

名古屋大学 学位論文

可変分解能処理による  
移動ロボットの運動特性を考慮した  
ロードマップ生成

名古屋大学 大学院 工学研究科

機械理工学専攻 電子機械工学分野

メカトロニクス制御講座 モビリティシステムグループ

項 警宇

平成 25 年



# 目次

第 1 章	序論	1
1.1	本研究の背景	1
1.2	移動ロボットにおける地図	2
1.2.1	経路計画のための地図表現	2
1.2.2	ロードマップにおける地図表現の分解能	3
1.2.3	ロードマップの分解能と経路計画	4
1.2.4	運動特性を考慮したロードマップの必要性	5
1.2.5	ロードマップを用いた経路計画における課題	6
1.2.6	物理情報量によるロードマップの拡張	6
1.3	本研究の目的	7
1.4	本論文の構成	8
第 2 章	グラフとグラフ探索	9
2.1	グラフ構造	9
2.2	グラフ探索	10
2.2.1	問題の定式化	10
2.2.2	グラフ探索手法	10
2.2.3	実時間グラフ探索手法	12
第 3 章	未知環境下での移動ロボットの運動特性に基づいた地図の自律生成	17
3.1	はじめに	17
3.2	問題設定および提案手法の概要	18

## 目次

---

3.2.1	問題設定	18
3.2.2	提案手法の概要	19
3.3	提案手法	22
3.3.1	可変分解能処理に基づくセル半径の決定	22
3.3.2	ノード及びリンクの生成手法	24
3.3.3	パラメータに関する考察	26
3.4	経路探索	27
3.5	検証実験	30
3.5.1	数値シミュレーション	30
3.5.2	実機実験	32
3.6	まとめ	36
第4章	全方向移動ロボットの運動特性に基づく可変分解能速度マップ生成	37
4.1	はじめに	37
4.2	想定するシステム	38
4.2.1	作業空間上のロードマップ	38
4.2.2	全方向移動ロボットのモデルと制御則	38
4.2.3	移動に関する制約条件	41
4.3	可変分解能速度マップ生成手法	42
4.3.1	速度情報を持つ拡張ロードマップの定義	43
4.3.2	実行可能条件の評価	44
4.3.3	可変分解能速度空間分割	45
4.4	実行可能領域の厳密な導出	47
4.4.1	移動時間	48
4.4.2	衝突回避	48
4.4.3	速度制限	49
4.4.4	加速度制限	51
4.5	数値例	53
4.6	まとめ	58



第 5 章	車両型移動ロボットの運動特性に基づく可変分解能姿勢マップ生成	59
5.1	はじめに	59
5.2	想定するシステム	60
5.2.1	作業空間上のロードマップ	60
5.2.2	車両型移動ロボットのモデルと制御則	61
5.2.3	軌道生成	62
5.2.4	移動に関する制約条件	67
5.3	可変分解能姿勢マップ生成手法	68
5.3.1	姿勢情報を持つ拡張ロードマップの定義	68
5.3.2	実行可能条件の評価	70
5.3.3	可変分解能姿勢空間分割	72
5.4	実行可能領域の導出	74
5.4.1	障害物回避	75
5.4.2	速度及び加速度制約	82
5.5	数値例	83
5.6	自動駐車システム	87
5.7	まとめ	90
第 6 章	結論	93
6.1	本研究の成果	93
6.2	今後の課題および展望	94
参考文献		
付録		
A.1	全方向移動ロボットの実行可能領域の導出例	107
A.1.1	衝突回避	107
A.1.2	速度制限	111
A.1.3	加速度制限	111
A.2	車両型移動ロボットの実行可能領域の導出例	112
A.2.1	衝突回避の $y$ 軸方向	112

A.2.2 旋回最小半径に関する実行可能領域 . . . . . 113

謝辞

# 目次

2.1	Illustration of RTA* : (a) $g(v, v_{d+1})$ is the cost from $v$ to $v_{d+1}$ and $\hat{h}(v_{d+1})$ is the heuristic cost from $v_{d+1}$ to $v_D$ , (b) $v^*$ is calculated by (2.3) and $\hat{h}(v_c)$ is updated by (2.5). . . . .	15
2.2	An example of RTA* . . . . .	16
3.1	Mobile robot for experiment . . . . .	18
3.2	Illustration of workspace and proposed graph-map . . . . .	19
3.3	Flowchart of overall procedure . . . . .	21
3.4	Measured distance $D^{(n)}(\theta)$ and cell radius $R^{(n)}$ . . . . .	22
3.5	Node generation rules . . . . .	25
3.6	Link generation rule . . . . .	26
3.7	Decision indicator of $R_{\min}$ and $R_{\text{th}}$ . . . . .	27
3.8	Illustration of path planning using RTA* : (a) $g(n, n_{d+1})$ is the cost from $n$ to $n_{d+1}$ and $\hat{h}(n_{d+1})$ is the heuristic cost from $n_{d+1}$ to a goal position $q_D$ , (b) $n^*$ is calculated by (3.21) and $\hat{h}(n_c)$ is updated by (3.23). . . . .	29
3.9	Result of simulation . . . . .	32
3.10	Graph-map constructed by simulation in large environment . . . . .	33
3.11	Environment for experiment using an autonomous mobile robot . . . . .	34
3.12	View of environment for the experiment . . . . .	34
3.13	Graph-map constructed by the mobile robot (a,b) and path of the mobile robot using path-planning on the graph-map with the search depth set as 5 (c) . . . . .	35

4.1	Illustration of extension a configuration roadmap to a variable-resolution velocity roadmap : green parts depict obstacles in the $(x, y, v)$ space. Note that obstacles in the $(x, y, v)$ space can not calculate easily. . . . .	39
4.2	Definition of information between two points . . . . .	40
4.3	Illustration of Variable-resolution Velocity Roadmap . . . . .	43
4.4	Condition for a directed link generation . . . . .	44
4.5	Examples of feasible region . . . . .	45
4.6	An example of variable-resolution partitioning of velocity space . . . . .	46
4.7	Flowchart of overall procedure . . . . .	46
4.8	A feasible velocity region satisfying condition (4.11) . . . . .	50
4.9	A feasible velocity region satisfying condition (4.12) . . . . .	50
4.10	A Feasible velocity region satisfying an acceleration condition (4.17): the hatched rectangular region depict $\tilde{\mathcal{F}}_v(q, q')$ . . . . .	52
4.11	A Feasible velocity region satisfying an acceleration condition (4.19): the hatched rectangular region depict $\tilde{\mathcal{F}}_v(q, q')$ . . . . .	52
4.12	An example of dividing velocity space and directed link generation . . . . .	54
4.13	An environment for experiment and a VVR constructed by the proposed method . . . . .	55
4.14	Path planning using VVR (Fig. 4.13(b)) and trajectories . . . . .	56
4.15	A velocity roadmap with constant resolution 6 and the optimal trajectory . . . . .	57
5.1	Definition of information between two positions . . . . .	62
5.2	Model of a car-like vehicle . . . . .	63
5.3	Trajectory map based on $\theta$ and $\theta'$ . . . . .	63
5.4	Forward trajectory generation . . . . .	64
5.5	Forward-Forward trajectory generation . . . . .	66
5.6	Illustration of Variable-resolution Orientation Roadmap . . . . .	69
5.7	Feasible trajectories and Infeasible trajectories for car-like robots . . . . .	69

---

5.8	Relation between a local coordinate system and a global coordinate system for orientation . . . . .	71
5.9	Examples of feasible region . . . . .	72
5.10	An example of variable-resolution partitioning of orientation space . . . . .	73
5.11	Flowchart of overall procedure . . . . .	73
5.12	A region about constraint condition of y-axis with $\theta_{pp'} + \theta_A \geq 0$ . . . . .	76
5.13	A region about constraint condition of y-axis with $\theta_{pp'} + \theta_A < 0$ . . . . .	76
5.14	A region about constraint condition of y-axis . . . . .	78
5.15	A region about constraint condition of x-axis with $\theta_{pp'} + \theta_A \geq 0$ . . . . .	79
5.16	A region about constraint condition of x-axis with $\theta_{pp'} + \theta_A < 0$ . . . . .	80
5.17	A region about constraint condition of x-axis . . . . .	81
5.18	An example of dividing orientation space and directed link generation . . . . .	83
5.19	An example of dividing orientation space and directed link generation . . . . .	85
5.20	Path planning with $(0, 0, 0)$ to $(500, 0)$ using VOR (Fig. 5.19(f)) . . . . .	86
5.21	Path planning with $(0, 0, \frac{\pi}{2})$ to $(500, 0)$ using VOR (Fig. 5.19(f)) . . . . .	86
5.22	Trajectory map with reverse trajectory based on $\theta$ and $\theta'$ . . . . .	87
5.23	Parking lot environment with a roadmap . . . . .	88
5.24	A Variable-resolution Orientation Roadmap for Fig. 5.23 . . . . .	88
5.25	A parking trajectory by reverse parking from $(q_2, \pi)$ to $(q_2^r, \frac{\pi}{2})$ . . . . .	89
5.26	A parking trajectory by forward parking from $(q_0, \pi)$ to $(q_0^f, -\frac{\pi}{2})$ . . . . .	90
5.27	An experiment result using an electric car . . . . .	91
A.1	Feasible region with two $R_{\min}$ . . . . .	114
A.2	Selection of feasible region with $R_{\min} \leq 0.20d(q, q')$ . . . . .	115
A.3	Selection of feasible region with $0.20d(q, q') < R_{\min}$ . . . . .	116



# 表目次

3.1	Definitions of terms and symbols . . . . .	20
3.2	Condition settings for $R_{\min}$ with $0 < R_{\text{th}} < d_r/2$ . . . . .	31
3.3	Condition settings for $R_{\text{th}}$ with $d_r/2 \leq R_{\min} < d_r$ . . . . .	31
4.1	Variable resolution velocity roadmap and constant resolution velocity roadmap . . . . .	58





# 第 1 章

## 序論

本章では，自律移動ロボットの走行安全性に基づいた地図を生成する研究を行うことの背景と意義を述べ，本研究の目的を示す．また，第 2 章以降の各章の構成について示す．

### 1.1 本研究の背景

近年，惑星探査，放射線危険区域や医療福祉などで移動ロボットの需要が高まっている．特に惑星探査や放射線危険区域などの多くの場所では，レールなどの専用インフラを整備することが困難であるため，ロボットが自律的に安全に移動することが求められている．

ロボットが自律移動を行うためには，対象とする環境をモデル化した地図が必要である．地図を用いないナビゲーション手法 [1], [2] もあるが，同じ作業空間上を繰り返し移動する場合は，情報の再利用の観点から地図を用いる方が有利である．地図を利用する目的には主として経路計画と自己位置推定が挙げられる．未知環境において地図生成と自己位置推定を同時並行的に行う問題は SLAM (Simultaneous Localization and Mapping) と呼ばれ，多くの研究成果が報告されている [3] ~ [8]．しかしながら，SLAM の枠組みにはロボットの移動経路の決定は含まれておらず，操作者により外部から指示される場合が多い．SLAM に対して，効率の良い地図生成を目的とした経路計画問題は NBV (Next Best View) と呼ばれ，多くの研究成果が報告されている [9] ~ [12]．しかしながら，これらの研究は精確な地図生成もしくは効率の良い地図生成を目的としているため，移動ロボットの経路計画のための地図を生成しているわけではない．

ロボットが移動を行う際，安全に移動する必要がある．ここで，安全な移動とは障害物に衝突せずに移動することを指す．地図に安全性に関する情報が含まれていない場合，経路計画の段階で安全性を考慮した計画を行うことは不可能である．そのため，オンラインで走行安全性を考慮して再計画をする必要がある．この問題に対して，安全性を考慮した

観測を行う手法が文献 [13] で提案されているが，オンラインで経路計画と同時に観測の計画を行うために計算コストが大きくなる．一方，文献 [14] ではオフラインで経路と観測地点の両方を計画する手法が提案されている．このように走行安全性を考慮した計画手法は提案されている．しかしながら，これらの手法では移動ロボットの走行安全性を考慮した地図を用いていないために，計画に必要な計算コストが大きくなる．以上より，本研究では移動ロボットの走行安全性を考慮した地図表現に着目し，走行安全性を考慮した地図の生成手法に取り組む．

## 1.2 移動ロボットにおける地図

本論文での地図とは環境を形式的にモデル化したものである．本節ではモデル化の概要について述べる．

### 1.2.1 経路計画のための地図表現

移動ロボットの経路計画問題を考慮した環境の地図表現方法で主に用いられている手法にはロードマップ法 (Roadmap Method) と人工ポテンシャル法 (Artificial Potential Fields) がある．ロードマップ法は移動ロボットの行動に直結するように環境にグラフを配置する手法であり [15] ~ [20]，人工ポテンシャル法は環境を場で表現する手法である．本節では両手法の概要について述べる [21]．なお，グラフの詳細は第 2 章で説明する．

#### ロードマップ法

ロードマップ法とは，作業環境内の位置を表現する頂点 (ノード) と位置の繋がりを表現する辺 (リンク) で構成されるグラフを用いて作業環境を表現する手法である．ロードマップの生成手法としてセル分割法，ポロノイ図法，可視グラフ，PRM (Probabilistic Roadmap Method) および RRT (Rapidly-Exploring Random Tree) などが提案されている [22]．セル分割法 [23] では作業空間を格子状に分割し，各領域をノードとし，隣り合う領域間にリンクを生成している．ポロノイ図法 [24] ~ [28] では空間を最近接障害物によって領域分割を行い，領域の境界をポロノイ境界とし，境界の交点をポロノイ点とする．このポロノイ点をノードとし，ポロノイ境界で連結するノード間にリンクを生成する．可視グラフ法 [29], [30] では，障害物の頂点にノードを生成し，障害物と交差しないノード間をリンクで連結する．これら 3 種類の生成手法は作業環境の詳細な幾何情報を用いた手法である．PRM [31] と RRT [32] では自由空間内にランダムでノードを生成し，近接するノード間をリンクで連結することでロードマップを生成している．ただし，PRM

では環境全体の地図を生成し，RRT では所定の初期位置から目標位置までの地図のみを生成している．これら 2 種類の生成手法は作業環境を詳細に表現しないが，前記の 3 手法よりも複雑な環境に適用できる．ロードマップ法の利点としては，経路計画としてグラフ探索手法を利用できることが挙げられる．また，未知環境に対して逐次的にロードマップを生成しながら目標位置に移動する手法が提案されている [33] ~ [36] ．

### 人工ポテンシャル法

人工ポテンシャル法とは，障害物を回避する斥力ポテンシャルと目標位置に引き寄せられる引力ポテンシャルによる場を用いて作業環境を表現する手法である [37] ．その場の勾配ベクトルに基づいた極値探索法を適用することにより，オンラインで経路計画を行える．グラフ探索的なアプローチに比べてアルゴリズムが簡単で実時間性に優れた障害物回避が可能であるが，ポテンシャルに極小値が存在する場合には，その位置から抜け出せないという問題がある．この問題に対して，極小値を持たない調和関数を用いた手法が提案されている [38] ~ [40] ．また，画像の中の特定の領域を抽出し，その時間変化をトラッキングする方法の 1 つであるレベルセット法を応用した手法も提案されている [41] ~ [43] ．人工ポテンシャル法の利点としては，オンラインでロボットの速度の計画も可能であることが挙げられる．しかしながら，場の生成が非常に重要であり，未知環境で逐次的に生成することが困難であることが欠点である ．

上記の性質より，未知環境で移動ロボットが地図生成と経路計画を並行して行うにはロードマップ法の方が適している．オンラインでの経路計画は実時間グラフ探索の利用もしくはサブゴールの設定で可能になる．また，オフラインでの計画可能という点を考慮して，本研究ではロードマップ法を用いるとする．以降の議論はロードマップ法で行う ．

### 1.2.2 ロードマップにおける地図表現の分解能

一般に地図は環境の複雑さに合わせて適切な分解能で表現される必要がある．分解能とは地図表現の精度であり，分解能を高くすることで精度の高い地図を生成できる．ロードマップにおけるノードの配置密度が地図の精度に該当する．地図の精度は移動の安全性，記憶領域，計算時間および観測精度と関係する [44], [45] ．ノードの配置密度が高い高精度な地図は作業環境を詳細に表現しているため，移動の安全性の向上が望める．しかしながら，地図の保持に必要な記憶領域量が過大になるばかりでなく，経路計画に必要な計算時間が増加する．また，地図を生成する際に行う観測精度も高くする必要があり，地図生成に必要な計算量および時間も増加する．一方，ノードの配置密度を低くし，地図の精度を

落とすことで、地図の保持に必要な記憶領域量を低減でき、経路計画に必要な計算時間も減少できる。また、観測の精度を低くすることが可能であり、地図生成に必要な計算量および時間も抑えられる。しかしながら、粗雑な地図を用いた計画では移動の安全性が低下する。以上の考察により、移動の安全性を考慮しつつ必要最低限の分解能で地図を生成することが望ましい。

### 1.2.3 ロードマップの分解能と経路計画

グラフ表現を基本とした地図表現に関する研究は多くなされている。環境をグラフで抽象化する際、ノードの配置密度で抽象度の度合を調節することができる [46], [47]。ノードの配置密度が低い場合、一つのノードで広い範囲を表現するため、地図の分解能は低く、抽象度の高い地図であると言える。それに対して、ノードの配置密度が高い場合、一定の範囲を多くのノードで表現するため、地図の分解能が高く、抽象度の低い地図であると言える。

グラフ表現の地図に限った場合、最も分解能が低い地図はトポロジカル地図 (Topological map) である [48] ~ [51]。トポロジカル地図では環境内の固有な場所 (例えば、部屋等) をノードにし、固有な場所間につながりがある場合にノード間をリンクで繋げる。この地図を利用することで、目標場所に行くために通るべき場所の大まかな系列を計画することができる。しかしながら、リンクが保持する情報はつながりの情報のみであり、ノード間の位相関係も距離関係もないため、実際にどのような経路を移動するべきかを計画することは不可能である。

それに対して、分解能の高い地図の例として空間を細かい格子状に分割したグリッド地図 (Grid-based map) が挙げられる [52] ~ [55]。グリッド地図はメトリック地図の情報を基に各グリッドが自由空間であるか否かを判断している。自由空間である領域をノードとし、隣り合う自由空間の領域をリンクで繋げている。この地図では空間を細かく分割することで、環境を詳細に表現することができ、詳細な経路を計画することができる。しかしながら、グラフのサイズが非常に大きくなるため、経路計画の計算量が非常に大きくなる。

これら地図の問題点に対して、トポロジカル地図とグリッド地図を組み合わせたハイブリット地図 (Hybrid map) が提案されている [56] ~ [60]。これらの地図では場所のつながりを表すトポロジカル地図の各ノードにグリッド地図が付随している。この組み合わせによって、トポロジカル地図で通過する場所の系列を計画した後に、グリッド地図を用いて詳細な経路を計画することができる。この時、トポロジカル地図のノードをサブゴールと設定することで、全体の経路計画問題を複数の経路計画問題に分割することができる。また、グリッド地図の分解能を可変にして経路計画する手法も提案されている [61] ~ [66]。

これらの手法では、隣接する複数の自由空間を表す領域を合わせて一つの自由空間領域とすることで、自由空間を表すノードの数を減らしている。これによって、地図の保持に必要なとする記憶領域量だけではなく、経路計画の計算量も削減することができる。しかしながら、これらの手法では移動ロボットの走行安全性を考慮していない。一方、得られた経路に対して、障害物との距離を考慮して経由点の個数や間隔を調節し、経路を修正する手法に Elastic Band 法（もしくは Bubble Band 法）と呼ばれる手法がある [67]。しかしながら、この手法は事前に得られた経路に対して適用するものである。このように、これらの手法は事前に得られた地図もしくは経路に対して適用するものであり、自律的な地図生成に適用された例はあるか、これらの情報を直接組み込んだ地図すら未だ提案されていない。

以上の背景より、逐次的に観測から得られた走行安全性に関する情報を基にして、環境表現を変更したロードマップ法の開発が必要である。

#### 1.2.4 運動特性を考慮したロードマップの必要性

ロボットの経路計画において、ロボットの運動特性を考慮する必要がある。従来のロードマップを用いた経路計画においては、移動ロボットが制約（例：速度制約、衝突回避など）を違反せずに経路を追従できることは保証されていない。なぜならば、それらの手法では移動ロボットの運動特性を考慮せず、作業空間の幾何学的な連結性のみに基づいてロードマップを定義しているためである。この問題に対して、経路計画後に直線分のリンクを弧に変更する手法 [68]、経路計画後にノードの位置の変更もしくは新たなノードの追加を行う手法 [69] やオンラインで軌道を計画し直す手法 [70] などが提案されている。しかしながら、これらの手法では経路計画と軌道計画を分離しているため、経路計画後に追加的な処理が必要となる。

一方で、移動ロボットの運動特性を考慮した Kinodynamic motion planning と呼ばれる軌道計画問題がある [71] ~ [73]。この問題では、コンフィギュレーションのみならず速度等をも含む状態空間上で軌道計画を行う。一般的にコンフィギュレーションと速度を考慮した状態空間内の二点を結ぶ直線分はロボットの実現可能な軌道を表現しない [74]。Kinodynamic motion planning に対して、いくつかの手法が提案されている [75] ~ [78]。これらの手法は複雑な動特性を持つ多くのシステムの行動計画に有効である。しかしながら、文献 [75] ~ [77] はオンライン軌道計画手法であり、文献 [78] は定められた現在位置と目標位置に対する軌道計画手法である。このため、いずれも異なる出発位置・目標位置に対して再利用可能な地図を生成する手法ではなく、同じ作業環境においても現在位置や目標位置が変わる度に再計算を要する。このように、移動ロボットの運動特性をも考慮した

軌道計画を行えるロードマップ法は未だ提案されていない。なお，移動ロボットの軌道計画を行うためには，移動ロボットが有する力学的拘束条件を考慮する必要がある。拘束条件はホロノミック拘束と非ホロノミック拘束に分類できるため，それぞれについて考慮する必要がある。このように，移動ロボットの運動特性を考慮したロードマップ法は未だ提案されていない。

### 1.2.5 ロードマップを用いた経路計画における課題

以上の議論をまとめると，移動ロボットのロードマップには以下の課題が残されている：

1. 移動ロボットが観測情報を基にして逐次的に分解能を変更しながら，地図生成と経路計画をオンラインで同時に行えない。
2. 移動ロボットの軌道計画において，以下に示す移動ロボットの力学的拘束をも考慮した計画を従来のロードマップ上で行えない。
  - ホロノミック拘束を有する移動ロボット
  - 非ホロノミック拘束を有する移動ロボット

これらの課題が解決されることで，未知な環境においても地図を生成しながら移動できるようになる。また，運動特性を組み込んだ地図を利用することでオフラインで動特性をも考慮した計画が行えるようになる。それによって，オンラインでの計算コストを削減できる。

### 1.2.6 物理情報量によるロードマップの拡張

前節で述べた課題が残されている一因として，従来のロードマップのノードが位置情報しか保持していないことが挙げられる。ノードが位置情報のみを保持しているため，計画段階で計画できることが位置の系列のみである。それに対して，文献 [68] ~ [70] などの解決策が提案されているが，ロードマップとしての性能は向上されていない。そこで，ロードマップのノードに位置情報以外の物理情報をも保持させることが解決策として考えられる。しかしながら，ノードに物理情報を付加する場合，付加すべき物理情報と付加する物理情報量の大きさを決定することが課題になる。特に後者が重要な課題である。一つのノードが保持する物理情報量を大きくする場合，物理情報が形成する空間を粗く分割すれば良い代わりに，情報の抽象度が高くなり，リンクの生成条件を満足することが困難になる [79]。一方，一つのノードが保持する物理情報量を小さくする場合，リンクの生成条件

を満足することが容易になる代わりに，物理情報が形成する空間を細かく分割する必要があり，空間全体を表現するために必要なノード数が膨大になる．以上の考察より，付加する物理情報量の大きさを適切な大きさにすることが望ましい．

### 1.3 本研究の目的

本研究は，前節で述べた課題を受け，可変分解能処理を用いて付加する物理情報量を決定することにより，移動ロボットの運動特性に基づいたロードマップ生成手法を提案する．具体的には，以下の三点を目的とする：

- (1) 未知環境下での移動ロボットの運動特性に基づいた地図の自律生成 未知環境下で移動ロボットがセンサ情報のみに基づいて適切な分解能を持つロードマップを自律的に生成する手法の提案と検証を行う．ここでは，全方向移動ロボットが未知の作業空間において，指示された目標地点へ向かいつつ，ロードマップを生成する問題について考える．ただし，議論を簡単にするために，ロボットは作業空間上の自身の位置を取得可能であるとする．
- (2) 全方向移動ロボットの運動特性に基づく可変分解能速度マップ生成 ホロノミック拘束を有する移動ロボットの動特性を考慮したロードマップ生成手法の提案と検証を行う．ここでは二次元平面上を全方向移動が可能なロボットを対象とし，対象とする移動ロボットの速度空間を適切な分解能で分割したロードマップ生成手法の提案と検証を行う．ロードマップを生成する空間として，移動ロボットの位置を表す二次元の位置空間に，移動ロボットの速度を表す二次元の速度空間を加えた四次元の状態空間を考える．ただし，位置を表す二次元のロードマップは既存の手法で与えられるものとし，その二次元のロードマップに領域で表される速度空間の情報を付加する．
- (3) 車両型移動ロボットの運動特性に基づく可変分解能姿勢マップ生成 非ホロノミック拘束を有する移動ロボットの動特性を考慮したロードマップ生成手法の提案と検証を行う．ここでは二次元平面上を移動する車両型移動ロボットを対象とし，対象とする移動ロボットの姿勢空間を適切な分解能で分割したロードマップ生成手法の提案と検証を行う．ロードマップを生成する空間として，移動ロボットの位置を表す二次元の位置空間に，移動ロボットの姿勢を表す一次元の姿勢空間を加えた三次元の配位空間を考える．ただし，位置を表す二次元のロードマップは既存の手法で与えられるものとし，その二次元のロードマップに領域で表される姿勢空間の情報を付加する．

### 1.4 本論文の構成

本論文は全 6 章から構成される。第 3 章と第 4 章では全方向移動ロボットを対象とし、第 5 章では車両型移動ロボットを対象とする。

第 2 章では、グラフについての基本知識を説明する。

第 3 章では、未知環境下で移動ロボットがセンサ情報のみに基づいて適切な分解能を持つロードマップを自律的に生成する手法について示す。なお、対象とする全方向移動ロボットは全方位観測が可能であるとする。

第 4 章では全方向移動ロボットの運動学を考慮した新しいロードマップの生成手法について示す。提案手法は第 3 章で示した手法および従来のロードマップ生成手法で生成されたロードマップが保持する幾何学的な安全性を受け継ぎ、さらに全方向移動ロボットの性能をも考慮したロードマップを生成する。

第 5 章では車両型移動ロボットの運動学を考慮した新しいロードマップの生成について示す。第 4 章ではホロノミック拘束を有する移動ロボットを対象としていたが、本章では非ホロノミック拘束を有する移動ロボットを対象とした手法を示す。

最後に第 6 章では、各章で得られた知見を総括し、本研究の結論を述べる。



## 第 2 章

# グラフとグラフ探索

本章では，グラフ表現とそれを用いた探索手法について紹介する．

### 2.1 グラフ構造

グラフは有限個のノード（節点，頂点） $v$  から成る集合  $\mathcal{N} = \{v_1, \dots, v_n\}$  と，有限本のリンク（辺，枝） $l$  から成る集合  $\mathcal{L} \subseteq [\mathcal{N}]^2$  で構成される集合  $\mathcal{G} = (\mathcal{N}, \mathcal{L})$  である．もしリンク  $l \in \mathcal{L}$  がノード  $v, v' \in \mathcal{N}$  を結ぶリンクならば  $l = (v, v')$  と書く．この時，ノード  $v$  と  $v'$  は隣接 (adjacent) の関係にあり，リンク  $l$  はノード  $v$  と  $v'$  に接合している (incident) という．リンクの両端が同じノードである場合，そのリンク  $l = (v, v)$  をループ (loop) という．また，2 個のノード間に複数のリンクが存在するとき，これらの辺を多重辺 (multiple edges) といい，ループや多重辺を含まないグラフを単純グラフ (simple graph) という．なお，断りがない限り，本論文ではグラフは単純グラフを意味する．

グラフ  $\mathcal{G}$  上のノード  $v_i \in \mathcal{N}$  に隣接するノードの集合  $\mathcal{N}(v_i) = \{v_j \in \mathcal{N} \mid (v_i, v_j) \in \mathcal{L}\}$  を近隣 (neighborhood) という．グラフ  $\mathcal{G}$  上の隣接するノード同士をたどった系列を経路 (path) と呼び， $(v_i)_{0:n}$  をノード系列  $(v_0, v_1, \dots, v_n)$  と定義すると，経路の集合は  $\Pi[\mathcal{G}] = \{(v_i)_{0:n} \mid n \in \mathbb{N}, v_i \in \mathcal{N}, \forall i \in [0, n], (v_j, v_{j+1}) \in \mathcal{L}, \forall j \in [0, n-1]\}$  と書ける．そして，あるノード  $v_S$  からあるノード  $v_D$  までの経路は  $\Pi[\mathcal{G}](v_S, v_D) = \{(v_i)_{0:n} \in \Pi[\mathcal{G}] \mid v_0 = v_S, v_n = v_D\}$  である．この時，経路内の任意の 2 つのノードが異なるノードである経路を単純経路 (simple path)

$$\Pi[\mathcal{G}](v_S, v_D) = \{(v_i)_{0:n} \in \Pi[\mathcal{G}] \mid v_0 = v_S, v_n = v_D, v_i \neq v_j, i \neq j, \forall i, j \in [0, n]\}$$

という．また，経路の始点  $v_S$  と終点  $v_D$  が同じである時にその経路を閉路 (cycle) という．閉路を持たないグラフを木 (tree) という．グラフ  $\mathcal{G}$  の任意の 2 つのノード頂点  $v_i, v_j$

に対して、 $v_i$  と  $v_j$  を結ぶ経路が存在する時に  $\mathcal{G}$  は連結 (connected) であるという。一方、連結でないグラフを非連結 (disconnected) であるという。なお、断りがない限り、本論文では経路は単純経路を意味する。

グラフにはリンクに方向を持つ有向グラフ (directed graph) と方向を持たない無向グラフ (undirected graph) の2種類がある。有向グラフにおいて、有向グラフの全てのリンクの向きを考慮しない状況で連結であれば、そのグラフを連結有向グラフという。また、リンクの向きを考慮しても連結であれば強連結 (strongly connected) であるといい、そうでないものを弱連結 (weakly connected) であるという。

本論文では、第3章においては無向グラフを用い、第4章および第5章においては有向グラフを用いる。

## 2.2 グラフ探索

グラフ探索とはグラフ上のあるノードから別のノードへの経路を探索する手法である。この節では問題の定式化および探索手法について説明する。

### 2.2.1 問題の定式化

グラフ  $\mathcal{G} = (\mathcal{N}, \mathcal{L})$  が与えられているとする。初期ノードと目標ノードをそれぞれ  $v_S$  と  $v_D$  とするときの経路  $\Pi[\mathcal{G}] = (v_S, \dots, v_D)$  を求める。与えられたグラフのそれぞれのリンク  $l_i \in \mathcal{L}$  に対して、非負のコスト  $c_i$  を与える。ここで、 $C = \sum_{i=0}^m c_i$  を評価関数と定義すると、任意の経路に対する評価値  $\sum_{i=S}^D c_i$  を計算できる。この評価値を最小にする経路が  $v_S$  から  $v_D$  への最適な経路である。

### 2.2.2 グラフ探索手法

グラフ探索手法は数多くあり、ここでは主な手法について述べる。

1. 深さ優先探索 (Depth-first search)

初期ノードから行き止まりになるまで探索を進めて、行き止まりになったら後戻りして、最も近くの未探索のノードまで戻って探索を行う手法。

2. 幅優先探索 (Breadth-first search)

初期ノードから隣接する全てのノードを探索する。そして、隣接するノードに隣接する全てのノードを探索する。この試行を繰り返して探索を行う手法。

## 3. 最良優先探索 (Best-first search)

幅優先探索に対して、次に探索する最も望ましいノードを何らかの規則を用いて決める手法。代表例としてダイクストラ法 (Dijkstra's algorithm) や A\* 探索 (A\* search) などが挙げられる [80], [81]。

## ダイクストラ法 (Dijkstra's algorithm)

ダイクストラ法はグラフ上の初期ノードからその他の全てのノードへの最短経路を求める手法である。なお、探索コストとして距離を用いる。その結果として、目標ノードまでの最短経路問題を解くことができる。アルゴリズムの概要を次に示す。

ここで、 $\mathcal{N}$  をノード集合とし、 $\mathcal{L}$  をリンク集合として、 $\mathcal{L}$  内の全ての  $(v, v')$  に対するコストを  $g(v, v')$  とする。また、 $f(v)$  を初期ノード  $v_S$  からノード  $v (\in \mathcal{N})$  までの距離とし、 $\mathcal{D}$  を最短経路が見つかったノード集合とする。目標ノードを  $v_D (\in \mathcal{N})$  とし、計算対象ノードを  $v_c$  とした場合の概要を以下に示す：

**Algorithm: Dijkstra's Algorithm**

**STEP1 (初期化):**

$$\mathcal{D} := \emptyset, f(v_S) := 0, f(v) := \infty (\forall v \in \mathcal{N} \setminus \{v_S\}).$$

**STEP2 (ノード選択):**

$$v_c := \arg \min_{v \in \mathcal{N} \setminus \mathcal{D}} f(v), \mathcal{D} := \mathcal{D} \cup \{v_c\}.$$

**STEP3 (距離更新):**

全ての  $v (\in \mathcal{N} \setminus \mathcal{D}, (v_c, v) \in \mathcal{L})$  に対して、 $f(v_c) + g(v_c, v) < f(v)$  ならば、 $f(v) := f(v_c) + g(v_c, v)$ 。

**STEP4 (繰り返し):**

もし、 $v_D \in \mathcal{D}$  ならば終了。そうでなければ STEP 2 へ戻る。

STEP 4 での終了条件を  $\mathcal{N} = \mathcal{D}$  とすることで、初期ノードから全てのノードまでの最短経路を求めることができる。

## A\* 探索 (A\* search)

A\* 探索はグラフ上の初期ノード  $v_S$  から目標ノード  $v_D$  への最短経路を求める手法である。なお、ここでも距離をコストとして用いている。基本的なアルゴリズムはダイクストラ法と同じであるが、A\* 探索は各ノード  $v$  から目標ノードまでの推定距離  $\hat{h}(v)$  を用いることで、初期ノードから目標ノードまでの最短経路問題を効率的に解くことができる。こ

のとき,  $\hat{h}(v)$  をヒューリスティック関数と呼び, 次の条件を満たす必要がある:

$$0 \leq \hat{h}(v) \leq h(v), \forall v \in \mathcal{N}. \quad (2.1)$$

なお,  $h(v)$  は  $v_S$  から  $v$  までの実際の経路距離である. アルゴリズムの概要を次に示す.

ここで,  $\mathcal{N}$  をノード集合とし,  $\mathcal{O}$  を計算中のノードの集合とし,  $\mathcal{C}$  を計算済みのノードの集合とする.  $\mathcal{L}$  をリンク集合として,  $\mathcal{L}$  内の全ての  $(v, v')$  に対するコストを  $d(v, v')$  とする. また,  $\hat{f}(v)$  を初期ノード  $v_S$  からノード  $v (v \in \mathcal{N})$  を経由して, 目標ノード  $v_G$  までの推定距離とし,  $\hat{g}(v)$  を  $v_S$  から  $v$  までの推定距離とし,  $\hat{h}(v)$  を  $v$  から  $v_G$  までの推定距離とする. 計算対象ノードを  $v_c$  とした場合の概要を以下に示す:

**Algorithm: A\* Search Algorithm**

**STEP1 (初期化):**

$\mathcal{O} := \{v_S\}, \mathcal{C} := \emptyset, \hat{g}(v_S) := 0, \hat{f}(v_S) := \hat{g}(v_S) + \hat{h}(v_S).$

**STEP2 (ノード選択):**

$v_c := \arg \min_{v \in \mathcal{O}} \hat{f}(v), \mathcal{O} := \mathcal{O} \setminus \{v_c\}$ . もし  $v_c = v_G$  ならば探索終了. そうでなければ  $\mathcal{C} := \mathcal{C} \cup \{v_c\}$ .

**STEP3 (距離更新):** 全ての  $v (v \in \mathcal{N}, (v_c, v) \in \mathcal{L})$  に対して,  $f'(v) = \hat{g}(v_c) + d(v_c, v) + \hat{h}(v)$  を計算する.  $v$  の状態に合わせて場合分けする:

1. if  $v \in \mathcal{O}$   
 もし  $f'(v) < \hat{f}(v)$  なら,  $\hat{f}(v) := f'(v)$  とする.
2. elseif  $v \in \mathcal{C}$   
 もし  $f'(v) < \hat{f}(v)$  なら,  $\hat{f}(v) := f'(v)$  として,  $\mathcal{C} := \mathcal{C} \cup \{v_c\}, \mathcal{O} := \mathcal{O} \setminus \{v_c\}$  とする.
3. else  
 $\hat{f}(v) = f'(v)$  として,  $\mathcal{O} := \mathcal{O} \setminus \{v_c\}$  とする.

**STEP4 (繰り返し):** もし  $\mathcal{O} = \emptyset$  ならば, 目標ノードにたどり着ける経路が存在しないということで終了する. そうでなければ STEP 2 へ戻る.

一般に  $\hat{h}(v) = 0$  としたときにダイクストラ法と等価である.

### 2.2.3 実時間グラフ探索手法

実時間グラフ探索とは計画と動作を繰り返し行うことで, 目標へと向かっていく手法である. 代表的な手法には, Trailblazer search[82], RTA\* 探索 (Real-Time A\* search)[83]

や D\* 探索 (D\* search)[84] およびその派生探索手法 [85] ~ [87] がある．これらの手法は次の制約を想定しているため，前節までの初期ノードから目標ノードまでの最短経路をオフラインで求める手法とは異なり，少量のメモリで探索を実行できる手法である：

1. 探索できる探索深さが有限である，
2. 一定時間で移動しなければならない．

オフラインで探索を行う A\* 探索などの手法は，探索時間や探索深さの制限を考慮せずに，初期ノードから目標ノードへの最適経路を求めている．そのため，初期ノードと目標ノードを含むグラフの全体構造が既知でなくてはならない．一方，オンラインで探索を行う RTA\* 探索などの手法は，計画時間や探索深さの制限を考慮し，一度の探索で目標ノードまでの経路を求めない．ある程度の深さまでの経路を繰り返し求めることで，目標ノードまでの経路を求めている．この性質により，グラフの全体構造ではなく，現在ノードの周辺のみが既知であれば計画できる手法であり，グラフ構造が部分的に既知の問題やグラフ構造が変化する問題に適用されている．ここでは，アルゴリズムが比較的簡単であり，実装が容易な RTA\* 探索を紹介する．

RTA\* 探索 (Real Time A\* search)

RTA\* 探索は A\* 探索をオンラインで行うために変更された手法である．RTA\* 探索は一定の探索深さに対して，探索と移動を繰り返しながら目標ノードまでの経路を求めている．また，この手法では A\* 探索と同様に各ノードが目標ノードまでのヒューリスティックコスト（推定距離）を保持している．ただし，A\* 探索とは異なり，各ノードのヒューリスティックコストはエージェントがそのノードを訪問する度に更新される．この更新により探索深さが有限であっても目標ノードへの到達性が保証される．RTA\* 探索のアルゴリズムの概要を次に示す．

ここで， $\mathcal{N}$  をノード集合とし， $\mathcal{L}$  をリンク集合とする．また， $g(v, v')$  をノード  $v (\in \mathcal{N})$  から別のノード  $v' (\in \mathcal{N})$  までのグラフ上の最短距離とし， $\hat{h}(v)$  を  $v$  から  $v_D$  までの推定距離とし， $f(v, v')$  を  $v'$  を経由する  $v$  の評価値とする．さらに， $\mathcal{N}_k^{(v)}$  ( $k \geq 0$ ) を次式のように定義する：

$$\begin{aligned} \mathcal{N}_0^{(v)} &:= v, \\ \mathcal{N}_k^{(v)} &= \{v' \mid l(\hat{v}, v') \in \mathcal{L}, \hat{v} \in \mathcal{N}_{k-1}^{(v)}, v' \notin \bigcup_{i=0}^{k-1} \mathcal{N}_i^{(v)}\}. \end{aligned} \quad (2.2)$$

これは，ノード  $v$  から  $k$  ステップで初めて到達できるノードの集合を意味する．現在ノードを  $v_c$  とし，先読み深さを  $d$  とした場合の概要を以下に示す：

**Algorithm: Real-Time A\* search Algorithm**

STEP1 (初期化): 存在する全てのノードの  $\hat{h}(v)$  を削除し, スタート地点のノード  $v_S$  を  $v_c$  にセットする:

STEP2 (先読み探索): 全ての  $v \in \mathcal{N}_1^{(v_c)}$  について次式より評価値  $f(v_c, v)$  を計算し,  $f(v_c, v)$  を最小とする  $v^*$  を求める (Fig. 2.1(a)):

$$\begin{aligned} f(v_c, v) &= g(v_c, v) + \hat{h}(v), \\ v^* &= \arg \min_{v \in \mathcal{N}_1^{(v_c)}} f(v_c, v). \end{aligned} \quad (2.3)$$

なお,  $\hat{h}(v)$  が  $v$  に登録されていない場合は次式で計算する:

$$h(v) = \min_{v_{d+1} \in \mathcal{N}_{d+1}^{(v_c)}} \left\{ g(v, v_{d+1}) + \hat{h}(v_{d+1}) \right\}. \quad (2.4)$$

この時,  $\hat{h}(v_{d+1})$  が  $v_{d+1}$  に登録されていない場合は  $\hat{h}(v_{d+1}) = \|v_{d+1} - v_D\|^2$  とする. なお,  $v$  から先読み可能なステップ数が  $d$  未満の場合は,  $v$  から  $d$  ステップ未満で先読み可能な最大ステップ数を  $d$  として, (2.4) 式を計算する. もし  $d = 0$  ならば  $\hat{h}(v) = \|v - v_D\|^2$  とする. また,  $v$  から  $d$  ステップ以内に  $v_D$  が存在する場合は,  $\hat{h}(v) = g(v, v_D)$  とする.

STEP3 (コスト更新):  $\hat{h}(v_c)$  の値を次式に従い, 二番目に最小なコスト値に更新する (Fig. 2.1(b)):

$$\hat{h}(v_c) = \min_{v \in \mathcal{N}_1^{(v_c)} \setminus \{v^*\}} f(v_c, v). \quad (2.5)$$

STEP4 (移動):  $v_c$  から  $v^*$  へ移動し,  $v^*$  を  $v_c$  にセットする.

STEP3 のコスト更新により, ノードを訪れるたびにコスト関数は単調増加し, エージェントが無限ループや行き止まりに陥ることはない.

ここで, Fig. 2.2 にてアルゴリズムの例を示す. なお, 図中の  $g(\cdot)$  は全て 1 とし, 先読み深さ  $d$  は 0 する.

- (1) エージェントは初期状態  $v_0$  にいるとする.
- (2)  $v_1, v_2, v_3$  へのそれぞれのコスト値を計算すると,  $f(v_0, v_1) = 2, f(v_0, v_2) = 3, f(v_0, v_3) = 4$  となるので,  $f(v_0, v_1)$  が最小値であり,  $f(v_0, v_2)$  が二番目に小さい値である.
- (3)  $f(v_0, v_2)$  が二番目に小さい値であるため,  $\hat{h}(v_0) = f(v_0, v_2) = 3$  と更新する.
- (4)  $f(v_0, v_1)$  が最小値でありため, エージェントは一番コスト値が小さい  $v_1$  に移動する.

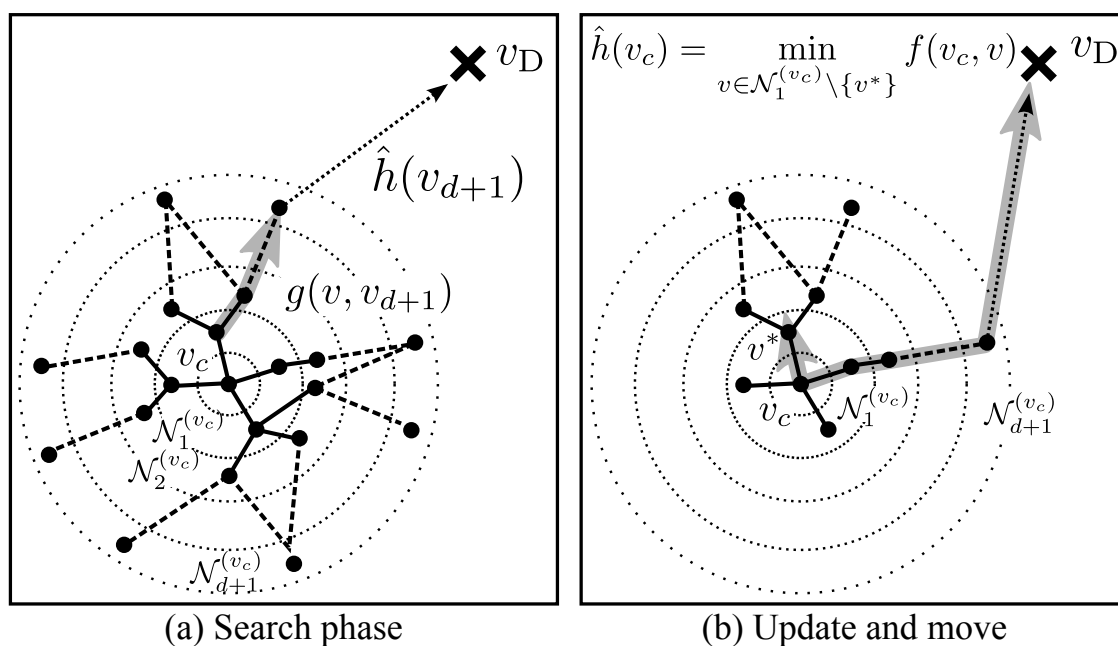


Fig. 2.1 Illustration of RTA\* : (a)  $g(v, v_{d+1})$  is the cost from  $v$  to  $v_{d+1}$  and  $\hat{h}(v_{d+1})$  is the heuristic cost from  $v_{d+1}$  to  $v_D$ , (b)  $v^*$  is calculated by (2.3) and  $\hat{h}(v_c)$  is updated by (2.5).

RTA\* 探索は以下の定理を有する .

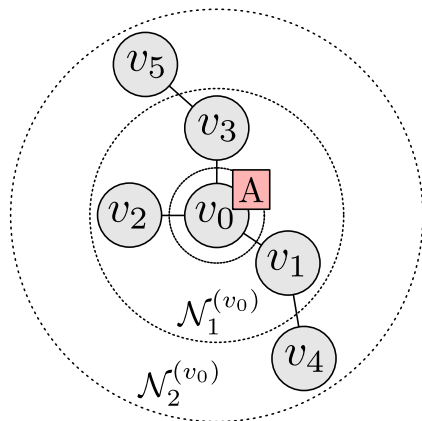
**定理 1 ( RTA\* 探索の完全性 )**

正のノード間コストと、有限のヒューリスティック値を持つ有限の探索空間において、全てのノードからゴールへ到達できるならば、RTA\* 探索は必ず解を持つ .

**定理 2 ( RTA\* 探索の正当性 )**

木における RTA\* 探索によるそれぞれの移動は、その時点における最先端ノードに基づいた、ゴールまでの推定コストが最小となる経路に沿う .

本論文では、第 3 章において RTA\* 探索を用い、第 4 章および第 5 章においてダイクストラ探索をそれぞれ用いる .



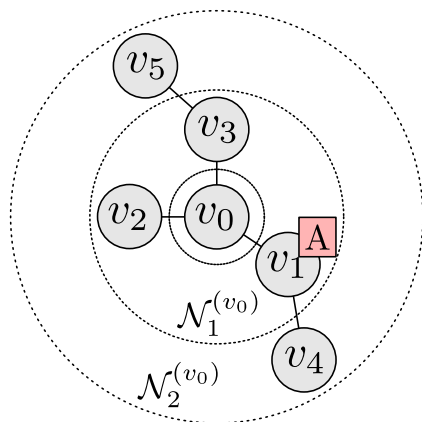
$$\begin{aligned} \hat{h}(v_0) &= 0 & \hat{h}(v_2) &= 2 & \hat{h}(v_4) &= 4 \\ \hat{h}(v_1) &= 1 & \hat{h}(v_3) &= 3 & \hat{h}(v_5) &= 5 \end{aligned}$$

$$\mathcal{N}_1^{(v_0)} = \{v_1, v_2, v_3\} \quad \mathcal{N}_2^{(v_0)} = \{v_4, v_5\}$$

Search:

$$\begin{aligned} f(v_0, v_1) &= g(v_0, v_1) + \hat{h}(v_1) = 1 + 1 = 2 \\ f(v_0, v_2) &= g(v_0, v_2) + \hat{h}(v_2) = 1 + 2 = 3 \\ f(v_0, v_3) &= g(v_0, v_3) + \hat{h}(v_3) = 1 + 3 = 4 \end{aligned}$$

(a) Search phase



Update:

$$\hat{h}(v_0) = 0 \Rightarrow \hat{h}(v_0) = f(v_0, v_2) = 3$$

Move:

$$A : v_0 \Rightarrow v_1$$

(b) Update and move

Fig. 2.2 An example of RTA\*



## 第 3 章

# 未知環境下での移動ロボットの運動特性に基づいた地図の自律生成

### 3.1 はじめに

#### 未知環境下での移動ロボットの運動特性に基づいた地図の自律生成

本章では、第 1 章で述べた未知環境下での移動ロボットの運動特性に基づいた地図の自律生成手法について述べる。1.2.3 節で述べたようにロードマップの分解能と経路計画には関係がある。しかしながら、その分解能を逐次的に観測により得られた情報を基に決定し、実時間で地図生成と経路計画を同時に行う手法は未だ提案されていない。そこで、移動ロボットがセンサ情報のみに基づいて適切な分解能を持つロードマップを自律的に生成する手法の提案と検証を行う。なお、未知環境下で自律的に地図を生成するため、この手法では地図生成と経路計画を並行して行う。具体的には、移動ロボットが障害物の置かれた平面状の作業空間において、指示されたゴール地点へ向かいつつ、同時に環境の地図を作成する問題について考える。自己位置推定問題を切り分けて議論を簡潔にするために、ロボットは作業空間上の自身の位置を取得可能であるとする。この問題に対して、本論文では全方位距離情報を用いた可変分解能処理によるグラフ地図生成法を提案する。提案するグラフ地図生成法は、障害物の近くでは高い環境表現分解能を用い、障害物が遠くにならない開かれた場所では低い環境表現分解能を用いてグラフ地図を生成する。これは障害物の近くでは安全性を考慮した正確な移動と高精度な計測が必要とされるという要求を反映している。さらに、提案手法はいくつかのルールセットから成り立っており、加えて経路計画に Real-Time A\*(RTA\*) 探索 [83] を用いるため、実時間での処理が十分に可能である。提案手法をシミュレーション環境とレーザレンジファインダ (Laser Range Finder: LRF) が取り付けられた全方向移動ロボットの実機に実装し、実験を通してその有用性を

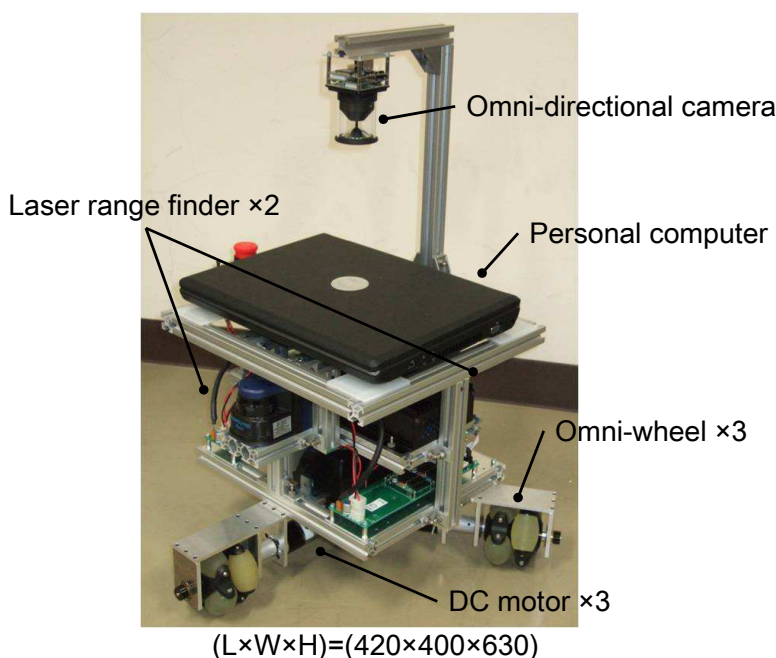


Fig. 3.1 Mobile robot for experiment

検証する。

以下に本論文の構成を示す。3.2 節にて問題設定及び提案手法の流れについて述べる。3.3 節にて空間の可変分解能処理に基づく空間のセル分割手法とルールベースのグラフ地図生成手法について述べ、3.4 節にて RTA\* 探索に基づく経路計画手法について述べる。3.5 節にてシミュレーションと実機実験の結果から提案手法の有用性を検証する。最後に3.6 節にてまとめと今後の課題について述べる。

## 3.2 問題設定および提案手法の概要

### 3.2.1 問題設定

本研究で使用したロボット (Fig. 3.1) と問題設定について説明する。ロボットは三個の DC モータとオムニホイールによる三輪駆動で、全方向移動が可能である。定格速度は約 500mm/s である。ロボットの前後には LRF, 中央には全方位カメラが装着されており、全方位観測が可能である。

作業空間の例を Fig. 3.2 に示す。空間は有界な閉空間であり、障害物と自由空間に分けることができる。ロボットは自由空間内を自由に移動できるが、障害物に触れることはできない。オペレータは作業空間に関する大まかな情報に基づいてロボットにゴール地点を

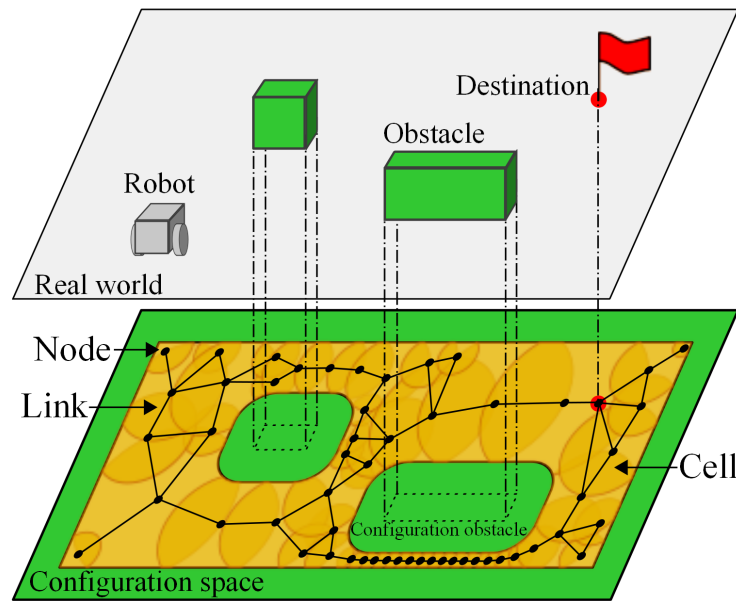


Fig. 3.2 Illustration of workspace and proposed graph-map

指示する．ただしロボットにとって詳細な障害物配置は未知であるため，ロボットは移動中に得られるセンサ情報をもとに近傍の地図を作成し，経路計画を行うことでゴールへ到達することが求められる．加えて，オペレータによって適切に指示された複数のゴール地点を巡回することで，最終的に空間全体の地図を獲得することが期待される．

### 3.2.2 提案手法の概要

3.2.1 節で述べた問題設定に対して提案手法では，グラフ探索手法を実空間での経路探索問題に適用するため，LRF の観測データを基にコンフィギュレーション空間（以降，C-space）上にグラフ構造の地図を生成する．C-space とはロボットの位置及び姿勢を表す空間である．なお，本研究ではロボットの重心位置をロボットの位置  $(x, y)$  と定義し，ロボットの正面の向きを姿勢  $\theta$  と定義する．C-space では，

- 実空間において障害物と干渉するロボットの位置及び姿勢から成る C-space 上の集合を C-space 上の障害物領域 (Configuration Obstacle) と呼ぶ．
- C-space 上の障害物の補集合を C-space 上の移動可能領域と呼ぶ．

本論文では 2 次元平面上を全方向に移動可能なロボットを対象とするため，C-space は実空間上の位置  $(x, y)$  からなる 2 次元空間である．また，円形状で近似可能なロボットを対

Table 3.1 Definitions of terms and symbols

提案手法における用語定義

ノード :	C-space 上の経路点 $(x, y)$
リンク :	ノードとノードを接続する線分
セル :	ノードを中心とした C-space 上の移動可能領域の部分集合
未到達ノード :	セルを保持していないノード
到達済ノード :	セルを保持しているノード
仮想リンク :	未到達ノードと $n_D$ を接続するリンク

記号定義

$q^{(n)}$ :	C-space 上のノード $n$ の位置	
$q_S$ :	スタート地点	$n_S$ : $q_S$ の位置を持つノード
$q_D$ :	ゴール地点	$n_D$ : $q_D$ の位置を持つノード

象とするため、実空間で矩形形状の障害物は C-space 上では角の丸い矩形形状の障害物となる (Fig. 3.2)。C-space 上の移動可能領域をノード及びリンクの集合で構成されるグラフで表すことでグラフ探索を適用できる。Table 3.1 に用語及び記号を示す。グラフ地図はノード、リンク及びセルによって構成される (Fig. 3.2) :

- ノードは C-space 上の経路点を表し、C-space 上の  $(x, y)$  座標を持つ。
- リンクはノード間の接続関係を表す線分である。
- ノード  $n$  のセルは  $q^{(n)}$  を中心とした所定の距離以内の領域である。ただし、障害物及びセンサの死角となる部分を含まない。

なお、セルは対応するノード上でのセンシング結果を基に生成されるため、未到達のノードはセルを持たない。これらのノードを未到達ノードと呼び、それ以外のノードを到達済ノードと呼ぶ。提案するグラフ地図は一般的なトポロジカルマップ [49] ~ [51] と異なり、リンクはノード間の幾何学的な接続のみならず、ロボットがノード間を直線移動可能であることを表す。この性質により、グラフ上で計画した経路から直接的にロボットの移動軌跡が得られる。その上、ノードを適切に配置することで移動時の安全性が考慮できる。提案手法では未完成な地図を用いてゴールへの経路を計画するために、各未到達ノードは  $n_D$  と「仮想的なリンク」で接続される。未到達ノードを初めて訪れる際に、当該ノードから  $n_D$  へ移動不能であることが判明すれば、仮想リンクは取り除かれる。

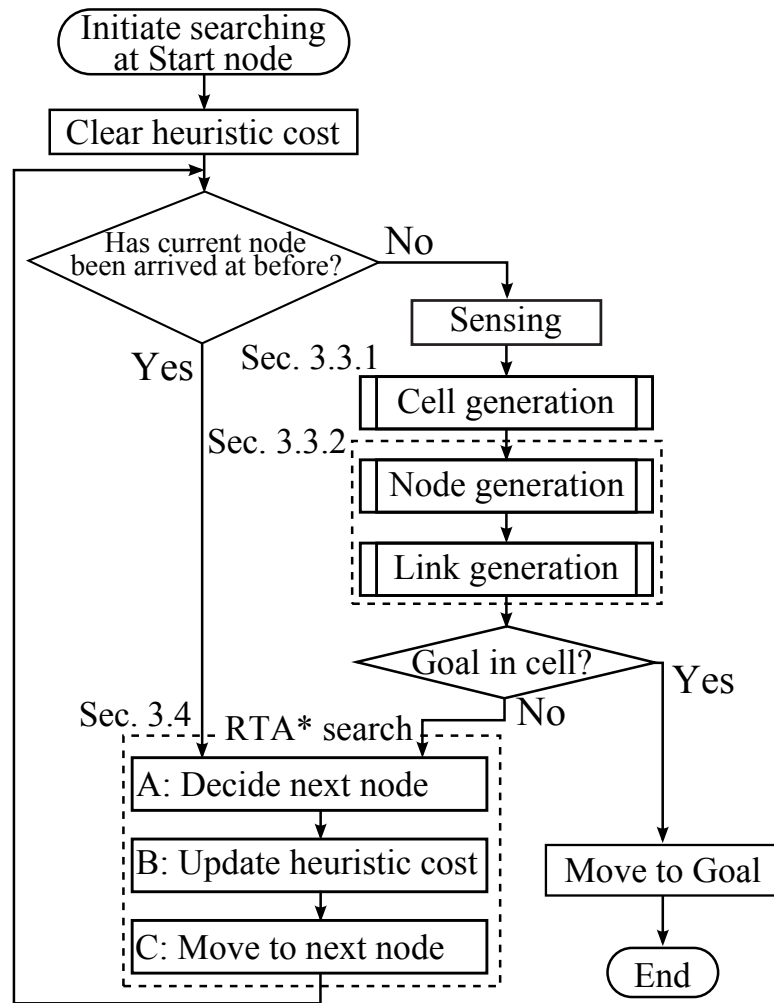


Fig. 3.3 Flowchart of overall procedure

提案手法の全体の流れを Fig. 3.3 に示す．なお，経路計画には 2.2.3 節で述べた RTA\* 探索 [83] を用いる．RTA\* 探索の具体的な適用方法については 3.4 節で説明する．スタート時，RTA\* 探索に用いるヒューリスティックコストを初期化し，未到達ノードである  $n_S$  に対して，3.3.1 節に示す手法により，センサ情報をもとにセル半径を決定し，セル生成を行う．次に 3.3.2 節に示す手法により，現在ノードの周囲に新たなノードを生成し，ノード間にリンクを生成する．なお，セル内に  $q_D$  が含まれていれば  $n_D$  を生成し， $n_D$  に移動して処理を終了する．セル内に  $q_D$  が含まれていなければ，3.4 節に示す処理により，グラフ探索を用いて次に向かうノードを決定し，ヒューリスティックコストを更新する．次のノードに到達後，以上の処理を繰り返す．この時，もし現在ノードと  $n_D$  の間に仮想的なリンクが存在していれば，探索を行う前にそのリンクを取り去る．なお，移動先のノードが到達済ノードであれば，セル，ノード及びリンクの生成を行わない．

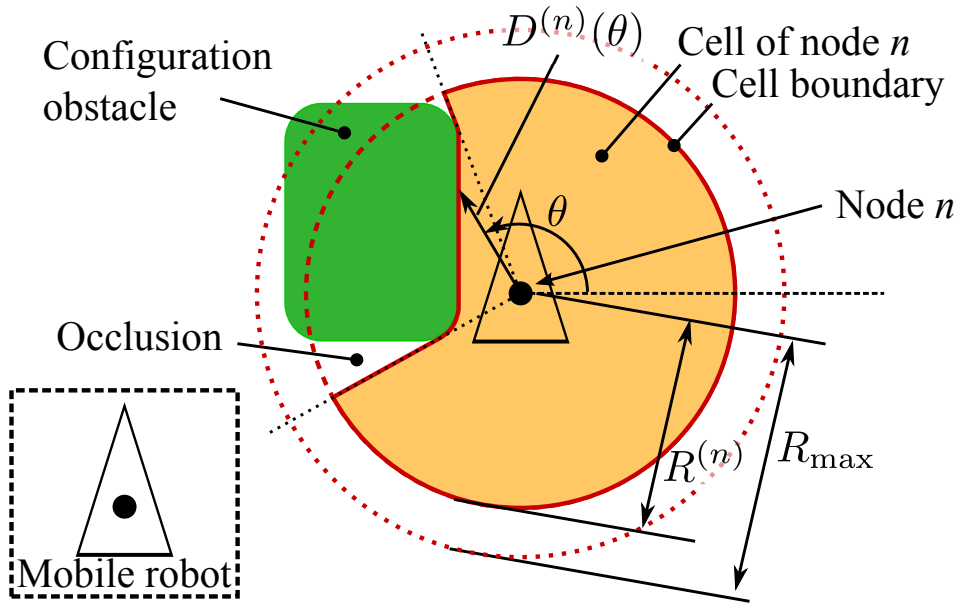


Fig. 3.4 Measured distance  $D^{(n)}(\theta)$  and cell radius  $R^{(n)}$

### 3.3 提案手法

本節では、全方位の距離計測能力を有するロボットに適した、ルールベースのグラフ生成手法を提案する。まず、3.3.1 節で可変分解能処理に基づくセル半径決定手法について説明する。この手法では状況に合わせたセル半径を反復的手続きから求める。次に 3.3.2 節でグラフのノード及びリンクの生成手法について説明する。この手法はいくつかの簡単なルールセットから成り立っている。これらの理由により、提案手法は計算コストが抑えられ、実時間処理に適している。最後に 3.3.3 節で提案手法に含まれるパラメータについて考察する。

#### 3.3.1 可変分解能処理に基づくセル半径の決定

セルの概要図を Fig. 3.4 に示す。Figure 3.4 のように、 $R_{max}$  をセンサ性能に基づく C-space 上での測定可能距離の最大値、 $R^{(n)}$  をノード  $n$  におけるセル半径、 $\theta \in \mathbb{R}^1 (0 \leq \theta < 2\pi)$  をロボットの右方向から反時計回りの対象点の角度、 $D^{(n)}(\theta)$  を C-space 内でノード  $n$  から  $\theta$  方向の障害物までの距離とする。ただし、測定可能距離内に障害物が検知されない場合は  $D^{(n)}(\theta) := R_{max}$  とする。なお、Fig. 3.4 の三角形はロボットの形状ではなく、ロボットの向きを指示するものである。これらを用いると、ノード  $n$  のセル  $C^{(n)}$

は次式で表される領域と定義される：

$$C^{(n)} = \left\{ q \mid \| q - q^{(n)} \| \leq \min \left( D^{(n)}(\arg(q - q^{(n)})), R^{(n)} \right) \right\}. \quad (3.1)$$

ただし， $\arg(\cdot)$  はベクトルの角度を表す．すなわち，セルとは位置  $q^{(n)}$  から直線移動可能かつ距離  $R^{(n)}$  以下の点の集合である．この時，セル半径  $R^{(n)}$  は次のように決定する：

$$R^{(n)} = \max(R_{\min}, \min(R_{\max}, \hat{R}^{(n)})), \quad (3.2)$$

$$\hat{R}^{(n)} = \max r \text{ s.t. } \frac{1}{2\pi} \int_0^{2\pi} \max(r - D^{(n)}(\theta), 0) d\theta \leq R_{\text{th}}. \quad (3.3)$$

(3.3) 式は半径  $\hat{R}^{(n)}$  の円と障害物との交差する深さの平均値が閾値  $R_{\text{th}}$  以下に収まることを表す．ここで， $\theta$  方向における障害物との交差する深さは (3.3) 式右辺の  $\max(r - D^{(n)}(\theta), 0)$  が表す．(3.3) 式を満たす最大の  $\hat{R}^{(n)}$  を基に (3.2) 式で  $R^{(n)}$  を決定する．この時， $R_{\min}$  はセルの大きさの下限を決定するパラメータである．また， $R_{\text{th}}$  は障害物との交差割合を決定するパラメータであり， $R_{\text{th}} = 0$  とすれば  $\hat{R}^{(n)}$  は最近接障害物までの距離となる．なお，良好な  $R_{\min}$  と  $R_{\text{th}}$  については 3.3.3 節で議論する．決められた  $R_{\text{th}}$  に対して (3.3) 式を満たす  $\hat{R}^{(n)}$  を直接的に求めるのは困難であるため，反復的手続きにより近似的に求める．そのために，(3.3) 式の左辺を次式のように近似して書きなおす：

$$\frac{1}{m} \sum_{k=0}^{m-1} \max(\hat{R}^{(n)} - D^{(n)}(k\Delta\theta), 0) \leq R_{\text{th}}. \quad (3.4)$$

ここで， $m$  はサンプリング点数である． $\Delta\theta$  は距離計測における角度  $\theta$  に関するサンプリング間隔である．また， $[*]$  は  $*$  を超えない最大の整数を表す．サンプリング点数  $m$  は  $m = \lceil 2\pi/\Delta\theta \rceil$  で与えられる．そして，(3.4) 式を用いて (3.2) 式を満たす  $R^{(n)}$  を次の STEP1~3 にしたがって決定する：

**Algorithm: Cell generation**

**STEP1 (初期化):**  $R_i^{(n)}$  に対して， $i := 0$  とし，次式により  $R_0^{(n)}$  に  $R_{\max}$  を代入して初期化する：

$$R_0^{(n)} := R_{\max}. \quad (3.5)$$

**STEP2 (終了条件):** 次の二式のいずれかが真であるならば，STEP4 へ．そうでないならば STEP3 へ：

$$\frac{1}{m} \sum_{k=0}^{m-1} \max(R_i^{(n)} - D^{(n)}(k\Delta\theta), 0) \leq R_{\text{th}}, \quad (3.6)$$

$$R_i^{(n)} \leq R_{\min}. \quad (3.7)$$

STEP3 (更新):  $i := i + 1$  とし, 次式により  $R_i^{(n)}$  を更新し, STEP2 へ:

$$R_i^{(n)} = \frac{1}{m} \sum_{k=0}^{m-1} \min(D^{(n)}(k\Delta\theta), R_{i-1}^{(n)}). \quad (3.8)$$

STEP4 (決定):

$$R^{(n)} = \max(R_i^{(n)}, R_{\min}). \quad (3.9)$$

この時,

$$\min(D^{(n)}(0), D^{(n)}(\Delta\theta), \dots, D^{(n)}((m-1)\Delta\theta)) \leq R_i^{(n)} \leq R_{i-1}^{(n)} \quad (3.10)$$

の関係が満たされるので,  $R_i^{(n)}$  は単調減少する. よって,  $R_i^{(n)}$  の収束性は保証される. 終了条件 (3.6) 式は, 半径  $R_i^{(n)}$  の円と障害物との交差する深さの平均値が閾値  $R_{\text{th}}$  以下に収まったときに更新を終了することを意味する. 終了条件 (3.7) 式は,  $R_i^{(n)}$  があらかじめ定めた最小値  $R_{\min}$  以下になったときに更新を終了することを意味する.

上記の手順により  $R^{(n)}$  が決定すると, ノード  $n$  を中心にセルが生成される. Fig. 3.4 に示すように, セルの境界は半径  $R^{(n)}$  の円弧, 障害物の境界および死角の端点を結ぶ線分の 3 つの要素から成る.

### 3.3.2 ノード及びリンクの生成手法

セルが生成されたノードに対し, 次のルール A-1 ~ 3 にしたがってノードを生成する (Fig. 3.5). ここで, ノード  $n$  のセルの  $\theta$  方向の境界の一点を  $b^{(n)}(\theta)$  と定義する:

$$b^{(n)}(\theta) = q \quad \text{s.t.} \quad \|q - q^{(n)}\| = \min(D^{(n)}(\theta), R^{(n)}), \theta = \arg(q - q^{(n)}). \quad (3.11)$$

また, ノードの集合を  $\mathcal{N}$  とし, ノードからセル境界までの距離を角度  $\theta$  の関数として  $f(\theta) = \min(D^{(n)}(\theta), R^{(n)})$  とする:

#### Rule: Node generation

A-1: ノードとゴールを結ぶ線分とセルの境界の交点が障害物と面していない時にその点にノードを生成する:

$$b^{(n)}(\theta_D) \quad \text{s.t.} \quad \|q_D - q^{(n)}\| = R^{(n)}, \theta_D = \arg(q_D - q^{(n)}). \quad (3.12)$$

A-2: セルの境界線の角点にノードを生成する:

$$\mathcal{K}^{(n)} = \{b^{(n)}(\theta') \mid f(\theta) \text{ is not differentiable at } \theta'\}. \quad (3.13)$$



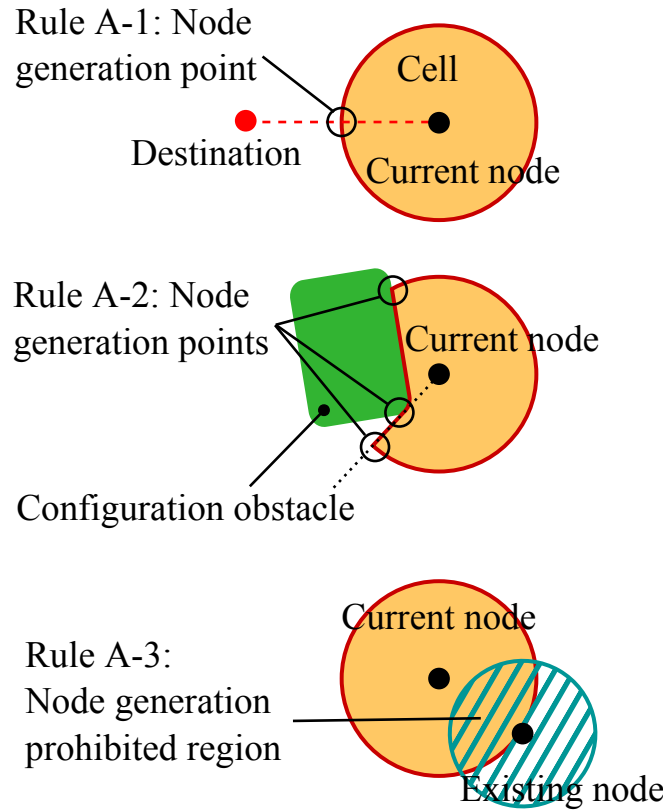


Fig. 3.5 Node generation rules

A-3: 他のセルと交差する部分にはノードを生成しない:

$$\mathcal{N}_{\text{new}}^{(n)} = \left\{ q \mid q \notin \bigcup_{n' \in \mathcal{N} \setminus \{n\}} C^{(n')}, q \in b^{(n)}(\theta_D) \cup \mathcal{K}^{(n)} \right\}. \quad (3.14)$$

A-3 は他のルールよりも優先して適用する禁止事項であり，ノードが無限に生成されることを防ぐ．これにより，グラフ構造が永久的に変化し続けることはない．新しく生成されたノードは未到達状態である．上記のルールでノードを生成した後に，次のルール B にしたがってリンクを生成する (Fig. 3.6) :

**Rule: Link generation**

B: 二つのノード  $n, n'$  を結ぶ線分上のすべての点が，二つのノードのセル  $C^n, C^{n'}$  のどちらかに含まれている時に，そのノード間にリンクを生成する:

$$tq^{(n)} + (1-t)q^{(n')} \in (C^{(n)} \cup C^{(n')}), \forall t \in [0, 1]. \quad (3.15)$$

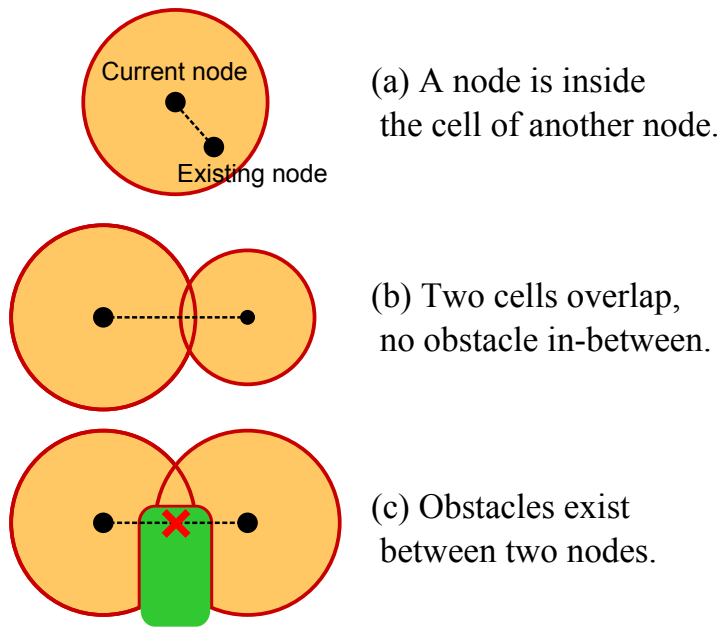


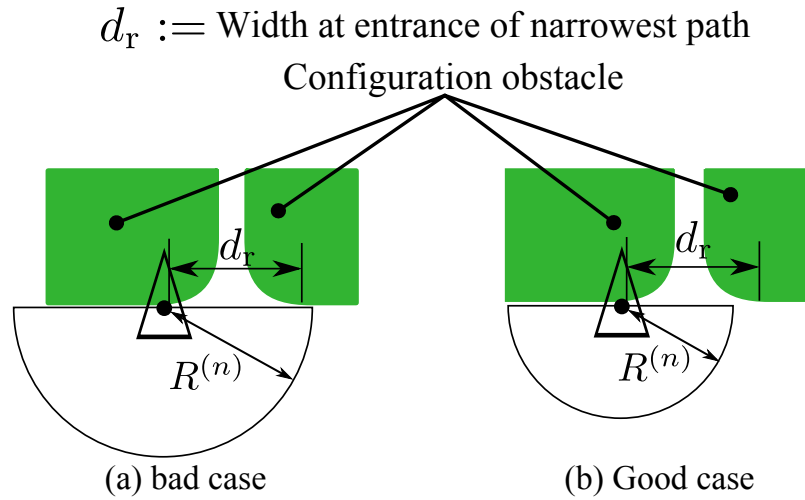
Fig. 3.6 Link generation rule

このルールにより，Figs. 3.6(a),(b) のような状況ではリンクが生成されるが，(c) のような状況ではリンクが生成されない．これにより，リンクで結ばれたノード間は障害物に衝突せずに直線移動できることが保証される．なお，C-space 上で Configuration obstacle と交差しない軌道は，実空間において（ロボットの大きさを考慮して）障害物と衝突しない軌道を表すことに注意されたい．以上のようにして，計四つのルールを用いて C-space 内にグラフ地図を逐次的に生成していく．

### 3.3.3 パラメータに関する考察

提案手法に含まれているパラメータの  $R_{\min}$  と  $R_{\text{th}}$  の決定指針について考察する．提案手法に対する要求事項の一つに，ロボットが走行可能な通路を見落とさず，通路の内部もしくは入り口にノードを生成することが挙げられる．特に環境に関する事前知識が無い場合は，ロボットの直径と等しい幅の通路まで考慮に入れる必要がある．ここでは Fig. 3.7 のように直線状の壁から直角に細い通路が伸びている状況に限定してパラメータに対する必要条件を求める．今， $d_r$  を空間内の最も細い通路の入り口の幅とする．Figure 3.7 のようにロボットが C-space 上の障害物境界上にいる状況を考えると，セルは半円になるため，(3.3) 式から次式が得られる：

$$\hat{R}^{(n)} = 2R_{\text{th}}. \quad (3.16)$$

Fig. 3.7 Decision indicator of  $R_{\min}$  and  $R_{th}$ 

そして，セル半径  $R^{(n)}$  は (3.2) 式より次式のように得られる：

$$R^{(n)} = \max(R_{\min}, 2R_{th}). \quad (3.17)$$

この時，通路の入り口にノードを生成するためには， $R^{(n)} < d_r$  となる必要がある．よって，以下の条件が得られる：

$$R_{\min} < d_r, \quad (3.18)$$

$$2R_{th} < d_r. \quad (3.19)$$

上記で求めた  $R_{\min}$  と  $R_{th}$  の条件の妥当性については，3.5 節で検証する．

## 3.4 経路探索

2.2 節で述べたように，経路計画においてその時点までに構築されたグラフ地図に未到達ノードからゴールへの仮想リンクを加えたものを用いてゴールへの最短経路を計画する．移動ロボットの経路計画には実時間性が求められるため，探索深さが限られる．しかし探索深さが不十分だとサイクルを持つグラフで無限ループに陥る可能性が生じる．この時，グラフのサイズが小さい場合には探索深さを気にせずに A\* 探索 [81] などのグラフ探索手法を用いても実時間性を保てるが，グラフのサイズが大きい場合には実時間性を保つために，有限の探索深さでゴールへの到達性を保証するオンライングラフ探索手法を用いる必要がある [83] ~ [86], [88]．そこで今回はアルゴリズムが単純であり，ノード数及び各ノードの保持しているリンク数が有限な空間であれば一定先読み深さの探索結果を基に解

を実行することで実時間性が保証されている RTA\* 探索 [83] を用いるとする。RTA\* 探索では各ノードのゴールまでの推定経路コストであるヒューリスティックコストを各ノードが保持し、ノードを訪問するたびにそのノードのヒューリスティックコストを更新するため、先読み深さが有限であってもゴールへの到達性が保証されている。

2.2.3 節で述べた RTA\* 探索を提案手法に組み込む方法について述べる。まず、グラフ地図内の全リンクを  $\mathcal{L}$ 、ノード  $n$  とノード  $n'$  間のリンクを  $l(n, n')$  として、 $\mathcal{N}_k^{(n)}$  ( $k \geq 0$ ) を次式のように定義する：

$$\begin{aligned} \mathcal{N}_0^{(n)} &:= n, \\ \mathcal{N}_k^{(n)} &= \{n' \mid l(\hat{n}, n') \in \mathcal{L}, \hat{n} \in \mathcal{N}_{k-1}^{(n)}, n' \notin \bigcup_{i=0}^{k-1} \mathcal{N}_i^{(n)}\}. \end{aligned} \quad (3.20)$$

これは、ノード  $n$  から  $k$  ステップで初めて到達できるノードの集合を意味する：

#### Algorithm: RTA\* search

- $n_c$  : 現在ノード
- $d$  : 先読み深さ
- $g(n, n')$  :  $n$  から  $n'$  までのグラフ上の最短経路距離
- $\hat{h}(n)$  :  $n$  のヒューリスティックコスト
- $f(n, n')$  :  $n'$  を経由する  $n$  の評価値

**STEP1** (初期化): 存在する全てのノードの  $\hat{h}(n)$  を削除し、スタート地点のノード  $n_S$  を  $n_c$  にセットする。

**STEP2** (先読み探索): 全ての  $n \in \mathcal{N}_1^{(n_c)}$  について次式より評価値  $f(n_c, n)$  を計算し、 $f(n_c, n)$  を最小とする  $n^*$  を求める (Fig. 3.8(a))：

$$\begin{aligned} f(n_c, n) &= g(n_c, n) + \hat{h}(n), \\ n^* &= \arg \min_{n \in \mathcal{N}_1^{(n_c)}} f(n_c, n). \end{aligned} \quad (3.21)$$

なお、 $\hat{h}(n)$  が  $n$  に登録されていない場合は次式で計算する：

$$\hat{h}(n) = \min_{n_{d+1} \in \mathcal{N}_{d+1}^{(n_c)}} \{g(n, n_{d+1}) + \hat{h}(n_{d+1})\}. \quad (3.22)$$

この時、 $\hat{h}(n_{d+1})$  が  $n_{d+1}$  に登録されていない場合は  $\hat{h}(n_{d+1}) = \|n_{d+1} - q_D\|$  とする。なお、 $n$  から先読み可能なステップ数が  $d$  未満の場合は、 $n$  から  $d$  ステップ未満で先読み可能な最大ステップ数を  $d$  として、(3.22) 式を計算する。もし  $d = 0$  ならば  $\hat{h}(n) = \|n - q_D\|$  とする。また、 $n$  から  $d$  ステップ以内に  $n_D$  が存在する場合は、

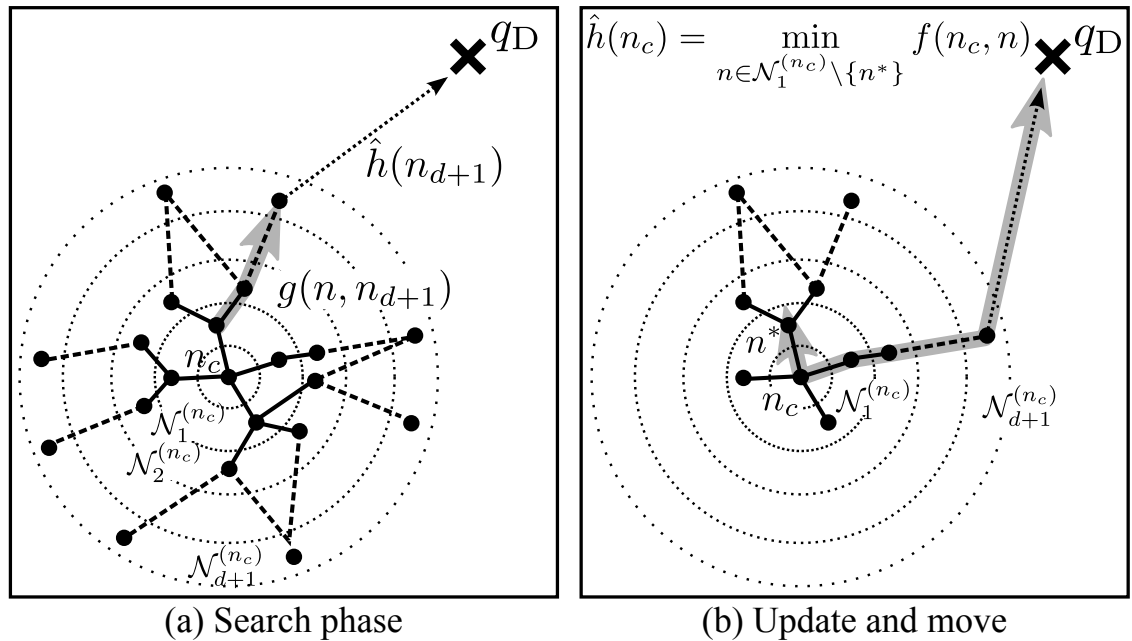


Fig. 3.8 Illustration of path planning using RTA\* : (a)  $g(n, n_{d+1})$  is the cost from  $n$  to  $n_{d+1}$  and  $\hat{h}(n_{d+1})$  is the heuristic cost from  $n_{d+1}$  to a goal position  $q_D$ , (b)  $n^*$  is calculated by (3.21) and  $\hat{h}(n_c)$  is updated by (3.23).

$\hat{h}(n) = g(n, n_D)$  とする .

STEP3 (コスト更新):  $\hat{h}(n_c)$  の値を次式に従い, 二番目に最小なコスト値に更新する (Fig. 3.8(b)) :

$$\hat{h}(n_c) = \min_{n \in \mathcal{N}_1^{(n_c)} \setminus \{n^*\}} f(n_c, n). \quad (3.23)$$

STEP4 (移動):  $n_c$  から  $n^*$  へ移動し,  $n^*$  を  $n_c$  にセットする .

STEP2~4 が Fig. 3.3 の RTA\* search 枠の A,B 及び C のそれぞれに該当する . STEP3 のコスト更新により, ノードを訪れるたびにコスト関数は単調増加し, エージェントが無限ループや行き止まりに陥ることはない .

移動するたびに新たにノードが生成されるが,  $R_{\min}$  のおかげで作業空間が有界ならば生成され得るノード数も有限であるため, RTA\* 探索を提案手法に組み込むことができる . また, 実時間性については先読み深さのみならずリンク数の有限性も必要となるが, この点については次節で検証する .

## 3.5 検証実験

提案手法をシミュレーションと実機により検証する．まず，シミュレーションを通じて，3.3.3 節で得られたグラフ地図生成アルゴリズムに含まれるパラメータの条件とゴールへの到達性との関係を定量的に考察する．次に，シミュレーションで求めた条件を満たす環境の元で行った実機 (Fig. 3.1) による実験について述べる．

### 3.5.1 数値シミュレーション

数値シミュレーションを通じて，3.3.3 節で得られたグラフ地図生成アルゴリズムに含まれるパラメータの条件とゴールへの到達性との関係を定量的に考察する．そこで Tables 3.2, 3.3 に示すそれぞれの設定を組み合わせた計 15 通りの設定でシミュレーションを行う．ただし，考察に用いた結果のみ載せる．また，RTA\* 探索の実時間性はノード数及び各ノードが保持しているリンク数が有限であれば保証されるため，ノード数及び各ノードが保持しているリンク数が有限であるということを確認する．なお，Table 3.2 の  $d_r$  は空間内の最も狭い通路の入り口の幅を表し， $R_{\min}$  と  $R_{th}$  はグラフ地図生成アルゴリズムに含まれるパラメータである．

一般に，RTA\* 探索は先読み深さ  $d$  を大きくすることで良い解が得られる傾向にあるが，グラフ地図を生成しながら探索を行う段階では， $\mathcal{N}_1^{(n_c)}$  に未到達ノードが存在していれば  $d$  の大きさによらずに未到達ノードが選択される傾向にあるため，ここでは  $d = 0$  とする．距離測定器の角度分解能を  $\Delta\theta \simeq 0.36^\circ$  とし，測定可能距離を 4000mm とする．

Table 3.2 は  $R_{\min}$  に関する 5 種類の設定のそれぞれにおいて，Fig. 3.9 に示す形状の環境にてスタート地点 ( $q_S$ )，ゴール地点 ( $q_D$ )， $R_{\min}$  及び  $R_{th}$  をそれぞれ 100 回変更して試行した結果である．Figure 3.9 では濃い灰色が障害物を，点がノードを，淡い灰色がセルで覆われた領域をそれぞれ表す．ただし， $0 < R_{th} < R_{\min}/2$  である．その結果として， $R_{\min}$  が次の条件を満たす時に全ての試行で  $p_D$  へ到達することが確認された：

$$0 < R_{\min} < d_r. \quad (3.24)$$

これにより (3.18) 式の妥当性が確認された． $q_D$  到達に要する移動回数についての比較も行った．比較は一組の  $q_S$ ， $q_D$  に対して，それぞれの設定で  $q_D$  到達に要する移動回数と Case 1-2 で要した移動回数との差の平均で行った．この時，(3.24) 式を満たしていても， $R_{\min}$  が過小だと必要な移動回数が多くなることが確認された．Case 1-1 の結果の一例を Fig. 3.9(a) に示す．移動回数は環境の大きさに依存するものであり，環境が拡大すればこの差はさらに広がることが予想される．また，Fig. 3.9(a) においてももしもセル半径を一

Table 3.2 Condition settings for  $R_{\min}$  with  $0 < R_{\text{th}} < d_r/2$ 

Case	Parameter setting	Success rate	Average of difference to Case 1-2
1-1	$0 < R_{\min} < d_r/2$	100%	+3.1 (steps)
1-2	$d_r/2 \leq R_{\min} < d_r$	100%	—
1-3	$R_{\min} = d_r$	87%	-0.9 (steps)
1-4	$d_r < R_{\min} < 3d_r/2$	76%	-2.1 (steps)
1-5	$3d_r/2 \leq R_{\min} \leq 2d_r$	70%	-4.3 (steps)

Table 3.3 Condition settings for  $R_{\text{th}}$  with  $d_r/2 \leq R_{\min} < d_r$ 

Case	Parameter setting	Success rate
2-1	$0 < R_{\text{th}} < d_r/2$	100%
2-2	$d_r/2 \leq R_{\text{th}} < d_r$	92%
2-3	$d_r \leq R_{\text{th}} < 3d_r/2$	71%

様に小さく設定した場合、障害物から遠い  $q_S$  と  $q_D$  付近でも一度当たりの移動距離が減り、全体の移動回数が多くなることが予想される。一方、 $R_{\min}$  が大きくなるにつれて一度当たりの移動量が増え、必要な移動回数が少なくなった。しかしながら、到達回数が減少することが確認された。Case 1-3 において  $R_{\min} = d_r$  としたことで、狭い通路の内側にノードが生成されずに  $q_D$  に到達できなかつた一例を (Fig. 3.9(b)) に示す。このことより、もしセル半径を一様に大きく設定した場合、目標地点へ到達できなくなることが予想される。

Table 3.3 は  $R_{\text{th}}$  に関する 3 種類の設定のそれぞれにおいて、Fig. 3.9 に示す形状の環境にてスタート地点 ( $q_S$ )、ゴール地点 ( $q_D$ )、 $R_{\min}$  及び  $R_{\text{th}}$  をそれぞれ 100 回変更して試行した結果である。ただし、 $\frac{d_r}{2} \leq R_{\min} < d_r$  である。その結果として、 $R_{\text{th}}$  が以下の条件を満たす時に全ての試行で  $q_D$  へ到達することが確認された：

$$0 < R_{\text{th}} < \frac{d_r}{2}. \quad (3.25)$$

これにより (3.19) 式の妥当性が確認された。Case 2-2, 3 では、障害物の周囲にいながらもセル半径  $R^{(n)}$  が  $R_{\min}$  にならずに、Fig. 3.9(a) と同様に狭い通路の内側にノードが生成されない試行があった。

上記で求めた条件を満たし、より実環境に近い環境 (20m × 20m) で行ったシミュレーション結果を Fig. 3.10 に示す。Figure 3.10 では二つの袋小路が存在するが、ゴールに

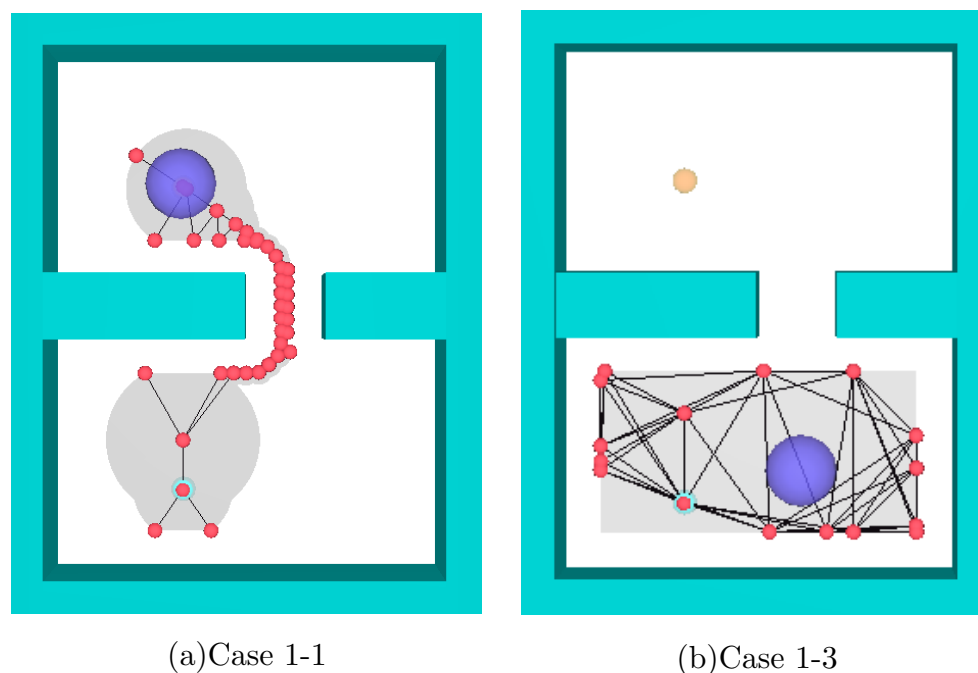


Fig. 3.9 Result of simulation

到達できている．可変分解能の効果により，自由空間が広い場所ではノード数が少なく，狭い場所ではノード数が多くなっている．また，シミュレーション実験の結果を通して，どのノードも保持しているリンク数が最大で 21 本であり，上限を持つことが確認された．これにより RTA\* 探索の実時間性が裏付けられた．

### 3.5.2 実機実験

シミュレーションで求めた条件を満たす環境の元で行った実機 (Fig. 3.1) による実験について述べる．有限空間内のスタート地点とゴール地点に対して本手法を二回試行し，グラフ地図を生成する．次に生成したグラフ地図を用いて，経路計画のみを一回試行する．ノード間の移動は，加速度  $500\text{mm/s}^2$ ，最高速度  $500\text{mm/s}$  の台形制御で行う．自己位置は事前に設置した位置が既知のランドマークを観測して計算する．RTA\* 探索の先読み深さは  $d = 0$  とする．ただし，一度グラフ地図が生成されたあとで，経路計画を行う場合は  $d \geq 1$  とする．LRF の角度分解能は  $\Delta\theta \simeq 0.36^\circ$  である．可変分解能セルの半径を決める時のパラメータの  $R_{\max}$  を LRF の測定可能距離の最大値  $4000\text{mm}$ ， $R_{\min}$  は環境と (3.24) 式を考慮して  $300\text{mm}$ ， $R_{\text{th}}$  は (3.25) 式を考慮して  $40\text{mm}$  とする．走行環境を Fig. 3.11 及び Fig. 3.12 に示す．問題設定で有限の広さを持つ閉じた環境を想定しているため，周囲には壁を配置する．ランダムに決めたいかなるスタート地点とゴール地点に対しても，経路が最低一本は存在するように障害物を設置する．なお，未知環境を想定して



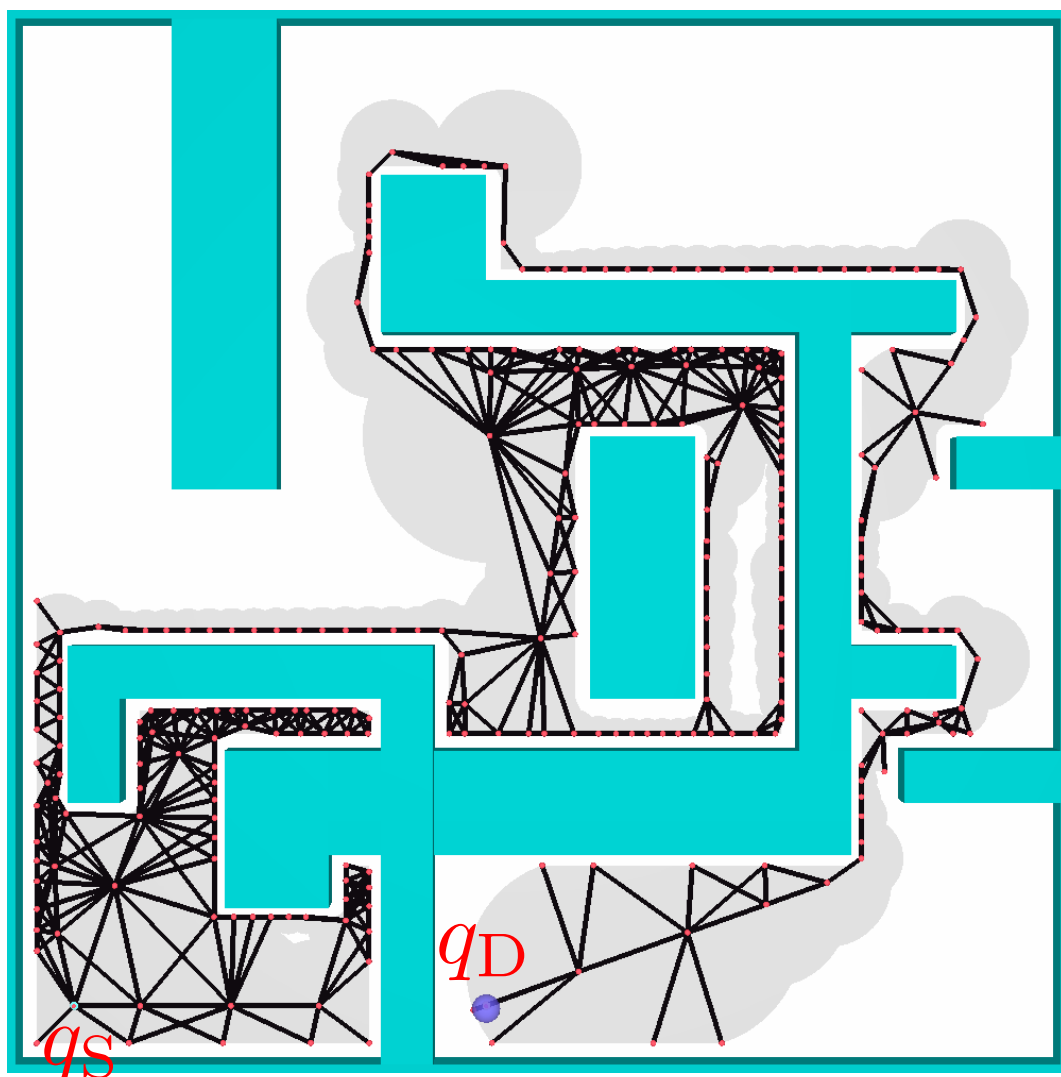


Fig. 3.10 Graph-map constructed by simulation in large environment

いるため、ロボットには障害物に関する情報を与えない。なお、提案手法は C-space 内にグラフを構築するものであるため、実際には LRF の観測データをもとに C-space 上の可視領域を計算する必要がある。

一連の実機実験の結果、得られた走行軌跡とグラフ地図を可視化したものを Fig. 3.13 に示す。Fig. 3.13(a) は一回目の走行後に生成された C-space 内のグラフ地図である。濃い灰色の点はグラフ地図のノードであり、濃い線はリンクである。薄く塗りつぶされた部分はセルである。また、矢印は移動ロボットの移動軌跡である。セルの半径を可変にしたことにより、スタート後や Obstacle3 を越えた後は、セル半径  $R^{(n)}$  が大きくなっていき、効率良くゴール地点へ向かうことができている。また、障害物が密集している領域では、セル半径  $R^{(n)}$  が小さくなり、安全に移動できている。Obstacle1,3 によって作られた袋

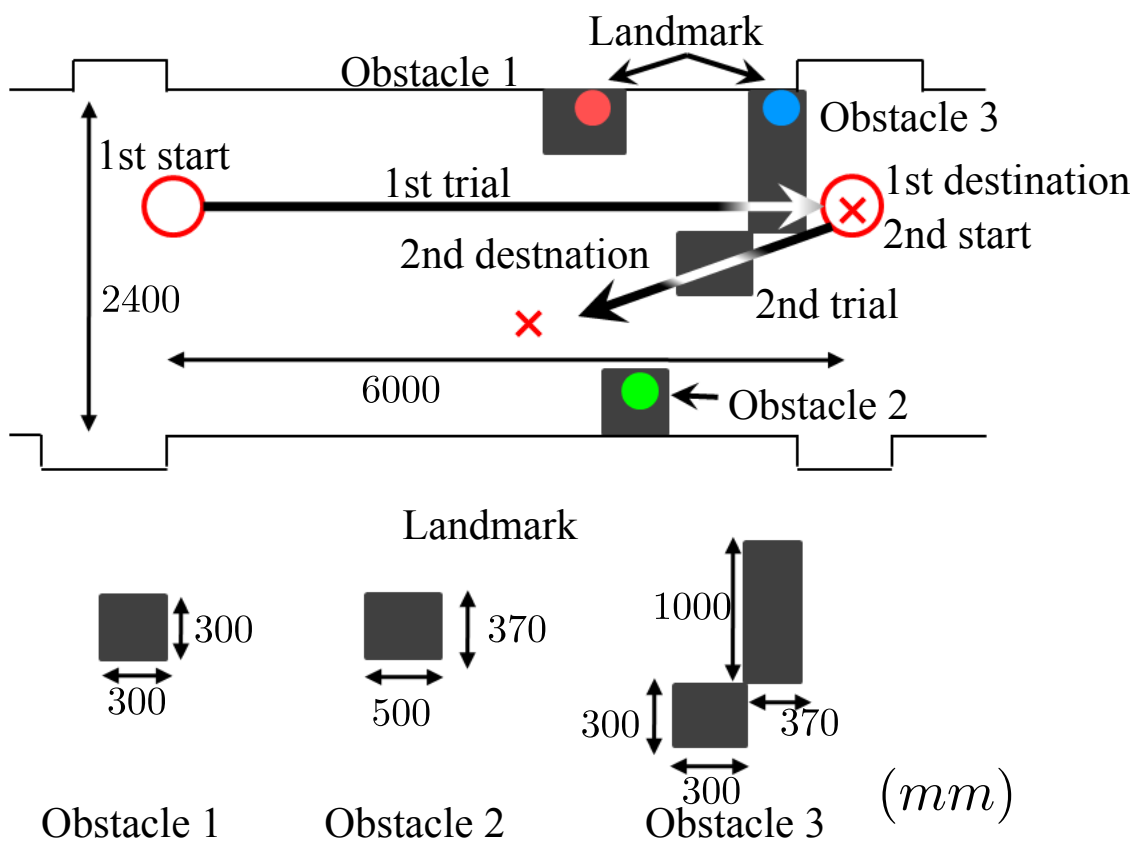


Fig. 3.11 Environment for experiment using an autonomous mobile robot



Fig. 3.12 View of environment for the experiment

小路内では、ルール A-2 と B により新しいノードを生成しつつ、既存のノードとの間にリンクを生成する。RTA\* 探索のメカニズムにより数ステップ後に袋小路から脱出する。脱出後は Obstacle 2, 3 によって作られた細道を通してゴール地点へ向かっている。このとき、セル半径  $R^{(n)}$  は最小値である  $R_{\min}$  となっている。

一回目の試行後に新たなスタート地点とゴール地点で二回目の試行を行うため、RTA\*

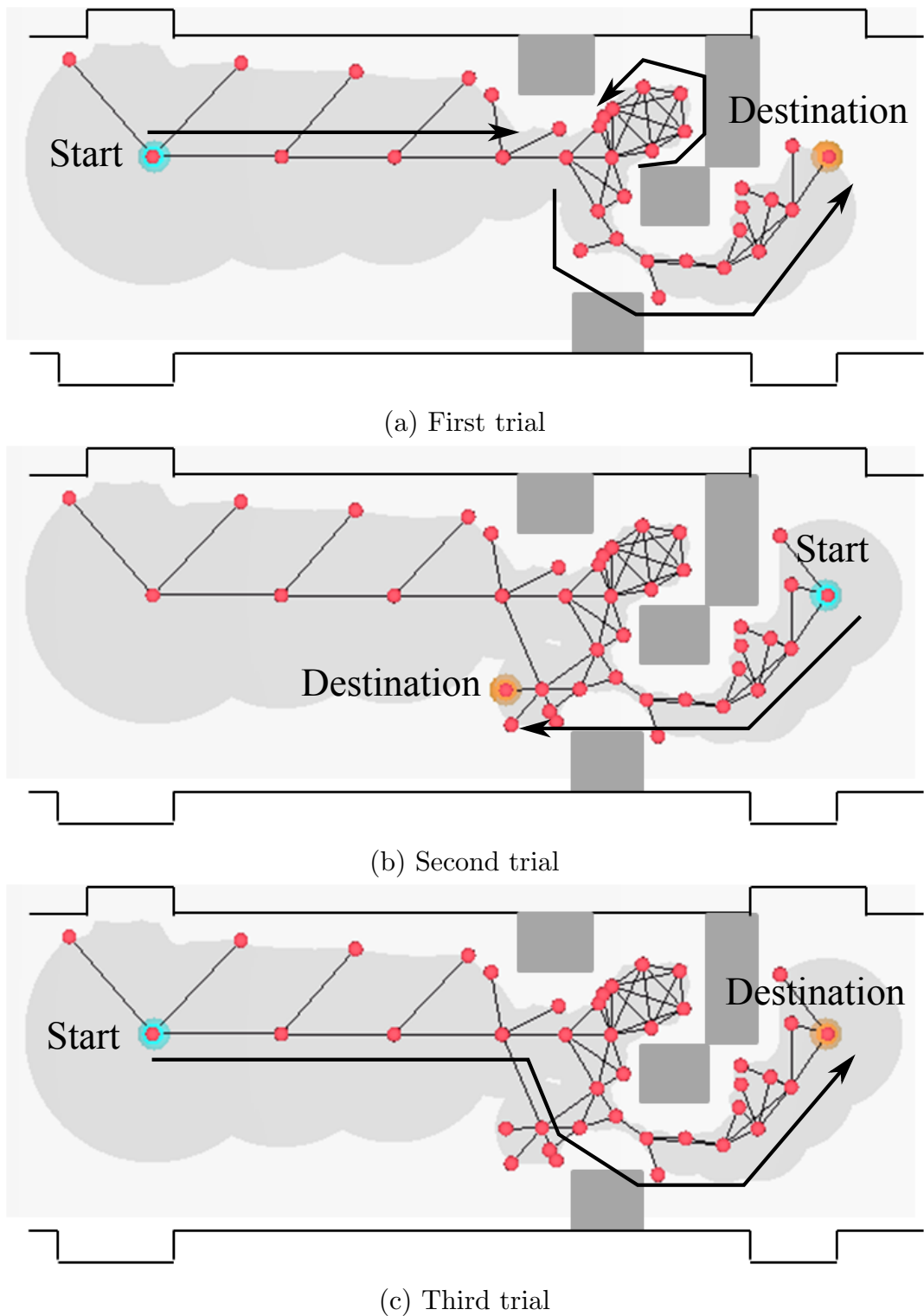


Fig. 3.13 Graph-map constructed by the mobile robot (a,b) and path of the mobile robot using path-planning on the graph-map with the search depth set as 5 (c)

探索のヒューリスティックコストを初期化する．二回目の試行では一回目の試行で生成したグラフ地図を利用しつつ，さらにグラフ地図を拡充している (Fig. 3.13(b)) ．

次に二回の試行で生成したグラフ地図 (Fig. 3.13(b)) を利用して，先読み深さを  $d = 5$  とした経路計画のみの試行を行う．三回目の試行でのスタート地点とゴール地点は，一回目の試行と同様に設定する (Fig. 3.13(c)) ．一回目の試行ではロボットは地図を保持していなかったために，袋小路に迷い込んでからゴール地点に向っているが，この試行では地図を保持し，さらに先読み深さを大きくすることで分岐路でゴール地点までの経路を先読みできているために，最適な経路を選択している．これにより，地図の逐次的な構築に加えて，獲得した地図を利用した効率的な経路計画が可能であることが確認された．

以上の実験結果から，試行を繰り返すことで，既存のグラフ地図を利用しつつ，グラフ地図を拡充する本手法の特性が確認された．

## 3.6 まとめ

本章では，未知環境において逐次的にグラフ地図を生成し，実時間で経路計画を行う手法を提案し，さらにその有用性をシミュレーションと実機実験にて検証した．提案手法は，単純な計算による可変分解能処理と，ルールベースのグラフ地図生成手法から成り立っているために，実時間処理に適している．可変分解能処理では，障害物との距離に応じてセル分割の細かさを変化させることで，移動時の安全性と効率に対する重視度の比重を連続的に切り替えることができる．しかしながら，提案手法で生成したロードマップは生成時のスタート位置及びゴール位置に依存している．また，生成されたロードマップでは幾何学的な安全性しか考慮されてなく，移動ロボットの動特性が考慮されていない．これらの理由により，スタート位置及びゴール位置を変更した場合や移動ロボットが各ノード上で静止せずに移動する場合に対しては，安全性が達成されていない．

## 第 4 章

# 全方向移動ロボットの運動特性に基づく可変分解能速度マップ生成

### 4.1 はじめに

本章では第 1 章で述べたホロノミック拘束を有する移動ロボットの動特性を考慮した新しいロードマップ手法について述べる．具体的には，提案手法では移動ロボットの位置を表す二次元のコンフィギュレーション空間に移動ロボットの速度を表す二次元の速度空間を加えた四次元の状態空間を考える．本章ではロボットの位置のみを考慮したロードマップを P-roadmap と呼び，提案するロードマップを可変分解能速度マップ (Variable-resolution Velocity Roadmap, VVR) と呼ぶ．P-roadmap は既存の手法によって得られるとし，そのマップに速度情報を付加して可変分解能速度マップを構築する方法を提案する．VVR の各ノードは，位置と速度からなる状態空間上の領域を表す．ここで，位置は P-roadmap と同様に点で表現されるが，速度は区間として表現される．もしあるノードが代表する状態空間上の領域の任意の点から，別のあるノードが代表する領域上の任意の点へ移動する適切な制御入力が存在するとき，二つのノード間は移動可能と判断され，有向リンクで結ばれる．このようなロードマップの拡張により，動特性を考慮した経路計画が可能となる．また，経路計画段階で移動ロボットが経路を追従するための適切な制御入力も得られるため，オンラインでの計算コストを大幅に削減できる．

以下に本章の構成を示す．4.2 節において本章で想定する移動ロボットについて述べる．4.3 節において提案手法の一般的な枠組みについて述べ，4.4 節において対象とする移動ロボットモデルへ適用する際の詳細について述べる．4.5 節において数値シミュレーション結果について述べる．最後に 4.6 節においてまとめと今後の課題について述べる．

## 4.2 想定するシステム

本章では，障害物と自由空間によって構成される二次元平面上を全方向移動が可能な移動ロボットを想定する． $p \in P \subset \mathbb{R}^2$  を移動ロボットの位置とし， $v \in V \subset \mathbb{R}^2$  を移動ロボットの速度とする．ただし， $P$  と  $V$  はそれぞれロボットの位置と速度が取り得る値の範囲を示す有界な領域である．ここで，想定する移動ロボットの状態を  $x = [p^T, v^T]^T \in X$ ， $X = P \times V$  とする．

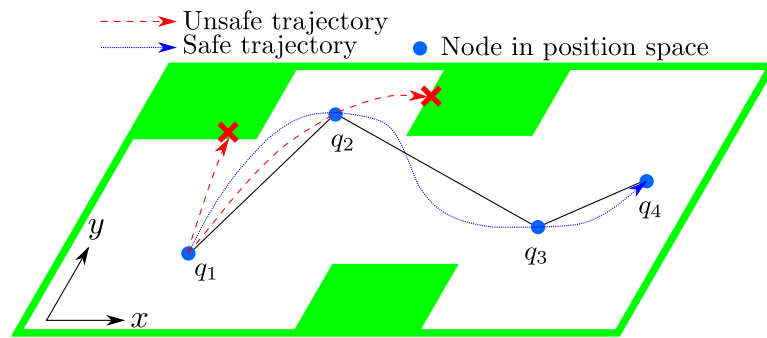
以下に本章の構成を示す．4.2.1 節において作業空間の定義について述べる．4.2.2 節において移動ロボットの制御則について述べる．4.2.3 節において移動ロボットの移動に関する制約条件について述べる．

### 4.2.1 作業空間上のロードマップ

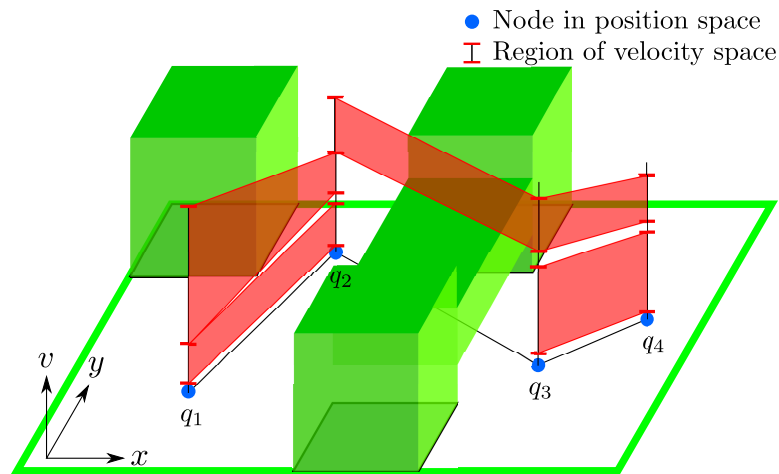
障害物  $P_{\text{obs}} \subset P$  と自由空間  $P_{\text{free}} \subset P$  で構成された作業空間を考える．自由空間上には P-roadmap  $\mathcal{M} = (P, \mathcal{L})$  が与えられており， $\mathcal{M}$  は連結グラフである (Fig. 4.1(a))．ここで， $P$  はノードの集合であり， $\mathcal{L} \subseteq [P]^2$  はリンクの集合である．Figure 4.1(a) では，点でノード  $q \in P$  を，点をつなぐ実線がリンクをそれぞれ表し，四角形が障害物を表す． $\mathcal{M}$  上の経路の集合を  $\Pi[\mathcal{M}] = \{(q_i)_{0:n} \mid n \in \mathbb{N}, q_i \in P, \forall i \in [0, n], (q_j, q_{j+1}) \in \mathcal{L}, \forall j \in [0, n-1]\}$  と書き，二地点  $q, q'$  間をむすぶ経路の集合を  $\Pi[\mathcal{M}](q, q') = \{(q_i)_{0:n} \in \Pi[\mathcal{M}] \mid q_0 = q, q_n = q'\}$  と書く．また，作業空間に関する以下の情報も与えられているとする．各  $(q, q') \in \mathcal{L}$  について，二点間の距離を  $d(q, q') = \|q' - q\| > 0$  とし， $(q, q')$  に対して垂直方向の最近傍障害物までの距離を  $w(q, q') > 0$  とする．また， $q$  と  $q'$  に対して接線方向の最近傍障害物までの距離をそれぞれ  $\epsilon_1(q, q') > 0$  および  $\epsilon_2(q, q') > 0$  とする．この作業空間において， $w(q, q')$ ， $\epsilon_1(q, q')$ ， $\epsilon_2(q, q')$  の値に応じて移動ロボットが十分低速であれば， $\mathcal{M}$  を用いて計画された任意の経路を障害物に衝突せずに走行可能であるとする．

### 4.2.2 全方向移動ロボットのモデルと制御則

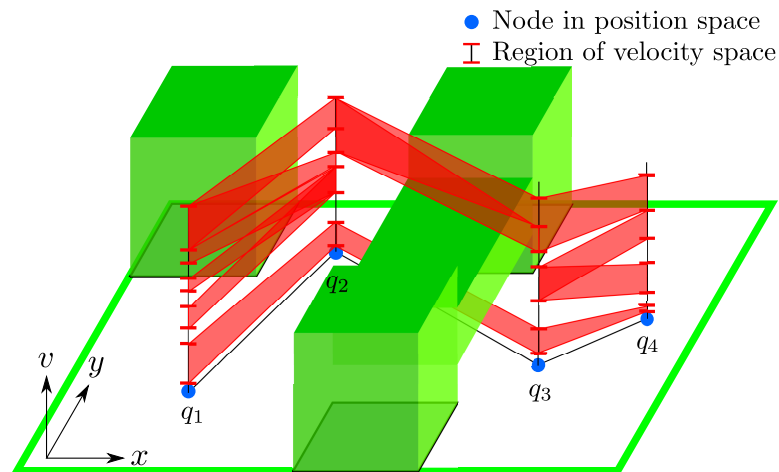
本節では作業空間に関する用語を述べ，移動ロボットの動特性と制御則について述べる．移動ロボットが  $\mathcal{M}$  上の隣接する  $q$  から  $q'$  に移動することについて考える (Fig. 4.2)．Figure 4.2(a) はグローバル座標系を表し，Fig. 4.2(b) は  $q$  を原点とし，線分  $qq'$  の接続方向を  $x$  軸，線分の垂直方向を  $y$  軸とするローカル座標系を表す． $(\hat{q}^X, \hat{q}^Y)$  と  $(q^x, q^y)$



(a) Illustration of a configuration roadmap



(b) Illustration of augmentation of velocity information



(c) Illustration of augmentation of velocity information in finely

Fig. 4.1 Illustration of extension a configuration roadmap to a variable-resolution velocity roadmap : green parts depict obstacles in the  $(x, y, v)$  space. Note that obstacles in the  $(x, y, v)$  space can not calculate easily.

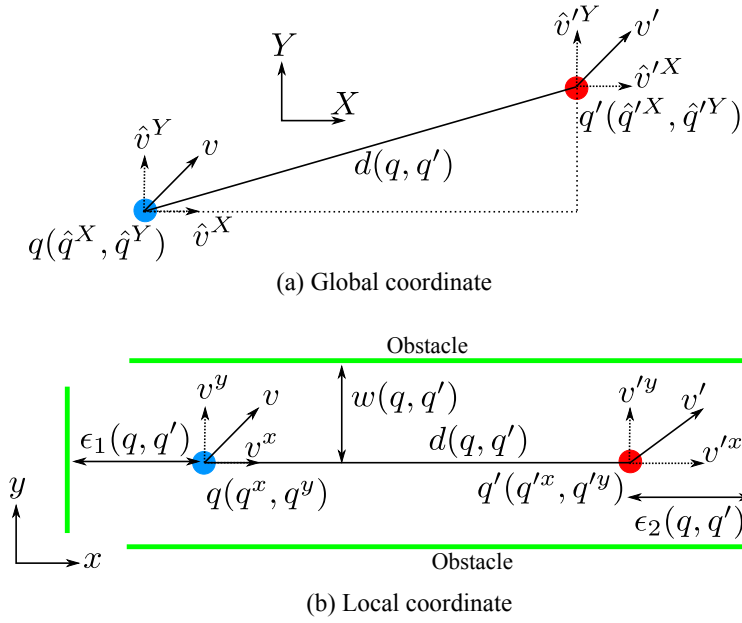


Fig. 4.2 Definition of information between two points

をそれぞれグローバル座標およびローカル座標とする．また， $q, q'$  を通過する瞬間の速度をそれぞれ  $v, v'$  とする．この時，グローバル座標系とローカル座標系の関係は以下の式で示される：

$$\begin{aligned} \begin{bmatrix} v^x \\ v^y \end{bmatrix} &= \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \hat{v}^X \\ \hat{v}^Y \end{bmatrix}, \\ \theta &= \arctan \left( \frac{\hat{q}'^Y - \hat{q}^Y}{\hat{q}'^X - \hat{q}^X} \right). \end{aligned} \quad (4.1)$$

なお， $(\hat{v}^X, \hat{v}^Y)$  はグローバル座標系での速度であり， $(v^x, v^y)$  はローカル座標系での速度である．本節の以降の議論ではローカル座標系を用いる．

本章では移動ロボットが二次元平面上を全方向移動可能であると想定しているため，次に示すロボットの運動方程式を得る：

$$\ddot{p}(t) = u(t), \quad (4.2)$$

なお， $p(t) = [x(t), y(t)]^T$  は時刻  $t$  での移動ロボットの位置であり， $u(t) = [u^x(t), u^y(t)]^T$  は時刻  $t$  での移動ロボットへ与える加速度入力である．決められた二地点間を移動する時のロボットの制御則を次に示す．後述する提案手法では任意に指定した始点と終点の位置・速度に対して軌道が一意に定まる性質，およびその二地点間を安全に移動できるそれぞれの地点での速度の範囲が求まる性質が要求させられるため，その性質が満たされるように  $x$  軸方向に関しては等加速度運動を行うとし， $y$  軸方向に関しては二段階の等加速度



運動を行うとする：

$$u(t) = \begin{cases} u_1, & \text{if } t \in [0, T/2), \\ u_2, & \text{otherwise.} \end{cases} \quad (4.3)$$

なお， $u_1 = [u^x, u_1^y]^T$ ， $u_2 = [u^x, u_2^y]^T$  はそれぞれ定数であり， $T \in \mathbb{R}^+$  は二地点間を移動するのに要する時間である．次に示す境界条件を考える：

$$\begin{aligned} p(0) &= (0, 0), & p(T) &= (d(q, q'), 0), \\ v(0) &= v, & v(T) &= v'. \end{aligned} \quad (4.4)$$

なお， $v(t) = [v^x(t), v^y(t)]^T$  は時刻  $t$  での移動ロボットの速度である．境界条件より， $u^x$ ， $u_1^y$ ， $u_2^y$  と  $T$  はそれぞれ以下のように求まる：

$$u_1 = \left[ \frac{-v^x + v'^x}{T}, -\frac{3v^y + v'^y}{T} \right]^T, \quad (4.5a)$$

$$u_2 = \left[ \frac{-v^x + v'^x}{T}, \frac{v^y + 3v'^y}{T} \right]^T, \quad (4.5b)$$

$$T = \frac{2d(q, q')}{v^x + v'^x}. \quad (4.5c)$$

これは制御入力  $u(t)$  と  $T$  は  $q$ ， $v$ ， $q'$  および  $v'$  が決定すれば一意に定まることを意味する．さらに，軌道  $p(t), t \in [0, T]$  と速度  $v(t), t \in [0, T]$  は  $u(t)$  と  $T$  によって一意に定まる：

$$\begin{aligned} p(t) &= \begin{cases} \frac{1}{2}u_1 t^2 + vt, & \text{if } t \in [0, T/2), \\ \frac{1}{2}u_2 t^2 + \left( \left( \frac{u_1}{2} - \frac{u_2}{4} \right) T + v \right) t + \frac{u_2 - u_1}{8} T^2, & \text{otherwise,} \end{cases} \\ v(t) &= \begin{cases} u_1 t + v, & \text{if } t \in [0, T/2), \\ u_2 t + (u_1 - u_2) \frac{T}{2} + v, & \text{otherwise.} \end{cases} \end{aligned} \quad (4.6)$$

そこで，これを関数として  $(u(t), v(t), p(t), T) = F(q, v, q', v')$  と書く．

### 4.2.3 移動に関する制約条件

本節ではローカル座標系を用いて移動ロボットが満たすべき制約条件について述べる．ロボットの移動において，二種類の制約条件を考える．一つは安全についての制約（以下，安全制約）であり，もう一つはロボットの性能についての制約（以下，性能制約）である．

まず, Fig. 4.2(b) を用いて安全制約について述べる. 線分  $qq'$  から大きく逸脱すると近傍の障害物に衝突するおそれがあるため, 移動ロボットが安全に二地点間を移動するためには, 線分  $qq'$  の  $y$  軸方向の移動に関しては線分から  $w(q, q')$  よりも遠く離れてはならない. また, 始点  $q$  と終点  $q'$  の近傍において線分からそれぞれ  $\epsilon_1(q, q')$ ,  $\epsilon_2(q, q')$  よりも遠く離れてはならない. 以上より次の条件が得られる:

$$|y(t)| \leq w(q, q'), \quad \forall t \in [0, T], \quad (4.7a)$$

$$q^x - \epsilon_1(q, q') \leq x(t) \leq q'^x + \epsilon_2(q, q'), \quad \forall t \in [0, T]. \quad (4.7b)$$

次に性能制約について述べる. 全ての移動ロボットには速度と加速度の上限が存在する. ここで  $V_{\max}$  を速度の上限とし,  $A_{\max}$  を加速度の上限とすると以下の条件が得られる:

$$|v(t)| \leq V_{\max}, \quad \forall t \in [0, T], \quad (4.8a)$$

$$|u(t)| \leq A_{\max}, \quad \forall t \in [0, T]. \quad (4.8b)$$

上述のように, 始点と終点における位置と速度を定めれば中間の軌道が一意に定まる. また, 軌道には安全性と性能に由来する一連の制約条件が課せられる. 以上の議論にもとづき, 始点・終点对の実行可能性を以下のように定義する:

**Definition 1** (Feasibility). もし  $(q, v)$  と  $(q', v')$  の組が与える軌道  $(u(t), v(t), p(t), T) = F(q, v, q', v')$  が制約条件 (4.7) 式と (4.8) 式を満たす場合, その組は実行可能 (*feasible*) であるという. また, 系列  $((q_0, v_0), (q_1, v_1), \dots, (q_n, v_n))$  について, すべての  $((q_i, v_i), (q_{i+1}, v_{i+1}))$ ,  $i = 0, 1, \dots, n-1$  が実行可能であるとき, この系列は実行可能であるという.

**Definition 2** (Feasible Region).  $q$  と  $q'$  が与えられた時,  $((q, v), (q', v'))$  が実行可能であるような  $(v, v')$  の集合を実行可能速度領域と呼び,  $\mathcal{F}_v(q, q')$  と書く.

もし  $\mathcal{F}_v(q, q') = \emptyset$  ならば, 始点と終点における速度をどのように選択しても, 制約条件を満たして  $q$  から  $q'$  へ移動することができないことを意味する.

### 4.3 可変分解能速度マップ生成手法

本節において可変分解能速度マップの生成法について述べる. Figure 4.1(a) には  $q_1$  から  $q_4$  への経路が存在する. ロボットがこの経路を追従するには各ノード上で適切な速度を選択する必要がある. なぜならば, 速度が不適切であると図中の Unsafe trajectory のように障害物に衝突したり, 速度制約や加速度制約を超過するおそれがあるからである.

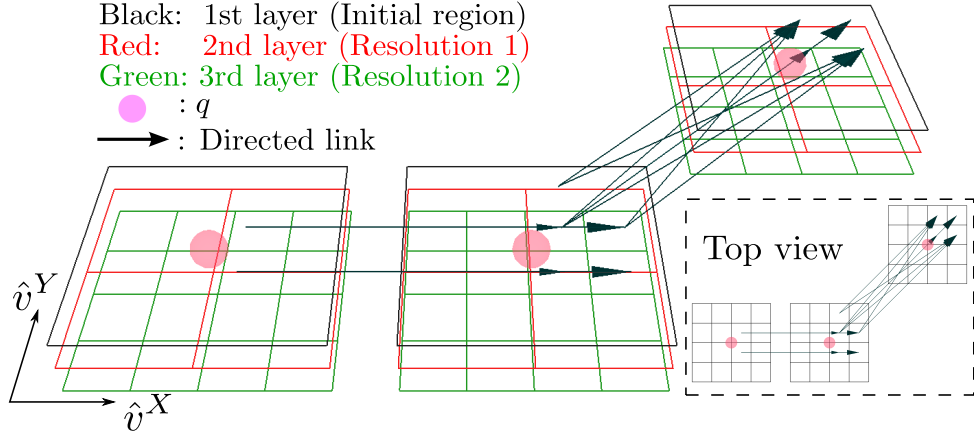


Fig. 4.3 Illustration of Variable-resolution Velocity Roadmap

そこで、本手法では、P-roadmap に対して実行可能な速度領域の情報を付加することで適切な速度の選択を可能にする (Figs. 4.1(b), 4.1(c))。なお、表現の都合上、Fig. 4.1 では速度を一次元とみなして三次元で示す。

以下に本節の構成を示す。4.3.1 節において提案するロードマップの定義について述べる。4.3.2 節において提案するロードマップでの有向リンクの生成条件について述べる。4.3.3 節において速度空間の分割方法について述べる。

#### 4.3.1 速度情報を持つ拡張ロードマップの定義

本節では速度情報を用いて P-roadmap  $\mathcal{M} = (\mathcal{P}, \mathcal{L})$  から可変分解能速度マップ (VVR) への拡張方法について述べる。提案するロードマップを  $\mathcal{M}_v = (\mathcal{X}_v, \mathcal{L}_v)$  で表記する (Fig. 4.3)。ここで  $\mathcal{X}_v$  はノードの集合であり、各  $x \in \mathcal{X}_v$  は位置と速度領域との二項組  $x = (q, \bar{v})$ ,  $q \in \mathcal{P}$ ,  $\bar{v} \subset V$  で与えられる。また、位置  $q$  に対応する速度領域の集合を  $\mathcal{V}(q) = \{\bar{v} \mid (q, \bar{v}) \in \mathcal{X}_v\}$  とする。 $\mathcal{L}_v \subseteq [\mathcal{X}_v]^2$  は有向リンクの集合である。 $\mathcal{M}_v$  の経路の集合を  $\Pi[\mathcal{M}_v] = \{(\mathbf{x}_i)_{0:n} \mid n \in \mathbb{N}, \mathbf{x}_i \in \mathcal{X}_v, \forall i \in [0, n], (\mathbf{x}_j, \mathbf{x}_{j+1}) \in \mathcal{L}_v, \forall j \in [0, n-1]\}$  と書く。ただし、速度領域  $\bar{v}$  は矩形であるとする。また、位置  $q$  から  $q'$  への  $\mathcal{M}_v$  上の経路の集合を  $\Pi[\mathcal{M}_v](q, q') = \{(\mathbf{x}_i)_{0:n} \in \Pi[\mathcal{M}_v] \mid q_0 = q, q_n = q'\}$  と書く。ノードの接続性について次のように定義する：

**Definition 3** (Connectivity of State Nodes).  $\mathcal{M}_v = (\mathcal{X}_v, \mathcal{L}_v)$  のノード  $x = (q, \bar{v})$  と  $x' = (q', \bar{v}')$  について以下が成り立つ：

$$(x, x') \in \mathcal{L}_v \Leftrightarrow (q, q') \in \mathcal{L} \text{ and } (\bar{v}, \bar{v}') \subseteq \mathcal{F}_v(q, q').$$

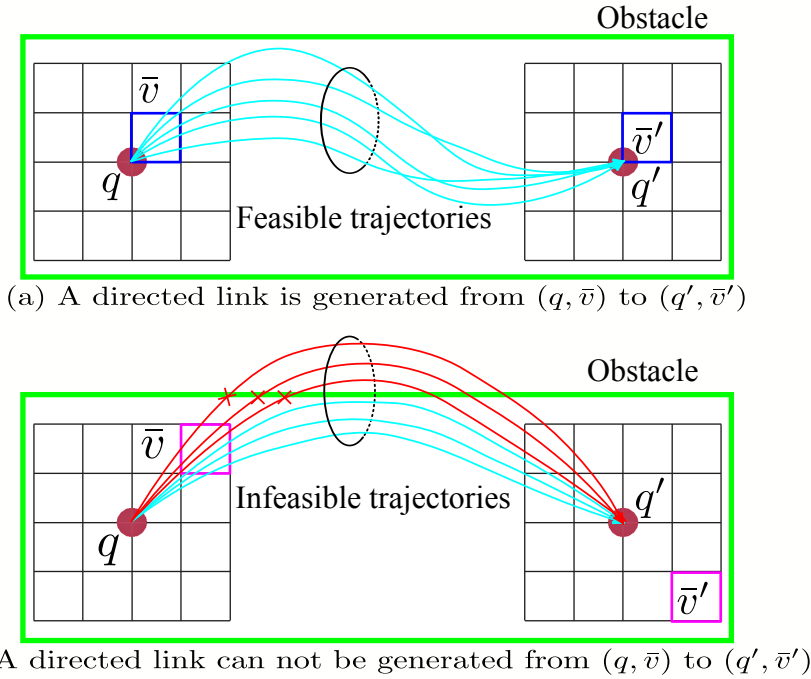


Fig. 4.4 Condition for a directed link generation

つまり、任意の  $v \in \bar{v} \in \mathcal{V}(q)$ ,  $v' \in \bar{v}' \in \mathcal{V}(q')$  について  $((q, v), (q', v'))$  が実行可能であるとき、 $x$  から  $x'$  へ向けて有向リンクが定義される (Fig. 4.4)。

上述のノードの接続性より次の補題が得られる：

**Lemma 1.**  $\mathcal{M}_v$  上の経路  $(x_i)_{0:n} \in \Pi[\mathcal{M}_v]$ ,  $x_i = (q_i, \bar{v}_i)$  を考える．このとき任意の系列  $((q_0, v_0), (q_1, v_1), \dots, (q_n, v_n))$ ,  $v_i \in \bar{v}_i, \forall i = 0, 1, \dots, n$  は実行可能である．

*Proof.* Definition 3 より自明である． □

すなわち、 $\mathcal{M}_v$  の任意の経路は実行可能な軌道を与える．

### 4.3.2 実行可能条件の評価

本節では実行可能条件  $(\bar{v} \times \bar{v}' \subseteq \mathcal{F}_v(q, q'))$  の判定に関する一般的な方法を述べる．一般的に、 $\mathcal{F}_v(q, q')$  が凸な領域であるとは限らないため、 $\bar{v} \times \bar{v}' \subseteq \mathcal{F}_v(q, q')$  を厳密に判定することは困難である (Fig. 4.5)．そこで、新たに凸な部分集合  $\tilde{\mathcal{F}}_v(q, q') \subseteq \mathcal{F}_v(q, q')$  を導入する．本論文で取り扱う問題における  $\tilde{\mathcal{F}}_v(q, q')$  の定義は後程詳しく述べる．なお、 $\bar{v}$  は矩形であるため、 $\bar{v} \times \bar{v}'$  は四次元の超直方体である．したがって、 $\bar{v} \times \bar{v}'$  のすべての頂

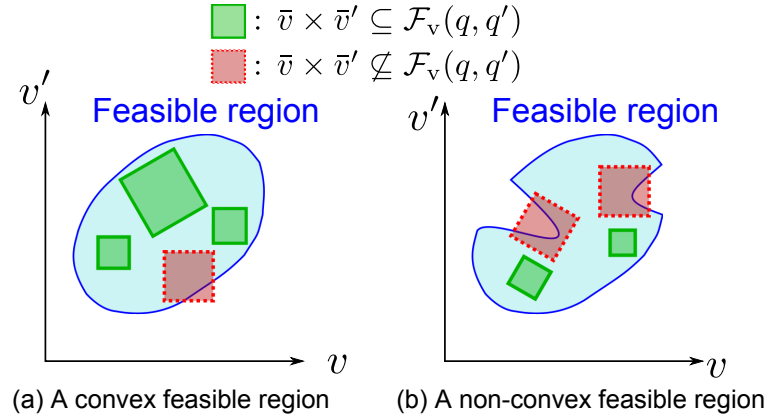


Fig. 4.5 Examples of feasible region

点が  $\tilde{\mathcal{F}}_v(q, q')$  に含まれていれば  $\bar{v} \times \bar{v}' \subseteq \tilde{\mathcal{F}}_v(q, q')$  であることは容易に示せる．ここで，超直方体  $\bar{v} \times \bar{v}'$  の頂点の集合を  $\mathcal{V}_v(\bar{v}, \bar{v}')$  とすると，以上の事柄から次の補題が得られる：

**Lemma 2.**  $\mathcal{V}_v(\bar{v}, \bar{v}')$  を  $\bar{v} \times \bar{v}'$  の頂点の集合とする．

$$\mathcal{V}_v(\bar{v}, \bar{v}') \subseteq \tilde{\mathcal{F}}_v(q, q') \Rightarrow \bar{v} \times \bar{v}' \subseteq \mathcal{F}_v(q, q').$$

*Proof.* 凸集合の定義より自明である． □

Lemma 2 より， $\bar{v} \times \bar{v}'$  の頂点のみを評価すれば十分であることが分かる．これによって，一組の  $(x, x')$  に対して，制約条件の評価を高々 16 回行えば良いということが分かる．

### 4.3.3 可変分解能速度空間分割

有向リンクの接続性は速度空間の分解能に依存する．例えば，Fig. 4.1(b) では分解能が不十分であるために， $q_1$  から  $q_4$  への速度を考慮した実行可能な経路が存在しない．それに対して，Fig. 4.1(c) では十分高い分解能であるため， $q_1$  から  $q_4$  への速度を考慮して実行可能な経路が存在する．一般に，速度空間の分解能を高めるほど実行可能な経路は増える．一方，分解能が過度に高いと，経路探索における探索木の規模が膨大となり探索時間の増大につながる．しかしながら，一般的に，速度空間をどのような粒度に分割すれば良いのかは事前には分からない．そこで，本論文では可変分解能に基づく分割手法を提案する．提案手法では，可変分解能処理によって速度空間をクアッドツリーで表現する．ある  $q \in \mathcal{P}$  における速度空間の分割例を Fig. 4.6 に示す．ここで，空間の分割度合を分割レベルと定義する (Fig. 4.6(a))．なお，速度領域の分割はグローバル座標系で行われる点

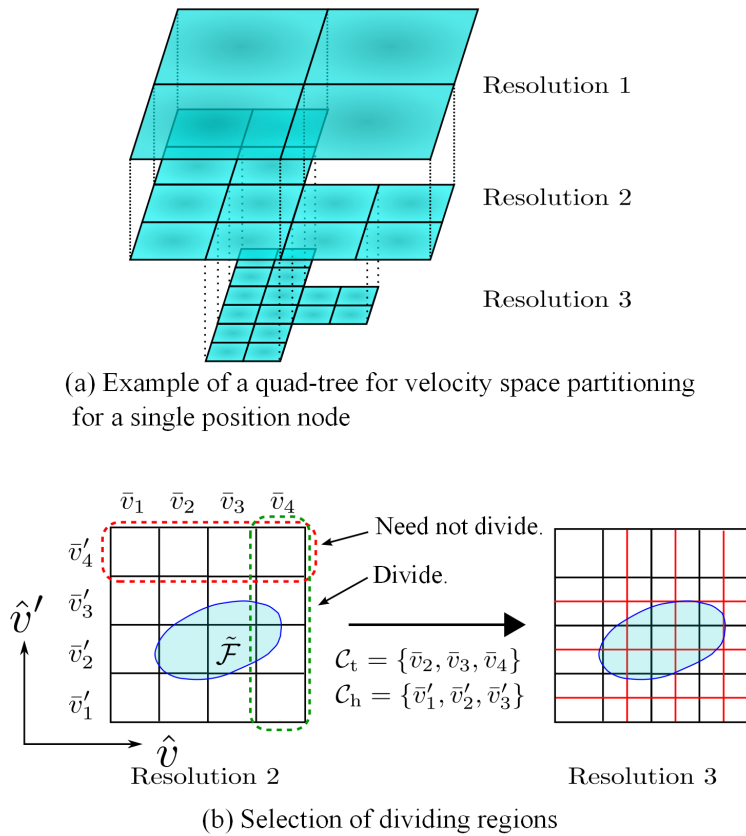


Fig. 4.6 An example of variable-resolution partitioning of velocity space

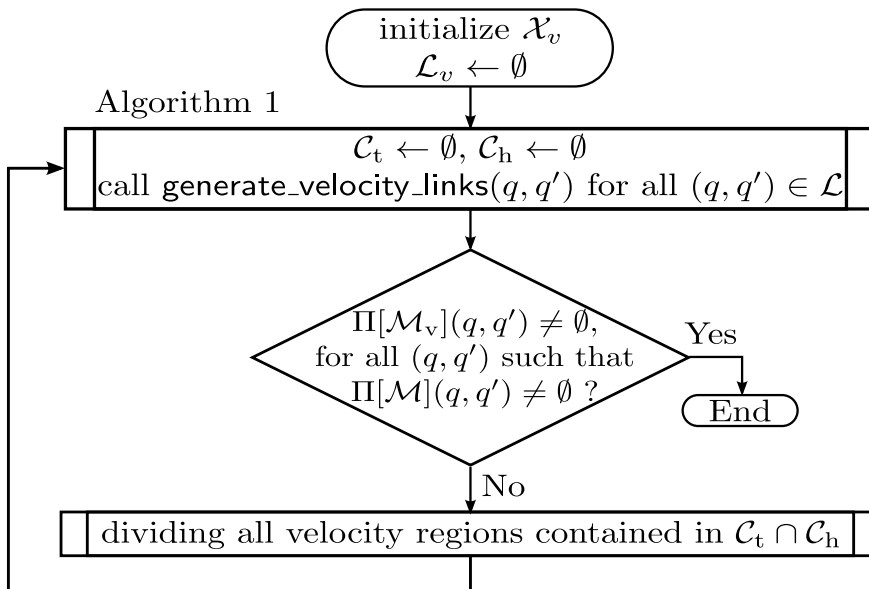


Fig. 4.7 Flowchart of overall procedure

について注意されたい．これは，ローカル座標系は隣接ノードごとに向きが異なるために，分割表現に適さないからである．

分割処理全体の流れを Fig. 4.7 に示す．初期状態では各  $q \in \mathcal{P}$  での速度空間が四等分されているとする（分割レベル 1）．次に，すべての組  $(q, q') \in \mathcal{L}$  について Algorithm 1 に示す `generate_velocity_links( $q, q'$ )` を実行する．なお，Algorithm 1 内の  $\mathcal{C}_t$  と  $\mathcal{C}_h$  は速度領域を格納する集合である．この時，Algorithm 1 の line 3–12 で各領域が分割すべき領域であるかどうかの判断も行い，分割すべき領域であると判断された領域  $\bar{v} \in \mathcal{V}(q)$  は  $\mathcal{C}_t$  に追加され， $\bar{v}' \in \mathcal{V}(q')$  は  $\mathcal{C}_h$  に追加される．なお，実際の分割は後述する終了条件が満たされなかった場合に行う．分割すべきかどうかの判断基準は対象の速度領域の頂点の集合  $\mathcal{V}_v(\bar{v}, \bar{v}')$  と  $\tilde{\mathcal{F}}_v(q, q')$  の関係にもとづく (Fig. 4.6(b)) ．

$\tilde{\mathcal{F}}_v(q, q')$  が  $\mathcal{V}_v(\bar{v}, \bar{v}')$  の全体を包含：  $(q, \bar{v})$  から  $(q', \bar{v}')$  に有向リンクを生成するため，該当領域を分割対象としない．

$\tilde{\mathcal{F}}_v(q, q')$  が  $\mathcal{V}_v(\bar{v}, \bar{v}')$  の一部を包含： 分割を行うことで  $\tilde{\mathcal{F}}_v(q, q')$  に包含される領域を生成できるため，該当領域を分割対象とする．

$\tilde{\mathcal{F}}_v(q, q')$  が  $\mathcal{V}_v(\bar{v}, \bar{v}')$  を非包含： 分割を行っても  $\tilde{\mathcal{F}}_v(q, q')$  に包含される領域を生成できないため，該当領域を分割対象としない．

この処理により，不必要な領域分割を行わなくて済む．Algorithm 1 終了後に P-roadmap 上の任意の二点  $q, q'$  に対応する経路が VVR 上に存在するかどうかを確認する．もしこの条件が成立すれば，速度空間の分割を終了する．そうでない場合， $\mathcal{C}_t \cap \mathcal{C}_h$  に含まれる全ての速度領域の分割を行う． $\mathcal{C}_t \cap \mathcal{C}_h$  に含まれる速度領域のみを分割するのは， $\mathcal{C}_t$  と  $\mathcal{C}_h$  のどちらかにしか含まれない速度領域は  $\mathcal{M}_v$  上の経路の端点にしかなりえないからである．本章では始点と終点において速度が 0 である軌道のみを対象とするため，このような速度領域を分割対象から除外する．領域の再分割を終えたら再度有向リンクの生成を行う．この一連の処理を終了条件が満たされるまで繰り返す．仮定より， $\mathcal{M}$  上の全ての経路が十分低速であれば制約を満たしつつ走行可能であるので，ある程度以上速度領域の分割が施された時点で必ず終了条件が満足される．

## 4.4 実行可能領域の厳密な導出

本章では，4.2 節で述べた全方向移動ロボットに関して実行可能速度領域  $\mathcal{F}_v(q, q')$  を厳密に導出する．また，非凸性を有する制約条件については  $\mathcal{F}_v(q, q')$  に包含される凸集合  $\tilde{\mathcal{F}}_v(q, q')$  の定義方法についても述べる．Fig. 4.2(b) と 4.2.1 節で定義した変数を用いて，ロボットの移動に関する制約条件を導出する．具体的には移動時間，衝突回避，速度制限

---

**Algorithm 1** generate\_velocity\_links( $q, q'$ )

---

```

1: for all  $(\bar{v}, \bar{v}') \in \mathcal{V}(q) \times \mathcal{V}(q')$  do
2:    $\mathcal{V}_v(\bar{v}, \bar{v}') \leftarrow$  set of vertices of  $\bar{v} \times \bar{v}'$ 
3:   if  $\mathcal{V}_v(\bar{v}, \bar{v}') \subset \tilde{\mathcal{F}}_v(q, q')$  then
4:     make directed link between  $(q, \bar{v})$  and  $(q', \bar{v}')$ 
5:   else if  $(\mathcal{V}_v(\bar{v}, \bar{v}') \cap \tilde{\mathcal{F}}_v(q, q')) \neq \emptyset$  then
6:     if  $\bar{v}$  is a leaf of quad-tree, then
7:        $\mathcal{C}_t \leftarrow \mathcal{C}_t \cup \{\bar{v}\}$ 
8:     end if
9:     if  $\bar{v}'$  is a leaf of quad-tree, then
10:       $\mathcal{C}_h \leftarrow \mathcal{C}_h \cup \{\bar{v}'\}$ 
11:    end if
12:   end if
13: end for

```

---

および加速度制限の 4 部に分かれる．それぞれの条件から導出される実行可能領域の積集合が想定する全方向移動ロボットの実行可能領域である．

#### 4.4.1 移動時間

移動時間  $T$  は正である必要があるため，(4.5) 式より次式が得られる：

$$v^x + v'^x > 0. \quad (4.9)$$

#### 4.4.2 衝突回避

本節では (4.7) 式について考える．最初に  $y$  軸方向の制約条件である (4.7a) 式について考える．なお，制約条件の全ての導出は複雑であるので，主に Fig. 4.8 を用いて説明する．

移動ロボットの  $y$  軸方向の軌道は大きく二種類に分類できる．一つは時刻  $t_\alpha$  ( $0 \leq t_\alpha \leq \frac{T}{2}$ ) において最も線分  $qq'$  から遠ざかる軌道，もう一つは時刻  $t_\beta$  ( $\frac{T}{2} \leq t_\beta \leq T$ ) において最も線分  $qq'$  から遠ざかる軌道である．なお， $y$  軸方向の軌道は  $x$  軸に対して線対称であるため，Fig. 4.8 には  $0 \leq v^y$  の場合についてのみ示す．以上を踏まえると以下の式が得



られる :

$$\begin{aligned}
 y_{\max} &= \begin{cases} |y(t_\alpha)|, & \text{if } -v^y \leq v'^y \leq v^y, \\ |y(t_\beta)|, & \text{otherwise,} \end{cases} \\
 y(t_\alpha) &= \frac{(v^y)^2 d(q, q')}{(v^x + v'^x)(3v^y + v'^y)}, \\
 y(t_\beta) &= -\frac{(v'^y)^2 d(q, q')}{(v^x + v'^x)(v^y + 3v'^y)}. \tag{4.10}
 \end{aligned}$$

なお,  $y_{\max}$  は線分  $qq'$  からの距離の最大値である. ここで (4.7a) 式, (4.10) 式より速度に関する不等式条件として次式が得られ, その領域を Fig. 4.8 に示す :

$$\begin{cases} 0 \leq (v^x + v'^x)w(q, q') + \frac{(v'^y)^2}{v^y + 3v'^y}d(q, q'), & \text{if } v'^y < -v^y, \\ 0 \leq (v^x + v'^x)w(q, q') - \frac{(v^y)^2}{3v^y + v'^y}d(q, q'), & \text{if } -v^y \leq v'^y \leq v^y, \\ 0 \leq (v^x + v'^x)w(q, q') - \frac{(v'^y)^2}{v^y + 3v'^y}d(q, q'), & \text{if } v^y < v'^y. \end{cases} \tag{4.11}$$

この時, Fig. 4.8 より  $v^y v'^y$  平面では任意の  $v^x$  に対する 3つの領域の和集合が凸集合であると判断できる. なお, 選択した領域が凸集合であることはヘッセ行列を用いて評価することもできる.  $v^y < 0$  に対しても同様な計算を行うことで,  $y$  軸方向に関する実行可能領域  $\mathcal{F}_v^y(q, q')$  を導出することができる.

次に  $x$  軸方向の制約条件を意味する (4.7b) 式について考える. (4.7b) 式および (4.5) 式を用いると速度に関する不等式条件として次式が得られ, その領域を Fig. 4.9 に示す :

$$\begin{cases} \text{don't care,} & \text{if } v^x > 0, v'^x > 0, \\ -\sqrt{d(q, q') + \epsilon_2(q, q')}v'^x \leq \sqrt{\epsilon_2(q, q')}v^x, & \text{if } v^x > 0, v'^x \leq 0, \\ -\sqrt{d(q, q') + \epsilon_1(q, q')}v^x \leq \sqrt{\epsilon_1(q, q')}v'^x, & \text{if } v^x \leq 0, v'^x > 0. \end{cases} \tag{4.12}$$

Figure 4.9 より (4.12) 式によって定められる領域が実行可能領域  $\mathcal{F}_v^x(q, q')$  であり, 凸集合であることが分かる.

#### 4.4.3 速度制限

本節では (4.8a) 式について考える. 本論文ではロボットの制御則を二段階の等加速度運動と想定しているため, 速度が最大となり得る時刻は  $t = 0, \frac{T}{2}, T$  である. したがって,

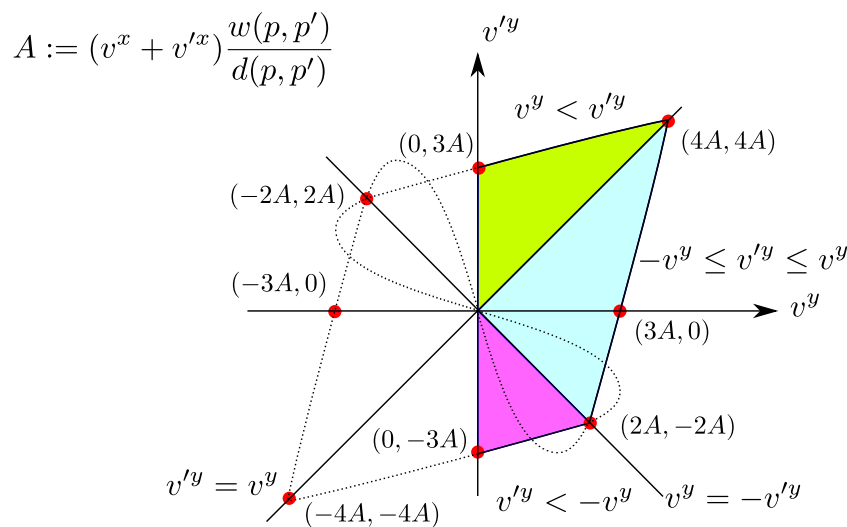


Fig. 4.8 A feasible velocity region satisfying condition (4.11)

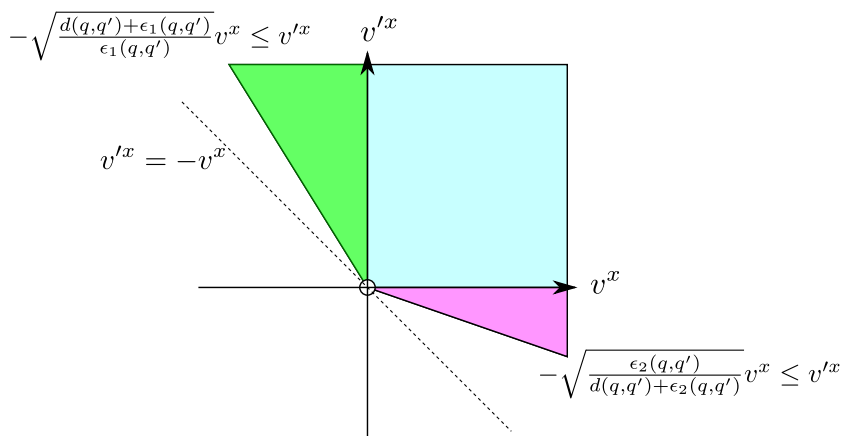


Fig. 4.9 A feasible velocity region satisfying condition (4.12)

次に示す全ての条件式が成立すればよい：

$$(v^x)^2 + (v^y)^2 \leq V_{\max}^2, \quad (4.13a)$$

$$\left(v^x + u^x \frac{T}{2}\right)^2 + \left(v^y + u^y \frac{T}{2}\right)^2 \leq V_{\max}^2, \quad (4.13b)$$

$$(v'^x)^2 + (v'^y)^2 \leq V_{\max}^2. \quad (4.13c)$$

(4.13a) 式および (4.13c) 式は領域が円周上およびその内部を表しているため，明らかに凸な領域である．一方，(4.13b) 式については (4.5) 式を用いることで次式が得られる：

$$\begin{aligned} 0 &< \left(v^x + u^x \frac{T}{2}\right)^2 + \left(v^y + u_1^y \frac{T}{2}\right)^2 \\ &= \left\{\frac{v^x + v'^x}{2}\right\}^2 + \left\{\frac{-v^y + v'^y}{2}\right\}^2 \leq V_{\max}^2. \end{aligned} \quad (4.14)$$

上式のヘッセ行列を評価することで (4.14) 式を満たす領域が凸な領域であることを確認できる．

以上より，(4.13a) 式，(4.13c) 式および (4.14) 式の三式を満たす領域が速度制約に関する実行可能領域  $\mathcal{F}_v^y(q, q')$  である．

#### 4.4.4 加速度制限

本節では (4.8b) 式について述べる．(4.8b) 式を直接用いた領域の導出が複雑であるため，次式に示すように x 軸方向と y 軸方向に分けて考える：

$$|u^x| \leq \frac{A_{\max}}{\sqrt{2}}, \quad |u_i^y| \leq \frac{A_{\max}}{\sqrt{2}}, \quad i \in \{1, 2\}. \quad (4.15)$$

最初に， $u^x$  について考える．(4.5) 式を用いると次式が得られる：

$$-\sqrt{2}d(q, q')A_{\max} \leq (v'^x)^2 - (v^x)^2 \leq \sqrt{2}d(q, q')A_{\max}. \quad (4.16)$$

(4.16) 式を満たす  $(v^x, v'^x)$  の領域を Fig. 4.10 に示す．Fig. 4.10 より，求められた実行可能速度領域  $\mathcal{F}_v^{u^x}(q, q')$  が凸な領域ではないことが分かる．提案手法では実行可能な速度領域が凸な集合である必要があるため，4.3.2 節に示すように凸な部分集合  $\tilde{\mathcal{F}}_v^{u^x}(q, q')$  を準備する．そこで， $\tilde{\mathcal{F}}_v^{u^x}(q, q')$  として領域の境界に内接する矩形領域を次式のように定義する：

$$\begin{aligned} -\sqrt{\sqrt{2}d(q, q')A_{\max}} &\leq v^x \leq \sqrt{\sqrt{2}d(q, q')A_{\max}}, \\ -\sqrt{\sqrt{2}d(q, q')A_{\max}} &\leq v'^x \leq \sqrt{\sqrt{2}d(q, q')A_{\max}}. \end{aligned} \quad (4.17)$$

次に， $u_1^y$  について考える．まず  $u^x$  と同様に (4.5a) 式および (4.5c) 式を用いると次式が得られる：

$$|u_1^y| = \left| \frac{-(3v^y + v'^y)(v^x + v'^x)}{2d(q, q')} \right| \leq \frac{A_{\max}}{\sqrt{2}}. \quad (4.18)$$

(4.18) 式を満たす  $(\frac{v^x + v'^x}{2}, \frac{3v^y + v'^y}{4})$  の領域  $\mathcal{F}_v^{u_1^y}(q, q')$  を Fig. 4.11 に示す．なお，4.4.1

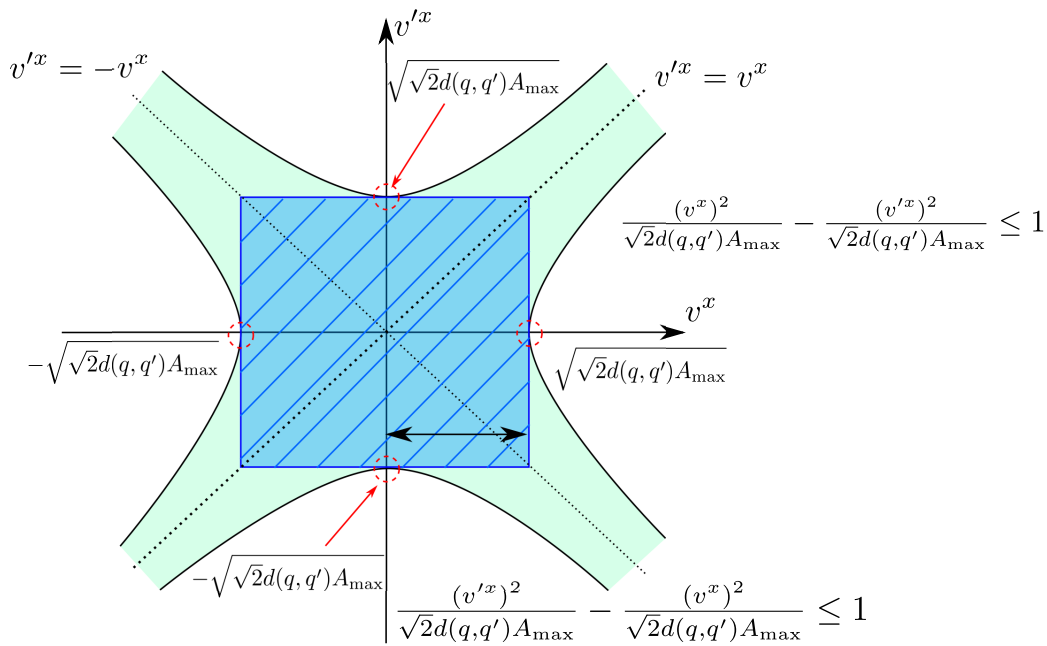


Fig. 4.10 A Feasible velocity region satisfying an acceleration condition (4.17): the hatched rectangular region depict  $\tilde{\mathcal{F}}_v(q, q')$ .

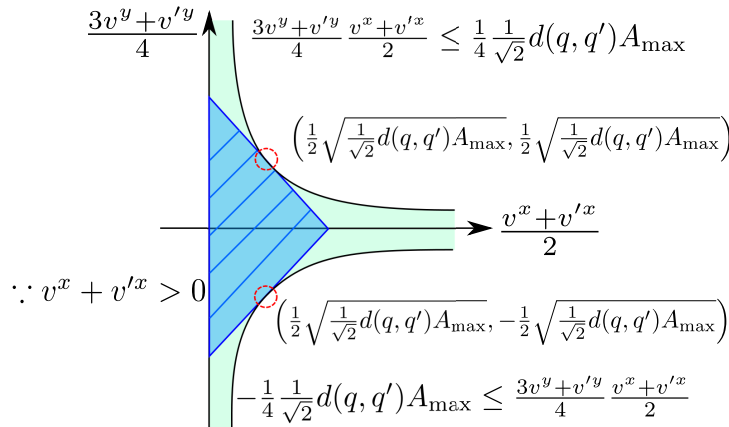


Fig. 4.11 A Feasible velocity region satisfying an acceleration condition (4.19): the hatched rectangular region depict  $\tilde{\mathcal{F}}_v(q, q')$

節で導出された  $v^x + v'^x > 0$  という条件を考慮する . Fig. 4.11 より ,  $\mathcal{F}_v^{u_1^y}(q, q')$  が非凸な領域であることが分かるため ,  $u^x$  の場合と同様に凸な部分集合  $\tilde{\mathcal{F}}_v^{u_1^y}(q, q')$  を次式のよ

うに定義する：

$$\begin{aligned}\frac{3v^y + v'^y}{4} &\leq -\frac{v^x + v'^x}{2} + \sqrt{\frac{d(q, q')A_{\max}}{\sqrt{2}}}, \\ \frac{3v^y + v'^y}{4} &\geq \frac{v^x + v'^x}{2} - \sqrt{\frac{d(q, q')A_{\max}}{\sqrt{2}}}.\end{aligned}\quad (4.19)$$

最後に， $u_2^y$  についても同様に考えることで実行可能領域  $\mathcal{F}_v^{u_2^y}(q, q')$  を導出できる：

$$\begin{aligned}\frac{v^y + 3v'^y}{4} &\leq -\frac{v^x + v'^x}{2} + \sqrt{\frac{d(q, q')A_{\max}}{\sqrt{2}}}, \\ \frac{v^y + 3v'^y}{4} &\geq \frac{v^x + v'^x}{2} - \sqrt{\frac{d(q, q')A_{\max}}{\sqrt{2}}}.\end{aligned}\quad (4.20)$$

## 4.5 数値例

本章においていくつかの例題に対する数値シミュレーション結果を示す．最初に二地点間の移動に注目し，どのように速度空間の分割とリンク生成が行われるかを Fig. 4.12 に示す．なお，ローカル座標系の  $x$  軸とグローバル座標系の  $X$  軸が平行であるとする．二地点間の距離を  $d(q, q') = 500$  とし，経路幅  $w(q, q')$  としては 100, 60 および 30 の三通りの場合を考える．また，移動ロボットの最大速度を  $V_{\max} = 400$  とし，最大加速度を  $A_{\max} = 400$  とする．速度領域の範囲を  $-200 \leq \hat{v}^X \leq 200$ ， $-200 \leq \hat{v}^Y \leq 200$  とする．図中の矢印は有向リンクを意味する．

分割レベルが 1 の時，どの場合もまだ有向リンクが生成されていない．分割レベルが 2 になると変化が現れる． $w(q, q') = 60, 30$  の場合はまだ有向リンクが生成されていないのに対して， $w(q, q') = 100$  では有向リンクが生成されている．分割レベルが 3 になると速度空間の分割にも変化が現れる．分割された領域は  $w(q, q') = 100, 60, 30$  の順に減少していることが確認できる．また，生成された有向リンクの本数も  $w(q, q') = 100, 60, 30$  の順に減少していることが確認できる．このように，経路幅が広いほど速度空間を高い粒度で分割しなくても有向リンクが生成されることが分かる．これは経路幅が広いほど選択可能な速度の範囲が広いことを意味し，対照的に経路幅が狭いほど選択可能な速度の範囲が狭いことを意味する．

次に事前に与えられた P-roadmap (Fig. 4.13(a)) に対して VVR (Fig. 4.13(b)) の生成を行う．次に，生成した VVR を用いて移動ロボットの軌道計画を行う．Figure 4.13(a) では，点が P-roadmap の各ノードを表し，直線分が各リンクを表す．Figure 4.13(b) で

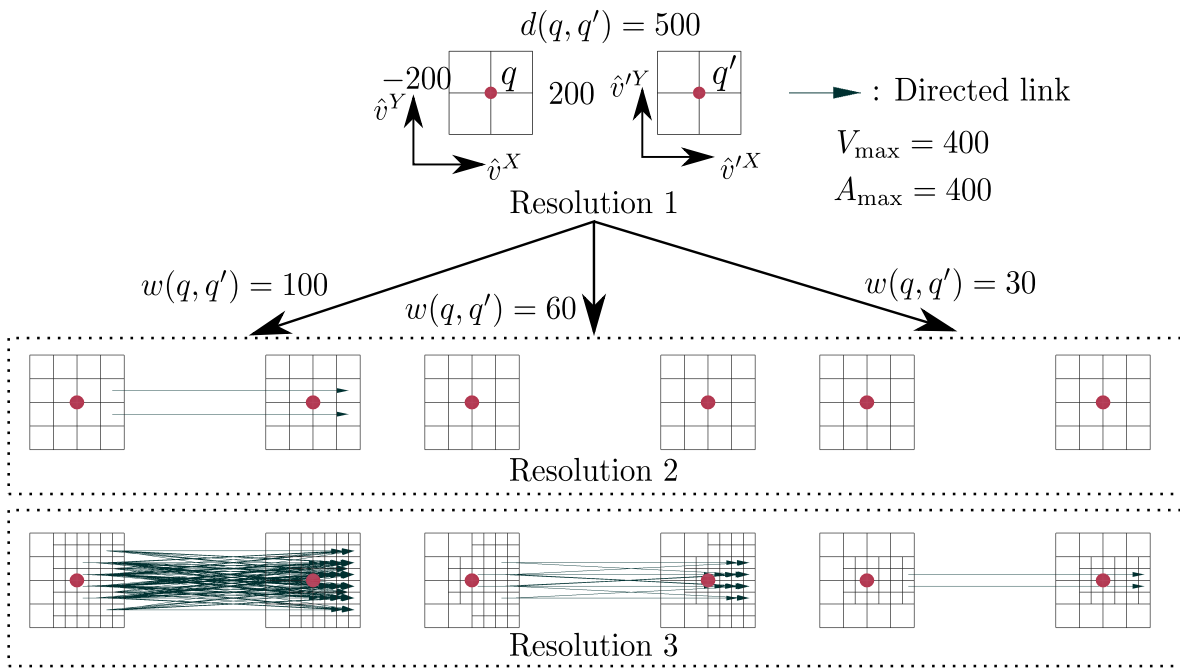


Fig. 4.12 An example of dividing velocity space and directed link generation

は、点が P-roadmap の各ノードを表し、各点周りの四角形が各点での分割された速度領域を表し、速度領域の分割レベルは 5 である。なお、表現の都合上、有向リンクは省略している。移動ロボットの最大速度を  $V_{\max} = 200$  とし、最大加速度を  $A_{\max} = 70$  とする。速度領域の範囲を  $-200 \leq \hat{v}^X \leq 200$ ,  $-200 \leq \hat{v}^Y \leq 200$  とする。全経路幅を  $\omega(q, q') = 150$  とする。

生成された VVR (Fig. 4.13(b)) について説明する。P-roadmap の端点  $((0, 0), (0, 1200), (1500, 0)$  と  $(1400, 600))$  と内点 (前記以外の位置) には速度空間の分割に違いがある。端点ではリンクの接続方向の速度領域が分割されず、リンクに対して垂直方向の速度領域のみが多く分割されている。一方、内点ではリンクの接続方向の速度領域も多く分割されている。これはそれぞれの位置を通過できる移動ロボットの速度の範囲を表している。このため、複数の方向に行き来可能な  $(1000, 500)$  では多くの速度領域が分割され、直角に折れ曲がっている  $(0, 500)$  および  $(1000, 0)$  では特定の方向の速度領域しか分割されていない。

生成された VVR (Fig. 4.13(b)) を用いて経路計画を行う (Fig. 4.14)。経路計画には 2.2.2 節で述べたダイクストラ法を用いる。Figure 4.14 では、曲線が軌道を表し、塗りつぶされた四角形が各位置で選ばれた速度領域を表し、矢印が各位置でのロボットの速度を表す。提案手法では (4.5) 式を用いることで移動ロボットの加速度入力が直接得られる。これは、経路計画の後にロボットの動特性を考慮して個別に軌道生成を行う必要のある文



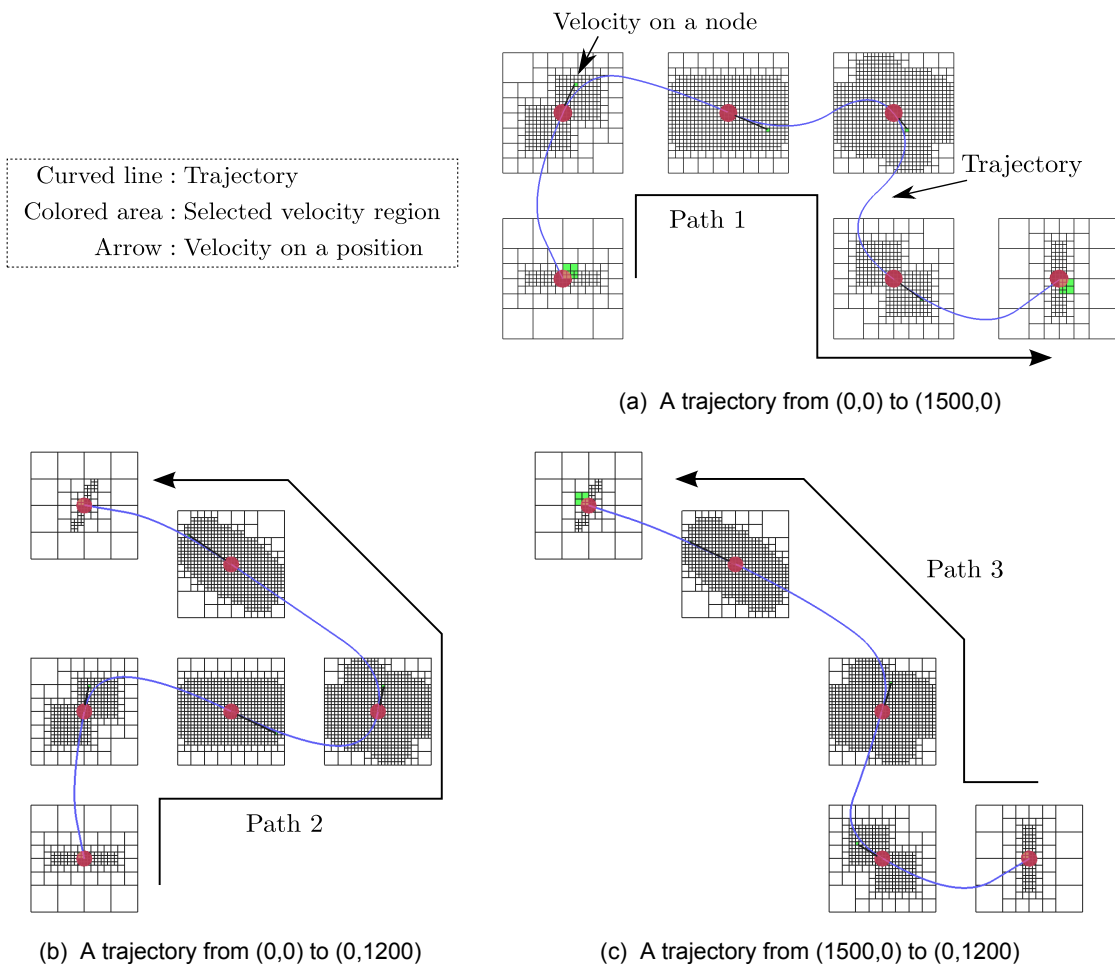


Fig. 4.14 Path planning using VVR (Fig. 4.13(b)) and trajectories

Figures 4.14(a) と (b) では通過する位置の系列の前半が同じであるが後半が異なる．これらに対して経路計画を行うと，(0, 500) 地点ですでに選択される速度領域が異なっていることが確認できる．次に異なる初期位置から同じ目標位置に対する経路計画結果を比較する (Figs. 4.14(b)(c)) ．Figures 4.14(b) と (c) では位置系列の前半は異なるが後半が同じである．これらに対して経路計画を行うと，(1000, 500) 地点以降では通過する位置は同じであるが，選択される速度領域が異なっていることが確認できる．これらの結果から，ある位置での適切な速度はその位置前後の位置系列に依存していることが分かる．提案する VVR では計画段階で各位置での適切な速度を算出することができる．

最後に分解能を固定して生成した速度地図との比較を行う．一般に，必要となる分解能が事前には未知であるため，高い分解能を指定する必要がある．分解能を 6 に固定した場合の速度マップを Fig. 4.15 に示す．移動ロボットの最大速度，最大加速度および速度領域の範囲は先の条件と同様とする．Table 4.1 に分解能を可変にした速度



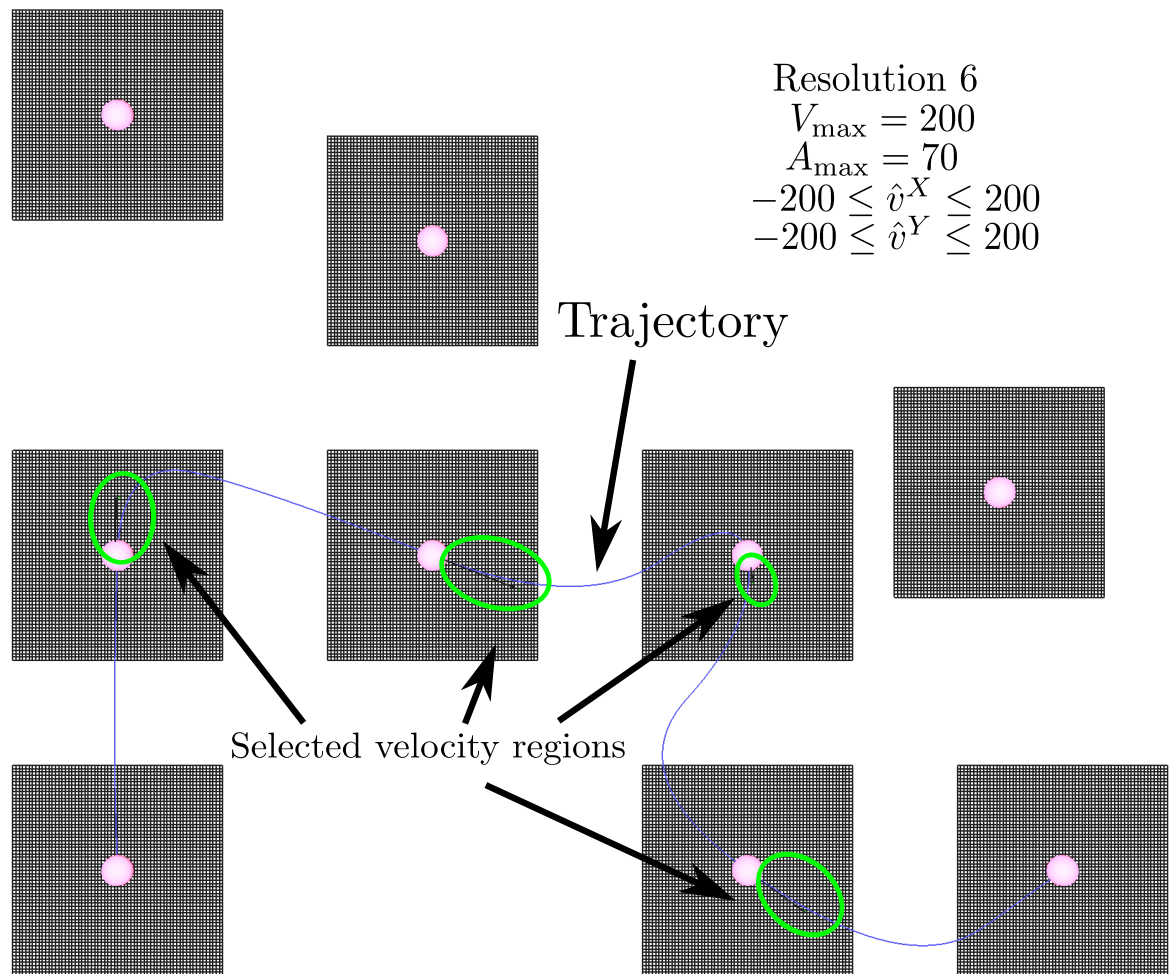


Fig. 4.15 A velocity roadmap with constant resolution 6 and the optimal trajectory

マップ (Fig. 4.13(b)) と分解能を一定にした速度マップ (Fig. 4.15) のそれぞれにおける Fig. 4.14(a) と同じ経路に対しての実行可能な経路数, 最短移動時間および計画時間を示す。なお, 計画には 2.2.2 節で述べたダイクストラ法を用いる。

VVR では Start から Destination までの軌道数が 2670 万個であるのに対して, 分解能を一定にした地図では軌道数が 9 億個を超えている。最短移動時間を比較すると分解能を一様に高くした地図の方の値が小さく, VVR の結果から 8% 短縮できることが分かる。しかしながら, 計画時間では分解能一定の方が VVR よりも 97 倍多くの時間がかかる。これは, 分解能一定の場合では単に細かく分割されるだけではなく, 経路に関与しない領域の分割も行われるためにツリー構造が膨張したのに対して, VVR では領域数が少なく, 地図生成段階で枝切りが行われたからである。以上より, 解の質のみを考慮した場合は分解能を一様に高くした方が良いが, 計算コストまで考慮した場合には VVR の方が優れていると言える。

Table 4.1 Variable resolution velocity roadmap and constant resolution velocity roadmap

Case	Resolution	Num. of Paths	Shortest transit time	Planning time
Variable	5	26.7 Million	40.76 (s)	34 (ms)
Constant	6	942 Million	37.54 (s)	$33 \times 10^2$ (ms)

## 4.6 まとめ

本章では、移動ロボットの位置と速度から成る状態空間上でロボットの動特性や種々の制約条件を考慮した新たなロードマップ表現とその構築方法を提案し、全方向移動ロボットへの適用例を示した。従来では、移動ロボットの動特性を考慮せずに経路計画を行っていたために、十分低速でなくては通れない経路を計画することもあるが、提案手法を用いることで、移動ロボットの制約条件を満足する経路を計画できるようになり、得られたグラフ経路よりロボットへの制御入力が直接得られる。可変分解能処理によって、作業環境の複雑さに応じて速度空間の分割粒度を変化させることができる。その結果として、不必要な空間細分化を抑えることができ、結果的に計算量を減らすことに繋がる。また、自動的に分解能を決定できるため、ヒューリスティック性を無くすことができる。

## 第 5 章

# 車両型移動ロボットの運動特性に基づく可変分解能姿勢マップ生成

### 5.1 はじめに

本章では、第 1 章で述べた車両型ロボットの動特性を考慮したロードマップ生成手法について述べる。第 4 章ではホロノミック拘束を有する移動ロボットの動特性を考慮した可変分解能速度マップの生成手法を提案したが、非ホロノミック拘束を有する移動ロボットには適用できない。そこで、本章では非ホロノミック拘束を有する移動ロボットの動特性を考慮したロードマップ生成手法を提案する。具体的には、提案手法では移動ロボットの位置を表す二次元の座標空間に移動ロボットの姿勢を表す一次元の姿勢空間を加えた三次元のコンフィギュレーション空間を考える。本論文ではロボットの位置のみを考慮したロードマップを P-roadmap と呼び、提案するロードマップを可変分解能姿勢マップ (Variable-resolution Orientation Roadmap, VOR) と呼ぶ。P-roadmap は第 4 章と同様に既存の手法によって得られるとし、そのマップに姿勢情報を付加して可変分解能姿勢マップを構築する方法を提案する。VOR の各ノードは、位置と姿勢からなる配位空間上の領域を表す。ここで、位置は P-roadmap と同様に点で表現されるが、姿勢は区間として表現される。もしあるノードが代表するコンフィギュレーション空間上の領域の任意の点から、別のあるノードが代表する領域上の任意の点へ移動する適切な軌道が存在するとき、二つのノード間は移動可能と判断され、有向リンクで結ばれる。このようなロードマップの拡張により、移動ロボットの姿勢をも考慮した経路計画が可能となる。また、経路計画段階で移動ロボットが経路を追従するための適切な軌道も得られるため、オンラインでの計算コストを大幅に削減できる。

さらに、提案ロードマップを用いた自動駐車システムを提案する。自動駐車システムは

数多く研究されており，複数段階の計画に分割する手法 [89], [90]，線分と円弧を用いて軌道を生成する手法 [91], [92]，駐車軌道のデータベースを生成して利用する手法 [93] やナビゲーションファンクションを利用する手法 [94] などがある．しかしながら，[89]～[92]で得られる軌道は自動車が追従できる軌道とは限らなく，データベースを利用する手法 [93] にはデータベースの容量と解の有無にトレードオフが存在し，ナビゲーションファンクションを用いた手法 [94] では切り替えしが自動には行えないなどの問題点がある．提案するロードマップを用いることでこれらの問題点を解決できることを示す．

以下に本章の構成を示す．5.2 節において本章で想定する車両型移動ロボットについて述べる．5.3 節において提案手法の一般的な枠組みについて述べ，5.4 節において対象とする移動ロボットモデルへ適用する際の詳細について述べる．5.5 節において数値シミュレーション結果について述べる．5.6 節において提案手法を用いた自動駐車システムについて述べる．最後に 5.7 節においてまとめと今後の課題について述べる．

## 5.2 想定するシステム

本章では障害物と自由空間によって構成される二次元平面上を移動する車両型移動ロボットを想定する． $p \in P \subset \mathbb{R}^2$  を移動ロボットの位置とし， $\phi \in \Phi \subset \mathbb{R}$  を移動ロボットの姿勢とする． $v \in V \subset \mathbb{R}^2$  を移動ロボットの線速度  $v$  と旋回角加速度  $\alpha$  で構成される二次元空間とする．ただし， $P$ ， $\Phi$  および  $V$  はそれぞれロボットの位置，姿勢，線速度および角加速度が取り得る値の範囲を示す有界な領域である．ここで，想定する移動ロボットの状態を  $x = [p, \phi, v]^T \in X$ ， $X = P \times \Phi \times V$  とする．

以下に本章の構成を示す．5.2.1 節において作業空間の定義について述べる．5.2.2 節において移動ロボットの制御則について述べ，5.2.3 節において軌道生成について述べる．5.2.4 節において移動ロボットの移動に関する制約条件について述べる．

### 5.2.1 作業空間上のロードマップ<sup>\*1</sup>

障害物  $P_{\text{obs}} \subset P$  と自由空間  $P_{\text{free}} \subset P$  で構成された作業空間を考える．自由空間上には P-roadmap  $\mathcal{M} = (P, \mathcal{L})$  が与えられており， $\mathcal{M}$  は連結グラフである (Fig. 4.1(a))．ここで， $P$  はノードの集合であり， $\mathcal{L} \subseteq [P]^2$  はリンクの集合である．Figure 4.1(a) では，点  $q \in P$  をノード  $q$  を，点をつなぐ実線がリンクをそれぞれ表し，四角形が障害物を表す． $\mathcal{M}$  上の経路の集合を

<sup>\*1</sup>作業空間の定義であり，4.2.1 節と同じである．

$\Pi[\mathcal{M}] = \{(q_i)_{0:n} \mid n \in \mathbb{N}, q_i \in \mathcal{P}, \forall i \in [0, n], (q_j, q_{j+1}) \in \mathcal{L}, \forall j \in [0, n-1]\}$  と書き，二地点  $q, q'$  間をむすぶ経路の集合を  $\Pi[\mathcal{M}](q, q') = \{(q_i)_{0:n} \in \Pi[\mathcal{M}] \mid q_0 = q, q_n = q'\}$  と書く．また，作業空間に関する以下の情報も与えられているとする．各  $(q, q') \in \mathcal{L}$  について，二点間の距離を  $d(q, q') = \|q' - q\| > 0$  とし， $(q, q')$  に対して垂直方向の最近傍障害物までの距離を  $w(q, q') > 0$  とする．また， $q$  と  $q'$  に対して接線方向の最近傍障害物までの距離をそれぞれ  $\epsilon_1(q, q') > 0$  および  $\epsilon_2(q, q') > 0$  とする．この作業空間において， $w(q, q')$ ， $\epsilon_1(q, q')$ ， $\epsilon_2(q, q')$  の値に応じて移動ロボットが十分低速であれば， $\mathcal{M}$  を用いて計画された任意の経路を障害物に衝突せずに走行可能であるとする．

### 5.2.2 車両型移動ロボットのモデルと制御則

本節では作業空間に関する用語を述べ，想定する移動ロボットのモデルと制御則について述べる．移動ロボットが  $\mathcal{M}$  上の隣接する  $q$  から  $q'$  に移動することについて考える (Fig. 5.1)．Figure 5.1 に示している  $(q^x, q^y)$  は位置  $q$  のグローバル座標であり， $\phi$  および  $\theta$  はそれぞれ  $q$  を通過する瞬間のロボットのグローバル座標系での姿勢および線分  $qq'$  に対するローカル座標系での姿勢である．また， $q(q')$  を通過する瞬間の線速度を  $v(v')$  とし， $q(q')$  を通過する瞬間の旋回角加速度を  $\alpha(\alpha')$  とする．車両型移動ロボットへの入力が線速度および旋回角加速度であるとした場合，次の運動方程式を得る (Fig. 5.2)：

$$\frac{d}{dt} \begin{bmatrix} x(t) \\ y(t) \\ \phi(t) \\ \dot{\phi}(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ y(t) \\ \phi(t) \\ \dot{\phi}(t) \end{bmatrix} + \begin{bmatrix} \cos \phi & 0 \\ \sin \phi & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} u(t). \quad (5.1)$$

ただし， $\phi(t)$  は時刻  $t$  での移動ロボットのグローバル座標系での姿勢であり， $u(t) = [u_v(t), u_\alpha(t)]^T$  は時刻  $t$  での移動ロボットへ与える線速度入力および旋回角加速度入力である．ここで，移動ロボットの時刻  $t$  での位置を  $p(t) = [x(t), y(t)]^T$  とする．この時，移動ロボットの入力が一定  $u(t) = [u_v(t), u_\alpha(t)]^T = [v_0, \alpha_0]^T$  であるとした場合，ロボットの移動軌道として次式を得る：

$$\begin{bmatrix} x(t) \\ y(t) \\ \phi(t) \end{bmatrix} = \begin{bmatrix} v_0 \int_0^t \cos\left(\frac{\alpha_0}{2}s^2\right) ds \\ v_0 \int_0^t \sin\left(\frac{\alpha_0}{2}s^2\right) ds \\ \frac{t^2}{2} \alpha_0 \end{bmatrix} = \begin{bmatrix} v_0 \sqrt{\frac{\pi}{|\alpha_0|}} C\left(\sqrt{\frac{|\alpha_0|}{\pi}} t\right) \\ v_0 \sqrt{\frac{\pi}{|\alpha_0|}} S\left(\sqrt{\frac{|\alpha_0|}{\pi}} t\right) \\ \frac{t^2}{2} \alpha_0 \end{bmatrix}. \quad (5.2)$$

ここで， $C(\cdot)$  および  $S(\cdot)$  はフレネル積分と呼ばれ，これによって得られる曲線はクロソイド曲線と呼ばれる．この時，グローバル座標系とローカル座標系の関係は以下の式で示

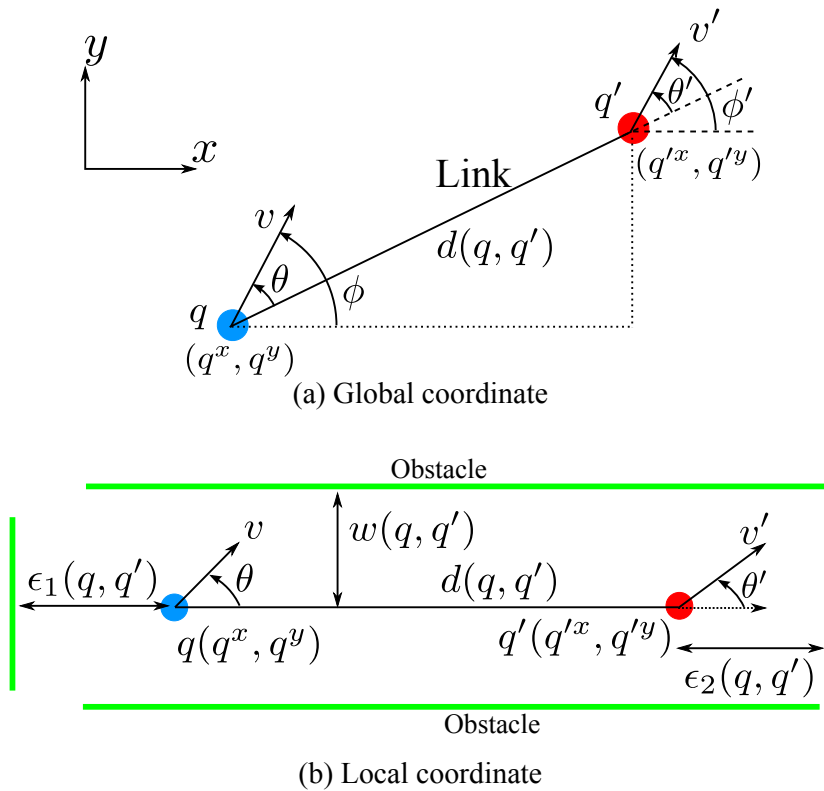


Fig. 5.1 Definition of information between two positions

される：

$$\theta = \phi - \arctan\left(\frac{q'^y - q^y}{q'^x - q^x}\right). \quad (5.3)$$

### 5.2.3 軌道生成

本節では、(5.2) 式を用いた軌道生成手法について述べる．なお、以降の議論はローカル座標系で行う．クロソイド曲線を用いた  $(q, \theta)$  から  $(q', \theta')$  への移動に関して次の境界条件を考える：

$$p(0) = q, \quad p(T) = q', \quad \theta(0) = \theta, \quad \theta(T) = \theta', \quad \omega(0) = 0, \quad \omega(T) = 0. \quad (5.4)$$

ただし、 $T \in \mathbb{R}^+$  は二地点間を移動するのに要する時間であり、 $\theta(t)$  は時刻  $t$  での移動ロボットのローカル座標系での姿勢であり、 $\omega(t)$  は時刻  $t$  での移動ロボットの旋回角速度である．本手法では二種類の軌道を用いる：一つは *Forward Trajectory* (F,F 軌道) であり、これは前進の軌道である．もう一つは *Forward-Forward Trajectory* (FF,FF 軌道)

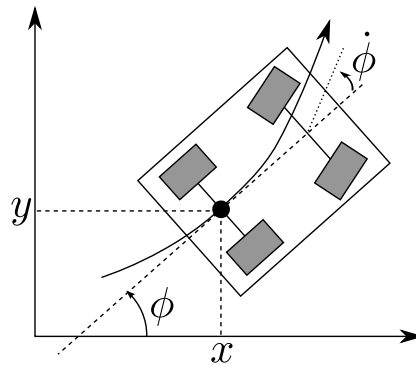
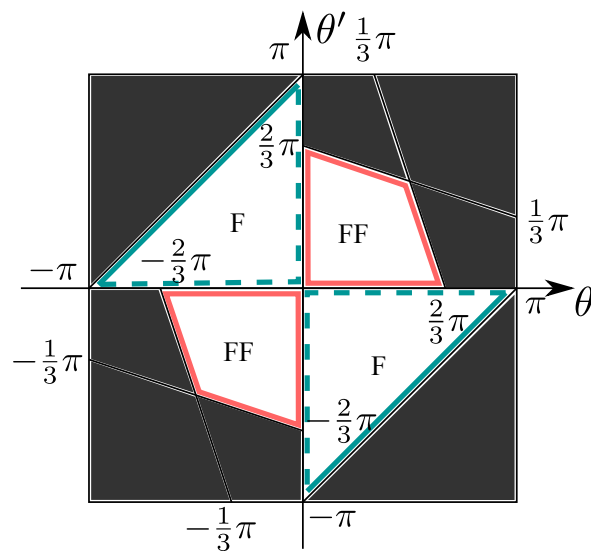


Fig. 5.2 Model of a car-like vehicle

Fig. 5.3 Trajectory map based on  $\theta$  and  $\theta'$ 

であり、これは F 軌道を二つ組み合わせた軌道である。これらの軌道は Fig. 5.3 に示すように  $\theta$  と  $\theta'$  によって場合分けされている。F 軌道はクロソイド曲線及び線分によって組み合わされた軌道であるため、車両型移動ロボットの動特性を考慮した軌道である。FF 軌道はその F 軌道を組み合わせた前進の軌道であるため、F 軌道と同様に車両型移動ロボットの動特性を考慮している。この二種類の軌道について述べる。

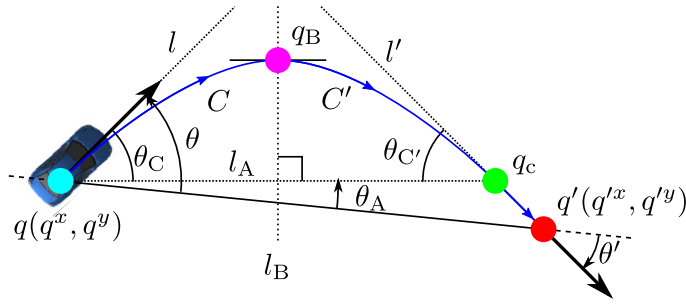


Fig. 5.4 Forward trajectory generation

### Forward trajectory

車両型移動ロボットが  $(q, \theta)$  から  $(q', \theta')$  に移動する F 軌道について述べる．この軌道は  $(\theta, \theta')$  について以下の条件

$$\begin{cases} 0 < \theta, \\ \theta - \pi \leq \theta' < 0, \end{cases} \quad (5.5)$$

または，

$$\begin{cases} \theta < 0, \\ 0 < \theta' \leq \theta + \pi, \end{cases} \quad (5.6)$$

を満たす．なお，対称性を有するため，(5.5) 式についてのみ詳細を述べる (Fig. 5.4) . Figure 5.4 では， $l$  は  $q$  から  $\theta$  方向に伸びた半直線であり， $l'$  は  $q'$  から  $\pi + \theta'$  方向に伸びた半直線であり， $l_A$  は  $\theta_C = \theta_{C'}$  を満たし，なおかつ  $l_A$  の端点の少なくとも片方が  $q$  もしくは  $q'$  であるような線分である．また， $\theta_A$  ( $-\frac{\pi}{2} < \theta_A < \frac{\pi}{2}$ ) は線分  $l_A$  と線分  $qq'$  の内角であり， $l_B$  は線分  $l_A$  の垂直二等分線である．なお， $0 < \theta_A < \frac{\pi}{2}$  の時は Fig. 5.4 に示されているように  $q$  が線分  $l_A$  の端点であり， $-\frac{\pi}{2} < \theta_A < 0$  の時は  $q'$  が線分  $l_A$  の端点であり， $\theta_A = 0$  の時は  $q$  と  $q'$  の両方が線分  $l_A$  の端点である．Figure 5.4 は  $0 < \theta_A < \frac{\pi}{2}$  の時を描画している．Figure 5.4 の軌道は二つのクロソイド曲線 ( $C$  と  $C'$ ) と線分  $qcq'$  を組み合わせたものである．クロソイド曲線  $C$  は次式で定義される：

$$\begin{aligned} \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} &= \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \tilde{x}(t) \\ \tilde{y}(t) \end{bmatrix} + q, \\ \begin{bmatrix} \tilde{x}(t) \\ \tilde{y}(t) \end{bmatrix} &= v \sqrt{\frac{\pi}{|\alpha|}} \begin{bmatrix} C(\sqrt{\frac{|\alpha|}{\pi}} t) \\ -S(\sqrt{\frac{|\alpha|}{\pi}} t) \end{bmatrix}. \end{aligned} \quad (5.7)$$



なお， $\alpha$  は曲線  $C$  上での旋回角加速度である．同様に，クロソイド曲線  $C'$  は次式で定義される：

$$\begin{aligned} \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} &= \begin{bmatrix} \cos \theta' & -\sin \theta' \\ \sin \theta' & \cos \theta' \end{bmatrix} \begin{bmatrix} \tilde{x}'(t) \\ \tilde{y}'(t) \end{bmatrix} + q_c, \\ \begin{bmatrix} \tilde{x}'(t) \\ \tilde{y}'(t) \end{bmatrix} &= -v' \sqrt{\frac{\pi}{|\alpha'|}} \begin{bmatrix} C\left(\sqrt{\frac{|\alpha'|}{\pi}} t\right) \\ S\left(\sqrt{\frac{|\alpha'|}{\pi}} t\right) \end{bmatrix}. \end{aligned} \quad (5.8)$$

なお， $\alpha'$  は曲線  $C'$  上での旋回角加速度である．ここで， $T_B \in \mathbb{R}^+$  を  $q$  から  $q_B$  への移動時間とし， $T_C \in \mathbb{R}^+$  を  $q$  から  $q_c$  への移動時間とすると，ロボットの  $q$  から  $q_c$  への移動軌跡は次のように得られる：

$$\begin{aligned} \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} &= \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \tilde{x}(t) \\ \tilde{y}(t) \end{bmatrix} + q, \\ \begin{bmatrix} \tilde{x}(t) \\ \tilde{y}(t) \end{bmatrix} &= v \sqrt{\frac{\pi}{|\alpha|}} \begin{bmatrix} C\left(\sqrt{\frac{|\alpha|}{\pi}} t\right) \\ -S\left(\sqrt{\frac{|\alpha|}{\pi}} t\right) \end{bmatrix}, & \text{if } t \in [0, T_B), \\ \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} &= \begin{bmatrix} \cos \theta' & -\sin \theta' \\ \sin \theta' & \cos \theta' \end{bmatrix} \begin{bmatrix} \tilde{x}'(t) \\ \tilde{y}'(t) \end{bmatrix} + q_c, \\ \begin{bmatrix} \tilde{x}'(t) \\ \tilde{y}'(t) \end{bmatrix} &= -v' \sqrt{\frac{\pi}{|\alpha'|}} \begin{bmatrix} C\left(\sqrt{\frac{|\alpha'|}{\pi}} (T_C - t)\right) \\ S\left(\sqrt{\frac{|\alpha'|}{\pi}} (T_C - t)\right) \end{bmatrix}, & \text{if } t \in [T_B, T_C). \end{aligned} \quad (5.9)$$

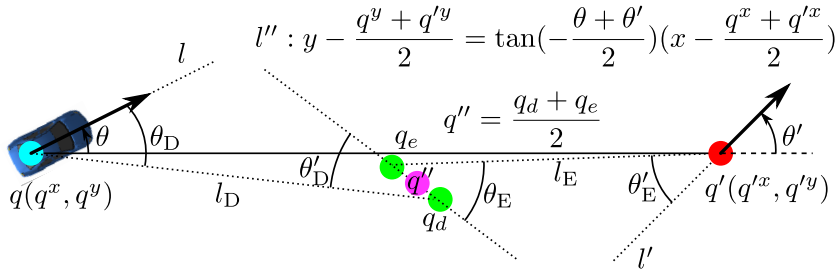
この時，クロソイド曲線の性質により，ロボットの姿勢  $\theta(t)$  および旋回角速度  $\omega(t)$  は  $q_B$  において連続である：

$$T_C = 2T_B, \quad (5.10a)$$

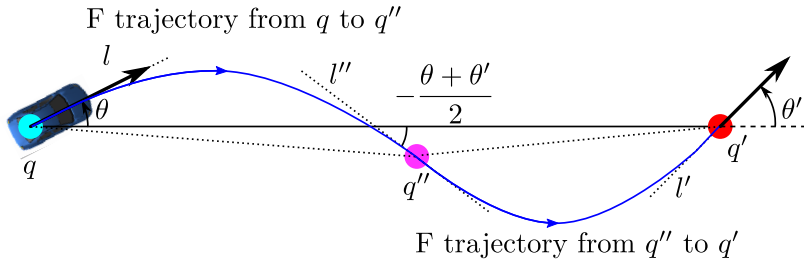
$$\theta(T_B) = \frac{\theta - \theta'}{2} = \theta_A, \quad (5.10b)$$

$$|\omega(T_B)| = |\alpha| T_B = |\alpha'| (T_C - T_B). \quad (5.10c)$$

さらに， $q$  と  $q_c$  がそれぞれのクロソイド曲線の原点であるため， $q$  と  $q_c$  での旋回角速度は  $\omega(0) = \omega(T_C) = 0$  である．また， $q_c$  から  $q'$  への軌道は直線分であるため， $q'$  での旋回角速度も 0 である．この手法によって生成された軌道では， $q \in \mathcal{P}$  上では旋回角速度は常に 0 である．



(a) Introduction of  $l''$  and  $q''$  for FF trajectory generation



(b) FF trajectory from  $q$  to  $q'$  via  $q''$

Fig. 5.5 Forward-Forward trajectory generation

### Forward-Forward Trajectory

車両型移動ロボットが  $(q, \theta)$  から  $(q', \theta')$  に移動する FF 軌道について述べる．この軌道は  $(\theta, \theta')$  について以下の条件

$$\begin{cases} 0 \leq \theta \leq -3\theta' + 2\pi, \\ 0 \leq \theta' \leq -3\theta + 2\pi, \end{cases} \quad (5.11)$$

または，

$$\begin{cases} -3\theta' - 2\pi \leq \theta \leq 0, \\ -3\theta - 2\pi \leq \theta' \leq 0, \end{cases} \quad (5.12)$$

を満たす．なお，対称性を有するため，(5.11) 式についてのみ詳細を述べる (Fig. 5.5)．これらの条件下では 5.2.3 節で述べた F 軌道を生成することができない．そこで Fig. 5.5(a) に示すような補助直線  $l''$  を用いて二つの F 軌道を組み合わせた軌道を生成する．補助直線  $l''$  は線分  $qq'$  の中心を通り，傾きが  $\tan(-(\theta + \theta')/2)$  であるような直線である．二つの F 軌道を生成するために  $l''$  上に  $q''$  を新たに導入する．まず， $\theta_D = \theta'_D$  となる線分  $l_D$  および， $\theta_E = \theta'_E$  となる線分  $l_E$  を導入する．そして， $l_D$  および  $l_E$  と  $l''$  との交点をそれぞれ  $q_d$  と  $q_e$  とし， $q_d$  と  $q_e$  の中点を  $q''$  とする．導入した  $q''$  に対して， $q$  から  $q''$  への

F 軌道と  $q''$  から  $q'$  への F 軌道を生成する (Fig. 5.5(b)) . この時 , F 軌道の始点と終点において旋回角速度が 0 であるため ,  $q''$  での旋回角速度は 0 であり , 軌道  $qq''$  から軌道  $q''q'$  に移行するときも旋回角速度は連続である .

#### 5.2.4 移動に関する制約条件

本節ではローカル座標系を用いて移動ロボットが満たすべき制約条件について述べる . ロボットの移動において , 二種類の制約条件を考える . 一つは安全についての制約 ( 以下 , 安全制約 ) であり , もう一つはロボットの性能についての制約 ( 以下 , 性能制約 ) である .

まず , 安全制約について述べる . 線分  $qq'$  から大きく逸脱すると近傍の障害物に衝突するおそれがあるため , 移動ロボットが安全に二点間を移動するためには , 線分  $qq'$  の法線方向の移動に関しては線分から  $w(q, q')$  よりも遠く離れてはならない . また , 始点  $q$  と終点  $q'$  の近傍において線分からそれぞれ  $\epsilon_1(q, q')$  ,  $\epsilon_2(q, q')$  よりも遠く離れてはならない . 以上より次の条件が得られる :

$$|y(t)| \leq w(q, q'), \quad \forall t \in [0, T], \quad (5.13a)$$

$$q^x - \epsilon_1(q, q') \leq x(t) \leq q'^x + \epsilon_2(q, q'), \quad \forall t \in [0, T]. \quad (5.13b)$$

次に性能制約について述べる . 全ての移動ロボットには速度と加速度の上限が存在する . ここで  $V_{\max}$  を速度の上限とし ,  $\Omega_{\max}$  を旋回角速度の上限とし ,  $A_{\max}$  を旋回角加速度の上限とすると以下の条件が得られる :

$$|v(t)| \leq V_{\max}, \quad \forall t \in [0, T], \quad (5.14a)$$

$$|\omega(t)| \leq \Omega_{\max}, \quad \forall t \in [0, T], \quad (5.14b)$$

$$|\alpha(t)| \leq A_{\max}, \quad \forall t \in [0, T]. \quad (5.14c)$$

上述のように , 始点と終点における位置と姿勢を定めれば中間の軌道が一意に定まる . また , 軌道には安全性と性能に由来する一連の制約条件が課せられる . 以上の議論にもとづき , 始点・終点对の実行可能性を以下のように定義する :

**Definition 4** (Feasibility). もし  $(q, \theta)$  と  $(q', \theta')$  の組が与える軌道

$$(u(t), p(t), \phi(t), \omega(t), T) = F_o(q, \theta, q', \theta')$$

が制約条件 (5.13) 式と (5.14) 式を満たす場合 , その組は実行可能 (*feasible*) であるという . また , 系列  $((q_0, \theta_0), (q_1, \theta_1), \dots, (q_n, \theta_n))$  について , すべての  $((q_i, \theta_i), (q_{i+1}, \theta_{i+1}))$  ,  $i = 0, 1, \dots, n-1$  が実行可能であるとき , この系列は実行可能であるという .

**Definition 5** (Feasible Region).  $q$  と  $q'$  が与えられた時,  $((q, \theta), (q', \theta'))$  が実行可能であるような  $(\theta, \theta')$  の集合を実行可能姿勢領域と呼び,  $\mathcal{F}_o(q, q')$  と書く.

もし  $\mathcal{F}_o(q, q') = \emptyset$  ならば, 始点と終点における速度をどのように選んでも, 制約条件を満たして  $q$  から  $q'$  へ移動することができないことを意味する.

## 5.3 可変分解能姿勢マップ生成手法

本節において可変分解能姿勢マップの生成法について述べる. 基本的な生成手法は第 4 章と同じであるため, ここでは異なる点についてのみ述べる. 第 4 章では, 一つの制約条件に対して一つの実行可能領域を定義している. それに対して, 本節では一つの制約条件に対して互いに素である複数の実行可能領域が定義される場合の対処について述べる. また, 第 4 章では実行可能領域が非凸の場合に凸な部分集合となる実行可能領域を導入したが, 本節では実行可能領域が非凸の場合においても, 導入する必要がない事例について述べる. なお, 以降の議論はローカル座標系で行う.

以下に本節の構成を示す. 5.3.1 節において提案するロードマップの定義について述べる. 5.3.2 節において提案するロードマップでの有向リンクの生成条件について述べる. 5.3.3 節において姿勢空間の分割方法について述べる.

### 5.3.1 姿勢情報を持つ拡張ロードマップの定義

本節では姿勢情報を用いて P-roadmap  $\mathcal{M} = (\mathcal{P}, \mathcal{L})$  から可変分解能姿勢マップ (VOR) への拡張方法について述べる. 提案するロードマップを  $\mathcal{M}_o = (\mathcal{X}_o, \mathcal{L}_o)$  で表記する (Fig. 5.6). ここで  $\mathcal{X}_o$  はノードの集合であり,  $x \in \mathcal{X}_o$  は位置と姿勢領域との二項組  $x = (q, \bar{\theta}), q \in \mathcal{P}, \bar{\theta} \subset \Theta$  で与えられる. また, 位置  $q$  に対応する姿勢領域の集合を  $\mathcal{O}(q) = \{\bar{\theta} \mid (q, \bar{\theta}) \in \mathcal{X}_o\}$  とする.  $\mathcal{L}_o \subseteq [\mathcal{X}_o]^2$  は有向リンクの集合である.  $\mathcal{M}_o$  の経路の集合を  $\Pi[\mathcal{M}_o] = \{(x_i)_{0:n} \mid n \in \mathbb{N}, x_i \in \mathcal{X}_o, \forall i \in [0, n], (x_j, x_{j+1}) \in \mathcal{L}_o, \forall j \in [0, n-1]\}$  と書く. ただし, 姿勢領域  $\bar{\theta}$  は線分であるとする. また, 位置  $q$  から  $q'$  への  $\mathcal{M}_o$  上の経路の集合を  $\Pi[\mathcal{M}_o](q, q') = \{(x_i)_{0:n} \in \Pi[\mathcal{M}_o] \mid q_0 = q, q_n = q'\}$  と書く. そして,  $q, q'$  での姿勢をグローバル座標系でそれぞれ  $\phi, \phi'$  と指定した時の経路の集合を  $\Pi[\mathcal{M}_o](q, q', \phi, \phi') = \{(x_i)_{0:n} \in \Pi[\mathcal{M}_o](q, q') \mid \phi \in \bar{\phi}, \phi' \in \bar{\phi}'\}$  と書く. ノードの接続性について次のように定義する:

**Definition 6** (Connectivity of Orientation Nodes).  $\mathcal{M}_o = (\mathcal{X}_o, \mathcal{L}_o)$  のノード  $x =$

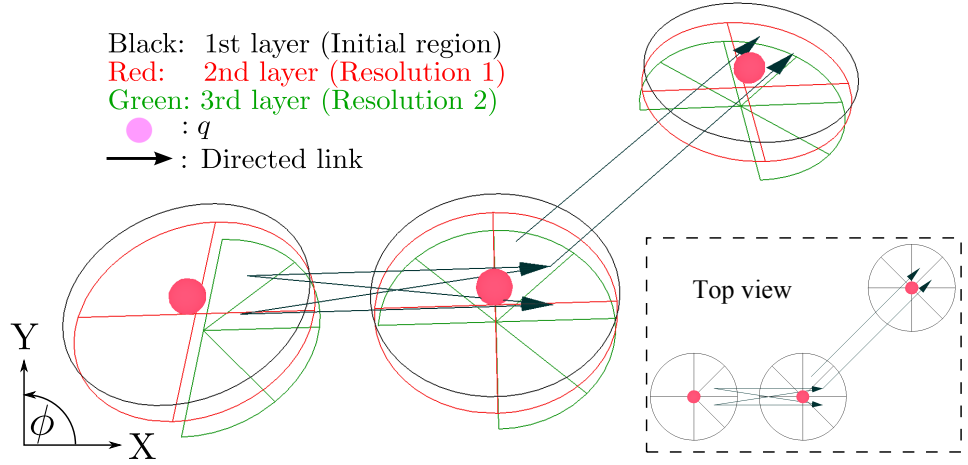
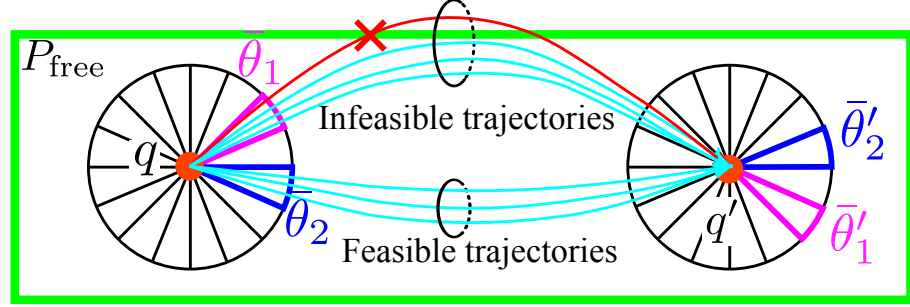


Fig. 5.6 Illustration of Variable-resolution Orientation Roadmap

A directed link can't be generated from  $(q, \bar{\theta}_1)$  to  $(q', \bar{\theta}'_1)$



A directed link is generated from  $(q, \bar{\theta}_2)$  to  $(q', \bar{\theta}'_2)$

Fig. 5.7 Feasible trajectories and Infeasible trajectories for car-like robots

$(q, \bar{\theta})$  と  $x' = (q', \bar{\theta}')$  について以下が成り立つ :

$$(x, x') \in \mathcal{L}_o \Leftrightarrow (q, q') \in \mathcal{L} \text{ and } (\bar{\theta}, \bar{\theta}') \subseteq \mathcal{F}_o(q, q').$$

つまり, 任意の  $\theta \in \bar{\theta}, \theta' \in \bar{\theta}'$  について  $((q, \theta), (q', \theta'))$  が実行可能であるとき,  $x$  から  $x'$  へ向けて有向リンクが定義される (Fig. 5.7) .

上述のノードの接続性より次の補題が得られる :

**Lemma 3.**  $\mathcal{M}_o$  上の経路  $(x_i)_{0:n} \in \Pi[\mathcal{M}_o]$ ,  $x_i = (q_i, \bar{\theta}_i)$  を考える . このとき任意の系列  $((q_0, \theta_0), (q_1, \theta_1), \dots, (q_n, \theta_n))$ ,  $\theta_i \in \bar{\theta}_i, \forall i = 0, 1, \dots, n$  は実行可能である .

*Proof.* Definition 6 より自明である . □

すなわち,  $M_0$  の任意の経路は実行可能な軌道を与える .

### 5.3.2 実行可能条件の評価

本節では実行可能条件 ( $\bar{\theta} \times \bar{\theta}' \subseteq \mathcal{F}_o(q, q')$ ) の判定方法を述べる .  $\mathcal{F}_o(q, q')$  が単一の凸領域であれば,  $\bar{\theta} \times \bar{\theta}'$  の 4 端点のみを確認すれば良いことは第 4 章で述べた . また, 単に実行可能領域は凸な領域でなくてはならないと述べた . それに対して, 本節では本手法で用いるための実行可能領域の具体的な必要条件について述べる . また, 次の状況についても議論する :

- $\mathcal{F}_o(q, q')$  が単一の領域ではなく, 互いに素な複数個 ( $M$  個) の領域の和集合である :

$$\mathcal{F}_o(q, q') = \bigcup_{i \in M} \mathcal{F}_{o,i}(q, q'), \quad M \in \mathbb{N}. \quad (5.15)$$

まず, 実行可能領域の具体的な必要条件について述べる . 本手法では, 各 2 地点間のローカル座標系に対して実行可能領域を定義している . 一方, 姿勢空間の分割はグローバル座標系で行う . ここで, ローカル座標系で定義される実行可能領域を  $\mathcal{R}(\theta, \theta') = \mathcal{F}_o(q, q')$  とし,  $\mathcal{R}(\theta, \theta')$  をグローバル座標系に変換したものを  $\mathcal{R}^G(\phi, \phi')$  とする .

新しい領域の性質を定義する :

**Definition 7** (Axis Convex Region).  $\mathbf{x}$  を系列  $(x_1, \dots, x_n)$  とし,  $\mathbf{x}_{-i}$  を系列  $(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$  とし,  $\mathcal{R}(\mathbf{x})$  を  $\mathbf{x}$  で定義される  $n$  次元の領域とする . もし, 領域  $\mathcal{R}(\mathbf{x})$  が次の条件を満たす場合, 領域  $\mathcal{R}(\mathbf{x})$  を軸凸領域と呼ぶ :

$$t\mathbf{x} + (1-t)(\mathbf{x}_{-i}, r) \in \mathcal{R}(\mathbf{x}), \quad \forall r(\mathbf{x}_{-i}, r) \in \mathcal{R}(\mathbf{x}), \quad \forall t \in [0, 1], \quad \forall i \in [1, n].$$

この領域は厳密に凸な領域ではないが, 各軸  $x_i$  に対して凸である . この性質により次の補題が得られる :

**Lemma 4.**  $\bar{\mathbf{x}}$  を  $n$  次元の超直方体  $\bar{x}_1 \times \dots \times \bar{x}_n$  とし,  $\mathcal{V}(\bar{\mathbf{x}})$  を  $\bar{\mathbf{x}}$  の頂点の集合とし,  $\mathcal{R}(\mathbf{x})$  を  $n$  次元の領域とし,  $\tilde{\mathcal{R}}(\mathbf{x})$  を軸凸領域となるような  $\mathcal{R}(\bar{\mathbf{x}})$  の部分集合とする . この時, 領域  $\bar{\mathbf{x}}$  も軸凸領域であるため, 以下の性質が成り立つ :

$$\mathcal{V}(\bar{\mathbf{x}}) \subseteq \tilde{\mathcal{R}}(\mathbf{x}) \Rightarrow \bar{\mathbf{x}} \subseteq \mathcal{R}(\mathbf{x}).$$

*Proof.* 凸集合の定義より自明である . □

Lemma 4 より, 提案手法を用いるため選択すべき実行可能領域の凸な部分集合が軸凸領域であれば良いことが分かる . つまり,  $\bar{\phi} \times \bar{\phi}'$  が軸凸領域であるので,  $\mathcal{R}^G(\phi, \phi')$  が

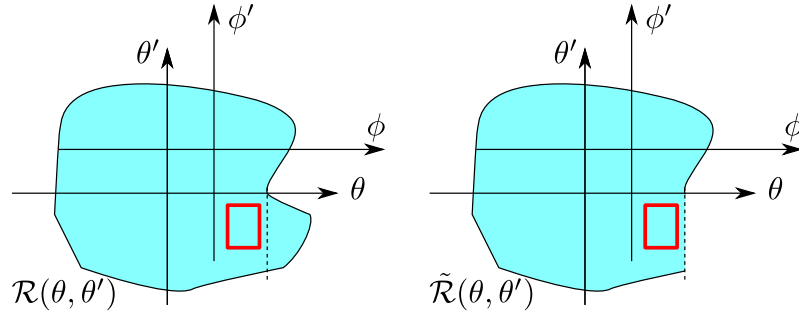


Fig. 5.8 Relation between a local coordinate system and a global coordinate system for orientation

軸凸領域であれば良いことになる．今回，ロボット姿勢のローカル座標系  $(\theta, \theta')$  からグローバル座標系  $(\phi, \phi')$  への変換は単なる軸の平行移動であるため， $\bar{\theta} \times \bar{\theta}'$  と  $\mathcal{R}(\theta, \theta')$  が軸凸領域であればそのまま利用できることを意味する．

以上より， $\mathcal{R}(\theta, \theta')$  が軸凸領域であれば良いことになる．満たさない場合には， $\mathcal{R}(\theta, \theta')$  に対して軸凸領域となる  $\tilde{\mathcal{R}}(\theta, \theta')$  を選択して用いる．Figure 5.8 では， $\mathcal{R}(\theta, \theta')$  も  $\tilde{\mathcal{R}}(\theta, \theta')$  も非凸な領域であるが， $\tilde{\mathcal{R}}(\theta, \theta')$  は軸凸領域である．

次に， $\mathcal{F}_o(q, q')$  が互いに素な複数の領域の和集合である状況について Fig. 5.9 を用いて述べる．Figure 5.9(a) は実行可能姿勢領域が単一の実行可能領域で成り立っている場合であり，Fig. 5.9(b) は実行可能姿勢領域が複数個 ( $M$  個) の実行可能姿勢領域で形成されている場合である．なお，Fig. 5.9(b) は  $M = 2$  であり，二つの実行可能姿勢領域の例である． $\bar{\theta} \times \bar{\theta}'$  が単一の実行可能姿勢領域内に含まれている場合は従来と同様に評価することが可能であるが，複数の実行可能姿勢領域にまたがっている場合には，端点のみで評価することが困難である．そのため，本研究ではこのような場合においては有向リンクを生成しないこととする．ここで，長方形  $\bar{\theta} \times \bar{\theta}'$  の頂点の集合を  $\mathcal{V}_o(\bar{\theta}, \bar{\theta}')$  とすると，以上の事柄から次の補題が得られる：

**Lemma 5.**  $\mathcal{V}_o(\bar{\theta}, \bar{\theta}')$  を  $\bar{\theta} \times \bar{\theta}'$  の頂点の集合とする． $\mathcal{F}_{o,i}(q, q')$  を  $i$  番目の実行可能姿勢領域とし， $\tilde{\mathcal{F}}_{o,i}(q, q')$  を  $i$  番目の実行可能姿勢領域の凸な部分集合であるとする：

$$\mathcal{V}_o(\bar{\theta}, \bar{\theta}') \subseteq \tilde{\mathcal{F}}_{o,i}(q, q'), \exists i \in M \Rightarrow \bar{\theta} \times \bar{\theta}' \subseteq \mathcal{F}_{o,i}(q, q'), \exists i \in M.$$

Lemma 5 より，各実行可能姿勢領域に対して  $\bar{\theta} \times \bar{\theta}'$  の頂点のみを評価すれば十分であることが分かる．

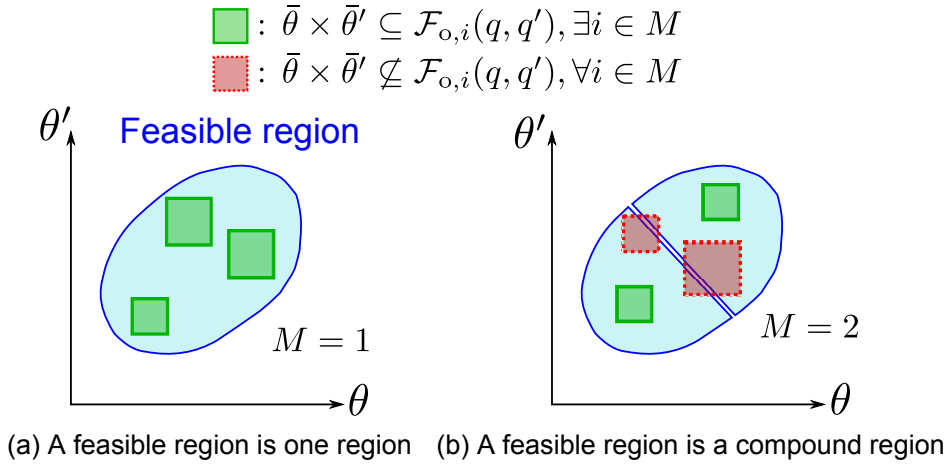


Fig. 5.9 Examples of feasible region

### 5.3.3 可変分解能姿勢空間分割

有向リンクの接続性は空間の分解能に依存することは 4.3.3 節で述べた。4.3.3 節では単一の実行可能領域についてしか考慮してなかったため、本節では複合実行可能姿勢領域について述べる。5.3.2 節で述べたように、直積集合がどれかの実行可能領域に包含された時にのみ有向リンクを生成する。提案手法では、可変分解能処理によって姿勢空間をバイナリツリーで表現する。ある  $q \in \mathcal{P}$  における姿勢空間の分割例を Fig. 5.10 に示す。ここで、空間の分割度合を分割レベルと定義する (Fig. 5.10(a))。

分割処理全体の流れを Fig. 5.11 に示す。初期状態では各  $q \in \mathcal{P}$  での姿勢空間が四等分されているとする (分割レベル 1)。次に、すべての組  $(q, q') \in \mathcal{L}$  について Algorithm 2 に示す `generate_orientation_links( $q, q'$ )` を実行する。なお、Algorithm 2 内の  $\mathcal{C}_o$  は姿勢領域を格納する集合である。この時、Algorithm 2 の line 3 – 12 で各領域が分割すべき領域であるかどうかの判断も行い、分割すべき領域であると判断された領域  $\bar{\theta} \in \mathcal{O}(q)$  と  $\bar{\theta}' \in \mathcal{O}(q')$  は  $\mathcal{C}_o$  に追加される。なお、実際の分割は後述する終了条件が満たされなかった場合に行う。分割すべきかどうかの判断基準は対象の姿勢領域の頂点の集合  $\mathcal{V}_o(\bar{\theta}, \bar{\theta}')$  と  $\mathcal{F}_o(q, q')$  の関係にもとづく：

$\mathcal{V}_o(\bar{\theta}, \bar{\theta}')$  の全体を包含する  $\mathcal{F}_{o,i}(q, q')$  が存在する：  $(q, \bar{\theta})$  から  $(q', \bar{\theta}')$  に有向リンクを生成するため、該当領域を分割対象としない。

$\mathcal{V}_o(\bar{\theta}, \bar{\theta}')$  の一部を包含する  $\mathcal{F}_{o,i}(q, q')$  が存在する： 分割を行うことである  $\mathcal{F}_{o,i}(q, q')$  に包含される領域を生成できるため、該当領域を分割対象とする。



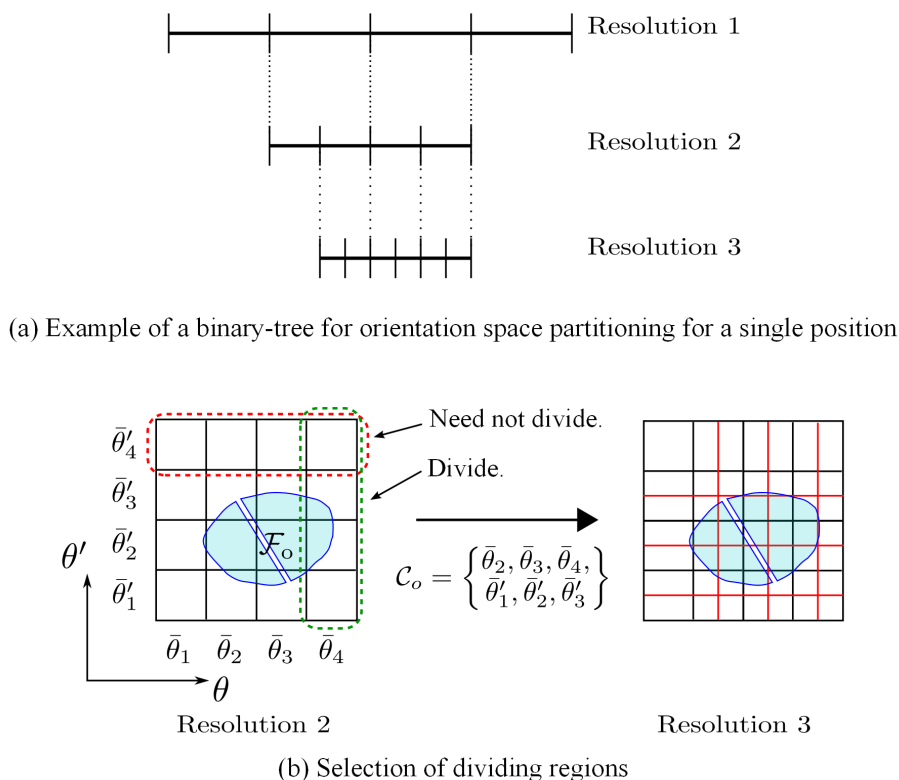


Fig. 5.10 An example of variable-resolution partitioning of orientation space

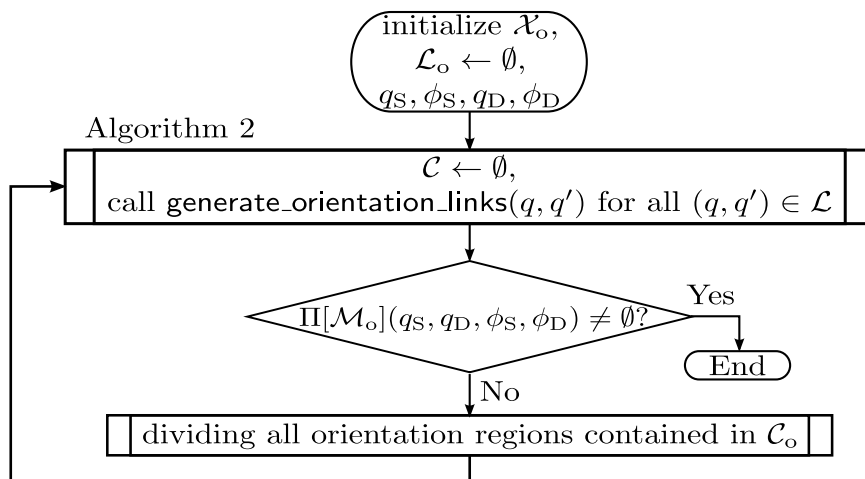


Fig. 5.11 Flowchart of overall procedure

全ての  $\mathcal{F}_{o,i}(q, q')$  が  $\mathcal{V}_o(\bar{\theta}, \bar{\theta}')$  を非包含： 分割によって生成された領域もどの  $\mathcal{F}_{o,i}(q, q')$  にも包含されないため，該当領域を分割対象としない。

---

**Algorithm 2** generate\_orientation\_links( $q, q'$ )
 

---

```

1: for all  $(\bar{\theta}, \bar{\theta}') \in \mathcal{O}(q) \times \mathcal{O}(q')$  do
2:    $\mathcal{V}_o(\bar{\theta}, \bar{\theta}') \leftarrow$  set of vertices of  $\bar{\theta} \times \bar{\theta}'$ 
3:   if  $\mathcal{V}_o(\bar{\theta}, \bar{\theta}') \subset \mathcal{F}_{o,i}(q, q')$ ,  $\exists i \in M$  then
4:     make directed link between  $(q, \bar{\theta})$  and  $(q', \bar{\theta}')$ 
5:   else if  $(\mathcal{V}_o(\bar{\theta}, \bar{\theta}') \cap \mathcal{F}_{o,i}(q, q')) \neq \emptyset$ ,  $\exists i \in M$  then
6:     if  $\bar{\theta}$  is a leaf of binary-tree, then
7:        $\mathcal{C}_o \leftarrow \mathcal{C}_o \cup \{\bar{\theta}\}$ 
8:     end if
9:     if  $\bar{\theta}'$  is a leaf of binary-tree, then
10:       $\mathcal{C}_o \leftarrow \mathcal{C}_o \cup \{\bar{\theta}'\}$ 
11:    end if
12:  end if
13: end for
    
```

---

この処理により，不必要な領域分割を行わなくて済む．Algorithm 2 終了後に P-roadmap 上の任意の二地点  $q, q'$  およびそれらでの姿勢  $\theta, \theta'$  に対応する経路が VOR 上に存在するかどうかを確認する．もしこの条件が成立すれば，姿勢空間の分割を終了する．そうでなければ，分割対象に選択した領域の分割を行う．領域の再分割を終えたら再度有向リンクの生成を行う．この一連の処理を終了条件が満たされるまで繰り返す．なお，第 4 章では  $\mathcal{C}_h$  と  $\mathcal{C}_t$  の二つ集合を用いたが，姿勢領域の分割においては今回は区別せずに  $\mathcal{C}_o$  のみを用いる．なぜならば，地図生成段階でロボットの姿勢を限定してはいけないからである．

ここで，分割対象領域の選択の様子を Fig. 5.10(b) に示す．Figure 5.10(b) では  $(\theta, \theta')$  空間について考えており，青の領域が  $\mathcal{F}_{o,i}(q, q')$  を表す．この例では  $\mathcal{V}_o(\bar{\theta}_3, \bar{\theta}'_2)$  の 4 端点がそれぞれ実行可能姿勢領域に含まれている．しかしながら，異なる実行可能領域に含まれているため， $(q, \bar{\theta}_3), (q', \bar{\theta}'_2)$  には有向リンクが生成されず， $\bar{\theta}_3$  と  $\bar{\theta}'_2$  が  $\mathcal{C}_o$  に登録される．この例では，最終的に  $\bar{\theta}_2, \bar{\theta}_3, \bar{\theta}_4, \bar{\theta}'_1, \bar{\theta}'_2$  および  $\bar{\theta}'_3$  が  $\mathcal{C}_o$  に登録される．

## 5.4 実行可能領域の導出

本章では，5.2 節で述べた車両型移動ロボットに関して実行可能姿勢領域  $\mathcal{F}_o(q, q')$  を導出する．5.2.1 節と Fig. 5.1(b) で定義した変数を用いて，ロボットの移動に関する制約条件を導出する．具体的には衝突回避および速度・加速度制約の 2 部に分かれる．それぞれの条件から導出した実行可能領域の積集合が想定する移動ロボットの実行可能領域となる．

## 5.4.1 障害物回避

本節では (5.13) 式について考える．(5.13a) 式は二地点  $q$  と  $q'$  の間を結ぶ直線分  $qq'$  の法線方向の制約条件に関する式であり，(5.13b) 式は直線分  $qq'$  の接線方向の制約条件に関する式である．(5.13a) 式および (5.13b) 式から求められる実行可能領域をそれぞれ  $\mathcal{F}_\circ^y(q, q')$  および  $\mathcal{F}_\circ^x(q, q')$  とする．

実行可能姿勢領域  $\mathcal{F}_\circ^y(q, q')$  最初に直線分  $qq'$  の法線方向に関する実行可能姿勢領域  $\mathcal{F}_\circ^y(q, q')$  を導出する． $\mathcal{F}_\circ^y(q, q')$  を導出するにあたり，5.2 節で述べたモデルを用いた場合の軌道において，直線分  $qq'$  から法線方向で最も遠ざかる時の距離を求める式を導出する．ここで，任意の地点  $p$  から任意の地点  $p' \in F$  軌道で進むことについて考える．ただし，直線分  $qq'$  上に存在するとは限定しない．また，直線分  $pp'$  に対する移動ロボットの  $p$  および  $p'$  での姿勢をそれぞれ  $\psi, \psi'$  とする．ここで，Fig. 5.12, 5.13 に示すように  $p_y = [x_{p_y}, y_{p_y}]^T$  を曲線  $pp'$  の直線分  $qq'$  から最も遠ざかる位置とし， $\theta_{pp'} (-\pi < \theta_{pp'} \leq \pi)$  を直線分  $qq'$  に対する直線分  $pp'$  の向きとする．なお，Fig. 5.12 と Fig. 5.13 は  $\theta_{pp'} + \theta_A$  によって場合分けされている．ただし， $\psi$  について対称性を有するため，( $\psi \geq 0, \psi' < 0$ ) についてのみ示す．

まず， $\theta_{pp'} + \theta_A \geq 0$  について述べる (Fig. 5.12)．この時，直線から最も遠ざかる位置は  $C'$  部分にあり，クロソイド曲線の性質より次式が得られる：

$$p_y = |v'| \sqrt{\left| \frac{2\theta_{p'}}{\alpha'} \right|} \begin{bmatrix} \cos \theta_{p'} & -\sin \theta_{p'} \\ \sin \theta_{p'} & \cos \theta_{p'} \end{bmatrix} \begin{bmatrix} -C \left( \sqrt{\left| \frac{2\theta_{p'}}{\pi} \right|} \right) \\ -S \left( \sqrt{\left| \frac{2\theta_{p'}}{\pi} \right|} \right) \end{bmatrix} + p_s,$$

$$\theta_{p'} = \psi' + \theta_{pp'},$$

$$p_s = \begin{cases} p_c & \text{if } \theta_A \geq 0, \\ p' & \text{otherwise.} \end{cases} \quad (5.16)$$

次に  $\theta_{pp'} + \theta_A < 0$  について述べる (Fig. 5.13)．この時，直線分  $qq'$  から最も遠ざかる

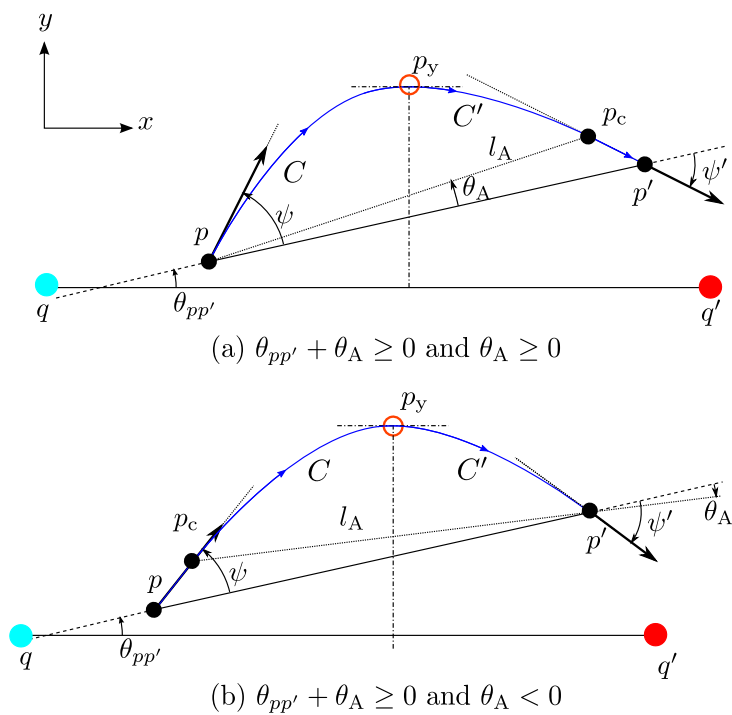


Fig. 5.12 A region about constraint condition of y-axis with  $\theta_{pp'} + \theta_A \geq 0$

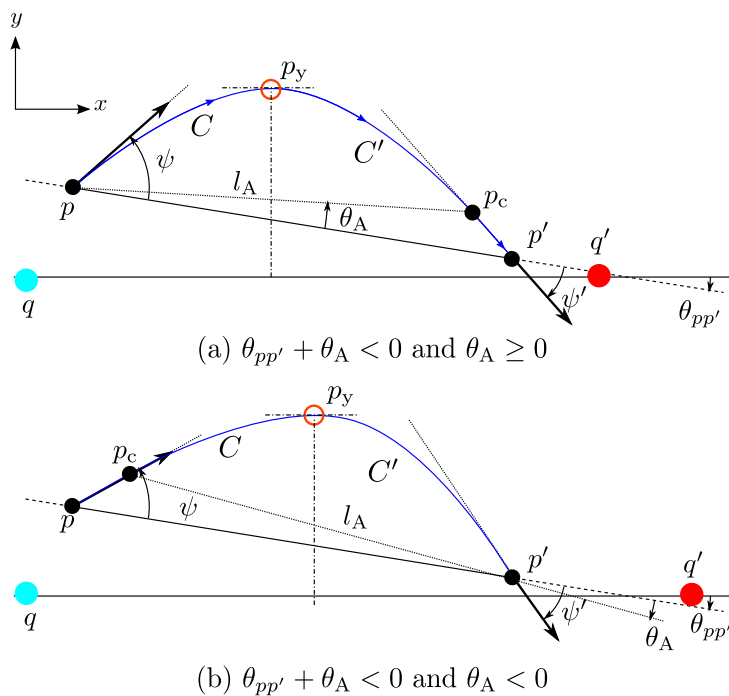


Fig. 5.13 A region about constraint condition of y-axis with  $\theta_{pp'} + \theta_A < 0$

位置は  $C$  部分にあり，クロソイド曲線の性質より次式が得られる：

$$p_y = |v| \sqrt{\left| \frac{2\theta_p}{\alpha} \right|} \begin{bmatrix} \cos \theta_p & -\sin \theta_p \\ \sin \theta_p & \cos \theta_p \end{bmatrix} \begin{bmatrix} C \left( \sqrt{\left| \frac{2\theta_p}{\pi} \right|} \right) \\ -S \left( \sqrt{\left| \frac{2\theta_p}{\pi} \right|} \right) \end{bmatrix} + p_s,$$

$$\theta_p = \psi + \theta_{pp'},$$

$$p_s = \begin{cases} p & \text{if } \theta_A \geq 0, \\ p_c & \text{otherwise.} \end{cases} \quad (5.17)$$

(5.16) 式及び (5.17) 式を組み合わせることで， $q$  から  $q'$  に移動する際に直線分  $qq'$  から法線方向で最も遠ざかる位置を計算できる．この時，一組の  $d(q, q')$  と  $w(q, q')$  に対して

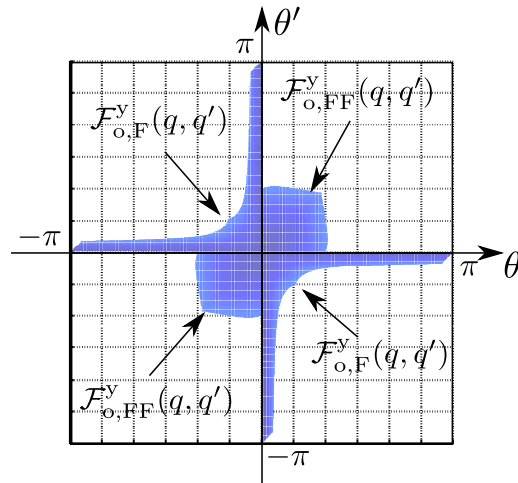
$$0 \geq y_{\max} - w(q, q'),$$

$$y_{\max} = \max_{t \in [0, T]} |y(t)| = \max_{i \in C_c} y_{p_y, i} \quad (5.18)$$

を満たす領域として Fig. 5.14 に示す実行可能姿勢領域  $\mathcal{F}_o^y(q, q')$  を得る．なお， $C_c$  は軌道  $qq'$  内に存在する F 軌道の数である．Figure 5.14 で色がついた部分が実行可能領域である． $w(q, q')$  が小さいと Fig. 5.14(a) のように領域が小さく， $w(q, q')$  が大きいほど Fig. 5.14(b) のように領域が大きくなる．ただし，解析的に求めることが困難であるため，数値計算を用いて求めている．なお，計算の詳細は付録を参照されたい．実行可能領域  $\mathcal{F}_o^y(q, q')$  は二つの FF 軌道の実行可能姿勢領域  $\mathcal{F}_{o,FF}^y(q, q')$  と二つの F 軌道の実行可能姿勢領域  $\mathcal{F}_{o,F}^y(q, q')$  が組み合わさった領域である．ただし，それぞれの実行可能領域  $\mathcal{F}_{o,F}^y(q, q')$  と  $\mathcal{F}_{o,FF}^y(q, q')$  は互いに素であるため， $(\bar{\theta}, \bar{\theta}')$  が実行可能領域に包含されているかどうかの判定は，単独の実行可能姿勢領域に包含されているかどうかで行う (Fig. 5.14(b))．そして， $(\bar{\theta}, \bar{\theta}')$  が複数の実行可能姿勢領域にまたがるときは，有向リンクは生成しないものとする．

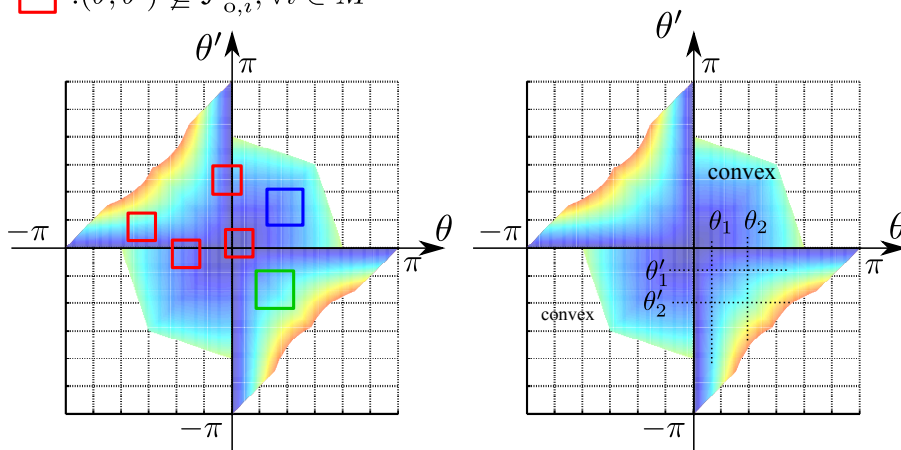
Figure 5.14 より  $\mathcal{F}_{o,FF}^y(q, q')$  は凸な集合であるが， $\mathcal{F}_{o,F}^y(q, q')$  が非凸な集合であることが分かる．第 4 章では非凸な集合の場合には，実行可能領域の凸な部分領域を導入していたが，今回は 5.3.2 節に示されている条件を満たしているため， $\mathcal{F}_F^y(q, q')$  の凸な部分領域を導入する必要はない．

実行可能姿勢領域  $\mathcal{F}_o^x(q, q')$  次に直線分  $qq'$  の接線方向に関する実行可能姿勢領域  $\mathcal{F}_o^x(q, q')$  を求める．直線分の接線方向の制約条件である (5.13b) 式について考える．前項と同様に 5.2 節で述べたモデルを用いた時の軌道において，直線分  $qq'$  から接線方向に最も遠ざかる時の距離を求める式を導出する．ここで，任意の地点  $p$  から任意の地点  $p'$  へ F 軌道で進むことについて考える．ただし，直線分  $qq'$  上に存在するとは限定しない．



(a) A combination of 4 feasible regions

- :  $(\bar{\theta}, \bar{\theta}') \subseteq \mathcal{F}_{o,FF}^y$
- :  $(\bar{\theta}, \bar{\theta}') \subseteq \mathcal{F}_{o,F}^y$
- :  $(\bar{\theta}, \bar{\theta}') \not\subseteq \mathcal{F}_{o,i}^y, \forall i \in M$



(b) Feasible regions and direct products

(c) Feasible region for F trajectory and sections

Fig. 5.14 A region about constraint condition of y-axis

また、直線分  $pp'$  に対する移動ロボットの  $p$  および  $p'$  での姿勢をそれぞれ  $\psi, \psi'$  とする。ここで、Fig. 5.15, 5.16 に示すように  $p_x = [x_{p_x}, y_{p_x}]^T$  を曲線  $pp'$  上の  $x$  成分の値が最小となる位置とし、 $p'_x = [x'_{p'_x}, y'_{p'_x}]^T$  を曲線  $pp'$  上の  $x$  成分の値が最大となる位置とし、 $\theta_{pp'}$  ( $-\pi < \theta_{pp'} \leq \pi$ ) を直線分  $qq'$  に対する直線分  $pp'$  の向きとする。なお、Fig. 5.15 と Fig. 5.16 は  $\theta_{pp'} + \theta_A$  によって場合分けされている。ただし、 $\psi$  について対称性を有するため、 $(\psi \geq 0, \psi' \leq 0)$  についてのみ示す。

まず、 $\theta_{pp'} + \theta_A \geq 0$  について述べる (Fig. 5.15)。ただし、 $\psi + \theta_{pp'} \geq \frac{\pi}{2}$  である。この時、直線分  $qq'$  の接線方向において、 $q$  から最も遠ざかる位置は  $C$  部分にあり、クロソイ

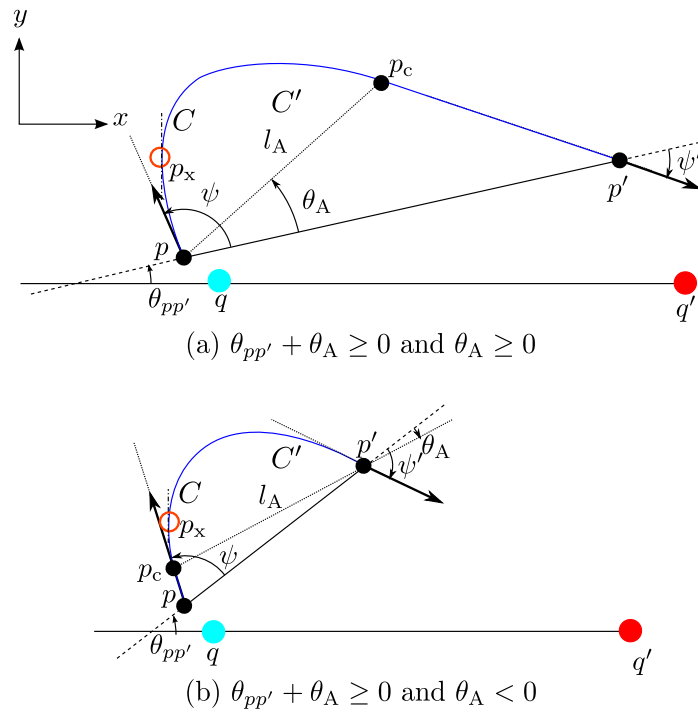


Fig. 5.15 A region about constraint condition of x-axis with  $\theta_{pp'} + \theta_A \geq 0$

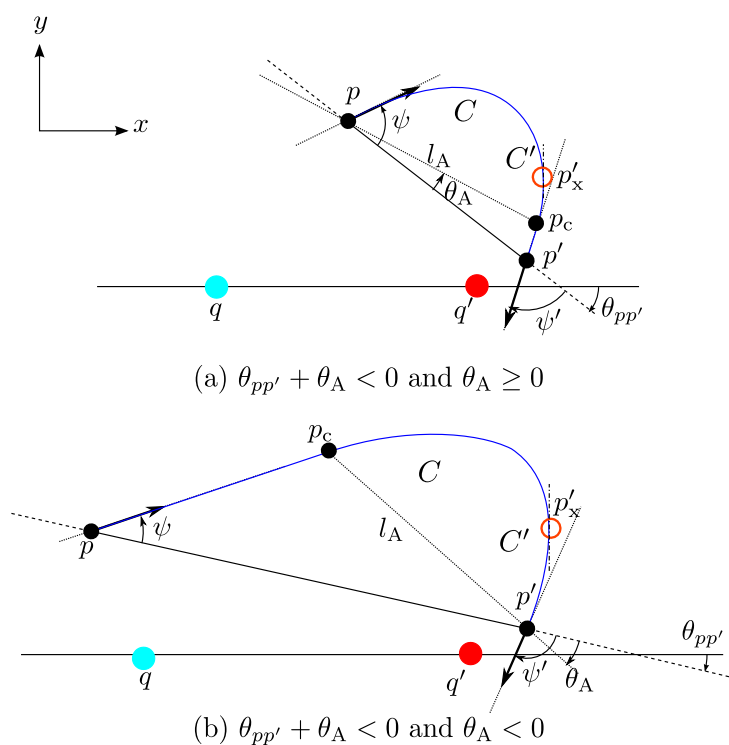
ド曲線の性質より次式のように計算できる：

$$p_x = |v| \sqrt{\left| \frac{2\theta_p - \pi}{\alpha} \right|} \begin{bmatrix} \cos \theta_p & -\sin \theta_p \\ \sin \theta_p & \cos \theta_p \end{bmatrix} \begin{bmatrix} C \left( \sqrt{\left| \frac{2\theta_p - \pi}{\pi} \right|} \right) \\ -S \left( \sqrt{\left| \frac{2\theta_p - \pi}{\pi} \right|} \right) \end{bmatrix} + p_s,$$

$$\theta_p = \psi + \theta_{pp'},$$

$$p_s = \begin{cases} p & \text{if } \theta_A \geq 0, \\ p_c & \text{otherwise.} \end{cases} \quad (5.19)$$

次に  $\theta_{pp'} + \theta_A < 0$  について述べる (Fig. 5.16) . この時, 直線分  $qq'$  の接線方向において  $q'$  から最も遠ざかる位置は  $C'$  部分にあり, クロソイド曲線の性質より次式が得ら


 Fig. 5.16 A region about constraint condition of x-axis with  $\theta_{pp'} + \theta_A < 0$ 

れる :

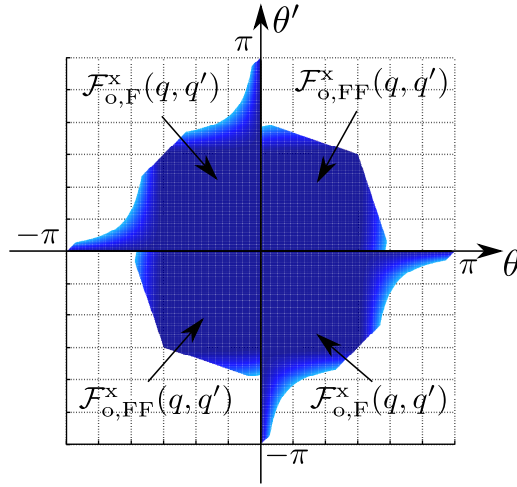
$$p'_x = |v'| \sqrt{\left| \frac{2\theta'_p + \pi}{\alpha'} \right|} \begin{bmatrix} \cos \theta'_p & -\sin \theta'_p \\ \sin \theta'_p & \cos \theta'_p \end{bmatrix} \begin{bmatrix} -C \left( \sqrt{\left| \frac{2\theta'_p + \pi}{\pi} \right|} \right) \\ -S \left( \sqrt{\left| \frac{2\theta'_p + \pi}{\pi} \right|} \right) \end{bmatrix} + p_s,$$

$$\theta'_p = \psi' + \theta_{pp'},$$

$$p_s = \begin{cases} p_c & \text{if } \theta_A \geq 0, \\ p & \text{otherwise.} \end{cases} \quad (5.20)$$

(5.19) 式及び (5.20) 式を組み合わせることで、 $q$  から  $q'$  に移動する際に直線分  $qq'$  から接線方向で最も遠ざかる位置を計算できる。この時、一組の  $d(q, q')$ 、 $\epsilon_1(q, q')$  および





(a) A combination of 4 feasible regions for x-axis

Fig. 5.17 A region about constraint condition of x-axis

$\epsilon_2(q, q')$  に対して

$$0 \leq x_{\min} - q^x + \epsilon_1(q, q'),$$

$$x_{\min} = \min_{t \in [0, T]} x(t) = \begin{cases} x_{p_x} & \text{if } \frac{\pi}{2} < |\psi + \theta_{pp'}| \leq \pi, \\ q^x & \text{if } 0 \leq |\psi + \theta_{pp'}| \leq \frac{\pi}{2}, \end{cases} \quad (5.21)$$

$$0 \geq x_{\max} - q'^x - \epsilon_2(q, q'),$$

$$x_{\max} = \max_{t \in [0, T]} x(t) = \begin{cases} x'_{p'_x} & \text{if } \frac{\pi}{2} < |\psi' + \theta_{pp'}| \leq \pi, \\ q'^x & \text{if } 0 \leq |\psi' + \theta_{pp'}| \leq \frac{\pi}{2}, \end{cases} \quad (5.22)$$

を満たす領域として Fig. 5.17 に示す実行可能姿勢領域  $\mathcal{F}_o^x(q, q')$  を得る．ただし，解析的に求めることが困難であるため，数値計算を用いて求めている．Figure 5.17 では，色がついた領域が実行可能領域である．なお，FF 軌道の場合は (5.21) 式の  $x_{p_x}$  には前半の F 軌道から得られる  $x_{p_x}$  を用い，(5.22) 式の  $x'_{p'_x}$  は後半の F 軌道から得られる  $x'_{p'_x}$  を用いる．実行可能姿勢領域  $\mathcal{F}_o^x(q, q')$  は二つの FF 軌道の実行可能姿勢領域  $\mathcal{F}_{o,FF}^x(q, q')$  と二つの F 軌道の実行可能姿勢領域  $\mathcal{F}_{o,F}^x(q, q')$  が組み合わさった領域である．ただし，それぞれの実行可能領域は互いに素であるため，前項と同様に  $(\bar{\theta}, \bar{\theta}')$  が実行可能姿勢領域に含まれているかどうかの判定は，単独の実行可能領域に含まれているかどうかで行う．そして， $(\bar{\theta}, \bar{\theta}')$  が複数の実行可能姿勢領域にまたがるときは，有向リンクは生成しないとする．

## 5.4.2 速度及び加速度制約

本節では線速度，旋回角速度および旋回角加速度に関する制約条件である (5.14) 式について考える．5.2 節で述べたモデルを用いた場合，線速度と旋回角加速度は一定であり，旋回角速度が最大となるのは  $C$  と  $C'$  が切り替わる地点である (Fig. 5.4 の  $q_B$ )．

提案軌道を用いた場合， $q$  から  $q_B$  の走行距離は  $\theta, \theta', d(q, q')$  を用いて次式で計算できる：

$$\begin{aligned}
 c_v &= vT_B = \frac{\tan \theta' (c(\theta, \theta') - \tan \theta) d(q, q')}{(\tan \theta - \tan \theta') (a(\theta, \theta') - b(\theta, \theta')c(\theta, \theta'))}, \\
 a(\theta, \theta') &= f(\theta, \theta') \sin \theta - g(\theta, \theta') \cos \theta, \\
 b(\theta, \theta') &= f(\theta, \theta') \cos \theta + g(\theta, \theta') \sin \theta, \\
 c(\theta, \theta') &= \tan \left( \frac{\theta + \theta' - \pi}{2} \right), \\
 f(\theta, \theta') &= C \left( \frac{1}{2} \alpha T_B^2 \right) = \sum_{k=0}^{\infty} \frac{1}{(2k)!(4k+1)} \left| \frac{\theta - \theta'}{2} \right|^{2k}, \\
 g(\theta, \theta') &= S \left( \frac{1}{2} \alpha T_B^2 \right) = \sum_{k=0}^{\infty} \frac{1}{(2k+1)!(4k+3)} \left| \frac{\theta - \theta'}{2} \right|^{2k+1}. \quad (5.23)
 \end{aligned}$$

$q_B$  での姿勢は次式で計算できる：

$$c_a = \frac{1}{2} \alpha T_B^2 = \frac{1}{2} |\theta - \theta'|. \quad (5.24)$$

この二式を連立して解くことで次式が得られる：

$$(\alpha, v, t) = \left( \frac{2c_a}{\tau^2}, \frac{c_v}{\tau}, \tau \right), \tau \in \mathbb{R}^+. \quad (5.25)$$

$\tau$  の条件については (5.14) 式から得られる：

$$\tau \geq \max \left( \frac{c_v}{V_{\max}}, \frac{2c_a}{W_{\max}}, \sqrt{\frac{2c_a}{A_{\max}}} \right). \quad (5.26)$$

(5.25) 式および (5.26) 式より任意の  $(\theta, \theta')$  から得られる軌道に対して  $(\alpha, v, t)$  が存在するため，これらの制約条件から得られる  $\mathcal{F}_o(q, q')$  は Fig. 5.3 で示された領域全体である．また，ロボットの性能が実行可能姿勢領域に影響しないと言える．なお， $(\theta, \theta')$  によって  $\tau$  の範囲の最小値が異なるため，移動時間に影響を与える．

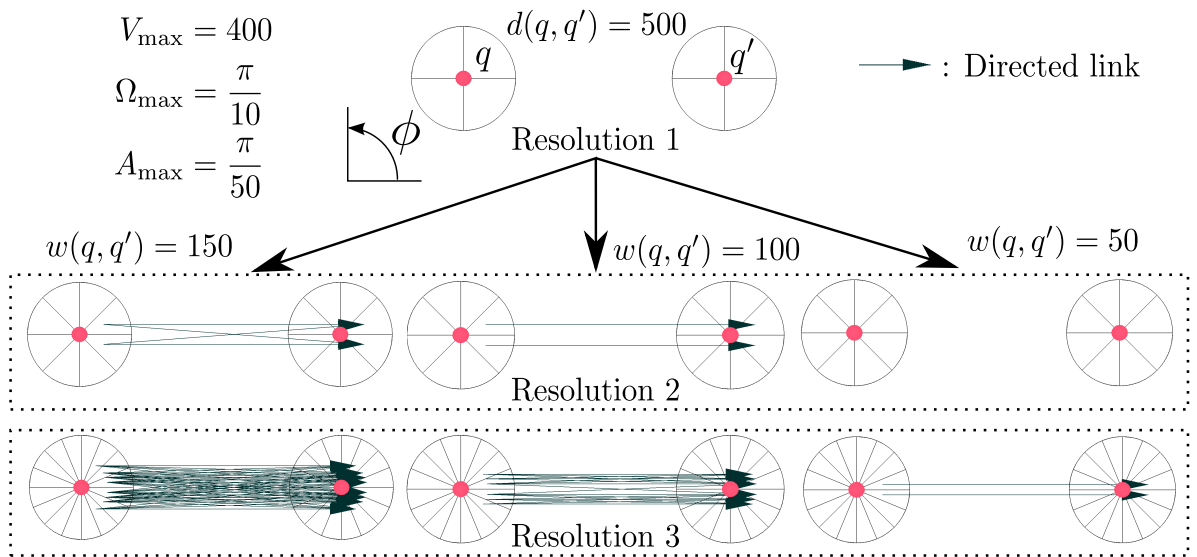


Fig. 5.18 An example of dividing orientation space and directed link generation

## 5.5 数値例

本節においていくつかの例題に対する数値シミュレーション結果を示す．最初に二地点間の移動に注目し，どのように姿勢空間の分割とリンク生成が行われるかを Fig. 5.18 に示す．なお，二地点間の距離を  $d(q, q') = 500$  とし， $\epsilon_1(q, q') = \epsilon_2(q, q') = 5$  とし，経路幅  $w(q, q')$  は 150, 100 および 50 の三通りを考える．また，移動ロボットの最大速度を  $V_{\max} = 400$  とし，最大旋回角速度を  $\Omega_{\max} = \frac{\pi}{10}$  とし，最大旋回角加速度を  $A_{\max} = \frac{\pi}{50}$  とする．図中の矢印は有向リンクを意味する．

分割レベルが 1 の時，どの場合もまだ有向リンクが生成されていない．分割レベルが 2 になると変化が現れる． $w(q, q') = 50$  の場合はまだ有向リンクが生成されていないのに対して， $w(q, q') = 100, 150$  では有向リンクが生成されている．後者についてさらに比較を行うと，150 の場合の方が多く生成されている．100 は FF 軌道を生成する領域間に有向リンクが生成され，150 では F 軌道を生成する領域間にも有向リンクが生成されている．これは，同じ二地点間に対して，F 軌道の方が FF 軌道よりも横方向に遠くまで移動する軌道の性質を反映していると言える．分割レベルが 3 になると有向リンクを生成している領域の差が顕著である． $w(q, q') = 150, 100, 50$  の順に有向リンクが生成される姿勢限界が狭くなっていることが分かる．これは経路幅が広いほど選択可能な姿勢の範囲が広いことを意味し，対照的に経路幅が狭いほど選択可能な速度の範囲が狭いことを意味する．なお，姿勢領域が一様に分割されているのは，姿勢の直積集合を考えた時に複数の実

行可能姿勢領域にまたがるために生じている。

次に事前に与えられた P-roadmap (Fig. 5.19(a)) に対して VOR の生成を行う (Fig. 5.19(b)-(f))。ただし、今回はロボットの初期設定を  $(x, y, \phi) = (0, 0, 0)$  とし、目標位置を  $(x, y) = (500, 0)$  で、目標姿勢は任意とする。Figure 5.19(a) では、点が P-roadmap の各ノードを表し、直線分が P-roadmap の各リンクを表す。Figure 5.19(b)-(f) では、各点周りの扇形が各点での分割された姿勢領域を表している。最終的な姿勢空間の分割レベルは5である。なお、表現の都合上、有向リンクは一方向のみを描画している。移動ロボットの最大速度を  $V_{\max} = 400$  とし、最大旋回角速度を  $\Omega_{\max} = \frac{\pi}{10}$  とし、最大旋回角加速度を  $A_{\max} = \frac{\pi}{50}$  とする。図より姿勢空間を分割していくにつれて有向リンクが増えている。Figure 5.19(c) では  $(0, 500)$  から  $(240, 400)$  への有向リンク生成において、赤点線の矢印が表す領域間は複数の実行可能姿勢領域をまたぐため、有向リンクが生成されていない。この影響により、全体が一様の分解能で分割されている。

実際に生成された VOR (Fig. 5.19(f)) を用いて経路計画を行う。経路計画には 2.2.2 節で述べたダイクストラ法を用いる。地図生成時と同様の設定にロボットの初期位置と姿勢を  $(x, y, \phi) = (0, 0, 0)$  とし、目標位置を  $(x, y) = (500, 0)$  とし、目標姿勢を任意とする (Fig. 5.20)。Figure 5.20(a) で緑色の領域が選択された姿勢領域である。Figure 5.20(b) が選択された姿勢領域を用いた軌道である。各経由地点での姿勢は領域の中央値を用いる。VOR では姿勢を選択することで (5.26) 式を用いて移動時間を計算できるため、移動時間を探索コストとしてグラフ探索で経路計画を行える。

次に、生成された地図を用いて初期姿勢のみを変更した設定  $(x, y, \phi) = (0, 0, \frac{\pi}{2})$  で経路計画を行う (Fig. 5.21)。初期姿勢を変更しているため、Fig. 5.20 の場合とは選択されている姿勢領域が異なっている。これにより、ある特定の姿勢に対して地図を生成しているが、別の姿勢に対しても利用できることが言える。

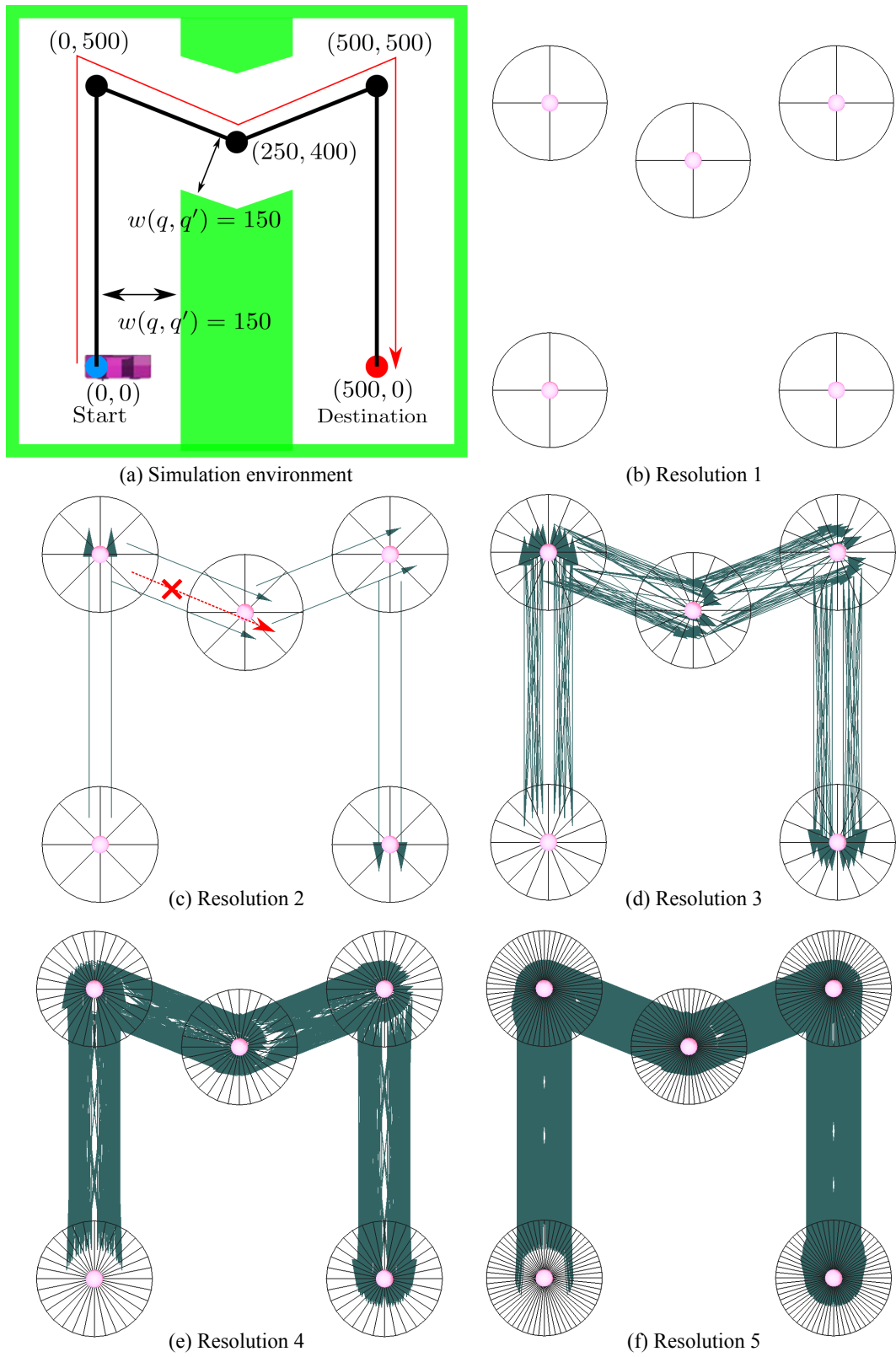


Fig. 5.19 An example of dividing orientation space and directed link generation

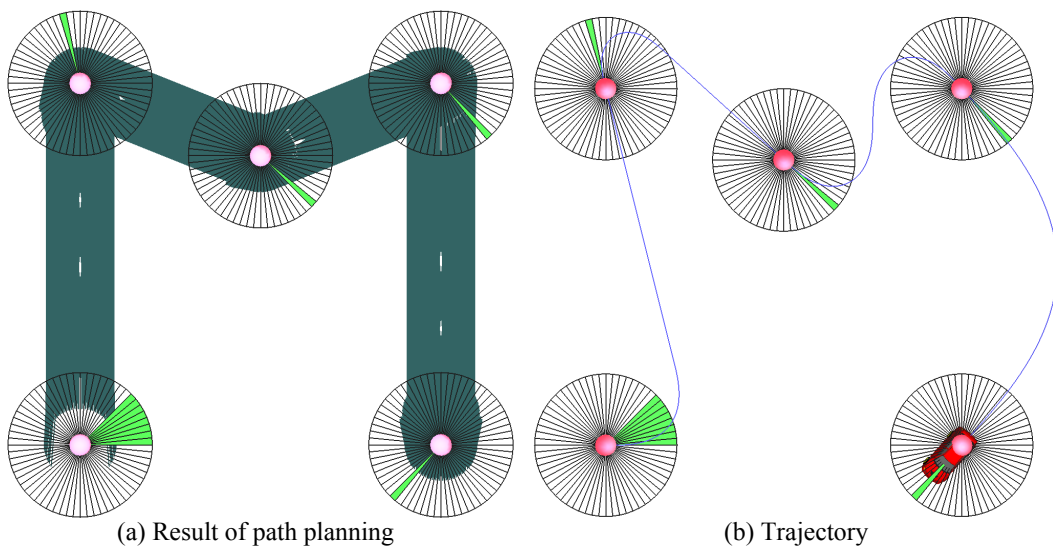


Fig. 5.20 Path planning with  $(0, 0, 0)$  to  $(500, 0)$  using VOR (Fig. 5.19(f))

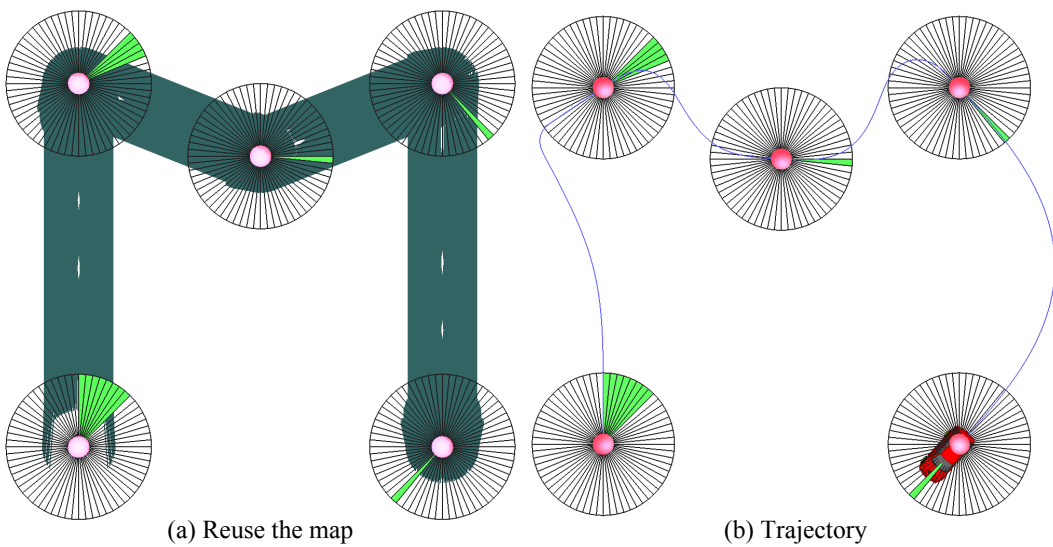


Fig. 5.21 Path planning with  $(0, 0, \frac{\pi}{2})$  to  $(500, 0)$  using VOR (Fig. 5.19(f))

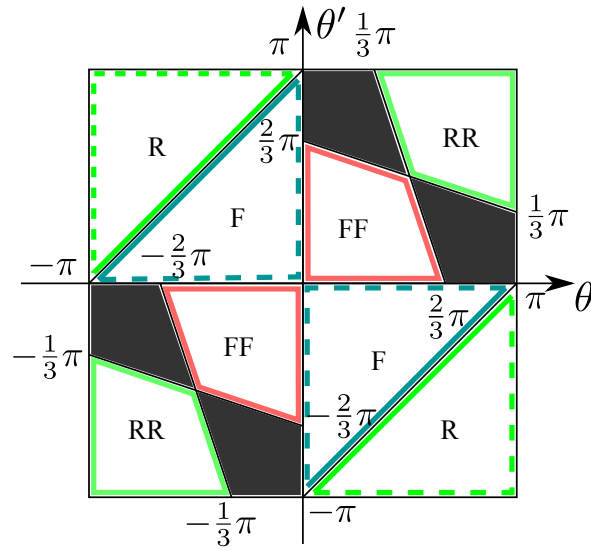


Fig. 5.22 Trajectory map with reverse trajectory based on  $\theta$  and  $\theta'$

## 5.6 自動駐車システム

本節では、提案する可変分解能姿勢マップを用いて構築した自動車の自動駐車システムについて示す．自動車の駐車では前進のみではなく、後進も行える必要がある．そこで、Fig. 5.22 に示す後進も可能なモデルを構築する．Figure 5.22 において、R と RR はそれぞれ Reverse Trajectory (R 軌道) と Reverse-Reverse Trajectory (RR 軌道) である．R 軌道は F 軌道と同様に生成でき、RR 軌道は FF 軌道と同様に生成できる．

本節でシミュレーションと実機実験を行う環境を表す P-roadmap を Fig. 5.23 に示す．自動車の寸法は実際に実機実験で用いる自動車の寸法である．自動車の最大速度を  $V_{\max} = 500[\text{mm/s}]$  であり、最大旋回角速度を  $\Omega_{\max} = \frac{\pi}{16}[\text{/s}]$  であり、最大旋回角加速度を  $A_{\max} = \frac{\pi}{32}[\text{/s}^2]$  である．さらに自動車の最小旋回半径は  $R_{\min} = 3.2[\text{m}]$  である．なお、最小旋回半径に関する実行可能領域の導出は付録 A.2 節を参照されたい．Figure 5.23 中の  $q_i, i \in [0, 20]$  は自動車能够通过できる位置であり、 $q_i^f, i \in [1, 3]$  は前向き駐車位置であり、 $q_i^r, i \in [1, 3]$  は後ろ向き駐車位置である．Figure 5.23 の P-roadmap のリンクが障害物と交差しているが、VOR の有向リンク生成時に厳密に衝突判定を行えば問題はない．むしろ、駐車というタスクを考慮した場合、障害物と交差しないリンクしか存在しない P-roadmap による駐車は困難である．Figure 5.23 に示す環境に対して生成した VOR を Fig. 5.24 に示す．なお、表現上の都合により有向リンクを省略する． $q_i^f, q_i^r$  は駐車位置で



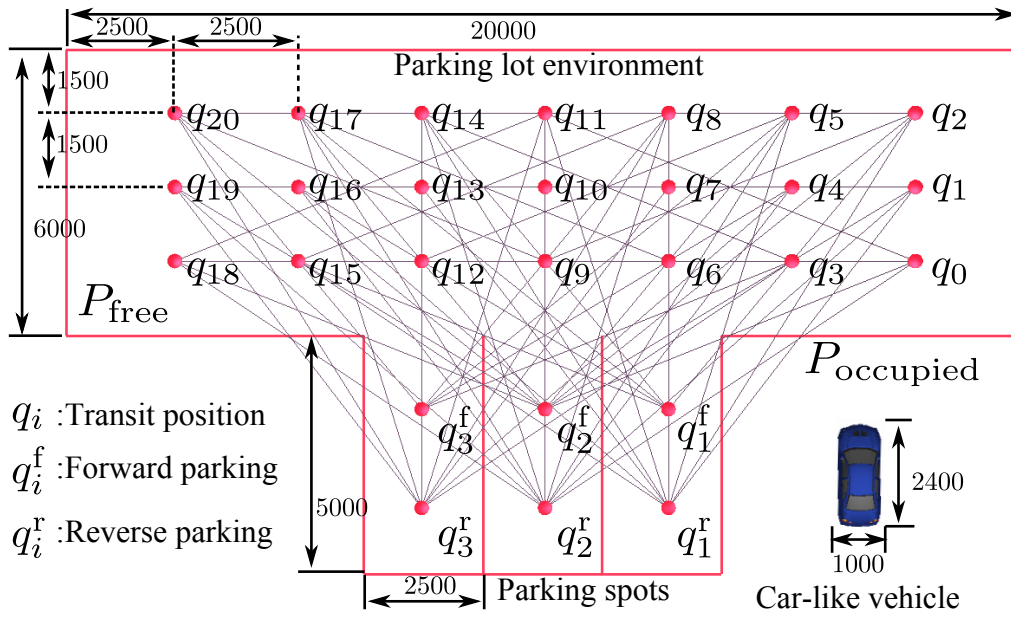


Fig. 5.23 Parking lot environment with a roadmap

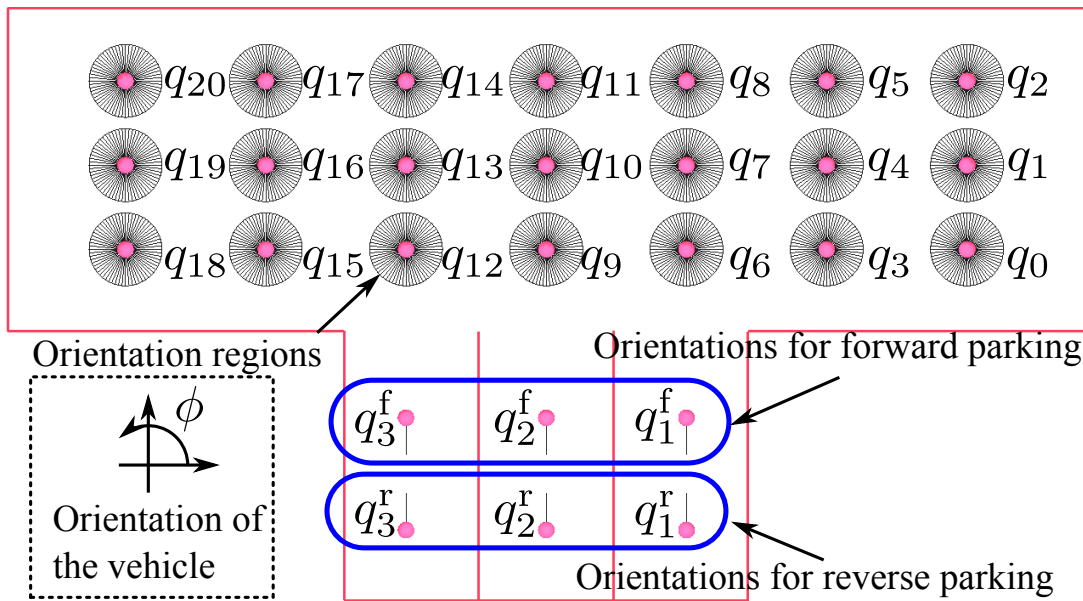


Fig. 5.24 A Variable-resolution Orientation Roadmap for Fig. 5.23

あるため、姿勢領域を一意に定めた地図を生成している。ただし、この地図生成では終了条件を分割レベルが5になった時としている。

生成された VOR に対して、二種類の設定で経路計画を行った。後進による駐車結果と前進による駐車結果をそれぞれ Fig. 5.25 と Fig. 5.26 に示す。これらの図では、



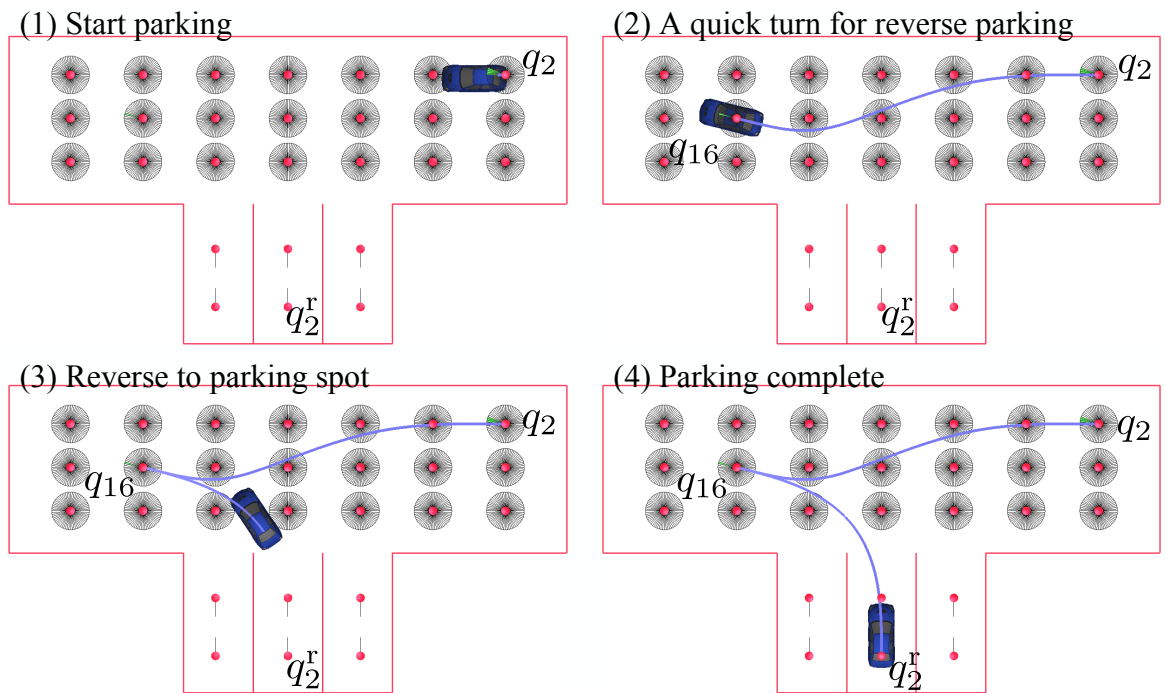


Fig. 5.25 A parking trajectory by reverse parking from  $(q_2, \pi)$  to  $(q_2^r, \frac{\pi}{2})$

緑色の領域が選択された姿勢領域であり，青色の曲線が移動ロボットの軌道である．Figure 5.25 は  $(q_2, \pi)$  から  $(q_2^r, \frac{1}{2}\pi)$  への後進による駐車である．経路計画段階では  $\Pi[\mathcal{M}_o]((q_2, \pi), (q_2^r, \frac{1}{2}\pi))$  の中で最も移動時間が小さい経路を選択している．目標位置・姿勢が後進による駐車でしか達成できないため， $q_{16}$  で切り替えを行う経路が自動的に計画されている．Figure 5.26 は  $(q_0, \pi)$  から  $(q_2^f, -\frac{1}{2}\pi)$  への前進による駐車である．経路計画段階では  $\Pi[\mathcal{M}_o]((q_0, \pi), (q_2^f, -\frac{1}{2}\pi))$  の中で最も移動時間が小さい経路を選択している．前進による駐車であるが， $(q_0, \pi)$  から直接駐車できないため， $q_{17}$  と  $q_8$  で切り替えを行う経路が計画されている．このように，グラフ探索を行うのみで，自動車は追従できる軌道を計画し，なおかつ切り替えしも自動的に計画される．

最後に実車実験について述べる．実車実験の様子を Fig. 5.27 に示す．実験に用いた電気自動車は自動的に前進，後進および停止できる．ただし，前進と後進の切り替えは手動で行わなくてはならない．この実験の環境は Fig. 5.23 と同じである．初期位置・姿勢は  $(q_3, \pi)$  であり，目標位置・姿勢は  $(q_2^r, \frac{\pi}{2})$  である．この設定に対して経路計画を行い，自動車に計画された軌道の追従を行った．自動車は  $q_3$  から  $q_{14}$  まで前進し，後進で  $q_2^r$  に駐車した．これにより，提案する自動駐車システムは実車においても利用可能である．

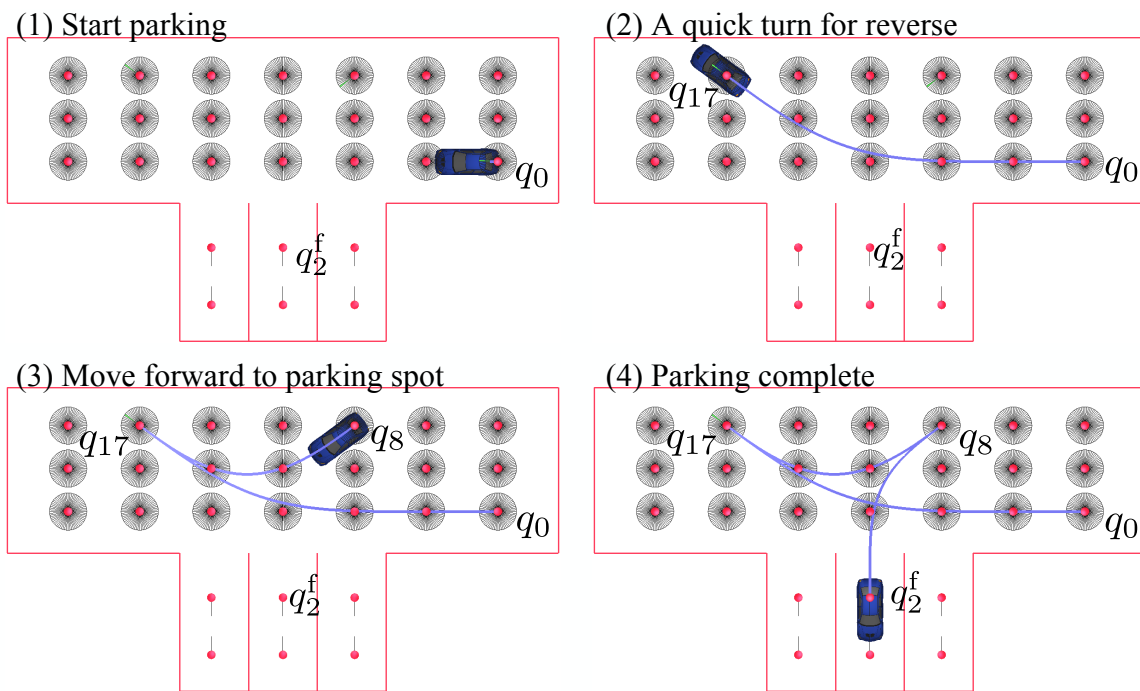


Fig. 5.26 A parking trajectory by forward parking from  $(q_0, \pi)$  to  $(q_0^f, -\frac{\pi}{2})$

## 5.7 まとめ

本章では、移動ロボットの位置と姿勢から成る配位空間上でロボットの動特性や種々の制約条件を考慮したロードマップ表現とその構築方法を提案し、車両型移動ロボットへの適用例を示した。従来では、移動ロボットの非ホロノミック特性を考慮せずに経路計画を行っていたために、非ホロノミック特性を考慮した軌道計画をオンラインで行う必要がある。それに対して、提案手法を用いることで、車両型移動ロボットの非ホロノミック特性をマップ生成時に計算してマップの姿勢情報として埋め込み、グラフ構造のロードマップで表現しているため、既存のグラフ探索手法を用いるだけで、移動ロボットの制約条件を満足する経路を計画できる。

最後に提案手法を用いた自動駐車システムを開発した。従来の自動駐車システムは自動車が観測を行いながら自律的に駐車スペースに移動するものであるが、提案手法を用いた自動駐車システムでは、事前に駐車場のマップを用意することで、計画にグラフ探索を適用することで、切り替えしを含んだ自動駐車ができている。

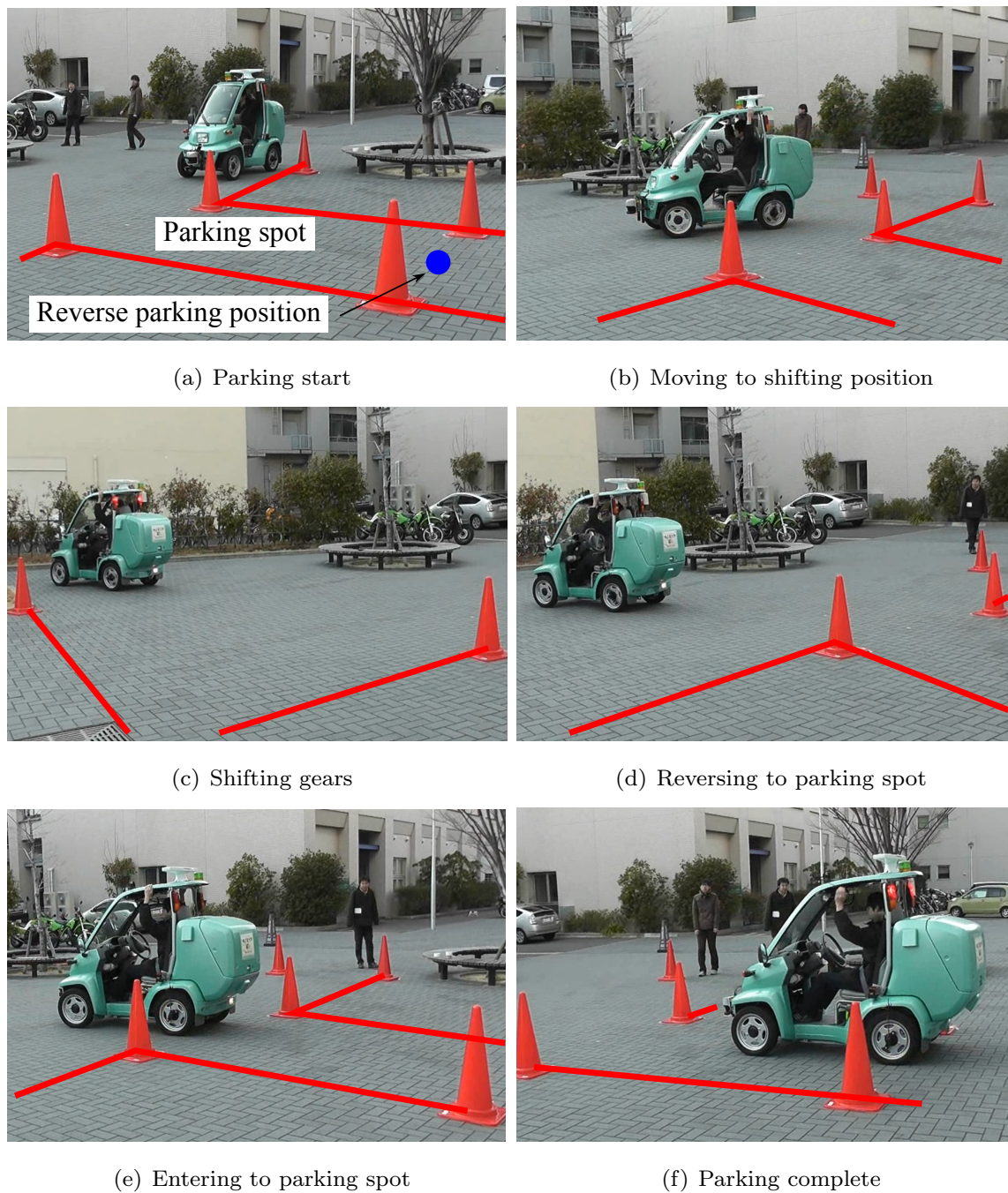


Fig. 5.27 An experiment result using an electric car



## 第 6 章

# 結論

本章では，本研究の成果と今後の課題について述べる．

### 6.1 本研究の成果

本論文での主な成果を以下に示す．

第 1 章および第 2 章では，移動ロボットのロードマップを用いた経路計画問題およびロードマップの分解能を可変にする意義について述べ，本研究の目的と本論文の構成について述べた．

第 3 章では，未知環境下での自律的な地図生成手法と実時間での経路計画手法を提案した．提案手法は，単純な反復計算による空間分割と，ルールベースのグラフ地図生成手法から成り立っているため，実時間処理に適している．また，経路計画には実時間処理が可能でなおかつ実装が容易な RTA\* 探索を用いた．これらの理由により，未知環境下での目的地到達タスクにおいて，観測情報を基に逐次的に地図を更新し，更新された地図を用いて経路の再計画が可能である．最後に，提案手法を数値シミュレーションで検証し，提案手法におけるパラメータの設定が妥当であることを示した．また，全方向移動可能なロボットを用いた実機実験の結果を示し，提案手法の有効性を示した．

第 4 章では，ホロノミック拘束を有する移動ロボットとして全方向移動ロボットを対象とし，全方向移動ロボットの運動特性に基づく可変分解能速度マップ生成手法を提案した．全方向移動ロボットのモデルとして，二地点の位置と二地点での速度から軌道が一意に求まる制御モデルを用いた．このモデルを用いたことで，速度領域から速度領域への実行可能性を定義でき，位置と速度領域の二項組をノードとした有向グラフ型のマップを生



成できることを示した。また、提案するロードマップを用いることで、経路計画段階において実際の制御入力計画も可能である。最後に、提案手法を用いた数値シミュレーションの結果を示し、分解能一定の速度マップと比較を行うことで提案手法の有効性を示した。

第5章では、非ホロノミック拘束を有する移動ロボットとして車両型移動ロボットを対象とし、車両型移動ロボットの運動特性に基づく可変分解能姿勢マップ生成手法を提案した。車両型移動ロボットは非ホロノミック拘束を有しているため、全方向移動ロボットのように速度のみを考えることが不可能である。そこで、移動ロボットの軌道をクロソイド曲線と線分を組み合わせた軌道生成手法を用いた。この軌道生成手法を用いたことで、姿勢領域から姿勢領域への実行可能性を定義でき、位置と姿勢領域の二項組をノードとした有向グラフ型のマップを生成できることを示した。また、提案するロードマップを用いることで、非ホロノミック拘束を有する車両型移動ロボットが走行可能な軌道を計画することが可能であり、その軌道を追従する制御入力計画も可能である。最後に、提案手法を用いた数値シミュレーションの結果を示し、提案するロードマップの有効性を示した。

さらに、提案するロードマップを用いた自動駐車システムを提案し、実機実験でその有効性を示した。

## 6.2 今後の課題および展望

本研究では、可変分解能処理に基づいたロードマップ生成手法を提案したが、実用化までには多くの課題が残されている。

第3章にて提案した手法では、目標位置に到達することを前提として地図生成と計画を行っている。しかしながら、未知環境下では目標位置到達を優先するのではなく、環境情報の獲得を優先すべき状況がある。そのため、状況によって行動の優先度を切り替えられる手法の開発が必要である。

第4章にて提案した手法では、実行可能領域が解析的に求められる必要があるために特殊な制御モデルを用いている。しかしながら、一般的には実行可能領域が解析的に求められるわけではない。そのため、制御モデルから実行可能領域を求める手法の確立が必要である。

第5章にて提案した手法でを用いて自動駐車システムを開発した。提案手法を適用することで自動車の初期位置から目標駐車位置までのグローバルな軌道を生成できる。しかしながら、初期位置が所定のノード上を想定しており、外乱に関しては他の自動車、想定外の障害物や歩行者などは想定しない。そのため、実用化するためには想定していなかつ

た外乱をローカルなプランニングで回避する必要がある。また、回避後にグローバルな軌道をロードマップ外の位置からロードマップを用いた計画が行えるように拡張する必要がある。





## 参考文献

- [1] 吉岡孝, 登尾啓史, 徳永昌治. “移動ロボットのセンサベースナビゲーションについて”. 日本ロボット学会誌, Vol. 19, No. 8, pp. 991–1002, 2001.
- [2] I. Kamon and E. Rivlin. “Sensory-Based Motion Planning with Global Proofs”. *IEEE Transaction on Robotics and Automation*, Vol. 13, No. 6, pp. 814–822, 1997.
- [3] M.W.M.G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. “A Solution to the Simultaneous Localization and Map Building(SLAM) Problem”. *IEEE Transaction on Robotics and Automation*, Vol. 17, No. 3, pp. 229–241, 2001.
- [4] H.Choset and K.Nagatani. “Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization”. *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 2, pp. 125–137, 2001.
- [5] A. Martinelli, A. Tapus, and K. O. Arras. “Multi-resolution SLAM for Real World Navigation”. In *Proceedings of the 11th International Symposium of Robotics Research*, 2003.
- [6] L.M. Paz, P. Pinies, J.D. Tardos, and J. Neira. “Large-Scale 6-DOF SLAM With Stereo-in-Hand”. *IEEE Transactions on Robotics*, Vol. 24, No. 5, pp. 946–957, 2008.
- [7] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer. “Fast and Incremental Method for Loop-Closure Detection Using Bags of Visual Words”. *IEEE Transactions on Robotics*, Vol. 24, No. 5, pp. 1027–1037, 2008.
- [8] Hyun Chul Roh, Chang Hun Sung, Min Tae Kang, and Myung Jin Chung. “Fast SLAM using polar scan matching and particle weight based occupancy grid map for mobile robot”. In *Ubiquitous Robots and Ambient Intelligence (URAI), 2011 8th International Conference on*, pp. 756–757, 2011.
- [9] H. Gonzalez-Banos and J.C. Latombe. “A randomized art-gallery algorithm

- for sensor placement”, 2001.
- [10] Hartmut Surmann, Andreas Nüchter, and Joachim Hertzberg. “An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments”. *Robotics and Autonomous Systems*, Vol. 45, No. 3-4, pp. 181–198, 2003.
- [11] Hector Gonzalez-Banos and Jean-claude Jean-claude Latombe. “Planning Robot Motions for Range-Image Acquisition and Automatic 3D Model Construction”, 1998.
- [12] A. Yamashita, S. Iwashina, and T. Kaneko. “View planning and 3D map building by a mobile robot equipped with two range sensors”, oct. 2010.
- [13] 文仁赫, 永谷圭司, 油田信一. “不確かさを考慮した観測位置と移動のオンライン計画手法”. 日本ロボット学会誌, Vol. 17, No. 8, pp. 1107–1113, 1999.
- [14] 永谷圭司, 油田信一. “衝突の危険性を評価関数とする移動ロボットの経路とセンシング点の計画”. 日本ロボット学会誌, Vol. 15, No. 2, pp. 197–206, 1997.
- [15] R.A. Brooks and T. Lozano-Perez. “A subdivision algorithm in configuration space for findpath with rotation”. *Systems, Man and Cybernetics, IEEE Transactions on*, Vol. SMC-152, number =, , 1985.
- [16] S. Kambhampati and L. Davis. “Multiresolution path planning for mobile robots”. *IEEE Journal of Robotics and Automatiion*, Vol. 2, No. 3, pp. 135–145, 1986.
- [17] G. Foux, M. Heymann, and A. Bruckstein. “Two-dimensional robot navigation among unknown stationary polygonal obstacles”. *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 1, pp. 96–102, 1993.
- [18] 太田順, 倉林大輔, 新井民夫. “知能ロボット入門: 動作計画問題の解法”. コロナ社, 2001.
- [19] Mankyu Sung, Lucas Kovar, and Michael Gleicher. “Fast and accurate goal-directed motion synthesis for crowds”. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 291–300, 2005.
- [20] Jonghoek Kim, Fumin Zhang, and M. Egerstedt. “An exploration strategy by constructing Voronoi diagrams with provable completeness”. In *Proceedings of the 48th IEEE Conference on Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009.*, pp. 7024–7029, 2009.
- [21] S.M. LaValle. “*Planning Algorithms*”. Cambridge University Press, Cam-

- 
- bridge, U.K., 2006. Available at <http://planning.cs.uiuc.edu/>.
- [22] Jean-Claude Latombe. “*Robot Motion Planning*”. Kluwer Academic Publishers, Norwell, MA, USA, 1991.
- [23] Jur P. van den Berg and Mark H. Overmars. “Using Workspace Information as a Guide to Non-uniform Sampling in Probabilistic Roadmap Planners”. *The International Journal of Robotics Research*, Vol. 24, , 2005.
- [24] H.Choset, K.Nagatani, and A.Rizzi. “Sensor Based Planning Using a Honing Strategy and Local Map Method to Implement the Generalized Voronoi Graph”. *SPIE Technical Conference on Mobile Robots*, Vol. 3210, pp. 72–83, 1997.
- [25] Howie Choset and Joel Burdick. “Sensor-Based Exploration: The Hierarchical Generalized Voronoi Graph”. *The International Journal of Robotics Research*, Vol. 19, No. 2, pp. 96–125, 2000.
- [26] H. Choset, K. Nagatani, and N.A. Lazar. “The Arc-Transversal Median Algorithm: A Geometric Approach to Increasing Ultrasonic Sensor Azimuth Accuracy”. *IEEE Transactions on Robotics and Automation*, Vol. 19, No. 3, pp. 513–522, 2003.
- [27] P. Beeson, N.K. Jong, and B. Kuipers. “Towards autonomous topological place detection using the extended voronoi graph”. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 4373–4379, 2005.
- [28] P. Bhattacharya and M.L. Gavrilova. “Roadmap-Based Path Planning - Using the Voronoi Diagram for a Clearance-Based Shortest Path”. *Robotics Automation Magazine, IEEE*, Vol. 15, No. 2, pp. 58–66, 2008.
- [29] D. Avis and G.T. Toussaint. “An Optimal Algorithm for Determining the Visibility of a Polygon from an Edge”. *Computers, IEEE Transactions on*, Vol. C-30, No. 12, pp. 910–914, 1981.
- [30] 劉雲輝, 有本卓. “一般的形状の障害物が存在する環境における移動ロボットの経路計画法”. *日本ロボット学会誌*, Vol. 9, No. 2, pp. 56–63, 1991.
- [31] L.E. Kavraki, P. Svestka, J.-C. Latombe, and M.H. Overmars. “Probabilistic roadmaps for path planning in high-dimensional configuration spaces”. *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 4, pp. 566–580, 1996.
- [32] Steven M. LaValle, James J. Kuffner, and Jr. “Rapidly-Exploring Random Trees: Progress and Prospects”. In *Algorithmic and Computational Robotics: New Directions*, pp. 293–308, 2000.

- [33] C.O'Dunlaing and C.K.Yap. "A "retracion" method for planning the motion of a disk". *J.Algorithms*, No. 6, pp. 104–111, 1985.
- [34] V.Boor, M.H.Overmars, and A.F.van der Stappen. "The Gaussian Sampling strategy for probabilistic roadmap planners". *IEEE Int. Conf. Robotics and Automation*, pp. 1018–1023, 1999.
- [35] R.bohlin and L.E.Kavaraki. "Path Planning Using Lazy PRM". *IEEE Int. Conf. Robotics and Automation*, pp. 521–528, 2000.
- [36] 竹澤聡, Tauseef GULREZ, Damith C. HERATH, Gamini DISSANAYAKE. "同時自己位置地図獲得手法を用いた自律型ロボットの環境認識：ダイナミカル局所ボロノイ分割による逐次経路計画 (機械力学, 計測, 自動制御)". 日本機械学会論文集. C 編, Vol. 71, No. 703, pp. 904–911, 2005.
- [37] O. Khatib. "Real-time obstacle avoidance for manipulators and mobile robots". In *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 500–505, 1985.
- [38] J.-O. Kim and P.K. Khosla. "Real-time obstacle avoidance using harmonic potential functions". *IEEE Transactions on Robotics and Automation*, Vol. 8, No. 3, pp. 338–349, 1992.
- [39] 辻繁夫, 千種和幸, 金子真. "人工ポテンシャル場の変形を用いた移動ロボットの軌道生成法". 日本機械学会論文集. C 編, Vol. 62, No. 597, pp. 1905–1911, 1996.
- [40] L. Huang. "Velocity planning for a mobile robot to track a moving target - a potential field approach". *Robotics and Autonomous Systems*, Vol. 57, No. 1, pp. 55 – 63, 2009.
- [41] Danping Peng, Barry Merriman, Stanley Osher, Hongkai Zhao, and Myungjoo Kang. "A PDE-Based Fast Local Level Set Method". *Journal of Computational Physics*, Vol. 155, No. 2, pp. 410–438, 1999.
- [42] 亮倉爪. "レベルセット法とその実装法について (チュートリアル).
- [43] M. Sabry Hassouna, Alaa E. Abdel-Hakim, and Aly A. Farag. Pde-based robust robotic navigation. *Image and Vision Computing*, Vol. 27, No. 1-2, pp. 10–18, 2009.
- [44] Michael Montemerlo and Sebastian Thrun. "A Multi-Resolution Pyramid for Outdoor Robot Terrain Perception.". In *AAAI*, pp. 464–469. AAAI Press / The MIT Press, 2004.
- [45] Ursula C. Benz, P. Hofmann, G. Willhauck, I. Lingenfelder, and M. Heynen. "Multi-resolution, object-oriented fuzzy analysis of remote sensing data for

- 
- GIS-ready information”. *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 58, No. 3-4, pp. 239–258, 2004.
- [46] Patrick Beeson, Joseph Modayil, and Benjamin Kuipers. Factoring the mapping problem: Mobile robot map-building in the hybrid spatial semantic hierarchy. *The International Journal of Robotics Research*, Vol. 29, No. 4, pp. 428–459, 2010.
- [47] Jean-Arcady Meyer and David Filliat. Map-based navigation in mobile robots: Ii. a review of map-learning and path-planning strategies. *Cognitive Systems Research*, Vol. 4, No. 4, pp. 283–317, 2003.
- [48] 一美及川, 秀則高氏, 貴紀江丸, 武士土谷, 重範大久保. 自律移動ロボットのためのトポロジカルマップを用いた環境表現. インテリジェント・システム・シンポジウム講演論文集 = FAN Symposium : fuzzy, artificial intelligence, neural networks and computational intelligence, Vol. 10, pp. 137–140, oct 2000.
- [49] F. Savelli and B. Kuipers. “Loop-closing and planarity in topological map-building”. In *Proceedings. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004. (IROS 2004).*, Vol. 2, pp. 1511 – 1517, 2004.
- [50] Tae-Bum Kwon and Jae-Bok Song. “Real-time building of a thinning-based topological map”. *Intelligent Service Robotics*, Vol. 1, No. 3, pp. 211–220, 2008.
- [51] Yu-Cheol Lee, Christiand, Seung-Hwan Park, Wonpil Yu, and Sung-Hoon Kim. “Topological map building for mobile robots based on GIS in urban environments”. In *Ubiquitous Robots and Ambient Intelligence (URAI), 2011 8th International Conference on*, pp. 790–791, 2011.
- [52] M. Bennewitz, C. Stachniss, S. Behnke, and W. Burgard. “Utilizing reflection properties of surfaces to improve mobile robot localization”. In *Proceedings of IEEE International Conference on Robotics and Automation, 2009. ICRA '09.*, 2009, pages=.
- [53] Bingfeng Wang, Shigang Cui, Li Zhao, Genghuang Yang, and Xuelian Xu. “Mobile Robot Map Building Based on Grid Arrangement”. In *Proceedings of International Conference on Artificial Intelligence and Computational Intelligence, 2009. AICI '09.*, Vol. 2, pp. 288–291, 2009.
- [54] Peng Li, Xinhan Huang, and Min Wang. “A novel hybrid method for mobile robot path planning in unknown dynamic environment based on hybrid DS<sub>m</sub> model grid map”. *Journal of Experimental and Theoretical Artificial*

- Intelligence*, Vol. 23, .
- [55] Sergio Almansa-Valverde, Jose Carlos Castillo, and Antonio Fernandez-Caballero. “Mobile robot map building from time-of-flight camera”. *Expert Systems with Applications*, Vol. 39, No. 10, pp. 8835–8843, 2012.
- [56] S. Simhon and G. Dudek. A global topological map formed by local metric maps. In *Proceedings., 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems, 1998.*, Vol. 3, pp. 1708–1714, 1998.
- [57] N. Tomatis, I. Nourbakhsh, and R. Siegwart. Hybrid simultaneous localization and map building: a natural integration of topological and metric, 2003.
- [58] R. Vazquez-Martin, P. Nunez, A. Bandera, and F. Sandoval. “Spectral clustering for feature-based metric maps partitioning in a hybrid mapping framework”. In *Proceedings of IEEE International Conference on Robotics and Automation, 2009. ICRA '09.*, pp. 4175–4181, 2009.
- [59] A.G. Ozkil, Zhun Fan, Jizhong Xiao, J.K. Kristensen, S. Dawids, K.H. Christensen, and H. Aanaes. “Practical indoor mobile robot navigation using hybrid maps”. In *Mechatronics (ICM), 2011 IEEE International Conference on*, pp. 475–480, 2011.
- [60] Emmanuel Battesti, Stéphane Bazeille, David Filliat, et al. Qualitative localization using vision and odometry for path following in topo-metric maps. In *Proceedings of the European Conference on Mobile Robotics (ECMR)*, pp. 303–308, 2011.
- [61] Naniwa-T. Noborio, H. and S. Arimoto. “A quadtree-based path-planning algorithm for a mobile robot”. *J. Robotic Syst.*, Vol. 7, pp. 555–574, 1990.
- [62] R.H.T. Chan, P.K.S. Tam, and D.N.K. Leung. “Robot navigation in unknown terrains via multi-resolution grid maps”. In *Proceedings of IECON '91., 1991 International Conference on Industrial Electronics, Control and Instrumentation*, Vol. 2, pp. 1138–1143, 1991.
- [63] D. Jung and K. K. Gupta. “Octree-based hierarchical distance maps for collision detection”. *J. Robotic Syst.*, Vol. 14, pp. 789–806, 1997.
- [64] S. Behnke. “Local Multiresolution Path Planning”. In *RoboCup 2003: Robot Soccer World Cup VII*.
- [65] Gerhard K. Kraetzschmar, Guillem Pages Gassull, Klaus Uhl, Guillem Pages, and Gassull Klaus Uhl. “Probabilistic Quadtrees for Variable-Resolution Mapping of Large Environments”. In *Eds.), Proceedings of the*

- 
- 5th IFAC/EURON, 2004.
- [66] R. Kirby, R. Simmons, and J. Forlizzi. “Variable sized grid cells for rapid replanning in dynamic environments”. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009. IROS 2009.*, 2009.
- [67] S. Quinlan and O. Khatib. “Elastic bands: connecting path planning and control”. In *Proceedings of IEEE International Conference on Robotics and Automation, 1993.*, Vol. 2, pp. 802–807, 1993.
- [68] Guang Song, S. Miller, and N.M. Amato. “Customizing PRM roadmaps at query time”. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation, 2001.*, Vol. 2, pp. 1500 – 1505, 2001.
- [69] M. Yamamoto, M. Iwamura, and A. Mohri. “Quasi-time-optimal motion planning of mobile platforms in the presence of obstacles”. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation, 1999.*, Vol. 4, pp. 2958–2963, 1999.
- [70] M. Kazemi, M. Mehrandezh, and K. Gupta. “An Incremental Harmonic Function-based Probabilistic Roadmap Approach to Robot Path Planning”. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005.*, pp. 2136–2141, 2005.
- [71] Bruce Donald, Patrick Xavier, John Canny, and John Reif. “Kinodynamic motion planning”. *J. ACM*, Vol. 40, No. 5, pp. 1048–1066, 1993.
- [72] Steven M. LaValle and James J. Kuffner. Randomized kinodynamic planning. “*The International Journal of Robotics Research*”, Vol. 20, No. 5, pp. 378–400, 2001.
- [73] A. Masoud. “Kinodynamic Motion Planning”. *IEEE Robotics Automation Magazine.*, Vol. 17, No. 1, pp. 85–99, 2010.
- [74] Steven M LaValle. “Planning Algorithms”, 2006.
- [75] D. Fox, W. Burgard, and S. Thrun. “The dynamic window approach to collision avoidance”. *IEEE Robotics Automation Magazine, IEEE*, Vol. 4, No. 1, pp. 23–33, 1997.
- [76] I.A. Sucas and L.E. Kavraki. “Kinodynamic Motion Planning by Interior-Exterior Cell Exploration”. In *Algorithmic Foundation of Robotics VIII*, Vol. 57 of *Springer Tracts in Advanced Robotics*, pp. 449–464. Springer Berlin Heidelberg, 2009.
- [77] R.V. Cowlagi and P. Tsiotras. “Multi-resolution H-cost motion planning:

- A new framework for hierarchical motion planning for autonomous mobile vehicles”. In *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3501–3506, 2011.
- [78] S. Karaman and E. Frazzoli. “Optimal kinodynamic motion planning using incremental sampling-based methods”. In *Proceedings of the 2010 49th IEEE Conference on Decision and Control (CDC)*, pp. 7681–7687, 2010.
- [79] Mihail Pivtoraiko, Ross A Knepper, and Alonzo Kelly. Differentially constrained mobile robot motion planning in state lattices. *Journal of Field Robotics*, Vol. 26, No. 3, pp. 308–333, 2009.
- [80] Edsger. W. Dijkstra. “A note on two problems in connexion with graphs.”. *Numerische Mathematik*, Vol. 1, pp. 269–271, 1959.
- [81] P.E. Hart, N.J. Nilsson, and B. Raphael. “A Formal Basis for the Heuristic Determination of Minimum Cost Paths”. *Systems Science and Cybernetics, IEEE Transactions on*, Vol. 4, No. 2, pp. 100–107, july 1968.
- [82] F. Chimura and M. Tokoro. “The Trailblazer search: A new method for searching and capturing moving targets”, oct. 1994.
- [83] R.Korf. “Real-time heuristic search”. *Artificial Intelligence*, Vol. 42, No. 2-3, pp. 189–211, 1990.
- [84] A. Stentz. ”Optimal and efficient path planning for partially-known environments”. In *Proceedings of 1994 IEEE International Conference on Robotics and Automation*, Vol. 4, pp. 3310–3317, may 1994.
- [85] A. Stentz. “The focussed D\* algorithm for real-time replanning”. In *Proceedings of the 14th international joint conference on Artificial intelligence*, Vol. 2, pp. 1652–1659, 1995.
- [86] S. Koenig and M. Likhachev. “Fast replanning for navigation in unknown terrain”. *IEEE Transactions on Robotics*, Vol. 21, No. 3, pp. 354–363, june 2005.
- [87] D. Ferguson and A. Stentz. “Field D\*: An Interpolation-Based Path Planner and Replanner”. In *Robotics Research*, Vol. 28 of *Springer Tracts in Advanced Robotics*, pp. 239–253. Springer Berlin Heidelberg, 2007.
- [88] R. Phillippsen. “A Light Formulation of the E\* Interpolated Path Replanner”. Technical report, Autonomous Systems Lab, Ecole Polytechnique Federale de Lausanne, 2006.
- [89] Cédric Pradalier, Jorge Hermosillo, Carla Koike, Christophe Brailon, Pierre Bessière, and Christian Laugier. The i<sub>c</sub> cyca<sub>i</sub>/i<sub>c</sub> b: a car-like robot navigat-



- 
- ing autonomously and safely among pedestrians. *Robotics and Autonomous Systems*, Vol. 50, No. 1, pp. 51–67, 2005.
- [90] Bernhard Muller, Joachim Deutscher, and Stefan Grodde. Trajectory generation and feedforward control for parking a car. In *Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE*, pp. 163–168. IEEE, 2006.
- [91] Antonio Bicchi, Giuseppe Casalino, and Corrado Santilli. Planning shortest bounded-curvature paths for a class of nonholonomic vehicles among obstacles. *Journal of Intelligent and Robotic Systems*, Vol. 16, No. 4, pp. 387–405, 1996.
- [92] Ming Feng Hsieh and Umit Ozguner. A parking algorithm for an autonomous vehicle. In *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 1155–1160. IEEE, 2008.
- [93] C Pradalier, S Vaussier, and P Corke. Path planning for a parking assistance system: Implementation and experimentation. *Sydney, Australia: Australian Robotics and Automation Association Inc*, 2005.
- [94] Moritz B Oetiker, Gion P Baker, and Lino Guzzella. A navigation-field-based semi-autonomous nonholonomic vehicle-parking assistant. *Vehicular Technology, IEEE Transactions on*, Vol. 58, No. 3, pp. 1106–1118, 2009.



# 付録

## A.1 全方向移動ロボットの実行可能領域の導出例

本付録では、4.4 節の導出の詳細について述べる。なお、定義は Fig. 4.2(b) および 4.2.2 節に準ずる。等加速度運動を行う移動ロボットの時刻  $t$  での位置  $p(t) = [x(t), y(t)]^T$  と速度  $v(t) = [v^x(t), v^y(t)]^T$  を表す運動方程式は次式である：

$$p(t) = p_0 + v_0 t + \frac{1}{2} a t^2, \quad (\text{A.1})$$

$$v(t) = v_0 + a t, \quad (\text{A.2})$$

この時、 $a = [a^x, a^y]^T$  は加速度であり、 $v_0 = [v_0^x, v_0^y]^T$  は初速度であり、 $p_0 = [p_0^x, p_0^y]^T$  は初期位置である。(A.1) 式および (A.2) 式を用いて制約条件の導出を説明する。

### A.1.1 衝突回避

本節では 4.4.2 節の導出の詳細について述べる。(A.1) 式を  $y$  方向と  $x$  方向に分け、それぞれについて述べる。

$y$  方向 (線分の垂直方向)

移動ロボットの二地点間の移動における垂直方向 ( $y$  軸方向) の運動方程式を次に示す：

$$y(t) = \begin{cases} p_0^y + v_0^y t + \frac{1}{2} u_1^y t^2, & \forall t \in [0, T/2], \\ y\left(\frac{T}{2}\right) + v^y\left(\frac{T}{2}\right) \left(t - \frac{T}{2}\right) + \frac{1}{2} u_2^y \left(t - \frac{T}{2}\right)^2, & \forall t \in [T/2, T], \end{cases} \quad (\text{A.3})$$

$$v^y(t) = \begin{cases} v_0^y + u_1^y t, & \forall t \in [0, T/2], \\ v^y\left(\frac{T}{2}\right) + u_2^y \left(t - \frac{T}{2}\right), & \forall t \in [T/2, T]. \end{cases} \quad (\text{A.4})$$

境界条件の (4.4) 式より次式が得られる :

$$y(t) = \begin{cases} v^y t + \frac{1}{2} u_1^y t^2, & \forall t \in [0, T/2], \\ v'^y (t - T) + \frac{1}{2} u_2^y (t - T)^2, & \forall t \in [T/2, T], \end{cases} \quad (\text{A.5})$$

$$v^y(t) = \begin{cases} v^y + u_1^y t, & \forall t \in [0, T/2], \\ v'^y + u_2^y (t - T), & \forall t \in [T/2, T]. \end{cases} \quad (\text{A.6})$$

ここで,  $y_{\max} = \max |y(t)|$  ( $0 \leq t \leq T$ ) と定義すると, (4.7a) 式は次式のように書ける :

$$y_{\max} \leq w(q, q'). \quad (\text{A.7})$$

そこで,  $y_{\max}$  の導出について述べる .

二段階の等加速度運動であるため,  $y_{\max}$  となる時刻  $t$  は  $0 \leq t \leq \frac{T}{2}$  もしくは  $\frac{T}{2} < t \leq T$  である . ここで, 時刻  $t$  ( $0 \leq t \leq \frac{T}{2}$ ) である区間を第一区間と定義し, 時刻  $t$  ( $\frac{T}{2} < t \leq T$ ) である区間を第二区間と定義する . そして, それぞれの区間で  $|y(t)|$  が最大となる時刻を  $t_\alpha$  ( $0 \leq t_\alpha \leq \frac{T}{2}$ ) および  $t_\beta$  ( $\frac{T}{2} < t_\beta \leq T$ ) と定義する .

$y(t)$  は速度と加速度によって定まる . なお,  $v^y \geq 0$  と  $v^y \leq 0$  には対称性が存在するため,  $v^y \geq 0$  の場合のみを示す . 場合分けは  $v^y, v'^y, u_1^y, u_2^y$  について 4 種類に分類できる .

(i)  $v^y \geq 0, v'^y < 0, u_1^y \geq 0, u_2^y < 0$  の場合 場合分けの条件, (4.5a) 式および (4.5b) 式より次式が得られる :

$$v'^y \leq -3v^y \leq 0. \quad (\text{A.8})$$

この場合では, 第一区間では  $v^y(t) \geq 0, \forall t \in [0, \frac{T}{2}]$  が成立するため,  $|y(t_\alpha)| \leq |y(t_\beta)|$  である :

$$y_{\max} = |y(t_\beta)|, \quad \text{if } v'^y \leq -3v^y \leq 0. \quad (\text{A.9})$$

(ii)  $v^y \geq 0, v'^y < 0, u_1^y < 0, u_2^y < 0$  の場合 場合分けの条件より場合分けの範囲は次式である :

$$-3v^y < v'^y < -\frac{v^y}{3}. \quad (\text{A.10})$$

この場合では, 速度  $v^y(\frac{T}{2})$  によって  $|y(t_\alpha)|$  と  $|y(t_\beta)|$  の大小関係が変わる . (4.5b) 式より  $v^y(\frac{T}{2}) = -\frac{1}{2}(v^y + v'^y)$  が得られる .  $v^y(\frac{T}{2}) < 0$  の時, すなわち  $-v^y < v'^y < 0$  の時, 第一区間ですでに方向転換しているため,  $|y(t_\alpha)| > |y(t_\beta)|$  である :

$$y_{\max} = |y(t_\alpha)|, \quad \text{if } -v^y < v'^y < -\frac{v^y}{3}. \quad (\text{A.11})$$

一方,  $v^y \left(\frac{T}{2}\right) \geq 0$  の時, すなわち  $v'^y \leq -v^y \leq 0$  の時, 第一区間ではまだ方向転換をしていないため,  $|y(t_\alpha)| \leq |y(t_\beta)|$  である:

$$y_{\max} = |y(t_\beta)|, \quad \text{if } -3v^y < v'^y \leq -v^y. \quad (\text{A.12})$$

(iii)  $v^y \geq 0, v'^y < 0, u_1^y < 0, u_2^y \geq 0$  の場合 場合分けの条件より場合分けの範囲は次式である:

$$-v^y/3 \leq v'^y < 0. \quad (\text{A.13})$$

この場合では,  $v'^y < 0, u_2^y \geq 0$  より  $y\left(\frac{T}{2}\right) > 0, v^y\left(\frac{T}{2}\right) < 0$  であるため,  $|y(t_\alpha)| > |y(t_\beta)|$  である:

$$y_{\max} = |y(t_\alpha)|, \quad \text{if } -\frac{v^y}{3} \leq v'^y < 0. \quad (\text{A.14})$$

(iv)  $v^y \geq 0, v'^y \geq 0, u_1^y < 0, u_2^y \geq 0$  の場合 場合分けの条件より場合分けの範囲は次式である:

$$0 \leq v'^y. \quad (\text{A.15})$$

この場合では, 第一区間と第二区間のそれぞれで極値を持つため,  $y(t_\alpha)$  と  $y(t_\beta)$  を比較する必要がある. そこで, (A.5) 式, (4.5a) 式および (4.5b) 式より次式が得られる:

$$y(t_\alpha) = \frac{(v^y)^2 T}{2(3v^y + v'^y)}, \quad t_\alpha = \frac{v^y T}{3v^y + v'^y}, \quad (\text{A.16})$$

$$y(t_\beta) = \frac{-(v'^y)^2 T}{2(v^y + 3v'^y)}, \quad t_\beta = \frac{v'^y T}{v^y + 3v'^y}. \quad (\text{A.17})$$

まず,  $|y(t_\alpha)| \geq |y(t_\beta)|$  について考える. この式について整理すると, 次式が得られる:

$$(v^y - v'^y) \left( (v^y)^2 + 4v^y v'^y + (v'^y)^2 \right) \geq 0. \quad (\text{A.18})$$

ここで,  $\left( (v^y)^2 + 4v^y v'^y + (v'^y)^2 \right) > 0$  より,  $v^y \geq v'^y$  の時に,  $|y(t_\alpha)| \geq |y(t_\beta)|$  である. 同様に  $v^y < v'^y$  の時に,  $|y(t_\alpha)| < |y(t_\beta)|$  である. 以上より次式が得られる:

$$y_{\max} = \begin{cases} |y(t_\alpha)|, & \text{if } 0 \leq v'^y \leq v^y, \\ |y(t_\beta)|, & \text{if } 0 \leq v^y < v'^y. \end{cases} \quad (\text{A.19})$$

それぞれの場合分けでの結果 (A.9), (A.11), (A.12), (A.14) および (A.19) 式をまとめると (4.10) 式が得られる.

× 方向（線分の接線方向）

移動ロボットの二地点間の移動における接続方向（ $x$  軸方向）の運動方程式を次に示す：

$$v^x(t) = v_0^x + u^x t \quad (\text{A.20})$$

$$x(t) = p_0^x + v_0^x t + \frac{1}{2} u^x t^2. \quad (\text{A.21})$$

ここで、境界条件  $x(0) = 0$ ,  $v_0^x = v^x$ ,  $x(T) = d(q, q')$  および  $v^x(T) = v'^x$  より, (4.5) 式が得られる.

まず, (4.9) 式より  $-v^x < v'^x$  である. この時,  $x$  軸方向の軌道は大きく 3 種類に分類できる.

(i)  $0 < v^x, 0 < v'^x$  の場合 この場合では次式が成立するため, 任意の速度を選択可能である:

$$0 \leq x(t) \leq d(q, q'), \quad \forall t \in [0, T]. \quad (\text{A.22})$$

(ii)  $0 < v^x, 0 \geq v'^x$  の場合 この場合では終端速度が負であるため,  $x$  軸方向の正側に極値が存在する. その時刻を  $t_\gamma$  とした場合, 以下の条件が得られる:

$$x(t_\gamma) \leq d(q, q') + \epsilon_2(q, q'). \quad (\text{A.23})$$

極値では  $v^x(t_\gamma) = 0$  となるため, (A.20) 式より  $t_\gamma$  が次式のように求まる:

$$t_\gamma = -\frac{v^x}{u^x} = \frac{2v^x d(q, q')}{(v^x + v'^x)(v^x - v'^x)} > 0. \quad (\text{A.24})$$

(A.24) 式を (A.23) 式に代入すると次式が得られる:

$$-\sqrt{d(q, q') + \epsilon_2(q, q')} v'^x \leq \sqrt{\epsilon_2(q, q')} v^x. \quad (\text{A.25})$$

(iii)  $0 \geq v^x, 0 < v'^x$  の場合 この場合では初速度が負であるため,  $x$  軸方向の負側に極値が存在する. その時刻を  $t_\gamma$  とした場合, 以下の条件が得られる:

$$0 - \epsilon_1(q, q') \leq x(t_\gamma). \quad (\text{A.26})$$

ここで,  $t_\gamma$  は上記の場合と同様に (A.24) 式になる. そして, (A.24) 式を (A.26) 式に代入すると次式が得られる:

$$-\sqrt{d(q, q') + \epsilon_1(q, q')} v^x \leq \sqrt{\epsilon_1(q, q')} v'^x. \quad (\text{A.27})$$

それぞれの場合分けでの結果 (A.22), (A.25) および (A.27) 式をまとめると (4.12) 式が得られる.

### A.1.2 速度制限

移動ロボットの速度は (A.6) および (A.20) 式で与えられる．等加速度運動であるため，二地点間の移動で速さが最大となる場所は各加速区間の両端のいずれかである．このため， $v(0)^2 \leq V_{\max}^2$ ， $v\left(\frac{T}{2}\right)^2 \leq V_{\max}^2$  および  $v(T)^2 \leq V_{\max}^2$  が成立する必要がある．この時，それぞれが次式で計算される：

$$v(0)^2 = v^2 = (v^x)^2 + (v^y)^2, \quad (\text{A.28a})$$

$$v\left(\frac{T}{2}\right)^2 = \left(v^x + u^x \frac{T}{2}\right)^2 + \left(v^y + u_1^y \frac{T}{2}\right)^2, \quad (\text{A.28b})$$

$$v(T)^2 = v'^2 = (v'^x)^2 + (v'^y)^2. \quad (\text{A.28c})$$

以上より (4.13) 式が得られる．

### A.1.3 加速度制限

本稿では加速度制約を  $x$  軸方向と  $y$  軸方向に分けている．最初に， $u^x$  について考える．(4.5) 式より次式が得られる：

$$u^x = \frac{-v^x + v'^x}{T} = \frac{(-v^x + v'^x)(v^x + v'^x)}{2d(q, q')}. \quad (\text{A.29})$$

(4.15) 式に代入すると次式が得られる：

$$|u^x| = \left| \frac{(v'^x)^2 - (v^x)^2}{2d(q, q')} \right| \leq \frac{A_{\max}}{\sqrt{2}}. \quad (\text{A.30})$$

ここで速度成分について整理すると (4.16) 式が得られる．

次に， $u_1^y$  について考える．(4.5a) 式および (4.5c) 式より次式が得られる：

$$u_1^y = \frac{-(3v^y + v'^y)}{T} = \frac{-(3v^y + v'^y)(v^x + v'^x)}{2d(q, q')}. \quad (\text{A.31})$$

(4.15) 式に代入すると次式が得られる：

$$|u_1^y| = \left| \frac{-(3v^y + v'^y)(v^x + v'^x)}{2d(q, q')} \right| \leq \frac{A_{\max}}{\sqrt{2}}. \quad (\text{A.32})$$

ここで速度成分について整理すると次式が得られる：

$$\left| \frac{(3v^y + v'^y)(v^x + v'^x)}{4} \right| \leq \frac{d(q, q') A_{\max}}{4\sqrt{2}}. \quad (\text{A.33})$$

この条件式は Fig. 4.11 に示す双曲線を表す .

最後に ,  $u_2^y$  について考える . (4.5b) 式および (4.5c) 式より次式が得られる :

$$u_2^y = \frac{(v^y + 3v'^y)}{T} = \frac{(v^y + 3v'^y)(v^x + v'^x)}{2d(q, q')}. \quad (\text{A.34})$$

(4.15) 式に代入すると次式が得られる :

$$|u_2^y| = \left| \frac{(v^y + 3v'^y)(v^x + v'^x)}{2d(q, q')} \right| \leq \frac{A_{\max}}{\sqrt{2}}. \quad (\text{A.35})$$

ここで速度成分について整理すると次式が得られる :

$$\left| \frac{(v^y + 3v'^y)}{4} \frac{(v^x + v'^x)}{2} \right| \leq \frac{d(q, q') A_{\max}}{4\sqrt{2}}. \quad (\text{A.36})$$

この条件式も Fig. 4.11 に示す双曲線と同様な双曲線を表す .

## A.2 車両型移動ロボットの実行可能領域の導出例

本付録では , 車両型移動ロボットの実行可能領域の導出について述べる .

### A.2.1 衝突回避の $y$ 軸方向

本節では , 車両型移動ロボットの衝突回避における実行可能領域の導出について述べる . ただし , 簡単のため F 軌道を想定して導出する . F 軌道であるため ,  $\theta = \psi$  および  $\theta' = \psi'$  である .

(5.16) 式は (5.23) 式および (5.24) 式を用いると次式のように変形できる :

$$p_y = c_v \sqrt{\left| \frac{\theta_{p'}}{c_a} \right|} \begin{bmatrix} \cos \theta_{p'} & -\sin \theta_{p'} \\ \sin \theta_{p'} & \cos \theta_{p'} \end{bmatrix} \begin{bmatrix} -C \left( \sqrt{\frac{|2\theta_{p'}|}{\pi}} \right) \\ -S \left( \sqrt{\frac{|2\theta_{p'}|}{\pi}} \right) \end{bmatrix} + p_s, \quad (\text{A.37})$$

$$\theta_{p'} = \theta' + \theta_{pp'},$$

$$p_s = \begin{cases} p_c & \text{if } \theta_A \geq 0, \\ p' & \text{otherwise.} \end{cases}$$

この時 , (5.23) 式および (5.24) 式より  $p_y$  の第一項は  $d(q, q')$  に対して線形であることが分かる . また ,  $p_c = [x_{p_c}, y_{p_c}]^T$  の  $y_{p_c}$  と  $d(q, q')$  は次の関係を満たす :

$$y_{p_c} \sin(\pi - \theta + \theta') = d(q, q') \sin \theta \sin \left( \frac{\theta + \theta'}{2} \right). \quad (\text{A.38})$$



さらに,  $\theta$  と  $\theta'$  は  $d(q, q')$  と無関係であるため, (A.37) 式および (A.38) 式より  $y_{py}$  は  $d(q, q')$  に対して線形であることが分かる:

$$y_{py} = f(\theta, \theta')d(q, q') \propto d(q, q'). \quad (\text{A.39})$$

なお,  $f(\theta, \theta')$  は (A.37) 式から得られる  $d(q, q')$  の係数である. ここで, (A.39) 式を (5.18) 式に代入することで次式が得られる:

$$\frac{w(q, q')}{d(q, q')} \leq f(\theta, \theta'). \quad (\text{A.40})$$

以上より, 一組の  $w(q, q')$  と  $d(q, q')$  に対して, (A.40) 式を満たす領域  $(\theta, \theta')$  が定まる.

(5.17) 式に関しても同様に計算でき, これらを組み合わせると Fig. 5.14 を獲得できる.

## A.2.2 旋回最小半径に関する実行可能領域

本節では, 旋回最小半径に関する拘束による実行可能領域の導出について述べる. ここで, 時刻  $t$  における移動ロボットの旋回半径, 線速度および旋回角速度をそれぞれ  $r(t)$ ,  $v(t)$  および  $\omega(t)$  と表す. 旋回半径  $r(t)$  は速度  $v(t)$  と旋回角速度  $\omega(t)$  を用いて次式で計算される:

$$r(t) = \frac{v(t)}{\omega(t)}. \quad (\text{A.41})$$

一つの軌道の中で旋回半径が最小となる位置は  $q_B$  であるため,  $q_B$  での旋回半径を  $r_{\min}$  と表すと, 次式で計算できる:

$$r_{\min} = r(T_B) = \frac{v(T_B)}{\omega(T_B)}. \quad (\text{A.42})$$

この時,  $v$  は一定であるので  $v(T_B) = v$  であり, 旋回角加速度  $\alpha$  が一定であるので  $\omega(T_B) = \alpha T_B$  である. これらより,  $r_{\min}$  は次式のように計算できる:

$$r_{\min} = \frac{v}{\alpha T_B} = \frac{\frac{1}{2}vT_B}{\frac{1}{2}\alpha T_B^2} = \frac{c_v}{2c_a}. \quad (\text{A.43})$$

(A.43) 式, (5.23) 式および (5.24) 式より  $r_{\min}$  と  $d(q, q')$  は以下の条件を満たす:

$$r_{\min} = f(\theta, \theta')d(q, q') \propto d(q, q'). \quad (\text{A.44})$$

なお,  $f(\theta, \theta')$  は (A.43) 式から計算される  $d(q, q')$  の係数である. ただし, この時,  $r_{\min}$  は  $d(q, q')$  に対しては線形であるが,  $r_{\min}$  は  $\theta$  と  $\theta'$  に対しては非線形である. ここで,

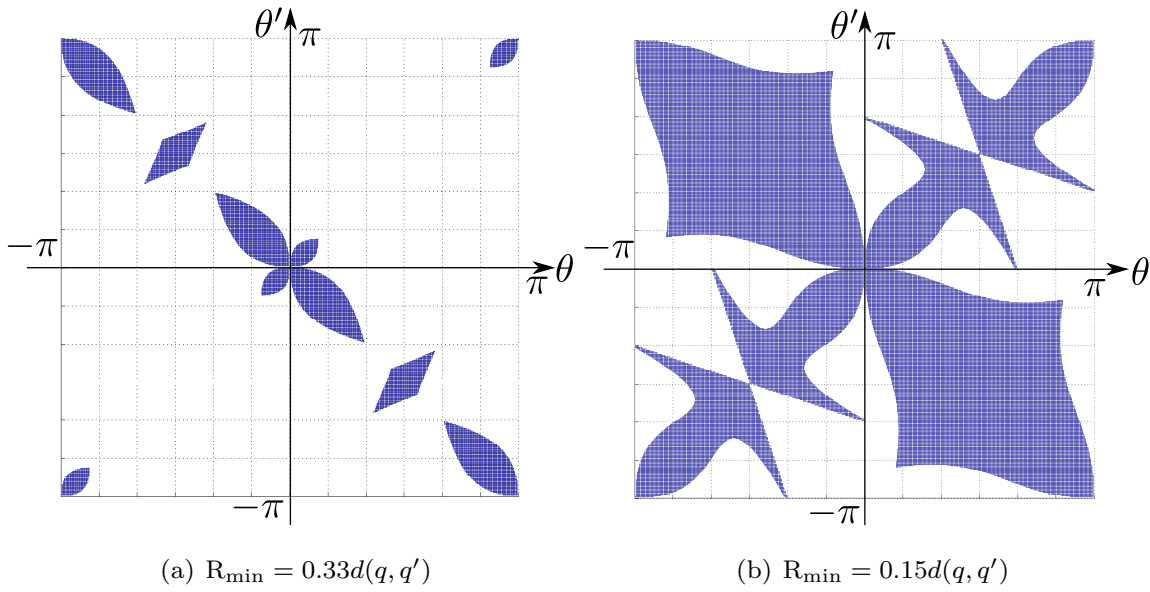


Fig. A.1 Feasible region with two  $R_{\min}$

移動ロボットの旋回最小半径を  $R_{\min}$  とおくと,  $r_{\min}$  が満たすべき条件は次式のように書ける :

$$R_{\min} \leq r_{\min}. \quad (\text{A.45})$$

整理すると, 次式が得られる :

$$\frac{R_{\min}}{d(q, q')} \leq f(\theta, \theta'). \quad (\text{A.46})$$

しかしながら, (A.46) 式を満たす領域  $(\theta, \theta')$  を解析的に求めることが困難である. そこで, 数値計算を用いて (A.45) 式を満たす領域  $(\theta, \theta')$  を求める.

$d(q, q')$  に対する  $R_{\min}$  の割合を変更して領域  $(\theta, \theta')$  を確認する.  $R_{\min} = 0.33d(q, q')$  の場合の領域と  $R_{\min} = 0.15d(q, q')$  の場合の領域を Fig. A.1 に示す. これらの図では, 青色領域が条件 (A.46) 式を満たす領域である. Figure A.1 より,  $d(q, q')$  と  $R_{\min}$  の関係によって実行可能領域の範囲が変化し,  $R_{\min}$  の  $d(q, q')$  に対する割合が小さいほど範囲が広くなるのが分かる (Fig. A.1(b)). しかしながら, 割合が小さいと領域の凸性が失われることも Fig. A.1(b) から分かる.

(A.45) 式を満たす領域  $(\theta, \theta')$  が上記の性質を有しているため, Figs. A.2, A.3 に示すように  $R_{\min}$  と  $d(q, q')$  の割合に対して個別に凸な実行可能領域を選択する. なお, これらの図では青色領域が元の実行可能領域であり, 緑線で囲まれた領域が選択して用いた元の実行可能領域の凸な部分領域である. さらに厳密に選択することが可能だが, 今回は八つに場合分けする. まず,  $R_{\min} \leq 0.05d(q, q')$  の範囲では, 実行可能領域が最も狭くなる

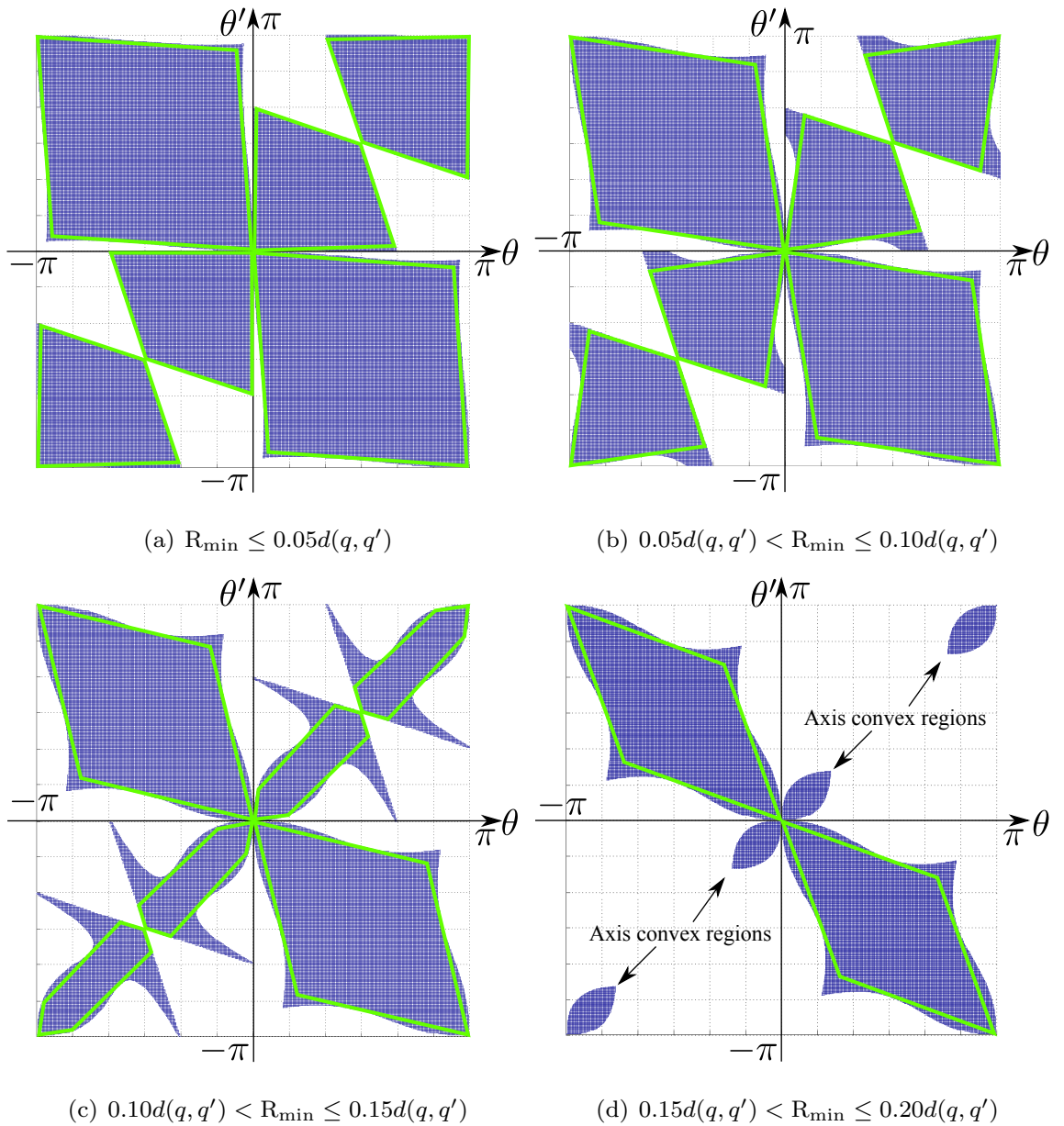
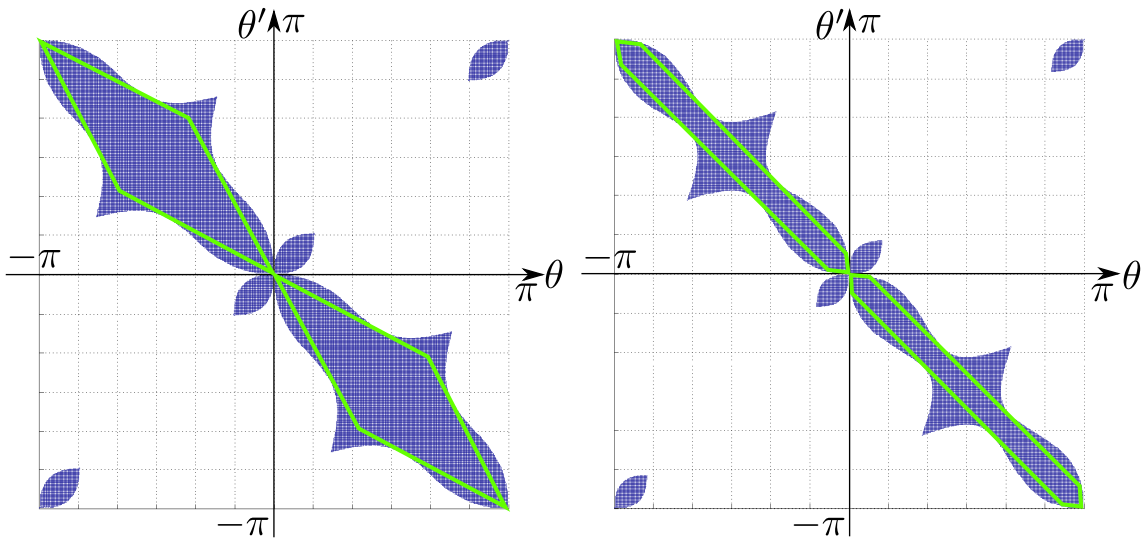


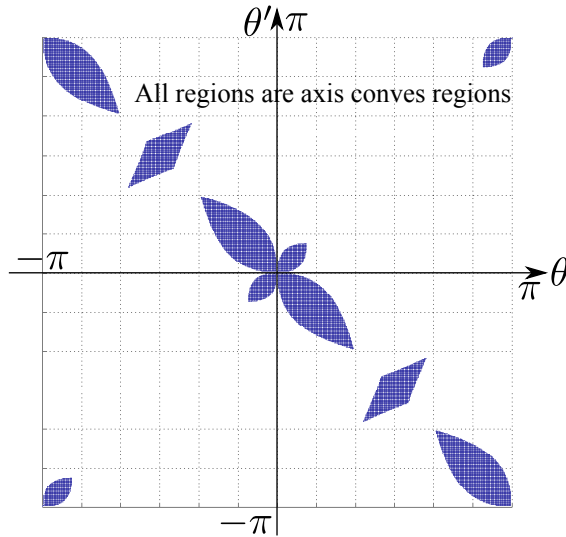
Fig. A.2 Selection of feasible region with  $R_{\min} \leq 0.20d(q, q')$

$R_{\min} = 0.05d(q, q')$  の時の実行可能領域の凸な部分領域を選択して,  $R_{\min} \leq 0.05d(q, q')$  での実行可能領域として利用する (Fig. A.2(a)). なぜなら,  $R_{\min}$  の  $d(q, q')$  に対する割合が大きいほど範囲が狭くなるためである. 同様に,  $0.05d(q, q') < R_{\min} \leq 0.10d(q, q')$  の範囲では, 実行可能領域が最も狭くなる  $R_{\min} = 0.10d(q, q')$  の時の実行可能領域の凸な部分領域を選択して, 実行可能領域として利用する (Fig. A.2(b)). このような選択を  $R_{\min} \leq 0.33d(q, q')$  まで行う (Figs. A.2(c) - A.3(c)). なお, Fig. A.2(d) 以降では元の実行可能領域がすでに凸な領域となる部分が存在する. そういった場合には, それら



(a)  $0.20d(q, q') < R_{\min} \leq 0.25d(q, q')$

(b)  $0.25d(q, q') < R_{\min} \leq 0.30d(q, q')$



(c)  $0.30d(q, q') < R_{\min} \leq 0.33d(q, q')$

Fig. A.3 Selection of feasible region with  $0.20d(q, q') < R_{\min}$

の部分に関しては部分集合を選択せずにそのまま利用し、非凸の部分のみで部分集合の選択を行う。また、 $R_{\min} = 0.33d(q, q')$  以降では元の実行可能領域が凸な領域となるので、 $R_{\min} \leq 0.33d(q, q')$  では  $R_{\min} = 0.33d(q, q')$  の実行可能領域をそのまま利用する。 $R_{\min} > 0.33d(q, q')$  以降に関しては、元の実行可能領域がすでに凸領域の集合であるため、全ての領域をそのまま利用する。

以上の作業により巡回最小半径に関する実行可能領域を準備できる。

# 謝辞

本研究の過程において、終始懇切なる御指導と御鞭撻を賜り、本論文をまとめるに際して、親身な御助言と力強い励ましを頂きました名古屋大学大学院 工学研究科機械理工学専攻 鈴木達也教授には、心より感謝を申し上げます。

本論文の審査過程において、数々の御助言と御指導を賜りました名古屋大学大学院 工学研究科機械理工学専攻 早川義一教授、豊田工業大学 スマートビークル研究センター 三田誠一教授、名古屋大学大学院 工学研究科機械理工学専攻 稲垣伸吉講師に深謝申し上げます。名古屋大学大学院 工学研究科機械理工学専攻 田崎勇一助教には、研究における貴重な御助言を賜りました。同 グリーンモビリティ連携研究センター 奥田裕之特任助教には、様々な観点から有益な御助言を賜りました。名古屋大学国際交流協力推進本部 Blaine Levedahl 特任講師には、英語論文執筆に関する多大な御助言を賜りました。深く御礼申し上げます。

学生生活においては、幸いにも多数の先輩、同期ならびに後輩たちとの出会いに恵まれ、大いなる刺激と笑いを提供してもらいました。諸先生方のご尽力は言うに及ばず、著者と同時期に名古屋大学大学院 工学研究科機械理工学専攻 鈴木研究室に在籍していた全ての方々に御礼申し上げます。特に、本研究を進めるにあたり、諸先輩、同期、後輩ならびに関係者に深く感謝いたします。名古屋大学大学院 工学研究科機械理工学専攻 鈴木研究室卒業生 伊藤隆洋氏ならびに向山将行氏には、研究生生活の準備ならびに研究方針についてご協力頂きました。同 島倉諭氏ならびに山下遼氏には、実機開発ならびに共同研究者としてご協力頂きました。名古屋大学大学院 工学研究科機械理工学専攻 博士課程前期課程 磯部芳朗氏ならびに藤紘至氏には、共同研究者として有益な議論・検討にご協力頂きました。名古屋大学大学院 情報科学研究科複雑系科学専攻 博士課程前期課程 山田貴之氏には、共同研究者ならびにプログラミングに関してご協力頂きました。また、鈴木研究室での修士課程を共に過ごし、切磋琢磨しあった黒川景亮氏、高橋徹氏、竹内秀典氏ならびに丹羽智哉氏には深く感謝いたします。さらには名古屋大学大学院 工学研究科技官の方々には、多大なるご助力賜りましたことを深く感謝いたします。

研究活動費においては、名古屋大学大学院工学研究科グローバル COE プログラム「マ

イクロ・ナノメカトロニクス教育研究拠点」ならびに名古屋大学 博士課程教育リーディングプログラム「フロンティア宇宙開拓リーダー養成プログラム」からの御支援を頂戴しました。関係者に大変感謝しております。

最後に、大学院博士課程へ進学し、6年の長きにわたり研究に専念することが出来たのは、一重に両親の理解と協力の賜物であります。未筆ではありますが、家族に深く感謝し、本論文の締めくくりとさせていただきます。