように、干渉チェックのような制約条件のチェック部分と経路探索戦略を独立させる方針のもとに、プラント設計にも応用可能な障害物回避動作計画手法を考える。特に、第7章では、プラント設計における機器の搬入計画に実際に適用した結果について述べる。

1. 6 障害物回避動作計画問題と本論文の提案する手法

以上述べたように、障害物回避動作計画の問題は、生産システム、極限作業ロボット、機構設計、プラント設計CADなど、物体の移動や形状を扱う様々な分野で広く必要とされる一般的で基礎的な問題である。また、障害物回避動作計画の問題は、3次元形状にからむ基本的な問題である干渉チェックに深く係わっており、干渉チェックの機能まで含めて考えると、何らかの形状との係わりのある問題ほとんどに関連している。さらに、干渉チェック機能をどのように実現するかは、どのように3次元形状を計算機のなかに表現するかに強く依存する。このような観点から、障害物回避動作計画に関しては、3次元形状の表現方法、形状間の干渉の有無の判別アルゴリズム、経路探索の方法や戦略など広い範囲にわたる情報処理技術のインプリメント手法や処理効率について考慮する必要がある。

また、ロボットのオフラインプログラミングからプラント設計における機器搬出入問題や配管ルーティングまで、幅広い応用分野を持つ障害物回避動作計画は、基本的な機能という点では共通していても、個々の問題ごとに特有の考慮事項があり、それを容易に反映できる枠組みを用意する必要があることも明らかになった。このような観点から、幅広く応用可能なシステムを実現するには、対象に依存しない一般的なアルゴリズムやデータ構造を用いた障害物回避の手法を開発しなければならない。そのためには、障害物回避動作計画の実現に必要な基本的な3要素、すなわち、3次元形状の表現方法、干渉チェックアルゴリズム、経路探索方法をできるだけ独立にインプリメントする必要がある。これによって問題の性質や対象に最も適した形状の表現方法、干渉チェックアルゴリズム、経路探索方法を用いた効率のよいシステムの構築が可能になり、様々な応用分野に広く適用することができる。

例えば、プラント設計CADのような場合には、計算効率の面で許される範囲で、できるだけ汎用的な形状表現方法や干渉チェックアルゴリズムを用いて、広い範囲の移動問題に対応で

きるようにシステムを構築するのが望ましい。一方、原子力プラント点検ロボットのように、障害物回避動作計画のシステムを装置に組み込み、特定の対象に絞って利用する場合には、その対象に適した近似や簡略化をして効率向上を図ったシステムとするのが良い。

ここでは、このような多様な応用分野に共通する基本的な問題として、障害物回避動作計画問題を改めて以下のように定義する。すなわち、ある整数 N があって、

「 N 自由度を持つ既知の移動物体があり、この移動物体の初期位置・姿勢と最終位置・姿勢が与えられたときに、周囲の既知の静止した障害物との幾何学的な干渉を避けて移動する経路を、自動的に算出する問題。」

とする。形状表現手法や干渉チェックアルゴリズムについては、すでに多くの研究が行われており、これらで用いられた手法を適切に組み合わせれば良い。そこで、本論文では、多様な分野に幅広く応用可能な一般性の高い障害物回避動作計画のための経路探索手法を提案する。本手法では、経路の探索を、コンフィギュレーション空間[Lozano83]と呼ばれる自由度の空間において、コンフィギュレーション空間の情報だけを参照する探索戦略を用いて実行している。このため、対象とする移動物体に関係なく、統一的にコンフィギュレーション空間において探索を実行することが可能になる。すなわち、経路探索アルゴリズムが、形状の表現形式や干渉チェックアルゴリズムと完全に独立しており、どのような問題にでも適用できるだけでなく、それぞれの応用に適した形状表現方法や干渉チェック手法と自由に組み合わせることが可能である。また、アプリケーションに特有の制約条件のチェックも、干渉チェックアルゴリズムに付加することにより容易にインプリメントすることができる。

このようなコンフィギュレーション空間の考え方と、その汎用性・一般性はすでに知られている。しかし、コンフィギュレーション空間において一般的に障害物回避動作計画を実行しようとすると、自由度の増加とともに爆発的に計算複雑さが増加する。そのため、自由度の高

い問題を扱おうとすると、組み合わせの爆発を抑える工夫が必要になる。詳細は障害物回避に関するこれまでの研究のサーベイを通して第2章で述べるが、従来の多くの手法が、応用分野特有の性質を利用したり、対象とする移動物体の構造的な特徴を利用して問題をうまく解くアプローチを用いており、高い次元の問題を扱うために、コンフィギュレーション空間の持つ一般性を犠牲にしているのが現状である。

これに対して本論文では、自由度の高い問題でも一般性を失うことなく効率的に扱うことができるように、探索によって参照すべきコンフィギュレーション空間を限定する方法を新しく提案する。ここで用いる探索戦略は、対象や応用分野とは完全に独立であり、本来の一般性は失われない。その結果、6自由度の移動物体のように次元の高いコンフィギュレーション空間を用いる場合でも、コンフィギュレーション空間の極く限られた情報のみを参照することで、効率よく汎用的に移動経路を算出することが可能になった。

さらに、本手法の効率を向上させるために、探索の際に複数の探索戦略を同時に実行させ、これら複数戦略の制御を探索努力の配分問題として考える新しい探索手法を提案する。そのために、本論文では、A*アルゴリズム [Hart68] に代表されるコストにもとづくヒューリスティックな探索戦略を拡張し、係数の設定によって容易に異なる性質を持つ探索戦略を実現することを可能にし、これらを簡便に制御する方法を新たに導入する。この手法の効果については、本論文の手法を実際にインプリメントし、工学的な観点、実用的な意味から十分複雑な様々な問題に適用することにより、実験的に実証した。

また、コンフィギュレーション空間において自由空間と障害物との境界面のみを限定的に探索する手法も提案し、経路の有無を効率的にチェックする手法を実現した。

本手法は、すでに現実のいくつかの問題に実際に適用されており、工業的な意味からも非常に有効であることが確認されている。

1. 7 本論文の構成

本論文では、以上の議論を踏まえて、3次元形状を扱う広い分野で必要とされる障害物回避動作計画の問題を、特定の対象や応用分野に限定されずに一般的に解くアルゴリズムについて具体的かつ詳細に議論する。

まず、第2章では、障害物回避動作計画を考える上で基礎となる考え方を紹介する。まず、様々な物体の移動問題を統一的に扱うために、コンフィギュレーション空間の概念を改めて説明し、移動物体が障害物と干渉せずに自由に移動できる空間（以下、自由空間と呼ぶ）をどのように定義するかを述べる。この上で、これまでの障害物回避動作計画に関する研究についてのサーベイを行い、本論文の提案する手法と対比させて、従来の問題点や本論文の手法の特長などをさらに明確にする。次に、障害物回避動作計画を考える上で重要な3次元形状の表現方法と形状間の干渉条件について考える。これにより、障害物回避動作計画のシステムを実際にインプリメントする上で、必要な基礎知識を整理する。さらに、移動物体について障害物との干渉以外の制約がある場合の考え方についても触れる。

次に、第3章では、本論文で提案する自由空間算出のための探索戦略について詳細に議論する。すなわち、探索の際すでに参照された自由空間の情報をもとに、経路が存在する確率の高い方向に探索の重点を変更しながら自由空間を算出する探索戦略を提案する。新たに導入したヒューリスティック関数により定義される特定の方向性を持つ探索戦略を複数使い、ヒューリスティクスを平均化するとともに、それぞれの探索戦略に投入する探索努力を制御することで探索の重点を変更する手法を提案する。さらに、初期点、最終点まわりの自由空間の状態の差を探索に反映させるために両方向探索を導入する。

第4章では、障害物回避動作計画を実行するために必要な技術で、第3章で述べなかったものについて議論する。まず、移動経路の有無を効率的にチェックするための探索戦略を提

案する。次に、階層的なデータ構造を用いてコンフィギュレーション空間の情報を効率よく管理する方法について述べる。またこれに関連して、階層的に自由空間を算出する方法を提案する。最後に、干渉以外の緩やかな制約条件を考慮した経路探索と算出された経路の修正について述べる。

第5章では、ここで提案する方法を実際にインプリメントし、複数の例題に適用した結果について報告する。さらに、新しく導入したいいくつかの手法のそれぞれの効果について、実験結果にもとづき考察する。これは、本論文で提案するアルゴリズムの基本部分について、その効率の高さを実験的に実証した結果であり、効率の面から本手法の有効性を確認できる。

第6章以下では、本論文で提案した手法がすでに現実の問題に適用されている事例についての紹介を行う。これを通じて、本論文で提案する手法を用いることによって、現実の問題がどのように解決されるかが具体的に議論される。

第6章では、極限作業ロボットへの適用について述べる。最初の事例は宇宙用マニピュレータであり、このマニピュレータシステムの概要とともに適用結果について報告する。次に、原子力発電プラントの点検ロボットへの適用例を、原子力発電プラントの概要とともに説明する。

第7章では、原子力発電プラント用CADにおける障害物回避動作計画の役割と適用結果について紹介する。

最後に第8章では、それまでの議論をまとめて、本論文の結論と展望を述べる。

第2章

障害物回避動作計画

2. 1 Piano Mover's Problem

第2章では、話題を障害物回避動作計画問題に限って、問題の整理とサーベイを行う。障害物回避動作計画の問題も様々な立場から研究されているので、以下では、障害物回避動作計画の中でも最も基本的な問題に絞って議論する。すなわち、多自由度の任意の移動物体の初期位置・姿勢と最終位置・姿勢が与えられたときに、既知の静止した障害物を避けて移動する移動経路を自動的に算出する問題について考える。このような問題は、Piano Mover's Problem などと呼ばれ古くから研究されており、また、ロボット工学、人工知能、計算幾何学などの立場から、近年でも活発に研究されている [Schwartz88][Brady82]。

障害物回避動作計画においては以下の疑問に対する答えを自動的に求めることが期待される。

- (1) 指定された初期位置・姿勢から最終位置・姿勢へ移動する経路が存在するか?
- (2) 経路が存在する場合には、どのように移動すれば良いか?

このような答えを多自由度の任意の移動物体について、一般的かつ効率よく求めるアルゴリズムを実現するのは、残念ながら極めて難しい。そこで、現実的には、対象とする問題の性質に応じて、様々な方法が用いられる。

簡単な例題として、図2.1の問題を考える [Coste89]。つまり、図に示された楕円が平面内を運動して直角に曲がる通路を通過することができるか、またどのように通過させれば良いかを算出したい。このような計算を実行するには、以下のような問題を解決する必要がある。

- (1) 移動物体の運動を計算機上にどのように表現するか?
- (2) 移動物体と障害物との干渉条件をどう表現し算出するか?
- (3) 移動経路をどのように計算するか?

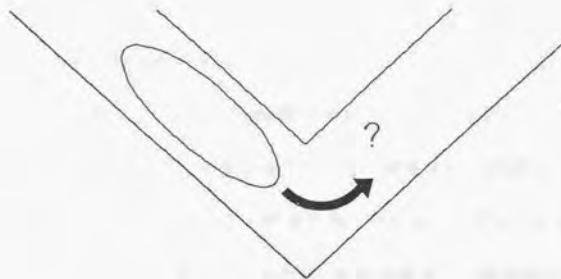


図 2. 1 簡単な障害物回避動作計画の例題

移動物体の運動や移動を計算機にどのように表現するかという観点からこれまでの研究をおおまかに分類すると、3次元実空間（作業空間）を用いるものと、自由度で定義されるコンフィギュレーション空間を用いるものに大別できる。与えられた移動経路にそって周囲との干渉の有無を調べる場合には、3次元実空間は直感的にも理解しやすく便利である。しかし、移動経路を探索する必要のある障害物回避動作計画では、あらゆる移動物体の運動を統一的に扱うことのできるコンフィギュレーション空間に基づく方法が有利である。また、コンフィギュレーション空間を用いることにすると、干渉チェックは実空間からコンフィギュレーション空間への写像をとることに対応し、コンフィギュレーション空間においてどのように経路を探索するかという経路探索手法と別々に考えることが可能になる。その結果、干渉チェックアルゴリズムと経路探索戦略を独立にインプリメントすることができるようになり、全体として汎用性の高いシステムが構築できる。このような観点から本論文では、コンフィギュレーション空間を用いることを前提とした経路探索手法を提案する。

第2章では、本論文で提案する手法の特長を明らかにするために、まずコンフィギュレーション空間についての明確な定義と説明を行い、それを踏まえて障害物回避動作計画に関するこれまでの研究について概観する。さらに、本論文で提案する方法の基本的な考え方についても述べ、本手法の特徴や新規性を整理する。また、形状表現方法や干渉チェックアルゴリズムについても紹介し、障害物回避動作計画システムを実現するために必要な技術をまとめ、第3章以降で経路探索手法に絞った議論をする準備とする。

2.2 コンフィギュレーション空間と自由空間

障害物回避動作計画問題は、障害物と干渉しない領域内での経路探索問題と考えることができる。このような障害物との干渉の無い領域を自由空間と呼ぶ。障害物回避動作計画を行うには、この自由空間を何らかの形で表現する必要がある。これまで様々な手法が提案されているが、自由空間を実空間において表現するか、別の空間において表現するかでおおまかに分類することができる。ここでは、障害物回避の手法を比較する準備としてコンフィギュレーション空間の概念を紹介する。

あらゆる移動物体は、その物体の運動を規定する特定の自由度を持っている。例えば、多関節マニピュレータは、その関節を動かすことによって姿勢を変える。これは関節の数だけの自由度を持っていることになる。3次元空間を自由に動く物体は、それ自身の変形を考えなければ、その位置・姿勢を定める並進3自由度と回転3自由度の合計6自由度を有する。物体の運動を、このような自由度と関連させて考えようとするのがコンフィギュレーション空間である。

N自由度を持つ移動物体の位置・姿勢は、N個のパラメータによって一意に定めることができる。例えば、多関節マニピュレータは関節角をすべて指定すれば、その姿勢が一意に決まる。このような移動物体の位置・姿勢を一意に定めるパラメータによって張られるN次元空間を考えると、この空間での点がパラメータの組を表し、実際の移動物体の位置・姿勢と一対一に対応する。このような考え方はUdupa[Udupa77]がStanford armの障害物回避の研究の中で示唆しており、Lozano-Pérez[Lozano81][Lozano83]は、このような移動物体の位置・姿勢を一意に定めるN次元空間をコンフィギュレーション空間と呼び、このコンフィギュレーション空間における点の移動問題として障害物回避動作計画を考えればよいことを明確にした。

例えば、図2.1の問題では、楕円の中心座標 (x_0, y_0) と長軸の傾き t の3つのパラメータ

で楕円の位置・姿勢を一意に定めることができる。これら3つのパラメータによって張られる3次元空間をコンフィギュレーション空間として考えれば、楕円の運動は3次元コンフィギュレーション空間における点の運動と等価である。このようにコンフィギュレーション空間を用いれば、あらゆる移動物体の移動問題が、N次元空間における点の移動問題として統一的に表現される。

一般的に、マニピュレータなどの移動物体の障害物回避の問題は、障害物と干渉しない領域内での経路探索問題である。これをコンフィギュレーション空間で考えれば、障害物と干渉しない領域内での点の移動問題となる。この障害物と干渉しない領域は自由空間と呼ばれ、障害物回避の問題は自由空間における点の移動経路探索問題として、指定された初期位置・姿勢と最終位置・姿勢に対応するコンフィギュレーション空間における初期点と最終点に関する以下の問題に置き換えられる。

- (1) 初期点と最終点が同一の連結した自由空間に含まれるか？
- (2) 自由空間内で初期点と最終点を結ぶ経路を少なくとも一つ求めよ。

前者は、自由空間の位相的な連結性を調べる問題であり、後者は実際の移動経路を求める問題である。図2.2に2自由度マニピュレータの例で、実空間とコンフィギュレーション空間において定義された自由空間を対比させて示す。

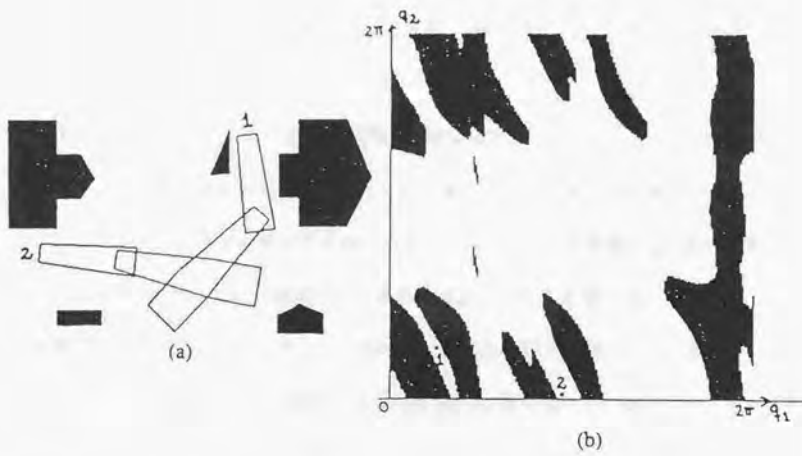


図2. 2 自由度マニピュレータと対応するコンフィギュレーション空間での障害物 [Lozano87]

2.3 種々の障害物回避手法

障害物回避手法は、自由空間を実空間で定義するかコンフィギュレーション空間で定義するかで大別できる。実空間で自由空間を表現しようとするアプローチは、直感的に理解し易いが、対象とする移動物体によっては適用しにくい場合もある。一方、コンフィギュレーション空間を用いる方法は一般的ではあるが、自由度が増えて次元が高くなると必要な計算量が爆発的に増大し、なんらかの工夫をしなければ実用的な意味で解を求めることは不可能になる。そのため、実用的な側面からと、理論的な側面から様々な方法が試みられている。ここでは、自由空間の表現方法や経路の探索方法などに着目して、これまでの研究の方法について紹介し、本論文で提案する方法と比較する。

以下では説明の都合上、これまでの研究を、仮説検定法、ポテンシャル法、近似的な自由空間の表現法、代数的な自由空間の表現法、その他の方法に分類し、本論文の方法と対比させる。ただし、このような分類は絶対的なものではなく、特定の種類にはっきりと色分けしにくいものもあり、それぞれの説明の中で適宜関連性についても述べる。

仮説検定法やポテンシャル法は、比較的初期の研究から試みられている方法であり、コンフィギュレーション空間を用いず、経路探索を実空間の情報を用いて制御しようとする考え方が基本になっている。そのため、問題がコンフィギュレーション空間を用いる場合のように単純な問題へ帰着するのが難しく、対象ごとに個別の方法になりがちである。

一方、近似的な自由空間の表現法や代数的な自由空間の表現法は、コンフィギュレーション空間における自由空間の表現を前提としており、適用範囲の広い一般的な方法である。しかし、一般に障害物回避の問題は、自由度についてPSPACE困難であるとされており、最大計算量は自由度の増大とともに爆発的に増大する。近似的な自由空間の表現法として紹介する方法は、主に対象とする移動物体の機構的な特徴を参照して、コンフィギュレーション空間

の次元を低く抑え、効率化を図るアプローチである。代数的な自由空間の表現法は、理論的、数学的に多次元コンフィギュレーション空間における自由空間を厳密に扱おうとする考え方である。しかし、これらの方法はともに、現実の問題に適用するには自由度の点などで問題が多い。

極く最近、現実的な問題を効率よく解くための方法として、平均計算量の低減という観点から、モンテカルロ法のような確率的な手法を用いるアプローチも提案されている。これらについては、確率的な手法をどのように導入するかによって異なる方法になるので、関連する仮説検定法やポテンシャル法の項で触れることにする。

本論文で提案する方法は、コンフィギュレーション空間を用い、経路探索の際に参照する部分を限定するアプローチを取る。そのため、高い次元のコンフィギュレーション空間であっても扱うことが可能であり、非常に高い一般性を持つ。特にこの方法は、確率的な手法と同様、平均計算量の低減という観点から極めて大きな効果があり、実験的にも確認済みである。

以下では、それぞれの方法について参考文献を参照しながら、さらに説明を加える。

2. 3. 1 仮説検定法

仮説検定法 [Brady82][Lewis73] は、最も初期の研究で試みられた単純な方法である。ここでは、移動経路候補の生成、移動経路を通った場合の障害物との干渉のチェック、干渉を回避するための移動経路の修正の3つのステップを繰り返し実行することによって、移動経路を決定する。この方法は単純ではあるが、自由空間という概念を用いておらず、経路の修正をどのように行うかが明確でない。そのため、経路探索を系統だてて実行できず、障害物が多い場合には、簡単な経路修正方法では経路を見つけることは困難である。

しかし、関連する研究として、コンフィギュレーション空間での探索と、乱数を用いた確率的な仮説の生成とを組み合わせ、効率を改善しようとする考え方が最近提案されている [Glavina90]。この方法では、コンフィギュレーション空間において、障害物にぶつかった場合には障害物に沿って移動するという探索戦略を用いて経路を探索を実行する。しかし、この方法では経路の発見が保証されておらず、経路計画に成功する確率は低い。そこで、もしこの方法で経路を発見できなかった場合には、ランダムに中間ゴールを設定して中間ゴール間の経路探索を実行しながら、問題を分割し、単純化して行く。[Glavina90] には3自由度の例題が紹介されている。

2.3.2 ポテンシャル法

この方法は、探索の制御に仮想的なポテンシャル場を用いる方法である [Brady82][Khatib86][登尾89]。マニピュレータなどの移動物体が移動する実空間において、障害物の内部で値が無限大、障害物から離れるに従って値が減少するポテンシャル場を考え、このポテンシャル場の勾配を反発力として移動物体を移動させようとする方法である。ただし、マニピュレータのようなリンク機構では、各リンクへの反発力が相殺して探索が停留するような場合があり (図2.3)、これを回避するには、ポテンシャル場の情報とは別に、停留点脱出のための戦略や情報が必要となる。

ただし、以上のような問題は、実空間で定義されたポテンシャル場によって探索を制御させることが原因であり、コンフィギュレーション空間で自由空間を求め、コンフィギュレーション空間において初期点から最終点への停留点のないポテンシャル場を定義できれば有効な方法となる [Rimon89]。

ポテンシャル法の特長は、実空間で定義されたポテンシャル場によって探索を制御する方法であり、自由空間全体を直接 N 次元コンフィギュレーション空間で扱わない点にある。このため、自由度の増加による組み合わせの爆発の問題がなく、自由度の高い問題に適用しやすいという長所を持つ。そこで、複雑なリンク機構などでの停留の問題などを回避するために、いくつかの新しい試みも行なわれている。これらは、主に停留点脱出のための戦略や情報に関するものであり、大局的な情報と組み合わせようとするものが多い。比留川ら [比留川87] は、エンドエフェクタの移動経路に関する大域的な情報によって探索を制御することにより、停留の問題を回避しようとする方法を提案している。Barraquand、Latombeら [Barra89][Barra90] は、停留点の間をむすぶグラフを生成するアルゴリズムと組み合わせて、様々な多自由度の障害物回避動作計画問題を解いている。この方法では、停留点に陥った場合

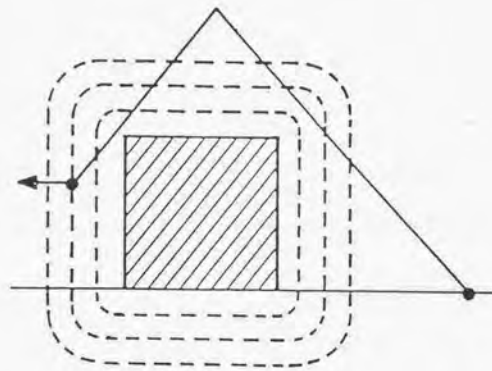


図 2. 3 ポテンシャル法による障害物回避

に、その位置・姿勢に対応するコンフィギュレーション空間の点のまわりで確率的にランダムに運動させ（ブラウン運動させ）、停留点から脱出させる。また、Faverjonら[Faverjon89]は、停留の問題を回避するために、コンフィギュレーション空間において自由空間を陽に表現する方法と組み合わせることを試みている。

2.3.3 近似的な自由空間の表現

ポテンシャル法のように実空間における局所的な情報を用いる場合の、停留のような欠点を克服する方法として、コンフィギュレーション空間において自由空間を大局的に陽に表現するアプローチがある。このようなコンフィギュレーション空間における自由空間を表現するアプローチの一つに、コンフィギュレーション空間を規則的にセル分割し、これらのセルの集合として自由空間を表現する方法がある。例えば、 N 次元コンフィギュレーション空間を N 次元配列状に等しくセル分割し、これらのセルそれぞれについて移動物体と障害物の干渉チェックを実行して自由空間であるかどうかを判定し、自由空間のセルの集合として近似的に自由空間を表現する方法が考えられる。

このように、自由空間がすべて大局的に陽に表現されれば、あとは、自由空間内の経路探索問題に帰着される。例えば、初期点と最終点を含むセルの連結性を調べることで、セル分割の際の量子化の精度の範囲において、経路が存在するかどうかの判定が可能となる。ただ、コンフィギュレーション空間の次元の増加とともに情報量が爆発的に増大するので、次元が高くなるとそのままでの適用が困難になる。一部の範囲に限定して自由空間を算出することになると、経路が存在するかどうかの判定ができるとは限らない。

Lozano-Pérez[Lozano87]は多関節マニピュレータについて、ベースに近いリンクから順番に、リンクの根元を固定した場合に障害物と干渉しない関節角の範囲を求める slice projection と呼ばれる方法を用いて自由空間を算出し、移動経路を生成させた。ただ、この方法も、 N 次元配列状に等しくセル分割する方法に似た方法であり、自由度が多くなると組み合わせの爆発が問題となる。

自由度の増加に伴う組み合わせの爆発を回避する方法として、マニピュレータの先端を無視するなどしてコンフィギュレーション空間の次元を下げる考え方がある

[Lozano87][Hasegawa85][長谷川86]。例えば、手首に3自由度を持つ6自由度のマニピュレータの場合、手首から先の部分がどのように動いてもそれらを必ず含むような球を考え、手首をこの球で近似し手首の自由度を無視すれば、6自由度のマニピュレータの動きが3自由度のコンフィギュレーション空間で表現できる。ただし、どのような点に着目してコンフィギュレーション空間の低次元化を図るかは、対象の機構的な特徴に強く依存してしまい、一般的な手法にはなりにくい。

2.3.4 代数的な自由空間の表現

より厳密に多次元空間における自由空間を扱おうとするアプローチとして、代数的な自由空間の表現を用いようとする考え方がある [Schwartz83][Canny88][Ge89]。

3次元的な複雑な立体の場合には式も複雑になり、見通しが悪くなるので、ここでは、図2.1の例題で考える。ここでは、楕円の中心座標 (x_0, y_0) と長軸の傾き t の3つのパラメータで楕円の位置・姿勢を一意に定めることができ、 (x_0, y_0, t) でコンフィギュレーション空間を定義する。曲り角を通過できるかどうかは、たとえば曲り角の位置を原点に取り、それぞれ左側と右側の位置 $(-1, 1, -1)$ と $(1, 1, 1)$ の間の経路を見つける問題に帰着される。さて、この3次元空間における自由空間は、以下の条件を満たす集合として代数的に定義される。すなわち、

$$\begin{aligned} & \forall x \forall y \{ [(x-x_0) + t(y-y_0)]^2 + 9[(y-y_0) - t(x-x_0)]^2 - (1+t^2) = 0 \\ & \Rightarrow \{ x \geq 0 \text{ AND } (y-x)^2 - \frac{1}{2} < 0 \} \text{ OR } \{ x \leq 0 \text{ AND } (y+x)^2 - \frac{1}{2} < 0 \} \end{aligned} \quad (2-1)$$

となる。つまり、楕円上のすべての点が通路の内部にあるという条件を満たす点の集合 (x_0, y_0, t) が自由空間である。ただ、このままでは x_0, y_0, t の他に変数 x, y が入っており、かつ \forall (全称記号)も含まれているので、障害物回避動作計画に直接的にはつながらない。自由空間の情報を得るためには、 \forall や \exists といった量記号(限定記号)を消去(quantifier elimination)し、コンフィギュレーション空間を定義する変数 x_0, y_0, t に関する式を導出する必要がある。

代数幾何の分野では、多項式によって表される等式、不等式の論理的な組み合わせ(AND, OR, NOT)で表現される条件を満たす n 次元空間の点の集合(semialgebraic set)に関する研究が行われ、幾何学的な様々な問題への適用が試みられている。このような集合(semialgebraic set)の例として身近なものに、CSG(Constructive Solid Geometry)表現による形状

表現法がある。ここでは、C.A.D.(Cylindrical Algebraic Decomposition) と呼ばれる代数幾何の手法の概略を簡単に紹介し、自由空間の代数的表現への適用について触れる。ただし、ここでは手法の概略を直観的に説明することに重点を置き、さらに詳しい説明は参考文献 [Coste89][Marchard89] に譲る。

n 変数 (x_1, \dots, x_n) からなる多項式系 (P_1, \dots, P_r) が与えられ、それに対応して n 次元空間 R^n を考える。このとき R^n 内の点を、各多項式 $P_i (i=1, \dots, r)$ の符号によって分類することを考える。このように、 R^n の部分集合 C においてそれぞれの多項式 P_i が一定の符号 ($=, <, > 0$) を取るとき、 C が (P_1, \dots, P_r) -invariant であるという。さて C.A.D. は、多項式系が与えられたときに、 n 次元空間 R^n を、 (P_1, \dots, P_r) -invariant なセルに分割する方法を与える。

C.A.D. の概略を理解するために、さらに簡単な多項式を考える。

$$P = x^2 + y^2 - 1 \quad (2-2)$$

これは明らかに 2 次元空間における単位円を与える多項式である。 $P > 0$ となるのは円の外部、 $P < 0$ となるのは円の内部、 $P = 0$ は円周上である。つまり (P) -invariant な領域は、直観的には、円周と円の内部、外部の 3 つになると思われる。さて、C.A.D. では、 n 次元空間を順次 $n-1$ 次元空間に射影し、 (x_1, \dots, x_{n-1}) からなる多項式系を生成するという手順を繰り返すことによって、 (P_1, \dots, P_r) -invariant なセルに分割して行く。単位円の例題で、2 次元問題を x 軸へ射影することを考える。直感的には $P = 0$ となる円を投影して、 $x = -1, 1$ の 2 点を境に分類されることが予想される。結果のみを示すと、射影された多項式系は

$$\{x-1, x+1\} \quad (2-3)$$

となり、単位円の両端 $x = -1, 1$ の 2 点で区切られる 5 つのセルに分けられる。つまり $x < -1$ 、 $-1 < x < 1$ 、 $1 < x$ の 1 次元開区間のセル 3 つと、 $x = -1$ 、 $x = 1$ の 0 次元のセル (点) 2 つと

なる。これに基づき、もとの2次元平面は、0次元セル(点)2つ、1次元セル6つ、2次元セル5つの合計13のセルに分割される(図2.4)。この場合、 $P > 0$ の部分は、 $x = -1$ および $y = 1$ の直線上の4つの1次元セルと4つの2次元セルに分割されている。詳細については割愛するが、ここで重要なことは、多項式系が与えられると、少なくとも理論的には、順次再帰的に射影を実行し、 (P_1, \dots, P_r) -invariantなセルに分割する手順が存在することである。

このように、代数的に (P_1, \dots, P_r) -invariantなセルに分割できると、これらのセルを適当に選ぶことにより \forall や \exists といった量記号を消去することができる。たとえば、

$$\forall y \{x^2 + y^2 - 1 > 0\} \quad (2-4)$$

は、 x, y の順に変数の順序を考えることにより、すでに述べたようにセル分解され、 xy 平面の上のセルが x 軸のどのセルに投影されるかという観点から分類できる。これから $x < -1$ 、 $-1 < x$ が条件を満たす x の範囲であることが分る。同様に、図2.1の楕円の例題でも x_0, y_0, t, x, y の順に変数の順序を考えることにより、自由空間を表す x_0, y_0, t からなる3次元空間の部分集合が、少なくとも理論的には算出できる。

このように代数的に分解されたセルについては、それらの間での位相的な接続関係の検出方法についても研究されており、初期点を最終点が連結した自由空間に含まれるかどうかも決定することができる。

しかし、このような代数的な方法はまだ理論的研究が中心であり、現実の大規模な問題や次元の高い問題を計算機で実際に効率よく扱うまでには至っていない。

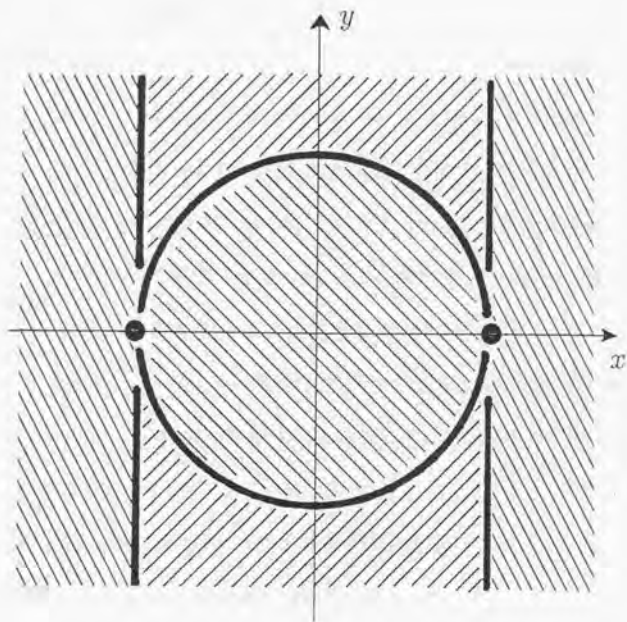


図 2. 4 $P = x^2 + y^2 - 1$ の C.A.D. によるセル分割

2.3.5 その他の方法

以上で紹介した方法以外にも、いくつかの障害物回避のための手法が提案されている。

コンフィギュレーション空間に基づく方法では、近似的な手法と代数的な手法の中間的な手法も提案されている。3次元空間を並進3自由度、回転3自由度の合計6自由度の剛体運動する移動物体についての障害物回避問題を扱ったDonald[Donald87]の方法はその例である。この方法では、多面体どうしの干渉に関する条件を、面と頂点、稜線と稜線の関係として個々に式で表現する。一方、6次元コンフィギュレーション空間に6次元配列状に格子点を定義し、移動経路をこの格子点列として扱う。探索を行う際には、注目している格子点の近傍の自由空間と障害物の境界面(C-Surface)を式から求め、この局所的な情報にもとづくヒューリスティックスを用いて探索を制御する。ここでは、移動物体と障害物との干渉条件を代数的な式として表現しているが、C.A.D.のような操作は行っていない。この方法は、干渉条件を式の形で持つことでデータの爆発を抑え、6次元配列状に定義した格子点の局所的な情報を参照させることで、組み合わせの爆発を回避し6次元コンフィギュレーション空間での探索を可能にしていると考えられる。

Brooks[Brooks84]は、3次元実空間において移動可能な領域を、一般化円筒の集合として表現し、この情報を用いてPUMAの障害物回避動作計画を行わせた。この方法は、一般化円筒の形で移動可能な領域を表現することで、データ量を削減し、簡単な表現形式を採用することで、大局的な形状に関する特徴を抽出し易くしたものだと思われる。実際にPUMAの経路を求めるときには、根元の2関節で張られるコンフィギュレーション空間の情報と併用しており、自由空間の表現方法を工夫して6次元の問題を効率的に解いた例である。

Lumelsky[LumeStep87]は、2次元空間において壁をなぞりながら目的位置まで移動するような探索戦略(図2.5)を用いて、障害物回避を実行する方法を提案している。様々なタ

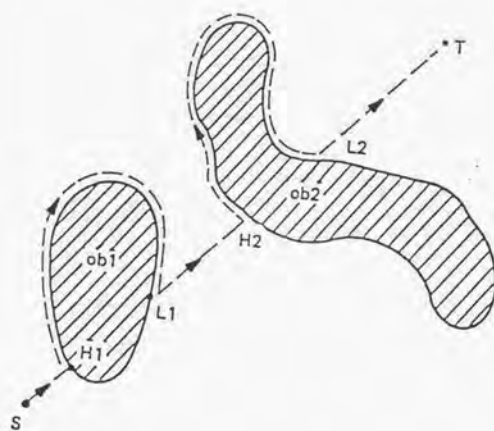


図 2. 5 Lumelsky らの探索方法 [LumeStep87]

イブのマニピュレータについて、実空間とコンフィギュレーション空間がどのように対応するかを調べた上で[Lumelsky87]、この手法の適用について検討がされている。この探索戦略は、障害物の情報が不完全であっても適用可能であり、近接センサを持つ多関節マニピュレータのセンサ情報にもとづく障害物回避に応用されている[Lumelsky88]。

2. 3. 6 本論文の提案する方法

本論文では、 N 次元コンフィギュレーション空間を N 次元配列状に規則的にセル分割し、それぞれのセルについて、対応する移動物体の実空間での位置・姿勢と周囲の障害物との干渉をチェックし、自由空間を自由空間のセルの集合として表現する考え方を取ることにする [Kondo88][Kondo89][Kondo91a][Kondo91b][近藤87][近藤88][近藤89]。この方法は、データ構造が規則的で単純であり、それぞれのセルについてそこで干渉があるかどうかをチェックできればよく、形状の表現方法や干渉チェックアルゴリズムと無関係にインプリメントすることが可能になる。また、移動物体の種類にも左右されないため、3次元空間を6自由度の剛体運動する立体であろうと、6自由度多関節形マニピュレータであろうとまったく同じように扱うことができる。実際のアプリケーションを考えると、序論でも議論したように対象や適用方法に応じて、最適な形状の表現方法や干渉チェックアルゴリズムを選択する必要があるので、自由空間の表現方法や算出アルゴリズムとが分離できる方が望ましいと考えられ、本手法はこれに容易に対応することができる。この点は、コンフィギュレーション空間における自由空間を代数的に表現しようとした場合に、形状の表現方法や変数の取り方が処理対象となる多項式に直接影響し、形状表現や干渉条件の表現が経路探索と密接に係わっていると対照的である。

ただし、6次元空間のような多次元コンフィギュレーション空間の場合、そのまま N 次元配列にセル分割しようとするデータ量が爆発し、現実的な計算時間で各セルについて干渉チェックを実行するのが不可能になる。ただ、 N 次元コンフィギュレーション空間は非常に広大な空間であるものの、個々の経路を探索する際に参照する必要がある領域はその極く一部である。そこでここでは、 N 次元コンフィギュレーション空間において、探索を実行しながら経路探索に関連する自由空間のみを算出する方法を採用した。すなわち、すべてのセルが自

自由空間であるかどうか不明な状態から探索を行い、探索の際に参照された部分についてのみ干渉チェックを実行して、自由空間を順次数え上げるように算出する方針をとった。図2.6に2次元コンフィギュレーション空間の例を、図2.7に指定された初期点と最終点の間の自由空間のセルを探索によって求めた結果を示す。

このように、部分的に自由空間を算出する方法は、[Dupont90]でも提案されている。しかし、この論文では、探索戦略が対象とする多関節マニピュレータの機構的な特徴を直接利用しており、本論文で提案する方法のような汎用性はない。

本論文で提案する方法では、指定された初期点と最終点を結ぶ自由空間のセルを、汎用性を犠牲にせず、いかに効率よく限定して算出するかがポイントとなる。自由空間の算出のための探索戦略や探索の制御方法については、第3章で詳しく議論するが、概略を述べると以下ようになる。すなわち、探索の際すでに参照された自由空間の情報をもとに、経路が存在する確率の高い方向に探索の重点を変更しながら自由空間を算出する手法を実現する。そのために、新たに導入したヒューリスティック関数により定義される特定の方向性を持つ探索戦略を複数使い、ヒューリスティクスを平均化するとともに、それぞれの探索戦略に投入する探索努力を制御することで探索の重点を変更する手法を用いる。さらに、初期点、最終点まわりの自由空間の状態の差を探索に反映させるために両方向探索を導入する。

これらにより、アプリケーションとの独立性を確保しつつ、極めて狭い範囲に探索を限定することが可能になり、汎用性、一般性と計算効率の高さを同時に実現した。

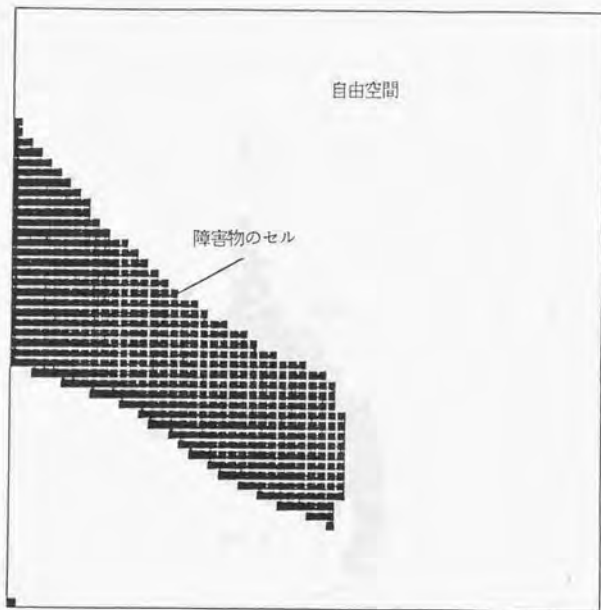


図 2. 6 2次元コンフィギュレーション空間の例

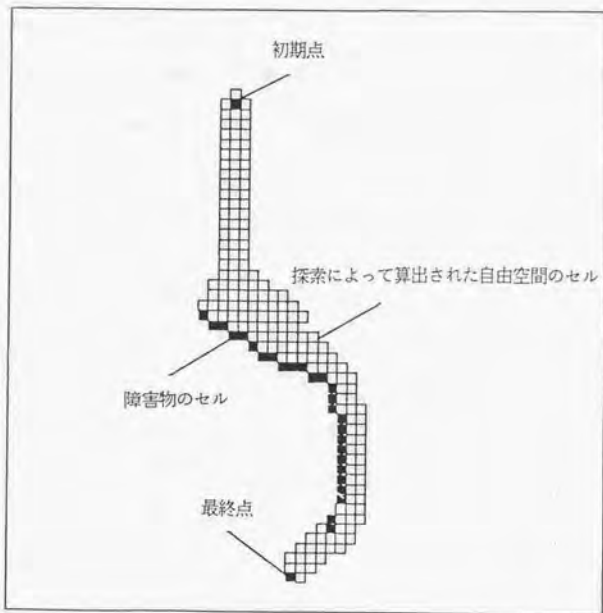


図 2. 7 探索によって求めた自由空間

2.4 形状表現と干渉条件

ここでは、3次元形状の表現方法や3次元形状どうしの干渉チェックの方法などについて考える。すでに述べたように本論文で提案する方法は、経路探索手法と形状表現や干渉チェックアルゴリズムを独立させることができるため、後で詳細に議論する自由空間算出のための探索手法とは直接の関連はない。しかし、これまでに研究されてきた障害物回避の手法の多くが、形状表現や干渉チェックアルゴリズムと密接に関連しており、このような点に関して議論しておくのは、本論文の手法との対比をするのに有効であると思われる。また、本論文の手法を用いる場合にも、以下に述べる干渉チェックアルゴリズムの中から対象に応じた方法を選択し、インプリメントする必要がある。

2. 4. 1 形状表現手法

3次元形状を計算機に表現する技術としては、ソリッドモデルが広く知られている。ソリッドモデルに関するサーベイとしては[Requicha80][Mantyla88]などがある。ソリッドモデルは大きく分けて境界表現(Boundary representation, B-rep)とCSG(Constructive Solid Geometry)の二つに分類できる。図2. 8に境界表現とCSGのそれぞれの表現方法を図式的に示した。

境界表現に基づくソリッドモデルでは、頂点、稜線、面といった形状要素とその位相的な接続関係から立体の境界を表現する。それぞれの面は向き付けされており、面に対してどちら側が立体の内部であるかが判断できる。このように境界表現のソリッドモデルでは、点の座標や面の方程式といった形状要素それぞれの幾何的な形状と、形状要素の位相関係を表現する部分とに大別され、形状要素間の位相関係をどのようなデータ構造を用いて表現するかでさらに分類される。位相関係を表現するためのデータ構造として代表的なのはwinged edgeやhalf edgeなどである。また最近では、従来のソリッドモデルでは扱えなかったワイヤフレーム、サーフェスを含んだ表現を可能にする非多様体形状モデルが提案されている。これは、境界表現の拡張であり、データ構造としてはradial edge [Weller86]などがある。

境界表現のソリッドモデルの場合、稜線、面にどのような幾何形状表現能力があるかで、ソリッドモデルとしての形状表現能力が決まる。例えば、稜線、面が直線と平面に限定されていると、ソリッドモデルは多面体しか扱うことができない。一般的なソリッドモデルでは、直線と平面の他に、円、楕円、双曲線、放物線、円筒面、円錐面、球面などの表現が可能であり、さらに自由曲線、自由曲面が含まれる場合もある。ただし、障害物回避を行う場合には、特殊な場合を除いて、干渉チェックアルゴリズムを必要以上に複雑にしないように単純な多面体近似表現が多く用いられる。また、さらに干渉チェック向きの表現として、多面体を凸多面体に限定することも比較的良好に行われる。

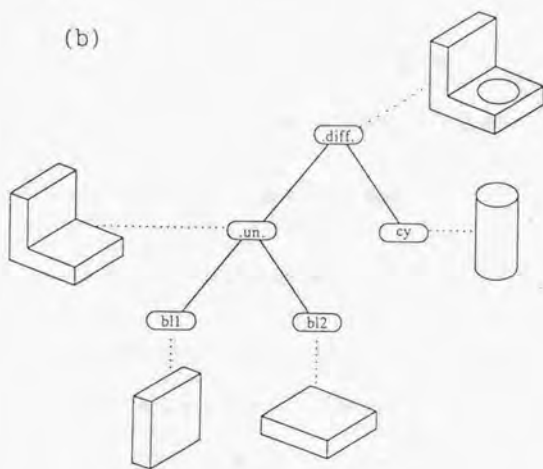
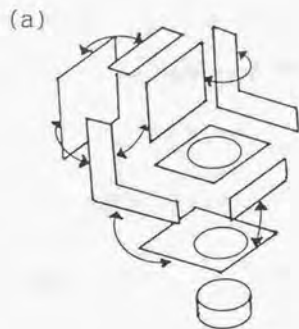


図 2. 8 形状表現方法 ((a)境界表現と(b)CSG)

CSG 表現のソリッドモデルでは、円筒、球、直方体、円錐などのプリミティブを和、差、積といった論理集合演算で組み合わせ複雑な形状を定義していく。このため、CSG 表現のソリッドモデルでは、境界表現のソリッドモデルとは異なり、プリミティブ間の干渉線など、プリミティブを組み合わせることによって生じる形状要素を陽に記述しない。例えば、一辺の長さが20の立方体は、

$$\{-10 \leq x \text{ AND } x \leq 10 \text{ AND } -10 \leq y \text{ AND } y \leq 10 \text{ AND } -10 \leq z \text{ AND } z \leq 10\} \quad (2-5)$$

と表され、半径5の円筒は、

$$\{x^2 + y^2 \leq 25\} \quad (2-6)$$

となる。これらの立体の差を考えると

$$\begin{aligned} & \{-10 \leq x \text{ AND } x \leq 10 \text{ AND } -10 \leq y \text{ AND } y \leq 10 \\ & \text{AND } -10 \leq z \text{ AND } z \leq 10 \text{ AND } x^2 + y^2 \geq 25\} \end{aligned} \quad (2-7)$$

となり、穴のあいた立方体となる。しかし、例えば穴の輪郭

$$\{x^2 + y^2 = 25 \text{ AND } z = 10\} \quad (2-8)$$

という記述はない。また、CSG 表現は、各プリミティブが多項式とその組み合わせで表され、論理集合演算の結果も同様であるため、すでに紹介した semialgebraic set の一例である。障害物回避などの応用では、円筒、直方体、球などの基本形状どうしでの干渉条件をあらかじめ算出しておき、それらの間での干渉チェックを高速に実行するような方法が取られる場合がある[小沢86]。このような場合は、CSG 的な形状表現方法を取っていると考えることができる。

さらに、障害物回避動作計画を行う場合には、より高速に干渉チェックを実行させる目的で、一つの立体について、簡単な球や円筒などによる近似形状から実際の形状まで複数の表現形式を併用した階層的な形状表現を行う場合がある(図2.9)[Faverjon89][近藤87][Kondo89]。

さらに、CADシステムに障害物回避動作計画を組み込むような場合を想定して、システム全体を考えた場合には、形状入力や形状変更の能力も重要になる。形状の入力や変更を効率よく行うには、形状表現能力だけではなく、寸法値の変更などによって自由に形状を変更できる柔軟性が必要である。例えば、著者自身は、ラフスケッチのような初期形状入力から、最終的なソリッドモデル生成までの形状入力履歴を、非多様体形状モデルの形状操作として管理することを提案した。また、さらにそれら形状操作を幾何拘束に関連させ、これらもとづいて寸法処理機能を実現した形状モデリングシステム[Kondo90]を開発した。このようなシステムでは、幾何拘束の表現や処理の方法と、幾何拘束を形状データとどのように関連付けるかが問題になるが、詳細は本論文の主題からずれる面があるので、ここではこれ以上議論しない。サーベイとしては、例えば[Roller89]などを参照するとよい。

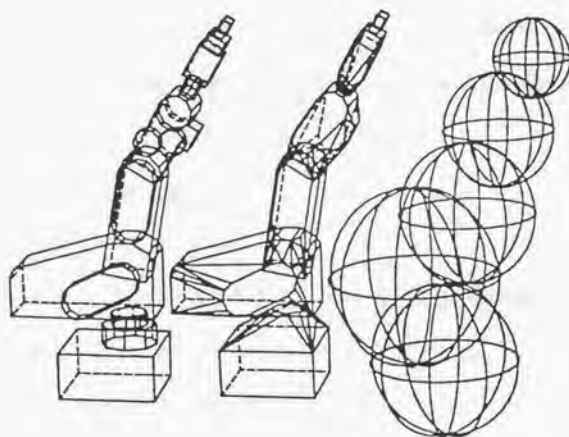


図 2. 9 複数の表現形式（多面体、凸包、球）を用いた階層的な形状表現

2.4.2 干渉条件

移動物体と周囲の障害物との干渉の有無を判定する機能は、障害物回避動作計画を行う上で不可欠である。そのためにはまず、個々の立体どうしの干渉をチェックできる必要がある。移動する物体に関する干渉判定の方法もあるが、ここではまず静止した立体どうしの干渉条件から考える。

ここでは、干渉チェックが比較的高速に実行できる凸多面体どうしの干渉チェックを考える。凸多面体は、平面によって区切られる半空間の積集合として考えることができ、これらの平面の方程式によって凸多面体が記述できる。今、凸多面体は境界表現のソリッドモデルとして表現されており、稜線の両側の面や両端点などは位相情報から検索できるとする。凸多面体の干渉は以下の2通りに分類できる(図2.10)。

(1) 一方の頂点と他方の面との干渉

(2) 両方の稜線どうしでの干渉

頂点と面の関係では、一方の凸多面体の全ての点が、他方のある面に対してその凸多面体と反対側にあれば絶対に干渉しないことを表している(図2.11)。今、凸多面体 P_1 の面を f_1, f_2, \dots, f_m とし、凸多面体 P_2 の頂点を v_1, v_2, \dots, v_n とする。ここで m と n は面の数、頂点の数を表す。ベクトル (a_i, b_i, c_i) が f_i の外側に向かう法線ベクトルとすると f_i は

$$a_i x + b_i y + c_i z + d = 0 \quad (2-9)$$

で表される。 $v_j = (x_j, y_j, z_j)$ であるとし、今、 $f_i \cdot v_j$ を以下のように定義する。

$$f_i \cdot v_j = a_i x_j + b_i y_j + c_i z_j + d \quad (2-10)$$

この表現を用いると、一方の頂点と他方の面との関係で干渉しない条件は、

$$\forall j \exists i \{f_i \cdot v_j > 0\} \quad (2-11)$$

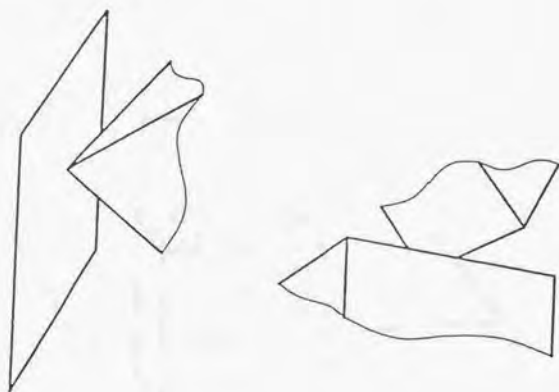


図 2. 1 0 凸多面体が接触する 2 つの場合

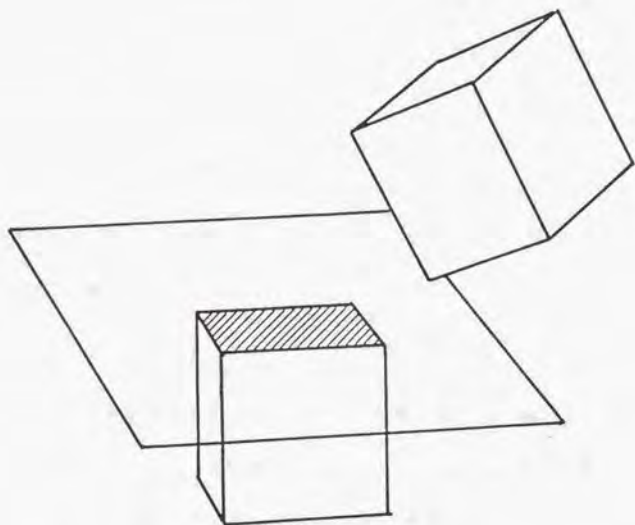


図 2. 1 1 頂点と面の関係で凸多面体が干渉しない場合

となる。この条件は P_1 の頂点と P_2 の面、 P_1 の面と P_2 の頂点の両方について考えなければならない。

次に、両方の稜線どうしでの干渉について考える。ここでは、凸多面体 P_1 の稜線はそれを構成する2つの面 (f_{i1}, f_{i2}) で表し、凸多面体 P_2 の稜線はその端点 (v_{j1}, v_{j2}) で表す。稜線どうしでの干渉を考える場合は、二つの凸多面体の向い合っている稜線のみを考慮すればよい。このためには、 P_2 のすべての頂点が少なくとも f_{i1} または f_{i2} のどちらかの外側になければならない。つまり、

$$\forall j \{ f_{i1} \cdot v_j > 0 \text{ OR } f_{i2} \cdot v_j > 0 \} \quad (2-12)$$

を満たす稜線 (f_{i1}, f_{i2}) のみを考えればよい。また、 v_{j1} と v_{j2} の両方が f_{i1} または f_{i2} の外側にある場合は、明らかに干渉しないので、

$$\begin{cases} f_{i1} \cdot v_{j1} \times f_{i1} \cdot v_{j2} < 0 \\ f_{i2} \cdot v_{j1} \times f_{i2} \cdot v_{j2} < 0 \end{cases} \quad (2-13)$$

となる稜線 (v_{j1}, v_{j2}) のみを考えればよい。以下では説明の都合上、 $f_{i1} \cdot v_{j1} > 0$ となるように v_{j1} を選ぶ。そうすると $f_{i1} \cdot v_{j2} < 0$ となるので、式(2-12)から $f_{i2} \cdot v_{j2} > 0$ となる。

以上の条件を満たす稜線 (f_{i1}, f_{i2}) と (v_{j1}, v_{j2}) に対し、稜線どうしが干渉する場合を考える(図2. 12)。図から分るように、

$$|f_{i1} \cdot v_{j1}| : a = |f_{i1} \cdot v_{j2}| : (b+c) \quad (2-14)$$

ゆえに、

$$\frac{|f_{i1} \cdot v_{j1}|}{|f_{i1} \cdot v_{j2}|} = \frac{a}{b+c} \quad (2-15)$$

同様にして、

$$|f_{i2} \cdot v_{j2}| : c = |f_{i2} \cdot v_{j1}| : (a+b) \quad (2-16)$$

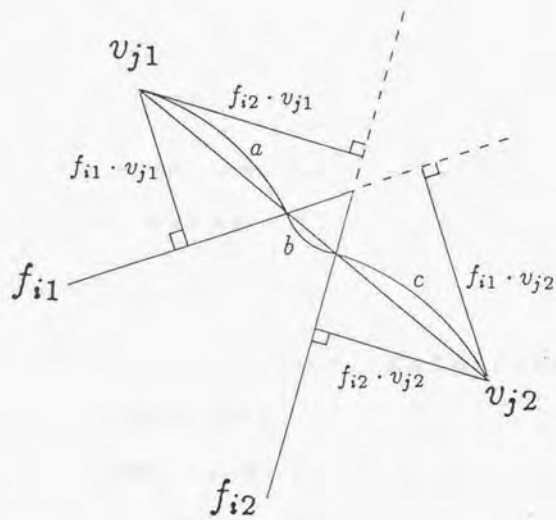


図 2. 1 2 稜線どうしでの凸多面体の干渉

$$\frac{|f_{i2} \cdot v_{j2}|}{|f_{i2} \cdot v_{j1}|} = \frac{c}{a+b} \quad (2-17)$$

従って、

$$\frac{|f_{i1} \cdot v_{j1}|}{|f_{i1} \cdot v_{j2}|} \times \frac{|f_{i2} \cdot v_{j2}|}{|f_{i2} \cdot v_{j1}|} = \frac{a}{b+c} \times \frac{c}{a+b} \leq 1 \quad (2-18)$$

つまり、

$$|f_{i1} \cdot v_{j1}| \times |f_{i2} \cdot v_{j2}| - |f_{i1} \cdot v_{j2}| \times |f_{i2} \cdot v_{j1}| \leq 0 \quad (2-19)$$

ここで、 $f_{i1} \cdot v_{j1} > 0$ 、 $f_{i2} \cdot v_{j2} > 0$ 、 $f_{i1} \cdot v_{j2} < 0$ 、 $f_{i2} \cdot v_{j1} < 0$ であるので、

$$f_{i1} \cdot v_{j1} \times f_{i2} \cdot v_{j2} - f_{i1} \cdot v_{j2} \times f_{i2} \cdot v_{j1} \leq 0 \quad (2-20)$$

となる。つまり、稜線どうしで干渉がある場合には式(2-20)が成立することになり、この対偶をとると干渉しないための以下の条件が導かれる。

$$f_{i1} \cdot v_{j1} \times f_{i2} \cdot v_{j2} - f_{i1} \cdot v_{j2} \times f_{i2} \cdot v_{j1} > 0 \quad (2-21)$$

以上で求めたのは、十分条件であるが、これは必要十分条件であることが証明できる。この証明については、参考文献[湯浅87]にゆずる。

以上で述べたように、凸多面体どうしの場合には、干渉する場合を一方の頂点と他方の面との干渉と両方の稜線どうしでの干渉の2通りに限定でき、簡単な計算で高速に干渉チェックを実行することが可能である。

一般の境界表現の形状モデルの場合には、基本的にはすべての面どうしの干渉計算を実行しなければならないので、凸多面体どうしの場合に比べてかなり多くの計算が必要となる。そこで、周囲の静止した障害物については、あらかじめオクトリーのデータ構造に変換しておき、移動物体のみを多面体の形状モデルとして扱うことにより、処理の高速化を実現する方法もある[登尾87]。また、移動する物体の近似形状表現方法として、大きさの異なる

球をオクツリーのように階層的に構造化して、回転を含む移動を行っても高速に干渉チェックを実行できるようにする方法がある[劉89]。

干渉チェックアルゴリズムには、静止した形状どうしに関するものだけではなく、移動を積極的に扱うものもある。指定された移動経路に沿って物体が移動する場合に、干渉の有無を判定し、干渉がある場合にはどこでどの部分が干渉するのかを特定する問題は、ロボットプログラミングなどの分野で重要である。嶋田[嶋田88]は、凸多面体について、干渉する場合が一方の頂点と他方の面との干渉と両方の稜線どうしでの干渉の2通りに限定できる点に着目し、移動物体の頂点や稜線の動きを求めて、移動中の干渉を厳密に計算する方法を提案している。また、ほぼ同じ考え方が[Canny88]にも述べられている。

2. 5 干渉以外の拘束条件の表現

アプリケーションによっては、障害物回避動作計画を実行する際に、障害物との干渉以外の拘束条件を考慮する必要がでてくる場合がある。例えば、マニピュレータが水の入ったコップを指定された位置まで移動させる問題を考えよう。このとき、マニピュレータは周囲の障害物を回避するのはもちろん、コップの姿勢を保って中の水をこぼさないようにしなければならない。中の水の量に従って、垂直方向とコップの軸のなす角の最大値が決まり、これが移動経路を計画する際の拘束条件となる。次に、丸太のような大きな物体を2つのマニピュレータで運ぶ問題を考える(図2.13)。この場合は、個々のマニピュレータが障害物を回避する外に2つのマニピュレータが同時にワークを把持しているという条件が加わる。機構学的に考えると、2つのマニピュレータとワークが全体として閉ループ機構を構成しており、閉ループ機構として成立つ条件になる。例えば、一方のマニピュレータの関節角でコンフィギュレーション空間を定義するとすると、他方のマニピュレータの手先の位置・姿勢について機構学的な逆問題の解が存在するための条件を求める必要がある。

コンフィギュレーション空間において自由空間を代数的に表現する場合には、このような干渉以外の拘束条件についても代数的に表現し、全体として自由空間を算出すればよい。ただ、代数的に表現することが不可能な拘束条件を考慮することが必要になった場合には、対応が難しい。

本論文で提案する方法のようにコンフィギュレーション空間をセル分割して自由空間を求めようとする場合には、各セルに対応する位置・姿勢について干渉をチェックすると同時に、考慮すべき干渉以外の拘束条件についてもチェックするようにすればよい。基本的には、経路探索の際に自由空間であるのか、禁止領域であるかが明確であればよいので、干渉以外の拘束条件をチェックする機能がインプリメントされていれば問題なく対応できる。

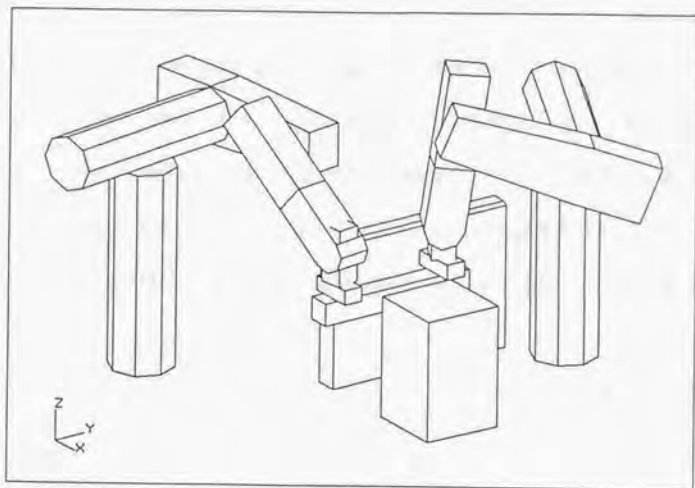


図 2. 1 3 2 アームの協調動作

以上のような拘束条件は、干渉以外の条件ではあるが動作計画に必ず満足されていなければならない条件であった。現実の問題ではさらに、必ずしも満足する必要はないが、満足された方がいい条件を考慮したくなる場合がある。例えば、あまり障害物の近傍を通過したくないといった条件である。このような条件は、自由空間、禁止領域といった明確な形でコンフィギュレーション空間において表現することが難しい。このため、このような条件は、自由空間の表現という観点からではなく、自由空間内部での経路探索において、条件をどのように考慮するか、あるいは自由空間のどの部分を優先的に参照するかという問題になる。このような問題は、問題特有の性質によって、考慮の方法が異なってくる。問題によっては最初の経路探索に拘束条件を反映させることが可能であるし、問題によっては、まず必ず満足する必要のある条件のみで経路を計画し、算出された経路をこのような条件を考慮しながら修正することで対応しなければならない。これに関しては、経路探索手法についての議論と関連して、なるべく障害物から離れた移動経路を算出する問題を例にして、第4章で詳しく考察する。

2.6 第2章のまとめ

第2章では、これまでの障害物回避動作計画の研究についてサーベイするとともに、形状表現や干渉チェックなどの関連技術についての整理をした。

これまでの研究に関しては、仮説検定法、ポテンシャル法、近似的な自由空間の表現法、代数的な自由空間の表現法、その他の方法に分類し、特徴と問題点を整理した。その結果、効率を重視すれば、対象の特徴を利用した専用アルゴリズムになりがちであり、汎用性を重視すれば、自由度の増加にともなう計算量の爆発的な増加が問題となることが分かった。その上で、汎用性を失うことなく現実的な問題を効率よく解くための方法として、平均計算量の低減という観点から、 N 次元コンフィギュレーション空間において、探索を実行しながら経路探索に関連する自由空間のみを算出する方法を提案した。

次に、形状表現技術や干渉チェックについて述べ、障害物回避動作計画システムを実現するために必要な技術について述べた。

また、アプリケーションによっては考慮が必要になる干渉以外の拘束条件についても言及した。

以上の議論を通じて、障害物回避動作計画を考える上での問題点を明確にし、周辺技術も含めて障害物動作計画問題について、技術的な側面から全体像を明らかにした。

第3章

自由空間算出のための探索戦略

3. 1 探索による自由空間算出

第3章では、多次元コンフィギュレーション空間において、経路探索に関連する自由空間を、探索によって限定して算出する方法について議論する。本章の議論は、自由空間をできるだけ狭い範囲に限定するための探索戦略に関するものが中心になる。

すでに触れたように、本論文で提案する方法では、汎用性や一般性を重視して、N次元コンフィギュレーション空間をN次元配列としてセル分割して扱うことにする。この場合、経路探索問題は、初期点のセルから最終点のセルまでを、障害物と干渉しない自由空間の領域に含まれるセルで連結する問題となる。このためにはまず、初期点のセルと最終点のセルを含む連結した自由空間の領域を算出する必要がある。ただ、個々の経路探索を実行する際に参照しなければならない自由空間の領域は、コンフィギュレーション空間の極く一部のはずである。必要な自由空間をなんらかの方法で狭い領域に限定できれば、次元の高いコンフィギュレーション空間でもそのまま扱うことが可能となる。

そこで、N次元コンフィギュレーション空間のセルを以下の3種類に分類する。

(1) 自由空間のセル

移動物体がこのセルに対応する位置・姿勢で障害物と干渉しない。

(2) 障害物のセル

移動物体がこのセルに対応する位置・姿勢でいずれかの障害物と干渉する。

(3) 未知のセル

自由空間のセルか障害物のセルかが不明のセル

さらに、個々の移動経路を探索する際には、広大なコンフィギュレーション空間の内、自由空間のセルと障害物のセルのみをメモリに記憶し、移動経路の探索に無関係な領域を未知のセルとして残しておく。つまり、自由空間の算出が始まる前は全てのセルを未知のセルとし

て残しておき、自由空間であるかどうかはそのセルについて干渉チェックを行わないと分らない状態にしておく。自由空間の算出は、自由空間のセルから順次隣接する未知のセルへ探索を進め、このときに、未知のセルについて干渉チェックを行って自由空間であるかどうかを確認する。このようにグラフ探索手法を利用して自由空間のセルを順次数え上げる方法を用いると、探索に関係しない部分のセルは未知のまま残されるので、広大な多次元コンフィギュレーション空間を用いても、効率よく経路探索を行なえる。

以下では、自由空間を算出するための探索戦略について議論する。この探索戦略が効率の高いものであればあるほど、必要な自由空間を限定して算出することができ、高速に経路を求めることが可能になる。参照する自由空間を狭い範囲に限定できれば、より高い次元のコンフィギュレーション空間を扱うことが可能になり、より汎用性が高くなる。ただし、自由空間算出のために実行される探索では、コンフィギュレーション空間の状態が不明の状態から探索を始めなければならず、探索の進行状態に応じて適切に探索の重点を変更していく必要がある。

第3章では、探索の際すでに参照された自由空間の情報をもとに、与えられた初期点と最終点を結ぶ経路が存在する確率の高い方向に探索の重点を変更しながら自由空間を算出する探索戦略を提案する。すなわち、すでに分かっている自由空間の部分的な情報から必要な自由空間の形状を予測し、探索を制御する。そこでまず、特定の方向を重点的に探索する戦略が、本論文で新しく提案したヒューリスティック関数によって実現できることを示す。さらに複数の探索戦略を同時に実行し、ヒューリスティクスを平均化するとともに、それぞれの探索戦略に投入する探索努力を制御することで探索の重点を変更する手法を提案する。そのために、探索努力の最適配分に関する一般的な理論について簡単に紹介し、その上で、障害物回避動作計画において探索努力配分のための評価法を提案する。さらに、初期点、最終

点まわりの自由空間の状態の差を探索に反映させるために両方向探索を導入する。また、探索戦略の入れ替えによりさらに効率の向上を図る。

3. 2 ヒューリスティックなグラフ探索アルゴリズム

ここでは、以下の議論の準備として、 A^* アルゴリズム [Hart68] に代表される評価関数に基づくグラフ探索アルゴリズムについて説明する。

グラフは節点の集合から構成され、節点の間は向きを持った弧で結ばれている。弧が節点 n_i から節点 n_j へ向かっていれば、節点 n_j を節点 n_i の後継者 (successor) と呼び、節点 n_i を節点 n_j の親 (parent) と呼ぶ。系列 $(n_{i_1}, n_{i_2}, \dots, n_{i_k})$ において、 $j = 2, \dots, k$ に対して n_{i_j} が $n_{i_{j-1}}$ の後継者ならば、この系列は n_{i_1} から n_{i_k} に至る長さ k の道と呼ぶ。節点 n_i から n_j へ道があれば、 n_j は n_i から到達可能 (accessible) であると言う。このとき節点 n_j は節点 n_i の子孫 (descendant) であり、節点 n_i は節点 n_j の祖先 (ancestor) である。

グラフ探索は、出発節点が与えられ、これから目標節点までの道を求める手続きである。このために、グラフ探索では後継者オペレータの概念が用いられる。後継者オペレータが節点に適用されると、その後継者をすべて与えてくれる。この後継者オペレータを節点に適用することを、節点を展開すると言う。グラフ探索の手続きは、出発節点から順次節点を展開して、この操作を目標節点が見つかるまで繰り返すことであると考えられる。このような過程は、以下のような手続きとして表わされる。

- (1) 出発節点 s だけからなる探索グラフ G をつくる。さらに s を OPEN と呼ばれるリストに入れる。
- (2) CLOSED と呼ばれるリストをつくり空にする。
- (3) LOOP: もし OPEN が空ならば探索は失敗し、終了する。
- (4) OPEN の中から節点を 1 つ選び、それを OPEN から取り除き、CLOSED に加える。この節点を n とする。
- (5) もし、 n が目標節点ならば、 n から s までのポイントをたどって解を得る。

(6) 節点 n を展開し、その後継者を M とする。

(7) M に含まれ、OPEN にも CLOSED にも含まれない要素から n へ向かうポイントをつくり、この要素を OPEN に入れる。

(8) GO TO LOOP

探索戦略は、ステップ 4 における節点の選択方法によって様々なものが考えられる。もしこのステップで、探索木の深さ (s からの道の長さ) の一番小さい節点を選択すると、横型探索になり、最後に OPEN に加えられた節点を選択するようにすると縦型探索になる。

横型探索や縦型探索は、目標節点に至る道をしらみつぶしに調べる方法である。しかし、このような方法では膨大な節点を展開することになり現実的でない問題が多い。そこで、対象とする問題に対する情報を利用して、探索によって展開する節点の数を減らす必要が出てくる。このような情報をヒューリスティックな情報と呼び、これに基づく探索をヒューリスティックなグラフ探索と呼ぶ。

ここでは、グラフの弧にそれぞれコストが与えられており、目標節点までのコスト最小の道を求める問題を考える。ヒューリスティックな探索においては、ヒューリスティックな情報を用いることにより探索に要する手間を大幅に削減することができる場合が多いが、中には必ずしもコスト最小の道となるとは限らないものも含まれる。もしあるアルゴリズムが、 s から目標節点に達する道が存在するあらゆる場合において、必ず最適な道を見つけて終了するならば、このアルゴリズムは適格 (admissible) であるという。また、求められた道のコストと探索に必要な手間の組み合わせがより低いものは、ヒューリスティックな力が強いと言う。

さて、節点を選択する段階 (ステップ 4) で、最も見込みの高い節点を見分けるには、何らかの評価関数が必要となる。そこでここでは、評価関数 f を、どの節点 n においてもその値 $f(n)$ が出発節点 s から節点 n までの最小コストの道のコストと節点 n から目標節点までの

最小コストの道のコストの和の推定値になるように定義する。つまり $f(n)$ は、節点 n を通る道の最小コストの予想値である。ここで、出発節点 s から節点 n までの真の最小コストを $g^*(n)$ 、節点 n から目標節点までの真の最小コストを $h^*(n)$ 、節点 n を通る道の真の最小コストを $f^*(n)$ とする。このとき、評価関数 $f(n)$ は $f^*(n)$ の予想値となり、次式で与えられる。

$$f(n) = g(n) + h(n) \quad (3-1)$$

ここで、 $g(n)$ は $g^*(n)$ の推定値、 $h(n)$ は $h^*(n)$ の推定値である。 $h^*(n)$ の推定である $h(n)$ に対しては、問題領域に関するヒューリスティックな情報を頼りにしなければならない。そこで h をヒューリスティック関数と呼ぶ。このような評価関数を用いて節点を選択するアルゴリズムを、アルゴリズム A と呼ぶ。

ここで、 h が h^* の下界 ($\forall n \{h \leq h^*\}$) ならば、アルゴリズム A は目標節点までの最短経路を見つけることを証明することができる [Hart68][Nilsson82]。このようなアルゴリズムを A* アルゴリズムと呼ぶ。A* アルゴリズムは適格である。

3.3 ヒューリスティック関数

ここでは、評価関数にもとづくグラフ探索を障害物回避動作計画における自由空間の算出に用いることを考える。特に、特定の方向を重視して探索する探索戦略を定義するためのヒューリスティック関数を提案する。自由空間を算出する場合、節点はN次元配列状に分割された自由空間のセルとなり、節点の展開は、展開される自由空間のセルに隣接する未知のセルについて干渉チェックを実行し、自由空間のセルであると分かったものを後継者とする操作になる。

例えば関節角空間として表されるN次元コンフィギュレーション空間において、関節角 $(c_1(1), \dots, c_1(N))$ で表される初期姿勢のセル C_1 から、関節角 $(c_f(1), \dots, c_f(N))$ で表される最終姿勢のセル C_f まで、単方向の探索をすることを考える。このとき、セル $C = (c(1), \dots, c(N))$ についての評価関数 $f(C)$ は、次のように定義される。

$$f(C) = g(C) + h(C) \quad (3-2)$$

ここで、 g は C_1 から C_f までのコスト（移動距離）であり、具体的にはセルの数をを用いる。一方、 h は C から C_f までのコストの予想値である。セルを展開する際には、展開候補の自由空間のセル（リストOPENに入っている自由空間のセル）のうちこの評価関数の値が最も小さいものが選ばれる。つまり、コストが少なく、早く最終姿勢のセルに到達しそうなセルが優先されて展開されることになる。もちろんここでも、ヒューリスティック関数 $h(C)$ が探索の効率を左右する。このような手順が、 C_1 と C_f が自由空間のセルで連結されるまで繰り返して実行され、必要な自由空間が算出される。自由空間が算出されれば、移動経路をこの自由空間内部で探索すればよい。もし、探索の途中でリストOPENが空になれば、 C_1 を含む自由空間と C_f を含む自由空間は位相的に連結していないことになり、移動することは不可能であると結論

できる。

さて、障害物回避動作計画では、多次元空間での探索であり、できるだけ探索の範囲を狭くしたいので、アルゴリズムの適格性を犠牲にしてもヒューリスティックな力を重視するとともに、後で議論する探索の制御のために探索戦略に方向性を持たせる目的で、 $h(C)$ を次のように決める。すなわち、

$$\begin{cases} f(C) = g(C) + h(C) = g(C) + A \cdot h'(C) \\ h'(C) = \sqrt{\sum_{i=1}^N (a(i) \times (c(i) - c_f(i))^2)} \end{cases} \quad (3-3)$$

とする。 A 、 $a(i)$ は係数であり、 A はコスト $g(C)$ との割合いを定めるパラメータで、探索の際に参照する範囲(以下では探索範囲と略称することもある)を狭めるために用いられる。なお、このようにヒューリスティック関数を定数倍することにより簡便にヒューリスティックな力を強めることは、一般の様々な問題においてよく行われる。一方、 $a(i)$ は関節に重みづけするパラメータであり、探索の際に重視する方向性を決定する。すべての i について $a(i) = 1$ ならば $h'(C)$ は N 次元空間における直線距離である。係数 $a(i)$ が異なる場合には、関節角が同じだけ目標値に近づいても係数の大きい関節の場合の方が評価関数が大きく減少するため、 $a(i)$ の大きい関節を動かす方向が優先的に展開されるようになり、これによって探索を優先して実行する方向に関して様々な特徴を持つ探索戦略を実現できる。

まずは簡単のために図3.1に示した2次元コンフィギュレーション空間の例で考える。ここで、白い部分が自由空間、黒い部分が障害物と干渉する領域である。今、横軸を1軸、縦軸を2軸として、ヒューリスティック関数 $h(C)$ を以下のように定める。

$$h(C) = A \sqrt{a(1)(c(1) - c_f(1))^2 + a(2)(c(2) - c_f(2))^2} \quad (3-4)$$

まず係数 A の影響を見るために、 $a(1) = a(2) = 1$ として $A = 1$ の場合と $A = 3$ の場合について自由空間を算出させてみた。その結果を図3.2(a)(b)に示す。 $A = 1$ のときは適格である

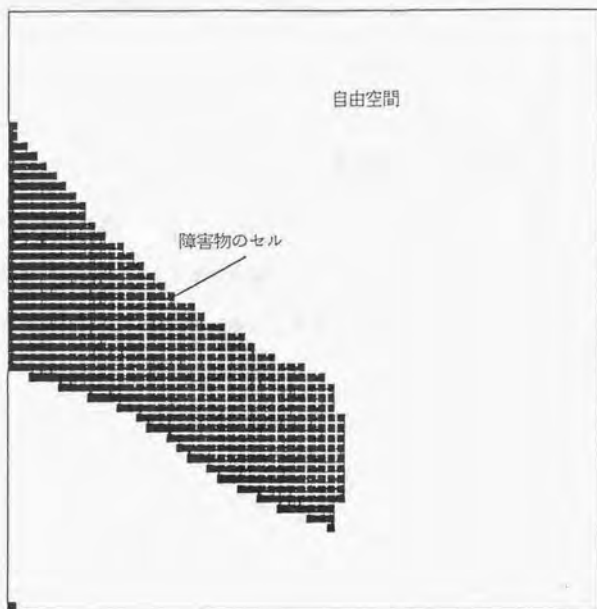


図 3. 1 2次元コンフィギュレーション空間の例

が探索した範囲は広がっている。一方 $A=3$ の場合は適格性は犠牲になっているが、少ないセルの展開で必要な自由空間が求まっている。

次に係数 $a(1), a(2)$ の影響を調べるために $A=3$ として、 $a(1), a(2)$ を変化させてみる。このとき $a(1)=1, a(2)=3$ としたときの自由空間の算出結果を図 3. 3 (b) に、同様に $a(1)=1, a(2)=1$ の場合の結果を図 3. 3 (c) に、また $a(1)=3, a(2)=1$ の場合の結果を図 3. 3 (d) に示す。図から分るように (b) の場合はまず縦軸方向を先に動かす方向に探索が進み、逆に (d) では横軸方向を先に動かす方向に探索が進んでおり、この例題では結果的に (d) が一番効率が良い。すなわち、係数 $a(i)$ によってそれぞれの探索戦略が重視する方向が決定され、結果として探索範囲に違いが生じている。しかし、本方法では、コンフィギュレーション空間がどのような状態であるかという知識を前提としていないため、自由空間を算出しようとする時点では、(d) が効率がよいことは直接的には分らない。

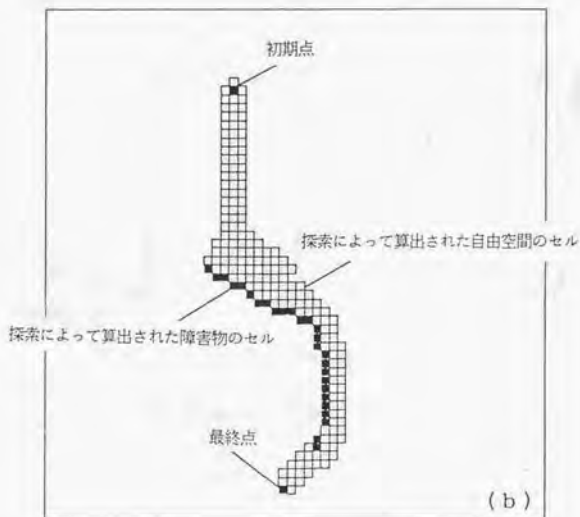
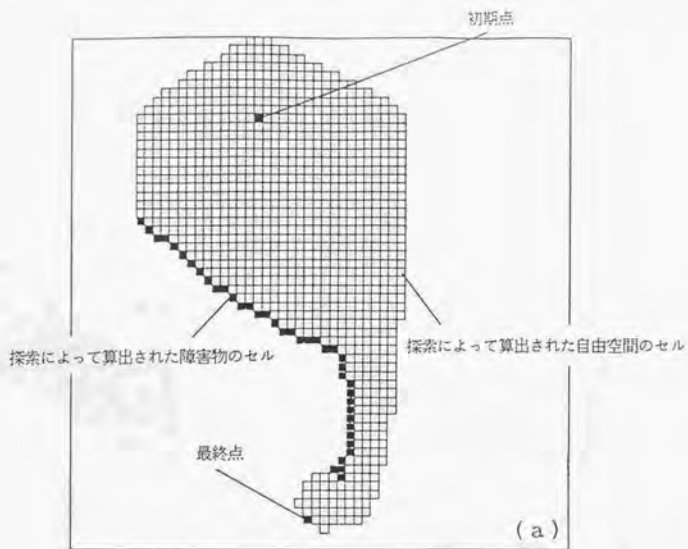


図 3. 2 係数 A の影響 ((a) $A = 1$, (b) $A = 3$)

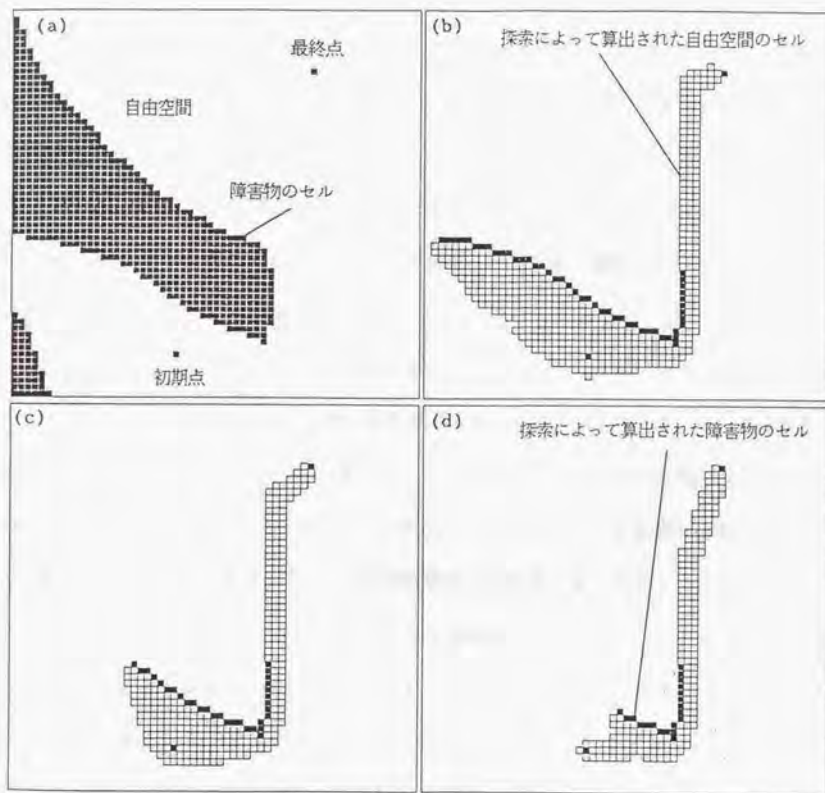


図 3.3 係数 $a(i)$ の影響
 (a) コンフィギュレーション空間の状態
 (b) $a(1) = 1, a(2) = 3$
 (c) $a(1) = 1, a(2) = 1$
 (d) $a(1) = 3, a(2) = 1$

3. 4 複数探索戦略の同時実行

以上で述べたように、障害物回避動作計画のために新しく導入したヒューリスティック関数を用いると、係数を適当な値に定めることによって様々な探索戦略を実現することが可能になる。あらかじめコンフィギュレーション空間の中で自由空間がどのようになっているかが分かれば、その情報にもとづいて適切な探索戦略を設定することも可能であるが、多次元のコンフィギュレーション空間の場合には、データ量の多さから、そのような情報を得ることは一般には不可能である。そこで、異なる探索戦略を用いた自由空間の算出手続きを同時に実行させ、ヒューリスティクスの平均化を図り、効率の低い戦略を用いることによる不必要な探索努力の実行を回避することを考える。

再び図3. 1の例で考える。自由空間算出前にはどのような探索戦略がよいかはわからないので、図3. 3 (b)(c)(d)の3つの探索戦略を同時に実行する。すなわち、3つの探索戦略によるセルの展開をかわるがわる順番に実行して行く。ここでは、セルの展開の際に、他の戦略による探索の進行状況は考慮しないとす。この場合は(d)の探索戦略が最も効率がよく、最も早く最終位置に到達する。1つの探索戦略で最終位置まで到達すれば初期位置と最終位置が自由空間のセルで連結されるので、他の探索をこれ以上実行する必要がない。この時点で(b)(c)の探索戦略による探索の途中状態は、図3. 4 (a)(b)に示されたようになり、これらの図を重ね合わせると図3. 5のようになる。同一のセルについては干渉チェックを1回だけ実行すればよいので、3つの探索戦略の同時実行に必要な干渉チェック回数は図3. 5の自由空間と障害物のセルの総数に相当する。自由空間算出の際には、幾何計算を多く必要とする干渉チェックの計算コストが大部分であるので、探索戦略を複数実行するために必要となる計算コストの影響は小さい。

しかし、以上の方法は同時に実行させた戦略すべてを同等に扱っており、単独の探索戦略

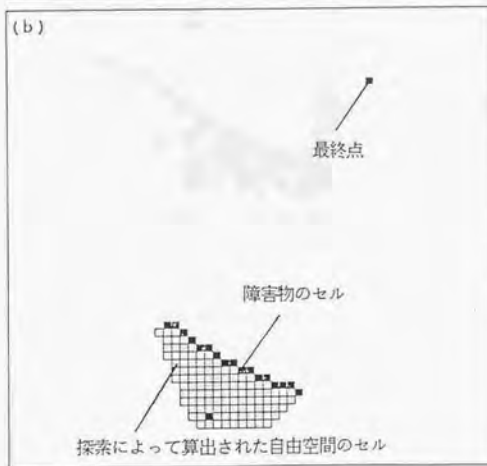
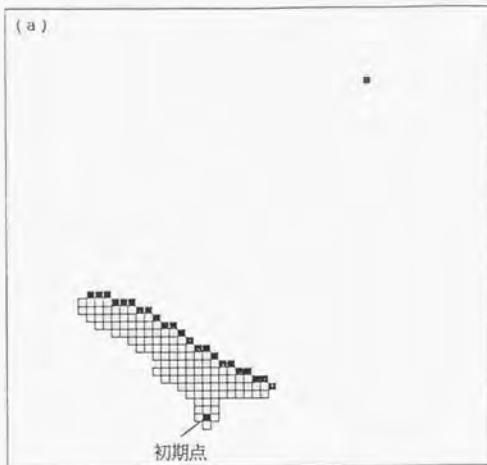


図3. 4 最も効率のよい戦略による探索が終了した時点の他の戦略による自由空間の状態
 ((a) $a(1)=1, a(2)=3$, (b) $a(1)=1, a(2)=1$)

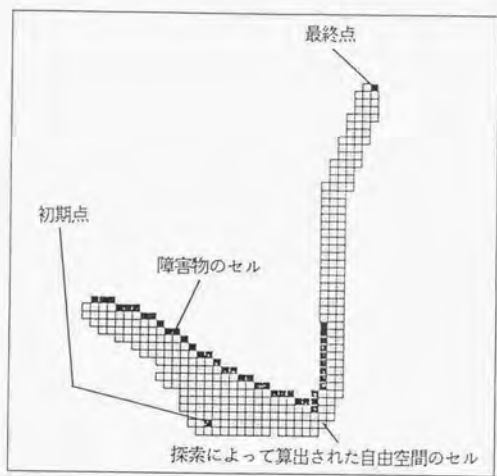


図 3. 5 3つの探索戦略によって算出された自由空間

を用いた場合に比べると、どうしても不必要な計算を実行する結果になってしまう。このような無駄な計算を少しでも実行させないようにするには、個々の探索戦略にどれだけの探索努力を投入し、全体として最も少ない計算で必要な自由空間を算出するかを計画する、さらに上位の戦略が必要になる。そこで、以下では、複数の探索戦略をどのように実行させるかについて考える。

3. 5 探索努力の最適配分問題

以上で述べたように、複数の探索戦略を用いて状況に応じて最も効率よく自由空間を算出するためには、それぞれの探索戦略にどれだけの探索努力(どれだけの計算)を投入するかを決定する必要がある。そこで本節では、このような議論をする準備として、オペレーションズリサーチの分野で研究されている探索理論[多田73]について触れる。ここで述べる探索理論は、軍事目的の敵艦艇の探索や海難事故の際の捜索などの応用を主に意図して研究されてきたものであるが、探索努力の最適配分という観点から、自由空間の算出の問題を議論する上で非常に参考になる。

ここで紹介する探索理論では、敵艦艇のような目標物の位置、出現などに関する確率法則や、発見に関する確率法則は既知であるとの前提のもとに、どこをどれだけ探すべきか、どこからどんな順序で探すべきかなどに関する理論を扱う。また、探索空間は探索によって荒らされない空間であると仮定する。すなわち、瞬間探知確率が探索に影響されず、探索努力のみの関数であると考えられる。

まず、いくつかの投資対象があって、1単位の資本をそのどれかに投資しようとする場合を考える。このようなとき普通は、それぞれの対象に1単位の資本を投資したときに得られる効用(利益)を調べ、その値が最大の対象に投資する。経済学の分野ではこの値を限界効用と呼ぶ。より数学的に表現すると、ある投資対象に対して努力 t を投入したとき、それから得られる効用を $g(t)$ とする。このとき、

$$g'(t) = \frac{dg}{dt} \quad (3-5)$$

を努力 t における限界効用と呼ぶ。

今、 n 個の投資対象 $A_i (i=1, 2, \dots, n)$ があると、微量の追加努力 e を $c_i (i=1, 2, \dots, n)$ に分

割して A_i に投入することによって、そこから得られる追加効用を最大にする問題を考える。
 ただし、全体の効用は個々の投資対象から得られる効用の和であるとし、 $g_i(t_i)$ を A_i の効用関数（微分可能）、 t_i を A_i に対する投入済みの努力量とする。一般性を失うことなく、

$$g'_1(t_1) \geq g'_2(t_2) \geq \dots \geq g'_n(t_n) \quad (3-6)$$

とする。努力 e の分割投入によって得られる追加効用を Δg と書けば、

$$\Delta g = \sum_{i=1}^n g'_i(t_i) e_i \quad (3-7)$$

と書ける。そこで上記の問題は、制約条件

$$\sum_{i=1}^n e_i = e; \quad e_i \geq 0, (i=1, 2, \dots, n) \quad (3-8)$$

の下で、

$$\Delta g \rightarrow \max \quad (3-9)$$

と表すことができる。

このとき、微小努力量 e を新たに投入して得られる効用を最大にするには、以下のよう
 すればよい。

- (1) $g'_i(t_i)$ が負（非正）であるような A_i に対しては努力を投入しない。
- (2) 正であるような $g'_i(t_i)$ があって、次の関係：

$$g'_1(t_1) = g'_2(t_2) = \dots = g'_k(t_k) > g'_{k+1}(t_{k+1}) \geq \dots \geq g'_n(t_n) \quad (3-10)$$

が成立しているとする。

- (a) $A_j (j = k+1, \dots, n)$ には努力を投入しない。
- (b) もし $g''_i(t_i) (i = 1, 2, \dots, k)$ のうち、非負なるものがあれば、その値の最大であるような対象に e を集中して投入する。

(c) もし $g_i''(t_i)$ ($i=1, 2, \dots, k$) のすべてが負ならば、 $a_i = -g_i''(t_i)$ とおいて、

$$c_i = \frac{a_i^{-1}}{\sum_{i=1}^k a_i^{-1}} c \quad i=1, 2, \dots, k \quad (3-11)$$

に従って c を分割投入する。

この定理を限界効用原理と呼ぶ。この原理の導出については参考文献[多田73]にゆずる。

次に、この限界効用原理を用いて、探索努力を最適に配分する問題を考える。例えば、探索空間が連続1次元空間である努力配分問題を取り上げる。1つの目標物が1次元空間のどこかにいるものとして、その存在の確率密度を $g(x)$; $-\infty < x < \infty$ とする。すなわち、位置を表す確率変数を X として Δx の範囲に存在する確率は、

$$Pr\{x \leq X \leq x + \Delta x\} = g(x)\Delta x, \quad g(x) \geq 0 \quad (3-12)$$

である。さらに、

$$\int_{-\infty}^{\infty} g(x)dx = 1 \quad (3-13)$$

としておく。また、区間 $(x, x + \Delta x)$ に投入された探索努力が $\varphi(x)\Delta x$ であるときの条件つき発見確率を $p(x, \varphi(x))$ とする。この $p(x, \varphi(x))$ については $p(x, 0) = 0$ 、 $p(x, \infty) = 1$ かつ $p(x, \varphi) \geq 0$ を仮定してよい。さらに現実の多くの問題を考えると、

$$p'(x, \varphi(x)) = \frac{\partial p(x, \varphi(x))}{\partial \varphi} \quad (3-14)$$

がすべての x について φ の単調減少関数 ($p(x, \varphi(x))$ が φ の増加凹関数) であると仮定できる。これを $p(x, \varphi(x))$ が効用通減の法則に従うという。このとき、全区間における目標物の発見確率 $P[\varphi]$ は、

$$P[\varphi] = \int_{-\infty}^{\infty} g(x)p(x, \varphi(x))dx \quad (3-15)$$

で与えられる。探索努力が Φ に制限されている条件のもとで、 $P[\varphi]$ を最大にしたい。つまり発見確率最大となる最適努力配分を求めたい。

この問題の解を限界効用原理に照らして予想してみる。今、全区間を幅 Δx の微小区間に区切って考え、その分点を x_i とすると、このときの限界効用は、 $g(x_i)p'(x_i, \varphi(x_i))\Delta x_i$ である。限界効用原理によれば、式(3-10)に対応して、探索努力を投入する区間ではこの量は一定でなければならない、さらに探索努力を投入しない区間についてはこの量は、投入する区間よりも小さくなければならないはずである。事実、このような連続区間の場合、 $\varphi(x)$ が最適解であるための必要十分条件は、 $\varphi(x) > 0$ であるような区間について、

$$g(x)p'(x, \varphi(x)) = \lambda (> 0) \quad (3-16)$$

また $\varphi(x) = 0$ であるような x については、

$$g(x)p'(x, 0) \leq \lambda \quad (3-17)$$

であることが証明できる[多田73]。

以上の議論では、探索努力を一括投入することを前提としていた。次に探索努力の一括投入と逐次投入の関係について述べる。ここでは、同じ1次元の問題について2通りの努力投入法を区別して考える。すなわち、

[一括投入型]

努力密度関数 $\varphi(x)$ にもとづき、探索努力を一括して投入する。この場合の最適解を φ^* とする。

[逐次投入型]

(1) まず、努力の総量 Φ を任意の2量 Φ_1 と Φ_2 に分ける。すなわち、 $\Phi = \Phi_1 + \Phi_2$

(2) Φ_1 だけが与えられているものとして、 $P[\varphi]$ を最大にするような $\varphi_1^*(x)$ を求め、これに従って Φ_1 を一括投入する。

(3) 目標物が見つからない場合には、 Φ_2 を配分するための努力密度 $\varphi_2(x)$ を

$$P[\varphi_2 | \varphi_1^*] = \int_{-\infty}^{\infty} g(x | \varphi_1^*) p(x, \varphi_2(x) | \varphi_1^*) dx \quad (3-18)$$

を最大にするように選ぶ。これを $\varphi_2^*(x)$ とする。これに従って Φ_2 を一括投入する。

このとき、はたして発見確率は等しいのか。すなわち、

$$P[\varphi^*] = P[\varphi_1^* | \varphi_1^*] \quad (3-19)$$

であるのか。また、努力密度の間に

$$\varphi^*(x) = \varphi_1^*(x) + \varphi_2^*(x) \quad (3-20)$$

という関係が成立つか、という疑問が生じる。ここでも、証明は参考文献[多田73]にゆずり、結論だけ述べる。この結果は、探索空間が離散的な場合でも成り立つ。すなわち、発見確率の最大化を目的とする探索努力配分において、次の2条件が満たされるならば、最適一括投入と最適逐次投入との間には式(3-19)と式(3-20)の関係が成り立つ。

条件1: 目標物の位置の確率分布が経時変化しないこと。すなわち、停留的な目標物であること。

条件2: 条件つき発見確率が投入努力量の増加凹関数であって、経時変化しないこと。

これが成り立つ場合、最適配分においては探索努力は加法的であるという。この結果は、努力分割の任意性を認めつつ、分割された個々の努力については、そのつど最適に配分することを要求し、結果として一括投入の場合の最適配分と一致することを主張している。

最後に、発見確率最大の努力配分を続けることが、期待努力最小型の探索努力配分に等しくなるかどうかについて考える。期待努力最小型の探索努力配分では、目標物が見つからな

ければ探索努力 Φ を $0 \rightarrow \infty$ と逐次増やさなければならず、 Φ の変化につれてそのつど $\varphi(x)$ を条件:

$$\int_{-\infty}^{\infty} \varphi(x) dx = \Phi$$

に従いつつ決めなければならない。これを明示するために、 $\varphi(x)$ を $\varphi(x, \Phi)$ と書けば、これが探索政策となる。今度は、発見までの期待努力を最小にするように最適な $\varphi(x, \Phi)$ を決定しなければならない。このとき、最適な $\varphi(x, \Phi)$ が、発見確率最大の立場でもとめた最適値 $\varphi^*(x)$ 一致するかどうか問題となる。具体的な証明は[多田73]にゆずるとして、結果のみを述べると、

「連続的な探索において、目標物が停留的であり条件つき発見確率が経時変化しないならば、発見確率最大化のための努力配分と、発見までの期待努力最小化のための努力配分とは一致する。」

すなわち、発見確率が最大になるように努力配分をつづけていけば、結果としては発見までの期待努力が最小になっている。これは、探索空間が連続的でも離散的でも成り立つ。

3.6 探索努力配分のための評価関数

前節で述べた探索理論によれば、発見までの期待努力最小化のための最適努力配分は、発見確率最大化の努力配分と一致し、このような探索を行うためには、探索努力を投入するたびに、その時点で最適な努力配分を行えばよい。この考え方を複数探索戦略を用いる自由空間の算出に当てはめると、それぞれの探索戦略を探索努力の投入対象として、干渉チェックを実行するセルの数を最小とするような努力配分を考えることになる。そのためには、それぞれの探索戦略について限界効用を評価し、最適な探索努力配分を決定する必要がある。自由空間を探索しながら順次算出する場合に、最も探索範囲が狭くなりそうな戦略を優先的に実行させ、必要な総計算量を最小化することで、次元の高いコンフィギュレーション空間を効率的に扱うことが可能になる。AIなどにおいて一般的に探索の効率を評価する尺度としては、例えば浸透度 [Nilsson82] や有効枝分れ係数 [Nilsson82] などがあるが、これらは探索が終了した時点での評価基準であり、探索を実行している状況で探索を制御するための評価方法とは異なる。自由空間の算出を探索によって行う場合には、探索の制御のために利用できる情報は、すでに探索によって参照された自由空間および障害物のセルの情報のみである。そこで、自由空間の算出において、これらの情報にもとづく探索努力配分のための評価法を新しく提案する。

ここでは、探索努力の適切な配分によって探索を制御するために、以下のように段階に分けた探索方法によって自由空間を算出させることにする。すなわち、探索の各段階で、それぞれの探索戦略について実行するセル展開の数を定め、探索戦略ごとに順番に決められた回数のセル展開を実行して行く。今、 S 種類の異なる探索戦略を用いるとし、 i 番目の戦略 ($i = 1, \dots, S$) は以下で示すヒューリスティック関数 h_i の係数 $\alpha_i(i)$ や A_i によって特徴づけられて

いるとする。

$$h_i(C) = A_i \sqrt{\sum_{i=1}^N (\alpha_i(i) \times (c(i) - c_j(i))^2)} \quad (i = 1, \dots, S) \quad (3-21)$$

さらに、 i 番目の戦略について、第 j 段階のセル展開回数を $E_i(j)$ とする。このとき、第 j 段階においては、まず最初の探索戦略によって $E_i(j)$ 回のセル展開を行い、続いて次の探索戦略によって $E_i(j)$ 回のセル展開を行う。同様に実行して、 S 番目の探索戦略によって $E_i(j)$ 回のセル展開を行ったあと、 $j+1$ 段階に進み、最初の探索戦略によって $E_i(j+1)$ 回のセル展開を実行する。ここで、 $E_i(j)$ の値を探索努力の投入量と考えて探索の制御を行う。このように探索を段階に分けた場合では、探索が連続的にならず、期待努力最小化のための努力配分と発見確率最大化の努力配分が一致することを厳密に証明することができなくなるが、実際上の話では、これらが一致すると考えてよいと思われる。

まず第1段階においては、それぞれの探索戦略の限界効用は未知である。そこで最初はすべてに同等の探索努力を投入する。すなわち、

$$E_i(1) = E_{init.} \quad (i = 1 \text{ to } S) \quad (3-22)$$

ただし $E_{init.}$ はあらかじめ決められた定数である。つまり、第1段階においては S 種類の各戦略について $E_{init.}$ 回のセル展開が順次実行される。

一方、第2段階以降では、それまでの探索の状況に応じて探索努力の投入量を変化させる。そのために、自由空間の算出過程で、 S 種類の戦略それぞれを評価する。ここで、 $P_i(j)$ を i 番目の戦略の第 j 段階における評価値とおくと、これは以下のようにして算出される。まず、自由空間算出の際には、セルが展開されるたび毎に、 i 番目の戦略の評価値を数値 $p_i(C)$ として計算する。

$$p_i(C) = \frac{g_i(C)}{f_i(C)} \quad (3-23)$$

ここで C は展開されたセルであり、 $q_i(C)$ は i 番目の戦略による探索の進み具合を表す正値をとる関数、 $r_i(C)$ は C を展開するまでに i 番目の戦略に対して実行した計算量(探索努力)を表す正値をとる関数とする。ここで、 $q_i(C)$ は探索が終了に近づく程、値が大きくなるように定義する。こうすると、 $p_i(C)$ の値が大きいいうことは、より少ない探索努力によって、より早く探索の終了状態に近づいていることになり、 $p_i(C)$ の値の大きい探索戦略はより有効であるということになる。これは、セル C を展開した時点での、そのセルを展開した戦略の評価値と考えることができ、近似的にこの時点での限界効用とみなすことが可能である。ただ、 $q_i(C)$ や $r_i(C)$ の具体的な関数形にもよるが、この数値の性質はあくまでも定性的なものであり、定量的な信頼性は余り高いとは言えない。そこで $P_i(j)$ は、 i 番目の戦略によって展開された最後の Q セルについての $p_i(C)$ の平均値として定めることにする。すなわち、

$$P_i(j) = \frac{\sum_{\text{(for last } Q \text{ cells)}} p_i(C)}{Q} \quad (3-24)$$

ただし、 Q はあらかじめ定めた定数である。

さて、第2段階以降におけるセル展開回数はこちらで定めた値を用いて以下のように計算することにする。この値の算出法も色々考えられるが、効用関数の実際の性質は未知であるので、効率の悪い戦略に投入される探索努力が相対的に他よりも小さくなるようにしながら、あまり極端な努力配分ではなく、かつ、単純に計算できるような形となるように考慮して決めてある。

$$E_i(j) = E_{\text{init}} \cdot \frac{P_i(j-1)}{\max\{P_1(j-1), \dots, P_S(j-1)\}} \quad (3-25)$$

さて、実際に計算を行うには、 $p_i(C)$ の関数形を具体的に与える必要がある。この与えかたは、 $a_i(i)$ や A_i と同様に全体としての自由空間の算出効率に影響をおよぼす。ここで、利用できる情報はすでに干渉チェックを実行して、自由空間であるか障害物であるかが分かっている

るセルに関するもののみである。この情報のみから、自由空間の形状にそった領域を参照する探索戦略を評価し、選定しなければならない。ここでは、初期点から速く(少ない探索努力で)遠ざかるものほど、探索戦略が持つ方向性と自由空間の広がり具合がうまく一致する傾向が高く、探索範囲が横に広がっていないと考えられるので、以下のような関数を定めた。

$$\begin{cases} g_i(C) = D_i(C)^N \\ r_i(C) = F_i(C) \\ p_i(C) = D_i(C)^N / F_i(C) \end{cases} \quad (3-26)$$

ここで、 $D_i(C)$ は i 番目の戦略によるセル展開で初期点からセル C までに通過したセル数(探索の深さ)、 N はコンフィギュレーション空間の次元、 $F_i(C)$ はセル C が i 番目の戦略によって展開されるまでに、すでに i 番目の戦略によって展開されている自由空間のセルの総数である。 $F_i(C)$ は、セル C が i 番目の戦略によって展開されるまでに i 番目の戦略に投入された探索努力であり、 N 次元の体積に相当する。一方、すべての探索戦略は同じ長さの経路を算出すると仮定すると、 $D_i(C)$ は、探索終了までに必要な探索努力とこれまでに投入した探索努力の比を長さの単位で表していると考えられる。そこで、ここではこれらの値の次元の違いを補正する意味で N を導入してある。ただ、これには物理的な根拠に基づいて探索戦略と自由空間の形状との適合具合を正しく反映しているとの保証はない。いずれにせよ、ここではそれぞれの探索戦略について、近似的であってもその限界効用を推定し、探索努力の配分を制御することに意味がある。

このようにして複数の探索戦略を実行すると、単一の探索戦略を実行する場合に比べて探索範囲が広がってしまう傾向を抑えることができ、全体として少ない期待努力投入量で必要な自由空間を算出することができる。こういった手法を導入することによって実際にどれだけの効果があるかについては、後で実験結果を踏まえて議論する。

3.7 両方向探索に基づく自由空間の算出

複数の探索戦略を用い、これらに投入する探索努力を適切に制御することで、自由空間の算出の効率を高め、戦略の選び方に係わらず安定して探索努力期待投入量を低く抑えることができる。しかし、以上の議論は単方向の探索に関してであり、これだけでは、効率が自由空間の算出の方向、すなわち探索の方向に左右され好ましくない。すなわち、初期点と最終点それぞれのまわりに障害物がどのように分布し、自由空間の状況にどのような差があるかを考慮しておらず、必要な計算量が大きく変化してしまう。そこで、自由空間の算出を初期点のセルと最終点のセルの両方から初め、自由空間の状態を評価しながら探索の方向についても制御することで、さらに効率の向上と安定化を図ることを考える。

例えば、図3.6のように、初期点のセルが障害物のセルに囲まれており、最終点のセル付近には障害物のセルがないような状況を考える。このとき、そのまま自由空間を算出すると(図3.6(a)の場合)、初期点のセルと最終点のセルを入れ替えて自由空間を算出する(図3.6(b)の場合)のでは効率に大きな差が生じてしまう。これは小さな穴に糸を通すが、すでに通してある糸を抜くのに比べてはるかに難しいのに似ている。このような場合に両方から自由空間の算出を始め、より少ない探索努力で必要な自由空間を算出できるのはどちらの方向の探索であるのかを評価し、その結果として図3.6(a)の方向の探索を優先するようにすれば効率の向上が図れる。

一般的には、障害物が複雑に配置される部分があっても、初期点、最終点が共にそこに含まれていなければ、その部分をひとまとまりとして大きな障害物として考えることにより、この部分の細部を考えなくても済む。一方、初期点、最終点の一方または両方が複雑に配置された障害物の中にある場合には、これらを回避する動きが不可欠であり、移動する方法はおのずから限られてくる。コンフィギュレーション空間において障害物(禁止領域)のセルに

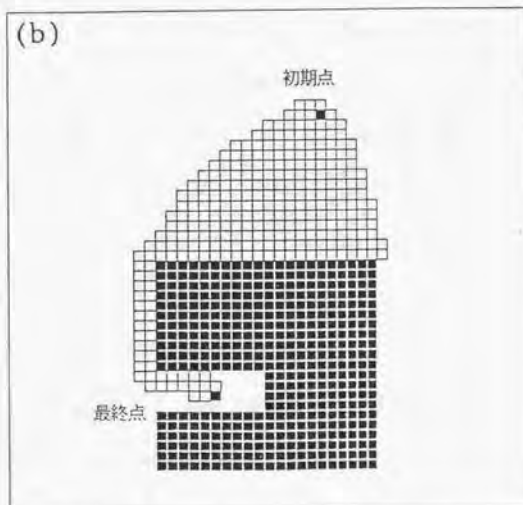
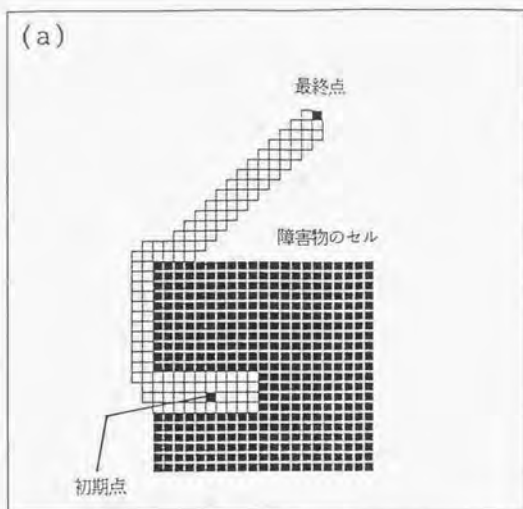


図3.6 探索の向き効率への影響
 (a) 障害物の多い方から少ない方へ (b) 障害物の少ない方から多い方へ

囲まれた領域で自由空間を展開しながら算出しようとする、自由空間の領域が広がりにくく、実際の移動経路も必然的にそこを通らなければならなくなるので、必要な自由空間が効率的に求まることが多い。また、周囲に障害物が密に配置されているときには、移動経路が存在しない場合もある。このときには、初期点を含む自由空間と最終点を含む自由空間が連結していないことになり、これを示すには、初期点から展開した自由空間ないしは最終点から展開した自由空間が、初期点と最終点を連結する前に障害物のセルに囲まれてそれ以上広がらなくなることを示す必要がある。この場合も、初期点からの領域と最終点からの領域のうちで狭い方を選んで、それがあるところ以上は展開できなくなることを示す方が、必要な計算量が少なく済む。多次元コンフィギュレーション空間では、次元が大きくなるにつれて、自由空間を膨張させることによる情報の増加の度合いが爆発的に増えるため、どちらの自由空間を広げて行くかは本質的な問題である。

そこで探索の方向については、同一方向で実行される複数の探索戦略とは扱いを変え、各段階において探索方向をどちらにするかを決定した上で、その探索方向について前節で述べたように複数の探索戦略を実行させることにする。すなわち、第2段階以降の自由空間の算出では、それ以前の探索の状況を評価した上で、前向き（初期点のセルから最終点のセルへ向かう方向）の探索、後向き（最終点のセルから初期点のセルへ向かう方向）の探索のいずれかのみを実行させる。つまり第2段階以降では、探索の進行とともに探索の方向が逆転することがあっても、前向き、後向きの探索が同時に実行されることはない。

図3.7は、両方向探索にもとづく自由空間算出の手順の概略を分かりやすい形で図示したものである。最初は両方の向きで自由空間の算出を実行し、適宜探索方向を変更しながら、初期点と最終点が自由空間のセルで連結されるまで探索を続ける。

今、前向きの探索に S_f 、後ろ向きの探索に S_b 種類の戦略を用い、合計 S_t 種類の戦略を

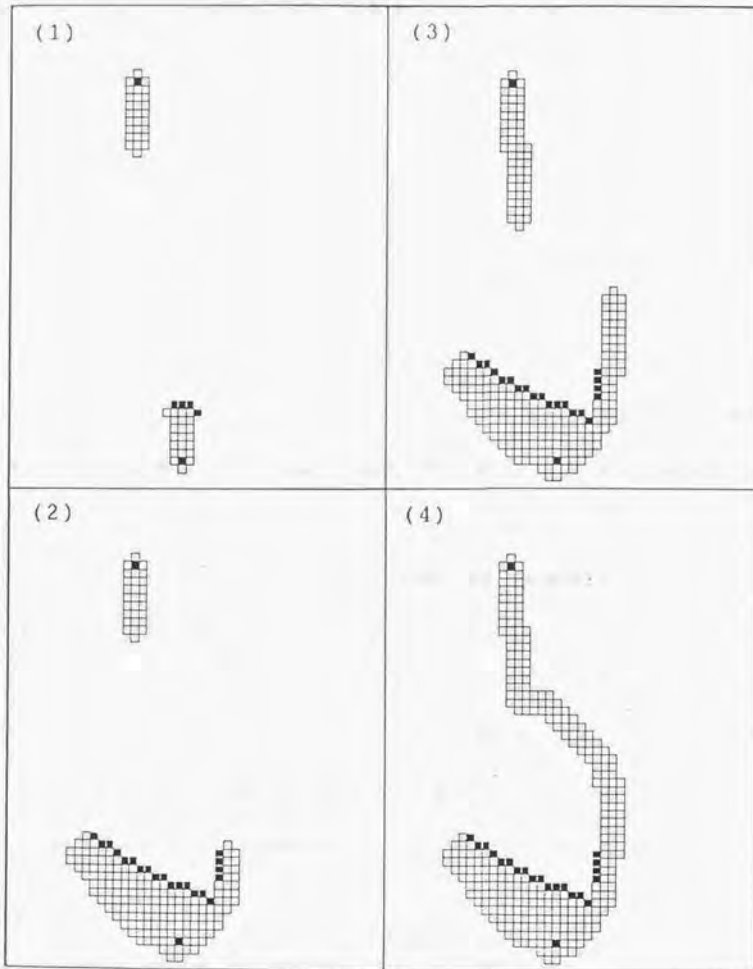


図 3. 7 両方向探索にもとづく自由空間算出手順

用いるとする。今、前向き探索の t 番目の戦略 ($t = 1, \dots, S_f$) と後向き探索の u 番目の戦略 ($u = 1, \dots, S_b$) は、それぞれ係数 $A_t, a_t(i)$ および $A_u, a_u(i)$ ($i = 1, \dots, N$) によって定められているとする。それぞれの段階において各探索戦略について実行するセル展開回数は、前節で述べたのと同じようにして計算される。すなわち、前向き探索の第 j 段階における t 番目の戦略のセル展開回数を $E_{f_t}(j)$ 、後向き探索の第 k 段階における u 番目の戦略のセル展開回数を $E_{b_u}(k)$ とすると、これら $E_{f_t}(j)$ と $E_{b_u}(k)$ の値を以下のように計算する。

$$\begin{cases} E_{f_t}(1) = E_{init.} \\ E_{f_t}(j) = E_{init.} \frac{P_{f_t}(j-1)}{\max\{P_{f_1}(j-1), \dots, P_{f_{S_f}}(j-1)\}} \quad (j > 1) \\ E_{b_u}(1) = E_{init.} \\ E_{b_u}(k) = E_{init.} \frac{P_{b_u}(k-1)}{\max\{P_{b_1}(k-1), \dots, P_{b_{S_b}}(k-1)\}} \quad (k > 1) \end{cases} \quad (3-27)$$

ただし $E_{init.}$ はあらかじめ決められた定数である。また、 $P_{f_t}(j)$ は前向き探索の第 j 段階における t 番目の戦略の評価値、 $P_{b_u}(k)$ は後向き探索の第 k 段階における u 番目の戦略の評価値である。

さて、探索の方向を決定するには、前向きの探索と後向きの探索について、必要となる期待探索努力を評価する必要がある。ここでは、以上で述べたように、障害物に阻まれて広がりにくい方が期待探索努力が小さいという考え方にもとづいて、前向きの自由空間の算出と後ろ向きの自由空間の算出の効率を、それぞれで展開した未知のセルのうち自由空間のセルになったものの割合で評価する。前向き探索、後向き探索の評価値を R_f 、 R_b とすると、これらの評価値は以下のように定義する。

$$\begin{cases} R_f = \frac{G_f}{H_f} \\ R_b = \frac{G_b}{H_b} \end{cases} \quad (3-28)$$

ここで G_f と G_b は、前向き、後向きの探索で展開された自由空間のセルの総数であり、 H_f と H_b は、前向き、後向きの探索で展開されたセルの総数（自由空間のセルと障害物のセルの

合計)である。この値が小さいということは、自由空間の領域が障害物のセルに阻まれて広がりにくいことを示しており、全体として少ないセルの展開で必要な自由空間を算出できる可能性が高い。

このときの自由空間の算出手順は以下ようになる。まず最初は、前向き探索と後ろ向き探索のどちらを実行したらよいかは分からないので、前向き、後ろ向き探索を共に実行させる。すなわち、 S_i 種類の戦略によるセル展開を順番に実行して行く。これに対して第2段階以降では、評価値 R_f と R_b に基づき前向き、後ろ向き探索によるセル展開のどちらか一方のみを実行させる。例えば、 R_f が R_b より小さければ、前向き探索がより少ない探索範囲で経路を発見する確率が高いと考えられ、次の段階では前向き探索によるセル展開のみを実行する。

このように、初期点付近と最終点付近の自由空間の状況を簡便な方法で評価し、それに応じて自由空間の算出の際に前向き、後ろ向きといった探索の向きを切り替えることで、さらに効率を高めることができる。

3. 8 探索戦略の入れ替え

以上に述べたように、複数探索戦略の利用、両方向探索の実行により、かなり複雑な場合でも効率良く経路を算出することができる。しかし、現実のインプリメンテーションでは、同時に実行する複数探索戦略の総数にも限界があり、これらの中でどのように努力配分を考へても、複雑な問題においては、なかなか解が求まらない場合も出てくると思われる。そこで、このようなインプリメンテーション上の問題を回避するため、あらかじめ定めた一定回数の干渉チェックを行っても経路が見つからない場合には、係数 $a(i)$ の変更により探索戦略を入れ替えて自由空間の算出をやり直すような方法が考えられる。やり直しにおいては、すでに干渉チェックが実行済みのセルについては、再び干渉チェックを実行する必要がない。そのため、干渉チェックに必要とされる演算が大部分を占める自由空間の算出においては、やり直しによる効率の低下は少ないと考えられる。

3.9 第3章のまとめ

第3章では、多次元コンフィギュレーション空間において、経路探索に関連する自由空間を、探索によって限定して算出する方法について議論した。特に、探索の際すでに参照された自由空間の情報をもとに、与えられた初期点と最終点を結ぶ経路が存在する確率の高い方向に探索の重点を変更しながら自由空間を算出する探索戦略を提案した。

まず、特定の方向を重点的に探索する戦略が、本論文で新しく提案したヒューリスティック関数によって実現できることを示した。

そのうえで、複数の探索戦略を同時に実行し、ヒューリスティクスを平均化するとともに、探索努力配分のための評価法を提案し、これを用いてそれぞれの探索戦略に投入する探索努力を制御することで探索の重点を変更する手法を提案した。

さらに、初期点、最終点まわりの自由空間の状態の差を探索に反映させるために両方向探索を導入するとともに、探索戦略の入れ替えによりさらに効率の向上を図る手法を提案した。

第4章

自由空間の表現と経路の算出

4.1 システム構築に必要な技術

第3章では、自由空間算出のための探索戦略のみに話題を絞って議論してきた。第4章では、障害物回避のプログラムをインプリメントしアプリケーションに組み込むために必要なその他の技術について議論する。

まず最初に、経路の有無をチェックするための手法を提案する。第3章の議論では暗黙に経路の存在を仮定し、その上で高い次元のコンフィギュレーション空間を効率よく扱うことを考えてきた。しかし、アプリケーションによっては経路の有無をまずチェックする必要がある、その際にはこれまでと異なる手法で効率化を図る必要がある。

次に、第3章で述べたような手法で限定的に算出したコンフィギュレーション空間の情報を効率的に管理・記憶するための階層的なデータ構造について述べる。また、これに関連して第3章で述べた手法を拡張して、階層的な自由空間の算出を行なう方法を提案する。

最後に、干渉以外の緩やかな制約条件を考慮した経路探索の手法を、幾何学的な問題に限定して議論する。

4. 2 経路の有無をチェックするための探索戦略

これまでの議論では、移動経路が存在することを前提とした上で、必要最小限の自由空間を如何にして限定的に算出するかを考えてきた。ここでは、経路の有無をチェックする場合に適切な探索戦略について述べる。

経路の有無をチェックするためには、与えられた初期点と最終点を含むそれぞれの自由空間が位相的に接続しているかどうかを調べれば良い。前節までで述べた方法では、移動経路が存在しない場合には、初期点を含む自由空間ないしは最終点を含む自由空間を自由空間のセルで埋めつくし、グラフを展開できなくなってプログラムが停止する。一方、本節で述べる方法では、コンフィギュレーション空間の境界に沿って限定的にセルをチェックする。すなわち、自由空間の境界面に沿って限定的にセルを選出し、自由空間を算出されたセルで埋め尽くすことがない。

自由空間を算出する手続きは以下のようになる。図4. 1は、2次元の例題を用いて本方法の概略を説明したのものである。探索の最初の段階では、初期点と最終点のセルを除いてすべてのセルが未知のセルである(図4. 1(1))。次に、これら2つのセルを結ぶ仮の経路が生成され、これに沿って干渉チェックを実行する。もしここで障害物のセルが見つかった場合は、これに隣接する自由空間のセルを自由空間の境界のセルとして記憶しておく(図4. 1(2))。これらの中で初期点に最も近い自由空間の境界のセルを、自由空間算出の出発点とする(図4. 1(3))。自由空間のセルは展開され、順次自由空間の境界に沿って算出される。この処理は初期点と最終点から自由空間のセルで連結されるか、境界の自由空間のセルがなくなるまで続けられる(図4. 1(4))。自由空間算出の後でも、境界面に隣接するセルのみについて干渉チェックが実行され、コンフィギュレーション空間の大部分のセルは干渉チェックを実行する必要のない未知のセルとして残っている。図4. 2は、図4. 1と同じ例題に、

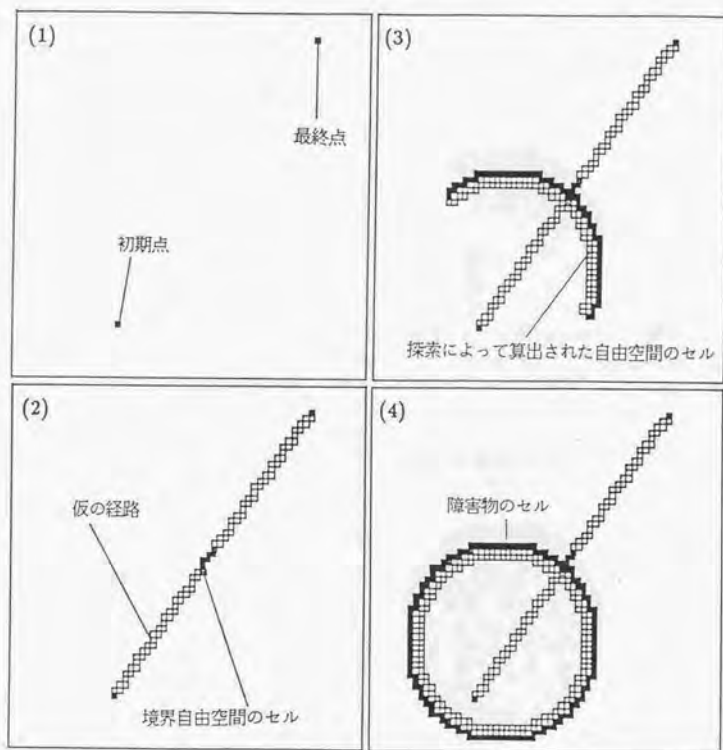


図 4. 1 経路の有無をチェックするための探索戦略

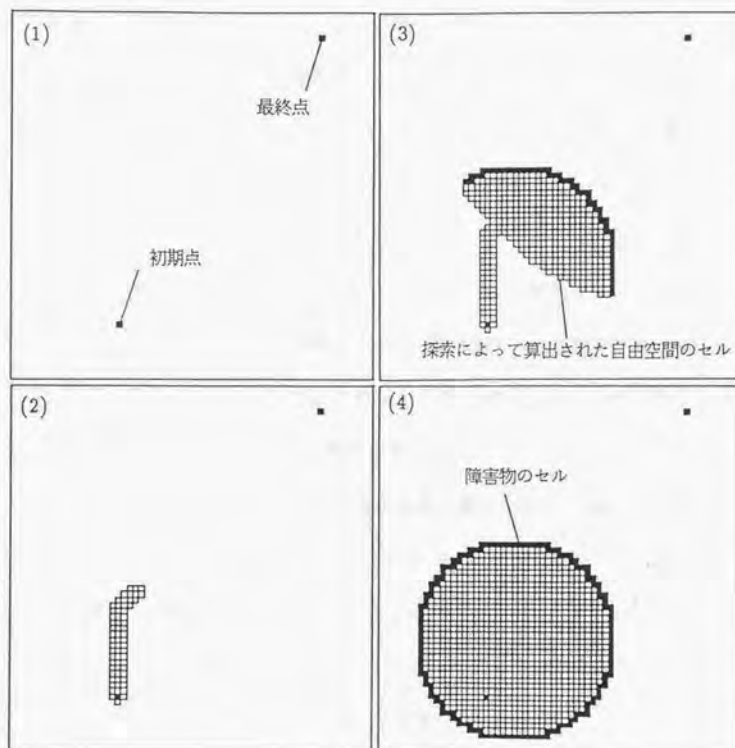


図4.2 通常の探索戦略を用いた場合の自由空間の算出

第3章の3.4節で議論したような単方向の探索戦略で自由空間の算出を実行した場合の様子を示している。ここでは、自由空間と障害物との境界面と関係に係わり無く、すべての隣接する自由空間のセルを展開候補のセルとして登録するため、自由空間が算出されたセルで埋められている。

本節で議論する経路の有無をチェックする探索も、基本的には、ヒューリスティックなグラフ探索に基づいている。ただし、展開候補のセルを選ぶ基準は極めて単純であり、式(3-2)与えられる評価関数のうち、 $g(C)$ は常にゼロであり、 $h(C)$ は最終点までの距離とする。つまり、最終点に最も近いセルが選択される。一方ここでは、自由空間の境界面をトレースするために、隣接するセルの一部のみを展開する新しい戦略を用いる。

以下の議論の準備として、まず、自由空間のセルをさらに、境界自由空間のセルと内部自由空間のセルに分類する。境界自由空間のセルは、障害物のセルに隣接する自由空間のセルであり、内部自由空間のセルは、自由空間のセルに取り囲まれている自由空間のセルである。このとき、自由空間は以下のように算出される。

自由空間の算出は、図4.3(1)のように境界自由空間のセルから開始される。次に、これに隣接するセルについて干渉をチェックする(図4.3(2))。隣接する自由空間のセルについては、さらにそれが境界自由空間のセルであるかどうかをチェックする。図4.3(2)の場合、展開されたセル(自由空間算出の出発点のセル)は下向きに障害物のセルと隣接するので、隣接する自由空間のセルが同じ下向きに障害物のセルと隣接するかどうかをチェックする。もしこの方向に障害物のセルと隣接すると、これらは連続した自由空間の境界面を形成するので、これらは、その後のセル展開の候補として記憶される(図4.3(3))。これら候補のセルは、候補のセルがなくなるまで順番に展開されて行く。図4.3(4)では、図4.3(3)の右側の境界自由空間のセルが展開されている。それぞれの境界自由空間のセルが展開

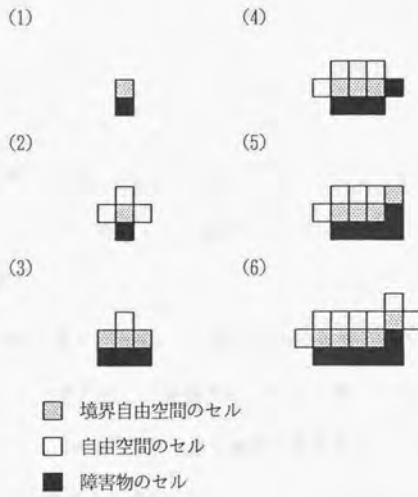


図 4. 3 セル展開のための局所的な探索戦略

される時には、隣接する障害物のセルについて、連続した境界面となるさらに隣接する境界自由空間のセルがあるかどうかをチェックされる。例えば、図4.3(4)右端の新しく見つかった障害物のセルの場合、上下の方向に境界自由空間のセルがあるかどうかをチェックされる(図4.3(5))。右方向に隣接する自由空間セルがあっても、図4.3(4)で展開された境界自由空間のセルと連続した境界とはならないので、ここではチェックされない。この障害物のセルについてのチェックによって、図4.3(5)右上の境界自由空間のセルのように、互いに直接隣接しない境界自由空間のセルを検出することが可能になる。図4.3(6)はさらに2つのセルが展開された状態を示している。

図4.4では、セル展開の処理手順をC風の構文で要約した。N次元コンフィギュレーション空間では、それぞれのセルが2N個のセルに隣接する。図4.4では、展開候補セルと隣接セルとの関係を i, j の整数で表現している。

以上のように、自由空間の算出は続けられ、展開候補の自由空間のセルがなくなるか、初期点と最終点が互いに自由空間のセルで連結された時点で終了する。展開候補の自由空間のセルがなくなった場合は、自由空間の境界面が境界自由空間のセルで覆い尽くされたことに対応し、これは、初期点と最終点に経路が存在しないことを意味する。

また、場合によっては、自由空間が複数の境界面をもつ場合がある。例えば、閉じた自由空間が、湖のなかの島のように、いくつかの独立した障害物を持つ場合を考える。この場合は自由空間は独立した複数の境界面を持っており、1つは湖の岸に相当する外側の境界面で、他は湖のなかの島に相当する内部の境界面である。このような場合、経路の存在を調べるためには、これら内部の境界面に沿って自由空間のセルを算出する必要はない。そこで、セルを展開するための局所的な戦略の他に、外側の境界面を発見するための大域的な戦略を導入する。

```

Cell expansion (candidate cell to be expanded) {
  for (i = 0; i < 2N; i++)
    /* Check all neighboring cells in N-dimensional space. */
    /* Direction from candidate cell (upward, right, etc.) is coded by integer: i. */
    if ( obstacle cell )
      Record direction code (i) for later checking;
  for (i = 0; i < 2N; i++) { /* Check all neighboring cells again. */
    if ( free space cell )
      if ( neighboring obstacle cell exists in same direction as candidate cell )
        Record this cell as candidate cell;
    if ( obstacle cell )
      for (j = 0; j < 2N; j++) /* Check neighboring cells of this obstacle cell */
        if ( i != j && free space cell which is not yet expanded )
          Record this cell as candidate cell;
  }
}

```

図 4 . 4 セル展開アルゴリズム

外側の境界面を他と区別するために、自由空間算出の最初に定義する仮の経路に沿った干渉チェックの結果に基づいて以下のようにラベル付けをする。まず、仮の経路に沿って干渉チェックをした際に見つかった境界自由空間のセルを、他の境界自由空間のセルと区別して扱い、これらをさらにスタートセルとスキップセルと呼ばれる2つに分類する。スタートセルは、仮の経路にそって最終点側に隣接する障害物のセルがあるもので、スキップセルは同様に初期点側に隣接する障害物のセルがあるものである。スタートセル、スキップセルのそれぞれの集合は、初期点からの距離でソートされているとする。つまり、初期点のセルは最初のスタートセルと仮の経路に沿って自由空間のセルで連結されており、 i 番目のスキップセルは j 番目のスタートセルと仮の経路に沿って自由空間のセルで連結されている(ただし $j=i+1$)。

自由空間の算出は、最初のスタートセルから開始され、これと同じラベルが算出された自由空間のセルに付加される。例えば、図4.5(1)は2次元コンフィギュレーション空間において、いくつかの境界自由空間のセルが算出された状態を示している。展開候補のセルは前節と同じ基準で選択する。図4.5に示した矢印は、自由空間のセルが算出されて行く方向を表している。算出された自由空間のセルが i 番目のスキップセルに隣接したときには、自由空間算出手続きの出発点が、 j 番目のスタートセルに変更される(図4.5(2))。このような状況では、それまで算出された自由空間のセルは内側の境界面に対応すると考えられ、 j 番目のスタートセルが外側の境界面に対応すると考えられる。そのため、自由空間の算出は j 番目のスタートセルからセルの展開を始めることで外側の境界面に変更され、それまでの探索は抑制される。また、算出された自由空間のセルに与えられるラベルも、境界面を区別するために変更される。異なるラベルを持つ自由空間のセルが互いにつぶかった場合には、これら2つの自由空間のセルの集合は1つにまとめられ、再度ラベル付けが行われる

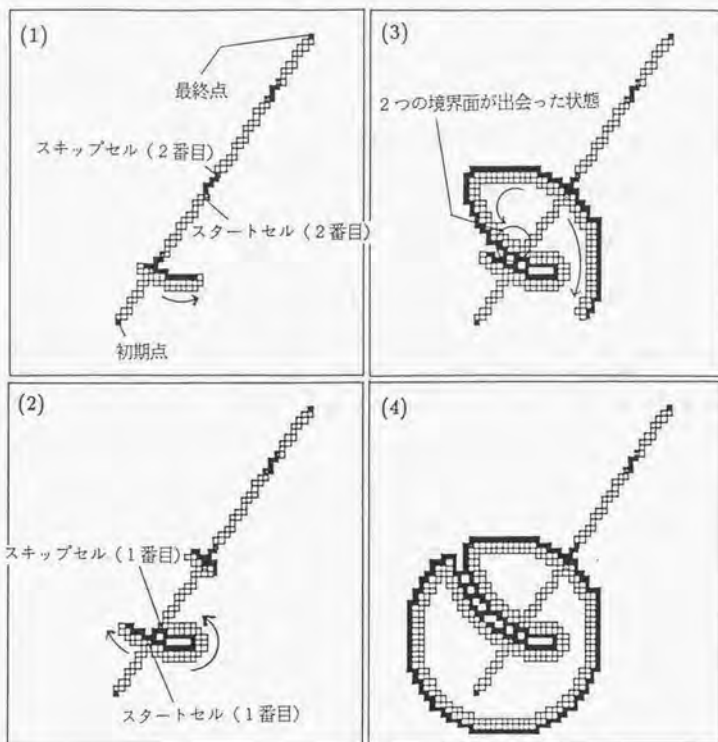


図4.5 外側の境界面を発見するための大域的な探索戦略

(図4.5(3))。自由空間の算出は外側の境界面のラベルを持つ展開候補の自由空間のセルがなくなったときに終了する(図4.5(4))。

図4.6は、干渉のない移動経路が存在する場合の自由空間の算出結果を示した例である。こういった状況は、算出された自由空間のセルが最後のスキップセルに隣接するかどうかで判定できる。

本方法は、 N 次元コンフィギュレーション空間の自由空間を埋め尽くす代わりに、自由空間の表面を埋め尽くすことによって、自由空間の接続関係をチェックしようとするものである。すなわち、 N 次元の問題を、 $N-1$ 次元の表面の問題に置き換え、効率化を図ろうとする考え方である。このため、コンフィギュレーション空間の次元が高くなると、覆い尽くすべき表面の次元も同様に増大し、情報量が爆発的に増大する。つまり、自由度が高い場合には、この方法でも経路の有無のチェックが困難になることが予想される。現実に本手法が有効となるのは、第7章で述べるプラントCADの例題のように、3から4自由度程度の問題であろうと考えられる。

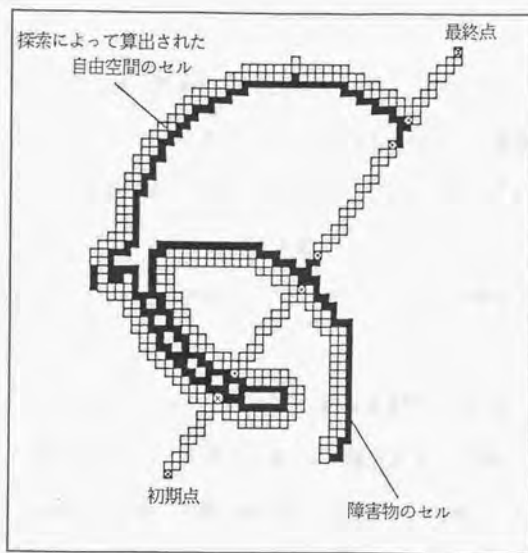


図 4. 6 経路が存在する場合の自由空間の算出結果

4.3 自由空間の階層表現

以上で述べたような探索戦略を用いることで、広大な多次元コンフィギュレーション空間の極く一部の自由空間のセルおよび障害物のセルのみで経路を探索することが可能になった。これまでの議論は、必要な自由空間のみを限定するための探索戦略と探索の制御に関してであったが、ここでは、広大なコンフィギュレーション空間の一部のデータのみを効率よく管理・記憶する方法について述べる。

ここで用いるデータ構造は、基本的には規則的にコンフィギュレーション空間を分割して得られる多次元配列である。しかし、膨大な多次元配列をそのまま計算機上に用意しようとするのは現実的でなく、探索戦略を工夫することによって自由空間を限定したことによるメリットを生かすことにならない。少ないメモリ量でなるべく多次元のデータが扱えるように、探索によって算出された自由空間のセル、障害物のセルのみに限定的にメモリを割り当てるようにする。

図4.7は、コンフィギュレーション空間の管理・記憶機構を、2次元コンフィギュレーション空間を例にとって概念的に示したものである。この図のように、各軸を半分に分けることによって全体空間を分割し、分割して得られたサブ空間に自由空間のセルまたは障害物のセルがある場合には再帰的に分割を繰り返す。この空間分割の履歴は木構造として表現され、この木構造の末端のノードに自由空間のセルであるか、障害物のセルであるか、またはいずれも含まないかという属性を記述することにより、空間の状態を表現できる。N次元空間の場合には 2^N 個のサブ空間が存在するので、このような手法は、2次元の場合にクワドトリー、3次元の場合にオクトリーと呼ばれる。例えば6次元コンフィギュレーション空間を扱う場合には、1つのノードに64個(2^6)のサブ空間が存在する木構造となる。一例として、各パラメータ(各軸)の最大分割数を128($=2^7$)とすると、7層の木構造としてコンフィ

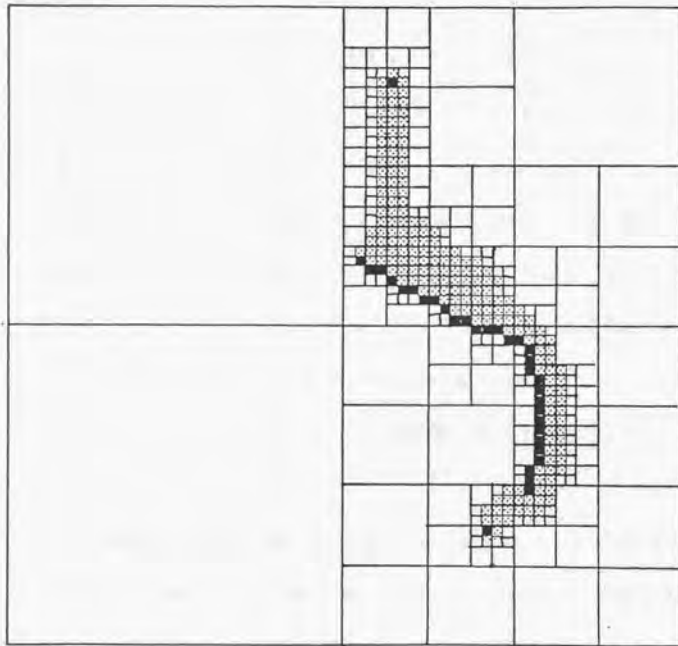


図 4、7 コンフィギュレーション空間の階層表現

ギュレーション空間が表現できる。このようにすると、自由空間のセルや障害物のセルが含まれるサブ空間は再帰的に細かく分割されるが、未知のセルのみの部分は分割されずに残るので、必要な部分の情報を効率よく管理することができる。

このように空間を再帰的に分割する考え方は、Faverjonら [Faverjon89] によっても採用されている。彼らは3次元コンフィギュレーション空間において自由空間すべてを陽に算出した結果を管理する方法として、オクトリー的な手法を用いている。

以上で述べたコンフィギュレーション空間の再帰的な分割は、探索による自由空間の算出の進行と共に実行して行く。最初は、初期点と最終点以外は未知のセルであり、これら2点を含むサブ空間のみがセルの大きさになるまで再帰的に分割されている。探索が実行されていくと、それに伴って新しく自由空間のセルや障害物のセルが算出されるので、これらを含むサブ空間は再帰的に分割され、新しく算出されたセルをノードに登録する。このように、最初は初期点と最終点のみを表現するのに最低限必要なノードでコンフィギュレーション空間を表現し、探索の進行と共に必要なノードを追加しながらコンフィギュレーション空間を表現する木構造を成長させて行く。

簡単な例として、縦軸、横軸を8分割した2次元コンフィギュレーション空間を考える。ここで、自由空間のセルが算出されるに従ってコンフィギュレーション空間を表す木構造がどのように変化するかを図4.8に示す。この図では、初期点と最終点のみの状態(図4.8(1))、初期点と最終点を1度展開した状態(図4.8(2))、初期点と最終点が自由空間のセルで連結した状態(図4.8(3))のそれぞれについて、コンフィギュレーション空間でのセルの様子、サブ空間分割の様子、それに対応する木構造(クワドトリー構造)を示した。クワドトリーでは1つのノードに4つの子孫ノードが存在する。この図では、子孫ノードの左から順に、左上、左下、右上、右下のサブ空間を示している。探索の進行とともに、サブ空間

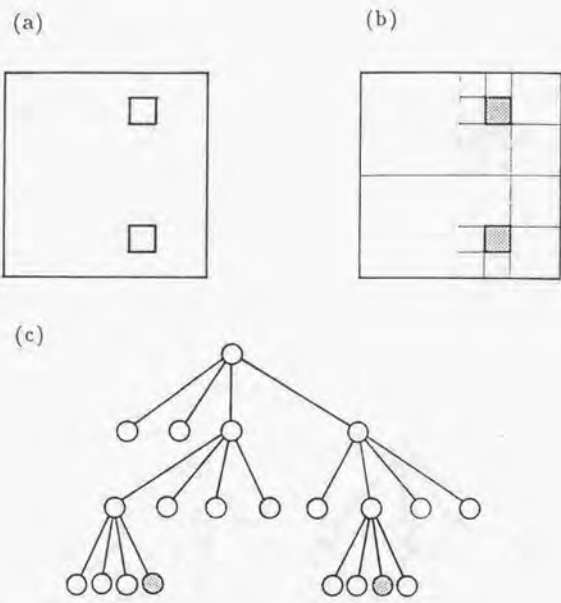


図 4. 8 (1) 自由空間算出に伴う木構造の変化 (探索開始時)
 (a) コンフィギュレーション空間の状態
 (b) コンフィギュレーション空間のサブ空間分割
 (c) コンフィギュレーション空間を表現する木構造

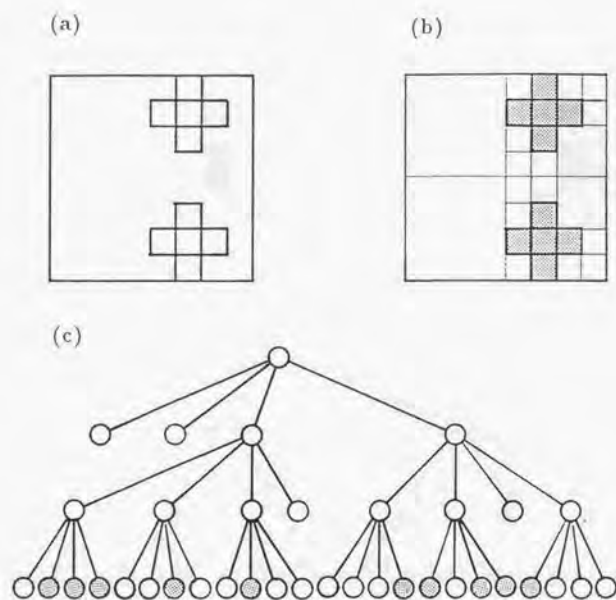


図 4. 8 (2) 自由空間算出に伴う木構造の変化 (探索中)
 (a) コンフィギュレーション空間の状態
 (b) コンフィギュレーション空間のサブ空間分割
 (c) コンフィギュレーション空間を表現する木構造

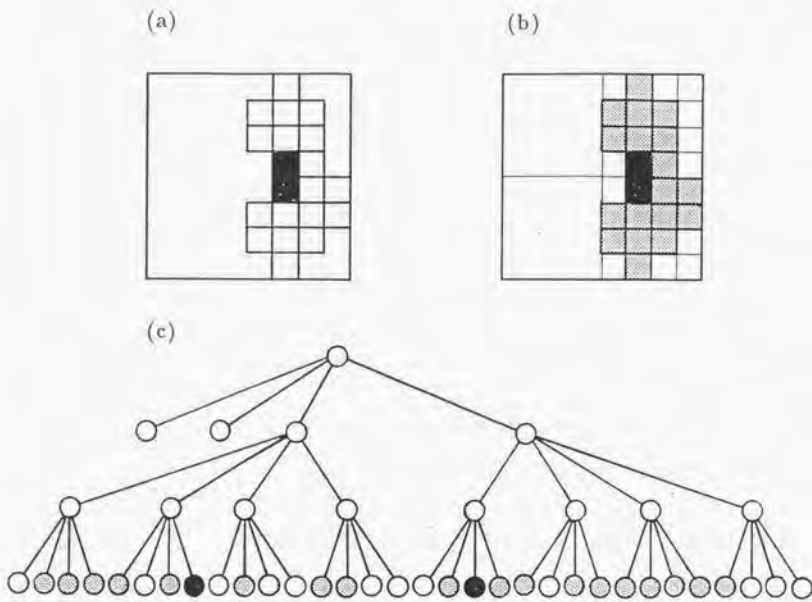


図 4. 8 (3) 自由空間算出に伴う木構造の変化 (探索終了時)
 (a) コンフィギュレーション空間の状態
 (b) コンフィギュレーション空間のサブ空間分割
 (c) コンフィギュレーション空間を表現する木構造

分割が探索結果に応じて実行され、ノードが付加されながら木構造が成長して行く過程が分かる。

4.4 階層的な自由空間の算出

前節では、規則的に等しくN次元配列状に分割されたコンフィギュレーション空間を、階層的なデータ構造を用いて効率よく記憶・管理する方法について述べた。このように必要に応じて階層的にコンフィギュレーション空間を細かく分割する考え方は、自由空間の算出そのものにも拡張することができる。例えば、障害物がかなり密集している部分と障害物がほとんど存在しない部分が両方ある場合などは、算出しようとする経路全体にわたって細かな分割数で探索する必要はない。最初は荒い分割数で大きな近似を行い、それにもとづく経路探索を行って、不都合のある部分のみでさらに詳細な経路探索を実行することは、アプリケーションによっては重要なアプローチとなる。

このような考え方は、Kambhampatiら [Kambham86] が移動ロボットの経路探索問題について提案している。ここでは、周囲の障害物が既知の場合の2次元空間における移動問題を扱っており、最初は大まかに探索を行い、順次詳細化することによって効率向上を図ることができる。

同様に、まずコンフィギュレーション空間を表現する木構造の深さを制限することによって荒く分割して経路探索を行い、それで求めた移動経路が必要な精度で安全であるかどうかをチェックし、条件が満足されない部分について木構造の深さを追加して、さらに細かく経路探索を行えばよい。

4.5 緩やかな制約条件を考慮した経路探索

これまでの議論は、障害物と干渉してはならないという厳しい制約条件のみを考えて自由空間を算出し、経路を求める方法について考えてきた。しかし、実際のアプリケーションでは、このような条件の他に緩やかな制約条件を考慮して経路を算出する必要がある場合もある。このような緩やかな制約条件としては、障害物からなるべく離れた経路を通りたいといった問題が考えられる。このように満足することが必ずしも要求されていない条件については、必ず満足しなければならない条件と同時に考えるのが難しい場合もある。そこで本節では、このような緩やかな制約条件の扱いについて考える。ここでは、これまでの議論で用いた知識のみで考えることのできる幾何学的な問題の一例として、障害物からなるべく離れた経路を求める問題について述べ、緩やかな制約条件を考慮した経路探索の基本的な考え方を示す。

まず最初は、実空間において移動物体と障害物との最小距離があまり小さくならないように移動経路を計画することを考える。この場合には、探索を実行しながら自由空間を算出している段階で、コンフィギュレーション空間の状態が分かっていなくても、判断基準である移動物体と障害物との最小距離を計算することができる。このような場合には、探索戦略の性質を決める評価関数に考慮すべき制約を反映させることで、緩やかな制約条件まで含めた自由空間の算出ができる。より具体的には、評価関数 $f(C)$ を以下のように変形する。

$$f(C) = g(C) + h(C) + q(C) \quad (4-1)$$

ここで $g(C)$ と $h(C)$ は第3章の式(3-2)で定義したのと同じである。 $q(C)$ はここで付加的に考慮しようとしている制約条件に関する項であり、この場合は移動物体と障害物との最小距離が小さいほど $q(C)$ が大きな値を取るように定義する。このようにすると、移動物体と障害物

との最小距離が小さい場合には、それに応じたペナルティが付加されるため、なるべくそこを通らないように自由空間が算出される。

次に、コンフィギュレーション空間において障害物との最小距離があまり小さくならないように移動経路を計画することを考える。この場合は、コンフィギュレーション空間の状態が分かっていると、付加的な緩やかな制約条件を考慮することができない。そこで、満足することが必ずしも要求されていないこのような条件については、必ず満足しなければならない条件と別々に扱うことにする。すなわち、まず必ず満足しなければならない条件のみを考慮して経路を決定し、次に緩やかな条件をできるだけ満足するように経路を修正するアプローチを取る。図4.9に単方向の自由空間算出を行って求めた経路の例を示す。この経路は、算出した自由空間のなかで最短経路をA*アルゴリズムで求めたものであり、レンガ状のハッチングを施したセルとして示してある。この移動経路は、障害物をかすめるようなものになっているので、これを障害物から離れた経路に修正することが必要になる。

移動経路修正の基本的な処理手順は以下のようになる。まず、算出された移動経路をもとに自由空間を広げる。ここで算出された自由空間が、修正した移動経路を求める際の探索範囲となる。広大なコンフィギュレーション空間においては、経路の修正の際にも自由空間すべてを考えることは不可能であり、このようにして修正に際して考慮する自由空間を拡大する。次に、この自由空間に隣接する障害物のセルから自由空間内に波面をひろげる。波面が到達した自由空間のセルには近くに障害物が存在し、波面が到達していないセルは近くに障害物がないことがわかる。移動経路としては、なるべく障害物からの波面の到達していないセル、すなわち近くに障害物のないセルを通るものが望ましい。このようなことを考慮して再び移動経路を探索しなおし、出力する。図4.9の例について上記の処理を行った結果を示す。図4.10は、図4.9の経路を3回膨脹させて得られた自由空間である。

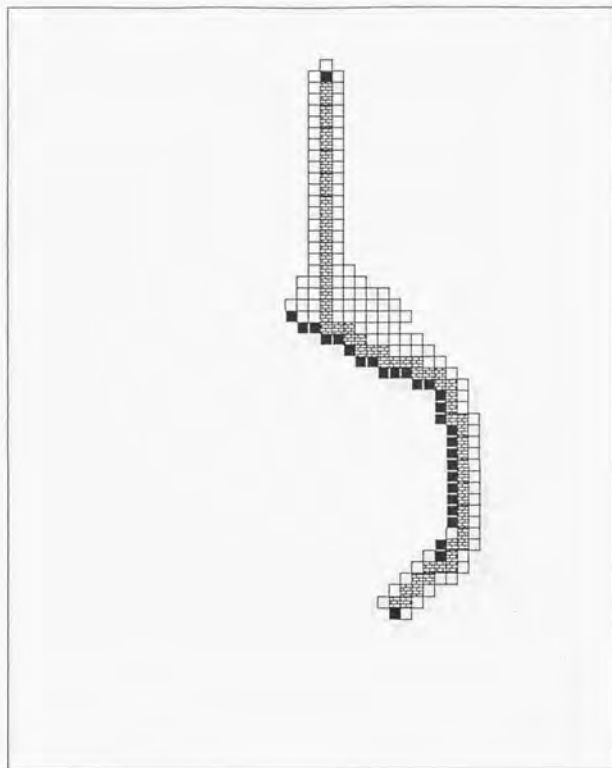


図 4 . 9 求まった移動経路の例

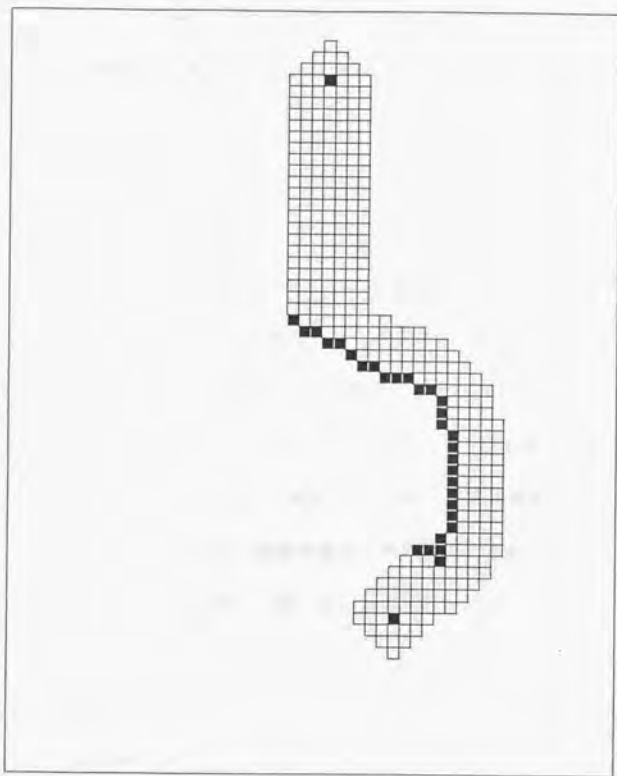


図 4. 1 0 移動経路を膨張させて得られた自由空間

図4.11はこの自由空間に隣接する障害物のセルから波面を3回膨脹させた結果である。

図4.12は障害物の近傍を通らないように探索をなおした結果である。

以上で示した方法は、コンフィギュレーション空間の限定された領域について、ある種のポテンシャル場を定義し、この情報を利用して経路修正を行っていると考えることができる。ここでは障害物からの距離の情報を表すポテンシャル場に相当するデータを障害物からの波面膨脹で作成したが、目的に応じて変更すればよい。

このようなポテンシャル場の生成や移動経路の膨脹などに必要な波面膨脹は、自由空間の算出に用いたグラフ探索手法を応用して実行できる。この場合には、ヒューリスティック関数を常に0とすることによって横型探索戦略とし、探索の最初に設定する展開候補のセル(第3章3.2節におけるリストOPENに相当)として波源となるセルを用いればよい。上の例で自由空間を拡大する場合には経路のセルが波源となり、ポテンシャルの生成では障害物のセルが波源となる。このとき、まず波源となるセルにコスト値0が付加され、展開候補の自由空間のセルとして登録される。セル展開によって新しく展開候補となったセルには、もとのセルのコスト値に1が加えられ、展開候補のセルとして登録される。セル展開の際には、展開候補のセルの中からコスト値の一番小さいセルが選ばれるので、コスト値0を持つ展開候補のセルから展開され、その結果波源のセルが1回膨脹し、次にコスト値1のセルが展開されて2段目の膨脹になる。例えば3回波面の膨脹を行ないたいければ、コスト値3のセルが初めて選ばれた時点で処理を中止すればよい。

また、ポテンシャル場を考慮して移動経路を生成しなおすときには、それぞれのセルを通過するときに、ポテンシャル場に応じて適当なコストの上乗せを行えばよい。図4.12の経路は、障害物から1回目、2回目、3回目の波面膨脹で波面が到達したセルについて、それぞれ9、4、1のコストを通常のコストに上乗せし、その上でコスト最小の経路をA*アル

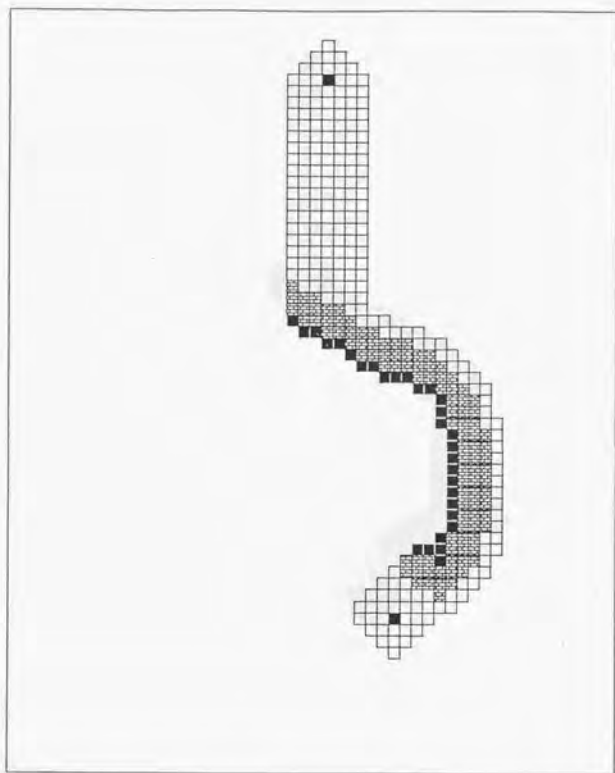


図4.11 障害物から波面膨張させた結果

ゴリズムで算出した。

以上の方法を6自由度PUMA型マニピュレータに適用した。すなわち、コンフィギュレーション空間において障害物との距離があまり小さくならないように経路を修正させた。コンフィギュレーション空間は6次元関節角空間であり、各軸5度おきに分割してセル分割してある。図4.13は修正前の移動経路であり、直方体のワークを立方体の障害物を避けて移動させている。図4.14は、この例題について、修正前のワークの動きと修正後のワークのみの動きを示したものである。修正前と修正後の移動経路の違いがよくわかる。ここでは、自由空間の拡大、障害物からのポテンシャル場の生成には、共に波面膨張を3回ずつ実行した。

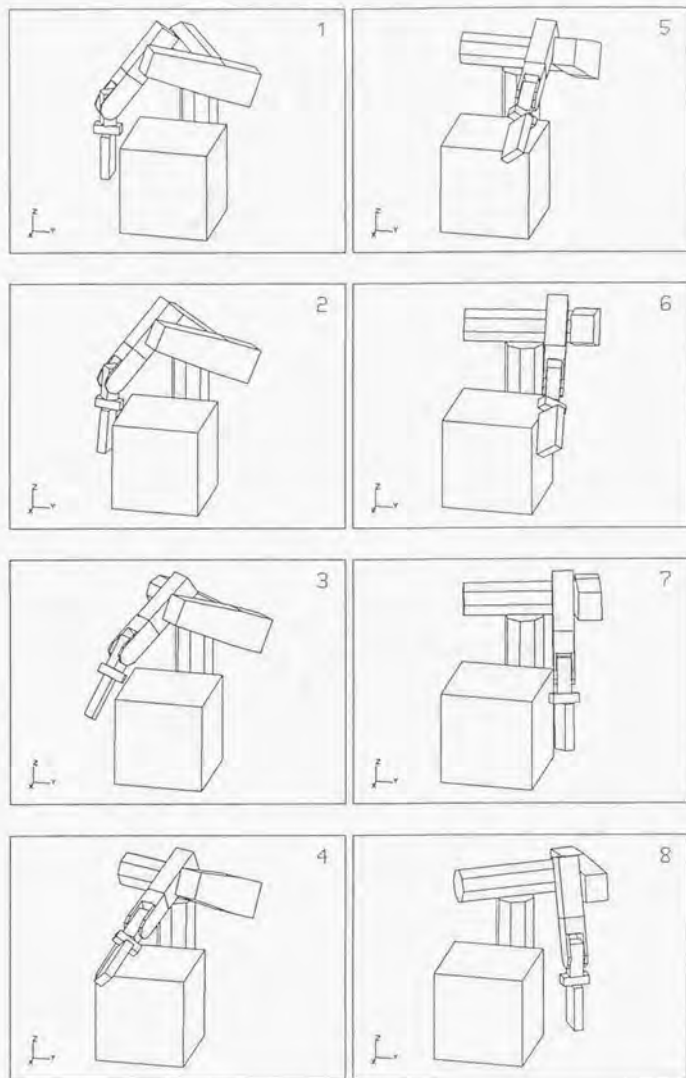


図4.13 6自由度マニピュレータの修正前の移動経路

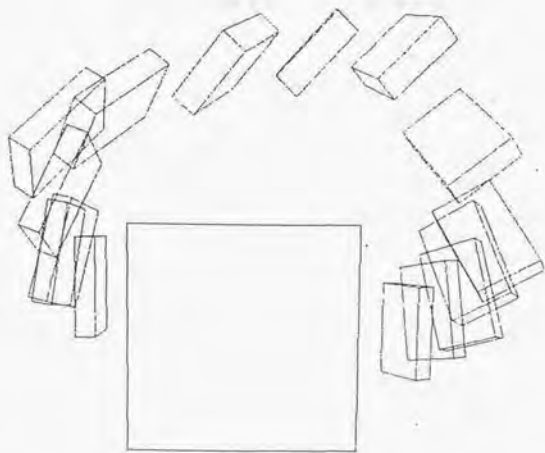
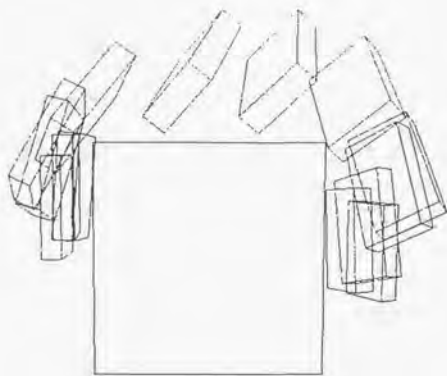


図 4. 1 4 6 自由度マニピュレータの場合の移動経路の修正

4.6 第4章のまとめ

第4章では、第3章で述べた自由空間算出のための探索戦略以外で、実際のアプリケーションで障害物回避を実行するのに必要な技術について議論した。

まず、移動経路の有無を効率的にチェックできるように、自由空間の境界面のみを探索する戦略を提案し、目的に応じて適切な戦略を利用でき、効率化を図ることが可能であることを示した。これにより障害物回避のアプリケーションの幅が広がり、工業的な意味での重要性は大きい。

次に、探索によって算出された自由空間、障害物のセルの情報を効率的に記憶・管理する方法について提案した。これにより、6次元空間のような多次元コンフィギュレーション空間を効率よく扱うことが可能になった。

さらに、自由空間を階層的に算出する方法、緩やかな制約条件を考慮した経路探索などにも触れ、主に第3章で議論した本手法の拡張性についても議論を行った。

第5章

実験による効率の評価

5.1 例題

第5章では、これまで述べた自由空間の算出方法の効果を実験的に評価した結果について述べる。特に、複数探索戦略の利用や両方向探索といった手法が、次元の高いコンフィギュレーション空間における自由空間の算出に、どの程度の効果があるかを確認する。

そこで、これまで述べた自由空間の算出方法を、実際に計算機上にインプリメントし、様々な例題に適用した。この実験では、これまで適用が困難とされてきた6次元のコンフィギュレーション空間を用いた障害物回避動作計画を取り上げ、様々な6自由度を持つ移動物体の例題に適用した。実験に用いた例題は全部で12あり、図5.1から図5.12までの図は、それぞれの例題についてインプリメントしたプログラムで算出した移動経路の例を示している。コンフィギュレーション空間は6次元空間であるので、各階層において64のサブ空間を持つ64分木で表現することができ、深さを7段に設定して各軸最大128個までセル分割できるようなプログラムを用いた。また、すべての形状は凸多面体とそれらの組み合わせに限定し、凸多面体どうしで直接干渉チェックを実行した。

図5.1から図5.4までは、PUMA型マニピュレータについて移動経路を算出した結果である。この例では、マニピュレータの関節角によって6次元関節角空間を定義しコンフィギュレーション空間とした。セル分割はすべての軸を5度間隔で量子化して行った。

図5.5から図5.7までは他の6自由度多関節マニピュレータ(JEMRMS)について移動経路を算出した結果である。このJEMRMS[Yamawaki89][山脇89]のものについては第6章で詳しく述べるが、基本的にはPUMAと同じ関節の配置になっている宇宙用マニピュレータである。ここでも、位置・姿勢は関節角によって指定し、6次元関節角空間をコンフィギュレーション空間とした。各軸5度おきに量子化を行いセル分割している。

図5.8から図5.10は、2つのPUMA型マニピュレータが協調して一つのワークを運

が例題である。この場合は2つマニピュレータについて合計12の関節があるが、ワークを介して閉ループ機構となっており、12自由度の系ではない。ここでは、図左側のマニピュレータの関節角で位置・姿勢を指定し、図右側のマニピュレータの関節角は、ワークの位置・姿勢から逆問題を解くことによって求めた。また、逆問題に解が無い場合は、障害物と干渉する場合と同等に扱い、禁止領域となるようにした。ここでも各軸5度おきに量子化を行いセル分割している。

図5.11、図5.12は、U字型をした物体が3次元空間において、並進3自由度(x方向、y方向、z方向)と回転3自由度(x軸回り、y軸回り、z軸回り)の合計6自由度の剛体運動する場合である。この例では、初期位置・姿勢と最終位置・姿勢はx座標、y座標、z座標、それぞれの軸回りの回転角で指定した。回転角についてはそれぞれ5度おきに分割し、並進自由度については80ごとに分割してセル分割した。ただし、ここでU字型をした物体の大きさは縦、横、高さは523、500、100である。

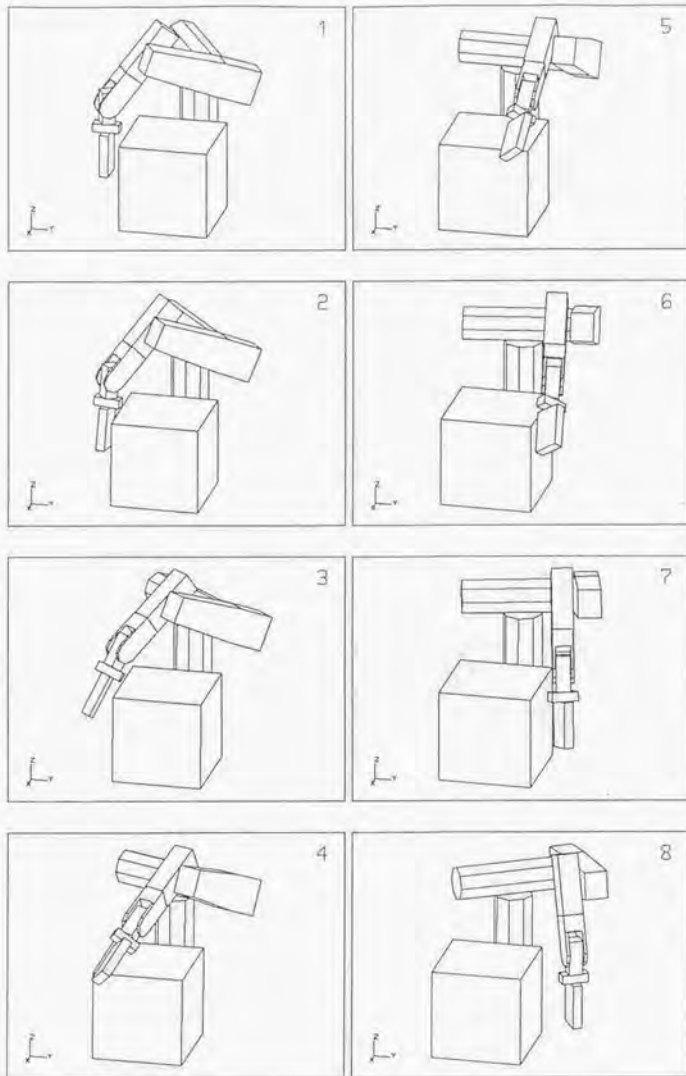


図5. 1 例題1 (PUMA型マニピュレータ)

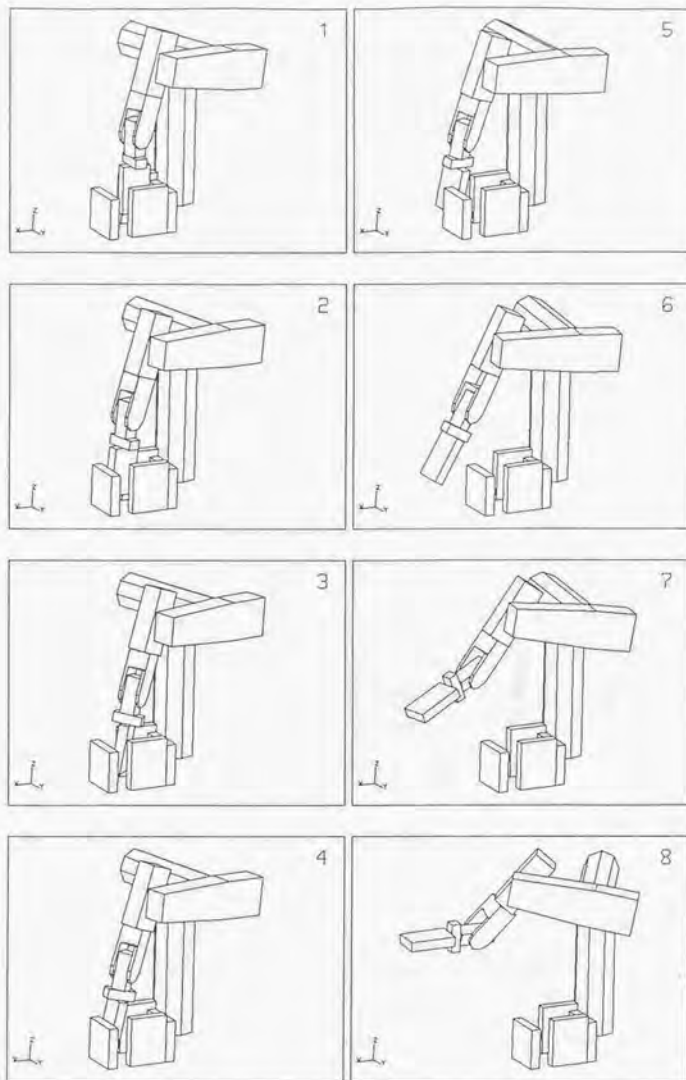


図 5. 2 例題 2 (PUMA 型マニピュレータ)

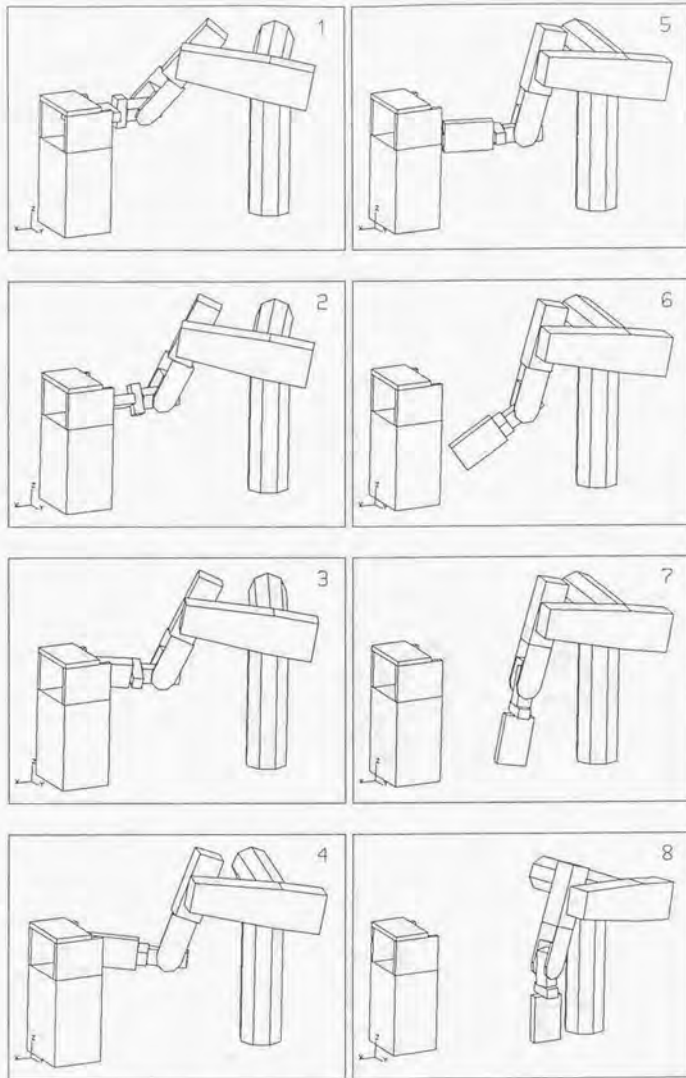


図 5. 3 例題 3 (PUMA 型マニピュレータ)

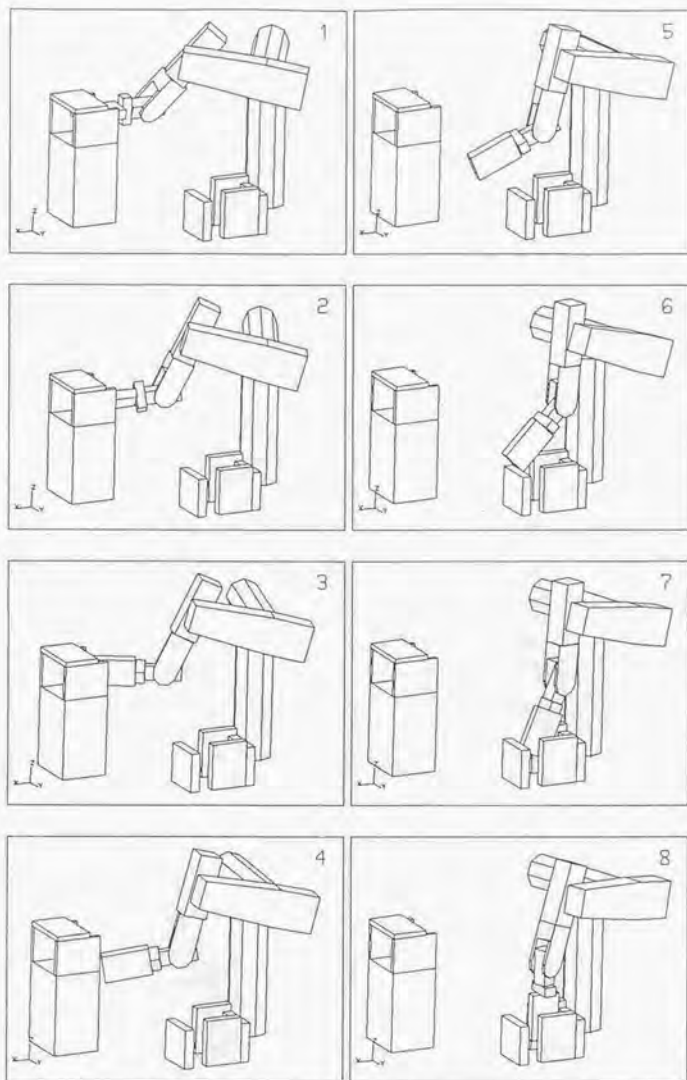


図5.4 例題4 (PUMA型マニピュレータ)

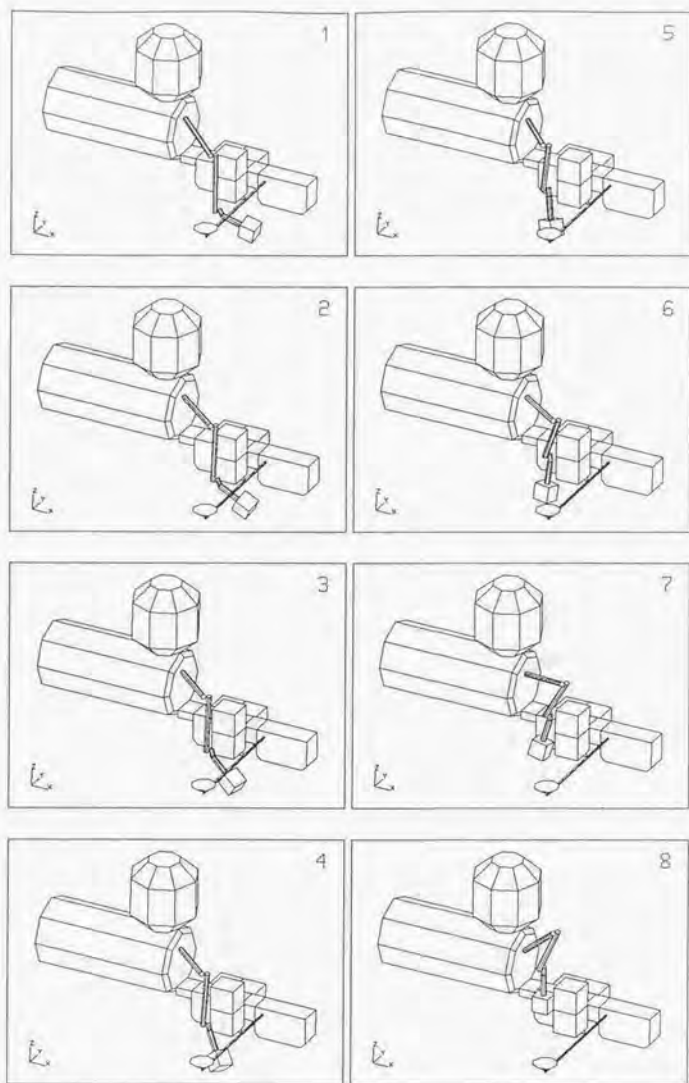


图 5.5 例题 5 (JEMRMS)

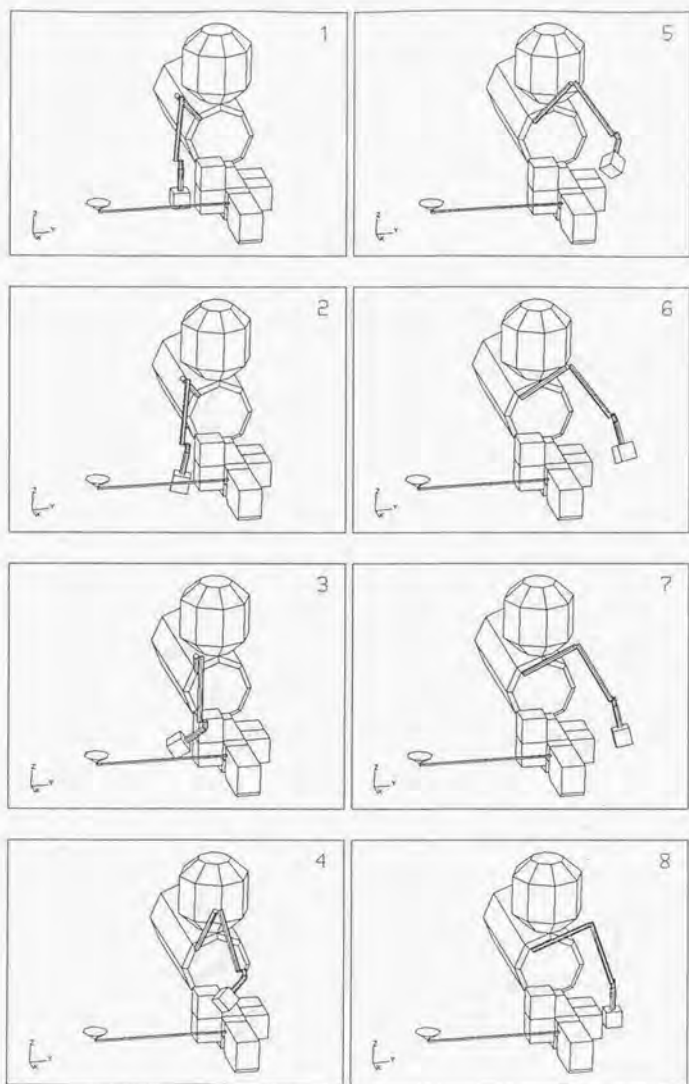


图 5.6 例题 6 (JEMRMS)

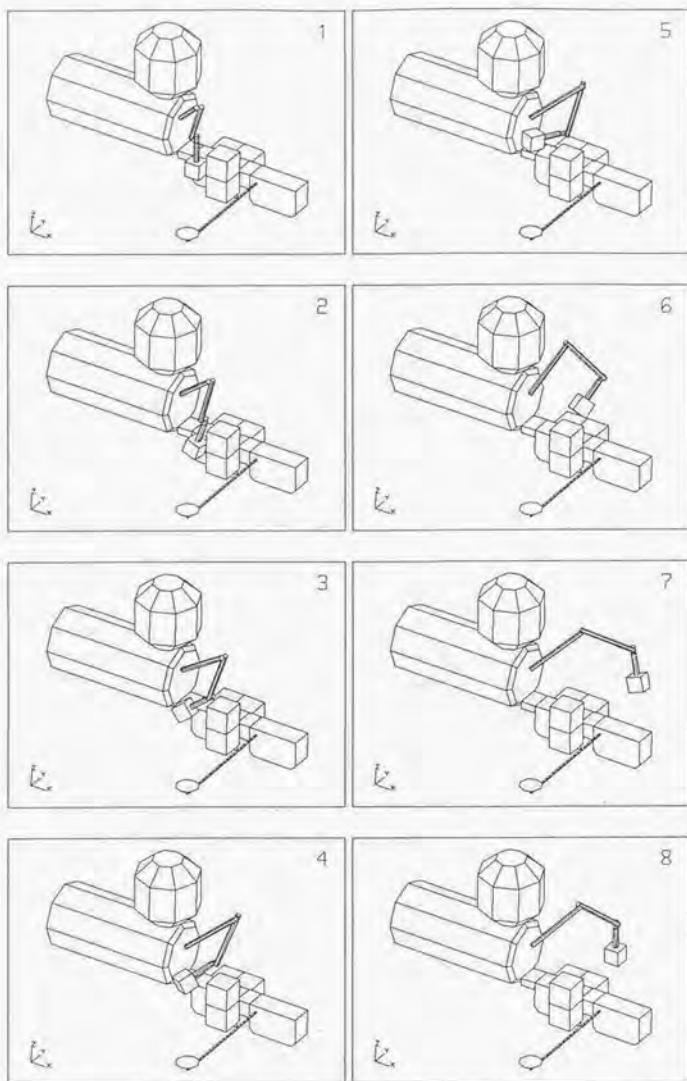


图 5. 7 例题 7 (JEMRMS)

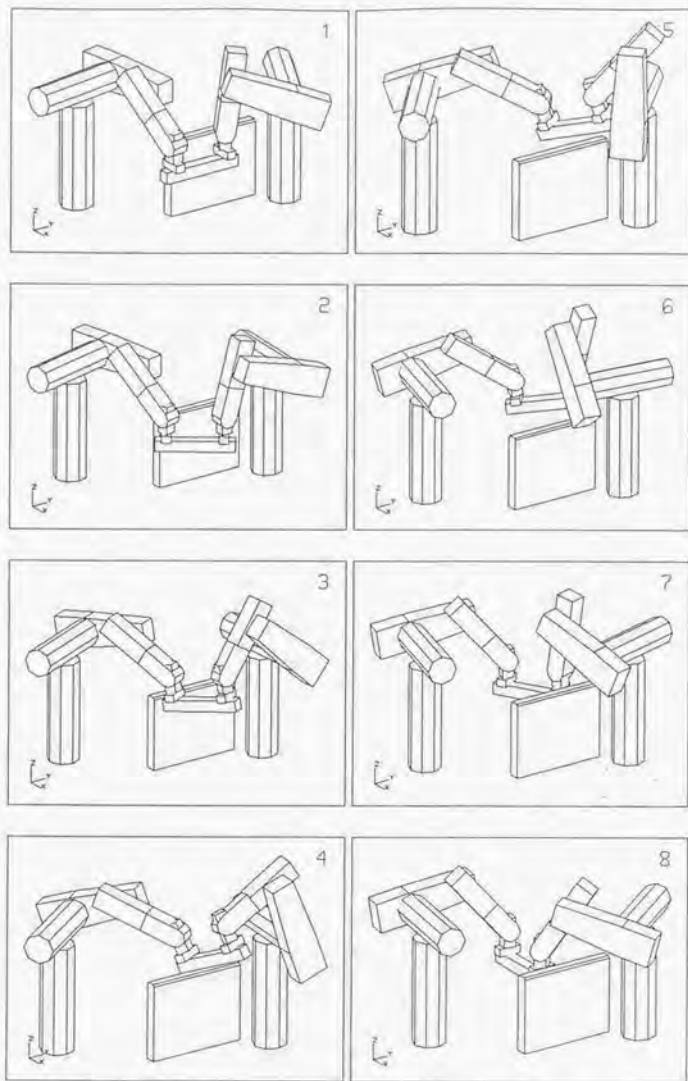


図 5. 8 例題 8 (2 アーム協調動作)

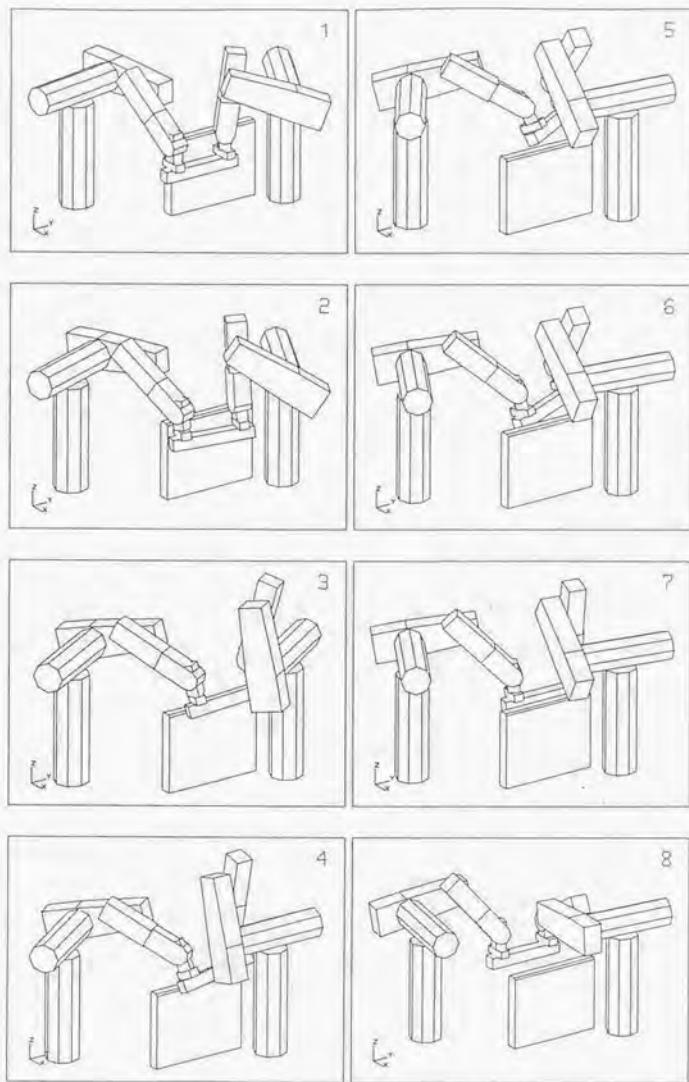


図 5. 9 例題 9 (2 アーム協調動作)

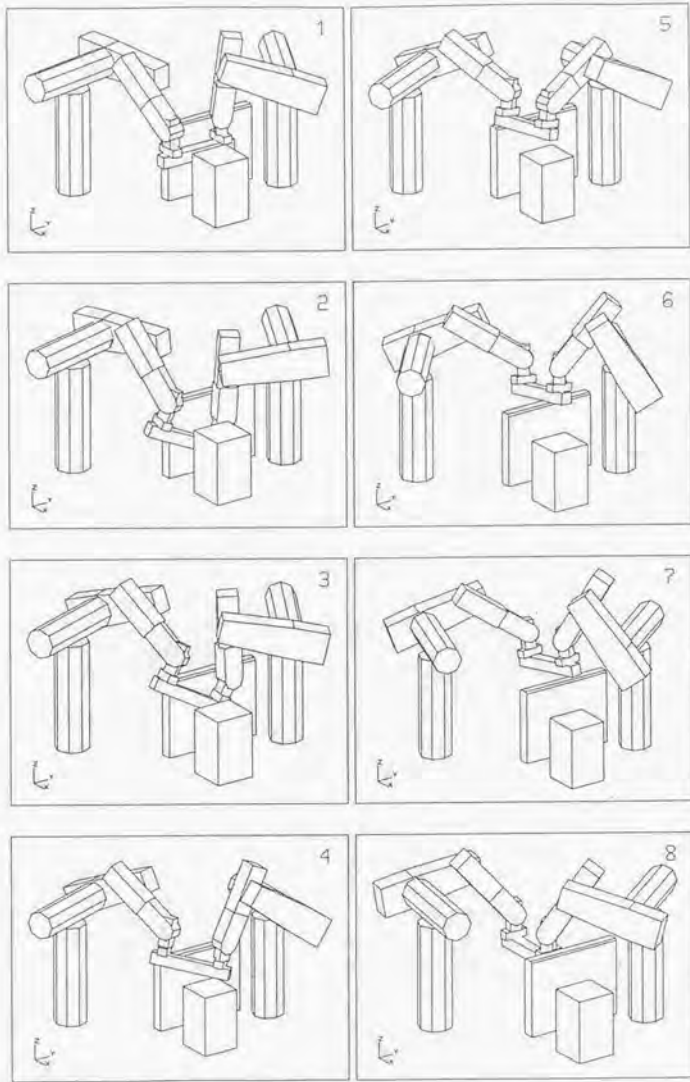


図5.10 例題10 (2アーム協調動作)

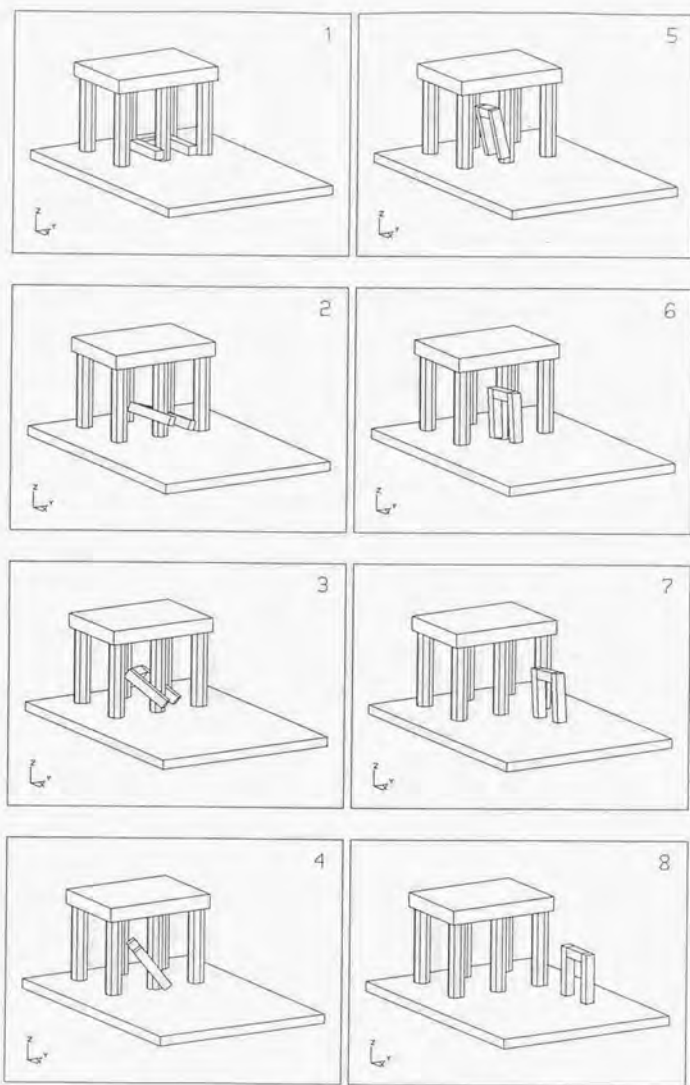


図 5. 1 1 例題 1 1 (U型物体の剛体運動)

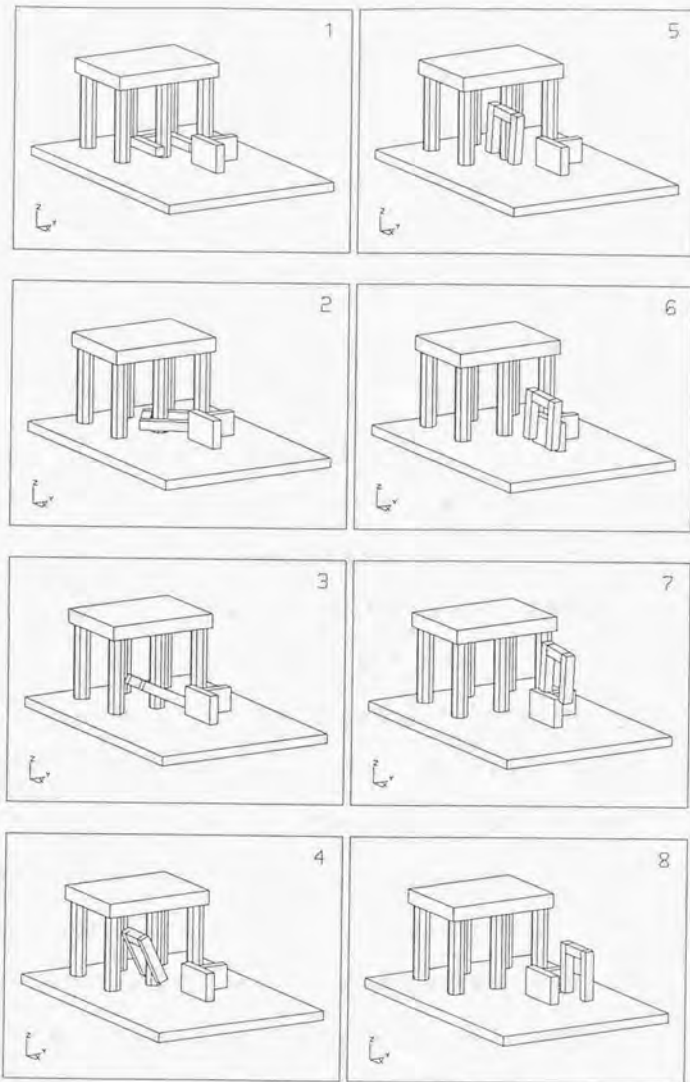


図 5. 1 2 例題 1 2 (U型物体の剛体運動)

5.2 実験結果

以上で示した例題それぞれについて、様々な自由空間算出手法を適用し、実験的に効率を評価した。それぞれの探索手法の効率は、初期点と最終点を連結する自由空間を算出するまでに干渉チェックを実行したセル数、つまり自由空間のセル数と障害物のセル数の合計と比較している。

実行した探索手法は以下に示す5種類(S1, S2, S3, S4, S5)である。最初の方法S1は、初期点から最終点へ向かう単方向の自由空間の算出法であり、各軸に均等にウェイトを置く単一の探索戦略を用いるものである。このときのヒューリスティック関数の係数は $A=3$ 、 $a(i)=(5,5,5,5,5,5)$ である。2番目の方法S2は、S1と同様に単方向(前向き)単一戦略の自由空間の算出であるが、ヒューリスティック関数の係数を $A=3$ に固定した上で、 $a(i), (i=1, \dots, 6)$ を1から9までの範囲で一様乱数によってランダムに設定した場合である。ただし、 $a(i)$ は実数である。3番目の方法S3は、単方向(前向き)複数戦略(4戦略)の自由空間算出である。ここでも、ヒューリスティック関数の係数 $a_t(i) (i=1, \dots, 6, t=1, \dots, 4)$ は1から9までの範囲で一様乱数によってランダムに設定し、すべての $A_t=3 (t=1, \dots, 4)$ に固定した。探索に際して必要な定数は $E_{init}=25$ 、 $Q=20$ にした。4番目の方法S4は、S3と同じであるが探索の向きが後向きになっている。最後の方法S5は、両方向複数戦略(8戦略)の自由空間探索である。定数はそれぞれ $Sf=4$ 、 $Sb=4$ 、 $E_{init}=25$ 、 $Q=20$ である。ヒューリスティック関数の係数 $a_t(i), a_u(i) (i=1, \dots, 6, t=1, \dots, 4, u=1, \dots, 4)$ は1から9までの範囲で一様乱数によってランダムに設定し、すべての A_t と A_u を3に固定した。つまり、前向き4戦略、後ろ向き4戦略の両方向探索である。乱数によってヒューリスティック関数の係数を決定する4つの方法、S2、S3、S4、S5については、300種類の異なる係数の組を生成して自由空間を算出させ、その分布を調べた。干渉チェック(自由空間のセル数と障害物のセル数の合計)

の平均回数、標準偏差 s 、最大値、最小値は表に示してある。また、それぞれの分布を図5.

1.3から図5.2.4にヒストグラムとして示してある。

300という試行回数は、計算時間の観点から実用的な範囲でできるだけ大きい数という
ことで設定した。一般に実験を行なう場合には、正規分布の母集団を仮定し、母集団の平
均を μ 、標準偏差 σ 、 n 個の標本平均を \bar{x} とすると、 \bar{x} は母平均 μ のまわりに

$$\frac{1}{\sqrt{n}}\sigma \quad (5-1)$$

の標準偏差で正規分布することから、実験の精度は試行回数 n について $1/\sqrt{n}$ の形で向上す
ると考えるのが普通である。また、標本の標準偏差を s とすると、

$$\mu = \bar{x} \pm t \frac{s}{\sqrt{n-1}} \quad (5-2)$$

の形で μ を区間推定することができることが知られている。ここで t は t 分布の値であり、実
用的には必要な信頼度と標本数 n に相当する t の値を t 分布表から読みとって用いる。 $n=300$
の場合は $n=\infty$ の値を用いてほとんど問題がなく、この点からも300という試行回数は十
分な値であると考えられる。なお、母集団の標準偏差 σ は以下の式により区間推定できる。

$$\chi^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{\sigma^2} \quad (5-3)$$

ここで χ^2 の値は必要な信頼度に応じて χ^2 分布表から読みとればよい。

表5. 1 自由空間算出のために実行した干渉チェックの回数(例題1)

探索戦略	平均	標準偏差 s	最小	最大
S1 (Fo,S,E)	1811	-	-	-
S2 (Fo,S,R)	4255	6225	833	59788
S3 (Fo,M,R)	2730	875	1465	6972
S4 (Ba,M,R)	8285	3090	3326	24061
S5 (Bi,M,R)	6210	4236	1992	34064

探索戦略は以下の記号で表示してある。

Fo: 単方向前向き探索

Ba: 単方向後向き探索

Bi: 両方向探索

S: 単一戦略による探索

M: 複数戦略による探索

E: 各軸に均等なウェイト付けする戦略 ($a(i) = (5, 5, 5, 5, 5)$)

R: 乱数によって定めたウェイト付けをした戦略

例えば S1 (Fo,S,E) は単一戦略による単方向探索で、 $a(i) = (5, 5, 5, 5, 5)$ であるものを意味する。

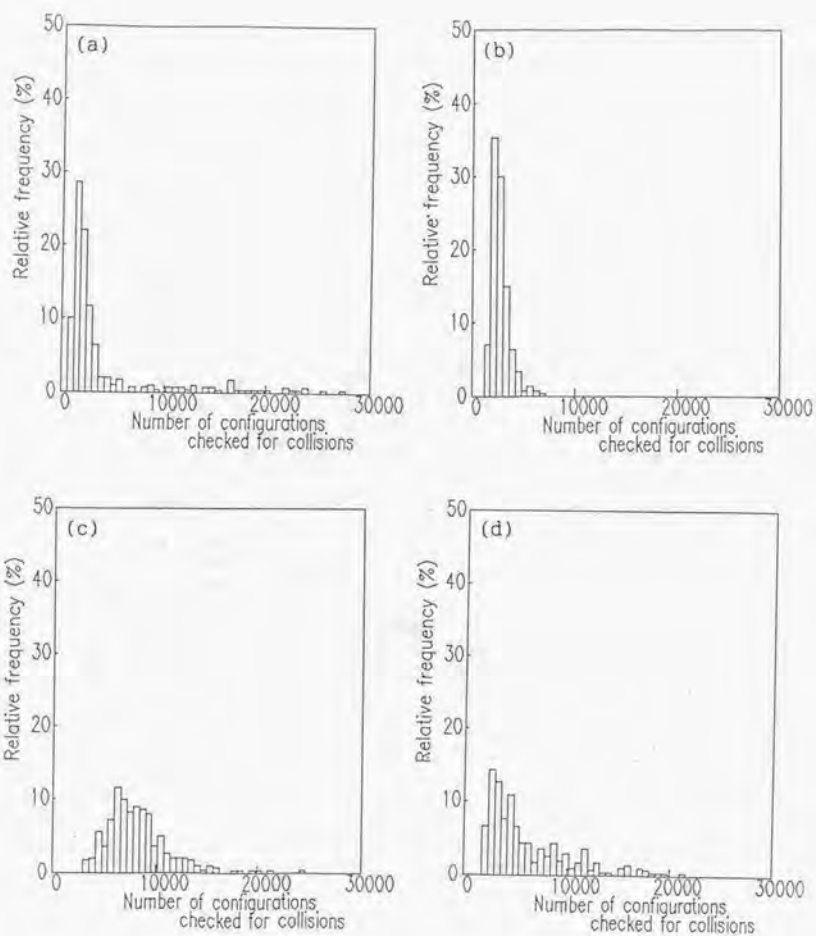


図5. 13 例題1 (PUMA型マニピュレータ)における乱数による試行結果 (a)S2, (b)S3, (c)S4, (d)S5

表5. 2 自由空間算出のために実行した干渉チェックの回数(例題2)

探索戦略	平均	標準偏差 s	最小	最大
S1 (Fo,S,E)	972	-	-	-
S2 (Fo,S,R)	1958	3231	567	31097
S3 (Fo,M,R)	1866	604	1153	6385
S4 (Ba,M,R)	4460	2358	1812	16502
S5 (Bi,M,R)	2433	294	1584	3621

探索戦略は以下の記号で表示してある。

Fo: 単方向前向き探索

Ba: 単方向後向き探索

Bi: 両方向探索

S: 単一戦略による探索

M: 複数戦略による探索

E: 各軸に均等なウェイト付けする戦略 ($a(i) = (5, 5, 5, 5, 5)$)

R: 乱数によって定めたウェイト付けをした戦略

例えば S1 (Fo,S,E) は単一戦略による単方向探索で、 $a(i) = (5, 5, 5, 5, 5)$ であるものを意味する。

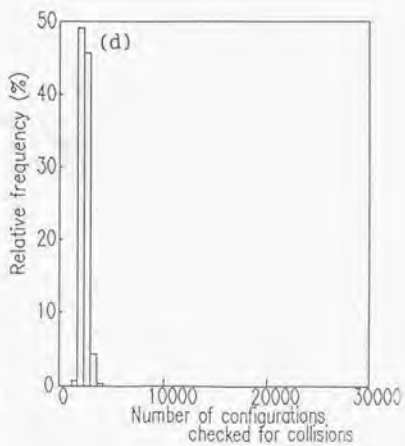
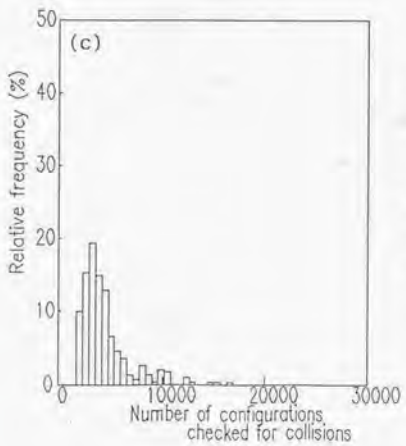
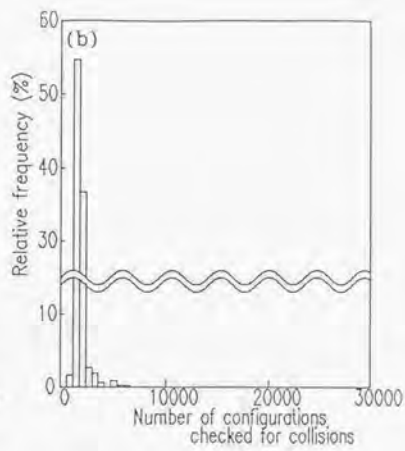
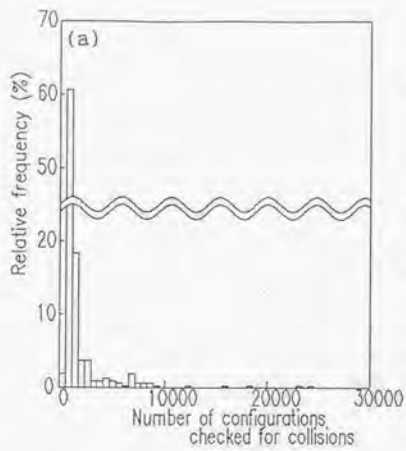


図5. 14 例題2 (PUMA型マニピュレータ)における乱数による試行結果 ((a)S2, (b)S3, (c)S4, (d)S5)

表5.3 自由空間算出のために実行した干渉チェックの回数(例題3)

探索戦略	平均	標準偏差 s	最小	最大
S1 (Fo,S,E)	20822	-	-	-
S2 (Fo,S,R)	8660	3932	1769	27073
S3 (Fo,M,R)	9775	4893	2280	31568
S4 (Ba,M,R)	*	-	-	-
S5 (Bi,M,R)	9623	2881	4700	22245

探索戦略は以下の記号で表示してある。

Fo: 単方向前向き探索

Ba: 単方向後向き探索

Bi: 両方向探索

S: 単一戦略による探索

M: 複数戦略による探索

E: 各軸に均等なウエイト付けする戦略 ($a(i) = (5, 5, 5, 5, 5, 5)$)

R: 乱数によって定めたウエイト付けをした戦略

例えば S1 (Fo,S,E) は単一戦略による単方向探索で、 $a(i) = (5, 5, 5, 5, 5, 5)$ であるものを意味する。

*: 30回の試行を実行したが、いずれも14万回以下の干渉チェックでは経路を見つけることができなかった。

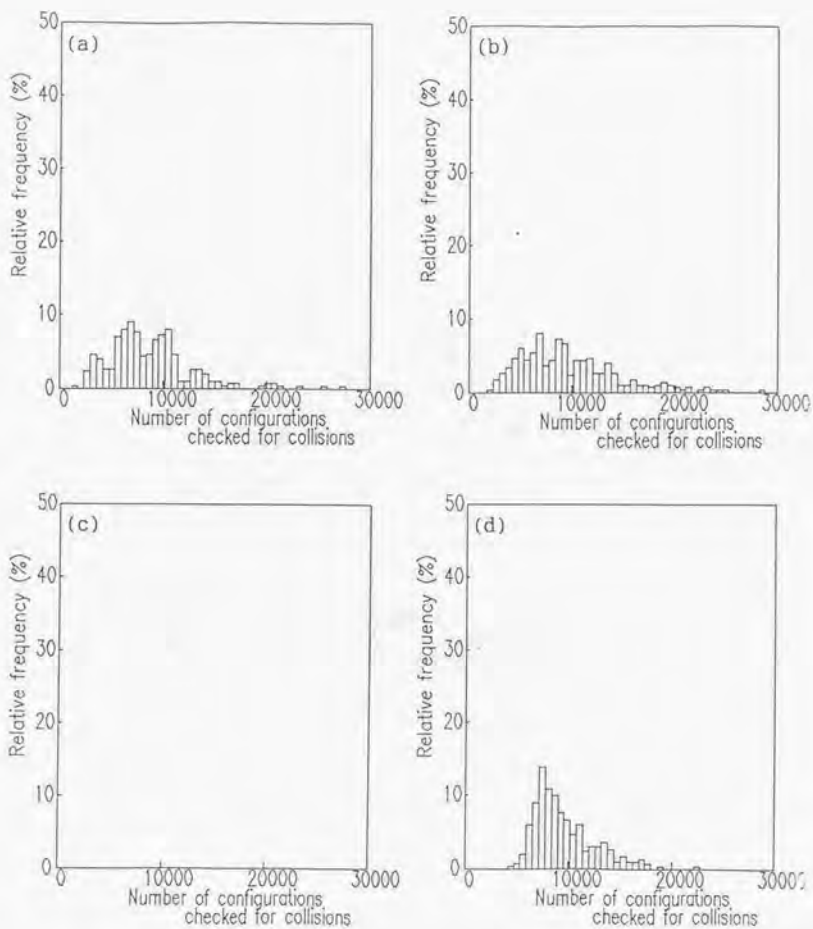


図5. 15 例題3 (PUMA型マニピュレータ)における乱数による試行結果 ((a)S2, (b)S3, (c)S4, (d)S5)

表5. 4 自由空間算出のために実行した干渉チェックの回数(例題4)

探索戦略	平均	標準偏差 s	最小	最大
S1 (Fo,S,E)	20891	-	-	-
S2 (Fo,S,R)	14549	9596	1867	74084
S3 (Fo,M,R)	13032	8144	2333	45098
S4 (Ba,M,R)	*	-	-	-
S5 (Bi,M,R)	17145	7647	4912	46316

探索戦略は以下の記号で表示してある。

Fo: 単方向前向き探索

Ba: 単方向後向き探索

Bi: 両方向探索

S: 単一戦略による探索

M: 複数戦略による探索

E: 各軸に均等なウエイト付けする戦略 ($a(i) = (5, 5, 5, 5, 5)$)

R: 乱数によって定めたウエイト付けをした戦略

例えば S1 (Fo,S,E) は単一戦略による単方向探索で、 $a(i) = (5, 5, 5, 5, 5)$ であるものを意味する。

*: 30回の試行を実行したが、いずれも12万回以下の干渉チェックでは経路を見つけることができなかった。

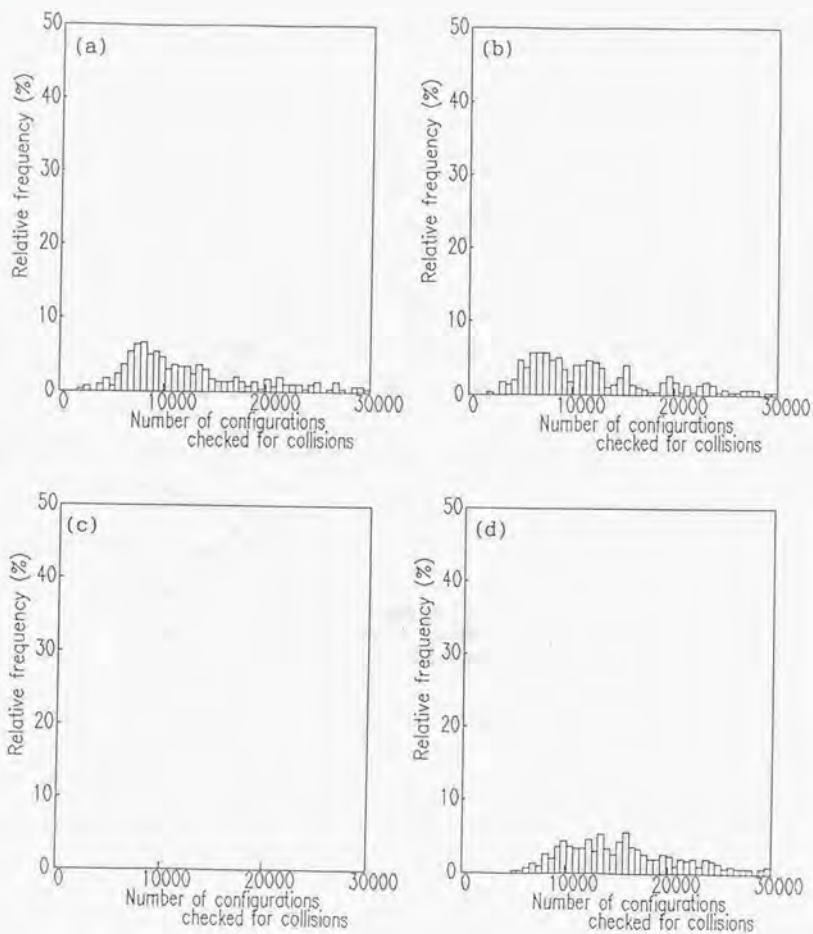


図5. 16 例題4 (PUMA型マニピュレータ)における乱数による試行結果 (a)S2, (b)S3, (c)S4, (d)S5)

表5. 5 自由空間算出のために実行した干渉チェックの回数(例題5)

探索戦略	平均	標準偏差 s	最小	最大
S1 (Fo,S,E)	533	-	-	-
S2 (Fo,S,R)	582	84	507	942
S3 (Fo,M,R)	1842	115	1432	2117
S4 (Ba,M,R)	1829	138	1508	2358
S5 (Bi,M,R)	2537	931	1077	6558

探索戦略は以下の記号で表示してある。

Fo: 単方向前向き探索

Ba: 単方向後向き探索

Bi: 両方向探索

S: 単一戦略による探索

M: 複数戦略による探索

E: 各軸に均等なウエイト付けする戦略 ($a(i) = (5, 5, 5, 5, 5, 5)$)

R: 乱数によって定めたウエイト付けをした戦略

例えば S1 (Fo,S,E) は単一戦略による単方向探索で、 $a(i) = (5, 5, 5, 5, 5, 5)$ であるものを意味する。

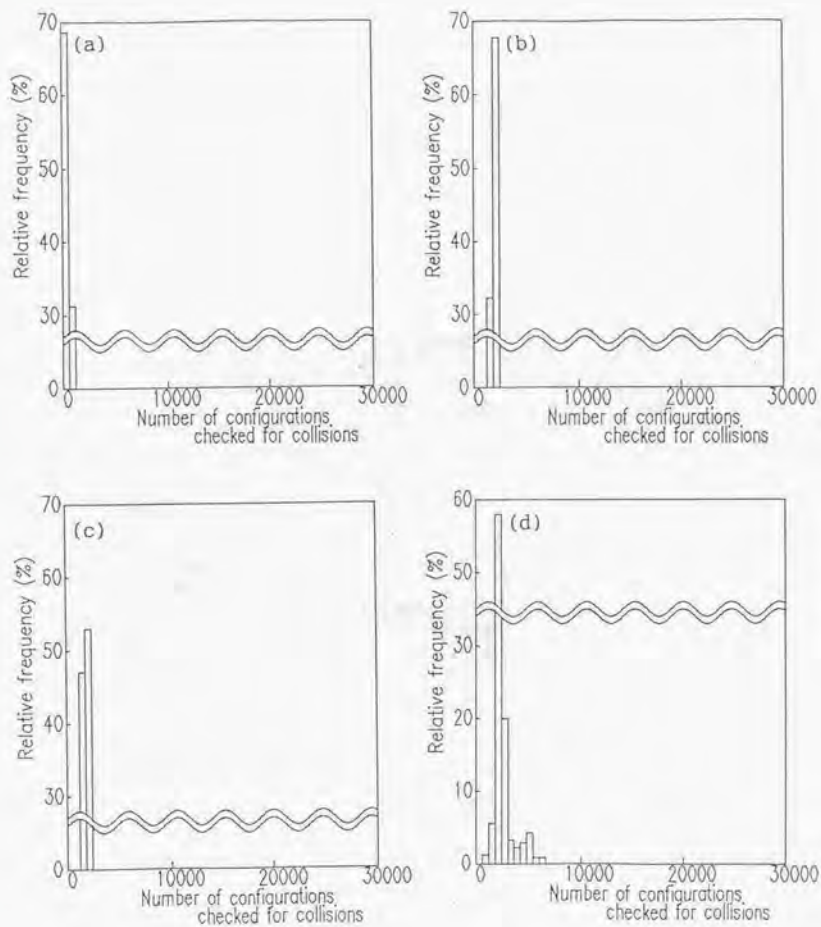


図5. 17 例題5 (JEMRMS) における乱数による試行結果 (a)S2, (b)S3, (c)S4, (d)S5

表5. 6 自由空間算出のために実行した干渉チェックの回数(例題6)

探索戦略	平均	標準偏差 s	最小	最大
S1 (Fo,S,E)	1193	-	-	-
S2 (Fo,S,R)	1664	1287	801	17914
S3 (Fo,M,R)	2527	520	1437	5237
S4 (Ba,M,R)	2346	3301	1288	34809
S5 (Bi,M,R)	6268	5627	1626	33815

探索戦略は以下の記号で表示してある。

Fo: 単方向前向き探索

Ba: 単方向後向き探索

Bi: 両方向探索

S: 単一戦略による探索

M: 複数戦略による探索

E: 各軸に均等なウェイト付けする戦略 ($a(i) = (5, 5, 5, 5, 5)$)

R: 乱数によって定めたウェイト付けをした戦略

例えば S1 (Fo,S,E) は単一戦略による単方向探索で、 $a(i) = (5, 5, 5, 5, 5)$ であるものを意味する。

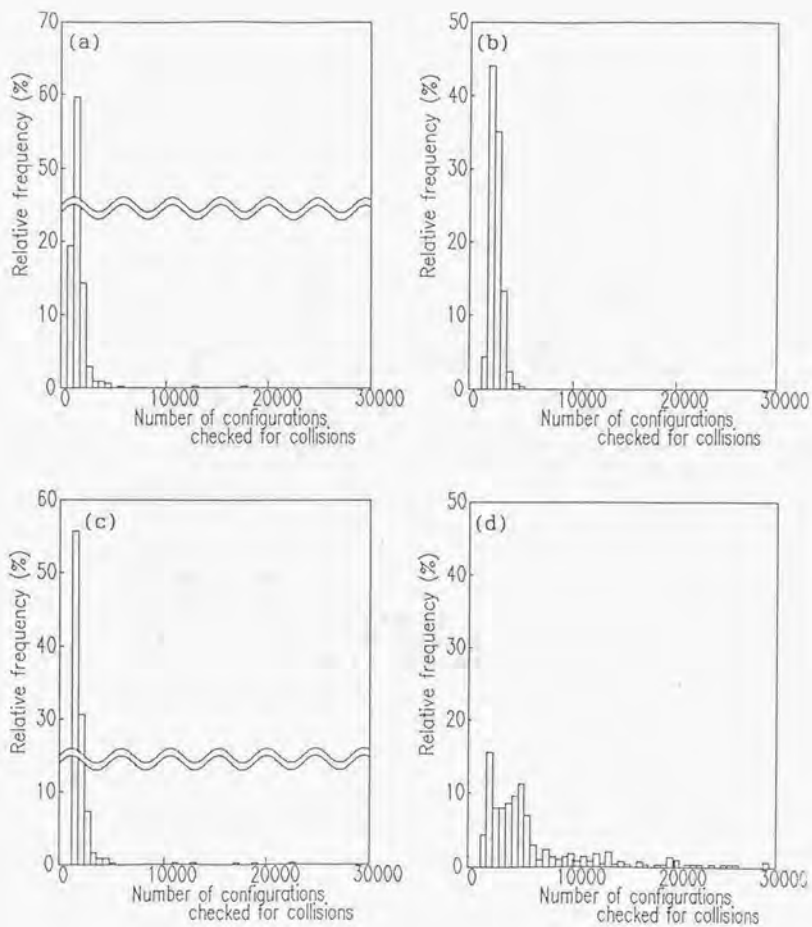


図5.18 例題6 (JEMRMS) における乱数による試行結果 (a)S2, (b)S3, (c)S4, (d)S5

表5.7 自由空間算出のために実行した干渉チェックの回数(例題7)

探索戦略	平均	標準偏差 s	最小	最大
S1 (Fo,S,E)	4196	-	-	-
S2 (Fo,S,R)	20372	25783	822	168806
S3 (Fo,M,R)	9579	8727	2101	60630
S4 (Ba,M,R)	4384	902	2981	10174
S5 (Bi,M,R)	17010	17946	3169	184498

探索戦略は以下の記号で表示してある。

Fo: 単方向前向き探索

Ba: 単方向後向き探索

Bi: 両方向探索

S: 単一戦略による探索

M: 複数戦略による探索

E: 各軸に均等なウエイト付けする戦略 ($a(i) = (5, 5, 5, 5, 5, 5)$)

R: 乱数によって定めたウエイト付けをした戦略

例えば S1 (Fo,S,E) は単一戦略による単方向探索で、 $a(i) = (5, 5, 5, 5, 5, 5)$ であるものを意味する。

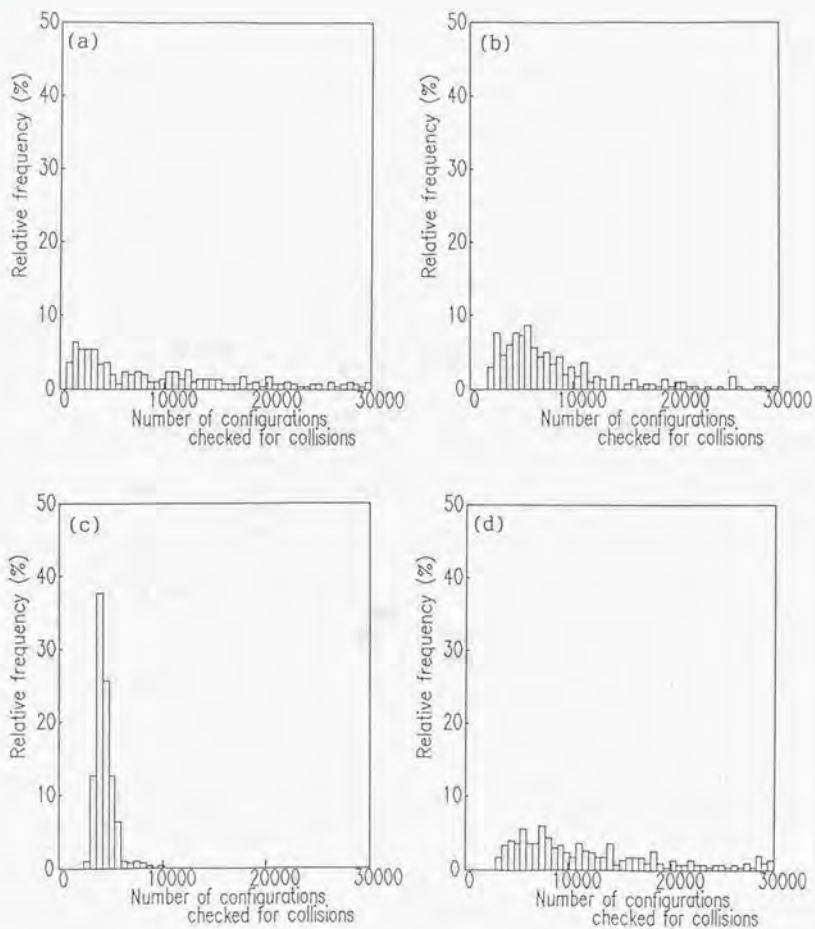


図5. 19 例題7 (JEMRMS) における乱数による試行結果 ((a)S2, (b)S3, (c)S4, (d)S5)

表5. 8 自由空間算出のために実行した干渉チェックの回数(例題8)

探索戦略	平均	標準偏差 s	最小	最大
S1 (Fo,S,E)	743	-	-	-
S2 (Fo,S,R)	814	97	678	1343
S3 (Fo,M,R)	1438	205	1023	2198
S4 (Ba,M,R)	30257	15124	5205	85972
S5 (Bi,M,R)	1938	667	1277	12167

探索戦略は以下の記号で表示してある。

Fo: 単方向前向き探索

Ba: 単方向後向き探索

Bi: 両方向探索

S: 単一戦略による探索

M: 複数戦略による探索

E: 各軸に均等なウエイト付けする戦略 ($a(i) = (5, 5, 5, 5, 5, 5)$)

R: 乱数によって定めたウエイト付けをした戦略

例えば S1 (Fo,S,E) は単一戦略による単方向探索で、 $a(i) = (5, 5, 5, 5, 5, 5)$ であるものを意味する。

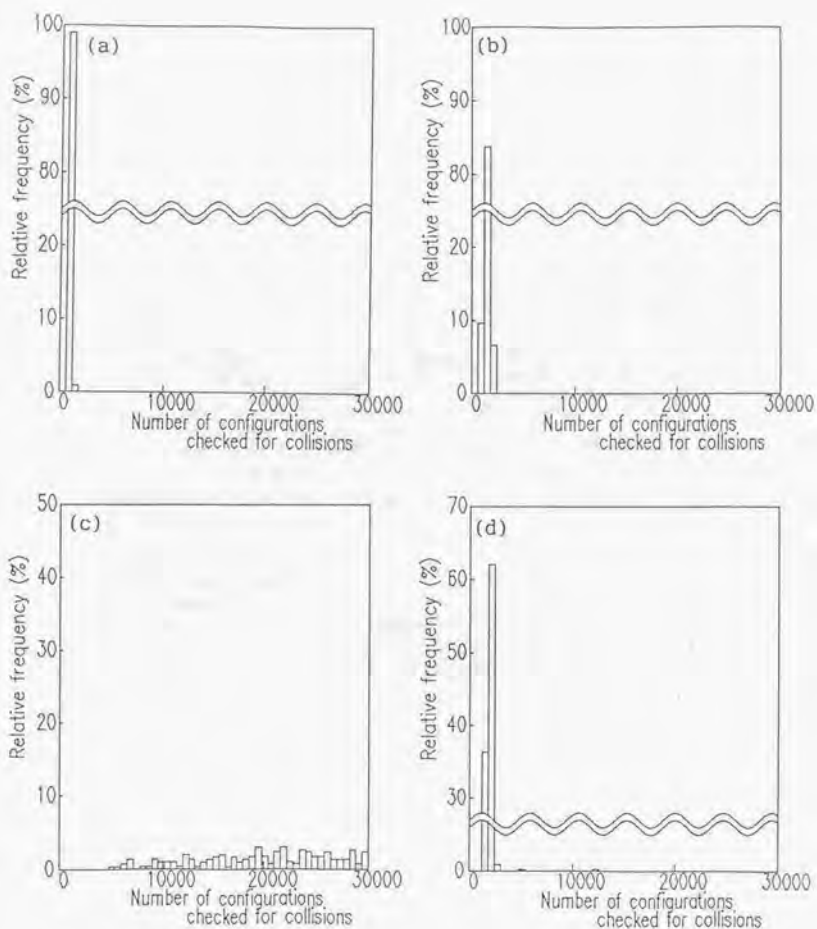


図5.20 例題8 (2アーム協調動作) における乱数による試行結果 (a)S2, (b)S3, (c)S4, (d)S5

表5. 9 自由空間算出のために実行した干渉チェックの回数(例題9)

探索戦略	平均	標準偏差 s	最小	最大
S1 (Fo,S,E)	956	-	-	-
S2 (Fo,S,R)	1551	1935	649	19219
S3 (Fo,M,R)	1966	409	1129	3573
S4 (Ba,M,R)	30150	15320	4983	87374
S5 (Bi,M,R)	4046	6458	1653	49339

探索戦略は以下の記号で表示してある。

Fo: 単方向前向き探索

Ba: 単方向後向き探索

Bi: 両方向探索

S: 単一戦略による探索

M: 複数戦略による探索

E: 各軸に均等なウエイト付けする戦略 ($a(i) = (5, 5, 5, 5, 5, 5)$)

R: 乱数によって定めたウエイト付けをした戦略

例えば S1 (Fo,S,E) は単一戦略による単方向探索で、 $a(i) = (5, 5, 5, 5, 5, 5)$ であるものを意味する。

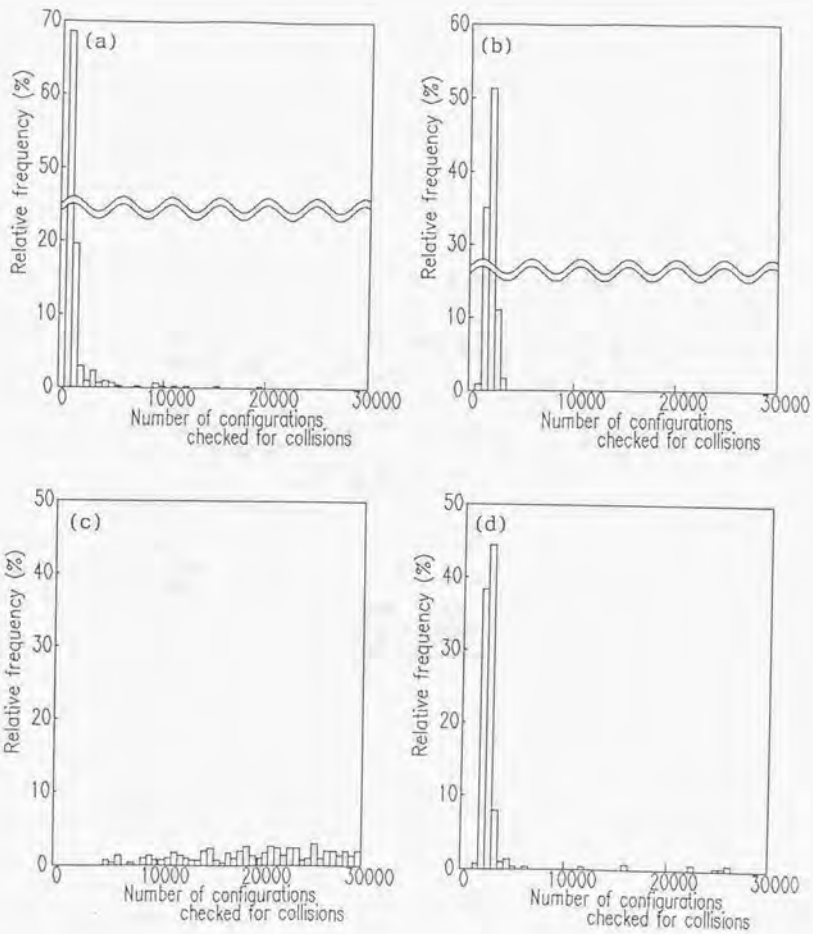


図5. 21 例題9 (2アーム協調動作) における乱数による試行結果 ((a)S2, (b)S3, (c)S4, (d)S5)

表5. 10 自由空間算出のために実行した干渉チェックの回数(例題10)

探索戦略	平均	標準偏差 s	最小	最大
S1 (Fo,S,E)	2502	-	-	-
S2 (Fo,S,R)	2344	1043	772	5605
S3 (Fo,M,R)	2704	976	1370	5797
S4 (Ba,M,R)	30262	15116	5205	87138
S5 (Bi,M,R)	3400	1018	1852	8614

探索戦略は以下の記号で表示してある。

Fo: 単方向前向き探索

Ba: 単方向後向き探索

Bi: 両方向探索

S: 単一戦略による探索

M: 複数戦略による探索

E: 各軸に均等なウェイト付けする戦略 ($a(i) = (5, 5, 5, 5, 5)$)

R: 乱数によって定めたウェイト付けをした戦略

例えば S1 (Fo,S,E) は単一戦略による単方向探索で、 $a(i) = (5, 5, 5, 5, 5)$ であるものを意味する。

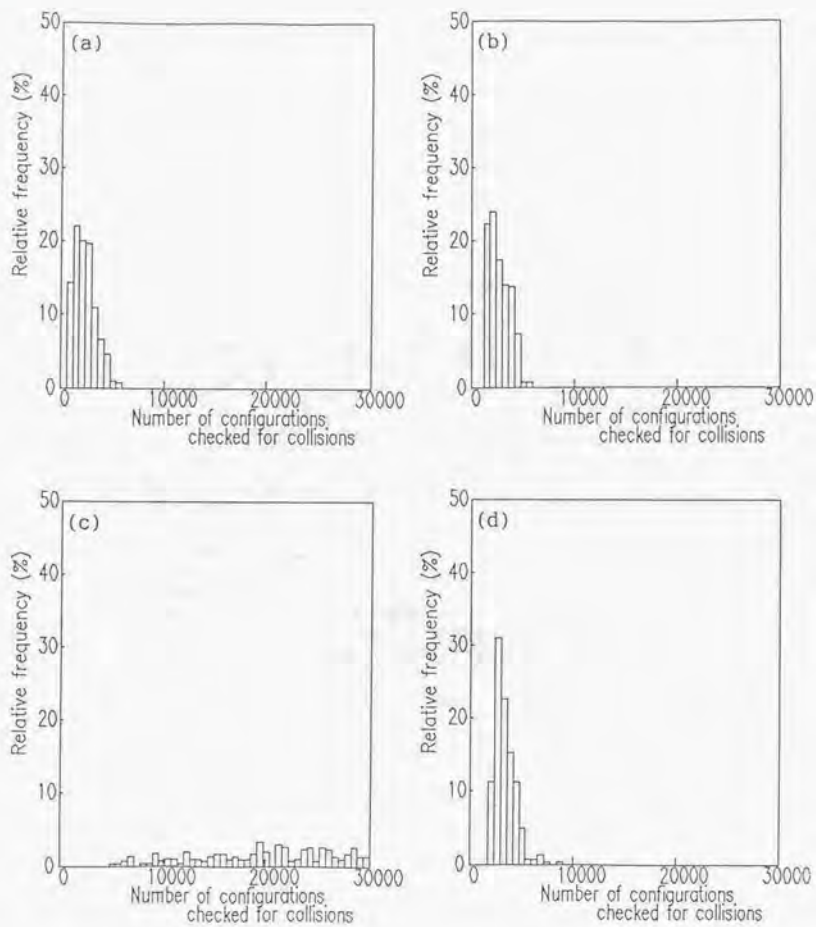


図5.22 例題10 (2アーム協調動作) における乱数による試行結果 ((a)S2, (b)S3, (c)S4, (d)S5)

表5. 11 自由空間算出のために実行した干渉チェックの回数(例題11)

探索戦略	平均	標準偏差 s	最小	最大
S1 (Fo,S,E)	2098	-	-	-
S2 (Fo,S,R)	3495	5217	790	44426
S3 (Fo,M,R)	2561	1111	1274	7837
S4 (Ba,M,R)	*	-	35815	-
S5 (Bi,M,R)	3566	1517	2002	17547

探索戦略は以下の記号で表示してある。

Fo: 単方向前向き探索

Ba: 単方向後向き探索

Bi: 両方向探索

S: 単一戦略による探索

M: 複数戦略による探索

E: 各軸に均等なウェイト付けする戦略 ($a(i) = (5, 5, 5, 5, 5)$)

R: 乱数によって定めたウェイト付けをした戦略

例えば S1 (Fo,S,E) は単一戦略による単方向探索で、 $a(i) = (5, 5, 5, 5, 5)$ であるものを意味する。

*: 30回の試行を実行したが、17万回以下の干渉チェックで経路を見つかることができたのは5回だけであった。

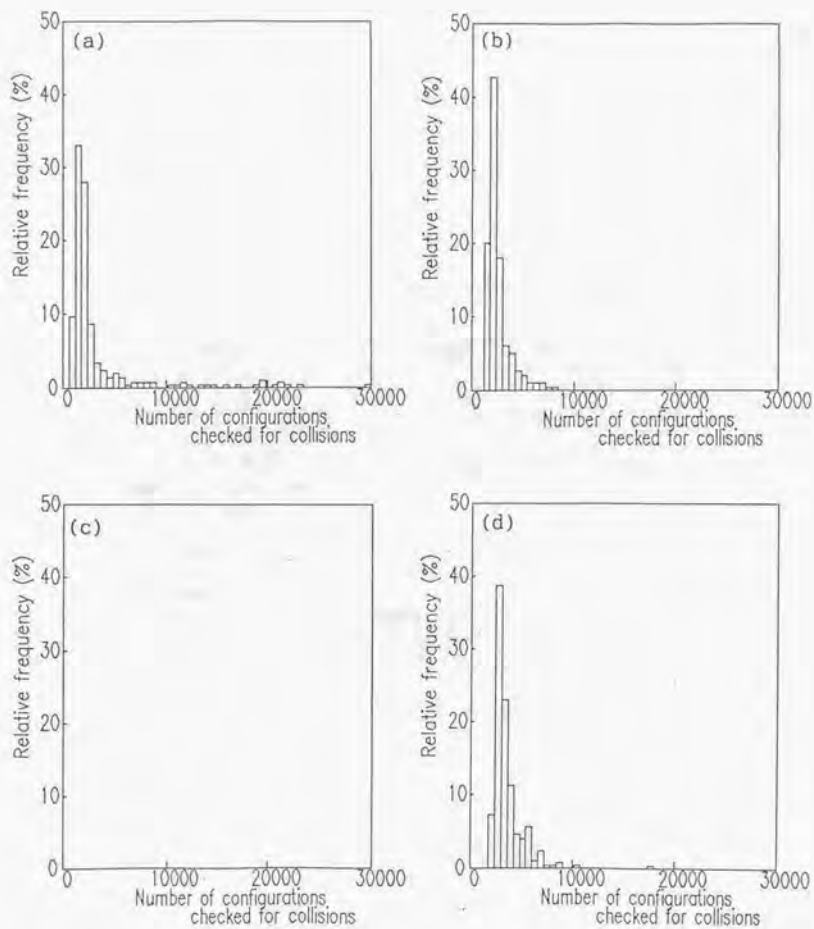


図5. 23 例題11 (U型物体の剛体運動) における乱数による試行結果 ((a)S2, (b)S3, (c)S4, (d)S5)

表5. 12 自由空間算出のために実行した干渉チェックの回数(例題12)

探索戦略	平均	標準偏差 s	最小	最大
S1 (Fo,S,E)	6099	-	-	-
S2 (Fo,S,R)	9414	7009	1878	47989
S3 (Fo,M,R)	8824	4540	2497	27555
S4 (Ba,M,R)	*	-	-	-
S5 (Bi,M,R)	11919	7916	3733	74455

探索戦略は以下の記号で表示してある。

Fo: 単方向前向き探索

Ba: 単方向後向き探索

Bi: 両方向探索

S: 単一戦略による探索

M: 複数戦略による探索

E: 各軸に均等なウェイト付けする戦略 ($a(i) = (5, 5, 5, 5, 5)$)

R: 乱数によって定めたウェイト付けをした戦略

例えば S1 (Fo,S,E) は単一戦略による単方向探索で、 $a(i) = (5, 5, 5, 5, 5)$ であるものを意味する。

*: 30回の試行を実行したが、いずれも12万回以下の干渉チェックでは経路を見つけることができなかった。

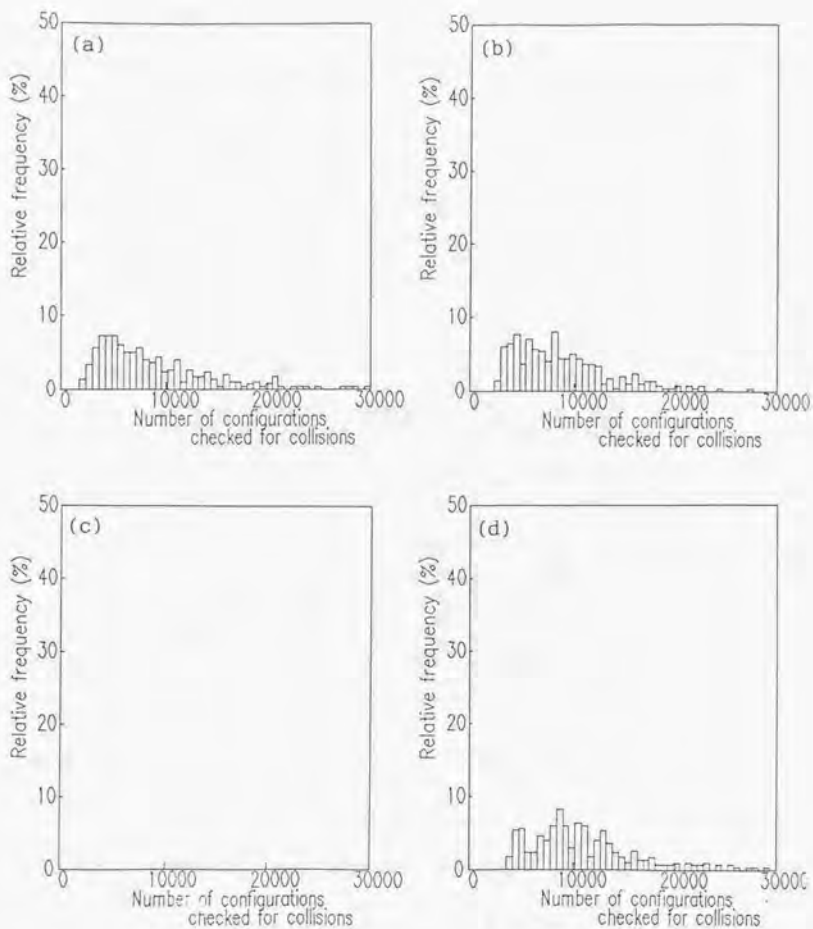


図5. 24 例題1 2 (U型物体の剛体運動) における乱数による試行結果 ((a)S2, (b)S3, (c)S4, (d)S5)

5.3 考察

以下では、以上で紹介した実験結果について、いくつかの観点から考察する。また、これを通じて本論文で提案したそれぞれの手法の効果についても考える。さらに、移動の自由度の増加にともなって本手法の計算量がどのように変化するかについても触れる。

5.3.1 係数 $a(i)$ の影響

まず最初に各軸にウェイトづけする係数の影響について考える。

表5.13は探索方法S2の結果をまとめたものである。

表5.13 探索方法S2の結果

例題	平均	標準偏差 s	最小	最大
Ex. 1	4255	6225	833	59788
Ex. 2	1958	3231	567	31097
Ex. 3	8660	3932	1769	27073
Ex. 4	14549	9596	1867	74084
Ex. 5	582	84	507	942
Ex. 6	1664	1287	801	17914
Ex. 7	20372	25783	822	168806
Ex. 8	814	97	678	1343
Ex. 9	1551	1935	649	19219
Ex.10	2344	1043	772	5605
Ex.11	3495	5217	790	44426
Ex.12	9414	7009	1878	47989

この結果から分かるように、コンフィギュレーション空間の各軸にウェイトづけする係数 $a(i)$ を変えることによって、探索に必要な計算量が大きく変化する。これは、本論文で提案したヒューリスティック関数を用いれば、係数を変化させるだけで容易に異なる性質を持つ探索戦略を定義できることを示している。また、解こうとする問題に合わせて適切に係数を設定すれば、かなり少ない計算量で経路を求めることができる可能性がある。

さらに、6次元配列状に区切ったコンフィギュレーション空間を用いた探索であることを踏

まえて、経路探索のために実行した干渉チェックの回数を見ると、すべての例題において、探索の際に参照したのは、コンフィギュレーション空間の極く一部であることが分かる。このような結果は、6次元コンフィギュレーション空間のように次元が高くなっても、個々の経路探索を実行する際は、コンフィギュレーション空間全体を参照する必要のないことを示している。これから、探索手法を工夫すれば、コンフィギュレーション空間の次元が高くなっても、計算量の爆発的な増加を抑えて、効率よく経路探索を実行できる可能性があることが、実験的にも確認されたことになる。

ただし、適切に係数を設定すれば必要な計算量を大幅に抑えることができるが、探索を実行して自由空間の状況が分かるまで、どのような係数の設定がよいのかは決定できない。例えば、探索方法S1はS2の一例であり、問題によっては探索方法S2の平均値より少ない干渉チェック回数で必要な自由空間が算出できている。ただこれは結果論であり、探索を実行する前にS1が平均より効率がよいかどうかを知ることは一般的には不可能である。また、多くの試行の中には、著しく多くの干渉チェックを実行する必要があったものもあり、最大計算量は非常に大きいことが予想される。これは、障害物回避動作計画の問題が、自由度についてPSPACE困難であるとされるのに対応し、単一の探索戦略を用いる場合には、非常に多くの計算を実行しなければならない危険性を覚悟しなければならない。

5. 3. 2 複数探索戦略を用いる効果

次に、単一戦略による自由空間算出と、複数探索戦略による自由空間算出を比較してみる。表5. 14は、探索手法S2とS3を比較するために、S2の場合の干渉チェック回数の少ない順にならべかえて、 t 分布、 χ^2 分布を用いて区間推定した95%信頼区間をあわせてまとめている。

表5. 14 探索方法S2とS3の比較

例題	探索戦略	平均(95%信頼区間)	標準偏差(95%信頼区間)
Ex. 5	x S2	582 (572.5 - 591.5)	84 (77.8 - 91.4)
	S3	1842 (1829.0 - 1855.0)	115 (106.6 - 125.1)
Ex. 8	x S2	814 (803.0 - 825.0)	97 (89.9 - 105.5)
	S3	1438 (1414.8 - 1461.2)	205 (190.0 - 223.0)
Ex. 9	△ S2	1551 (1331.7 - 1770.3)	1935 (1793.0 - 2105.2)
	S3	1966 (1919.6 - 2012.4)	409 (379.0 - 445.0)
Ex. 6	△ S2	1664 (1518.1 - 1809.9)	1287 (1192.6 - 1400.2)
	S3	2527 (2468.1 - 2585.9)	520 (481.8 - 565.7)
Ex. 2	⊗ S2	1958 (1591.8 - 2324.2)	3231 (2993.9 - 3515.2)
	S3	1866 (1797.5 - 1934.5)	604 (559.7 - 657.1)
Ex.10	△ S2	2344 (2225.8 - 2462.2)	1043 (966.5 - 1134.7)
	S3	2704 (2593.4 - 2814.6)	976 (904.4 - 1061.9)
Ex.11	○ S2	3495 (2903.7 - 4086.3)	5217 (4834.2 - 5675.9)
	S3	2561 (2435.1 - 2686.9)	1111 (1029.5 - 1208.7)
Ex. 1	○ S2	4255 (3549.4 - 4960.6)	6225 (5768.3 - 6772.6)
	S3	2730 (2630.8 - 2829.2)	875 (810.8 - 952.0)
Ex. 3	x S2	8660 (8214.3 - 9105.7)	3932 (3643.5 - 4277.9)
	S3	9775 (9220.4 - 10329.6)	4893 (4534.0 - 5323.4)
Ex.12	⊗ S2	9414 (8619.5 - 10208.5)	7009 (6494.7 - 7625.5)
	S3	8824 (8309.4 - 9338.6)	4540 (4206.9 - 4939.3)
Ex. 4	⊗ S2	14549 (13461.3 - 15636.7)	9596 (8891.9 - 10440.1)
	S3	13032 (12108.9 - 13955.1)	8144 (7546.4 - 8860.4)
Ex. 7	○ S2	20372 (17449.5 - 23294.5)	25783 (23891.2 - 28050.9)
	S3	9579 (8589.8 - 10568.2)	8727 (8086.7 - 9494.6)

複数探索戦略を用いることにより

- x: 平均干渉チェック回数、標準偏差ともに悪化した例
- △: 平均干渉チェック回数は増加したが、標準偏差が減少した例
- ⊗: 平均干渉チェック回数、標準偏差ともに改善されたが、95%信頼区間の値からは断言できない例
- : 95%信頼区間の値から平均干渉チェック回数、標準偏差ともに改善されたことが明白な例

複数探索戦略を用いたことによる効果としては、例外的にかなり多くの干渉チェックを要する場合がなくなっている点あげられる。これは、ほとんどの例で探索手法S2よりもS3の方が、標準偏差 s の値がかなり小さくなっていることに表れている。これは、解こうとする問題に対して不適切な係数 $a(i)$ による探索戦略が正しく識別され、その実行が制限された結果である。また、ここで用いた例題のレベルの複雑さにおいては、実空間がかなり複雑であっても、6次元のコンフィギュレーション空間においては、それほど障害物が密になっておらず、4つの戦略を同時に実行すれば、それらの中の少なくとも1つは容易に経路を発見することができる程度に自由空間が広がっていると考えられる。

一方、干渉チェックの最小値がS2よりもS3の方が大きいことから分かるように、複数探索戦略を実行するに際して、努力配分を決定するために多少余分な計算が必要になることも事実である。しかし、結果を平均値として見た場合には、一般的に言って複数探索戦略を用いた方が効率がよい。

より細かく考察すると、問題が単純で容易に解けるときには、複数の戦略を用いるためのオーバーヘッドの割合がおおきくなり効率の改善があまり見込めないが、問題が複雑になると、複数の戦略の中から効率のよいものを選択して優先的に実行する効果が現れると思われる。実際、表5.14でも必要とする干渉チェック回数がおおきいものほど複数探索戦略利用の効果が顕著である。これを定量的に評価するために、S2の平均干渉チェック回数を x_i 、S3の平均干渉チェック回数を y_i とし、

$$\alpha_i = \log \frac{x_i}{y_i} \quad (5-4)$$

を定める。複数戦略により効率があがるほど α_i は大きくなり、 α_i と正の相関があれば、複数戦略の利用は特に複雑な問題で効果があることを結論づけることができると考えられる。ここで対数を取ったのは、効率が2倍向上したのと半分になったのを同等に扱うためであ

る。ここでは、12の例題があるので、標本の相関係数 r は、

$$r = \frac{\sum_{i=1}^{12} (x_i - \bar{x})(x_i - \bar{x})}{\sqrt{\sum_{i=1}^{12} (x_i - \bar{x})^2 \times \sum_{i=1}^{12} (x_i - \bar{x})^2}} \quad (5-5)$$

となる。この値を計算すると $r=0.65$ となる。この値から母集団の相関係数 ρ を簡易的に求めるには、図5.25のようなグラフを用いる[大村80]。このグラフでは標本数ごとに2つの曲線があり、それぞれが ρ の上限と下限を与える。標本数が12であることを注意してこのグラフを読みとると、 $r=0.65$ の場合、 ρ の95%の信頼区間は

$$0.13 < \rho < 0.88 \quad (5-6)$$

となる。これから、 x_1 と x_2 に正の相関があることが確認され、複数探索戦略を用いることにより、特に複雑な問題で効率が改善されることが明かになった。実用的な観点からは、単純な問題での効率の低下は、もともとの計算時間が短いためにそれほど不利とはならない。一方複雑な問題では、計算時間の短縮が大きな課題であり、複数戦略の利用による実効的な効果は大きいといえる。

ただし、平均値として見た場合に、複数戦略の効率と単一戦略の効率に、大幅な改善が見られないのは、探索努力配分のために用いた効率の評価値の信頼性がそれほど高くないのが原因であると考えられる。この手法においては、探索戦略の評価値もヒューリスティクスの一つであり、より信頼度の高い評価法を用いればさらに効率向上が期待できる。

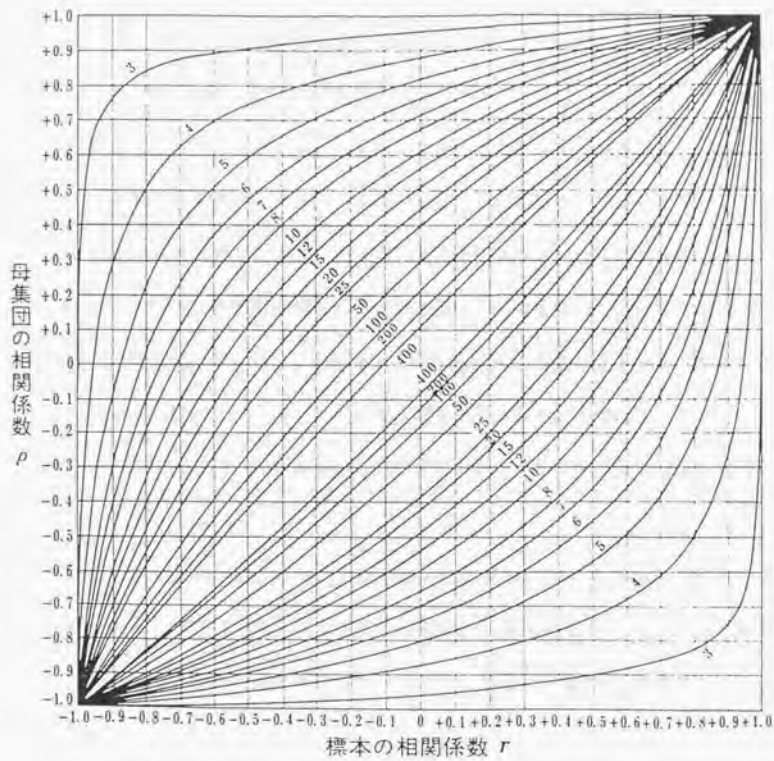


図5、25 母集団の相関係数を区間推定するためのグラフ [大村 80]

5.3.3 両方向探索の効果

一方、探索の方向が効率にどのような影響を与えるかは探索手法のS3とS4の結果から読み取ることができる。前向き探索、後ろ向き探索のどちらが効率がよいかがあらかじめ分かっているならば、その方向の単方向探索を行うことでかなり少ない干渉チェック回数で必要な自由空間を算出することができる。しかし、一般には3次元実空間とコンフィギュレーション空間との対応関係は複雑で、3次元実空間における障害物の状況からどちらの探索方向がよいかを判断するのは困難である。例えば、PUMA型マニピュレータの例題4では、初期姿勢、最終姿勢の両方が障害物に囲まれた状況であり、前向き探索、後ろ向き探索に大きな違いは無いように思われる。しかし、実験結果では後ろ向き探索の方が前向き探索よりはるかに多くの計算を必要としている。このような観点から、単方向の探索は、実行すべき探索の方向が分かっているような特定の場合には有効であるが、あらゆる問題に適用可能な一般性は無い。

以上の議論からも明かなように、あらゆる問題に適用できるようにするためには両方向探索戦略は必要不可欠である。探索手法S5の結果は、複数戦略を用いた両方向探索による手法は、探索の方向を特に指定することなく少ない計算量で必要な自由空間を算出することができることを示している。

さらに、問題によっては両方向探索手法を用いなければ実的な計算時間で解を求めることができないことが実験結果から推察できる。例えば、コンフィギュレーション空間において初期点と最終点が両方も例題3のように障害物によって囲まれている場合を考える。このような場合、前向き探索も後ろ向き探索も障害物を避けてその目標点へ近づくためにかなり多くの計算を必要とし、単方向探索では実的な計算時間で経路を算出することができない。つまり、前向き探索と後ろ向き探索の両方が、例題3の後ろ向き探索のような状況になっ

しまふ。一方、両方向探索戦略を用いると、初期点から最終点へ向かって離れる動きと、最終点から初期点へ向かって離れる動きが同時に探索され、これら両方によって算出された自由空間が比較的少ない計算量で互いに連結し、必要な自由空間を限定して求めることが可能になる。このように初期点、最終点の両方が障害物に囲まれているような非常に複雑な場合でも、両方向探索戦略を用いることで効率的に解くことができる。

表 5. 1 5 探索方法 S5 の結果

例題	平均	標準偏差 s	最小	最大
Ex. 1	6210	4236	1992	34064
Ex. 2	2433	294	1584	3621
Ex. 3	9623	2881	4700	22245
Ex. 4	17145	7647	4912	46316
Ex. 5	2537	931	1077	6558
Ex. 6	6268	5627	1626	33815
Ex. 7	17010	17946	3169	184498
Ex. 8	1938	667	1277	12167
Ex. 9	4046	6458	1653	49339
Ex.10	3400	1018	1852	8614
Ex.11	3566	1517	2002	17547
Ex.12	11919	7916	3733	74455

表 5. 1 5 は探索方法 S5 の結果をまとめたものである。両方向の複数探索戦略による自由空間算出方法では、性質の異なる 12 の例題すべてにおいて、6次元コンフィギュレーション空間のデータ量と比較して極めて限定された部分を参照するのみで、移動経路を発見することに成功している。この点から、本論文の提案する方法は、現実の問題において自由度が高い場合においても十分に利用できる効率と一般性を持つことが実証できた。

5. 3. 4 自由度の影響

これまでの議論では、コンフィギュレーション空間の次元を6に固定し、6自由度の問題のみを扱ってきた。ここでは、自由度が増加するに従って、本手法の計算量がどのように変化するかを述べる。

ここでは、図5. 26のような初期姿勢、最終姿勢の間を移動する例題を考える。マニピュレータ、障害物は例題1と同じである。この場合、マニピュレータの先端3自由度の関節角は初期姿勢、最終姿勢で同じである。このとき、ベースに近い N 関節($N=3\cdots 6$)のみを動かして N 自由度の障害物回避を行ったときに、必要な干渉チェックの総数を調べた。探索には複数戦略による両方向探索を用いた。表5. 16にその結果を示す。

表5. 16 探索の際の自由度と干渉チェック回数の関係

自由度	平均(95%信頼区間)	標準偏差s	最小	最大
3	105.3 (105.1 - 105.5)	1.75	102	108
4	147.3 (147.1 - 147.5)	1.75	144	150
5	1278.3 (1222.9 - 1333.6)	488.28	771	4220
6	2934.8 (2736.8 - 3132.7)	1746.30	1076	19421

この結果から、コンフィギュレーション空間の次元 N に関してセルの総数は N 乗のオーダーで増加しているのに対して、干渉チェックの総数の増加が極めて緩やかであることが分かる。

またこの例題では、立方体の障害物を避けるために、第1関節を動かすとともに、第2、第3、第5関節を動かしてワークを上を持ち上げる必要がある一方、第4、第6関節はこの障害物を回避するには余り重要でない。実験結果では、第4、第6関節の自由度を追加したときの干渉チェック総数の増加が少なく、不必要な方向への探索が極めて効率的に抑制されていることが分かる。

すなわち、本手法は次元が高い問題においても必要最小限の自由空間を参照して効率よ

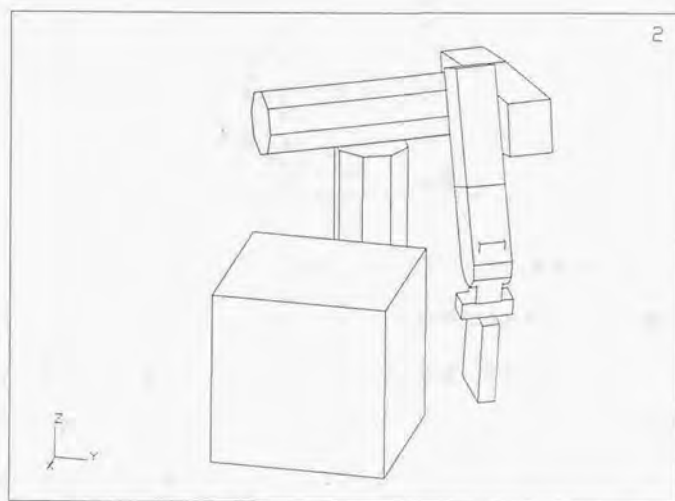
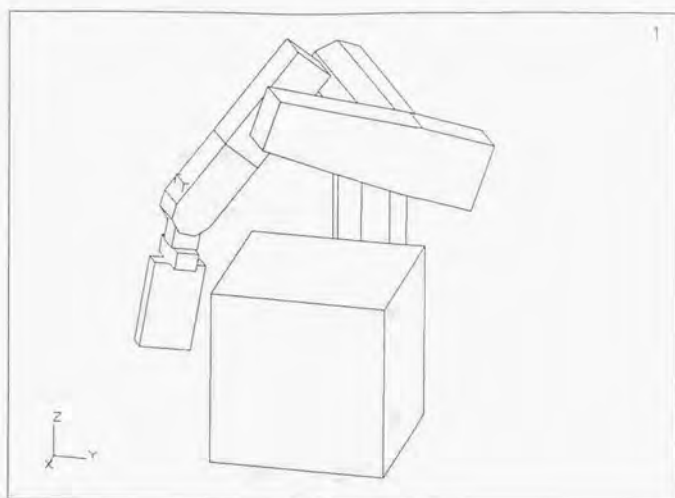


図 5. 2 6 自由度の影響を調べるために用いた例題

く障害物回避を行えることが確認された。

5.3.5 計算時間

ここでは、それぞれの例題の複数戦略両方向探索の場合の計算時間(平均値)を示しておく。計算を実行したのはSunのSpareStation370(17.5MIPS)であり主記憶は32Mbyteである。本システムは、Fortranでインプリメントされており、ワークステーションのほか32bitラップトップパソコンなどでも実行可能で、主記憶も8Mbyte程度以上あれば実行できる。

表5.17 計算時間

例題	計算時間(秒)
Ex. 1	78.2
Ex. 2	29.9
Ex. 3	122.8
Ex. 4	277.9
Ex. 5	55.0
Ex. 6	88.9
Ex. 7	458.0
Ex. 8	55.9
Ex. 9	86.9
Ex.10	89.9
Ex.11	35.8
Ex.12	137.8

計算時間はおおまかに考えて、1回の干渉チェックにかかる計算時間と干渉チェック回数
の積に比例すると考えられる。1回の干渉チェックに必要な計算時間は、実空間にどれだけ
の障害物があるかに影響される。ここでは、移動物体を構成する立体と障害物の立体のすべ
での組み合わせについて干渉チェックを行っているので、同じマニピュレータでも障害物の数
が倍になれば、1回の干渉チェックの計算時間もほぼ倍になると考えられる。本論文の手法
は、探索アルゴリズムと干渉チェックアルゴリズムは独立しており、実際のインプリメント
では、効率の高い干渉チェック手法と組み合わせるのが重要である。

これまでの研究との比較の意味で、代表的な2つの論文から計算時間を引用しておく。これらはともに Symbolics3600 上で計算されており、Lisp でインプリメントされていることが予想される。図5.27は[Lozano87]から引用したものであり、PUMA型マニピュレータの障害物回避である。ここでは先端を直方体で近似し、3次元コンフィギュレーション空間で動きを計画している。関節角は3度おきに量子化している。計算時間は約3分である。図5.28は[Donald87]から引用したものであり、6次元コンフィギュレーション空間を用いている。計算時間は数時間のオーダーである。計算機の処理速度、プログラミング言語の差があるので正確には議論できないが、本手法がこれまでの手法にくらべて高速であると言えるであろう。

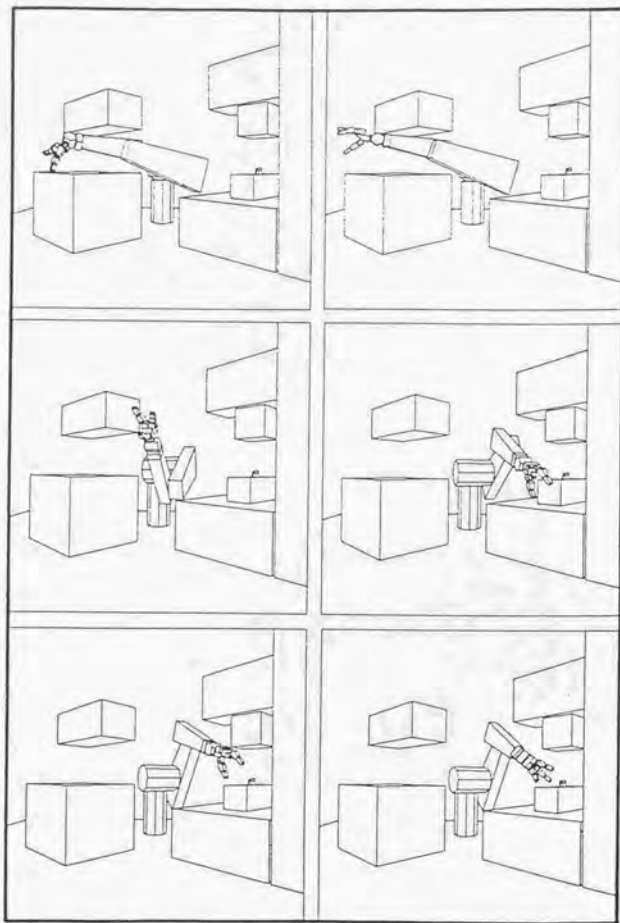


図 5. 27 PUMA 型マニピュレータの障害物回避 [Lozano87]

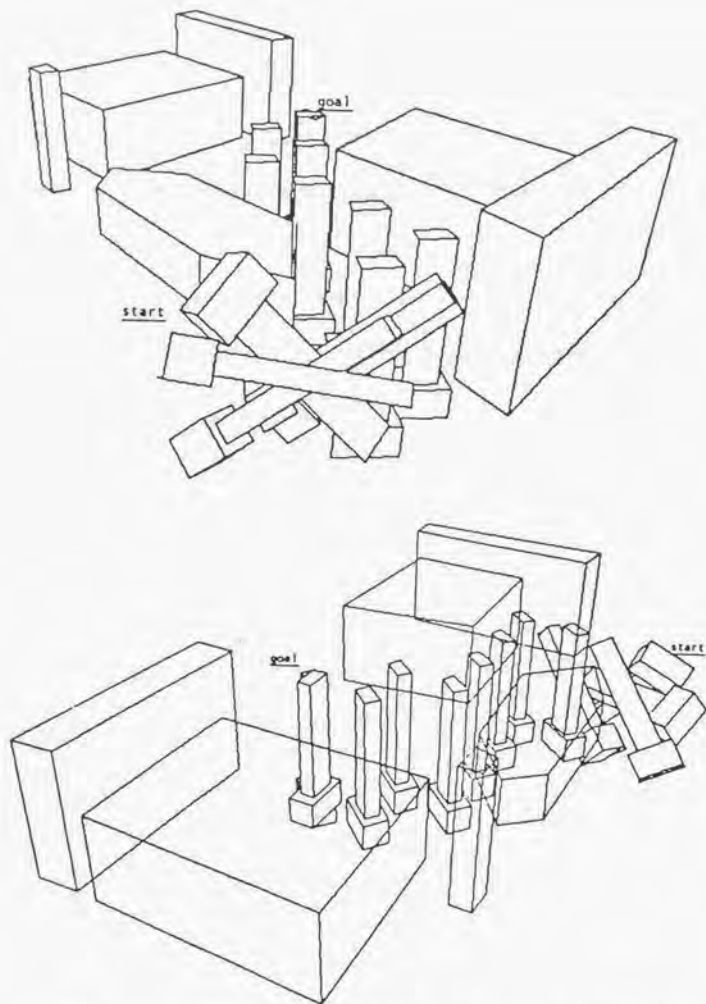


図 5. 2 8 6 自由度の剛体運動をする物体の障害物回避 [Donalds7]

5.4 第5章のまとめ

第5章では、これまで述べた自由空間の算出方法の効果を実験的に評価した結果について述べた。

ここでは、これまで一般的に適用するのが困難とされてきた6次元コンフィギュレーション空間を用いた6自由度の障害物回避動作計画を取り上げ、性質の異なる12の例題に適用した。

これらの実験結果をもとに、本論文で提案した方法の効果について考察し、現実の問題において自由度が高い場合においても十分に利用できる効率と一般性を持つことが実証された。

第6章

極限作業ロボットにおける 障害物回避動作計画

6. 1 宇宙用マニピュレータ JEMRMS

第6章では、これまで議論してきた障害物回避動作計画の手法を実際にインプリメントし、現実の極限作業ロボットの問題に応用した例について紹介する。これを通じて、本論文の提案する方法が工業的な観点から、どのように利用され、どのような効果があるかを考える。第6章では、すでに探索効率を検証する際に用いた宇宙用マニピュレータ JEMRMS と、原子炉内点検ロボットについて述べる。第1節では宇宙用マニピュレータを取り上げ、まず JEMRMS の概要について紹介し、次に JEMRMS のようなシステムにおいて、障害物回避動作計画がどのように使われ、どういった意味があるかを考える。

6. 1. 1 JEMRMS の概要

ここでは、日米欧加の国際協力プロジェクトとして進められている宇宙ステーション計画と、日本実験モジュールにおけるマニピュレータ JEMRMS [山脇 89][Yamawaki 89] について述べる。この宇宙ステーション計画は、フェーズ B における全体構想についての技術的な検討を終え、1988年9月に参加12か国の間で、本格的な開発段階以降の協力に関する宇宙基地協力協定が結ばれている。米欧加の各国では、この協定に基づいてフェーズ C (基本設計、詳細設計) 作業が開始され、わが国においても、本格的な開発段階であるフェーズ C 作業スタートの時期を迎えている。

日本は、与圧部、暴露部、補給部、マニピュレータなどで構成する日本実験モジュール (JEM: Japanese Experimental Module) (図 6. 1) を NASA が建設する宇宙ステーション本体 (図 6. 2) に結合し、無重力宇宙環境を利用した各種の軌道上実験を計画している。JEM の設計検討は、宇宙開発事業団を中心として進められ、東芝が設計を担当しているマニピュレータシステムは JEMRMS (JEM Remote Manipulator System) と呼ばれ、予備設計、関連技術研究

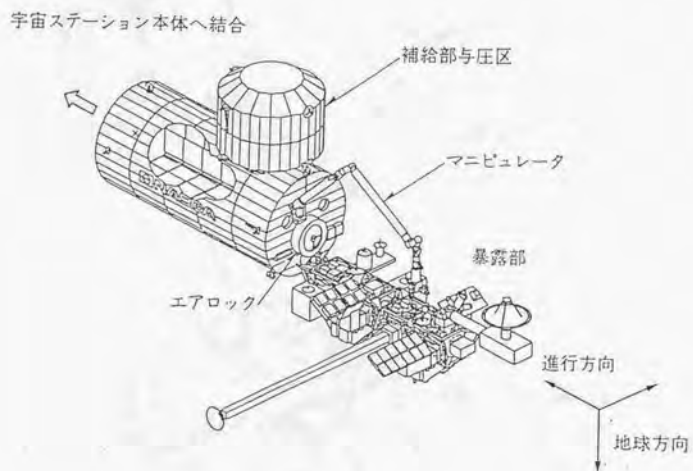


図 6. 1 日本実験モジュールの概要

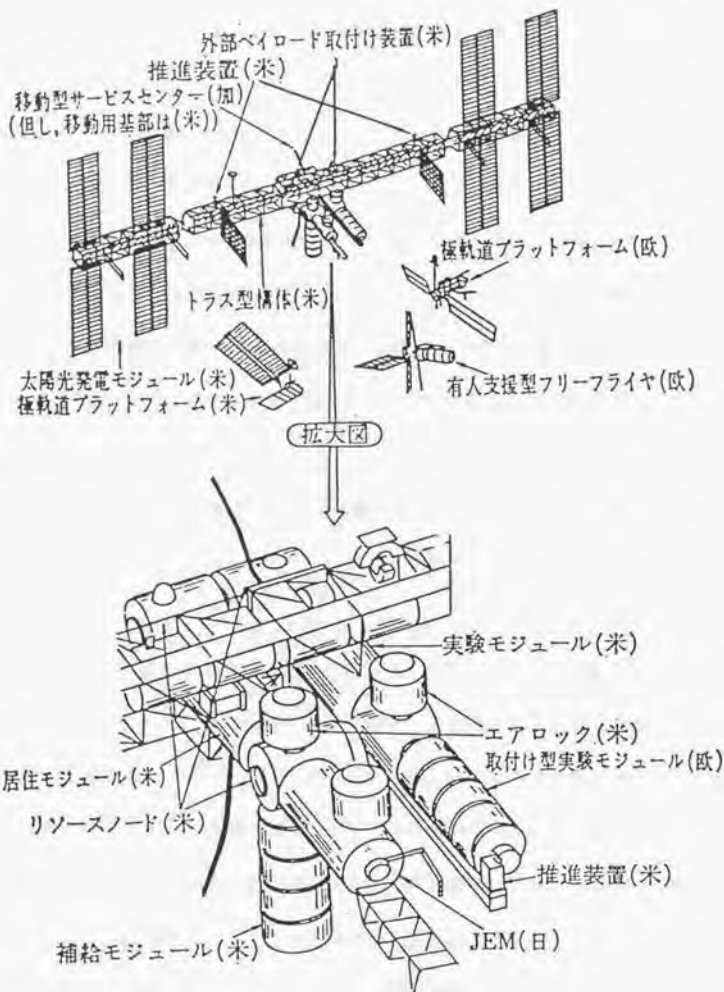


図6.2 宇宙ステーションの全体像

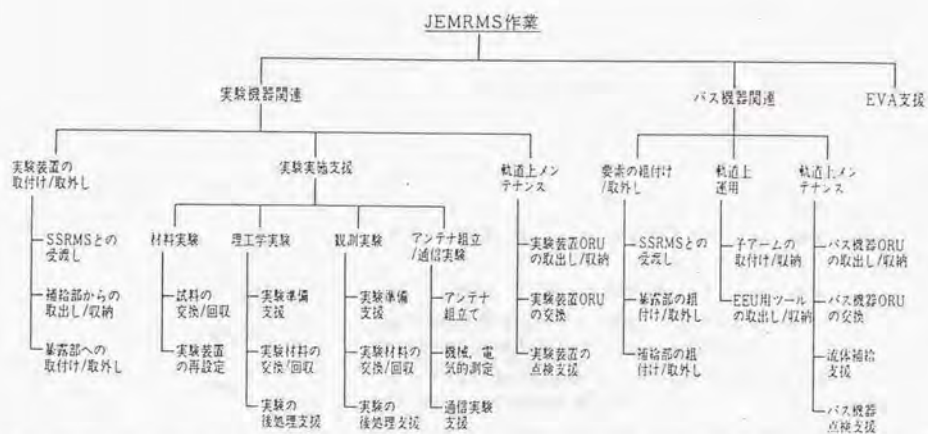
を経て、現在、本格的開発段階に向けてシステム設計および基礎試験を進めている。

JEMRMSは、与圧部内の搭乗員の遠隔操作により暴露環境下において、JEM構成要素、暴露部ペイロードのハンドリング、取付け／取外し、および実験支援などの図6.3に示す各種の軌道上作業を行う。

マニピュレータは、補給部や暴露部のような大型大質量物（最大7トン、4.5m×4mφ）のハンドリングから、実験用材料、試料およびアンテナ鏡面片などの小型小質量物までの広範囲な対象を取り扱う。また、作業対象物の把持／解放、移動などの単純な作業から、アンテナの組み立てや実験用材料、試料の交換など宇宙飛行士の船外活動を代替する精密かつ器用な作業までの広範囲な作業内容が求められる。特に、JEMRMSは、軌道上への打上げ後10年にわたって上記作業およびサービスが行なえるよう、軌道上の保全を含めて10年の設計寿命をもっている。

JEMRMSのマニピュレータは、上記の広範囲な作業対象および作業内容に対応して、二種のマニピュレータ（親アームと子アーム）で構成されている。主な構成要素を表6.1に示す。

親アーム系は、大型大質量物のハンドリング、宇宙ステーション本体マニピュレータ（SSRMS）とのペイロードの受渡しなどを行う親アーム機構、アーム操作装置、アーム制御装置などから構成されている。一方、子アーム系は小型小質量物を作業対象として、細密かつ器用な作業を行う子アーム機構で構成され、アーム操作装置、アーム制御装置は、親アーム系各装置と共用している。RMS管理系は、JEMRMSの運用、システムレベルの制御など各種管理をつかさどる計算機システムから構成され、視覚系は、親アームと子アームに搭載された視覚装置により、搭乗員にマニピュレータの遠隔操作および管理監視に必要な視覚情報を提供するテレビカメラ（単眼および立体）、雲台、照明装置、テレビモニタ、ビデオ制御装



EVA : Extravehicular Activity (宇宙飛行士の船外活動) EEU : Equipment Exchange Unit (暴露部装置交換機構)

図 6. 3 JEMRMS の作業

表 6. 1 JEMRMS の主要構成

系 名 称	主 要 構 成 品
親アーム系	親アーム機構, 親アーム制御装置, 親アーム操作装置 (マスタアーム, アーム操作エレクトロニクス, 操作盤)
子アーム系	子アーム機構, 子アーム制御装置, 子アーム操作装置
RMS 管理系	RMS 管理制御装置 (ホスト計算機, 外部記憶装置), RMS 操作監視装置 (表示装置, キーボード, 操作盤など)
視覚系	親/子アーム視覚装置 (単眼/立体テレビカメラ, 雲台, 照明視 覚制御装置, 視覚操作監視装置) (テレビモニタなど)
インタフェース系	インタフェース系制御装置, 親アーム保持解放機構, 操 作者拘束機構, インタフェース系操作装置
インテグレーション系	制御ラック, 操作ラック, モニタラック, ぎ装品

置などから構成される。インターフェース系は、JEM電力系および通信制御系とのインターフェースにかかわる制御装置、打上げ時に親アームを与圧部に固定保持するための機構、保持解放のためのエレクトロニクス機器、無重力状態でのマニピュレータ操作のための操作者拘束機構などから構成される。インテグレーション系は、与圧室内に設置される上記各系の機器を所定の配置に従い、与圧部に固定保持し、温度制御や火災検知、消火の各機能を備えた三種のラックで構成される。以上をまとめたJEMRMSの軌道上コンフィギュレーションは図6.4に示した通りである。

子アームの運用は、親アーム先端のエンドエフェクタにより、子アーム肩部のグラブリングフィクスチャを把持した状態で行う。子アーム作業場所の移動は親アーム動作モードに切換え、親アームを駆動することによって行う。また、子アームは非運用時、暴露部に設置された収納装置に保管し、温度制御や宇宙環境からの保護を行う。

親アームおよび子アームは種々の制御モードをもっている。親アームの基本制御モードは、あらかじめ設定された移動経路に従い、親アームを制御するプログラム制御モードである。このほか、マスタアームを使用するマニュアル操作への対応、各関節の独立した駆動、親アーム先端に作用する外部からの力やトルクに応じて関節が従動するリンプモードなど、目的に応じて各制御モードを使い分ける。

子アームの基本制御モードは、バイラテラル制御モードであり、子アーム先端に作用する力やトルクを与圧室内のマスタアームを介して操作者に帰還し、立体テレビ画像をモニタしつつ、種々の細密作業を行う。また、比較的質量の大きい作業対象物のハンドリングには、力帰還を行わないユニラテラル制御モードを使用する。このように、子アームによる作業は、マスタアームを操作して子アームの先端位置および姿勢を制御するマスタスレーブ方式を採用しているが、保管収納形状に自動的に変更するプログラム制御モード、バックアップ

与圧室内
コンフィギュレーション

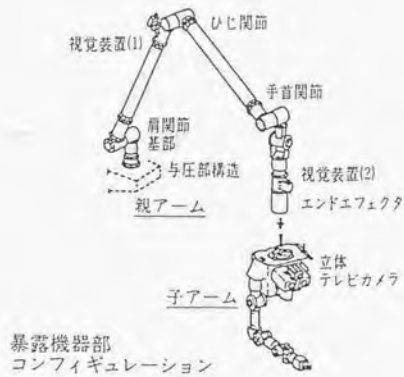
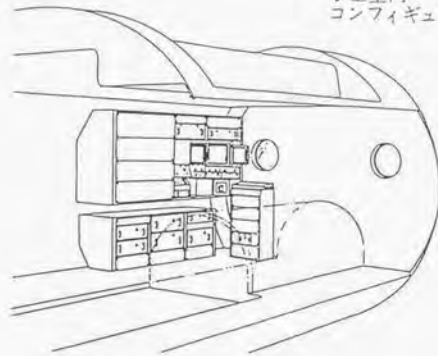


図 6. 4 JEMRMS の軌道上コンフィギュレーション

モードとして単軸駆動などもある。

6. 1. 2 JEMRMS における障害物回避動作計画

以上で述べたように、JEMRMS は、プログラムモードやマスタスレーブモードを適切に組み合わせることにより、様々な作業を実行する。特に、JEM は、有人システムであることから、作業の効率化などとともに、安全性についての十分な考慮がなされていなければならない。

JEMRMS における安全設計としては、マニピュレータ運用時の衝突防止システムの構築が挙げられる。これは、アームの誤動作による暴走の防止およびアーム動作域内の JEM 構成要素や船外クルーへの衝突の防止であり、オペレータによるアームの動作監視とともに、アーム動作域内の移動物体のモニタなどのシステムにより安全確保を図っていく必要がある。この際、アームとの干渉を計算機によって検出するための干渉チェック機能が不可欠となる。

一方、作業の効率化という観点からは、干渉チェック機能を一步進めた障害物回避動作計画の機能が重要となる。特に親アームはプログラム制御モードを基本モードとしており、主に大局的な移動の動作を実行するため、途中の移動経路は、始点と終点を指定するだけで自動生成されることが望ましい。これにより、オペレータのミスによる事故を未然に防ぐ効果が期待できるとともに、オペレータの負担を大幅に軽減することができると考えられる。

6. 1. 3 例題

ここでは、JEMRMSに本論文で提案する障害物回避動作計画システムを適用した結果を紹介する。ここで紹介する例題は、JEMRMSの親アームによるペイロードの移動を想定して、障害物回避動作計画を適用したものであり、プログラム制御モードでマニピュレータを動かす場合に対応する。

例題1(図6.5)は、アンテナ下方にアームがある状態から、ペイロードを暴露部側面に移動させる問題である。アンテナのような細い障害物も見落とすことなく回避している。

例題2(図6.6)は、ペイロードを暴露部をまたいで暴露部の反対側に移動させる問題である。

例題3(図6.7)は、暴露部側面にあるペイロードを暴露部の反対側に移動させる問題である。この例題では、暴露部をまたぐという意味で例題2と同じであるが、異なった移動の仕方をしているのが分る。

JEMRMSは6関節を持つ6自由度マニピュレータであり、以上のような移動経路を算出させるために、6次元関節角空間をコンフィギュレーション空間とした。関節角は各軸5度おきに量子化を行いセル分割している。このように、比較的障害物の少ない環境で動く宇宙用マニピュレータであるが、自由度が多いために、従来の方では障害物回避動作計画を適用することができなかった。本手法を用いることにより、第5章でも述べたように、これら3つの例題すべてにおいて極めて効率よく経路を生成することに成功している。

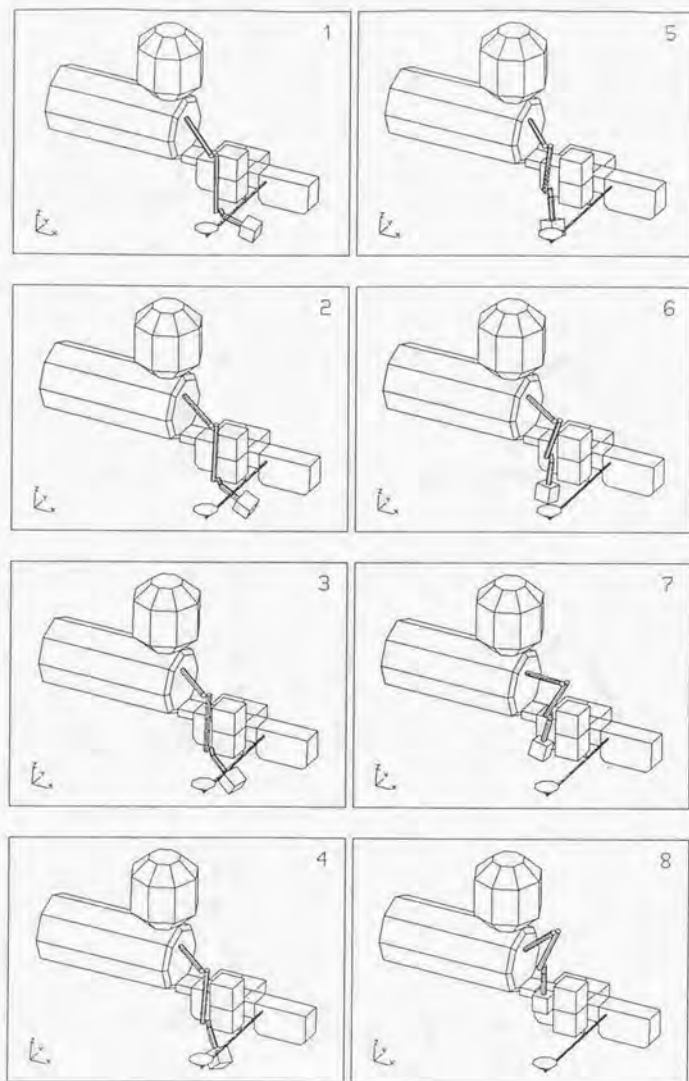


図 6. 5 算出された JEMRMS の移動経路 (例題 1)

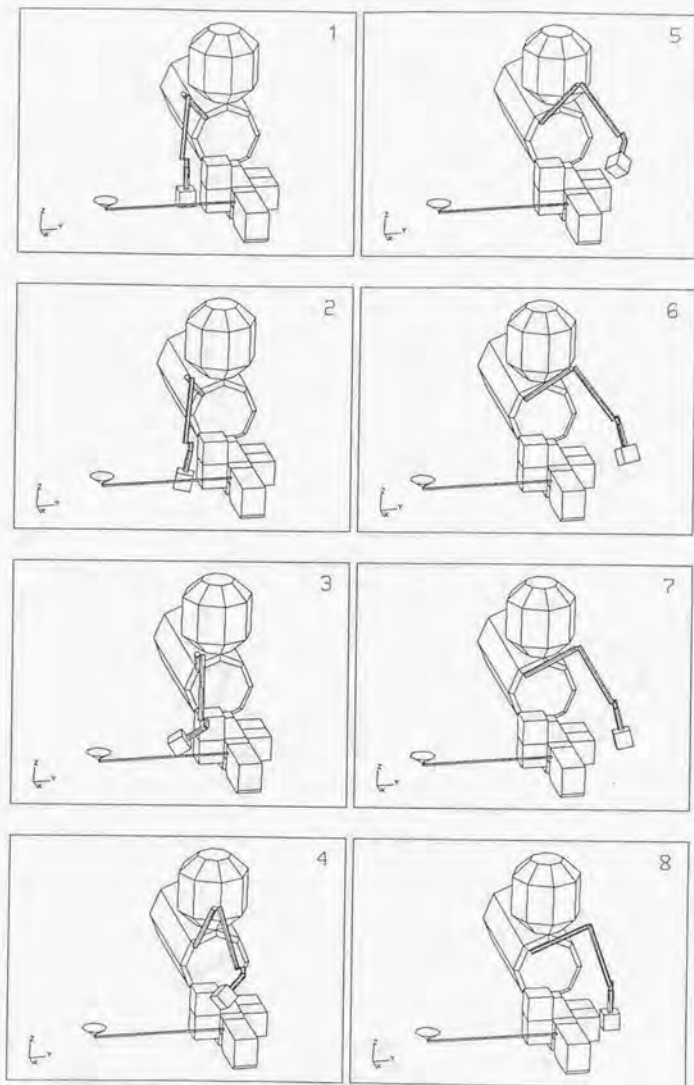


図 6. 6 算出された JEMRMS の移動経路 (例題 2)

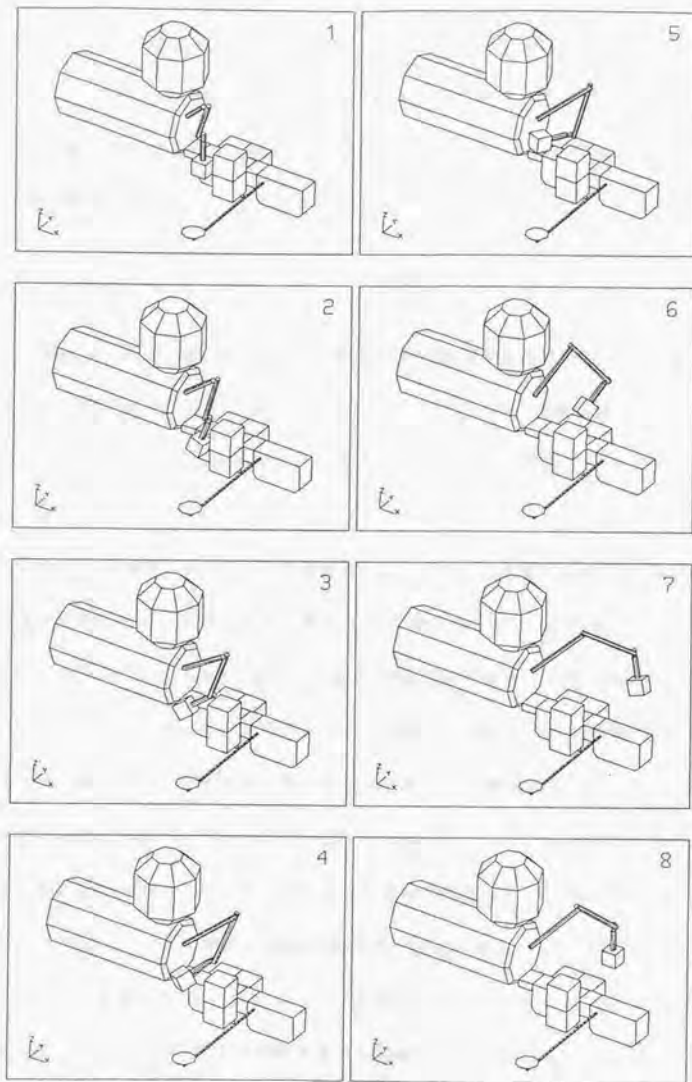


図 6. 7 算出された JEMRMS の移動経路 (例題 3)

6. 2 原子炉内下部構造物点検ロボット

本節では、プラントの点検ロボットの例として原子力への応用について述べる。ここでは、まず原子力プラントの概要を紹介し、これをもとにプラントの点検ロボットに必要なとされる機能を明確にする。さらに、原子力プラントの点検ロボットへの応用を通じて障害物回避動作計画の意義を考える。

6. 2. 1 BWR原子炉の概要

原子力発電所は、用いられている原子炉の構造から沸騰水型原子力発電所と加圧水型原子力発電所に大別される。ここでは、後から紹介する原子炉内下部構造物点検ロボットが検査対象とする沸騰水型原子力発電所と、そこで用いられている沸騰水型原子炉の概要について簡単に説明する。

沸騰水型原子力発電所(BWR原子力発電所)は、1960年に運転を開始した米国のドレスデン1号機を始めとして、すでに世界で約85基が運転中であり、日本国内においては現在18基が運転中(総出力約1510万kW)で多くの運転実績を積み重ねている。沸騰水型原子力発電所は、直接サイクルで構成がシンプルであり、運転がしやすい、出力制御性や日負荷応答性が良い、低い原子炉圧力で質の良い蒸気が得られるなどの特長がある。

沸騰水型原子力発電所は大きく分けると原子炉系設備とタービン・発電機設備および放射性廃棄物設備とから構成されている。原子炉系設備は、低濃縮ウランを燃料とし、減速材と冷却材に水を用いて、これを炉心で直接沸騰させ、蒸気を発生させる方式の沸騰水型原子炉-Boiling Water Reactor(BWR)が中心となる。沸騰水型原子炉の炉心は、原子炉圧力容器の中に納められており、この中で水を沸騰させる。発生した蒸気は、蒸気出口ノズルより取り出されタービンに導かれて、タービンと発電機を回転させ、電力を発生する。沸騰水型原子

力発電所で使用する蒸気は、火力発電所とは異なり低圧の飽和蒸気のため、タービン羽根が長い、羽根に水きりがついている、高圧・低圧タービン間に湿分分離器がついているなどの特徴がある。また、タービン系の主要機器や配管は放射線のしゃへいが考慮されている。タービンを通過した蒸気は復水器などを通して水となり、再び原子炉へ送られる。また、原子力発電所には、放射性廃棄物処理設備が備えられており、放射性廃棄物は気体、液体および固体に分け、それぞれに適した方法で処理される。

原子炉圧力容器や再循環系等の一次系は、鋼製の原子炉格納容器内に収納されている。原子炉格納容器にはMark-I改良型、Mark-II改良型等があり、万一事故が起こっても原子炉一次系からの放射線物質の放出を防ぐ役割がある。図6.8にMark-II改良型格納容器を用いた沸騰水型原子力発電所の全体像を、図6.9にその系統構成を示す。原子炉圧力容器は、炉心を内部に形成する原子力発電所の主要機器であり、内部には、燃料集合体、制御棒および炉内構造物が収納されている。炉内構造物は、燃料集合体を支持する炉心支持構造物と気水分離器、蒸気乾燥器等の内部構造物とから構成されている。図6.10は炉内構造物、炉内機器等の説明図である。

炉心は、燃料集合体が装荷される部分であり、制御棒案内管と燃料支持金具を介して燃料集合体を保持する炉心支持板、燃料集合体の上端を保持する炉心支持板、および燃料集合体を包んで炉心支持板と上部格子板を保持する炉心シュラウドに囲まれた部分である。炉心シュラウドは、シュラウドサポートリング、サポートシリンダ、サポートレグを介して原子炉圧力容器に固定されている。

炉心は、冷却材である水が満たされており、さらに、炉心から熱を有効に取り出し、蒸気を発生しやすくするために、冷却材(水)は炉心で強制循環させられている。冷却材の強制循環は、炉心外側に配置されたジェットポンプや原子炉格納容器の外部に設けられた2台の

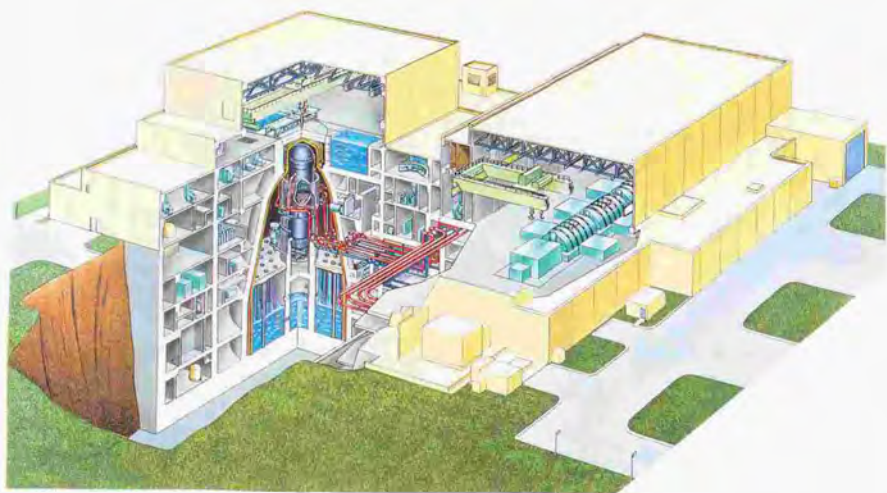


図 6. 8 沸騰水型原子力発電所の全体像

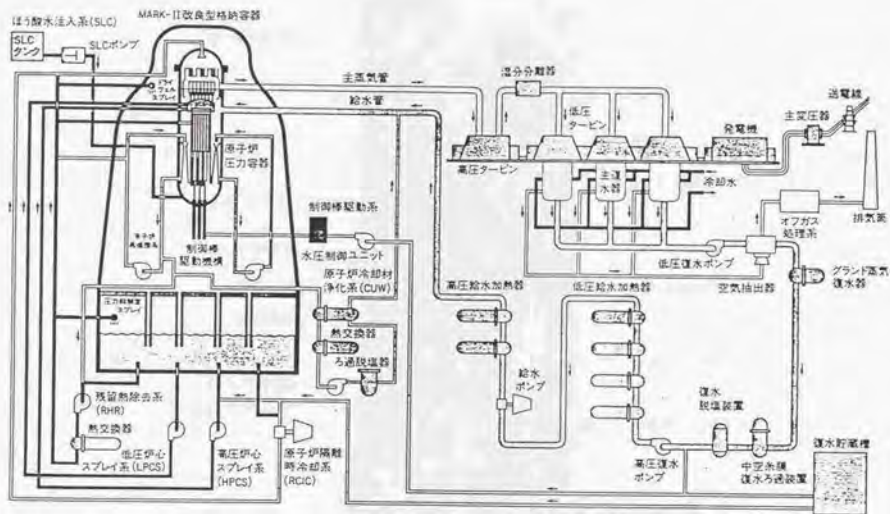


図 6. 9 沸騰水型原子力発電所の系統構成

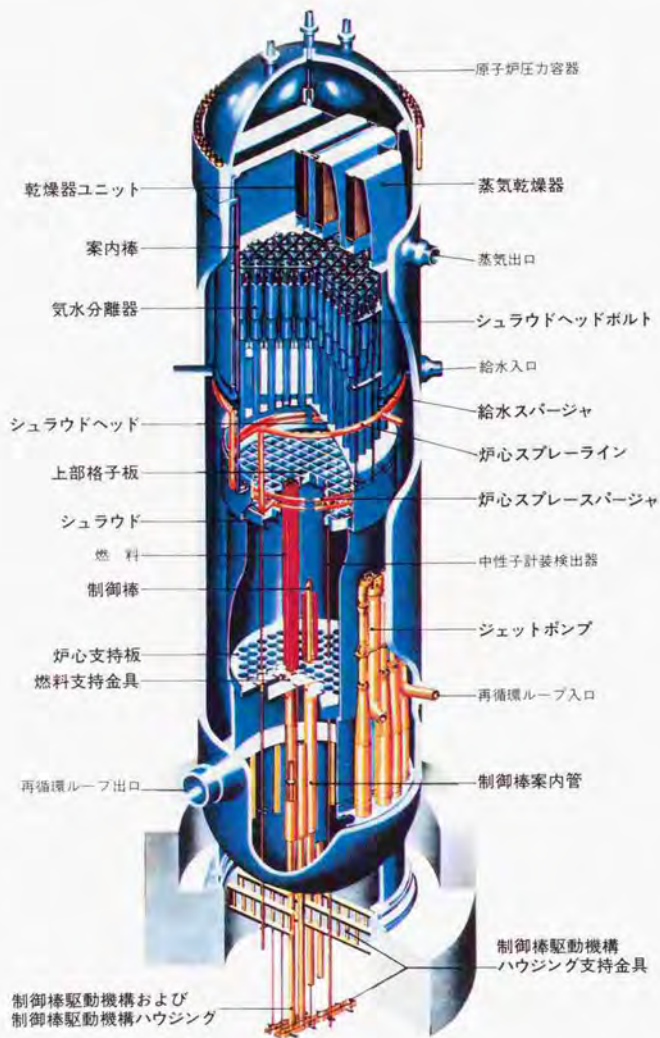


図 6. 1 0 炉内構造物と炉内機器

冷却材再循環ポンプなどから成る原子炉冷却材再循環設備により行われる(図6.11)。また、万一の原子炉系配管破断による冷却材の流出に備えて、非常用炉心冷却設備が備えられている。これには、事故の際、炉心に冷却水を注入・補給するとともに、炉圧を下げる装置である、高圧炉心スプレイ系、高圧炉心注水系、原子炉隔離時冷却系や、事故により冷却材が少なくなり、十分に冷却ができなくなった場合に炉心にスプレイ水を吹きつけて燃料を冷やすための低圧炉心スプレイ系、事故のあと、圧力容器内に大量の冷却水を注入し、水づけにする低圧力注水系などがある。

炉心での核分裂の連鎖反応は、余分の中性子を吸収する機能を持つ制御棒を燃料集合体の間に挿入し、その挿入量を調節することによって制御される。この制御棒は十字形をしており、制御棒案内管の中を上下動し、先端部が4体の燃料集合体の中央に挿入される。制御棒駆動機構は水圧制御ユニットにより水圧で遠隔駆動する構造になっている。図6.12に制御棒の構造を示す。また、原子炉に異常事態が発生した場合、全制御棒は自動的に急速挿入され、原子炉の緊急停止を行う。

炉心で加熱された冷却材は、沸騰二相流となる。炉心からのこの沸騰二相流は、気水分離器により蒸気と水に分離され、分離された蒸気は、蒸気乾燥器によりさらに水分が除去されて、蒸気出口ノズルより取り出される。

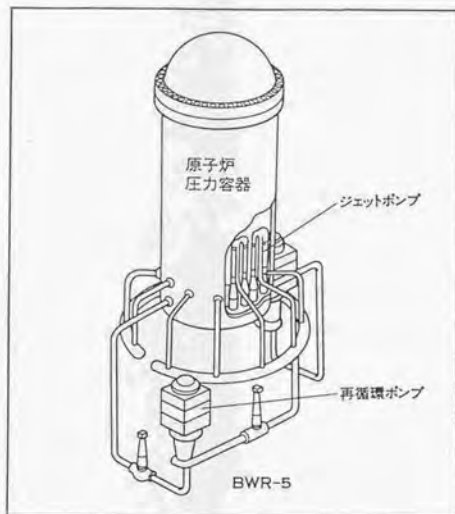


図 6. 1 1 原子炉冷却材再循環設備

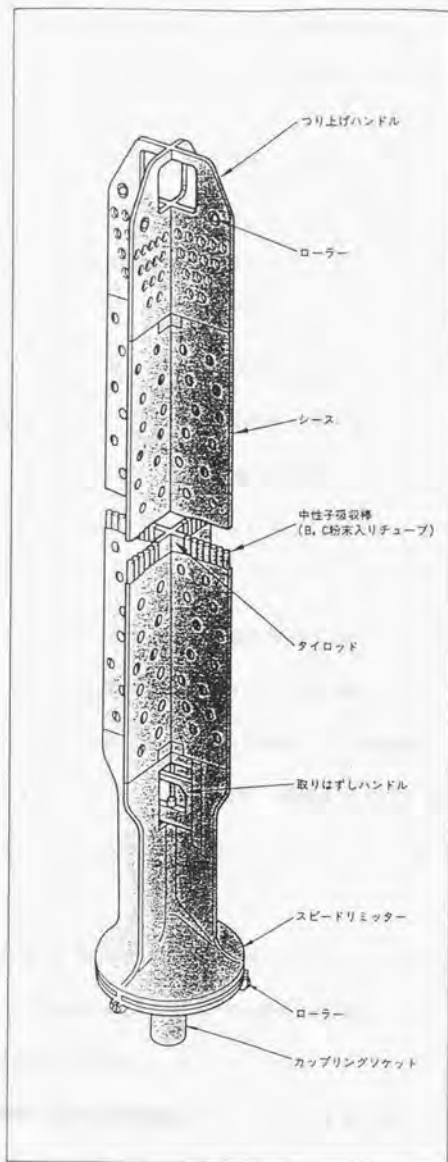


図 6. 1 2 制御棒

6. 2. 2 原子炉内下部構造物点検ロボットにおける障害物回避動作計画

原子力発電所のように、特に安全性が強く要求されるシステムにおいては、万一の事故の際に、事故の被害を最少限にとどめ、安全に運転を再開することが可能なように、様々な非常用の設備が設けられている。しかし、プラントの各種設備の信頼性の確保や健全性維持の確認の観点から、プラントの運転開始後も、定期的に各種設備を点検することは必要不可欠のことである。

原子力発電所においては、日本電気協会規程「軽水型原子力発電所用機器の供用期間中検査」に基づき、炉内構造物についても、定期点検時に供用期間中検査(ISI)を実施している。原子炉圧力容器内については、水中テレビで接近可能な範囲についての目視検査や、さらに検査精度を高めるための超音波探傷検査を行うための各種遠隔点検装置が開発されている。

このような、点検装置においては、必要な点検箇所において、必要な点検項目のチェックを確実に行うことのほかに、点検の効率向上を図り、短い時間でより多くの項目について点検できるようになっている必要がある。点検の効率向上は、定期点検によってプラントを休止させる期間を短縮することにつながり、稼働率や経済性の向上を高い安全性を確保しながら達成することが可能になる。また、効率的な点検の実行は、より一層の被爆低減化にもつながると考えられる。

ここでは、本論文で述べた障害物回避動作計画のアルゴリズムを実際に適用した原子炉内下部構造物点検ロボットを例にとり、障害物回避動作計画がプラントの保全・点検などの場面においてどのように活用されるかについて考える。

原子炉は原子力発電所の最も重要な設備であり、炉内構造物は、原子力発電所の信頼性を確保するために、完全にその健全性が確保されなければならない。炉内構造物のうち、炉内

上部については、比較的接近が容易であり、気水分離器遠隔目視検査装置や、シェラウドヘッドボルト遠隔超音波探傷装置、上部格子板グリッドプレート遠隔超音波探傷装置などが実用化されている。これに対して、炉心内部、炉内底部は、接近する上で形状が複雑で狭隘な箇所があり、点検は点検箇所への接近などの点から炉内上部に比べて困難な状況にある。そのため、炉内下部構造物点検装置においては、点検箇所への接近などの面を自動化し、オペレータの負担低減と操作性の向上を図るとともに、検査実施前の十分な検討によって効率をあげる必要がある。

ここで述べる原子炉内下部構造物点検装置（ロボット）は、炉内下部の以下の対象を必要に応じて目視検査ないし超音波探傷検査する。

- (1) シェラウド胴とシェラウドサポートリングの溶接部
- (2) シェラウドサポートリングとサポートシリンダの溶接部
- (3) バッフルプレートと原子炉圧力容器の溶接部
- (4) バッフルプレートとアダプタ（ジェットポンプ取り付け部）の溶接部

この検査装置は、制御棒案内管と同じ形状のアーム収納部に収納されたアーム先端にテレビカメラないし超音波探傷検査装置を取り付けた構造である。点検時には、炉内最外周の制御棒案内管を引き抜き、そこにアーム収納部を燃料交換機補助ホイスで下ろして固定し、必要な点検箇所までアームを動かす。アームには、円筒形アーム収納部の軸回りの回転、同じく軸沿いの上下動並進、アーム横出しの回転の基本3自由度を持ち、超音波探傷検査装置の首振りの自由度のように必要に応じて自由度が追加される。

この検査装置では、原子炉圧力容器やシェラウドの外周に沿って多くの溶接部の点検を実行しなければならない。効率的に点検作業を実行するには、どの制御棒案内管を引き抜き、どの位置にアームをセットするかをあらかじめ十分検討し、無駄のない作業手順を作ってお

く必要がある。原子炉内下部構造物の点検のように、狭隘な環境でアームを移動させなければならないような場合には、無駄のない作業手順を作成するのは容易ではない。考えられるいくつかの作業手順について、それぞれ障害物回避動作計画を行わせ、移動のための手間と時間を見積もることにより、短時間のうちに無駄のない作業手順を作成することが可能になる。また、検査実施中において、何か付加的な動きをしなければならないような場合でも、その作業や動きについてのみ障害物回避動作計画を実行させることにより、オペレータのミスを未然に防いで能率的に作業を進めることができる。この結果、オペレータは、本来の点検作業に集中することができ、点検全体の効率向上も期待することができる。

また、本システムのように対象があらかじめ決まっており、対象となる施設側の設計変更が困難な場合には、障害物回避動作計画は、機構設計における有効な設計ツールとなる。とくにここで検査対象としている炉内下部のような狭隘な空間で移動する機構の場合、リンクの長さなどを、周囲との干渉がないように、また確実に移動できるように慎重に決定しなければならない。こういった場面で、リンクの長さなどの設計パラメータに対応したモデルを用いて必要に応じて干渉チェックを実行し、確認の目的で障害物回避動作計画を実行させれば、設計の効率化に大きく貢献する。

6. 2. 3 例題

ここでは、原子炉内下部構造物点検ロボットに障害物回避動作計画を適用した結果について紹介する。

作業環境は、原子炉の1/4に相当する中心角90度の扇形の部分についての炉内下部構造物を凸多面体の集合として表現した。形状は不必要に細かくなることのないように、適切な簡略化を行っている。制御棒案内管は外周部分のみをモデル化し、本装置の点検対象としては関係のない内側の制御棒案内管を省略した。また、原子炉圧力容器底部の球面状の部分と制御棒駆動機構の部分は、一体として簡単な形状としてモデル化してある。バッフルプレートは、穴のあいた凹形状を凸多面体の集合として表現するために、複数の多面体に分割してある。

例題として、テレビカメラによるバッフルプレートとアダプタ(ジェットポンプ取り付け部)の溶接部の目視検査を考える。この作業を実行するためには、アーム先端に取り付けられた側視ヘッド(カメラ)をバッフルプレートの穴の中に挿入する必要があり、この動作を障害物回避動作計画の問題として計画させた。この結果を図6. 13に示す。

ここでは、コンフィギュレーション空間を、円筒形アーム収納部の軸回りの回転、同じく軸沿いの上下動並進、アーム横出しの回転の基本3自由度から定義される3次元空間とした。アーム収納部の軸回りの回転角とアーム横出しの回転角については、3度おきに区切ってセル分割し、上下動並進は15mmおきに分割した。なお、原子炉圧力容器の直径は約6000mmである。この問題は、3自由度の問題であり、もともとのコンフィギュレーション空間の大きさが6自由度の場合に比べてはるかに小さいため、ヒューリスティクスによる探索範囲の限定の効果はそれ程顕著には現れない。むしろ、このように障害物となる立体の数が多い場合(この場合の障害物数83)には、明らかに干渉しないと分っている障害物とは干渉チェックを

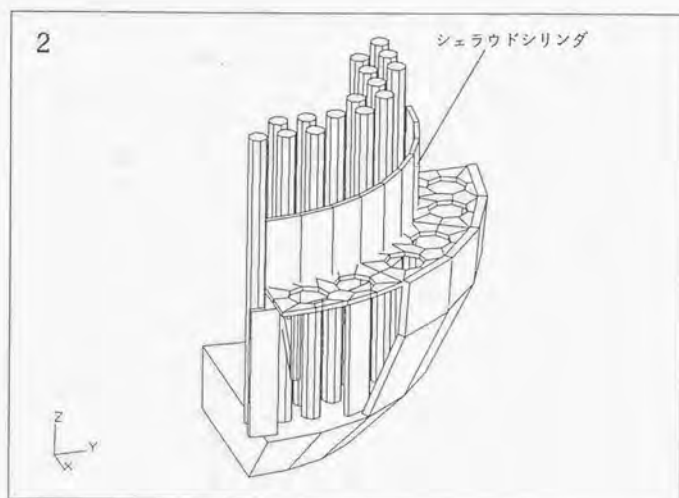
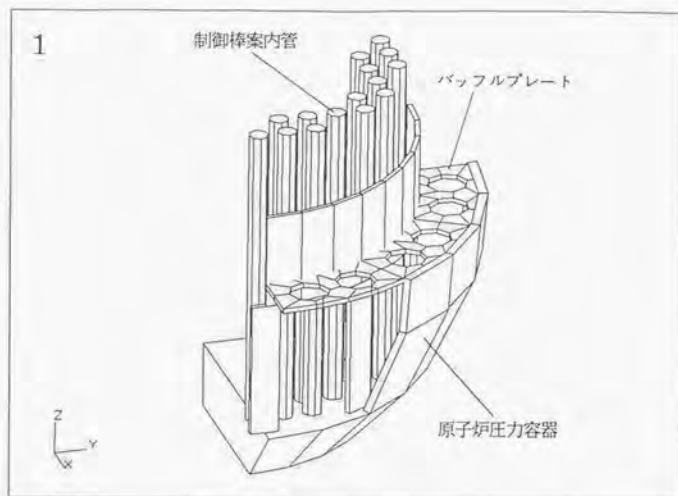


図6.13 炉内点検ロボットの移動経路(1)

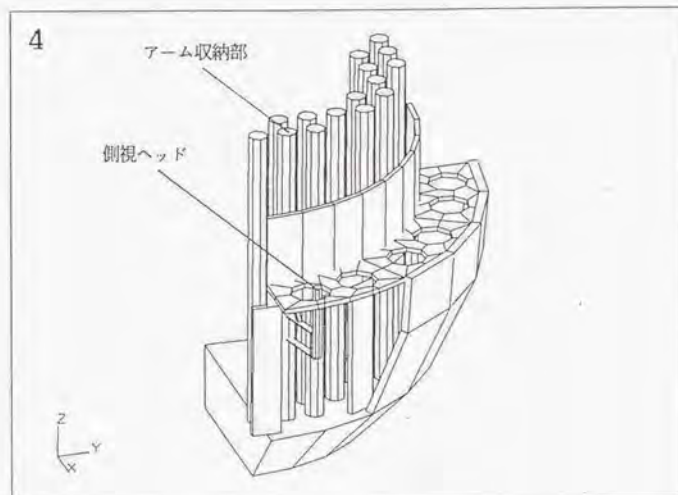
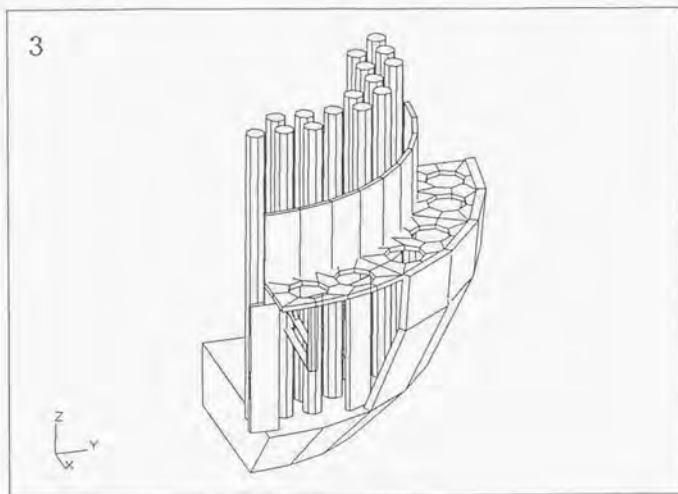


図 6 . 1 3 炉内点検ロボットの移動経路 (2)

しない工夫をすることが、実用上は重要である。この例題では、アームの設置位置(どの制御棒案内管の位置に設置するか)で障害物を限定している。

6.3 第6章のまとめ

第6章では、本論文で提案する障害物回避動作計画の手法を実際にインプリメントし、現実の極限作業ロボットの問題に応用した例について紹介した。これを通じて、本論文の提案する方法が極限作業ロボットという観点から、どのように利用され、どのような効果があるかを考察した。

まず宇宙用マニピュレータ JEMRMS を取り上げ、JEMRMS の概要を説明した上で、宇宙用機器での障害物回避動作計画の役割について、例題を通じて考えた。特に、自由度が高いため従来困難であった宇宙用マニピュレータの動作計画を自動化できたことを示した。

次に、原子炉内点検ロボットを取り上げ、BWR 原子炉の概要を説明した上で、プラントの点検ロボットにおける障害物回避動作計画の役割を、現実の応用の観点から考えた。特に、周囲の環境が複雑なため、人手に頼らざるを得なかった、点検ロボットの経路生成が自動化されたことを示した。

以上の議論を通じて、極限作業ロボットにおいては、障害物回避動作計画の役割が非常に重要であり、本論文で提案する方法を適用することにより、ロボットシステムの効率や安全性の向上が図れることが確認された。

第7章

プラント設計における 障害物回避動作計画

7.1 プラント設計CADにおける物体の移動の表現と推論

第7章では、プラントCADシステムにおける障害物回避動作計画の役割について、現実の問題への本論文の方法の適用結果を通して考える。まず、プラントCADと障害物回避との関係を一般的に考察した後、実際の例題について紹介する。

原子力発電プラントに代表されるようなプラント設計においては、決められた立地条件のもとで、必要な建屋を配置し、また、それぞれの建屋のなかに、所定の機器を配置し、これらの機器間の配管の引き回しを決定する必要がある。これらの建屋、機器、配管などの配置は、設計上の様々な制約条件を満足することが要求され、これらの制約を満たす設計解を求めるのは容易ではない。制約条件としては、メンテナンススペースなどのように配置される機器それぞれに適用される条件や、放射線レベルによる部屋の分離など機器が配置される領域に関する条件など多種多様なものが存在する。

プラント設計CADにおいては、これらの設計上の制約条件をすべて満足する解を自動的に導出しようとするのは現実的ではなく、むしろ設計者の作業の進行とともに、制約条件を自動的にチェックする対話的なシステムとするのが良い。このような観点から対話性に重点をおき、制約条件の自動チェック機能を実現したシステムとして、[Kameyama90]を開発した。これは、フロアごとの図面を基本とした2次元のシステムであり、主にプラント内での機器配置問題を対象としている。このシステムにおいては、機器個々に適用される制約条件を、オブジェクト指向によって管理される機器の属性と属性に応じて制約条件を適用するルールベースシステムによって表現し、部屋ごとの放射線レベルのように領域に対する制約を、形状モデルとして表現される位相情報に関係付けることによって、多様な制約条件に対応している。

さらに、プラント設計においては、機器の搬出入の可能性のチェック、定期点検時などに分

解した機器の移動方法や、点検中の部品の一時保管場所などについて詳細に検討しておく必要がある。機器の搬出入に関しては、トラック等による機器のプラントへの搬入から設置位置までの移動方法や、搬入手順について検討を行い、クレーン、モノレール、ホイストなどの移動設備が、適切にプラントの設備として設置される。これら移動機器の位置関係は、万が一の故障の際の機器の交換などの場合も考慮しながら、慎重に決定しなければならない。一般にプラントにおいては、複雑に引き回された配管などで環境が非常に複雑であり、その中で干渉しないような移動経路を求めることが要求されるので、極めて面倒な作業を行わなければならない。また、定期点検時の機器の移動や一時保管などの点からも、機器の分解手順と外した機器の移動方法や、メンテナンススペースを考慮した一時保管位置などを、プラント設計の段階で詳細に決定する必要がある。この場合には、作業の効率が重要な考慮事項となり、例えばハッチの開閉回数などで全体の作業効率を評価しながら、効率よく点検作業ができるような設計案をみつけなければならない。また、機器の搬出入が極めて困難な場合には、その機器を分解可能な形に設計変更することが必要になる場合もある。

このように、プラント設計において搬出入の問題は極めて重要な問題であり、機器の搬出入計画や、定期点検時の運用法などの観点から、機器の移動の問題を頻繁に扱う必要がある。特にクレーンやモノレールなどの位置を決定する場合には、それぞれの設計案に対して、実際に機器の搬出入を安全に実行できるかどうかを確認する必要がある。図7-1は、原子力発電所の原子炉複合建屋の一部をCADモデルを用いて表示したものである。このように、配管、機器、架台などが複雑に配置された環境において、図面によるチェックのみで、移動の安全性を確認するのは極めて困難である。そこで従来は、最終的な検討や確認を、現実のプラントを忠実に縮小したプラスチックモデルを作製して行っていた。最近では、これらプラスチックモデルに代わるものとして、CADデータからの3次元形状モデルが利用され

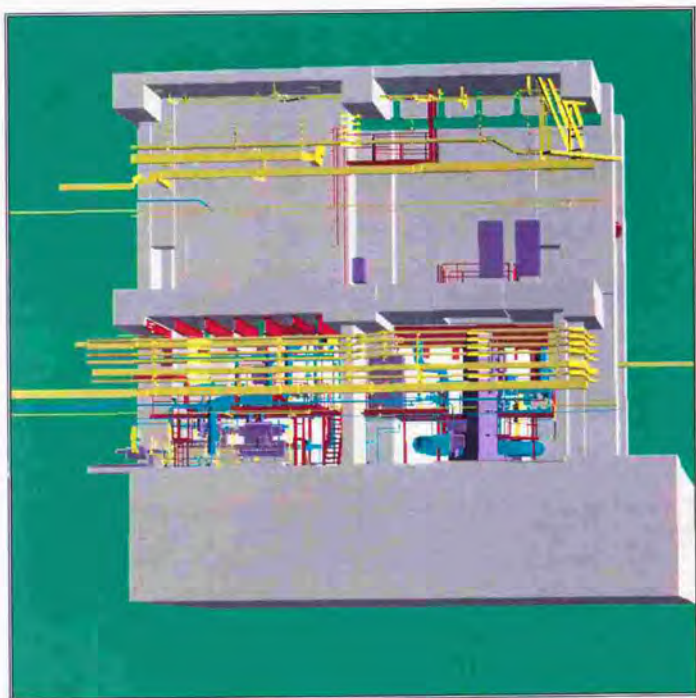


図 7 . 1 原子力発電プラントの CAD モデルの一部

るようになってきているが、この場合にも、3次元グラフィックスによる目視の確認のみでは十分とは言えない。干渉チェックや障害物回避動作計画の機能を利用して、ミスを事前にチェックすることが重要である。

7. 2 機器搬出入計画における障害物回避動作計画

以上で述べたようなプラント設計における機器の移動問題は、そのまま障害物回避動作計画問題として考えることができる。例えば、機器搬出入の問題を障害物回避動作計画の問題としてとらえ、搬出入経路を、機器の設置位置と搬出入ハッチの位置などの間の移動経路として自動生成させることができる。設計者はこの結果に基づき、プラントの設計案の妥当性の確認をすることが可能になり、搬出入の手順や必要な時間などを容易に見積もることができる。また、移動経路の有無を自動的にチェックすることにより、設計上のミスを容易に発見することが可能になる。

ただし、プラントにおける機器搬出入においては、クレーン、モノレールなどそれぞれの場所や対象に応じた移動用の設備が使用される。また、重量物が多いため、同一フロア内では、コロを敷いて移動させる場合も多い。そのため、機器の移動はこれらの移動方法や設備を前提としなければならず、移動設備それぞれに特有な移動方法の制約や能力の問題を柔軟に反映した上で、移動経路が算出可能なシステムでないと対応が困難となる。

コンフィギュレーション空間に基づく方法は、すべての移動問題を、移動物体の位置・姿勢を一意に定めるパラメータによって張られる空間における点の移動問題に帰着させ、一般的に解こうとする方法であり、多種多様な移動設備を扱わなければならないプラントCADに適した方法であると言える。

さらに本論文で提案した方法は、経路探索を行う部分と、コンフィギュレーション空間のパラメータから実空間での位置・姿勢を算出して制約条件や干渉のチェックを行う部分を分離してインプリメントすることが可能であるので、個々の移動設備に特有の制約を、経路探索の方法と無関係に容易に組み込むことができる。例えば、前後に長い機器の2点吊りのモノレールによる移動(図7.2)を考える。この場合には、前後で機器を吊る位置は、モノ

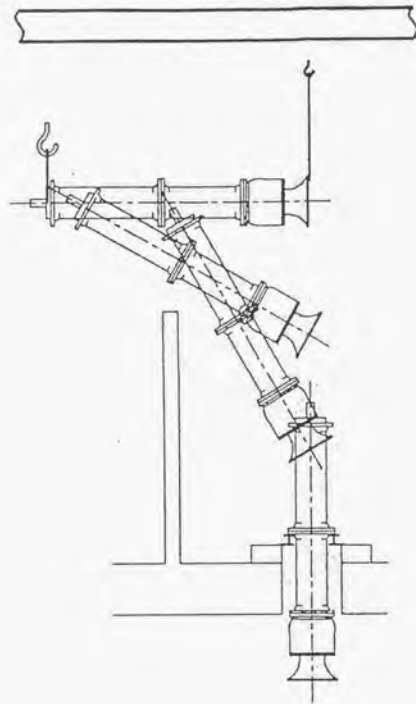


図 7. 2 2点吊りのモノレールによる移動

レールのレール上にあることが必要であり、かつカーブではどれくらい機器が内側にはみだすか(自動車の内輪差に相当)を機器の長さによって算出しなければならない。このように自由度がそれほど多くなくても、モノレールという設備に特有の拘束が多い場合には、本論文で提案した方法が特に有効である。

7. 3 例題

ここでは、現実のプラントのモデルに本論文の障害物回避動作計画システムを適用した結果を紹介する。

図7. 3のモデルは原子力発電所の熱交換器建屋である。ここのフロアには、淡水の冷却水と海水との熱交換を行うために3台の熱交換器が設置されている。このフロアは地下最下階のフロアであり、他のフロアの荷重を支えるために多くの柱が並んでいる。機器の搬出のために、図7. 4の位置に搬出入ハッチが設置されており、熱交換器を搬出する場合には、この搬出入ハッチの位置まで同じフロア内をコロを用いて平面的に移動させ、搬出入ハッチからクレーンで吊り上げる。フロア内の平面移動の際には、建屋内の柱などの障害物を避けて移動する必要があり、図面上の検討などでは移動経路が確保できているかどうか分かりにくい。図7. 5は、障害物回避問題として自動的に算出させた移動経路を、熱交換器の重ね書きによって表示したもので、図7. 6はコマ送り形式で表示した結果である。なお、この熱交換器移動問題では、重心まわりの回転1自由度と並進2自由度で移動を表現し、3次元コンフィギュレーション空間で探索を実行した。回転に関しては3度おきに、並進運動に関しては熱交換器の全長の2.5パーセントの幅で量子化した。

次に全長を8.6パーセント長くした熱交換器について経路の有無をチェックした。この場合は経路は存在せず、自由空間を埋め尽くす探索手法では、8505のセルについて干渉チェックを実行し、4. 2節で述べた方法では4297セルについて干渉チェックを実行した。この場合は、障害物である柱が密に存在し、自由空間が狭くなっていると考えられるが、自由空間の境界のみを探索することによって、2倍以上の効率化が達成されている。

次の例題は、同じく原子力発電所の原子炉複合建屋において、ポンプを分解・点検する問題である。図7. 7と図7. 8はそれぞれ、ポンプの分解前と分解後の状態を示している。

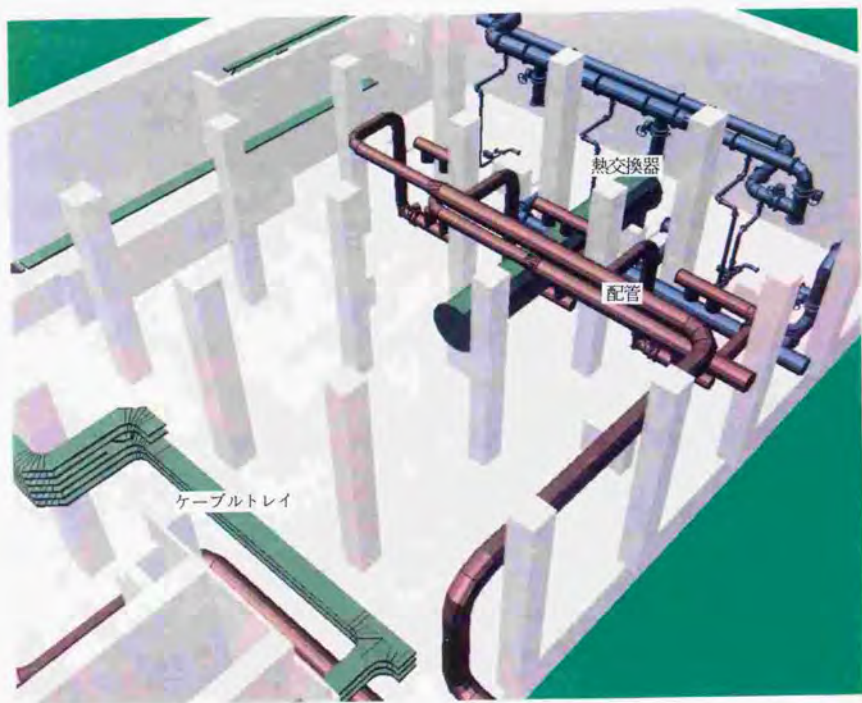


図 7. 3 熱交換器建屋

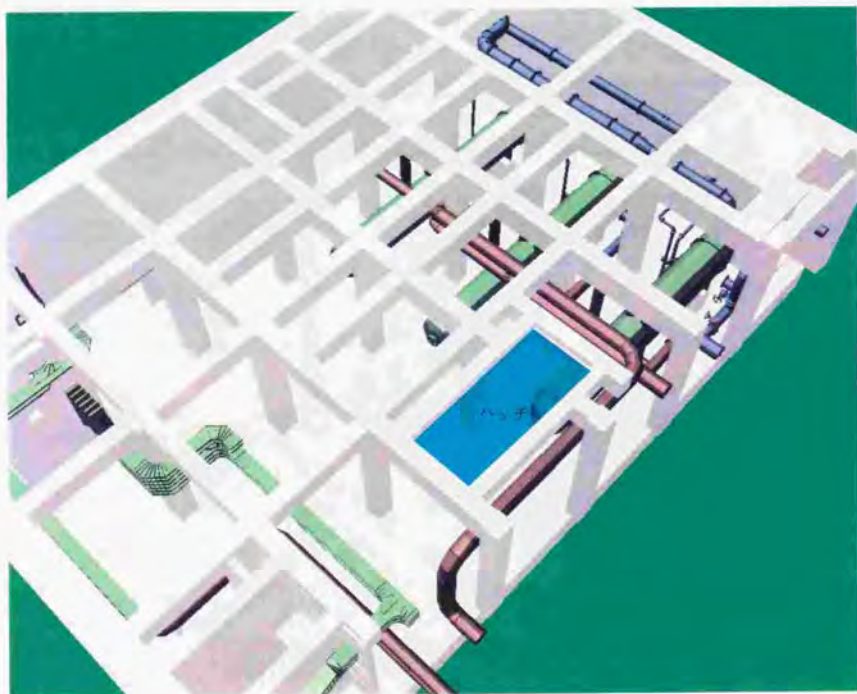


図7.4 熱交換器建屋における搬出入ハッチ

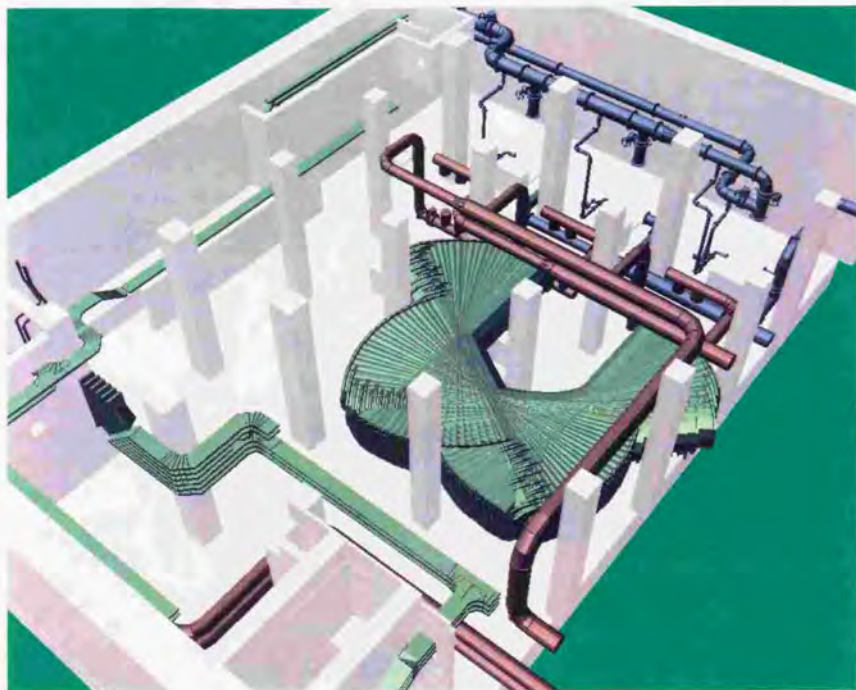


図 7. 5 熱交換器の移動経路（重ね書き表示）

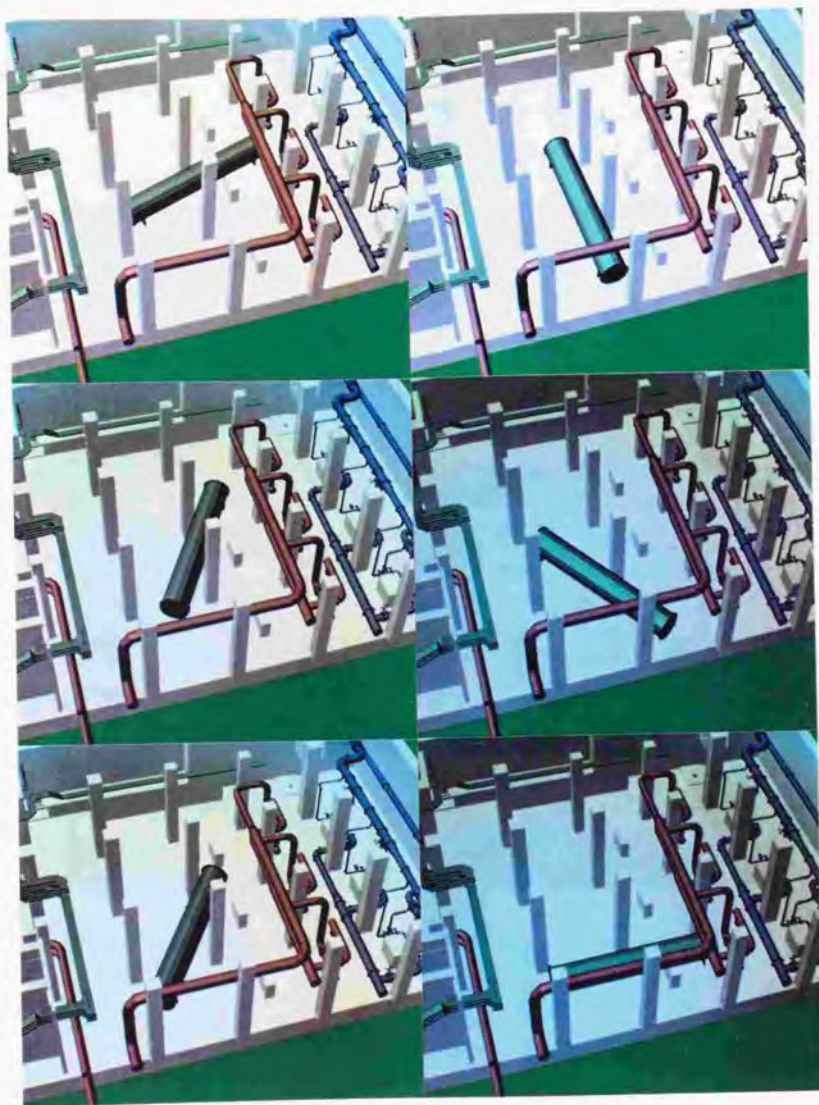


図 7. 6 熱交換器の移動経路(コマ送り表示)

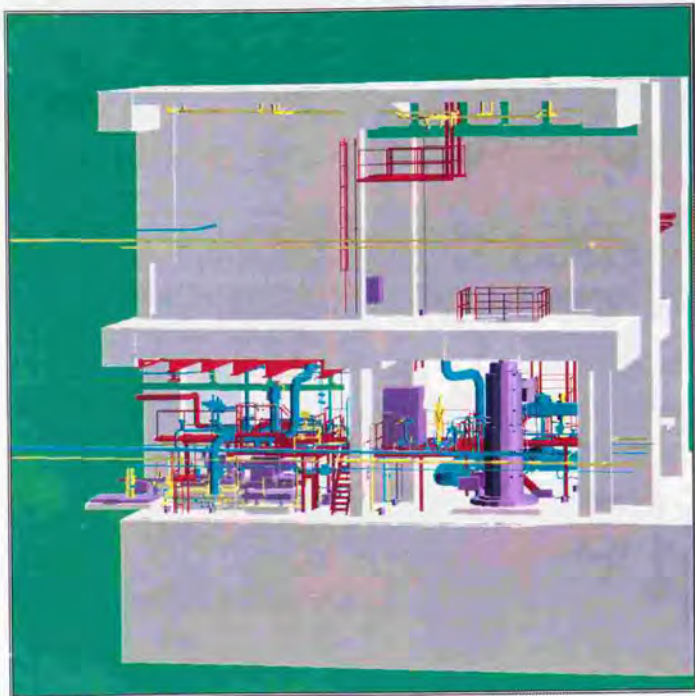


図 7. 7 ポンプ分解前の状態

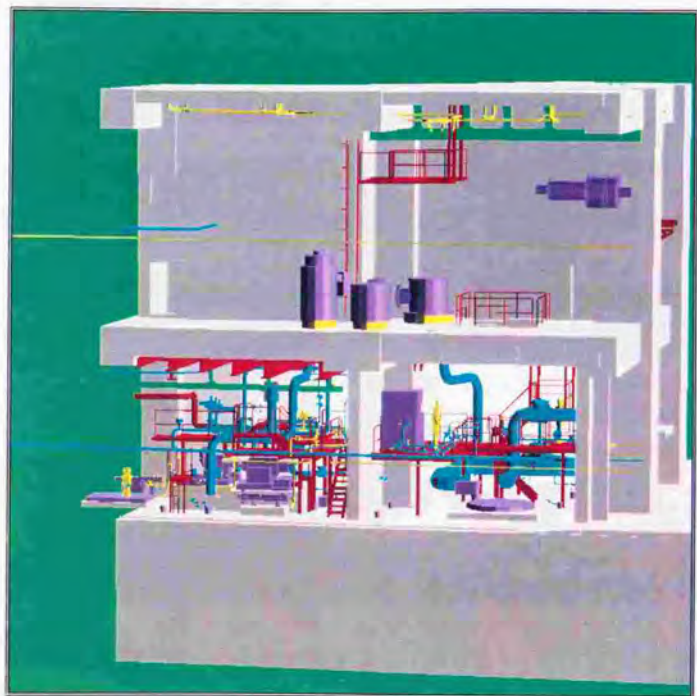


図 7. 8 ポンプ分解後の状態

このポンプは、原子炉複合建屋の最下階に設置されており、1つ上の階の天井に設置されたモノレールを用いて、点検ハッチから吊り上げる。このポンプは、上からモータ、ベダスタル、ディスチャージヘッドの順に上から分解可能であり、これらをモノレールで順番に1点吊りで引き上げて行くと、内部に垂直に設置されたポンプインターナルが現れる。図7. 8において、モノレールから吊られているのはポンプインターナルであり、このあとまっすぐ床に降ろされ、点検を行う。このように、ポンプインターナルを仮置きする場合には水平でなければならないので、ポンプインターナルの搬出はモノレールのよる2点吊りによって行う。ここでは、このポンプインターナルの吊り上げ経路を、障害物回避の問題として計画させた。

この場合は直線状のモノレールであり、ポンプインターナルの動きは平面内の3自由度の動きになる。ここでは、ポンプインターナル上部で吊っている位置に基準を置き、この点の並進2自由度とこの点まわりの回転1自由度でコンフィギュレーション空間を定義した。図7. 9には、移動経路をポンプインターナルの重ね書きによって表示したものであり、図7. 10はコマ送り形式で表示したものである。

また、このようなプラントCADへの適用に際しては、膨大な数の立体すべてと干渉チェックを実行するのは現実的ではない。そこで、ユーザにあらかじめ機器が移動する範囲を大きめの直方体で指定させ、この直方体に含まれる立体とのみ干渉チェックを実行するような方式にした。その結果、ここで紹介したような移動経路が、ワークステーションを用いた場合でも数分で算出されている。

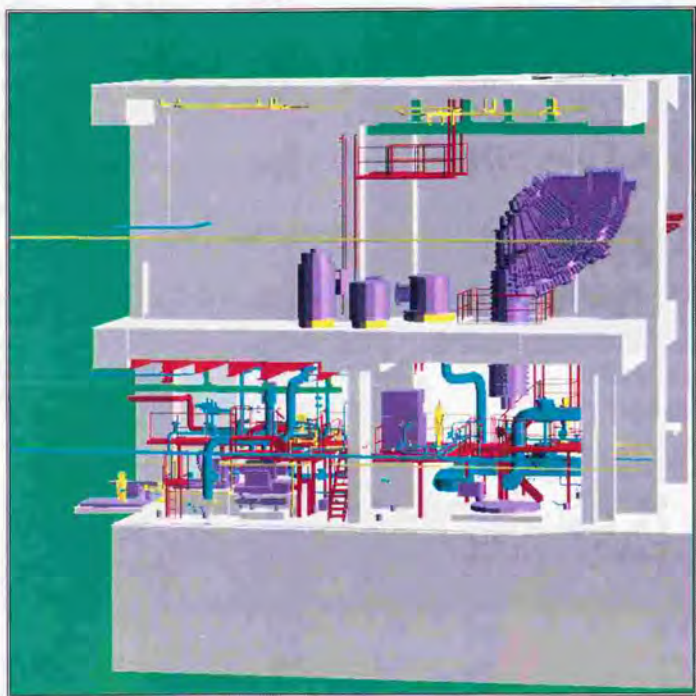


図 7. 9 ポンプインターナルの移動経路(重ね書き表示)

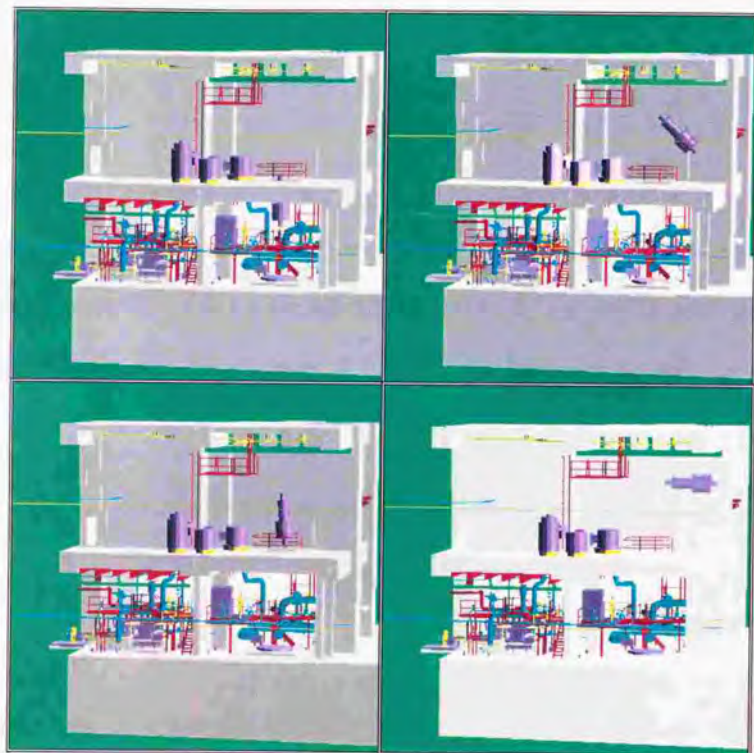


図 7. 1 0 ポンプインターナルの移動経路(コマ送り表示)

7.4 第7章のまとめ

第7章では、プラントCADシステムにおける障害物回避動作計画の役割について考え、障害物回避動作計画システムを、プラントCADシステムの一部として実際に適用した結果について紹介した。

特に、ここでは、熱交換器の搬出問題、ポンプの分解問題の2つの実際の問題についての適用結果を紹介した。

現実の原子力プラントのような場合には、非常に多くの機器や配管、架台、ダクト、トレイなどが複雑に配置されており、こういった複雑な環境下で機器の搬出入経路や点検スペースの確保は、設計作業の中でも極めて面倒な作業である。こういった問題への障害物回避動作計画の適用が、移動問題のチェックや検討の大幅な効率向上につながることで、現実の問題への適用を通じて確認された。

第8章

結論と展望

8.1 結論

本論文では、機械系CAD/CAM/CAE、生産システム、極限作業ロボット、プラント設計CADなどの幅広い分野において応用可能な動作計画の問題を取り上げた。特に、一般の多自由度の移動をする物体が、周囲の障害物との干渉を避けて、指定された初期位置・姿勢から最終位置・姿勢まで移動する経路を自動生成させる障害物回避問題を取り上げ、一般的で効率の高い新しい方法を提案した。

障害物回避動作計画の問題は、移動物体の自由度を規定するパラメータによって張られるコンフィギュレーション空間において、初期点から最終点まで点が障害物を回避して移動する問題に帰着される。しかし、高い次元のコンフィギュレーション空間を一般的に扱うのは困難であり、これまで自由度の高い場合でも効率よく移動経路を算出するのは難しかった。

本論文では、個々の移動経路を探索する際に参照されるのは広大なコンフィギュレーション空間の極く一部である点に着目し、 N 次元配列状にセル分割されたコンフィギュレーション空間において、探索によって必要な自由空間のセルを数え上げる手法を取った。その結果、これまで困難であった6次元コンフィギュレーション空間を用いる問題を含めて、一般の多自由度の移動をする物体の障害物回避動作計画を実行することが可能になった。

第1章では、生産システム、極限作業ロボット、機構設計、プラント設計などの立場から、物体の移動の表現や推論に関して、問題点や要求事項を整理した。その結果、様々なアプリケーションに対応できる柔軟なソフトウェア的な枠組みが必要であり、障害物回避動作計画問題のように基本的な問題を一般的に解く方法が強く望まれていることが明らかになった。

第2章では、話題を障害物回避動作計画問題にしぼり、これまでの研究のサーベイを行い、技術的な側面から現状の問題を明らかにした。その上で、探索によって必要な自由空間のセルを数え上げるように算出し、自由空間を限定するという、本論文の基本的な考え方を

提案した。また、形状表現や干渉チェックの問題など、障害物回避動作計画のシステムを実現するために必要な技術についても整理した。

第3章では、自由空間を限定して算出するための探索戦略について詳細に説明した。すなわち、探索の際すでに参照された自由空間の情報をもとに、経路が存在する確率の高い方向に探索の重点を変更しながら自由空間を算出する探索戦略を提案した。新たに導入したヒューリスティック関数により定義される特定の方向性を持つ探索戦略を複数用い、ヒューリスティクスを平均化するとともに、それぞれの探索戦略に投入する探索努力を制御することで探索の重点を変更する手法を提案した。さらに、初期点、最終点まわりの自由空間の状態の差を探索に反映させるために両方向探索を導入した。

第4章では、障害物回避動作計画を実行するために必要な技術で、第3章で述べなかったものについて議論した。まず、移動経路の有無を効率的にチェックするための探索戦略を提案した。次に、階層的なデータ構造を用いてコンフィギュレーション空間の情報を効率よく管理する方法について述べた。またこれに関連して、階層的に自由空間を算出する方法を提案した。最後に、干渉以外の緩やかな制約条件を考慮した経路探索と算出された経路の修正について述べた。

第5章では、本手法を実際に計算機上にインプリメントし、これまで困難とされてきた6自由度の問題について、複数の例題に適用することにより、個々の手法の効果を実験的調べた結果について述べた。また、これらの実験結果についていくつかの観点から考察を加え、本論文の提案する方法が十分な一般性と効率の高さをもっていることを確認できた。

第6章では、極限作業ロボットへの、本手法の応用について紹介した。ここでは、宇宙用マニピュレータ JEMRMS と原子炉内点検ロボットの2つの事例について述べ、本論文で提案する方法が極限作業ロボットの分野で極めて有用であることが分かった。

第7章では、プラント設計CADに本論文で提案する手法を適用した結果について述べ、プラント設計という立場からも、障害物回避動作計画が重要であり、本手法の適用によって大幅に効率が改善されることが分かった。

以上の議論から、本論文の提案する障害物回避動作計画手法は、これまで困難であった多自由度の移動をする物体の動作計画を可能にし、幅広い応用分野において極めて有用であることが確認できた。本研究の成果としてこれらをまとめると以下ようになる。

- (1) 自由空間の一部のみを限定的に参照するアルゴリズムを提案した。その結果、従来よりも高い自由度の問題をより効率的に解くことが可能になった。
- (2) 自由度が高いため、従来困難であった宇宙用マニピュレータの動作計画が自動化された。
- (3) 周囲の環境が複雑なため、従来人手に頼っていた原子炉内点検ロボットの移動経路生成が自動化された。
- (4) これまで計算機化が考慮されていなかったプラント設計における機器の搬出入の可否のチェックが自動化された。

8.2 展望

本論文で提案した手法を、さらに広い分野で応用し、一層の効果を達成するために、今後の発展性と研究課題について考えてみる。

本論文では、自由度が高い場合でも、移動経路を早く効率よく見つけるための探索戦略を中心に議論してきた。さらに、与えられた初期位置・姿勢と最終位置・姿勢の間に経路が存在しないことを証明したい場合に適用可能な探索戦略についても考察した。ただ、経路が存在しないことを示すには、どのような位置・姿勢をとっても絶対に移動できないことを示さなければならず、自由度が多い場合には、膨大な計算量を必要とすることが予想される。これは障害物回避動作計画問題がPSPACE困難であるとされている点に対応し、最大計算量が自由度の増加とともに爆発的に増大する問題を解決しなければならない。自由度が6以上となるような場合についても経路の有無をチェックするためには、基本的なアルゴリズムの点から検討しなおす必要があり、極めて難しい問題であるので、今後の基礎的な研究の大きな課題であると思われる。

一方、応用システムの面から考えると、様々なシステムとの統合化やユーザインタフェースの問題を解決して、全体として効率のよい使いやすいシステムとしなければならない。例えば、形状入力の問題については、寸法関係を幾何拘束と捉えて寸法処理機能を実現した、著者自身の形状モデリングシステム[Kondo90]について、第2章で簡単に触れた。このような様々なシステムを統合して行くことによって、それぞれの目的にあったシステムを実現できると考えられる。ただ、システム全体の問題は、目的によってその構成法が大きく異なってくるので、一般的に議論するのが難しく、本論文では触れなかった。

近年の計算機の処理能力の向上はめざましく、また、様々な応用ソフトウェアも開発されてきている。今後、これらが統合され、より高度なシステムが構築されると考えられる。こ

のような中で、障害物回避動作計画はCADを含む様々なエンジニアリングの問題にかかわるソフトウェアシステムにおいて、核となる機能の1つとして重要な役割を演じると考えられる。

[参考文献]

- [Ambler75] A.P. Ambler and R.J. Popplestone, "Inferring the positions of bodies from specified spatial relationships," *Artificial Intelligence*, vol. 6, pp.157-174, 1975.
- [Barber86] J. Barber, R. Volz, D. Desai, R. Runinfeld, B. Schipper and J. Wolter. "Automatic two-fingered grip selection." *Proc. IEEE Int. Conf. Robotics and Automation*, San Francisco, CA, pp. 890-896, 1986.
- [Barra89] J. Barraquand and J.C. Latombe, "Robot motion planning: a distributed representation approach," *Technical report STAN-CS-89-1257*, Stanford University, 1989.
- [Barra90] J. Barraquand and J.C. Latombe, "A monte-carlo algorithm for path planning with many degrees of freedom," *Proc. IEEE Int. Conf. Robotics and Automation*, Cincinnati, O., pp. 1712-1717, 1990.
- [Brady82] M. Brady, et al., *Robot Motion: Planning and Control*, Cambridge, MA: MIT Press, 1982, ch.3.
- [Brooks84] R. A. Brooks, "Planning collision free motions for pick and place operations," *Robotics Research*, Cambridge, MA: MIT Press, pp. 5-38, 1984.
- [Canny88] J. F. Canny, *The complexity of robot motion planning*, Cambridge, MA: MIT Press, 1988.
- [Coste89] M. Coste, "Effective semialgebraic geometry," *Geometry and Robotics (Lecture notes in computer science 391)* Springer-Verlag, pp. 1-27, 1989.
- [Dario89] P. Dario, "Contact sensing for robot active touch," *Robotic Science*, MIT Press, pp.138-163, 1989.
- [Donald87] B. R. Donald, "A search algorithm for motion planning with six degrees of freedom," *Artificial Intelligence*, vol. 31, pp. 295-353, 1987.
- [Dupont90] P.E. Dupont and S. Derby, "Two-phase path planning for robots with six or more joints," *Trans. ASME J. Mechanical Design*, vol. 112, pp. 50-58, 1990.
- [Faugeras89] O.D. Faugeras, "A few steps toward artificial 3-D vision," *Robotic Science*, MIT Press, pp.39-137, 1989.

- [Faverjon89] B. Faverjon and P. Tournassoud, "Motion planning for manipulators in complex environments," *Geometry and Robotics (Lecture notes in computer science 391)* Springer-Verlag, pp. 87-115, 1989.
- [Ge89] Q. Ge and J. M. McCarthy, "Equations for boundaries of joint obstacles for planar robots," *Proc. IEEE Int. Conf. Robotics and Automation*, Scottsdale, AZ, pp. 164-169, 1989.
- [Glavina90] B. Glavina, "Solving findpath by combination of goal-directed and randomized search," *Proc. IEEE Int. Conf. Robotics and Automation*, Cincinnati, O., pp. 1718-1722, 1990.
- [Grimson85] W.E. Grimson and T. Lozano-Pérez, "Search and sensing strategies for recognition and localization of two and three dimensional objects," *Prep. 3rd I.S.R.R.*, pp.81-88, 1985.
- [Hauafusa77] H. Hanafusa and H. Asada, "Stable prehension of objects by robot hand with elastic fingers," *Proc. 7th I.S.I.R.*, pp.361-368, 1977.
- [Hart68] P. E. Hart, N. J. Nilsson and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Systems Science and Cybernetics*, vol. SSC-4, no. 2, 1968.
- [Hasegawa85] T. Hasegawa, "Collision avoidance using characterized description of free space," *Proc. '85 I.C.A.R.* pp. 69-76, Tokyo, 1985.
- [Holzmann86] W. Holzmann and J.M. McCarthy, "Computing the friction forces associated with a three-fingered grasp," *Proc. IEEE Int. Conf. Robotics and Automation*, San Francisco, CA, pp. 594-600, 1986.
- [Iba88] H. Iba, H. Matsubara and H. Inoue, "View and visibility for managing incomplete environment models," *Proc. IEEE Intern. Workshop on Intelligent Robots and Systems (IROS'88)*, pp. 411-416, Tokyo, 1988.
- [Jameson86] J.W. Jameson and L.J. Leifer, "Quasi-static analysis: a method for predicting grasp stability," *Proc. IEEE Int. Conf. Robotics and Automation*, San Francisco, CA, pp. 890-896, 1986.

- [Jennings89] J. Jennings, B. Donald and D. Campbell, "Towards experimental verification of an automated compliant motion planner based on a geometric theory of error detection and recovery," *Proc. IEEE Int. Conf. Robotics and Automation*, Scottsdale, AZ, pp. 632-637, 1989.
- [Kambhampati86] S. Kambhampati and L. S. Davis, "Multiresolution path planning for mobile robots," *IEEE J. Robotics and Automation*, vol. RA-2, no. 3, pp. 135-145, 1986.
- [Kameyama90] K. Kameyama, K. Kondo and K. Ohtomi, "An intelligent interactive layout CAD system for industrial plants," *Proc. ASME 2nd Int. Conf. Design Theory and Methodology*, pp. 33-38, Chicago, 1990.
- [Khatib86] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Intern. J. Robotics Research*, vol. 5, no. 1, 1986.
- [Kondo88] K. Kondo, "A simple motion planning algorithm using heuristic free space enumeration," *Proc. IEEE Intern. Workshop on Intelligent Robots and Systems (IROS'88)*, pp. 751-756, Tokyo, 1988.
- [Kondo89] K. Kondo and F. Kimura, "Collision avoidance using a free space enumeration method based on grid expansion," *Advanced Robotics*, vol. 3, no. 3, pp.159-175, 1989.
- [Kondo90] K. Kondo, "PIGMOD: parametric and interactive geometric modeller for mechanical design," *Computer Aided Design*, vol. 22, no. 10, pp. 633-644, 1990.
- [Kondo91a] K. Kondo, "Motion planning with six degrees of freedom by multi-strategic bi-directional heuristic free space enumeration," *IEEE Trans. Robotics and Automation*, vol. 7, no. 3, pp. 267-277, 1991.
- [Kondo91b] K. Kondo and K. Ohtomi, "Motion planning in plant CAD systems," *Proc. ASME Design Automation Conf.*, pp. 195-202, Miami, 1991.
- [Laugier86] C. Laugier and J. Pertin-Troccez, "SHARP: A system for automatic programming of manipulation robots," *Prep. 3rd I.S.R.R.*, pp. 118-125, 1986.
- [Laugier89] C. Laugier, "Planning fine motion strategies by reasoning in the contact space," *Proc. IEEE Int. Conf. Robotics and Automation*, Scottsdale, AZ, pp. 653-659, 1989.

- [Lewis73] R.A. Lewis and A.K. Bejczy, "Planning considerations for a roving robot with arm," *Proc. 3rd IJCAI*, pp. 308-316, 1973.
- [Lieber77] L.I. Lieberman and M.A. Wesley, "AUTOPASS: An automatic programming system for computer controlled mechanical assembly," *IBM J. Res. and Develop.*, vol. 21, no. 4, pp. 321-333, 1977.
- [Lozano81] T. Lozano-Pérez, "Automatic planning of manipulator transfer movement," *IEEE Trans. Sys. Man Cybernetics*, vol. 11-10, pp. 681-698, 1981.
- [Lozano83] T. Lozano-Pérez, "Spatial planning: A configuration space approach," *IEEE Trans. Comput.* vol. C-32, pp. 108-120, Feb. 1983.
- [Lozano84] T. Lozano-Pérez, M.T. Mason and R.H. Taylor, "Automatic synthesis of fine motion strategies for robots," *Int. J. Robotics Research*, vol. 3, no. 1, pp. 3-24, 1984.
- [Lozano85] T. Lozano-Pérez and R.A. Brooks, "An approach to automatic robot programming," *M.I.T. A.I. memo*, no. 842, 1985.
- [Lozano87] T. Lozano-Pérez, "A simple motion-planning algorithm for general robot manipulators," *IEEE J. Robotics and Automation*, vol. RA-3, no. 3, pp. 224-238, 1987.
- [Lozano89] T. Lozano-Pérez and R.H. Taylor, "Geometric issues in planning robot tasks," *Robotic Science*, MIT Press, pp.227-261, 1989.
- [Lumelsky87] V. J. Lumelsky, "Effect of kinematics on motion planning for planer robot arms moving amidst unknown obstacles," *IEEE J. Robotics and Automation*, vol. RA-3, no. 3, pp. 207-223, 1987.
- [LumStep87] V. J. Lumelsky and A.A. Stepanov, "Path planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape," *Algorithmica*, vol. 2, pp. 403-430, 1987.
- [Lumelsky88] V. J. Lumelsky, "Motion planing for robot arm manipulators with proximity sensing," *Proc. IEEE Int. Conf. Robotics and Automation*, Philadelphia, 1988.

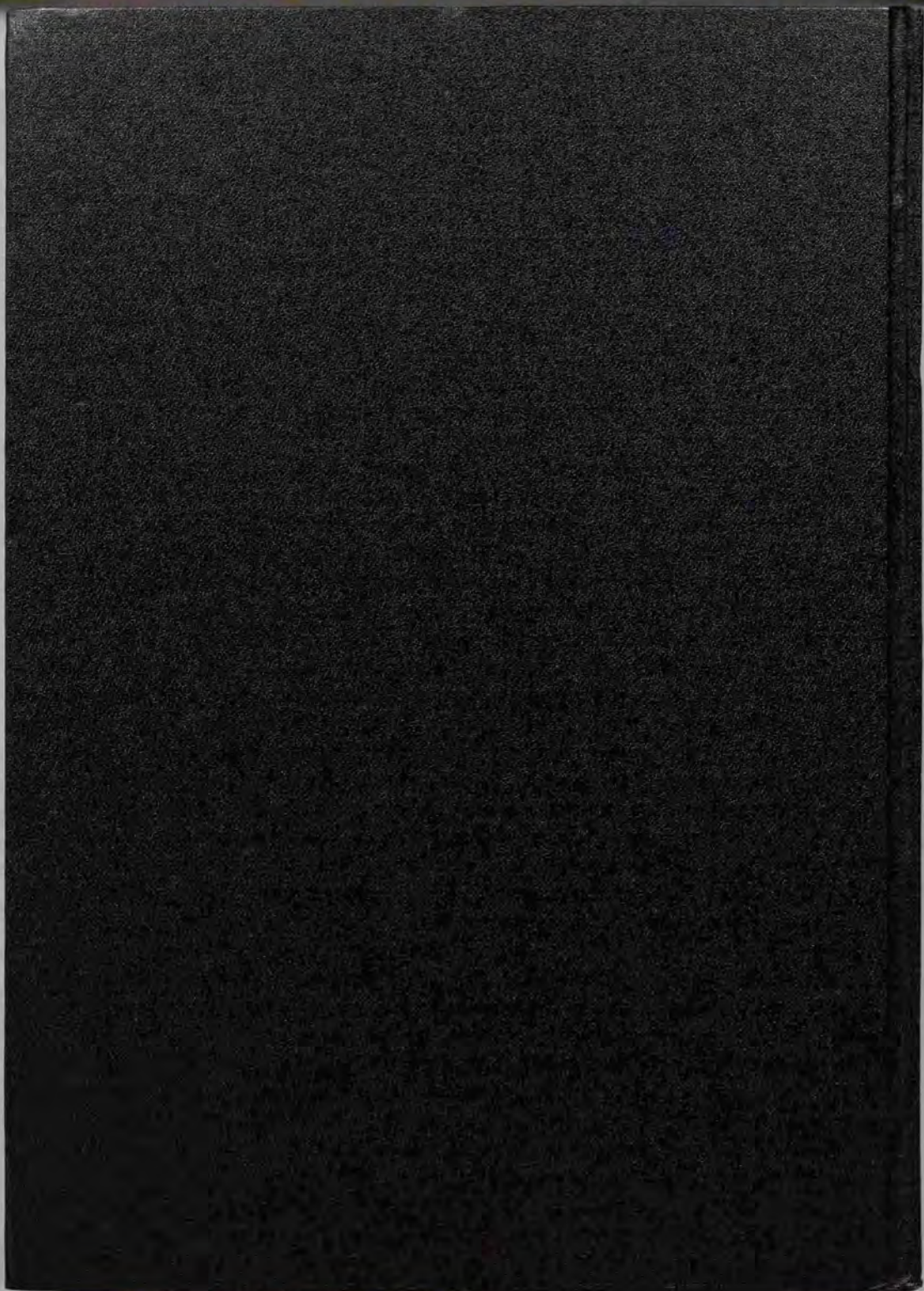
- [Marchard89] J. Marchard "The algorithm by Schwartz, Sharir and Collins on the piano mover's problem," *Geometry and Robotics (Lecture notes in computer science 391)* Springer-Verlag, pp. 49-66, 1989.
- [Maityla88] M. Mäntylä, *An Introduction to solid modeling*, Computer Science Press, 1988.
- [Nilsson82] N. K. Nilsson, *Principles of Artificial Intelligence*, Palo Alto, CA: Tioga Press, 1982, ch. 2.
- [Orlandea77] N. Orlandea, M.A. Chace and D.A. Calahan, "A sparsity-oriented approach to the dynamic analysis and design of mechanical systems," *Trans. ASME J. Engineering for Industry*, vol. 99-3, pp. 773-784, 1977.
- [Popple78] R.J. Popplestone et al., "RAPT: a language for describing assemblies," *Industrial Robot*, vol. 5, No. 3, pp. 131-137, 1978.
- [Popple80] R.J. Popplestone, "An interpreter for a language for describing assemblies," *Artificial Intelligence*, vol. 14, pp.79-107, 1980.
- [Requicha80] A.A.G. Requicha, "Representation for rigid solids: theory, methods and systems," *ACM Computing Surveys*, vol. 12, no. 4, pp. 437-464, 1980.
- [Rimon89] E. Rimon and D. E. Koditschek, "The construction of analytic diffeomorphisms for exact robot navigation on star world," *Proc. IEEE Int. Conf. Robotics and Automation*, Scottsdale, AZ, pp. 21-26, 1989.
- [Roller89] D. Roller, F. Schonek and A. Verroust, "Dimension-driven geometry in CAD: a survey," *Theory and Practice of Geometric Modeling*, Springer-Verlag, 1989.
- [Schwartz83] J.T. Schwartz and M. Sharir, "On the piano mover's problem II. General techniques for computing topological properties of real algebraic manifolds," *Advances in Applied Mathematics*, vol. 4, pp. 298-351, 1983.
- [Schwartz88] J.T. Schwartz, "A survey of motion planning and related geometric algorithms," *Artificial Intelligence*, vol. 37, pp. 157-169, 1988.
- [Sheth72] P.N. Sheth and J.J. Uicker Jr. "IMP (Integrated Mechanism Program) - A computer aided design analysis system for mechanism and linkages," *Trans. ASME J. Engineering for Industry*, vol. 94-2, pp. 454-464, 1972.

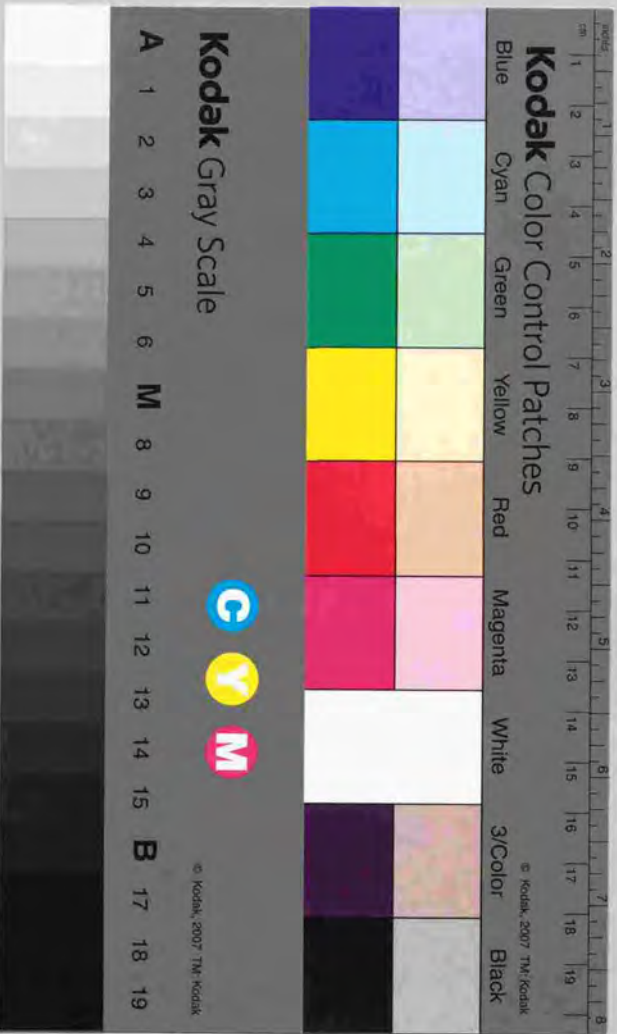
- [Udupa77] S.M. Udupa, "Collision detection and avoidance in computer controlled manipulators," *Proc. 5th IJCAI*, pp. 737-748, 1977.
- [Weiler86] K.J. Weiler, "Topological structures for geometric modelling," *Ph.D Thesis*, Rensselaer Polytechnic Institute, 1986.
- [Whitney79] D.E. Whitney and J.L. Nevis, "What is RCC and What can it do?," *Proc. 9th L.S.I.R.*, pp. 135-152, 1979.
- [Yamawaki89] K. Yamawaki, K. Kuraoka, M. Shigehara, K. Goma, T. Sumi and R. Okamura, "Design and evaluation of man-in-the-loop control system of Japanese Experimental Module Remote Manipulator System," *Proc. 40th Congress Inter. Astronautical Federation*, Malaga, Spain, 1989.
- [浅田 90] 浅田, "センサ情報の統合と理解による移動ロボットのための世界モデルの構築", 日本ロボット学会誌, vol. 8, no. 2, pp.28-38, 1990.
- [石川 88] 石川, "センサフュージョンシステム", 日本ロボット学会誌, vol. 6, no. 3, pp. 79-83, 1988.
- [井上 84] 井上, "ロボット言語の研究課題", 日本ロボット学会誌, vol. 2, no. 2, pp. 3-6, 1984.
- [狼 89] 狼, "宇宙ロボットプロジェクト", 日本ロボット学会誌, vol. 7, no. 1, pp. 78-86, 1989.
- [大村 80] 大村, "統計解析のはなし", 日科技連, 1980.
- [小沢 86] 小沢, 熊本, 明石, 中田, "オフラインロボット教示における高速干渉チェックの一方式", 日本ロボット学会誌, vol. 4, no. 2, pp. 5-14, 1986.
- [黒岩 88] 黒岩, "ロボットの機構設計用シミュレーションシステム", 日本ロボット学会誌, vol. 6, No. 2, pp. 65-71, 1988.
- [近藤 87] 近藤, 木村, "迷路法に基づく自由空間算出法を用いた障害物回避動作計画", 日本ロボット学会誌, vol. 5, No. 4, pp. 263-272, 1987.
- [近藤 88] 近藤, "ヒューリスティックなグラフ探索にもとづく自由空間算出法による障害物回避動作計画", 日本ロボット学会誌, vol. 6, no. 6, pp. 489-498, 1988.
- [近藤 89] 近藤, "複数探索戦略を用いた自由空間算出による障害物回避動作計画", 日本ロボット学会誌, vol. 7, no. 4, pp. 352-362, 1989.

- [嶋田 88] 嶋田, "ジオメトリックシミュレーションにおける衝突検出アルゴリズム", 精密工学会誌, vol. 54, no. 8, pp.1562-1567, 1988.
- [白井 87] 白井, 井上, "知能ロボット研究の展望 (モデルベースト・ロボティクス)", 日本ロボット学会誌, vol. 5, no. 6, pp. 34-41, 1987.
- [末広 88] 末広, 高瀬, "接触運動の表現と制御およびその組立作業への応用", 日本ロボット学会誌, vol. 6, no. 6, pp. 499-506, 1988.
- [多田 73] 多田, "探索理論", 日科技連, 1973.
- [坪内 89] 坪内, 油田, "移動ロボット用視覚のための建物内環境地図の表現法と視野像の生成", 日本ロボット学会誌, vol. 7, no. 3, pp. 87-98, 1989.
- [成川 88] 成川, 山本, 佐々木, "原子力プラント自動配管エキスパートシステム", 計測と制御, vol. 27, no. 10, pp.899-900, 1988.
- [登尾 87] 登尾, 福田, 有本, "オクトツリーを用いた高速干渉チェック法", 日本ロボット学会誌, vol. 5, no. 3, pp. 21-30, 1987.
- [登尾 89] 登尾, 有本, "オクトツリーを利用したロボットと障害物の近接2点決定アルゴリズム", 日本ロボット学会誌, vol. 7, no. 3, pp. 39-48, 1989.
- [長谷川 86] 長谷川, "自由空間分類表現法によるマニピュレータの衝突回避動作の計画", 計測自動制御学会論文集, vol. 22, no. 6, pp. 20-26, 1986.
- [比留川 87] 比留川, 北村, "安全第一アルゴリズムとポテンシャル関数に基づくマニピュレータの障害物回避", 日本ロボット学会誌, vol. 5, no. 3, pp. 3-11, 1987.
- [山脇 89] 山脇, 胡間, 津田, "宇宙ステーション用リモートマニピュレータシステム", 東芝レビュー, vol. 44, no. 9, pp. 710-714, 1989.
- [湯浅 87] 湯浅, 伊藤, "マニピュレータの障害物回避動作計画を考慮した凸多面体間の干渉チェック法", 日本ロボット学会誌, vol. 5, no. 4, pp. 39-44, 1987.
- [劉 89] 劉, 登尾, 有本, "移動物体間の干渉が効率的にチェックできるソリッドモデル HSM の提案", 日本ロボット学会誌, vol. 7, no. 5, pp. 26-34, 1989.

[論文リスト]

- (1) 近藤, 木村, "迷路法に基づく自由空間算出法を用いた障害物回避動作計画", 日本ロボット学会誌, vol. 5, no. 4, pp. 263-272, 1987.
- (2) K. Kondo, "A simple motion planning algorithm using heuristic free space enumeration," *Proc. IEEE Intern. Workshop on Intelligent Robots and Systems (IROS'88)*, pp. 751-756, Tokyo, 1988.
- (3) 近藤, "ヒューリスティックなグラフ探索にもとづく自由空間算出法による障害物回避動作計画", 日本ロボット学会誌, vol. 6, no. 6, pp. 489-498, 1988.
- (4) K. Kondo and F. Kimura, "Collision avoidance using a free space enumeration method based on grid expansion," *Advanced Robotics*, vol. 3, no. 3, pp.159-175, 1989.
- (5) 近藤, "複数探索戦略を用いた自由空間算出による障害物回避動作計画", 日本ロボット学会誌, vol. 7, no. 4, pp. 352-362, 1989.
- (6) K. Kondo, "PIGMOD: parametric and interactive geometric modeller for mechanical design," *Computer Aided Design*, vol. 22, no. 10, pp. 633-644, 1990.
- (7) K. Kondo, "Motion planning with six degrees of freedom by multi-strategic bi-directional heuristic free space enumeration," *IEEE Trans. Robotics and Automation*, vol. 7, no. 3, pp. 267-277, 1991.
- (8) K. Kondo and K. Ohtomi, "Motion planning in plant CAD systems," *Proc. ASME Design Automation Conf.*, pp. 195-202, Miami, 1991.
- (9) K. Kondo, "Collision avoidance by free space enumeration based on heuristic graph search," *Advanced Robotics*, vol. 5, no.3, pp. 293-308, 1991.
- (10) K. Kondo, "Collision avoidance by free space enumeration using multiple search strategies," *Advanced Robotics*, vol. 5, no.4, pp. 411-427, 1991.
- (11) K. Kondo, "Algebraic method for manipulation of dimensional relationships in geometric models," *Computer Aided Design*, vol. 24, no. 3, pp. 141-147, 1992.





Kodak Color Control Patches

Blue Cyan Green Yellow Red Magenta White 3/Color Black

Kodak Gray Scale

A 1 2 3 4 5 6 M 8 9 10 11 12 13 14 15 B 17 18 19



© Kodak, 2007 TM Kodak