



人との共存を考慮した自律移動システムに関する研究

小林, 聖人

(Degree)

博士 (工学)

(Date of Degree)

2022-09-25

(Date of Publication)

2024-09-25

(Resource Type)

doctoral thesis

(Report Number)

甲第8474号

(URL)

<https://hdl.handle.net/20.500.14094/0100477900>

※ 当コンテンツは神戸大学の学術成果です。無断複製・不正使用等を禁じます。著作権法で認められている範囲内で、適切にご利用ください。



博士論文

人との共存を考慮した
自律移動システムに関する研究

2022年 7月

神戸大学大学院海事科学研究科

小林 聖人

目次

第1章 序論	1
1.1 研究背景	1
1.2 研究目的	3
1.3 本稿の構成	5
第2章 自律移動システム	6
2.1 自律移動概要	6
2.2 Robot Operating System	7
2.3 Navigation Stack	9
2.4 移動ロボットのモデリング	11
第3章 死角領域を考慮した自律移動	16
3.1 はじめに	16
3.2 ROS Navigation	18
3.3 LRF による死角領域を考慮した自律移動	22
3.4 RGB-D カメラによる死角領域を考慮した自律移動	41
3.5 まとめ	57
第4章 速度制約に基づく動的障害物を考慮した自律移動	58
4.1 はじめに	58
4.2 Dynamic Window Approach (DWA)	60
4.3 Virtual Manipulator (VM)	67
4.4 Dynamic Window Approach with Virtual Manipulators (DWV)	72
4.5 シミュレーション	80
4.6 実験	90
4.7 まとめ	96

第5章 結論	97
5.1 本稿の結論	97
5.2 今後の展望	98
参考文献	99
謝辞	109
業績	111

目次

1.1	研究概要図	3
1.2	論文概要	5
2.1	自律移動概要	6
2.2	ROS 通信イメージ図	7
2.3	Gazebo と Rviz の動作例	8
2.4	ROS Navigation Stack	9
2.5	Global Planner の動作例	10
2.6	Local Planner の動作例	10
2.7	2 輪駆動 2 キャスター型車輪ロボット	11
2.8	座標系	12
2.9	速度関係図	13
2.10	Turtlebot2 の外観	14
2.11	LRF の座標系	15
3.1	ROS Navigation Stack (Default)	18
3.2	階層コストマップ (ROS Default)	19
3.3	Static Layer の概要図	20
3.4	Obstacle Layer の概要図	21
3.5	Inflation Layer の概要図	21
3.6	ROS Navigation Stack (LRF)	22
3.7	死角を考慮した階層コストマップ (LRF)	23
3.8	Blind Spots Layer のフローチャート	24
3.9	センサ値取得 (LRF)	25
3.10	死角境界座標推定 (LRF)	25
3.11	死角境界座標のフローチャート (LRF)	26

3.12	危険領域の中心座標推定	27
3.13	コスト伝播	28
3.14	コストの移り変わり	29
3.15	BSL の動作例 (LRF)	31
3.16	地図作成環境	32
3.17	作成した地図	32
3.18	シミュレーション環境	33
3.19	Case A-1 の動作	36
3.20	Case A-2 の動作	37
3.21	Case B-1 の動作	39
3.22	Case B-2 の動作	40
3.23	ROS Navigation Stack (RGB-D Camera)	41
3.24	死角を考慮した階層コストマップ (RGB-D Camera)	42
3.25	RGB-D カメラによる死角境界座標の推定	43
3.26	Voxel Grid Filter (Step2-1b)	44
3.27	Path Through Filter (Step2-2b)	44
3.28	Euclidean Cluster Extraction (Step2-3b)	45
3.29	Blind Spots Boundary Position (Step2-4b)	45
3.30	コスト生成	46
3.31	BSL の動作例 (RGB-D Camera)	48
3.32	地図作成環境	49
3.33	作成地図	50
3.34	Simulation Environment	51
3.35	シミュレーション結果 (Case C)	54
3.36	シミュレーション結果 (Case D)	55
3.37	コスト生成の様子	56
4.1	DWA フローチャート	60
4.2	ダイナミクスを考慮した速度領域	61
4.3	DWA 経路候補生成	62
4.4	DWA 最適経路選択	66
4.5	反射的動作生成の概念図	67

4.6	仮想マニピュレータを搭載したロボット	69
4.7	仮想補助マニピュレータ生成	71
4.8	DWV フローチャート	72
4.9	DWV 経路候補生成の概念図	73
4.10	DWV 経路候補	74
4.11	DWV 最適な経路選択	79
4.12	SLP の概要図	80
4.13	シミュレーション環境	82
4.14	シミュレーション結果 (<i>Case S1</i>)	85
4.15	シミュレーション結果 (<i>Case S2</i>)	86
4.16	シミュレーション結果 (<i>Case S3</i>)	87
4.17	シミュレーション結果 (<i>Case S4</i>)	88
4.18	Experiment Setup	90
4.19	実験システム概要図	90
4.20	実験環境 (<i>Case E1</i>)	91
4.21	実験環境 (<i>Case E2</i>)	92
4.22	実験結果 (<i>Case E1</i>)	94
4.23	実験結果 (<i>Case E2</i>)	95

表 目 次

2.1	TurtleBot2 の主な仕様	14
2.2	レーザ式測域センサの主な仕様	15
3.1	シミュレーション結果 (Case A)	35
3.2	シミュレーション結果 (Case B)	38
3.3	シミュレーション方式	52
3.4	パラメータ	52
3.5	シミュレーション結果 (Case C)	54
3.6	シミュレーション結果 (Case D)	55
4.1	パラメータ	81
4.2	シミュレーション条件	82
4.3	シミュレーション結果	84

第 1 章 序論

1.1 研究背景

我が国は超高齢社会であり、内閣府が公開している高齢化社会白書によれば 2021 年 9 月 15 日時点で、総人口に占める高齢者人口の割合は 29.1%となっており、過去最高の値を示している [1]。超高齢社会は様々な問題を引き起こし、その問題の一つが労働力の欠如である。それゆえ、人を支援または人の代用となるロボットに関する多くの研究が報告されており、ロボットによる労働力の創出が期待されている [2,3]。多くのロボットは工場等の定められた空間内で動作し、現在も人間の代替えとして労働力を創出している。工場等で稼働する産業用ロボットの多くは、安全柵等の囲まれた空間で動作する。動作方法の多くはティーチング・プレイバック方式(教示再生方式)という、人間があらかじめ教示したロボット動作をロボットが再生することで動作する [4,5]。しかしながら、ティーチング等の教示作業は膨大な時間と労力を要するため、人の教示無しに自動で動作生成を行う技術も導入されている [6,7]。また人協調ロボットも開発され、安全柵等のない開かれた環境で人とロボットが協調して動作することが可能となっている [8-10]。ハードウェアやソフトウェア技術の進歩により、工場だけでなく病院などの施設や商業施設等の人々が暮らす生活空間において動作するロボットも導入されている [11-14]。現在、人が暮らす生活空間においてロボットが導入されている例を以下に示す。

- 物流 例：配達ロボット
- 公共空間 例：清掃ロボット
- 医療 例：人支援ロボット
- 家庭 例：清掃ロボット
- 飲食店 例：配膳ロボット

これらのロボットは移動 [15–17]、認識 [18,19]、操作 [20,21]、物体保持 [22,23] 等の能力を必要とする。物流の現場では、人が物を運ぶ代わりにロボットが物品の運搬を行う [24]。医療の現場では、人の代わりにロボットが検査や消毒等の動作を行う [25]。公共空間や家庭の現場では、人が清掃する代わりにロボットが床等の清掃を行う [26]。飲食店では、ロボットは人等が存在する環境において食品の運搬や下膳等を行う [27]。ここで、上記のロボットの移動の要素に注目すると、多くのロボットは車輪を用いて移動している。今後よりいっそう車両型移動ロボットが使用され、人との共存を考慮した移動手法の研究開発が活発に進んでいくと考えられる。

前述の観点から、車輪型移動ロボットに関する多くの研究が報告されている [28,29]。移動ロボットの移動手法は2つのタイプに大別される。1つは遠隔移動手法で、もう1つは自律移動手法である。遠隔移動手法においては、ロボットは操作者の判断を用いて作業を行う [31,32]。一方、自律移動手法においては、ロボットは複数のセンサ情報を用いて作業を行う [33,34]。自律移動手法はオペレータが必要でない事が利点である。本稿では、労働力の創出の観点から自律移動手法を中心に述べる。

人等が混在する生活空間においてロボットと人の共存を考慮した自律移動の実現には、人の行動を妨げず、人に危害を与えないロボット動作が求められる。しかしながら、人が混在するような時々刻々と変化する環境下で自律移動を実現するのは困難である [35]。自律移動ではロボットはセンサ等から取得した環境情報をもとに、どのように行動するか決定する。ロボットに搭載できるセンサには限りがあり、またセンサが取得できない死角領域も生じる [36]。死角領域とは障害物の背後や廊下等での曲がり角の直前等、センサでは計測できない領域のことである。この死角領域が原因によりロボットと人が衝突する可能性が高くなる。また人が混在する環境下では動的障害物が存在するために、時々刻々と変化する環境下においてロボットは自律移動を実現しなければならず、ロボットが障害物と衝突せずに目的地までたどり着くのは未だ難しい問題である。

1.2 研究目的

本稿では、研究背景から人が生活する空間での自律移動について着目し、「死角領域からの人の飛び出しが衝突の可能性を高めること」、「人等の動的障害物が混在するような時々刻々と変化する環境下ではロボットと障害物との衝突の可能性が高いこと」の2点を課題としてとりあげる。そこで本稿では、図 1.1 に示すように、それぞれ2つの課題に対して以下の提案を行う。

- 死角領域を考慮した自律移動手法の提案
- 速度制約に基づく動的障害物を考慮した自律移動手法の提案

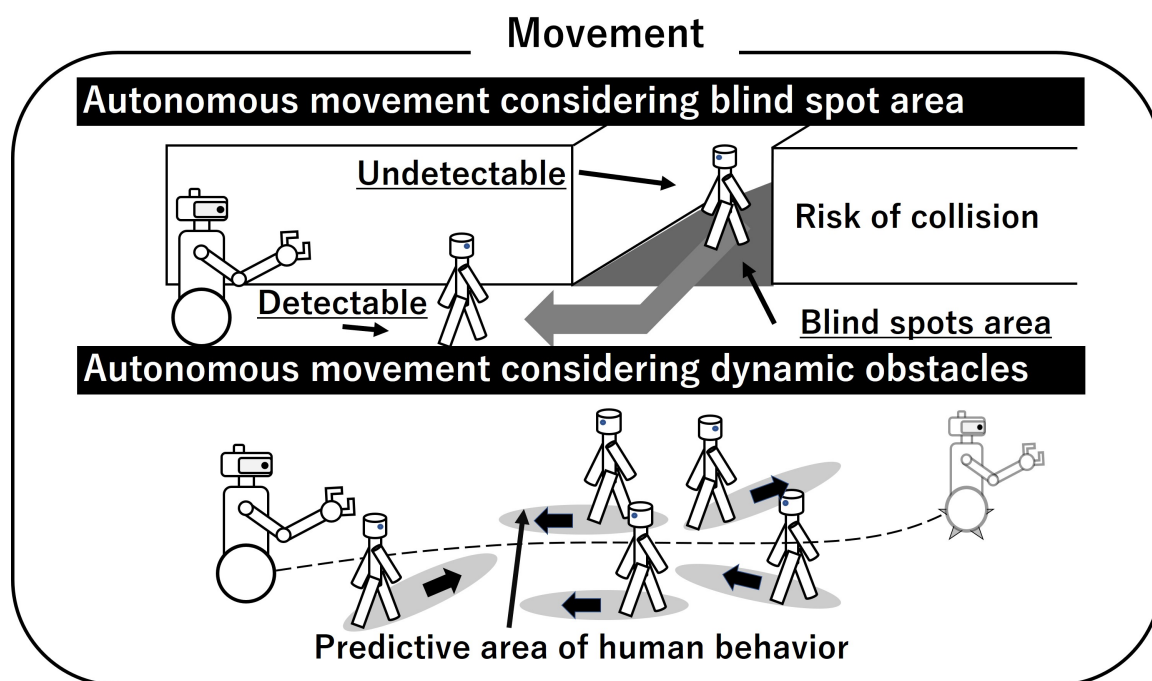


図 1.1 研究概要図

まず1つ目の「死角領域を考慮した自律移動手法」では、リアルタイムでの死角領域の危険度の表現と危険度を考慮した経路計画の実現が重要である。死角領域を考慮する上で問題なのは、死角領域に潜む障害物はセンサで検出できないことである。つまり空間内にセンサや補助機器を設置しない限り、死角領域に人等の動的障害物等が存在するかは未知である。そこで本稿では、LRF や RGB-D カメラにより取得した環境情報を用いて死角領域を考慮し

た自律移動を行う。取得した環境情報から死角領域において、人の位置を推定する。推定した位置から人が全方向移動すると仮定して円状に危険度を表すコストをコストマップに伝播する。その後、ロボットの運動性能を考慮した局所的経路計画手法である Dynamic Window Approach (DWA) において危険領域のコストマップを考慮し、死角領域とロボットの運動性能を考慮した経路計画がリアルタイムで可能となる。本稿ではシミュレーションにより提案手法の有効性を評価する。

次に2つ目の「速度制約に基づく動的障害物を考慮した自律移動手法」では、ロボットが実現可能な静的・動的障害物と衝突しない経路の生成と選択を行う経路計画の実現が重要である。そこで本稿では、ロボットの実現可能な速度制約に基づく仮想マニピュレータと DWA に基づく局所経路探索手法である Dynamic Window Approach with Virtual Manipulators (DWV) を提案する。DWV は、仮想マニピュレータと静的・動的障害物の予測位置によって修正された可変速度によって経路候補を生成する。そのため静的・動的な障害物が存在する環境において障害物回避可能な経路を生成する。本稿では提案した手法をシミュレーションおよび実験により提案手法の有効性を評価する。

1.3 本稿の構成

図 1.2 に本稿の構成を示す。第 1 章では人との共存を考慮した自律移動手法に関する研究背景と研究目的を述べる。第 2 章では自律移動の概要や、Robot Operating System で用いられている自律移動システムの Navigation Stack、本稿で使用したロボットのモデリングについて紹介する。第 3 章では死角領域からの人の飛び出しによる衝突を防ぐために、リアルタイムで死角領域を考慮した自律移動手法について述べる。第 4 章では人が混在する環境下での自律移動の実現を目指し、速度制約に基づく静的・動的障害物を考慮した自律移動手法について述べる。第 5 章にて本稿の結びとする。

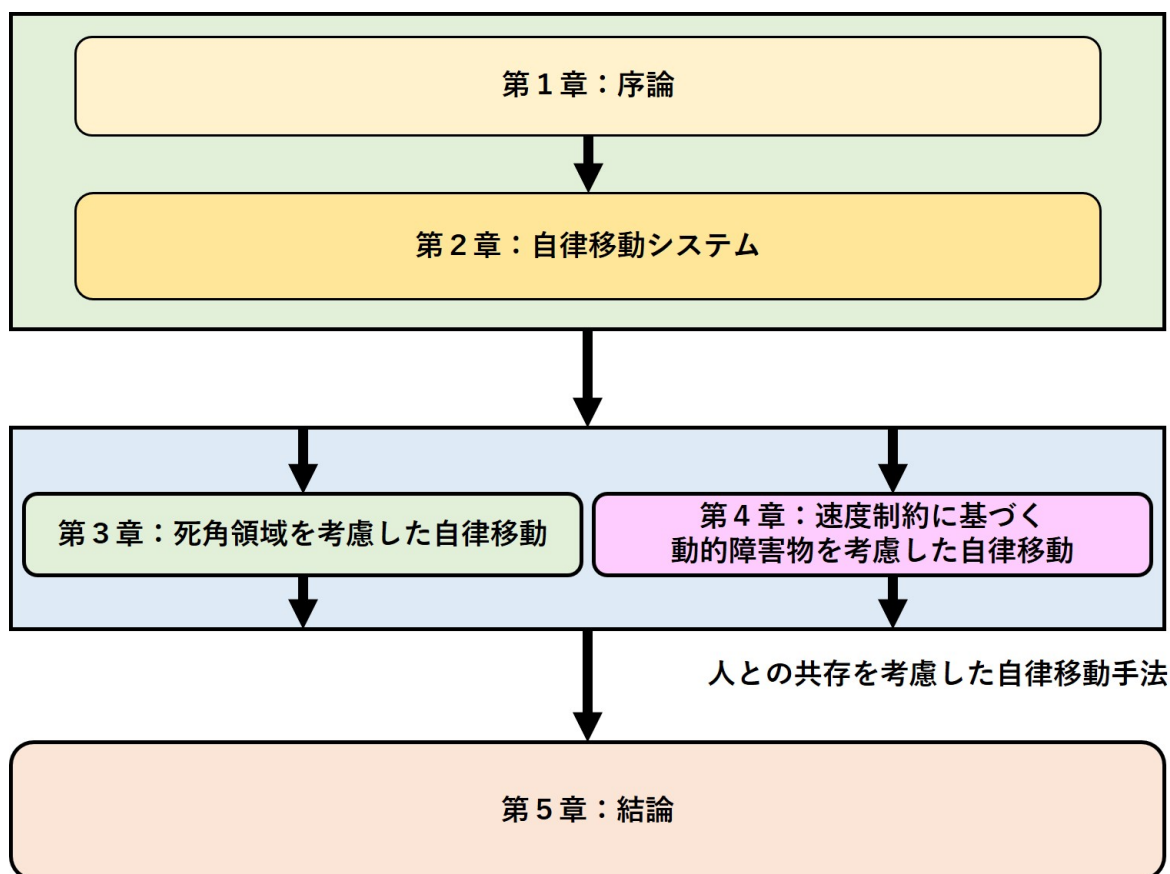


図 1.2 論文概要

第2章 自律移動システム

2.1 自律移動概要

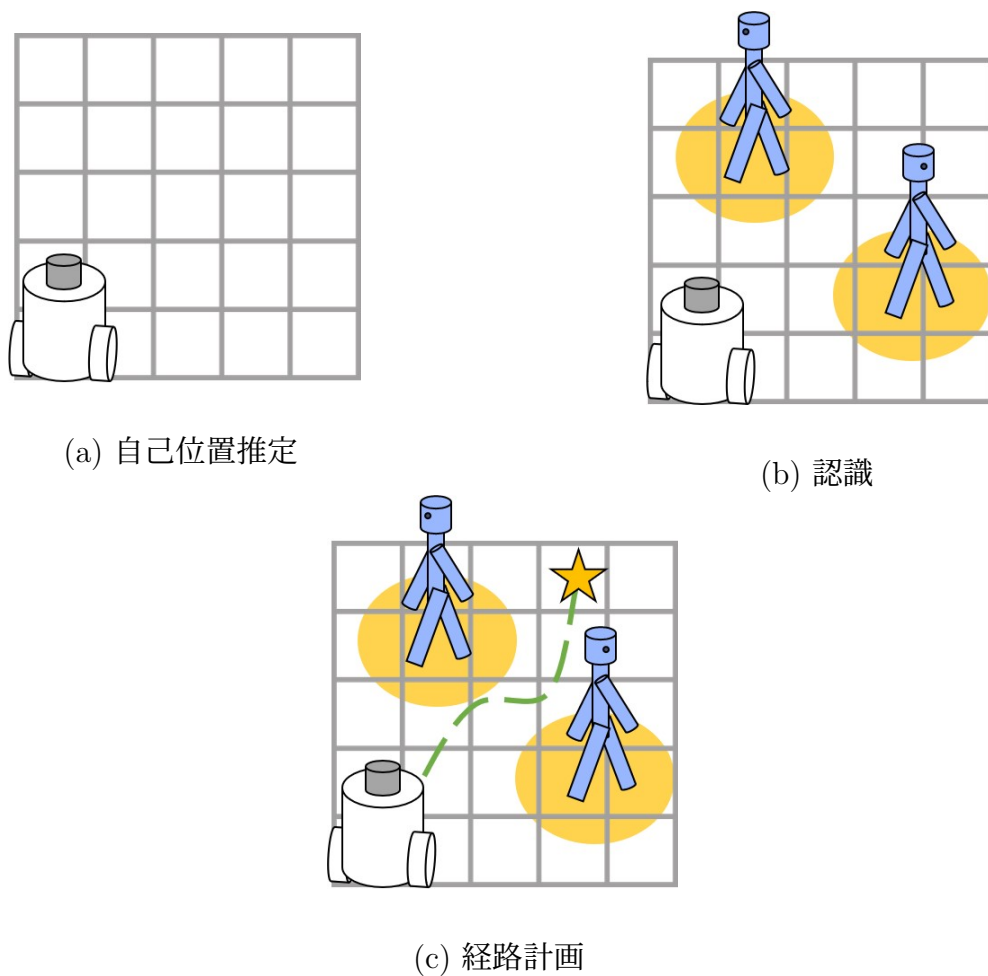


図 2.1 自律移動概要

自律移動の実現には、自己位置推定、地図生成、認識、経路計画等が必要となる。図 2.1 に自律移動における自己位置推定、認識、経路計画の概要図を示す。図 2.1(a) のように自己位置推定では、測域センサや車輪のエンコーダ情報等から地図上でのロボットの位置を

推定する。図 2.1(b) のように認識では、センサ等を用いてロボット周囲の環境情報を計測する。図 2.1(c) のように経路計画では、取得した環境情報とロボットの自己位置等の情報をもとに、障害物との衝突を回避する経路計画を行う。これらの処理を実時間で行うことで自律移動が実現されている。

2.2 Robot Operating System

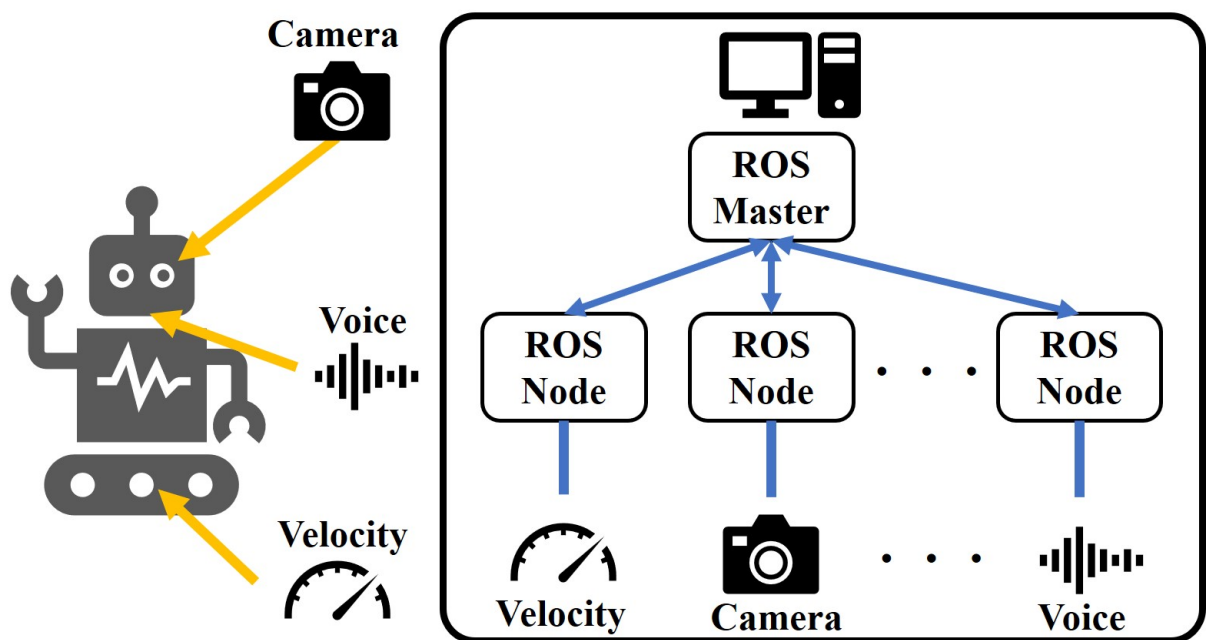


図 2.2 ROS 通信イメージ図

Robot Operating System (ROS) はオープンソースのロボットソフトウェアである。ROS には ROS1 や ROS2 が存在するが、本稿では ROS1 について述べる。ROS は機器の制御、ソフトウェアの実装、パッケージ管理、プロセス間のメッセージの受け渡し等を提供する。ROS はオープンソースのソフトウェアであり、世界的にユーザーも多くコミュニティ間の議論が活発に行われている。ROS ではノードと呼ばれる最小単位のプロセスを組み合わせることで、開発したソフトウェアの機能が実現可能となる。それぞれのノードは独立して並列に動作し、双方向のデータ送受信が可能である。それゆえ開発したいノードのソースコードを改良することによって、効率的に動作変更が可能となる。図 2.2 に ROS のデータ通信のイメー

図 2.2 を示す。図 2.2 に示すようにロボットにはカメラ等の視覚センサ、ロボットの動作を担うモータ等が搭載されている。それらセンサ情報等の処理を ROS ノードを組み合わせることで、様々なロボットアプリケーションの実現が可能となる。

ROS にはコミュニティや通信等の利便性に加えて、物理シミュレーターの Gazebo や可視化ツールの Rviz があり、シミュレーションでロボットソフトウェアを検証することが可能である。またシミュレーションで使ったソフトウェアと同等のソフトウェアで、実機のロボットを動作させることが可能であり、効率的に開発が可能である。図 2.3 にロボットにカメラと測域センサを搭載した時の Gazebo と Rviz の様子を示した。このように測域センサやカメラ等の様々なセンサにも対応しており、様々な機器を使用した開発を行うことが可能である。

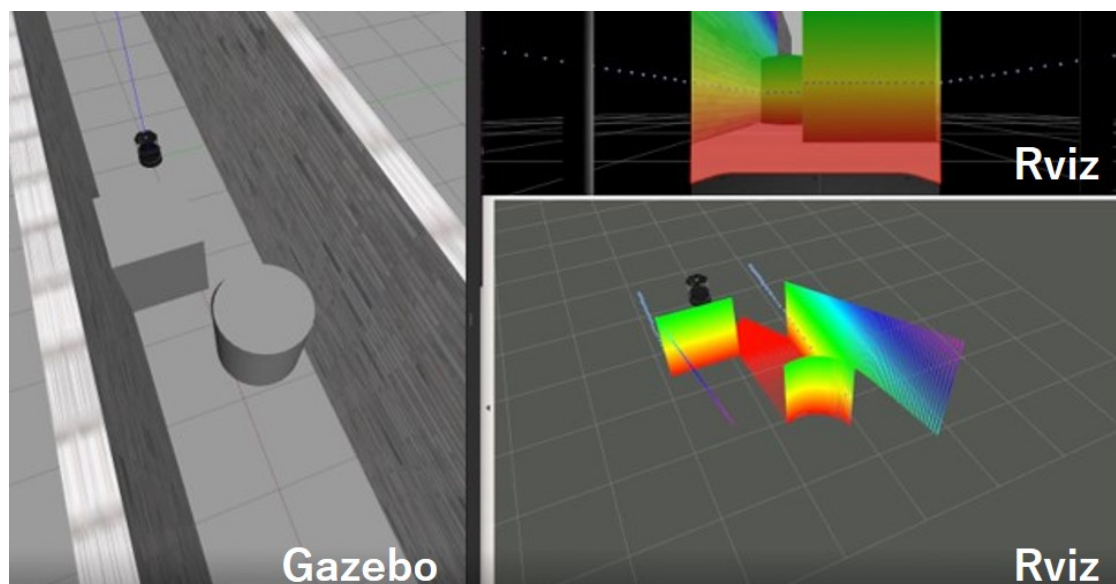


図 2.3 Gazebo と Rviz の動作例

2.3 Navigation Stack

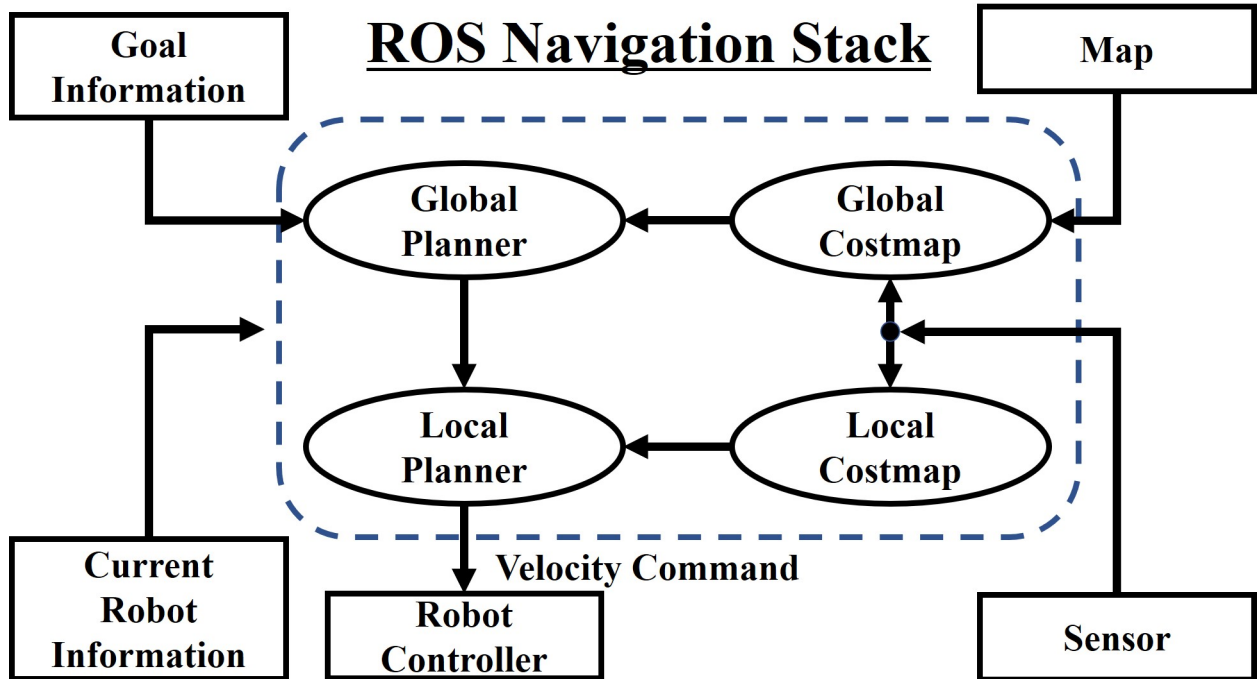


図 2.4 ROS Navigation Stack

Navigation Stack は ROS を用いた自律移動システムである。図 2.4 に ROS Navigation Stack のイメージ図を示す。Navigation Stack ではモータのエンコーダーの情報や測域センサ等から推定した自己位置や速度情報をもとに、指定された目的地までロボットと環境との衝突回避を考慮しながら経路計画を行うことで自律移動の実現が可能である。図 2.4 が示すように、ROS Navigation Stack では Global Planner と Local Planner の 2 種類の経路計画により構成されている。図 2.5 に Global Planner、図 2.6 に Local Planner の動作例を示す。Global Planner ではロボットの現在の位置から目的地まで大まかな経路（緑色）を事前に作成した地図情報を用いて生成する。Local Planner では Global Planner で生成した緑色の経路を追従するとともにセンサ等でリアルタイムで環境情報をローカルコストマップに反映し、Global Planner では対応できない障害物に対して経路計画を行う。図 2.6 が示すように、黄色の扇状で示されている経路が Local Planner で生成された経路であり、青色で示されているローカルコストマップを避けるようにロボットは行動する。

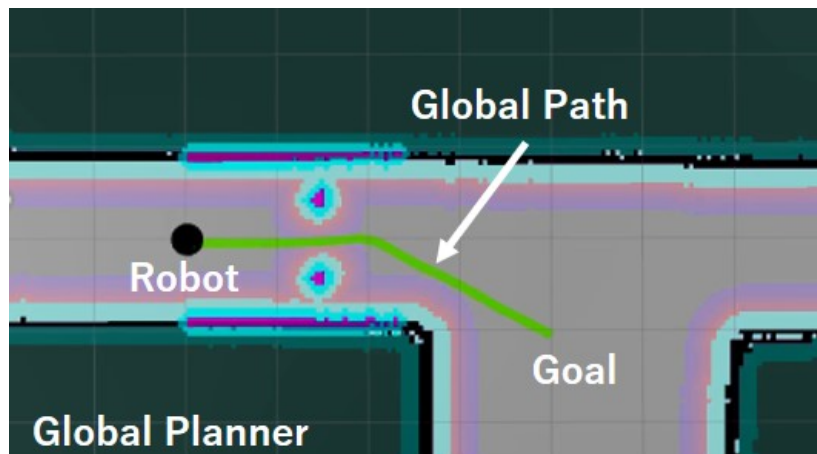


図 2.5 Global Planner の動作例

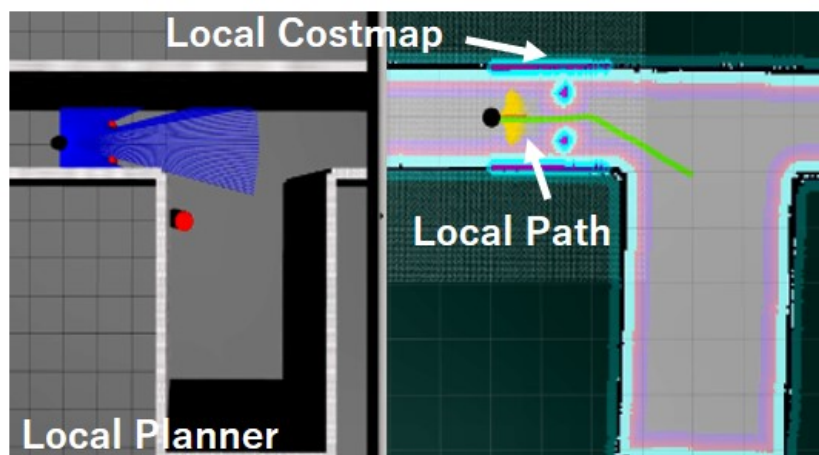


図 2.6 Local Planner の動作例

2.4 移動ロボットのモデリング

本節では、本研究でを使用した自律走行ロボット開発用プラットフォーム TurtleBot2 について説明する。本研究で使用する TurtleBot2 は対向 2 輪型と呼ばれる形態で、前後にキャスターがついた 4 点接触型の移動機構を有している。図 2.7 に 2 輪駆動 2 キャスター型車輪ロボットを上から見た図を示す。

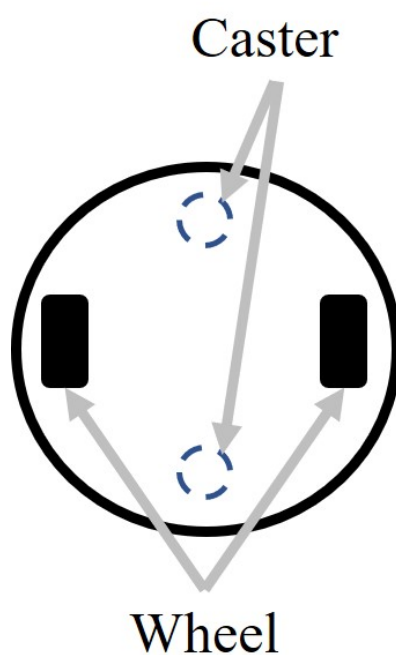


図 2.7 2 輪駆動 2 キャスター型車輪ロボット

2.4.1 座標系定義

図 2.8 に本論文で扱う座標系を定義する。二つの車輪の中心点をロボット座標系の原点としロボット座標系の X 軸をロボットの進行方向に合わせる。 X 軸に垂直な軸を Y 軸と定める。ロボット座標の原点をグローバル座標系におけるロボットの位置 (x, y) とする。またグローバル座標系の X 軸とロボット座標系の X 軸のなす角をロボットの姿勢角 θ と定義する。変数の左上添字は、座標系を表すこととする。例えば、 ^{GB}x はグローバル座標系における x 座標のことである。左上添字が付かない場合は、ロボット座標系における変数とする。

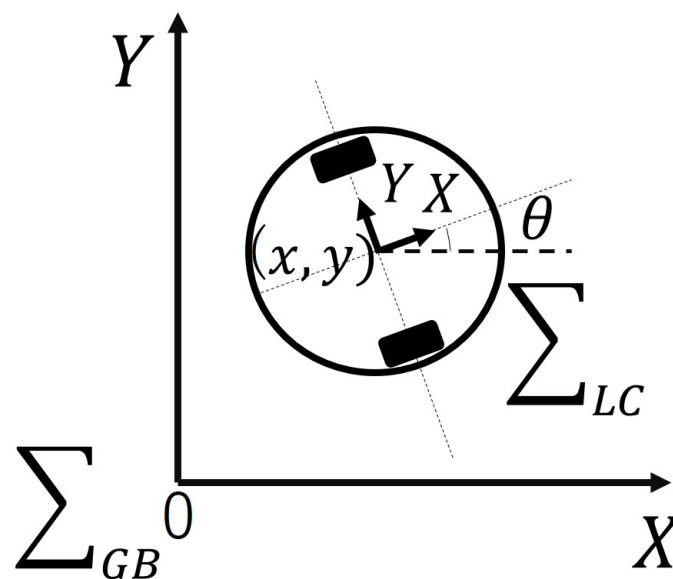


図 2.8 座標系

2.4.2 移動ロボット運動学

図 2.9 に移動ロボットの速度関係図を示す。ロボットの操作量は一般に車輪の回転速度である。左右各々の車輪の回転角度を ϕ_r 、 ϕ_l とすると、左右の車輪の接地点での速度 v_r 、 v_l は車輪の半径を μ として次のように算出される。

$$v_r = \mu \dot{\phi}_r \quad (2.1)$$

$$v_l = \mu \dot{\phi}_l \quad (2.2)$$

ここで、旋回中心回りの運動を考える。図 2.9 のように、ロボットの並進速度を v 、ロボットの旋回速度を ω 、旋回半径を ρ とすると、これらの関係式は次のように示される。

$$v = \rho\omega \quad (2.3)$$

一方、ロボット中心から車輪までの距離を B とすると、それぞれの車輪の旋回半径は B だけ増減するため、左右の車輪の接地点での速度は次のように算出される。

$$v_r = (\rho + B)\omega \quad (2.4)$$

$$v_l = (\rho - B)\omega \quad (2.5)$$

速度と旋回半径の関係は次のように示される。

$$\omega = (v_r - v_l)/2B \quad (2.6)$$

$$v = (v_r + v_l)/2 \quad (2.7)$$

$$\rho = B(v_r + v_l)/(v_r - v_l) \quad (2.8)$$

このとき、ロボットの位置や角度は以下のように算出される。

$$\dot{x} = v \cos \theta \quad (2.9)$$

$$\dot{y} = v \sin \theta \quad (2.10)$$

$$\dot{\theta} = \omega \quad (2.11)$$

以上より、ロボットの運動が両輪の回転角速度をもとに記述できることが確認できる。

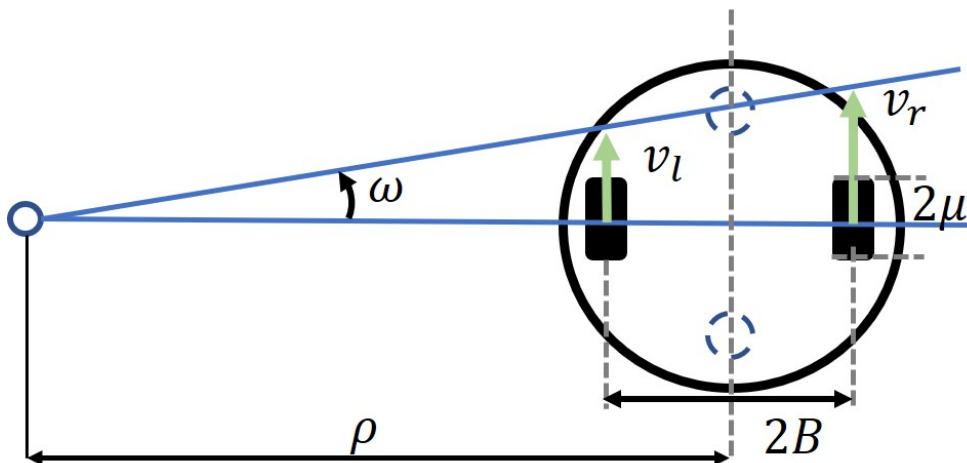


図 2.9 速度関係図

2.4.3 TurtleBot2

図 2.10 に TurtleBot2 の外観、表 2.1 に Turtlebot2 の仕様を示す。TurtleBot2 は、オープンソースソフトウェアを搭載したパーソナルロボットキットであり、ROS を利用した動作が可能である。ソフトウェアのバグ等の不具合時には ROS のコミュニティを利用することにより、ユーザーと意見交換することが可能である。

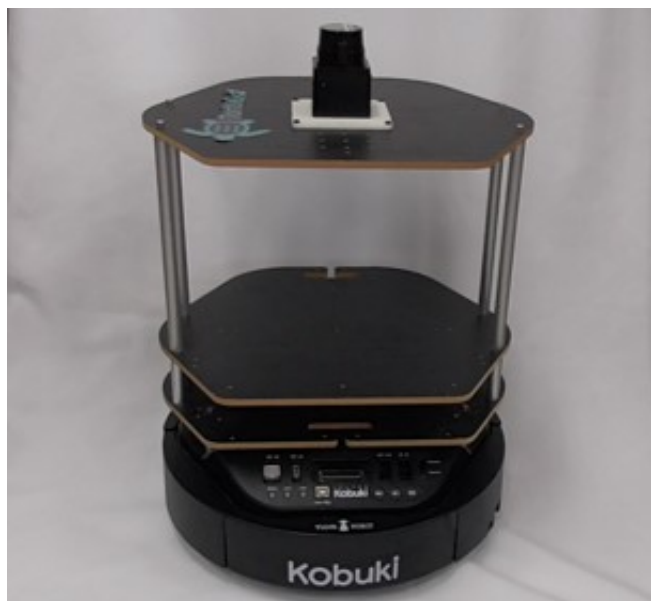


図 2.10 Turtlebot2 の外観

表 2.1 TurtleBot2 の主な仕様

名称	TurtleBot2
外形寸法	354 × 354 × 420 mm
重量	6.3 kg
最高速度	0.65 m/s
最大旋回速度	180 °/s
バッテリー	2200mAh リチウムイオン電池

2.4.4 センサー座標系

本稿で使用したレーザ式測域センサである Laser Range Finder (LRF) は URG-04LX-UG01 であり、内部にあるミラーを回転させることでレーザ光を走査し二次元平面を認識する。表 2.2 に LRF の仕様を示す。図 2.11 に LRF の座標系を図示する。本稿では、LRF の座標系を図 2.8 のロボット座標系と同様とし、ロボットの進行方向を X 軸、ロボットの進行方向左側を Y 軸と定める。LRF は反時計回りに探索し、ロボット座標系で極座標 $[Z, \theta]$ として取得し、 $P_i = [Z_i, \theta_i] (i = 1, 2, \dots, K)$ とする。

表 2.2 レーザ式測域センサの主な仕様

名称	レーザ式測域センサ
外形寸法	70 × 50 × 50 mm
測域範囲	距離:0.02~5.6 m 角度:240 度
光源	半導体レーザ $\lambda=785$ nm
角度分解能	約 0.36 度

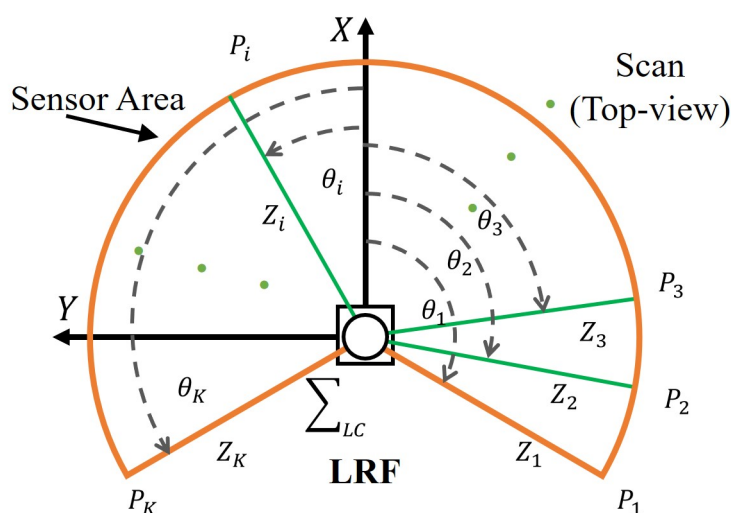


図 2.11 LRF の座標系

第 3 章 死角領域を考慮した自律移動

3.1 はじめに

近年、労働力の創出の観点から工場、飲食店、病院、空港等においてロボットが人のサポートや代替えとして導入されている [37-41]。また 2035 年に向けたロボット産業市場予測では、サービスロボットの需要拡大が予測されている [42]。サービス業界で動作するロボットにおいて、人との共存を考慮した場合、ロボットの移動が重要な要素となる。特に本章では自律移動について述べる。

人等が混在する生活空間においてロボットと人の共存を考慮した自律移動を実現には、人の行動を妨げず、人に危害を与えないロボット動作が求められる。本章では、死角領域を考慮した自律移動手法について着目する。死角領域とは障害物の背後や廊下等での曲がり角の直前等、センサでは計測できない領域のことである。この死角領域が原因によりロボットと人が衝突する可能性が高くなる。上述の問題を解決するために、従来より死角領域を考慮した自律移動手法が提案されており、「環境設備依存型」と「非環境設備依存型」に大別して述べる。環境設備依存型では、カーブミラーや反射鏡等の環境も設置された設備を利用することで死角領域内の状況予測を行う。内海等は、死角領域に潜む人物の位置をカーブミラーを用いる事で推定する [46]。田村等は、人の行動を予測するデータベースを環境に設置されたセンサから生成し、死角領域を考慮した移動手法を報告した [47]。上述の手法は大変有用な手法であるが、事前に反射鏡等を設置する必要があり、環境により利用可能な場面が限定される。本章では環境の設備に依存しない死角領域を考慮した自律移動手法の実現を目指し、非環境設備依存型の手法について述べる。

非環境設備依存型ではロボットに搭載されているセンサ等の情報のみから死角領域を考慮する手法である。Woojin 等は事前に作成した地図から死角となる領域を予め定め、死角領域付近での走行において、安全な速度変化を行う手法を報告した [48]。しかしながら、事前に作成した地図上に存在しない障害物により生じる死角領域には対応できないという問題がある。関等は、センサ毎に定められているセンサの可視領域を利用し、「見晴らし」という測

量でセンサの可視領域範囲が多い経路を通るように大域的な経路計画のダイクストラ法を用いて死角領域を考慮した経路計画を提案した [49]。しかし、事前に作成された地図を前提としており、環境が事前の地図から変化している場合、障害物や死角領域を考慮しない経路が生成されることがある。萬等は事前地図上に無い障害物が発生する等、周辺環境変化に合わせて速度修正が行えるように Laser Range Finder (LRF) と事前に作成した地図から得た死角領域と可視領域の面積割合から危険度を求め、リアルタイムで速度を変化させる手法を報告した [50]。しかしながら、ロボットの加減速性能が加味されておらず、算出された速度が安全であるかの議論はされていない。黒坂等はセンサ情報と事前に作成した地図を利用し、リアルタイムでロボットの速度制御を可能とした [51]。速度の安全性を考慮した手法であったが、事前地図をもとに計画された移動経路に基づくため、計画された移動経路上に物体等の障害物が置かれた場合、ゴールにたどり着くことができないことがある。刈谷等は、事前に作成した地図とロボットに設置されているセンサから取得した情報から作成した局所的な地図の差分により死角領域を算出し、算出した領域に対しポテンシャル法の斥力を発生させることで経路生成を行った [52]。しかしながら、ロボットの運動制約を考慮して生成した経路では無いために、生成された経路通りにロボットが経路追従できるとは限らない。このように従来より、事前地図から死角領域を特定し大域的な経路計画を行う方法や、死角領域を考慮したリアルタイムなロボット速度制御が報告されている。しかし、あらかじめ計画された経路上しかロボットが移動できないことや、ロボットの運動制約を考慮できないといった問題があった。

そこで本章では、上述の問題を解決するために、リアルタイムに死角領域とロボットの運動制約を考慮する経路計画手法の実現を目指した「車輪型移動ロボットにおける死角領域を考慮したコストマップに基づく局所的経路計画手法」について述べる [53, 54]。提案したシステムは ROS Navigation Stack を基盤のシステムとして採用しており、死角領域に潜む人間を推定する認識部と死角領域を考慮した経路計画部をそれぞれ並列動作させることで実現した。死角領域を考慮した認識部では初めに、LRF や RGB-D カメラから取得した環境情報から死角境界座標を算出し、死角領域において人が存在する可能性がある位置を推定する。推定位置を危険領域の中心座標とし、人が全方向に移動することを想定して、コストをコストマップに円形に伝播させることで危険領域を表現する。経路計画部においてロボットの運動制約を考慮した局所経路計画法である DWA (Dynamic Window Approach) の経路評価時に危険領域のコストを考慮することで、死角領域とロボットの運動制約を考慮した経路計画をリアルタイムに実行することができる。シミュレーションにより、提案手法の有用性を確認した。

3.2 ROS Navigation

図 3.1 に自律移動システムの基盤となる ROS Navigation Stack のデフォルトの構成を示す。本章ではデフォルトの構成から Local Planner (Dynamic Window Approach) と Local Costmap (Master Costmap) を変更するため、本節では Local Planner と Local Costmap について紹介する。詳解については次項で述べる。

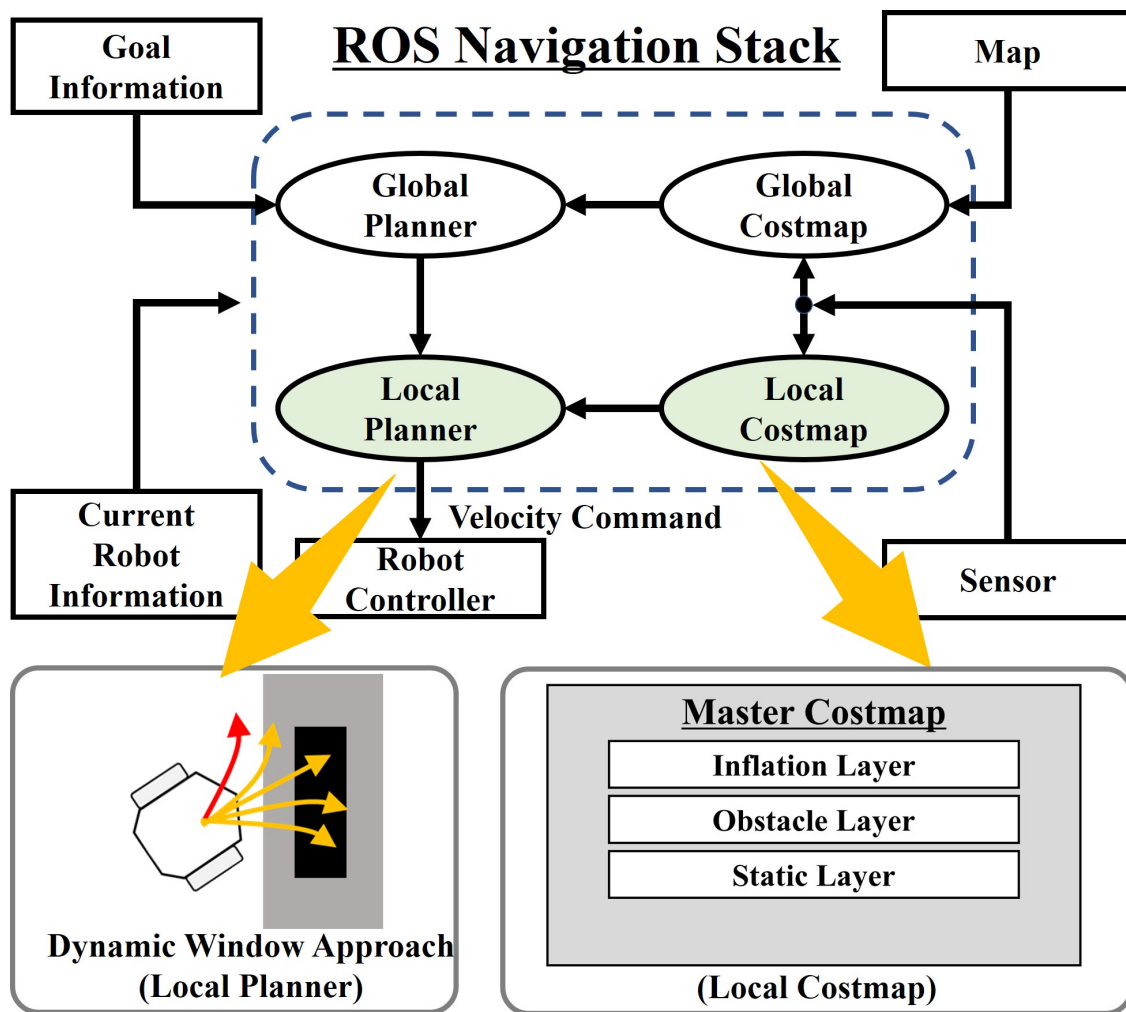


図 3.1 ROS Navigation Stack (Default)

3.2.1 Dynamic Window Approach

Dynamic Window Approach (DWA) とは、ロボットの運動性能を考慮した局所的経路計画の一つである。はじめに DWA では、ロボットの最大・最小速度・加速度を用いて動作可能範囲である速度領域を算出する。次に速度領域内から選択した並進・旋回速度と予測時間 T^{pre} を用いて予測経路を生成する。予測経路は各並進・旋回速度における等速運動を仮定することで、 T^{pre} 後までの予測経路候補を算出する [45]。評価関数を用いて算出した経路候補を評価し、最も評価関数を満たす経路を生成した並進・旋回速度を速度指令値として採用する。

Robot Operating System (ROS) の Navigation システムで用いられている DWA の評価関数を以下に記す [55]。ROS では評価関数の値が最小となる経路を選択する。

$$c^{all} = W^p \cdot c^p + W^g \cdot c^g + W^o \cdot c^o \quad (3.1)$$

ここで、 W^p はグローバルパスへの追従の重み、 c^p はローカルパス端点からグローバルパスまでの距離 [m]、 W^g はゴールに近づく重み、 c^g はローカルパス端点からゴールまでの距離 [m]、 W^o はコストマップ上の障害物を考慮した重み、 c^o はローカルパス上での障害物を考慮した最大マップコストである。

3.2.2 コストマップ

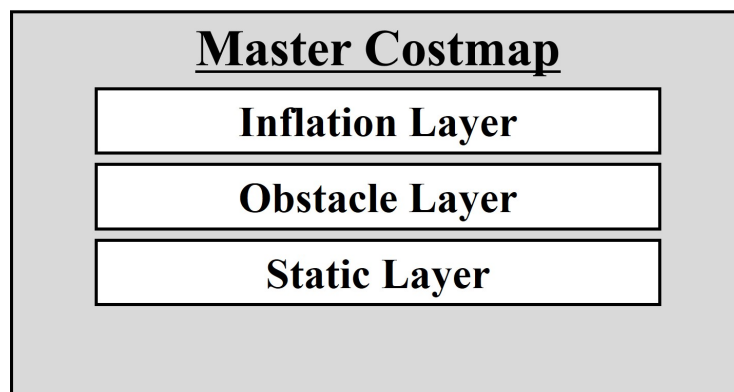


図 3.2 階層コストマップ (ROS Default)

図 3.2 に ROS の Navigation システムの階層コストマップを示す [56]。DWA では障害物を考慮したコストマップを用いて、ロボットと障害物の衝突回避が可能となる。コストマップ

は格子状に分割されており、センサ情報に基づいて、「自由」、「未知」、「占有」の 3 状態のいずれかである。

- 自由：障害物が存在しない安全な空間。
- 未知：障害物の存在が不明な空間。
- 占有：障害物が存在する空間。

階層コストマップには標準で 3 つの層が設定されており、Static Layer、Obstacle Layer、Inflation Layer の各処理が行われる。各階層は、マスターコストマップに統合される。マスターコストマップのコストと (3.1) 式を用いて、リアルタイムで経路計画を行う。

Static Layer

図 3.3 に Static Layer の概要図を示す。図 3.3(a) が示すように、LRF 等を用いて Simultaneous Localization and Mapping (SLAM) [57] で事前地図を生成する。図 3.3(b) が示すように、Static Layer では生成した地図情報を格納する。

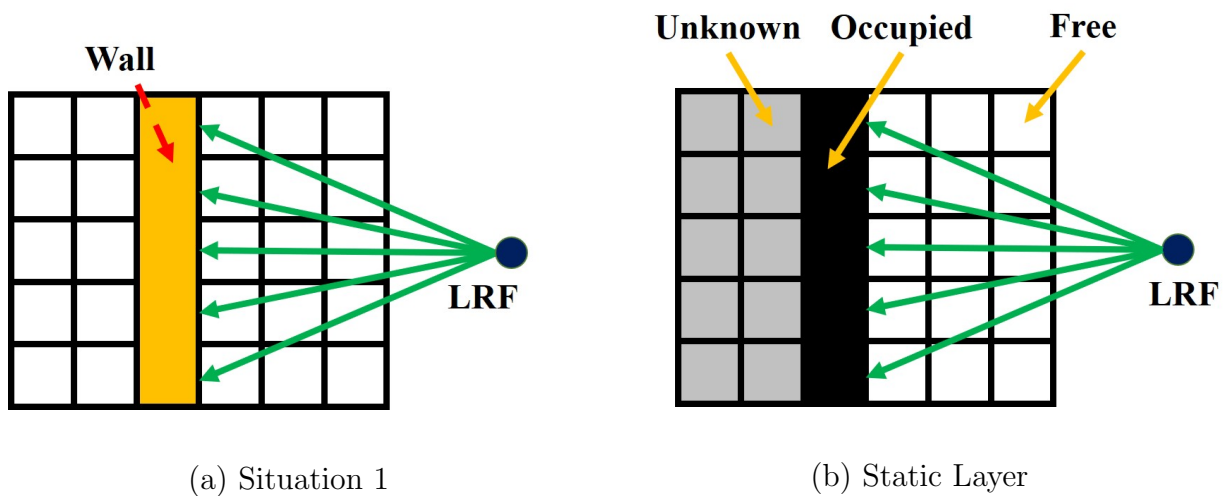


図 3.3 Static Layer の概要図

Obstacle Layer

図 3.4 に Obstacle Layer の概要図を示す。図 3.4(a) が示すように、測域センサである LRF を用いて環境情報を取得する。図 3.4(b) が示すように、Obstacle Layer ではセンサで取得した環境情報を格納する。

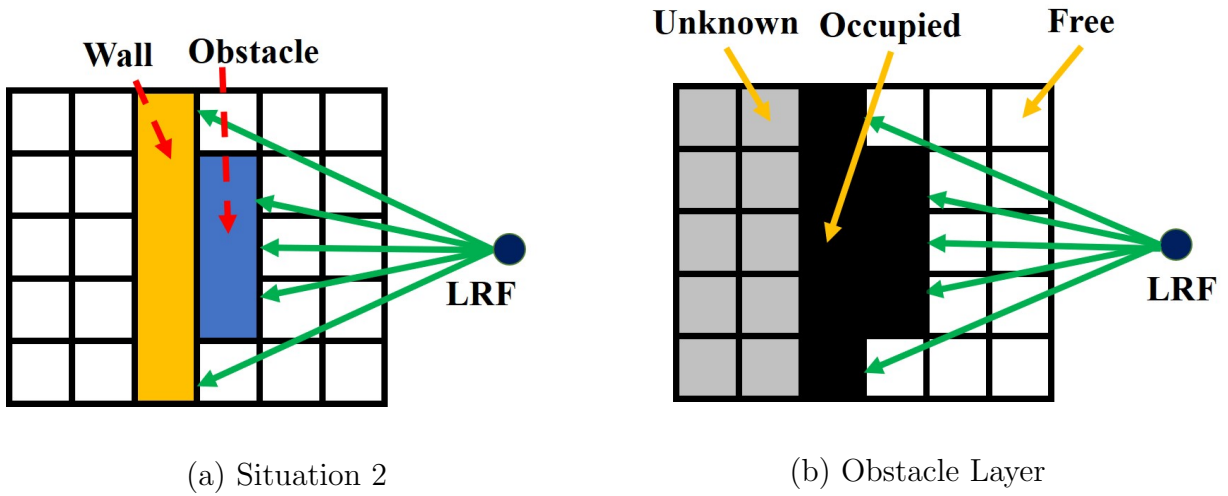


図 3.4 Obstacle Layer の概要図

Inflation Layer

図 3.5 に Inflation Layer の概要図を示す。図 3.5(a) が示すように、測域センサである LRF を用いて環境情報を取得する。図 3.5(b) が示すように、Obstacle Layer で格納した障害物情報を用いて、障害物近辺のコストを膨張させ、障害物とロボットの間に安全な距離を確保するコストを格納する。

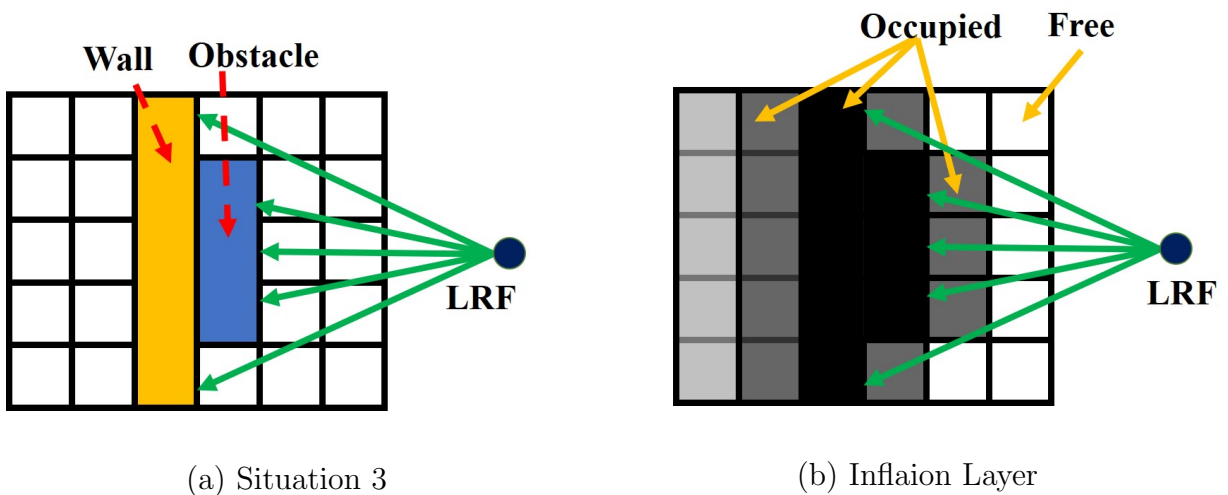


図 3.5 Inflation Layer の概要図

3.3 LRF による死角領域を考慮した自律移動

従来の死角領域を考慮した手法では、ロボットの運動制約を考慮しない経路計画と事前に計画された経路上のみしか移動できない問題があった。そのため死角領域とロボットの運動制約を考慮した経路計画が希求されている。本節では自律移動システムの基盤となる構成は ROS Navigation Stack を採用し、リアルタイムで死角領域を考慮した Local Costmap を提案する。死角領域を考慮したコストを用いて、運動制約を考慮した Local Planner である DWA の評価関数に含めることで上述の問題を解決する。

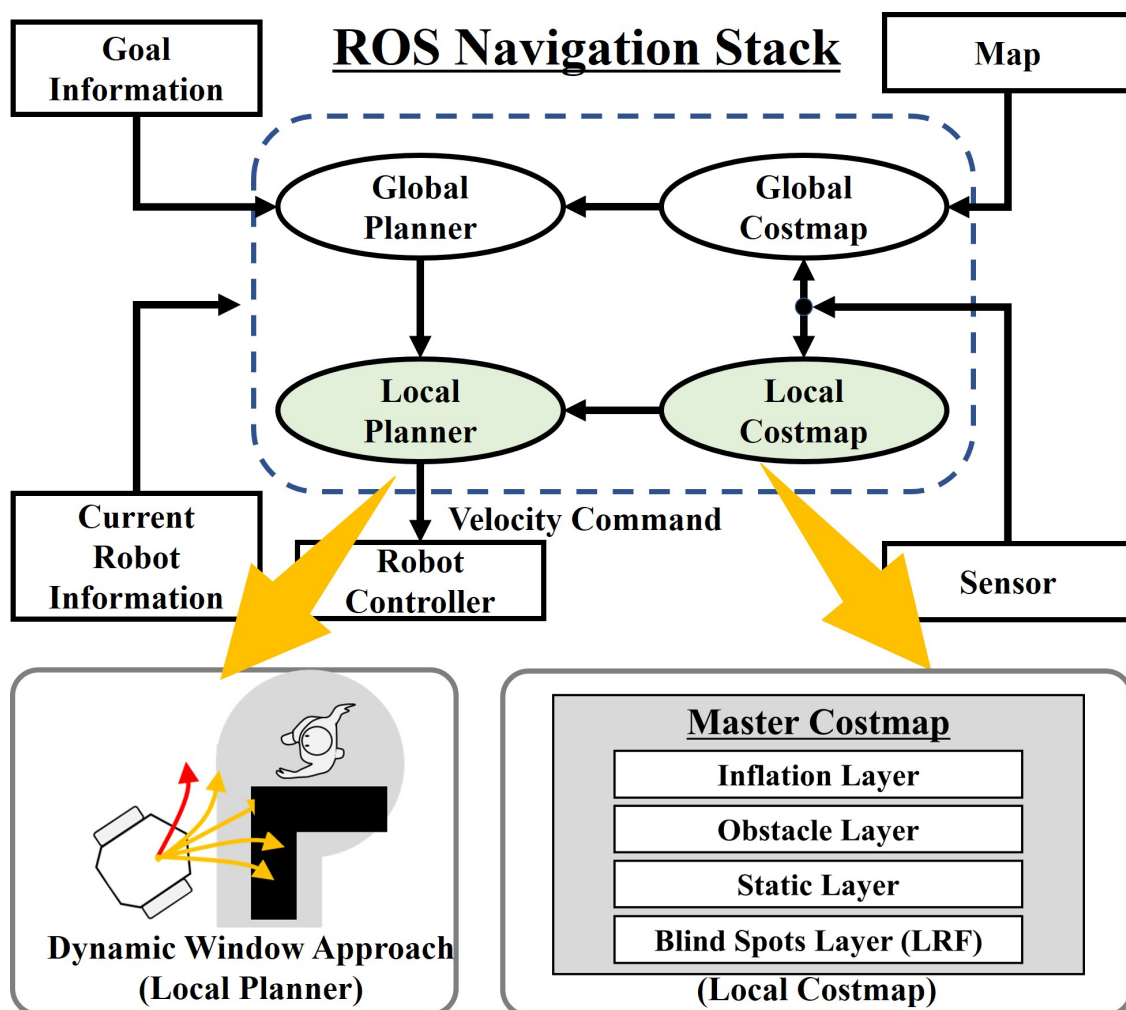


図 3.6 ROS Navigation Stack (LRF)

3.3.1 DWA 評価関数

(3.1) 式で示す DWA の評価関数に、死角領域を含めることで、死角領域を考慮した経路選択を行う。そこで提案手法では、DWA に用いる評価関数を (3.2) 式のように定義する。

$$c^{all} = W^p \cdot c^p + W^g \cdot c^g + W^d \cdot c^d \quad (3.2)$$

ここで、 W^d はコストマップ上の死角領域や障害物を考慮した重み、 c^d は DWA により生成された経路上での障害物や死角領域に関する最大マップコストである。この様に、 c^d 項に死角領域や障害物を考慮したコストを用いる事で、死角や障害物を考慮した経路生成が可能になる。これにより障害物情報に加えて、死角領域も考慮可能となる。

3.3.2 提案コストマップ

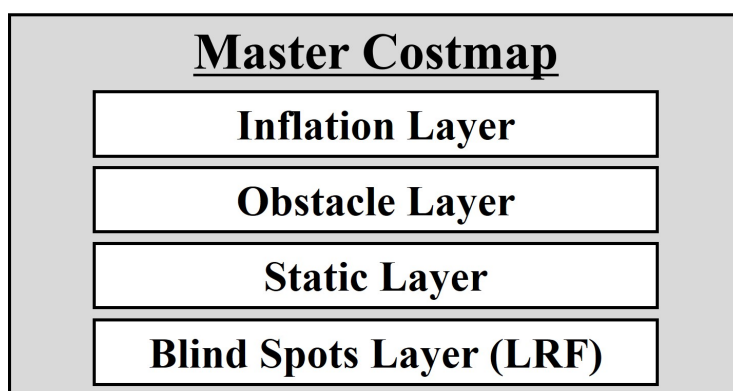


図 3.7 死角を考慮した階層コストマップ (LRF)

既存のコストマップでは測域センサの死角領域を考慮していないために、ロボットが人との出会い頭に衝突する可能性が高い。また実環境において安全性を考慮すると、人等が死角領域から飛び出してくる可能性を考慮した経路計画が必要である。そこで本研究ではコストマップにおいては、人が潜んでいる可能性がある地点を推定し、その地点から人は全方向に動けると仮定しコストを配置する。このコストマップを Blind Spots Layer (BSL) として提案する。既存の階層コストマップに加えて、BSL を新たに追加したコストマップシステムを図 3.7 に示す。BSL を追加することにより、死角領域からの人や移動障害物の飛び出しを考慮した経路計画がリアルタイムで可能となる。

3.3.3 死角検出方法

図 3.8 に BSL のフローチャートを示す。初めに LRF から測域情報を取得し、死角境界座標を算出する。そして死角境界座標から、死角領域に人が存在する可能性がある位置を推定する。その推定した位置を危険領域の中心座標とし、人が全方向移動すると仮定して円状にコストをコストマップに伝播させる事で危険領域を表現する。図 3.8 に示すフローチャートについて各 Step ごとに詳述する。

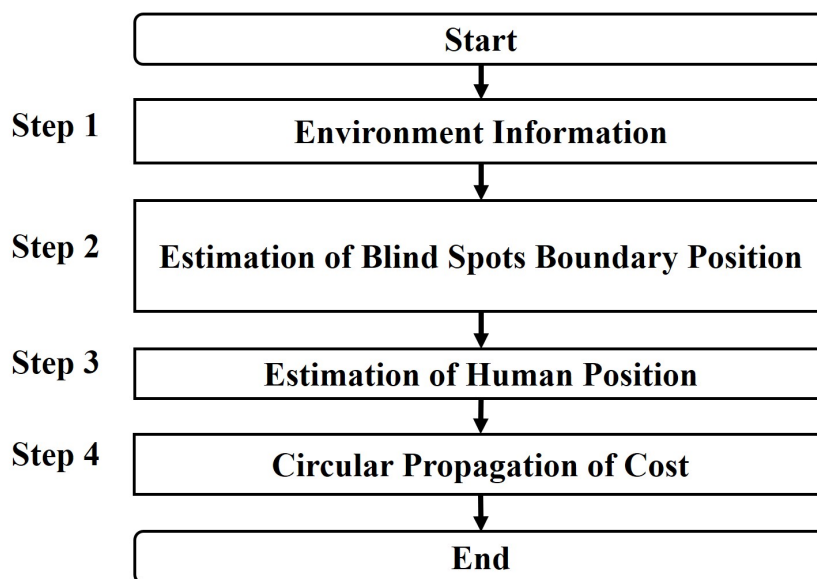


図 3.8 Blind Spots Layer のフローチャート

センサ値取得 (Step1)

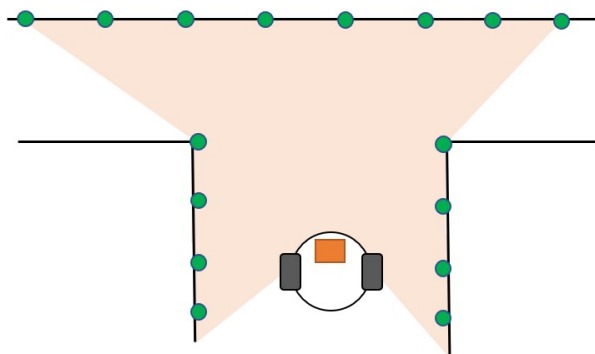


図 3.9 センサ値取得 (LRF)

図 3.9 に T 字路における環境情報取得の例を示す。図 3.9 における緑色の円が LRF で取得した距離と方位角から導いたロボット座標系における位置である。ロボットはこの取得した情報を用いて、死角領域を考慮した自律移動を行う。LRF はロボットの取り付け位置の水平平面上の環境情報しか取得できないことには注意されたい。

死角境界座標推定 (Step2)

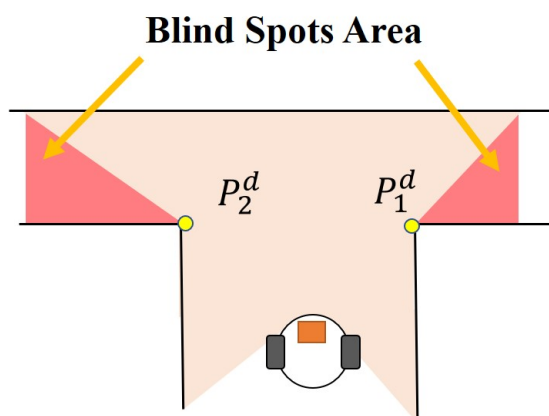


図 3.10 死角境界座標推定 (LRF)

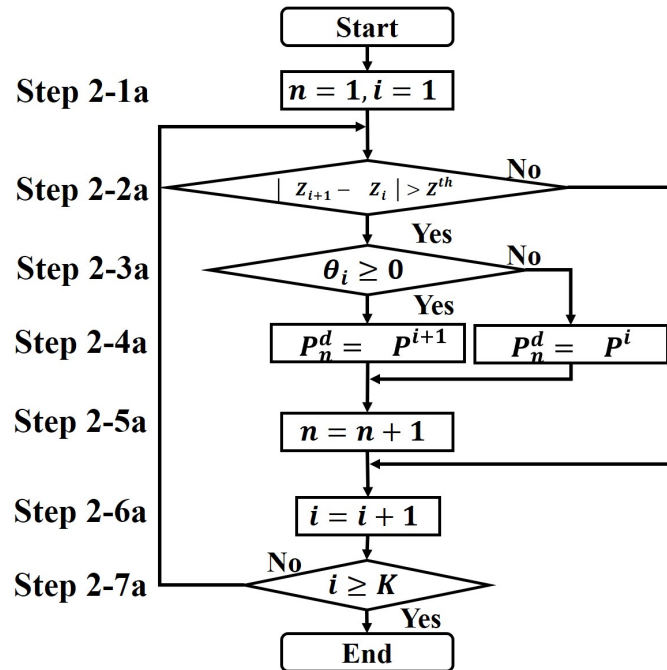


図 3.11 死角境界座標のフローチャート (LRF)

図 3.10 に死角領域の概念図を示す。図 3.10 の赤色で塗りつぶされている部分が LRF の測域情報が遮蔽されている死角領域、黄色の点が死角境界座標である。本章では、図 2.11 が示す座標系を用いて死角境界座標 $P_n^d = [Z_n^d, \theta_n^d]^T$ を定義する。ここで、 n は死角境界座標の個数である。図 3.11 に死角境界座標の検出方法のフローチャートを示す。死角境界座標の算出は、となりあう LRF 測域情報の差分値 $(z_{i+1} - z_i)$ が閾値 Z^{th} を上回る事で表現する。

危険領域の中心座標推定 (Step3)

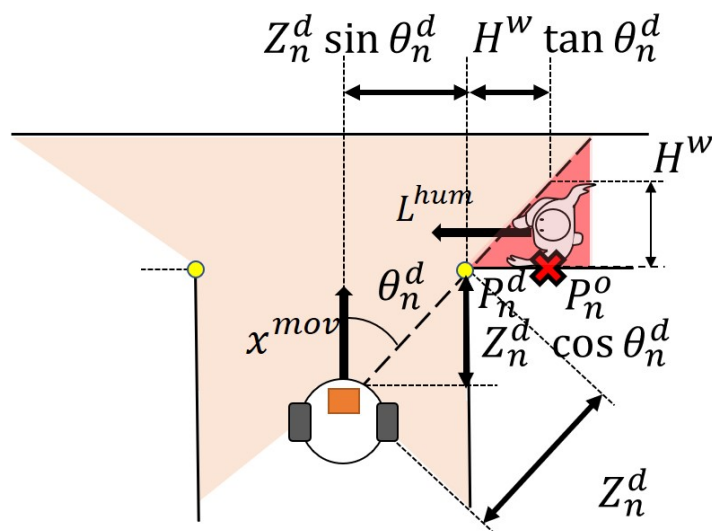


図 3.12 危険領域の中心座標推定

BSL を実装するにあたって、危険領域の中心を定義する。危険領域の中心とは人が存在する可能性がある領域でロボットに一番近い位置である。危険領域の中心は先ほど求めた死角境界座標 $P_n^d = [Z_n^d, \theta_n^d]^T$ と人の肩幅 H^w を利用して求める。図 3.12 に危険領域の中心を図示する。図 3.12 のように幾何的に危険領域の中心 $P_n^o = [x_n^o, y_n^o]^T$ を求める事が可能である。(3.3) 式が求めた危険領域の中心座標である。

$$\begin{bmatrix} x_n^o \\ y_n^o \end{bmatrix} = \begin{bmatrix} Z_n^d \cos \theta_n^d \\ Z_n^d \sin \theta_n^d + H^w \tan \theta_n^d \end{bmatrix} \quad (3.3)$$

コスト伝播 (Step4)

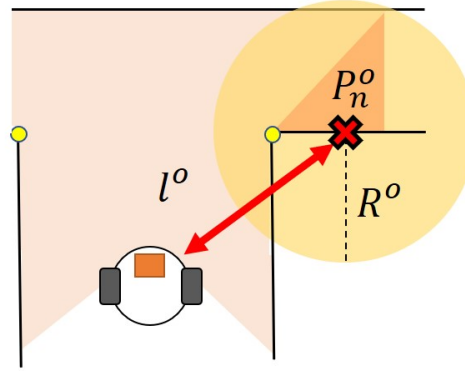


図 3.13 コスト伝播

BSLでは危険領域中心から円状にコストをコストマップに伝播させる。コストマップにコストを伝播させる範囲をロボットと人の停止可能距離から算出する。ロボットが速度 v^{mov} [m/s] の時、加速度 a^{mov} [m/s²] で減速した場合の停止までの距離を x^{mov} [m]、停止までの時間を t^{mov} [s] とする。加速度 a^{mov} で停止までの t^{mov} 秒間進んだときの距離 x^{mov} を (3.4) 式で求める。

$$x^{mov} = v^{mov}t^{mov} + a^{mov}\frac{(t^{mov})^2}{2} \quad (3.4)$$

速度 v^{mov} の時、加速度 a^{mov} で減速した際、停止するまでの時間 t^{mov} を (3.5) 式で求める。

$$t^{mov} = -\frac{v^{mov}}{a^{mov}} \quad (3.5)$$

(3.4) 式に、(3.5) 式を代入し (3.6) 式を求める。

$$x^{mov} = -\frac{(v^{mov})^2}{2a^{mov}} \quad (3.6)$$

速度 v^{mov} の時に停止までに距離 x^{mov} が必要である。

次に人が停止するまでの距離を求める。本章では人が停止しようとしてから、一歩で止まれることとする。よって、人の歩幅を L^{hum} [m] とし、人が停止するまでの距離とした。図 3.13 に危険領域の中心 P_n^o からロボットまでの直線距離 l^o [m] を示した。 l^o の値が $x^{mov} + L^{hum}$ を大きく上回るほど、ロボットと人の距離が大きく、衝突の可能性が低くなり安全といえる。このことから、危険領域の中心 P_n^o から半径を R^o としてコストを等心円状に伝播する。

ロボットの最高速度 v^{max} の時の x^{mov} を X^{max} とし、 R^o を (3.7) 式に示す。

$$R^o = X^{max} + L^{hum} + X^{off} \quad (3.7)$$

ここで X^{off} [m] とは、オフセット距離であり、ロボットと人との間の距離に余裕を持たせるために設定する。また、図 3.13 にコストを配置したイメージ図を示す。危険領域の中心 P_n^o から距離が R^o までは (3.8) 式で計算されるコストをコストマップに格納する。

$$c^{bsl} = \exp(-1.0 \times S^{cst} \times l^{dan}) \times H^{cst} \quad (3.8)$$

ここで、 c^{bsl} は危険領域の中心 P_n^o までの距離によって決められるコスト値、 S^{cst} はコストのスケーリング係数、 l^{dan} は危険距離の中心までの距離、 H^{cst} はコスト最大値である。(3.8) 式では変数は l^{dan} のみである。ここで、(3.8) 式のコスト c^{bsl} と危険距離の中心までの距離 l^{dan} の関係の一例を図 3.14 に示す。縦軸はコスト c^{bsl} を示しており、横軸は危険領域の中心座標からの距離 l^{dan} である。また、 S^{cst} の値を 1、5、10、かつ R^o の値を 1.25 とした。図 3.14 に示すように l^{dan} の増加に応じてコストが減少する。つまり危険領域の中心座標 P_n^o から、ロボットの速度と人の歩幅を考慮した距離 R^o までコストを円状に伝播させることで、死角領域付近で人との接触を考慮した経路生成が可能となる。

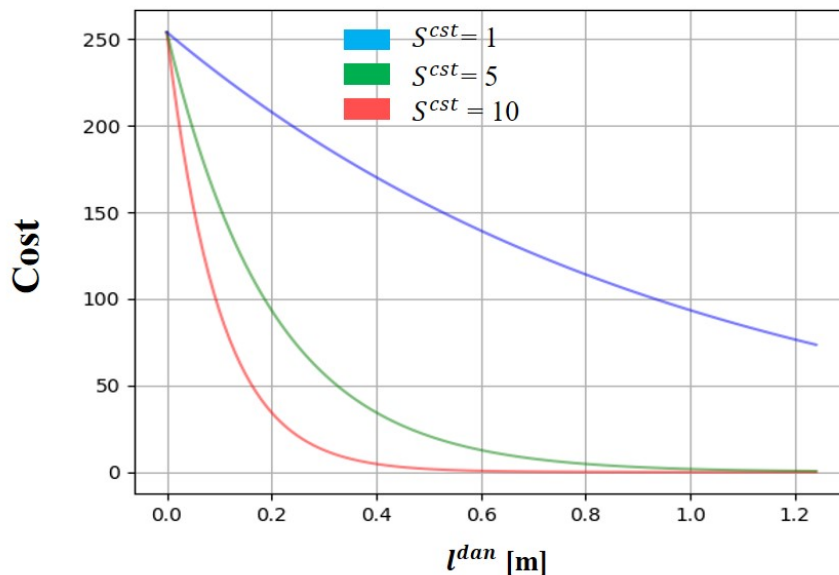
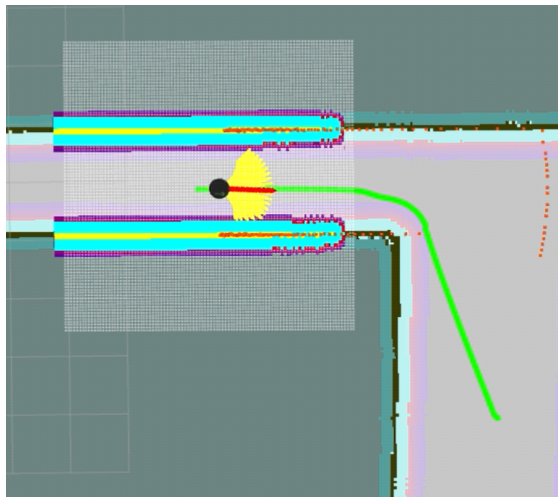


図 3.14 コストの移り変わり

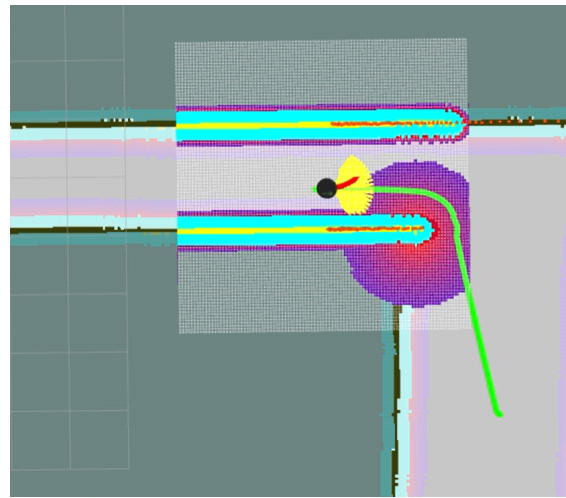
3.3.4 BSL を含めたコストマップの動作例

図 3.15 に BSL の動作例を示す。BSL では死角を考慮したコストをローカルコストマップ上で表現することで、死角を考慮した自律移動を実現する。ローカルコストマップはロボットの周りの白い四角で囲まれた領域であり、ロボットはこの領域で局所的経路計画 (DWA) を行う。ローカルコストマップの領域はロボットを中心に $5\text{m} \times 5\text{m}$ の正方形とする。自律移動では、初めに大域的経路計画を用いて、ゴール地点まで大まかな経路 (緑線) を生成する。次に大域的経路計画で生成された経路、障害物、死角を考慮した経路計画を局所的経路計画である DWA により生成する。ロボットから伸びている黄色の扇状の線が DWA により算出された経路の候補であり、赤太線が DWA の経路候補から、大域的経路計画で生成された経路、障害物、死角を考慮した評価関数に基づいて決定された経路である。ロボットは赤太線を生成した速度指令値を用いて行動を行う。

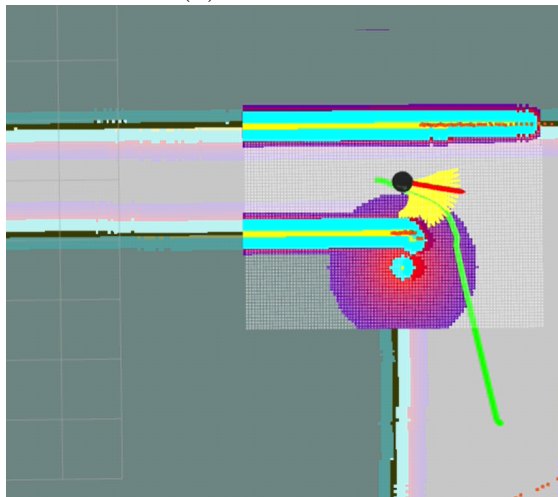
図 3.15(a) では、大域的経路計画に沿うように DWA の経路選択が行われている。このように死角領域がローカルコストマップ内に存在しない場合は、大域的経路で算出された経路と障害物のみを考慮した自律移動が行われる。図 3.15(b) では死角領域を検知し、コストマップにコストが円状に伝播している。図 3.15(c) では死角領域を回避するように、DWA の赤線が選択されている。図 3.15(d) では、死角領域がなくなり、大域的経路計画に沿うように DWA の経路が選択されている。



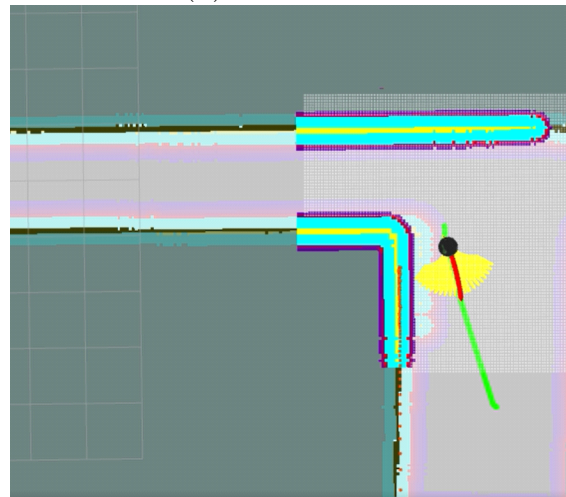
(a) Situation 1



(b) Situation 2



(c) Situation 3

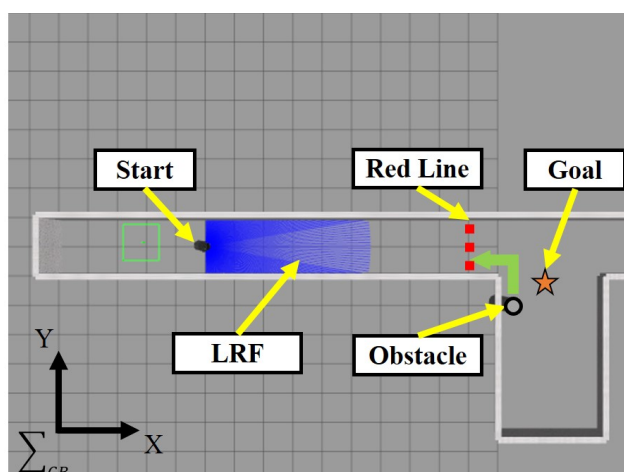


(d) Situation 4

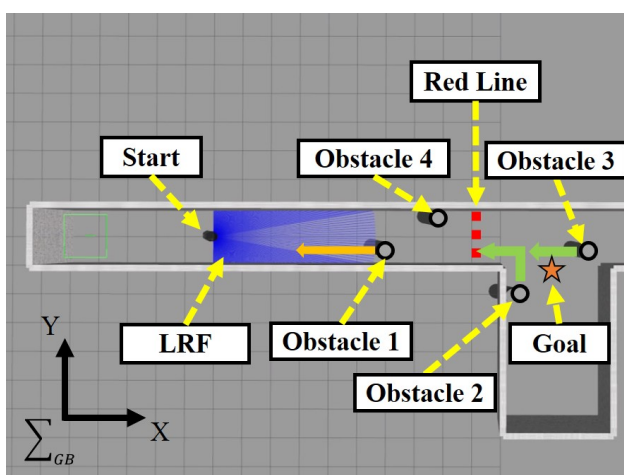
図 3.15 BSL の動作例 (LRF)

シミュレーション環境

シミュレーション環境は図 3.18 が示すように 2 つの環境を用意した。まず図 3.18(a) の環境を Case A とし、動的障害物が 1 つ存在する環境下で従来手法と提案手法の比較を行った。次に図 3.18(b) の環境を Case B とし、静的・動的障害物が存在する複雑な環境下で従来手法と提案手法に対してシミュレーションを行い、提案手法の有用性を確認した。また Case A、Case B ともに障害物情報は事前に与えず、ロボットに搭載されている LRF から取得した。



(a) Case A



(b) Case B

図 3.18 シミュレーション環境

• Case A (動的障害物が1つ存在する環境)

図 3.18(a) が示すように、ロボットと青線の LRF では認識できない位置に人と想定した動的障害物を設置した。また、移動ロボットが赤線を超えた際に動的障害物は緑色の矢印上を人の歩行速度と見立てて時速 4.5[km/h] 移動し、ロボットは最大時速 2.3[km/h] で DWA で走行することとした。この環境下で従来手法 (Case A-1) と提案手法 (Case A-2) を用いてシミュレーションを行った。

• Case A-1 ROS 標準

局所的経路計画・・・DWA (BSL なし)

• Case A-2 ROS 標準 + BSL(LRF)

局所的経路計画・・・DWA (BSL あり)

• Case B (静的・動的障害物が複数存在する環境)

図 3.18(b) が示すように、図 3.18(a) のシミュレーション環境に静的・動的障害物を加えてシミュレーションを実施した。動的障害物は Obstacle 1~3 で、静的障害物は Obstacle 4 とした。図 3.18(b) が示すように、移動ロボットのスタートと同時に Obstacle 1 はオレンジ色の矢印上を人の歩行速度と見立てて時速 4.5[km/h] 移動し、ロボットは最大時速 2.3[km/h] で DWA で走行することとした。またロボットが赤線を超えた際に Obstacle 2、3 は緑色の矢印上を人の歩行速度と見立てて時速 4.5[km/h] 移動することとした。この環境下で従来手法 (Case B-1) と提案手法 (Case B-2) を用いてシミュレーションを行った。

• Case B-1 ROS 標準

局所的経路計画・・・DWA (BSL なし)

• Case B-2 ROS 標準 + BSL(LRF)

局所的経路計画・・・DWA (BSL あり)

シミュレーション結果

・ Case A (動的障害物が 1 つ存在する環境)

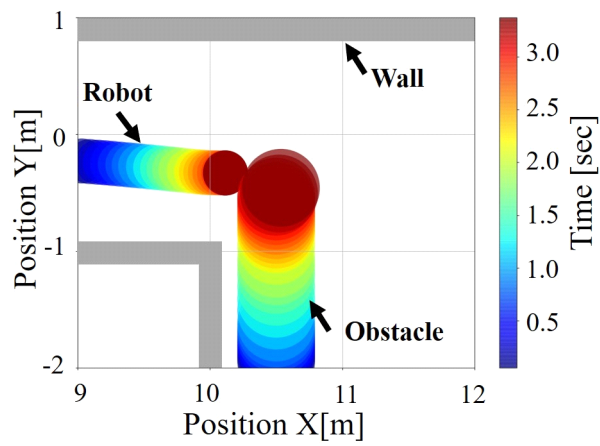
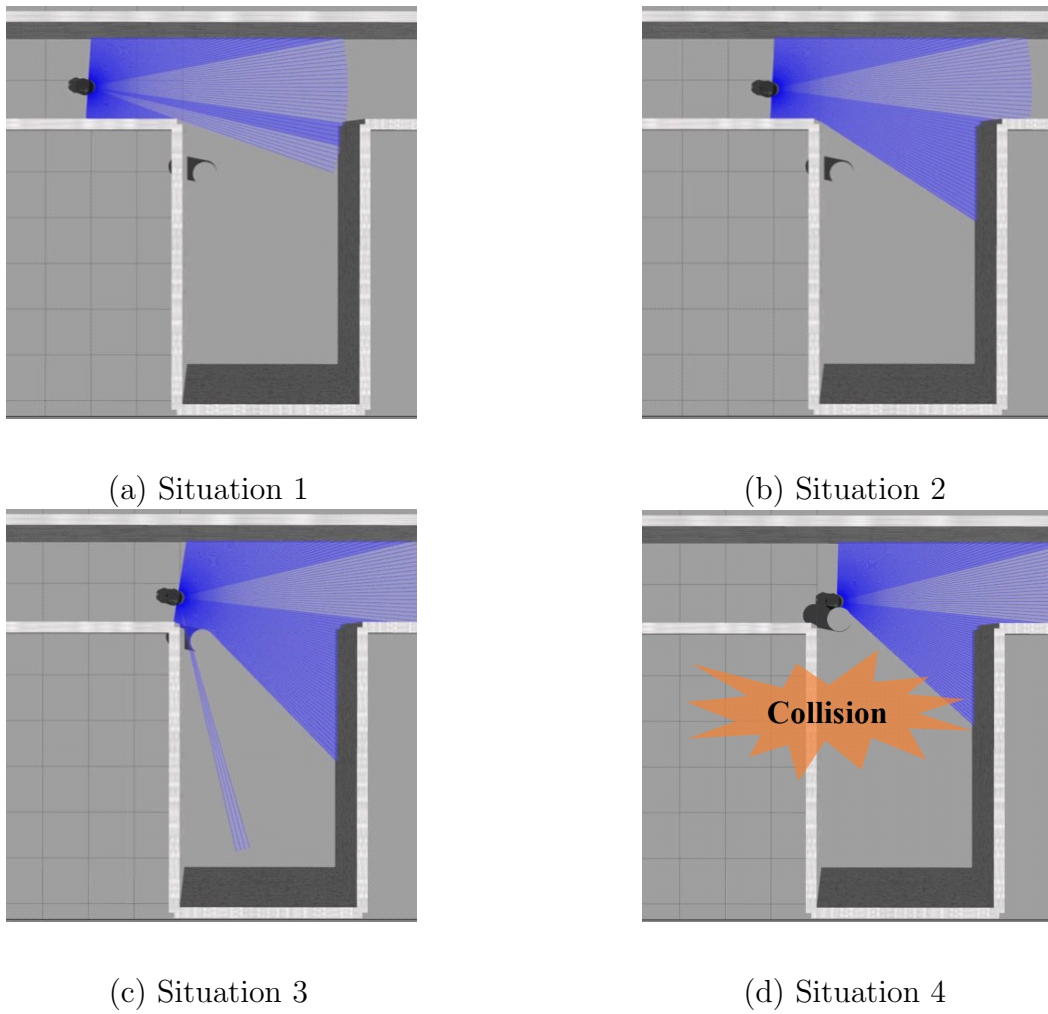
表 3.1 にゴール到達の可否を示した。図 3.19、3.20 に従来手法と提案手法を用いたシミュレーションでのロボットと障害物の移動状況とロボットと障害物の時間毎の軌跡を示した。

表 3.1 シミュレーション結果 (Case A)

Case	Goal
Case A-1 : ROS Default	×
Case A-2 : ROS Default + BSL (LRF)	○

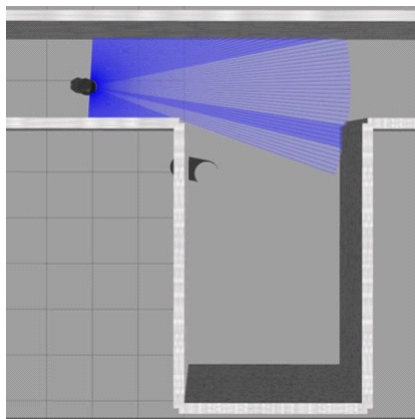
CaseA-1 の従来手法の結果について述べる。図 3.19(a) が示すように、ロボットはゴール地点へ向かっているものの、死角領域に潜む障害物を認識できていない状況であった。図 3.19(b) が示すように、ロボットは死角領域を考慮した経路計画を行っていないため、曲がり角付近を走行した。図 3.19(c) が示すように、ロボットは進行方向右折時に曲がり角の先に潜んでいた障害物を検知した。図 3.19(d) が示すように、障害物を検知したタイミングが遅かったためにロボットは障害物と衝突した。CaseA-1 の従来手法では図 3.19 が示すように、死角領域を考慮していないのでロボットと障害物は衝突した。

CaseA-2 の提案手法の結果について述べる。図 3.20(a) が示すように、ロボットはゴール地点へ向かった。図 3.20(b) が示すように、ロボットは死角領域を考慮した経路計画を行ったため、曲がり角付近を回避するように走行した。図 3.20(c) が示すように、ロボットは障害物検知前に事前に曲がり角付近を回避するように走行したため、曲がり角から咄嗟に現れる障害物を回避した。図 3.20(d) が示すように、ロボットは障害物と衝突することなく目標地点へ到達した。CaseA-2 の提案手法では図 3.20 が示すように、死角領域を考慮しているためにロボットは障害物と衝突せずに走行した。

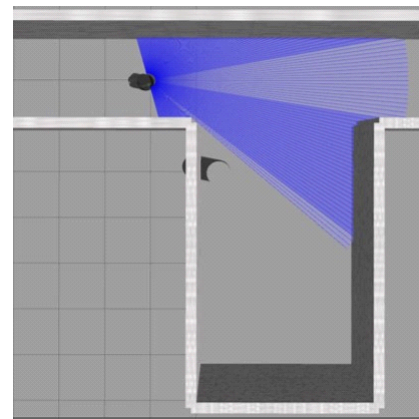


(e) Trajectory Result

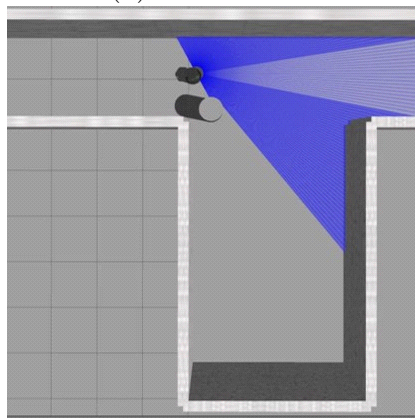
図 3.19 Case A-1 の動作



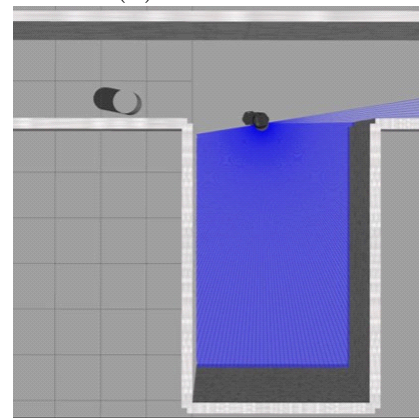
(a) Situation 1



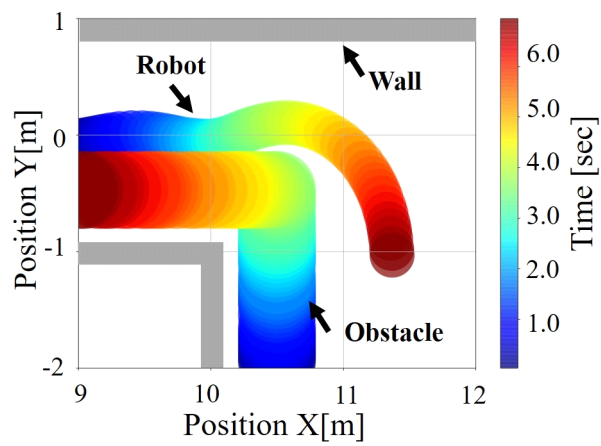
(b) Situation 2



(c) Situation 3



(d) Situation 4



(e) Trajectory Result

図 3.20 Case A-2 の動作

• Case B (静的/動的障害物が複数存在する環境)

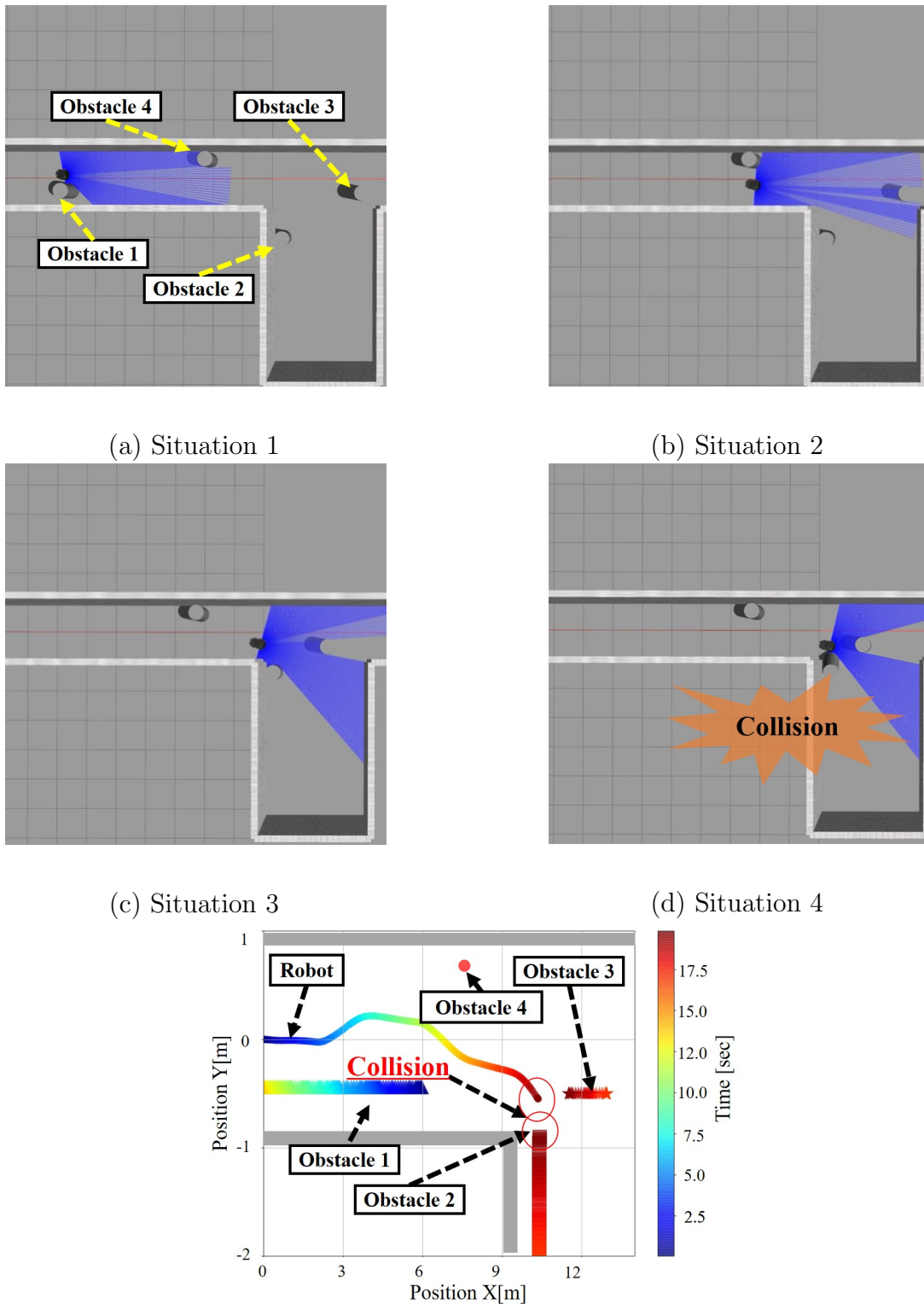
表 3.2 にゴール到達の可否を示した。図 3.21、3.22 に従来手法と提案手法を用いたシミュレーションでのロボットと障害物の移動状況とロボットと障害物の時間毎の軌跡を示した。

表 3.2 シミュレーション結果 (Case B)

Case	Goal
Case B-1 : ROS Default	×
Case B-2 : ROS Default + BSL (LRF)	○

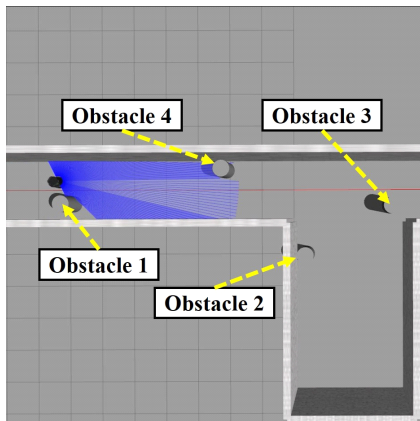
CaseB-1 の従来手法の結果について述べる。図 3.21(a) が示すように、ロボットは動的障害物である Obstacle1 を回避した。図 3.21(b) が示すように、ロボットは静的障害物である Obstacle4 を回避した。図 3.21(c) が示すように、ロボットは死角領域を考慮しない経路計画を行ったために、曲がり角付近を走行した。図 3.21(d) が示すように、死角領域を考慮していない経路計画を行ったため、ロボットは動的障害物である Obstacle2 と衝突した。図 3.21(e) のように、従来手法では死角領域や障害物を考慮していないためにロボットは障害物と衝突した。

CaseB-2 の提案手法の結果について述べる。図 3.22 に提案手法を用いたシミュレーションでのロボットと障害物の移動状況とロボットと障害物の時間毎の軌跡を示した。図 3.22(a) が示すように、ロボットは動的障害物である Obstacle1 を回避した。図 3.22(b) が示すように、ロボットは静的障害物である Obstacle4 を回避した。図 3.22(c) が示すように、ロボットは死角領域を考慮した経路計画を行ったために、曲がり角から飛び出してきた動的障害物である Obstacle2 との衝突を回避した。図 3.22(d) が示すように、全ての障害物と衝突することなくゴール地点へ到達した。図 3.22(e) のように、提案手法では死角領域や障害物を考慮しているためにロボットは障害物と衝突せずに走行した。静的・動的障害物が複数ある環境下においても提案手法によりゴール地点にたどり着くことができるために、提案手法の有用性が確認された。

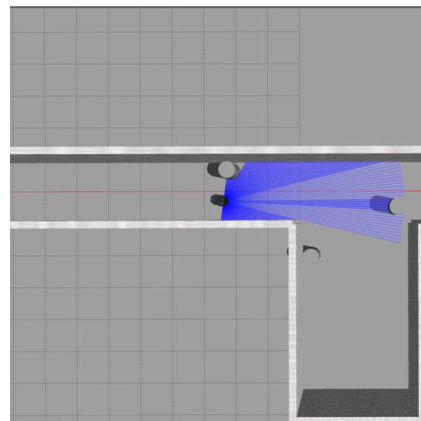


(e) Trajectory Result

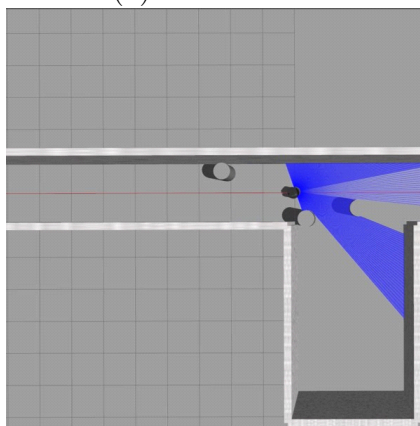
図 3.21 Case B-1 の動作



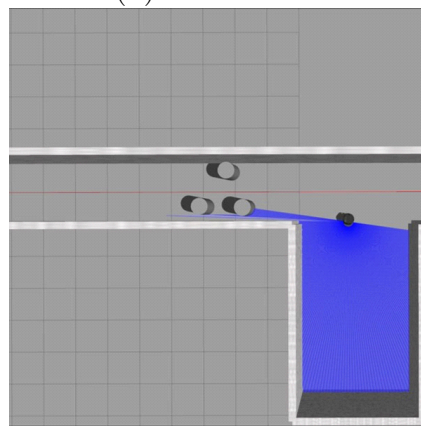
(a) Situation 1



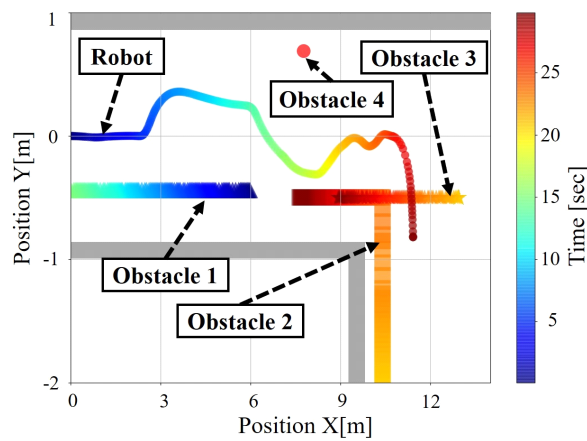
(b) Situation 2



(c) Situation 3



(d) Situation 4



(e) Trajectory Result

図 3.22 Case B-2 の動作

3.4 RGB-D カメラによる死角領域を考慮した自律移動

LRF での死角領域を考慮した手法の問題点は、LRF で取得可能な環境情報がセンサ取り付け位置での水平平面の情報しか取得できないために、危険領域を意味するコストを過剰に生成してしまう問題があった。そこで3次元情報が取得できる RGB-D カメラを用いる事で、LRF よりもロバストな死角領域の検出を行い、死角領域を考慮した自律移動を行う。図 3.23 が示すように、本節でも同様に自律移動システムの基盤となる構成は ROS Navigation Stack を採用し、リアルタイムで RGB-D カメラによる死角領域を考慮した Local Costmap を提案し、死角領域を考慮したコストを Local Planner である DWA の評価関数に含めることで上述の問題を解決する。

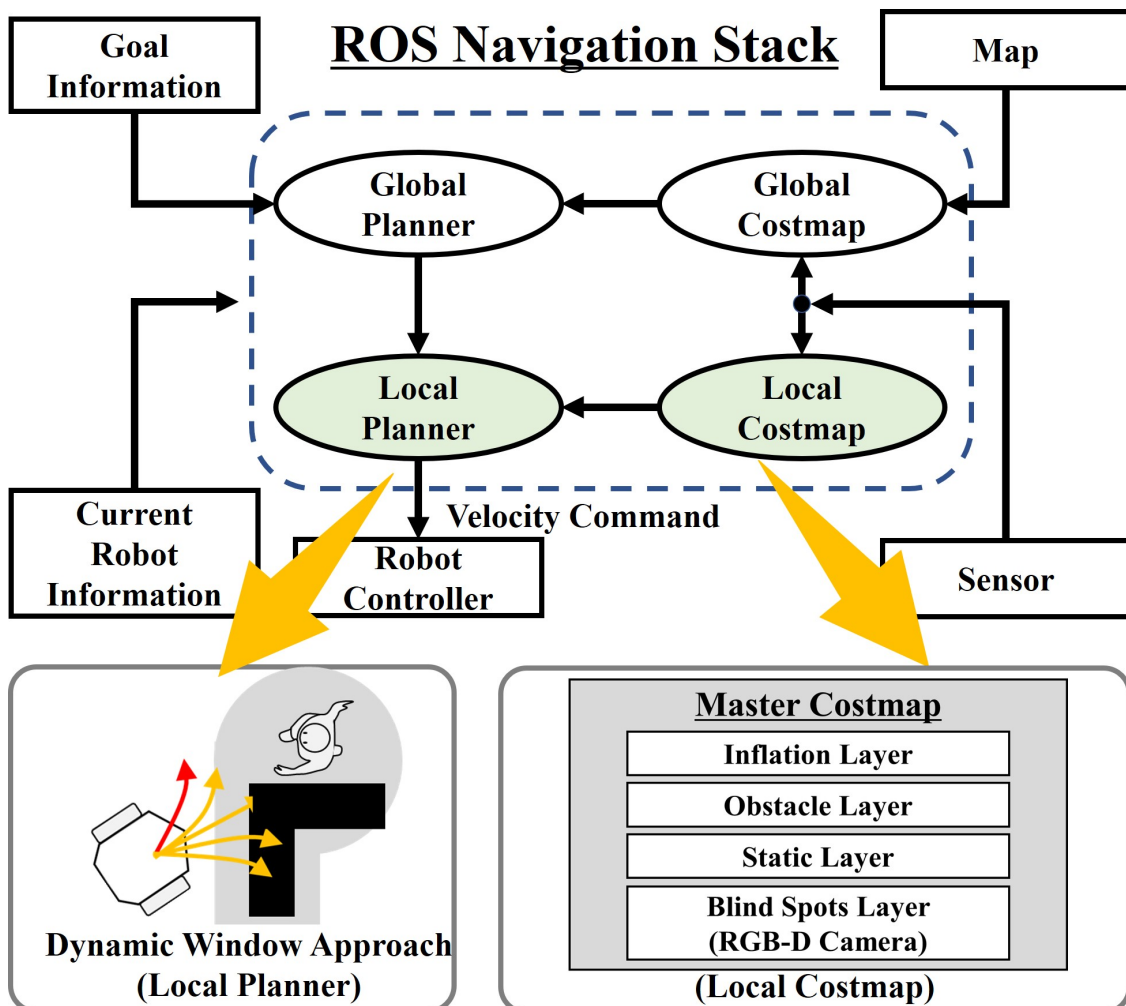


図 3.23 ROS Navigation Stack (RGB-D Camera)

3.4.1 DWA 評価関数

(3.2) 式で示す DWA の評価関数にはゴールとの距離、障害物との距離、死角領域は考慮されているが、ロボットの速度については考慮されていない。そこで提案手法では、DWA に用いる評価関数を次式のように定義する。

$$c^{all} = W^p \cdot c^p + W^g \cdot c^g + W^d \cdot c^d + W^v \cdot c^v \quad (3.9)$$

ここで、 W^v はロボットの速度に関する重み、 c^v はロボットの速度の逆数である。この様に c^v 項にロボットの速度を用いる事で、より大きな速度を使用した経路候補を選択することとなり、ゴール地点への到達時間の減少が見込まれる。

3.4.2 提案コストマップ

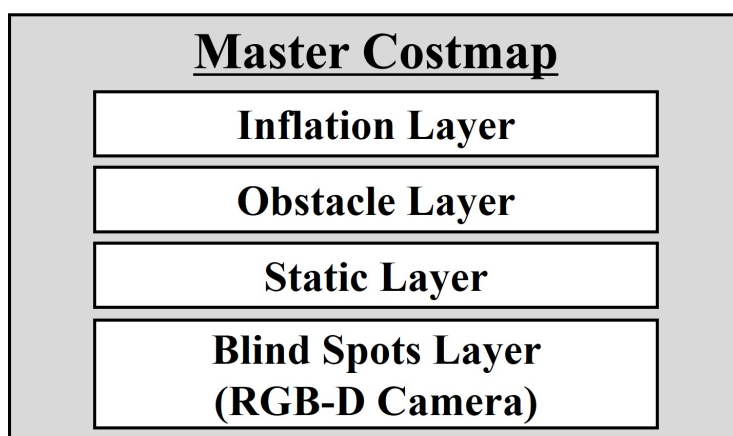
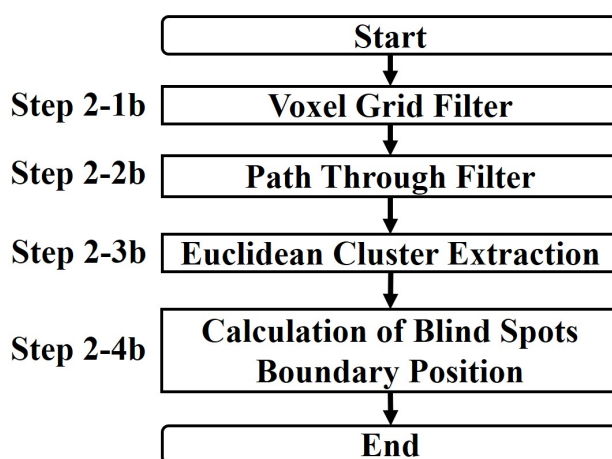


図 3.24 死角を考慮した階層コストマップ (RGB-D Camera)

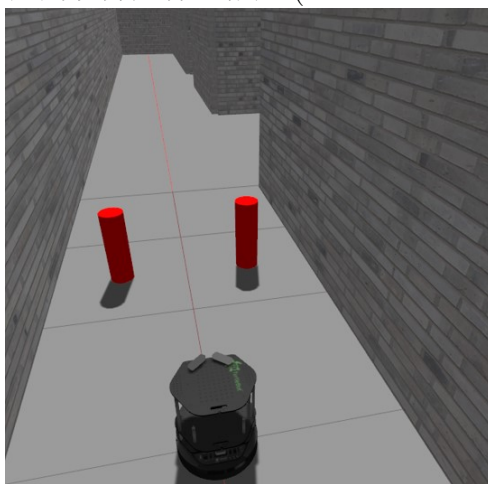
図 3.24 に提案したコストマップを示す。提案コストマップでは、RGB-D Camera を用いた Blind Spots Layer により死角領域を考慮する。これによりローカルコストマップに死角領域を考慮したコストを生成可能となり、ロボットは死角領域を考慮した経路計画を行うことが可能となる。

3.4.3 死角検出方法

死角検出では、初めに RGB-D カメラから 3 次元点群情報を取得し、死角境界座標を算出する。そして死角境界座標から、死角領域に人が存在する可能性がある位置を推定する。その推定した位置を危険領域の中心座標とし、人が全方向移動すると仮定して円状にコストをコストマップに伝播させる事で危険領域を表現する。図 3.8 に示すフローチャートの Step2 について、図 3.25 を用いて各 Step ごとに詳述する。



(a) 死角境界座標の推定 (RGB-D Camera)



(b) 点群を取得した環境

図 3.25 RGB-D カメラによる死角境界座標の推定

Voxel Grid Filter (Step2-1b)

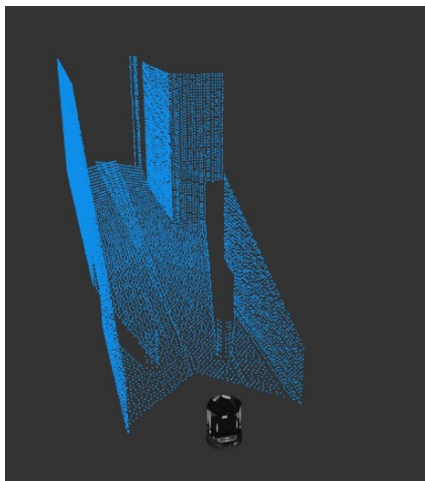


図 3.26 Voxel Grid Filter (Step2-1b)

図 3.26 が示すように、図 3.25 で取得した点群を Voxel Grid Filter を用いて点群の量を削減する。

Path Through Filter (Step2-2b)

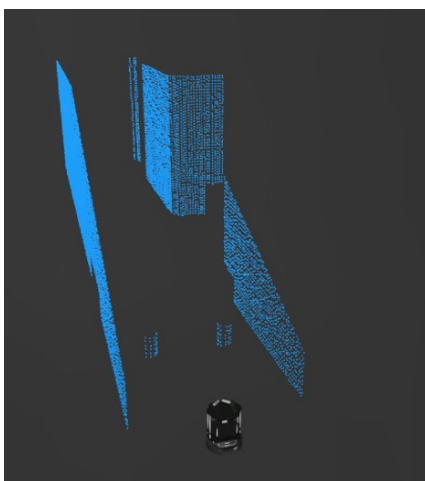


図 3.27 Path Through Filter (Step2-2b)

図 3.27 が示すように、Path Through Filter を用いて床面の点群を削除する。

Euclidean Cluster Extraction (Step2-3b)

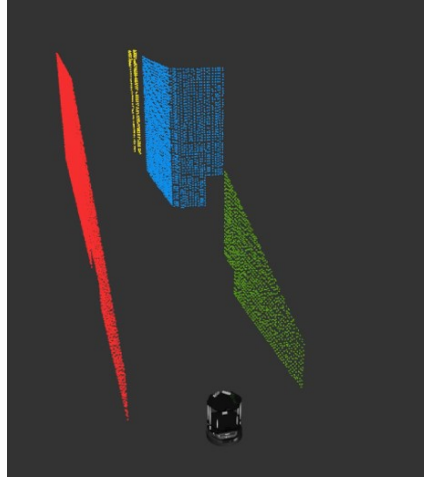


図 3.28 Euclidean Cluster Extraction (Step2-3b)

図 3.28 が示すように、Euclidean Cluster Extraction を用いて距離と点群数に応じてクラスターを分類する。クラスターとみなす点群数を設定することで、点群数の少ない物体等を除外する事が可能である。

Blind Spots Boundary Position (Step2-4b)

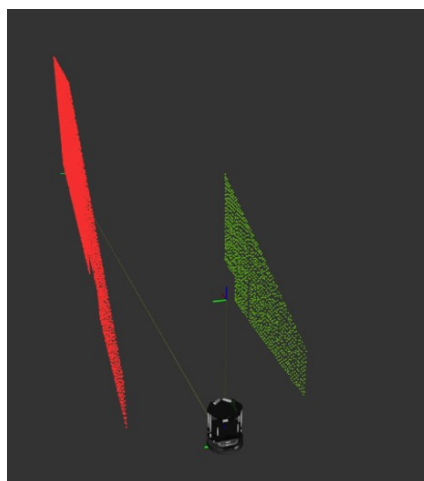


図 3.29 Blind Spots Boundary Position (Step2-4b)

図 3.29 が示すように、最近傍の点群から死角境界座標を次のように算出する。

$$\begin{bmatrix} x_n^d \\ y_n^d \end{bmatrix} = \begin{bmatrix} \arg \max(\Gamma_n^x) \\ \frac{\arg \max(\Gamma_n^y) + \arg \min(\Gamma_n^y)}{2} \end{bmatrix} \quad (3.10)$$

ここで Γ_n^x は n 番目の点群の X 軸方向の値、 Γ_n^y は n 番目の点群の Y 軸方向の値である。

コスト伝播 (Step3-4)

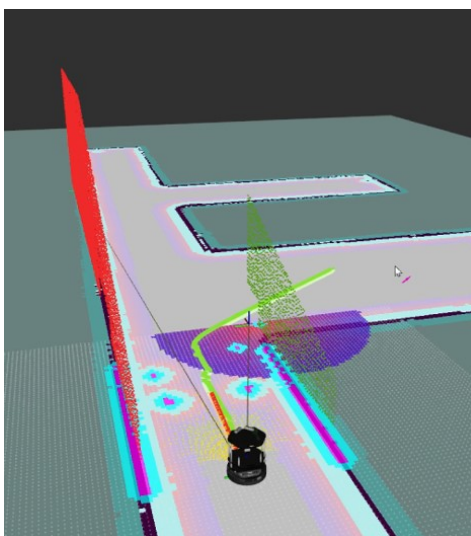


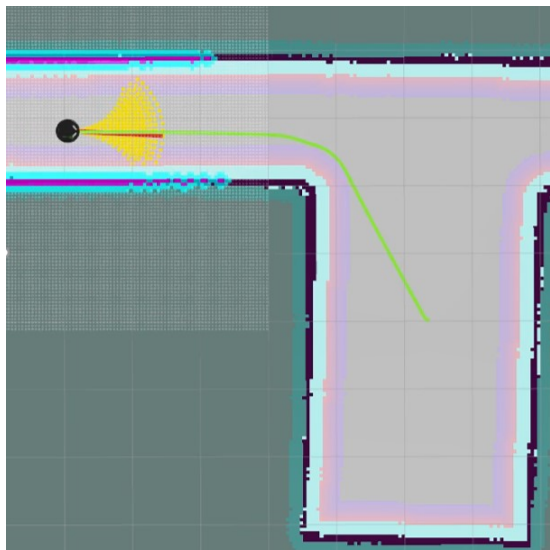
図 3.30 コスト生成

図 3.30 が示すように、危険領域にコストを伝播させる。ロボットは円状に伝播されたコストを回避する経路選択を行うために、死角領域近傍をさける自律移動を行うことが可能となる。

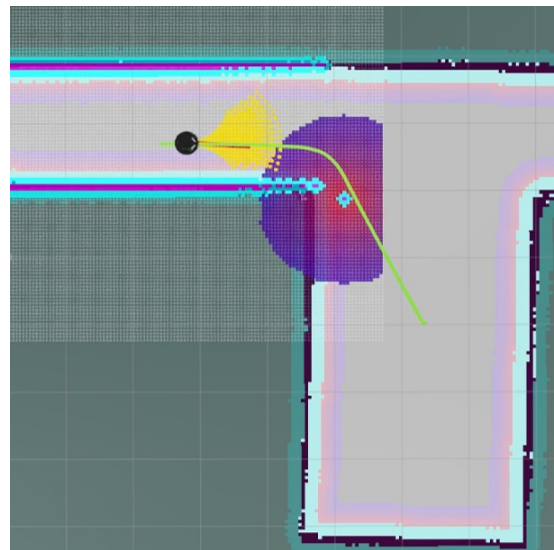
3.4.4 危険領域の移り変わり

図 3.31 に RGB-D カメラを用いた場合での BSL の動作例を示す。BSL では死角を考慮したコストをローカルコストマップ上で表現することで、死角を考慮した自律移動を実現する。ローカルコストマップはロボットの周りの白い四角で囲まれた領域であり、ロボットはこの領域で局所的経路計画 (DWA) を行う。ローカルコストマップの領域はロボットを中心に 5m × 5m の正方形とする。自律移動では、初めに大域的経路計画を用いて、ゴール地点まで大まかな経路 (緑線) を生成する。次に大域的経路計画で生成された経路、障害物、死角を考慮した経路計画を局所的経路計画である DWA により生成する。ロボットから伸びている黄色の扇状の線が DWA により算出された経路の候補であり、赤太線が DWA の経路候補から、大域的経路計画で生成された経路、障害物、死角を考慮した評価関数に基づいて決定された経路である。ロボットは赤太線を生成した速度指令値を用いて行動を行う。

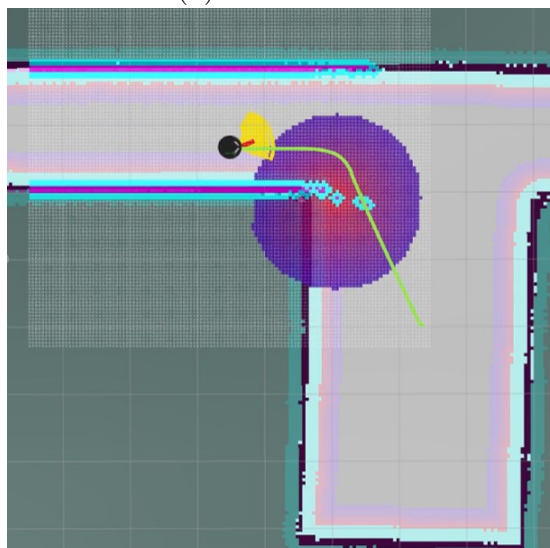
図 3.31(a) では、ローカルコストマップ内に死角が現れていないので、死角を考慮していない経路計画が行われている。図 3.31(b) では死角領域を検知し、コストマップにコストが円状に伝播している。図 3.31(c) では死角領域を回避するように、DWA の赤線が選択されている。図 3.31(d) では、死角領域がなくなり、大域的経路計画に沿うように DWA の経路が選択されている。このように、RGB-D カメラを用いた場合でも BSL は動作可能である。



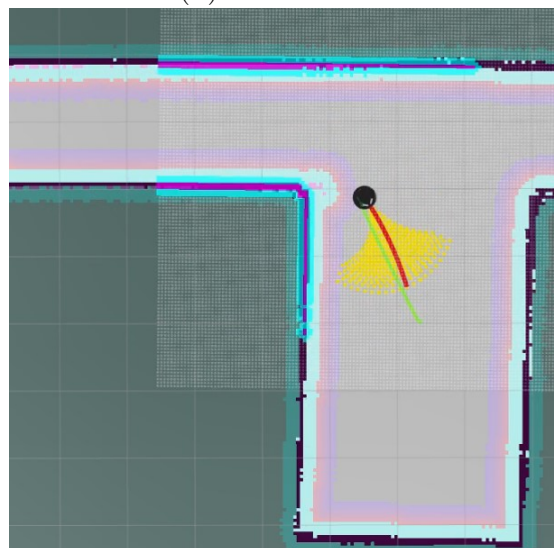
(a) Situation 1



(b) Situation 2



(c) Situation 3



(d) Situation 4

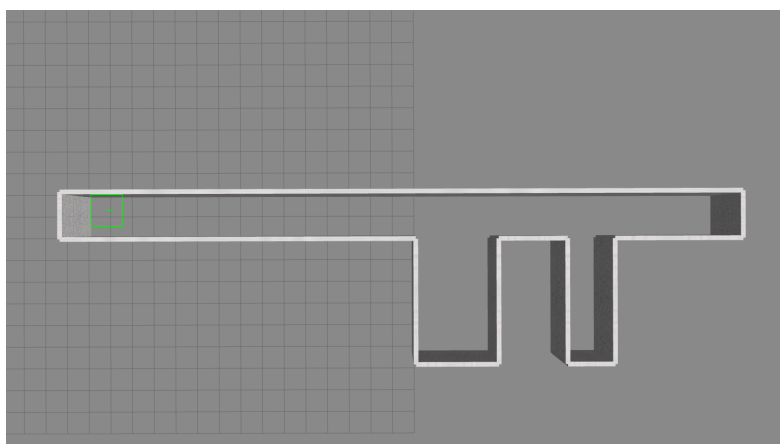
図 3.31 BSL の動作例 (RGB-D Camera)

3.4.5 シミュレーション

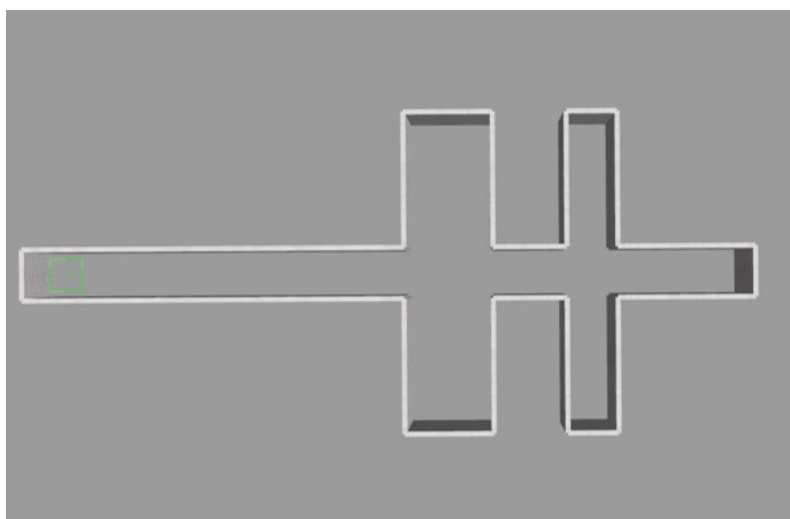
本節では、RGB-D カメラによる死角領域を考慮した自律移動のシミュレーションを行う。

シミュレーション事前準備

シミュレーションを行うために、図 3.32 の環境下を用意した。自律移動で使用する事前地図を作成するために、事前地図作成時には障害物は配置せずに、SLAM の gmapping [57] を用いて地図生成を行った。図 3.33 に作成した地図を示した。

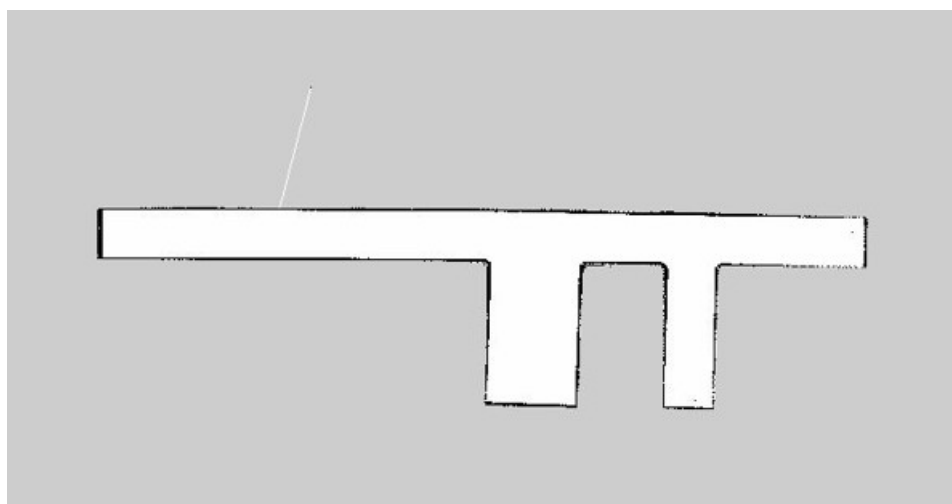


(a) L 字路環境

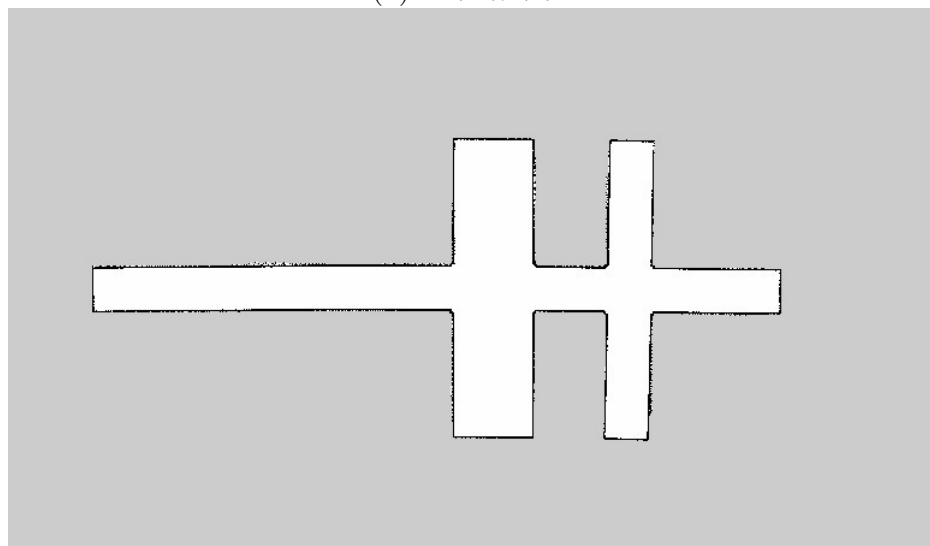


(b) 十字路環境

図 3.32 地図作成環境



(a) L 字路環境



(b) 十字路環境

図 3.33 作成地図

シミュレーション条件

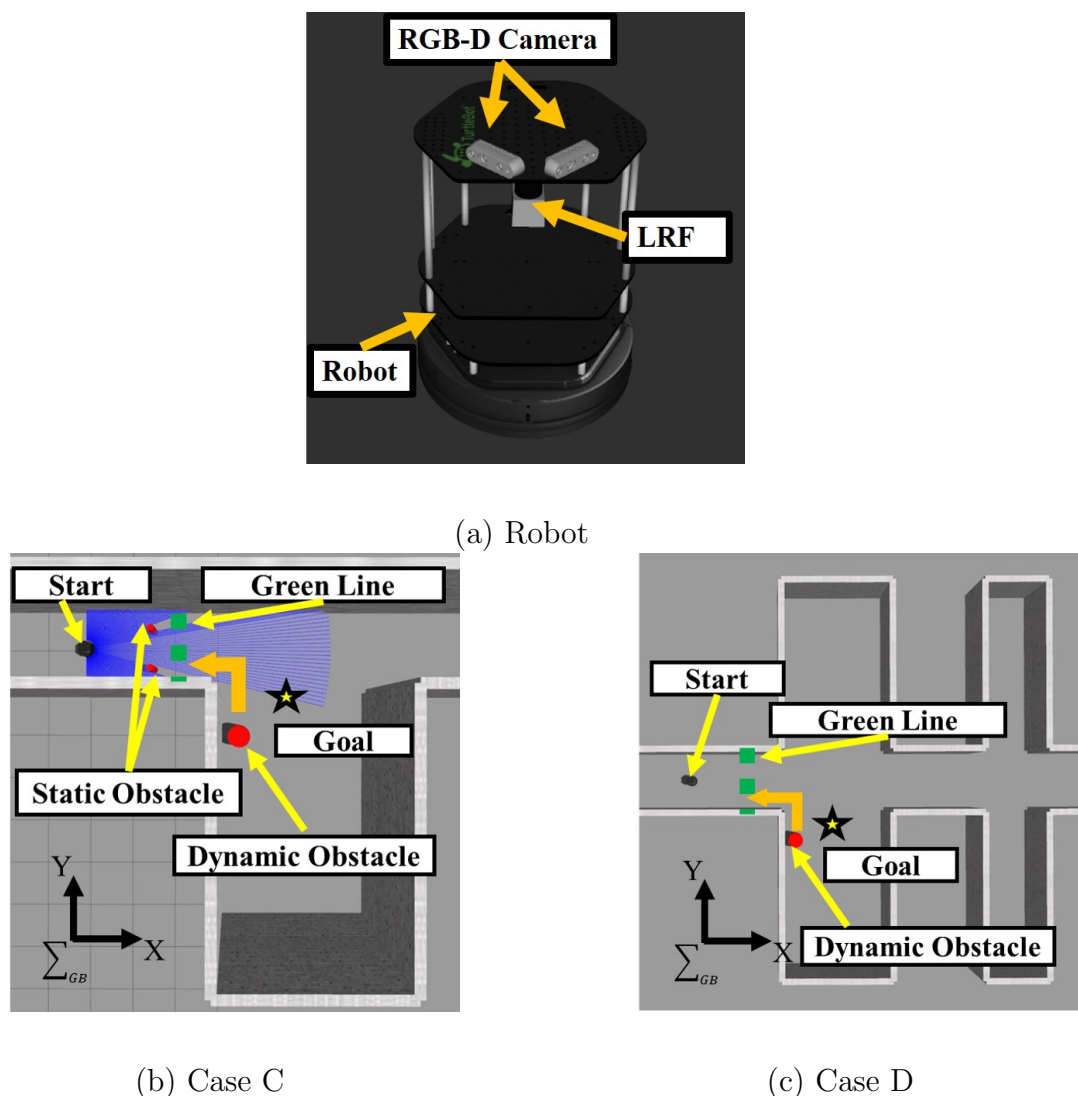


図 3.34 Simulation Environment

図 3.34(a) が示すように、ロボットには RGB-D カメラ 2 台と LRF が搭載した。図 3.34(b) の環境を Case C とし、L 字路に静的障害物を 2 つ、動的障害物が 1 つ存在する環境下で評価した。図 3.34(c) の環境を Case D とし、十字路に動的障害物が 1 つ存在する環境下で評価した。図 3.34(b)(c) が示すように、ロボットから認識できない位置に人と想定した動的障害物を設置した。また、移動ロボットが緑線を超えた際に動的障害物は橙色の矢印上を人の歩行速度と見立てて時速 4.0[km/h] 移動し、ロボットは最大時速 2.0[km/h] で DWA により走行することとした。障害物情報は事前に与えず、ロボットに搭載されている LRF や RGB-D カ

メラからリアルタイムで取得した。表 3.3 にシミュレーションに使用する手法を示す。本項では従来手法を *Method 1* と *Method 2*、提案手法を *Method 3* と *Method 4* して扱う。環境情報の取得として、*Method 1* と *Method 2* では LRF のみ、*Method 3* と *Method 4* では RGB-D Camera と LRF を用いた。表 3.4 はシミュレーションで用いたパラメータを示す。パラメータは試行錯誤により決定した。この環境下で従来手法と提案手法を用いてシミュレーションを行った。

表 3.3 シミュレーション方式

Method	Cost Map	Cost Function
<i>Method 1</i>	ROS Default	<i>eq.(3.1)</i>
<i>Method 2</i>	ROS Default + BSL (LRF)	<i>eq.(3.2)</i>
<i>Method 3</i>	ROS Default + BSL (RGB-D)	<i>eq.(3.2)</i>
<i>Method 4</i>	ROS Default + BSL (RGB-D)	<i>eq.(3.9)</i>

表 3.4 パラメータ

Character	Value	Description
L	0.8[m]	Human Stride
X^{off}	0.2[m]	Offset Distance
H^w	0.5[m]	Human Shoulders
S^{cst}	1	Cost Scaling Factor
H^{cst}	253	Maximum Cost
W^p	2	Weight Coefficient for Global Path
W^g	1	Weight Coefficient for Goal Position
W^o	10	Weight Coefficient for Obstacles
W^d	10	Weight Coefficient for Obstacles and Blind Spots Region
W^v	0.5	Weight Coefficient for Velocity
T^{pre}	4.0[s]	Predicted Time
Z^{th}	1.0	Threshold of BSBP

シミュレーション結果

図 3.35 と図 3.36 に Case C と Case D それぞれのシミュレーション結果を示した。

Case C のシミュレーション結果について述べる。Method 1 では、図 3.35(a) のように、死角領域を考慮した経路計画を行っていないためにロボットと障害物は衝突した。Method 2-Method 4 では、図 3.35(b)-(d) のように、経路計画時に死角領域を考慮しているためにロボットは障害物と衝突せずにゴール地点へ到達した。表 3.5 のように、従来手法 (Method 2) ではゴール到達時間が 25.3[sec]、提案手法 (Method 3) では 20.2[sec]、提案手法 (Method 4) では 18.3[sec] であった。提案手法 (Method 4) のゴール到達時間は 27.7% 従来手法 (Method 2) と比べて向上した。

Case D のシミュレーション結果について述べる。Method 1 では、図 3.36(a) のように、死角領域を考慮した経路計画を行っていないためにロボットと障害物は衝突した。Method 2-Method 4 では、図 3.36(b)-(d) のように、経路計画時に死角領域を考慮しているためにロボットは障害物と衝突せずにゴール地点へ到達した。表 3.6 のように、従来手法 (Method 2) ではゴール到達時間が 23.8[sec]、提案手法 (Method 3) では 21.2[sec]、提案手法 (Method 3) では 20.5[sec] であった。提案手法 (Method 4) のゴール到達時間は 13.9% 従来手法 (Method 2) と比べて向上した。

提案手法が従来手法よりも目的地に早く到達したのには 2 つの理由があった。1 つ目は従来手法では危険領域を LRF で生成したため、図 3.37(b) が示すように従来手法 (Method 2) では障害物に対しても死角境界座標を生成し、障害物が多々ある環境下では危険領域のコストを過剰に生成してしまった。

提案手法 (Method 3-4) では危険領域は RGB-D カメラにより推定するために、障害物を除外可能であった。そのために、提案手法では危険領域のコスト生成を抑えることができ、移動時間の短縮となった。2 つ目に、提案手法はコスト関数に速度項を追加した。そのために従来よりも速度項を重視することとなり、移動時間の短縮につながった。以上より提案手法の有効性を確認した。

表 3.5 シミュレーション結果 (Case C)

Navigation Method	Cost Function	Goal	Time [sec]
<i>Method 1</i> : ROS Default	eq. (3.1)	×	-
<i>Method 2</i> : ROS Default + BSL (LRF)	eq. (3.2)	○	25.3
<i>Method 3</i> : ROS Default + BSL (RGB-D)	eq. (3.2)	○	20.2
<i>Method 4</i> : ROS Default + BSL (RGB-D)	eq. (3.9)	○	18.3

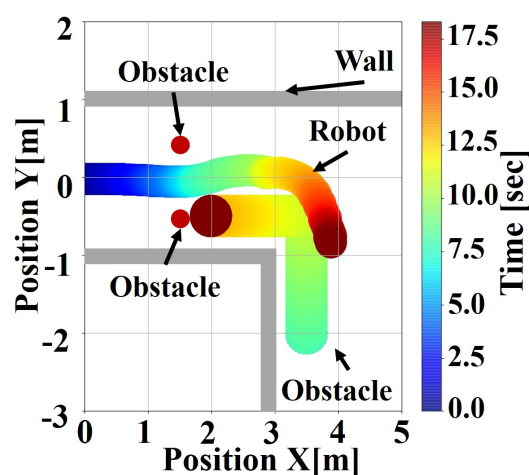
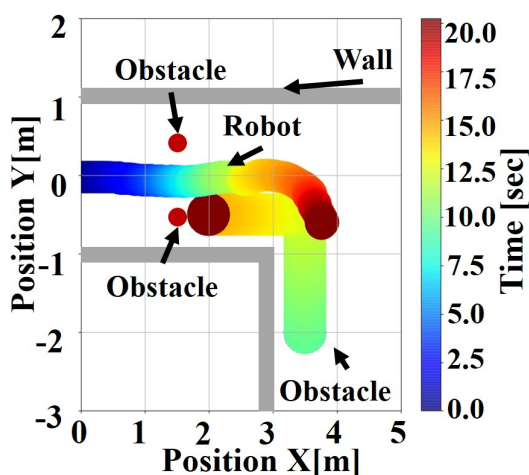
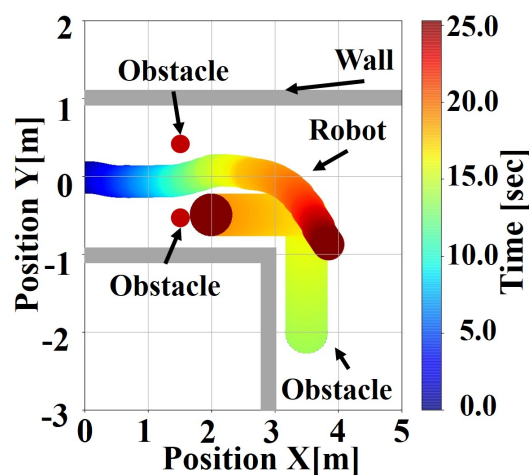
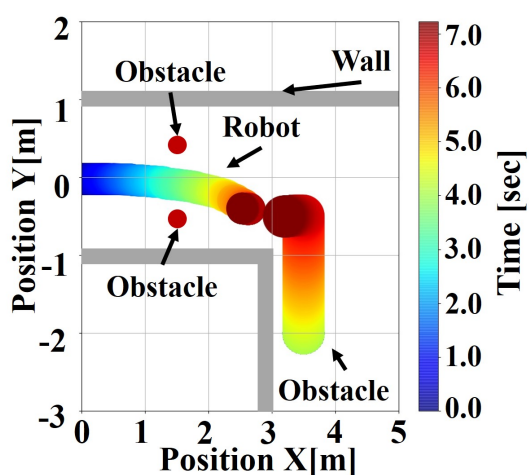


図 3.35 シミュレーション結果 (Case C)

表 3.6 シミュレーション結果 (Case D)

Navigation Method	Cost Function	Goal	Time [sec]
<i>Method 1</i> : ROS Default	eq. (3.1)	×	-
<i>Method 2</i> : ROS Default + BSL (LRF)	eq. (3.2)	○	23.8
<i>Method 3</i> : ROS Default + BSL (RGB-D)	eq. (3.2)	○	21.2
<i>Method 4</i> : ROS Default + BSL (RGB-D)	eq. (3.9)	○	20.5

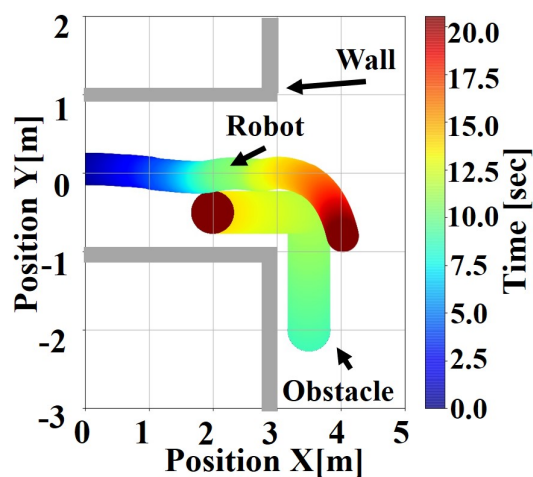
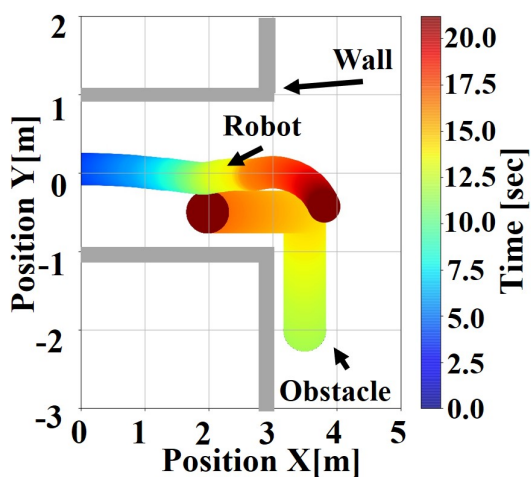
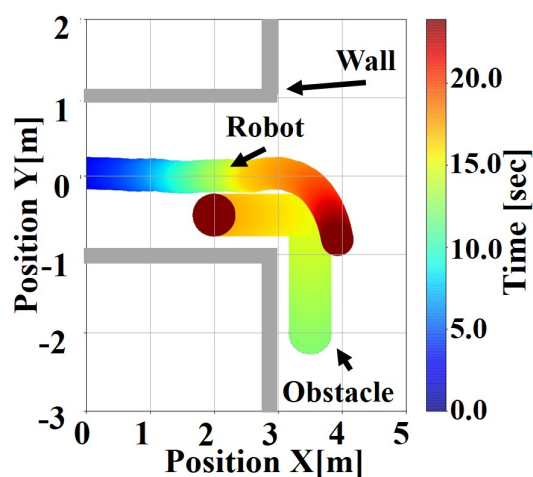
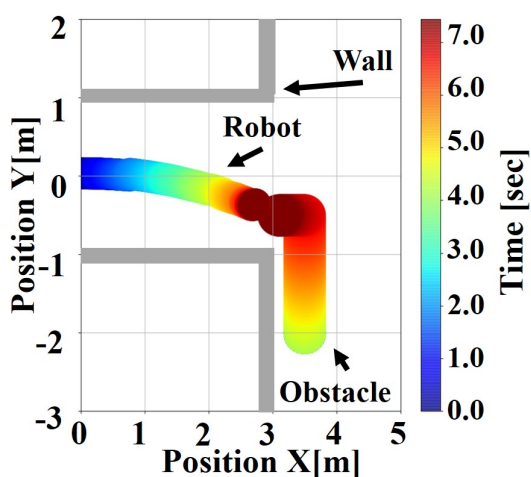
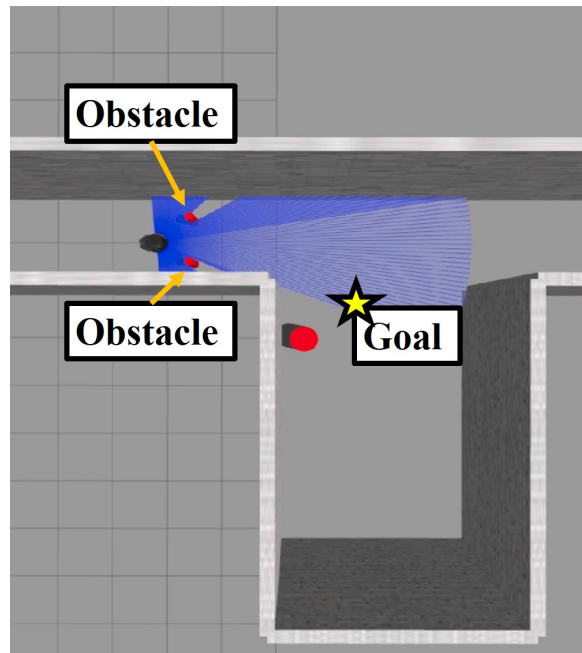
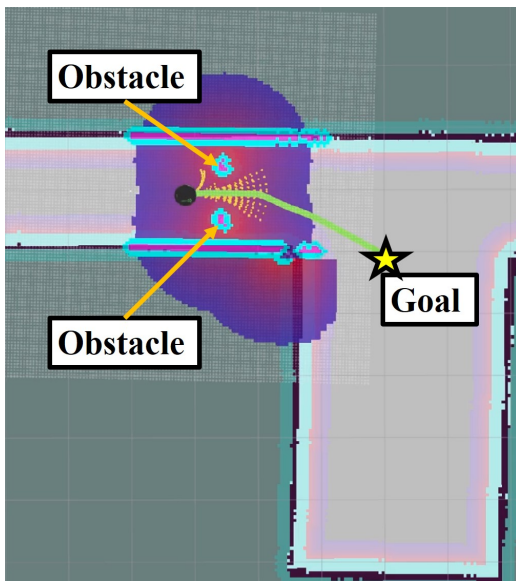


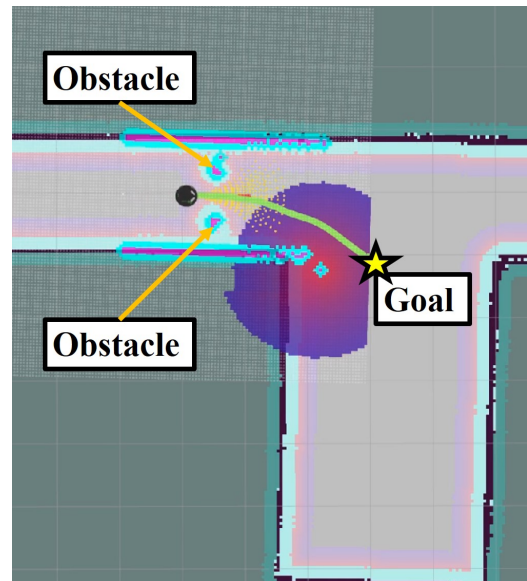
図 3.36 シミュレーション結果 (Case D)



(a) 環境



(b) 従来手法 (Method 2)



(c) 提案手法 (Method 3-4)

図 3.37 コスト生成の様子

3.5 まとめ

人等が混在する生活空間においてロボットと人の共存を考慮した自律移動を実現には、人の行動を妨げず、人に危害を与えないロボット動作が求められる。本章では、人に危害を加える可能性がある場面として、死角領域が発生する場面に着目した。死角領域とは障害物の背後や廊下等での曲がり角の直前等、センサでは計測できない領域のことである。この死角領域が原因によりロボットと人が衝突する可能性が高くなる。

本章では、上記の問題を解決するために、車輪型移動ロボットにおける死角領域を考慮したコストマップに基づく局所経路計画法を提案した。LRF から得られる環境情報を基に、死角境界座標を算出し、死角領域内に人が存在する可能性のある位置を推定する。推定した位置を危険領域の中心座標とし、人が全方向に移動すると仮定して、コストをコストマップに円状に伝播させることにより危険領域を表現した。その後、ダイナミックウィンドウアプローチの経路算出において危険領域のコストを考慮し、死角領域とロボットの運動制約を考慮した経路計画が可能となった。しかしながら LRF では環境情報がセンサ取り付け位置での水平平面の情報しか取得できないために、危険領域を意味するコストを過剰に生成してしまう問題があった。そこで 3次元情報が取得できる RGB-D カメラを用いる事で、LRF よりもロバストな死角領域を考慮した自律移動を実現した。シミュレーションにより提案手法の有用性を確認し、死角領域を考慮した自律移動の実現に躍進したと考える。

第 4 章 速度制約に基づく動的障害物を考慮した自律移動

4.1 はじめに

近年、少子高齢社会の影響によりサービスロボットによる労働力創出が希求されている [58–61]。サービスロボットが人と共存する空間で移動するためには、時々刻々と変化する動的な環境下に対応する必要がある [62–64]。

自律移動ロボットは自己位置推定 [65]、地図生成 [66]、認識 [67]、経路計画 [68] から構成されている。経路計画では静的・動的障害物を考慮した経路生成を行う必要がある。経路計画は大別して 2 つの手法により構成されており、大域的経路計画と局所的経路計画である。大域的経路計画では、事前に作成した地図の情報を利用して目的地まで障害物と衝突しない経路を生成する [69–71]。しかしながら大域的経路計画では、事前地図の情報のみを利用しているために、事前地図にない障害物には対応できない問題がある。そこで局所的経路計画では、リアルタイムでセンサから取得した情報を利用して経路計画を行う。つまり局所的経路計画では、事前地図上にない障害物を考慮した経路計画を行うことが可能である。

本章では局所的経路計画に着目する。静的・動的障害物を考慮した多くの局所的経路計画手法が報告されている [74] [75]。Fiorini 等は、マルチロボットにおける局所的経路計画手法である Velocity Obstacle (VO) を提案した [76]。VO は速度空間で動的障害物を考慮している。Berg 等は VO を拡張し振動のない滑らかな経路探索を実現した [77]。また Xu 等は、ロボットの最高速度より速い障害物速度を考慮した VO を報告している [78]。この他にもマルチロボットのための VO に基づく有用な局所経路計画手法が報告されている [76–79]。VO では障害物回避可能な経路候補を生成する場合、速度空間内で障害物と衝突しない速度を探索する。更に VO では、ロボットの速度は一定と仮定しているために、ロボットの経路候補は直線や円弧経路となる。そのため狭路環境や動的な環境では、障害物と衝突しないロボットの経路候補が減少する。また VO ではロボットのダイナミクスに基づく速度制約を考慮していないため、生成された速度をロボットが実現できない場合も存在する。

ロボットのダイナミクスに基づく速度制約を考慮した局所的経路計画も報告されている。Fox 等は Dynamic Window Approach (DWA) を報告した。DWA はダイナミクス制約を考慮した速度空間 (VSD) を用いて経路候補を生成する。VSD はロボットが現在の速度から生成できる速度範囲を意味する。DWA は VSD 内から生成された経路候補の中から最適な経路を選択することで障害物回避を行いながらゴールへ到達可能である。Dobrevski 等は、DWA と深層強化学習に基づく局所経路計画により、経路最適化を改善することを報告した [81]。Liu 等は、jump-A* アルゴリズムと DWA を組み合わせたグローバルな動的経路計画の融合アルゴリズムを開発した [82]。この他にも DWA に基づく有用な局所的経路計画法が報告されている [80–82]。しかし、DWA は一定時間の間、一定の速度を仮定して経路候補を生成するため経路候補は直線や円弧の経路となる。つまり、経路候補生成時には障害物を考慮していないため、動的障害物等と衝突する経路が多く生成されることとなり、動的な環境や狭路環境下での移動には適さない。

非直線経路や非円弧経路を含む経路候補を生成する局所経路計画手法が存在する [83–85]。Howard 等は State Lattice Planner (SLP) [83] [84] を報告した。SLP は、ロボットの状態とロボットの制約条件のデータセットを用いて、経路候補を生成する。しかし、ロボットの正面付近に障害物がある場合、SLP は経路候補を生成できないことがある。

そこで本章では、Virtual Manipulator (VM) と DWA に基づく局所経路探索手法である Dynamic Window Approach with Virtual Manipulators (DWV) について述べる [86]。VM は反射的に障害物回避動作を生成する手法であり、山崎等によって報告された [87]。DWV は、非直線・非円弧経路を含む経路候補を DWA と VM を用いて生成する。DWV は、VM と静的・動的障害物の予測位置によって修正された可変速度によって経路候補を生成する。そのため静的・動的な障害物が存在する環境において非直線経路や非円弧経路を含む障害物回避可能な経路を生成する。

4.2 Dynamic Window Approach (DWA)

4.2.1 DWA 概要

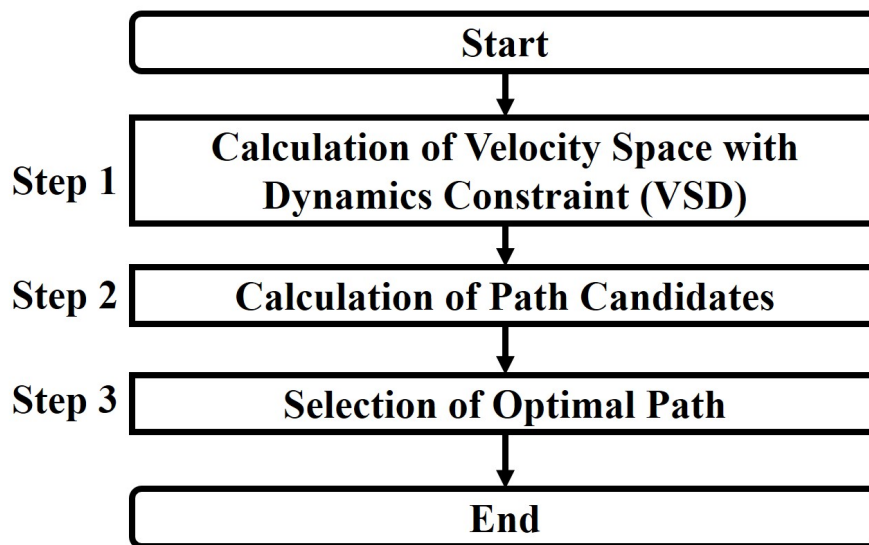


図 4.1 DWA フローチャート

DWA は局所的経路計画の 1 つである。図 4.1 に DWA のフローチャートを示す。DWA は次の 3 つステップにより実行する。

- ダイナミクスを考慮した速度領域の生成 (Step 1)
現在の速度やロボットの加速度から算出される速度領域である。
- 経路候補生成 (Step 2)
Step 1 で算出した速度領域から経路候補を生成する。
- 最適経路選択 (Step 3)
Step 2 で算出した経路候補から最適な経路を選択する。

4.2.2 ダイナミクスを考慮した速度領域

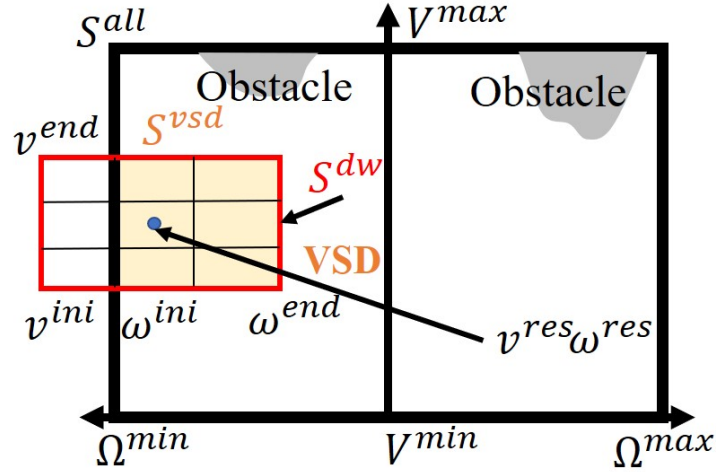


図 4.2 ダイナミクスを考慮した速度領域

図 4.2 は Velocity Space with Dynamics Constraint (VSD) のイメージ図である。VSD は 3つのステップにより生成される。

Step 1-1: 全速度領域 S^{all}

全速度領域 S^{all} は以下のように算出される。

$$S^{all} = \{(v, \omega) \mid v \in [V^{min}, V^{max}] \wedge \omega \in [\Omega^{min}, \Omega^{max}]\} \quad (4.1)$$

ここで、 V^{min} 、 V^{max} 、 Ω^{min} 、 Ω^{max} はロボットが算出可能な最小・最大並進・旋回速度である。

Step 1-2: ダイナミックウィンドウ速度領域 S^{dw}

ダイナミックウィンドウ速度領域 S^{dw} は、ロボットが次のステップで算出可能な速度領域である。

$$\begin{aligned} S^{dw} = & \{(v, \omega) \mid v \in [v^{res} - A^{max} \Delta T, v^{res} + A^{max} \Delta T] \\ & \wedge \omega \in [\omega^{res} - \Pi^{max} \Delta T, \omega^{res} + \Pi^{max} \Delta T]\} \end{aligned} \quad (4.2)$$

ここで v^{res} と ω^{res} は並進・旋回速度応答値であり、現在の速度とみなす。 A^{max} と Π^{max} は最大並進・旋回加速度であり、 ΔT はタイムステップである。

Step 1-3: ダイナミクスを考慮した速度領域 S^{vsd}

ダイナミクスを考慮した速度領域 S^{vsd} は以下のように算出される。

$$S^{vsd} = S^{all} \cap S^{dw} \quad (4.3)$$

$$= \{(v, \omega) \mid v \in [v^{ini}, v^{end}] \wedge \omega \in [\omega^{ini}, \omega^{end}]\} \quad (4.4)$$

ここで v^{ini} と ω^{ini} は S^{vsd} でダイナミクスを考慮した最小並進・旋回速度である。 v^{end} と ω^{end} は S^{vsd} でダイナミクスを考慮した最大並進・旋回速度である。

4.2.3 経路候補生成

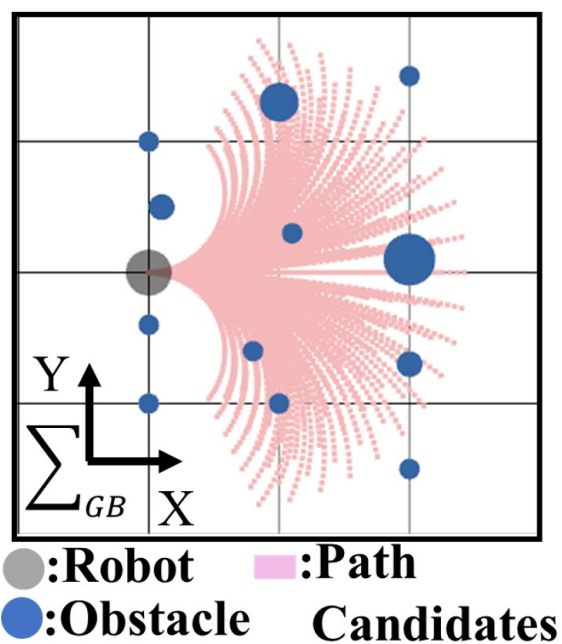


図 4.3 DWA 経路候補生成

Algorithm 1 Path Candidates by DWA

```

1:  $g \leftarrow 1, i \leftarrow 1$ 
2: //  $\blacktriangledown$ Step 2-1a
3: while  $N^{tlv} > g$  do
4:    $h \leftarrow 1$ 
5:    $v^{dwa} \leftarrow v^{ini} + \Delta v(g - 1)$ 
6:   //  $\blacktriangledown$ Step 2-2a
7:   while  $N^{agv} > h$  do
8:      $\omega^{dwa} \leftarrow \omega^{ini} + \Delta \omega(h - 1)$ 
9:     //  $\blacktriangledown$ Step 2-3a
10:    while  $f^{max} > f$  do
11:       $\langle x_f^{rob}, y_f^{rob}, \theta_f^{rob} \rangle \leftarrow onePath(v^{dwa}, \omega^{dwa})$ 
12:       $\langle \mathbf{X}_i^{rob}, \mathbf{Y}_i^{rob}, \Theta_i^{rob} \rangle \leftarrow \langle x_f^{rob}, y_f^{rob}, \theta_f^{rob} \rangle$ 
13:       $f \leftarrow f + 1$ 
14:    end while
15:     $\mathbf{P}_i^{rob} \leftarrow \langle \mathbf{X}_i^{rob}, \mathbf{Y}_i^{rob}, \Theta_i^{rob} \rangle$ 
16:     $\mathbf{O}^{rob} \leftarrow \mathbf{P}_i^{rob}$ 
17:     $i \leftarrow i + 1$ 
18:     $h \leftarrow h + 1$ 
19:  end while
20:   $g \leftarrow g + 1$ 
21: end while
22: return  $\mathbf{O}^{rob}$ 

```

図 4.3 に DWA の経路候補を示す。Algorithm 1 は、DWA による経路候補の擬似コードである。 N^{tlv} , N^{agv} , N^{all} は、並進速度、旋回速度、パス候補の最大数である。DWA は速度空間 S^{vsd} を並進速度軸 N^{tlv} と旋回速度軸 N^{agv} に分割し、合計で $N^{all} (= N^{tlv} \cdot N^{agv})$ の並進速度と旋回速度のペアが生成される。したがって、 N^{all} 個の速度ペアは、 N^{all} 個の経路候補 \mathbf{O}^{rob}

を生成する。ロボットの経路候補 \mathbf{O}^{rob} は以下のように表される。

$$\mathbf{O}^{rob} = [\mathbf{P}_1^{rob} \dots \mathbf{P}_i^{rob} \dots \mathbf{P}_{N^{all}}^{rob}]^T \quad (4.5)$$

$$\mathbf{P}_i^{rob} = [\mathbf{X}_i^{rob} \ \mathbf{Y}_i^{rob} \ \Theta_i^{rob}]^T \quad (4.6)$$

ここで、 \mathbf{P}_i^{rob} は f^{max} 行 3 列の行列であり、 i 番目のロボットの経路候補を示す。 $f^{max}(= \frac{T^{max}}{\Delta T})$ はタイムステップの最大カウント数であり、最大予測時間 T^{max} と時間刻み ΔT から算出される。 \mathbf{X}_i^{rob} , \mathbf{Y}_i^{rob} , Θ_i^{rob} はロボットの位置と姿勢を示す。つまり、 \mathbf{P}_i^{rob} には 1 本の経路候補の位置、姿勢 \mathbf{X}_i^{rob} , \mathbf{Y}_i^{rob} , Θ_i^{rob} が格納されている。Algorithm 1 の詳細について説明する。

Step 2-1a 並進速度選択 (Algorithm 1 lines 2-5)

DWA は速度領域 S^{vsd} を並進速度方向に N^{tlv} 個分割する。DWA では並進速度 v^{dwa} を次のように選択する。

$$v^{dwa} = v^{ini} + \Delta v(g - 1) \quad (4.7)$$

ここで、 Δv は並進速度の変化量、 g ($1 \leq g \leq N^{tlv}$) は並進速度のカウント数である。

Step 2-2a 旋回速度選択 (Algorithm 1 lines 6-8)

DWA は速度領域 S^{vsd} を旋回速度方向に N^{agv} 個分割する。DWA では旋回速度 ω^{dwa} を次のように選択する。

$$\omega^{dwa} = \omega^{ini} + \Delta \omega(h - 1) \quad (4.8)$$

ここで、 $\Delta \omega$ は旋回速度の変化量、 h ($1 \leq h \leq N^{agv}$) は旋回速度のカウント数である。

Step 2-3a 経路生成 (Algorithm 1 lines 9-13)

i 番目の経路候補 \mathbf{P}_i^{rob} は速度ペア $[v^{dwa}, \omega^{dwa}]$ から算出される。Algorithm 1(line 11) の $onePath(v^{dwa}, \omega^{dwa})$ で次のように算出される。

$$\mathbf{X}_i^{rob} = [x_1^{rob} \dots x_f^{rob} \dots x_{f^{max}}^{rob}]^T \quad (4.9)$$

$$\mathbf{Y}_i^{rob} = [y_1^{rob} \dots y_f^{rob} \dots y_{f^{max}}^{rob}]^T \quad (4.10)$$

$$\mathbf{\Theta}_i^{rob} = [\theta_1^{rob} \dots \theta_f^{rob} \dots \theta_{f^{max}}^{rob}]^T \quad (4.11)$$

$$\theta_f = \sum_{k=1}^f \int_{t_{k-1}}^{t_k} \omega^{dwa} dt \quad (4.12)$$

$$x_f = \sum_{k=1}^f \int_{t_{k-1}}^{t_k} v^{dwa} \cdot \cos \theta_k dt \quad (4.13)$$

$$y_f = \sum_{k=1}^f \int_{t_{k-1}}^{t_k} v^{dwa} \cdot \sin \theta_k dt \quad (4.14)$$

ここで、 f ($1 \leq f \leq f^{max}$) はタイムステップのカウンタ数である。

Step 2-1a から Step 2-3a を N^{all} 回行うことで、ロボットの経路候補 \mathbf{O}^{rob} が算出される。

4.2.4 最適経路選択

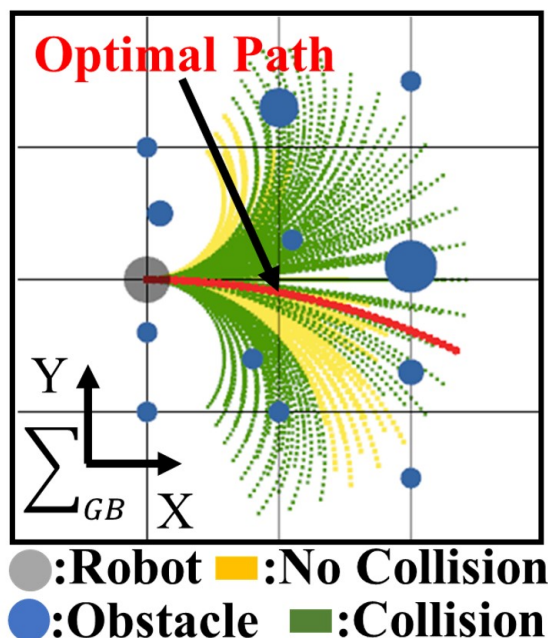


図 4.4 DWA 最適経路選択

図 4.4 に DWA の最適経路選択の様子を示す。ロボットの経路候補 O^{rob} はコスト関数によって評価される。コスト関数は次の様に導出される。

$$c^{dwa} = W^{ang} \cdot c^{ang} + W^{vel} \cdot c^{vel} + W^{obs} \cdot c^{obs} \quad (4.15)$$

ここで W^{ang} , W^{vel} , W^{obs} は重み係数である。 c^{ang} はゴールへの方向を考慮したコスト、 c^{vel} は選択した速度、 c^{obs} は障害物とロボットとの距離である。DWA ではコスト関数 (4.15) を最大化する経路を最適な経路として選択する。

最後に最適な経路の並進速度 v^{opt} 、旋回速度 ω^{opt} とし、ロボットの速度指令値を算出する。

$$v^{cmd} = v^{opt} \quad (4.16)$$

$$\omega^{cmd} = \omega^{opt} \quad (4.17)$$

ここで v^{cmd} 、 ω^{cmd} はロボットの並進・旋回速度指令値である。算出された速度指令値を用いることでロボットは障害物を回避しながらゴールへ到達することが可能となる。

4.3 Virtual Manipulator (VM)

本節では障害物回避手法の一つとして、山崎らが提案した仮想マニピュレータ (VM) を用いた反射的動作生成 [87] [88] について説明する。

4.3.1 VMの動作生成例

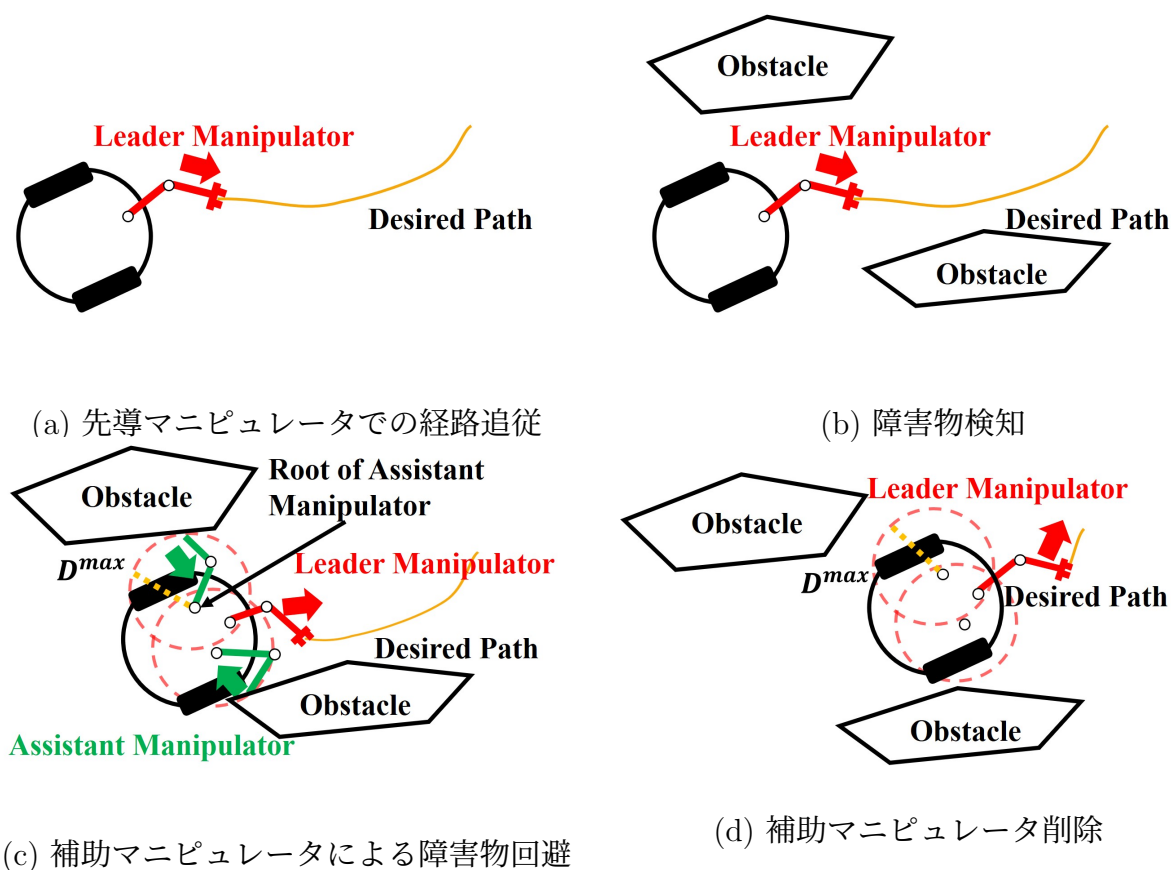


図 4.5 反射的動作生成の概念図

図 4.5 は VM を用いた反射的動作生成による障害物回避の概念図である。VM は移動ロボットに搭載されている。任意の数の VM が移動ロボットに搭載されている。所望の経路の追跡する役割を果たす「先導マニピュレータ」が 1 本搭載されており、他のマニピュレータは、ロボットの周囲の環境と VM の手先を接触させ、障害物回避の役割を果たす「補助マニピュレータ」と呼ぶことにする。この手法では、先導マニピュレータが所望の経路を追従し、補助マニピュレータにより障害物回避動作を生成する。図 4.5(a) では、移動ロボットが

先導マニピュレータによって所望の経路を追従することを示している。図 4.5(b) では、移動ロボットはセンサーを用いて障害物を検出する。図 4.5(c) では、障害物と補助マニピュレータの付け根までの距離が D^{max} よりも小さい障害物が存在する場合に、補助マニピュレータを生成する。 D^{max} は、障害物と移動ロボットの間の距離閾値である。移動ロボットは、補助マニピュレータにより障害物を回避する。図 4.5(d) に示すように、障害物と補助マニピュレータの根元間の距離は D^{max} より大きくなると、補助マニピュレータを削除する。従って、ロボットは所望の経路を追従しながら、障害物を回避することが可能である。

4.3.2 VM を用いた反射的動作生成

自由度数の合計が m 個とした VM を 1 本搭載したロボットの状態ベクトルを $\mathbf{q} = [q_1, q_2, \dots, q_m]$ とする。VM の手先状態を $\mathbf{p}_e = [x_e, y_e, \theta_e]$ とし、 $\dot{\mathbf{p}}_e$ と $\dot{\mathbf{q}}$ の関係は式 (4.18) で表すことができる。

$$\dot{\mathbf{p}}_e = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \quad (4.18)$$

ここで、 $\mathbf{J}(\mathbf{q})$ はヤコビ行列である。ヤコビ行列は次のように算出される。

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} \frac{\partial x_e}{\partial q_1} & \frac{\partial x_e}{\partial q_2} & \dots & \frac{\partial x_e}{\partial q_m} \\ \frac{\partial y_e}{\partial q_1} & \frac{\partial y_e}{\partial q_2} & \dots & \frac{\partial y_e}{\partial q_m} \\ \frac{\partial \theta_e}{\partial q_1} & \frac{\partial \theta_e}{\partial q_2} & \dots & \frac{\partial \theta_e}{\partial q_m} \end{bmatrix} \quad (4.19)$$

4.3.3 移動マニピュレータのヤコビ行列（全方位ロボット）

全方向に移動可能な全方位ロボットのヤコビ行列は式 (4.20) で表される。

$$\mathbf{J}_{omn} = \begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b & \mathbf{J}_v \\ 0 & 0 & 1 \end{bmatrix} \quad (4.20)$$

ここで、 \mathbf{J}_v は、ロボットや VM を考慮したヤコビ行列である。この \mathbf{J}_{omn} を用いれば、手先に与えられた速度 $\dot{\mathbf{p}}_e = [\dot{x}_e, \dot{y}_e, \dot{\theta}_e]$ からロボットの並進速度・旋回速度とマニピュレータの各関節角速度を算出可能となる。

図 4.6 のように回転関節 $[\theta_{01}, \theta_{02}]$ を 2 つ持つ 2 リンクマニピュレータの場合、ヤコビ行列は式 (4.21) のように 3 行 5 列となる。

$$\mathbf{J}_{omn} = \begin{bmatrix} 1 & 0 & a & c & e \\ 0 & 1 & b & d & f \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \quad (4.21)$$

ここで、 $e = -l_2 \sin(\theta + \theta_{01} + \theta_{02})$, $f = l_2 \cos(\theta + \theta_{01} + \theta_{02})$, $c = e - l_1 \sin(\theta + \theta_{01})$, $d = f + l_1 \cos(\theta + \theta_{01})$, $a = c - x_r \sin \theta - y_r \cos \theta$, $b = d + x_r \cos \theta - y_r \sin \theta$ である。 x_r, y_r はマニピュレータの根本座標値であり、 l_1, l_2 はマニピュレータのリンク長である。

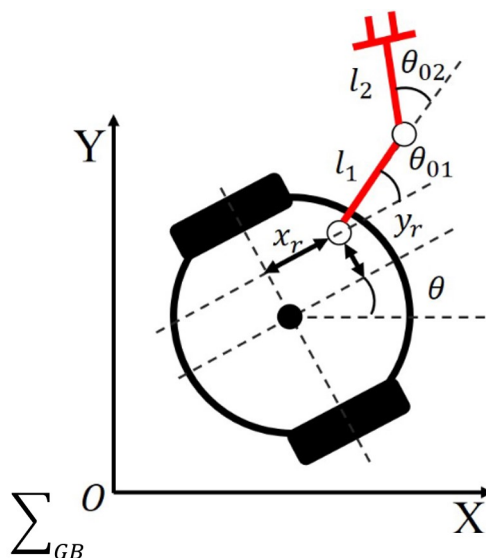


図 4.6 仮想マニピュレータを搭載したロボット

4.3.4 二輪駆動ロボット: 移動マニピュレータのヤコビ行列

二輪駆動ロボットのノンホロミックな拘束を含めたヤコビ行列は、 \mathbf{J}_{omn} を行列 $\mathbf{U}(\theta)$ と掛け合わせることで生成可能である。

$$\mathbf{U}(\theta) = \begin{bmatrix} \cos \theta & 0 & 0 \\ \sin \theta & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \mathbf{I}_2 \end{bmatrix} \quad (4.22)$$

具体的には、 $\mathbf{J}_{pws} = \mathbf{J}_{omn}\mathbf{U}$ とする。ここで、 \mathbf{I} は単位行列、下付添字の 2 はマニピュレータの関節の数である。この演算により、ノンホロミックな拘束に基づく並進速度 v と旋回速度 ω に対応するよう変換可能となる。これより、手先姿勢 $\dot{\mathbf{p}}_e = [\dot{x}_e, \dot{y}_e, \dot{\theta}_e]$ への速度入力によるロボット動作は次のように算出される。

$$\begin{bmatrix} v^{vm} \\ \omega^{vm} \\ \dot{\theta}_{01} \\ \dot{\theta}_{02} \end{bmatrix} = \mathbf{J}_{pws}^+ \dot{\mathbf{p}}_e \quad (4.23)$$

\mathbf{J}_{pws}^+ は \mathbf{J}_{pws} の疑似逆行列であり、ロボットのノンホロミックの拘束を考慮した並進速度・旋回速度とマニピュレータの関節角速度を計算することが可能となる。

4.3.5 反射的動作の調整

反射的な動作の調整を行うために以下のような対策をする。

- 零空間を用いた最適化

ロボットと周囲の環境との距離を一定に保つために、VM の所望関節角度に基づく評価関数を設定する。また VM が特異姿勢に陥るのを防ぐ役割を担う。

- 重み行列 \mathbf{W}

重み行列 \mathbf{W} を用いることで、ロボットの動作量の調整を行う。

以上から VM を用いたロボットの反射的動作生成を式 (4.24) のように構成する。

$$\dot{\mathbf{q}} = \mathbf{J}_w^+ \dot{\mathbf{x}} + \Lambda(\mathbf{I} - \mathbf{J}_w^+ \mathbf{J})(\Theta_{nj}^{ref} - \theta_{nj}^{res}) \quad (4.24)$$

ここで、 $\dot{\mathbf{q}} = [v^{vm}, \omega^{vm}, \dot{\theta}_{01}, \dot{\theta}_{02}, \dot{\theta}_{11}, \dot{\theta}_{12}, \dots, \dot{\theta}_{n1}, \dot{\theta}_{n2}]^T$ はロボットとマニピュレータの各関節角の速度・角速度であり、 $\dot{\mathbf{x}}$ は手先目標速度である。 v^{vm} と ω^{vm} は VM により生成されたロボットの並進速度と旋回速度である。 $\dot{\theta}_{n1}$ と $\dot{\theta}_{n2}$ は n 番目の VM の角速度である。 \mathbf{J}_w^+ は重み行列 \mathbf{W} を掛けたものである。 Λ は零空間の重み係数である。 Θ_{nj}^{ref} と θ_{nj}^{res} は関節角の参照値と応答値を示す。 $j(= 1, 2)$ は第一関節と第二関節を示す。

4.3.6 実際の動作例

図 4.7 に VM の動作例を示す。図 4.7 の左側には物理シミュレーター内のロボットと障害物の様子を示している。また青色の線は LRF の環境取得範囲を示している。右側は可視化ツール Rviz でロボットと障害物を可視化している。また VM は左右 1 本ずつ搭載されており、左の VM を黒色、右の VM を緑色とした。ロボットから出ている赤色の線は VM により移動する経路を示している。図 4.7 が示すように、LRF 等のセンサから取得した情報を用いて、ロボット近傍に障害物がある場合には VM により反射的な動作生成を行う。

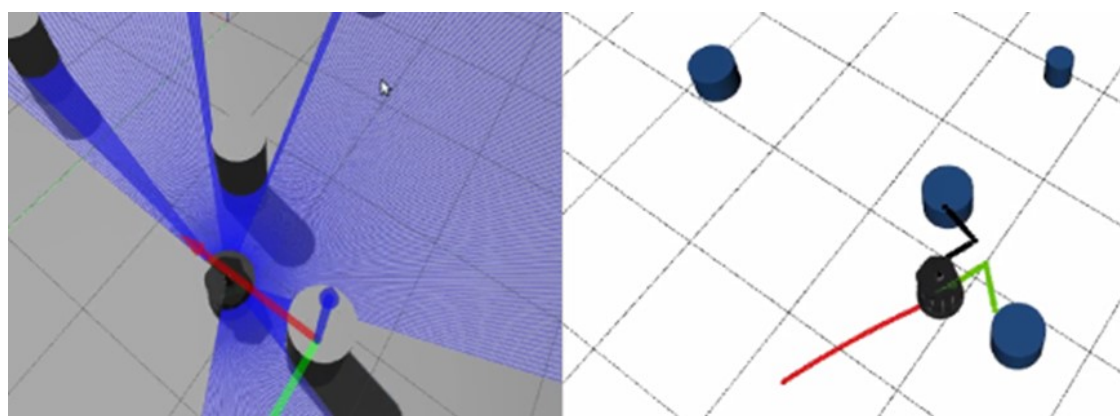


図 4.7 仮想補助マニピュレータ生成

4.4 Dynamic Window Approach with Virtual Manipulators (DWV)

4.4.1 DWV 概要

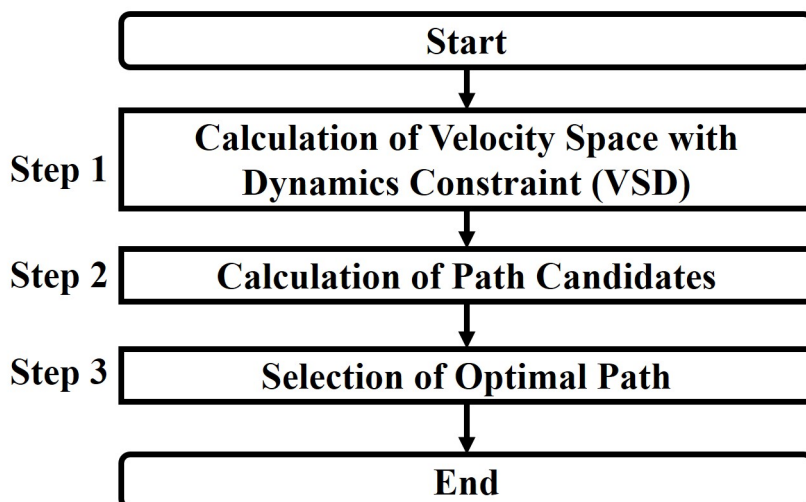


図 4.8 DWV フローチャート

図 4.1、4.8 が示すように、DWV のフローチャートは DWA と同じである。DWV は DWA から経路候補生成 (Step 2) と最適経路算出 (Step 3) を変更している。

- ダイナミクスを考慮した速度領域 (Step 1)
現在の速度やロボットの加速度から算出される速度領域である。
- 経路候補生成 (Step 2)
Step 1 で算出した速度領域から経路候補を生成する。DWV では VM と障害物の予測経路を用いて経路候補を算出する。
- 最適経路選択 (Step 3)
Step 2 で算出した経路候補から最適な経路を選択する。

4.4.2 経路候補生成 (Step 2)

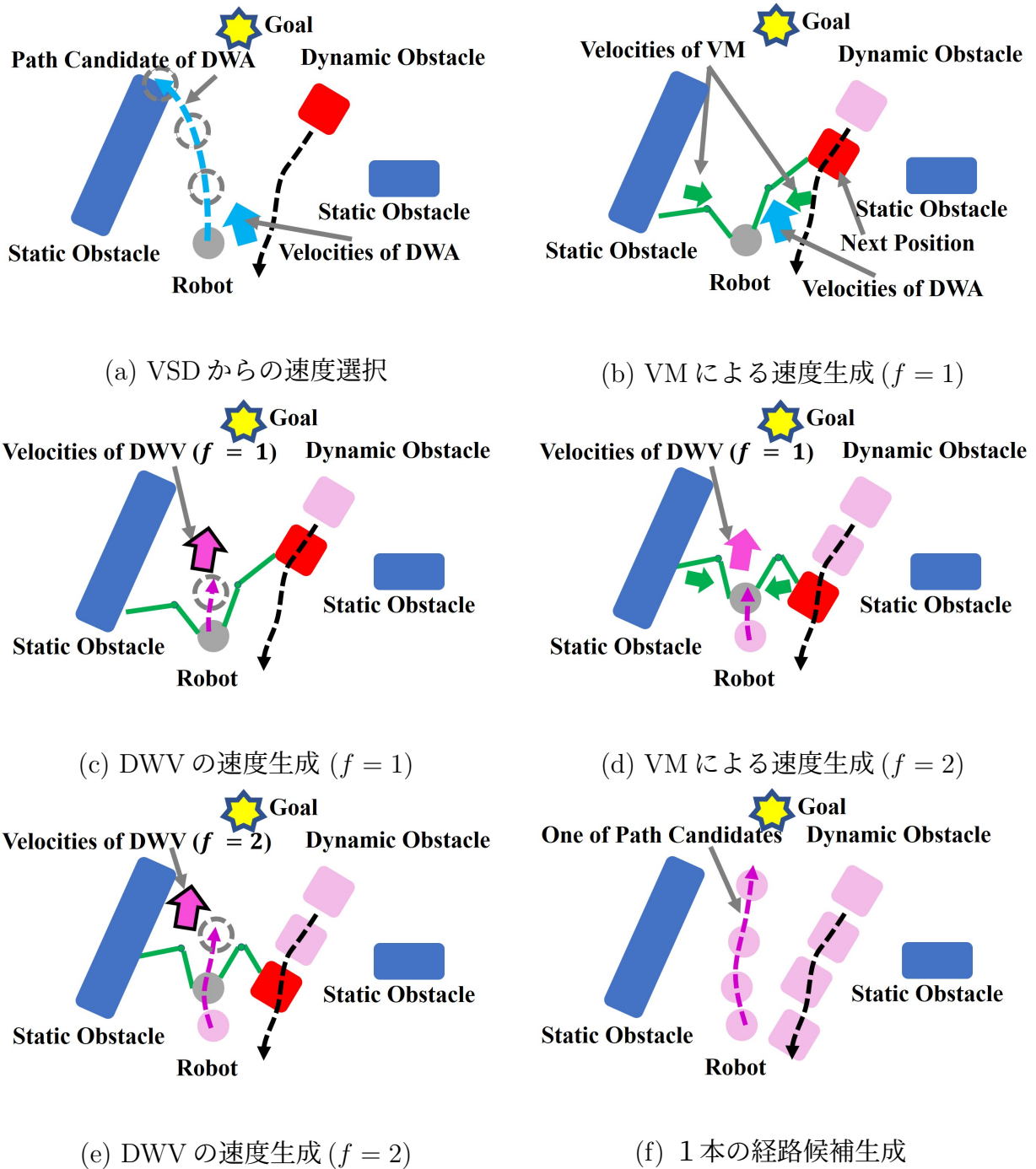


図 4.9 DWV 経路候補生成の概念図

図 4.9 は DWV の経路候補生成の概念図を示している。

1. 初めに DWV では DWA と同様に VSD から速度を選択する。DWA では VSD から選択した一定の速度を使用して経路候補を算出する。したがって DWA の経路候補は円弧の経路となる。障害物を考慮した経路候補を生成するために次ステップでは VM を用いた動作生成を行う (図 4.9(a))。
2. DWV はステップ f 毎に VM を用いて障害物回避可能な経路を生成する。VM は障害物の次ステップの予測位置から障害物回避を考慮した速度を生成する (図 4.9(b))。
3. ($f = 1$) の時の DWV の速度は DWA と VM から生成する (図 4.9(c))。
4. VM は障害物の次ステップの予測位置から障害物回避を考慮した速度を生成する (図 4.9(d))。
5. ($f = 2$) の時の DWV の速度は 1 ステップ前 ($f = 1$) の DWV の速度と VM の速度により生成される (図 4.9(e))。
6. この処理を f^{max} 回繰り返すことで、DWV の経路候補の一つ P_i^{rob} が生成される (図 4.9(f))。

これら操作を N^{all} の速度ペアで行うことで図 4.10 のように経路候補が生成される。

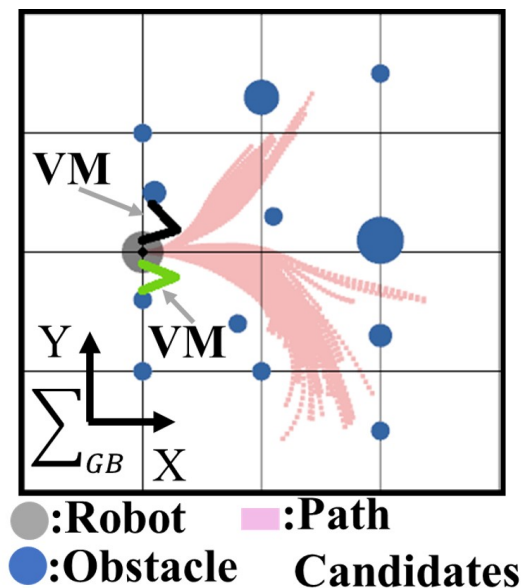


図 4.10 DWV 経路候補

Algorithm 2 は DWV の経路候補算出についての疑似コードを示す (Step 2)。DWV の経路候補生成方法について Algorithm 2 を用いて説明する。

Algorithm 2 Path Candidates by DWV

```

1:  $g \leftarrow 1, i \leftarrow 1$ 
2: //  $\blacktriangledown$ Step 2-1b
3: while  $N^{tlv} > g$  do
4:    $v^{dwa} \leftarrow v^{ini} + \Delta v(g - 1)$ 
5:    $h \leftarrow 1$ 
6:   //  $\blacktriangledown$ Step 2-2b
7:   while  $N^{agv} > h$  do
8:      $\omega^{dwa} \leftarrow \omega^{ini} + \Delta\omega(h - 1)$ 
9:      $f \leftarrow 1$ 
10:    while  $f^{max} > f$  do
11:      //  $\blacktriangledown$ Step 2-3b
12:       $\omega_f^{vm} \leftarrow calcVM(\mathbf{O}^{obs})$ 
13:      //  $\blacktriangledown$ Step 2-4b
14:       $v^{dvw} \leftarrow v^{dwa}$ 
15:      if  $f = 1$  then
16:         $\omega_f^{dvw} \leftarrow \omega^{dwa} + \omega_f^{vm}$ 
17:      else
18:         $\omega_f^{dvw} \leftarrow \omega_{f-1}^{dvw} + \omega_f^{vm}$ 
19:      end if
20:      //  $\blacktriangledown$ Step 2-5b
21:       $\langle v^{dvw}, \omega_f^{dvw} \rangle \leftarrow checkVelocity(v^{dvw}, \omega_f^{dvw})$ 
22:      //  $\blacktriangledown$ Step 2-6b
23:       $\langle x_f^{rob}, y_f^{rob}, \theta_f^{rob} \rangle \leftarrow onePath(v^{dvw}, \omega_f^{dvw})$ 
24:       $\langle \mathbf{X}_i^{rob}, \mathbf{Y}_i^{rob}, \Theta_i^{rob} \rangle \leftarrow \langle x_f^{rob}, y_f^{rob}, \theta_f^{rob} \rangle$ 
25:       $f \leftarrow f + 1$ 
26:    end while
27:     $\mathbf{P}_i^{rob} \leftarrow \langle \mathbf{X}_i^{rob}, \mathbf{Y}_i^{rob}, \Theta_i^{rob} \rangle$ 
28:    //  $\blacktriangledown$ Step 2-7b
29:    if  $collision(\mathbf{O}^{obs}, \mathbf{P}_i^{rob})$  then
30:       $delete(\mathbf{P}_i^{rob})$ 
31:    else
32:       $\mathbf{O}^{rob} \leftarrow \mathbf{P}_i^{rob}$ 
33:       $i \leftarrow i + 1$ 
34:    end if
35:     $h \leftarrow h + 1$ 
36:  end while
37:   $g \leftarrow g + 1$ 
38: end while
39: return  $\mathbf{O}^{rob}$ 

```

Step 2-1b 並進速度選択 (Algorithm 2 lines 2-4)

DWV は速度領域 S^{vsd} を並進速度方向に N^{tlv} 個分割する。DWV では並進速度 v^{dwa} を次のように選択する。

$$v^{dwa} = v^{ini} + \Delta v(g - 1) \quad (4.25)$$

Step 2-2b 旋回速度選択 (Algorithm 2 lines 6-8)

DWV は速度領域 S^{vsd} を旋回速度方向に N^{agv} 個分割する。DWV では旋回速度 ω^{dwa} を次のように選択する。

$$\omega^{dwa} = \omega^{ini} + \Delta\omega(h - 1) \quad (4.26)$$

Step 2-3b 仮想マニピュレータによる速度生成 (Algorithm 2 lines 11-12)

DWV では、静的・動的障害物を考慮した速度生成を障害物の予測位置 \mathbf{O}^{obs} と VM により生成する。障害物の予測位置 \mathbf{O}^{obs} は次のように示す。

$$\mathbf{O}^{obs} = [\mathbf{P}_1^{obs} \dots \mathbf{P}_s^{obs} \dots \mathbf{P}_{s^{max}}^{obs}]^T \quad (4.27)$$

$$\mathbf{P}_s^{obs} = [\mathbf{X}_s^{obs} \ \mathbf{Y}_s^{obs}]^T \quad (4.28)$$

$$\mathbf{X}_s^{obs} = [x_{s,1}^{obs} \dots x_{s,f}^{obs} \dots x_{s,f^{max}}^{obs}]^T \quad (4.29)$$

$$\mathbf{Y}_s^{obs} = [y_{s,1}^{obs} \dots y_{s,f}^{obs} \dots y_{s,f^{max}}^{obs}]^T \quad (4.30)$$

ここで、 s ($1 \leq s \leq s^{max}$) は障害物の数を表す。DWV では VM は補助マニピュレータのみで構成される。 f 番目の障害物の位置と VM の根本位置の距離 d_f^{rob} から、(4.24) を用いて旋回速度 ω^{vm} を生成する。 f 番目の旋回速度 ω_f^{vm} は次のように生成される。

$$\omega_f^{vm} = \begin{cases} \omega^{vm}, & \text{if } d_f^{rob} \leq D^{max} \\ 0, & \text{otherwise} \end{cases} \quad (4.31)$$

Step 2-4b 速度合成 (Algorithm 2 lines 13-19)

f 番目の DWV の速度は次のように生成される。

$$v^{dvw} = v^{dwa} \quad (4.32)$$

$$\omega_f^{dvw} = \begin{cases} \omega^{dwa} + \omega_f^{vm}, & \text{if } f = 1 \\ \omega_{f-1}^{dvw} + \omega_f^{vm}, & \text{otherwise} \end{cases} \quad (4.33)$$

ここで、 v^{dvw} と ω_f^{dvw} は DWV の並進速度と旋回速度を示す。

Step 2-5b 速度制限 (Algorithm 2 lines 20-21)

DWV の f 番目の速度である v^{dvw} と ω_f^{dvw} を f 番目速度領域 S_f^{dvw} 内に存在するか確認する。 f 番目の速度領域 S_f^{dvw} は f 番目の S_f^{pdw} と S^{all} から構成される。 S_f^{dvw} は以下のように算出される。

$$S_f^{pdw} = \{(v, \omega) | v \in [v^{dvw} - A^{max} \Delta T, v^{dvw} + A^{max} \Delta T] \\ \wedge \omega \in [\omega_{f-1}^{dvw} - \Pi^{max} \Delta T, \omega_{f-1}^{dvw} + \Pi^{max} \Delta T]\} \quad (4.34)$$

$$S_f^{dvw} = \begin{cases} S^{all} \cap S^{dw}, & \text{if } (f = 1) \\ S^{all} \cap S_f^{pdw}, & \text{otherwise} \end{cases} \quad (4.35)$$

$$= \{(v, \omega) | v \in [v_f^{ini}, v_f^{end}] \wedge \omega \in [\omega_f^{ini}, \omega_f^{end}]\} \quad (4.36)$$

ここで v_f^{ini} 、 ω_f^{ini} は速度領域 S_f^{dvw} 内の最小並進・旋回速度である。 v_f^{end} 、 ω_f^{end} は速度領域 S_f^{dvw} 内の最大並進・旋回速度である。 v^{dvw} か ω_f^{dvw} が速度領域 S_f^{dvw} 内に含まれない場合には、 v^{dvw} と ω_f^{dvw} は次の様に算出される。

$$v^{dvw} = \begin{cases} v_f^{ini}, & \text{if } v^{dvw} < v_f^{ini} \\ v_f^{end}, & \text{if } v^{dvw} > v_f^{end} \\ v^{dvw}, & \text{otherwise} \end{cases} \quad (4.37)$$

$$\omega_f^{dvw} = \begin{cases} \omega_f^{ini}, & \text{if } \omega_f^{dvw} < \omega_f^{ini} \\ \omega_f^{end}, & \text{if } \omega_f^{dvw} > \omega_f^{end} \\ \omega_f^{dvw}, & \text{otherwise} \end{cases} \quad (4.38)$$

Step 2-6b 経路生成 (Algorithm 2 lines 22-24)

ロボットの位置は v^{dvw} と ω_f^{dvw} から算出し、 i 番目の経路候補は次のように算出する。

$$\mathbf{X}_i^{rob} = [x_1^{rob} \dots x_f^{rob} \dots x_{fmax}^{rob}]^T \quad (4.39)$$

$$\mathbf{Y}_i^{rob} = [y_1^{rob} \dots y_f^{rob} \dots y_{fmax}^{rob}]^T \quad (4.40)$$

$$\mathbf{\Theta}_i^{rob} = [\theta_1^{rob} \dots \theta_f^{rob} \dots \theta_{fmax}^{rob}]^T \quad (4.41)$$

$$\theta_f = \sum_{k=1}^f \int_{t_{k-1}}^{t_k} \omega_k^{dvw} dt \quad (4.42)$$

$$x_f = \sum_{k=1}^f \int_{t_{k-1}}^{t_k} v^{dvw} \cdot \cos \theta_k dt \quad (4.43)$$

$$y_f = \sum_{k=1}^f \int_{t_{k-1}}^{t_k} v^{dvw} \cdot \sin \theta_k dt \quad (4.44)$$

v^{dvw} と ω_f^{dvw} を用いることで、DWV の経路候補には非円弧経路が含まれる。これは、旋回速度 ω_f^{dvw} を各ステップ毎に障害物の状態に合わせて変更しているからである。

Step 2-7b 衝突確認 (Algorithm 2 line 28-34)

ロボットが障害物と衝突しているか確認する。ロボットが障害物と衝突している経路を経路候補に含めないこととする。

Step 2-1b から Step 2-7b を N^{all} 回繰り返すことで、ロボットの経路候補 \mathbf{O}^{rob} が算出される。

4.4.3 Optimal Path (Step 3)



図 4.11 DWV 最適な経路選択

図 4.11 に DWV の最適経路選択の様子を示す。ロボットの経路候補 O^{rob} はコスト関数によって評価される。コスト関数は次の様に導出される。

$$c^{dvw} = W^{pos} \cdot c^{pos} + W^{vel} \cdot c^{vel} + W^{sdo} \cdot c^{sdo} \quad (4.45)$$

ここで、 W^{pos} と W^{sdo} は位置と障害物に関する重み係数である。 c^{pos} はロボットの予測位置と目的地までの距離、 c^{sdo} はロボットと障害物の予測位置との最短距離である。

最後にコスト関数を最大化する経路の並進速度 v^{opt} 、旋回速度 ω_1^{opt} とし、ロボットの速度指令値を算出する。

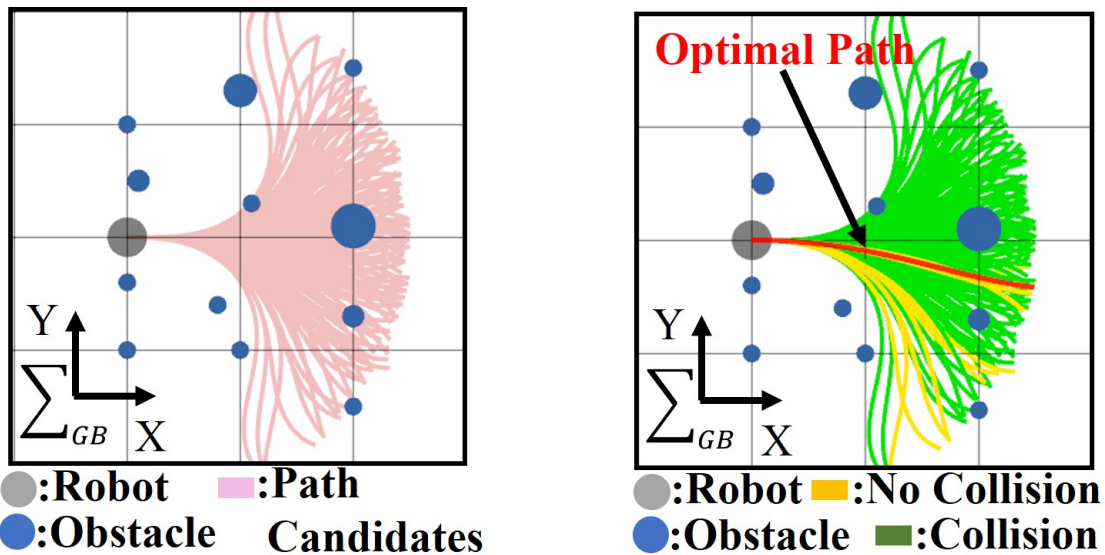
$$v^{cmd} = v^{opt} \quad (4.46)$$

$$\omega^{cmd} = \omega_1^{opt} \quad (4.47)$$

ここで v^{cmd} 、 ω^{cmd} はロボットの並進・旋回速度指令値である。算出された速度指令値を用いることでロボットは障害物を回避しながらゴールへ到達することが可能となる。

4.5 シミュレーション

4.5.1 局所経路計画手法



(a) 経路候補生成

(b) 最適な候補選択

図 4.12 SLP の概要図

本シミュレーションでは、“SLP”、“DWA”、“DWV”の3つの手法を比較した。SLPとDWAを従来手法、DWVを提案手法とした。表4.1にシミュレーションで使用したパラメータを記載する。本シミュレーションではROSを使用している[89–92]。図4.12にSLPの経路生成の様子と最適な経路選択の図を示す。本シミュレーションではDWVに2つのVMを搭載し、VMの左右根本位置をロボット座標系において(0.0, -0.1)と(0.0, 0.1)とした。

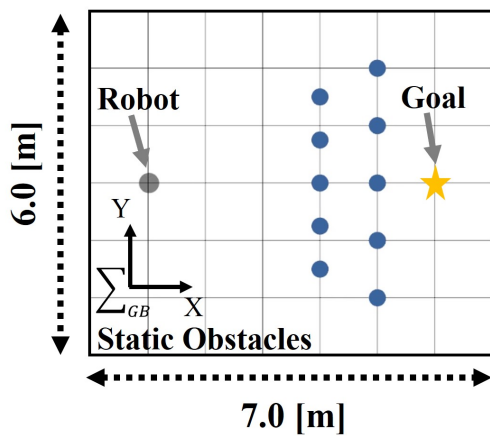
表 4.1 パラメータ

V^{max}	Maximum Translational Velocity	0.55 [m/s]
V^{min}	Minimize Translational Velocity	-0.3 [m/s]
Ω^{max}	Maximum Angular Velocity	5 [rad/s]
Ω^{min}	Minimize Angular Velocity	-5 [rad/s]
A^{max}	Maximum Translational Acceleration	2 [m/s ²]
Π^{max}	Maximum Angular Acceleration	5 [rad/s ²]
N^{tlv}	Number of Translational Velocity	6
N^{agv}	Number of Angular Velocity	20
T^{max}	Maximum Predicted Time	4 [s]
ΔT	Time Step	0.1 [s]
W^{ang}	Weight Coefficient of Angle	1
W^{vel}	Weight Coefficient of Velocity	5
W^{obs}	Weight Coefficient of Obstacle	0.1
W^{pos}	Weight Coefficient of Position	20
W^{sdo}	Weight Coefficient of Static and Dynamic Obstacle	0.1
D^{max}	Generation Value of Virtual Manipulators	0.5 [m]
L^{ass}	Length of Virtual Manipulators	0.3 [m]
Θ_{11}^{ref}	Reference 1st Angle of 1st Virtual Manipulators	90 [deg]
Θ_{12}^{ref}	Reference 2nd Angle of 1st Virtual Manipulators	-180 [deg]
Θ_{21}^{ref}	Reference 1st Angle of 2nd Virtual Manipulators	-90 [deg]
Θ_{22}^{ref}	Reference 2nd Angle of 2nd Virtual Manipulators	180 [deg]
Λ	Weight Coefficients of Virtual Manipulators	0.075
N^{pos}	Number of Samples in Terminal State Position	40
N^{hea}	Number of Samples in Terminal State Heading	3
L^{pat}	Path Length of SLP	2.5[m]
Υ^{min}	Minimum Angular Range of Terminal Position	-60[deg]
Υ^{max}	Maximum Angular Range of Terminal Position	60[deg]
Φ^{min}	Minimum Angular Range of Heading Angle	-30[deg]
Φ^{max}	Maximum Angular Range of Heading Angle	30[deg]

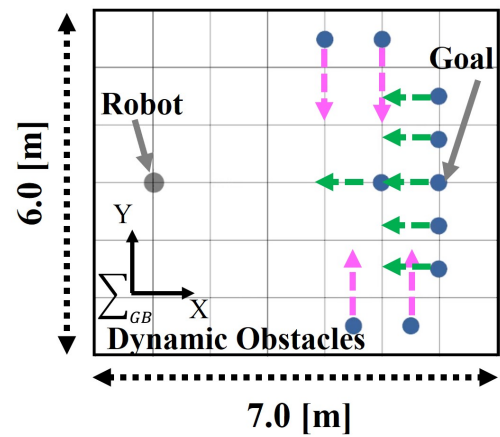
4.5.2 シミュレーション設定

表 4.2 シミュレーション条件

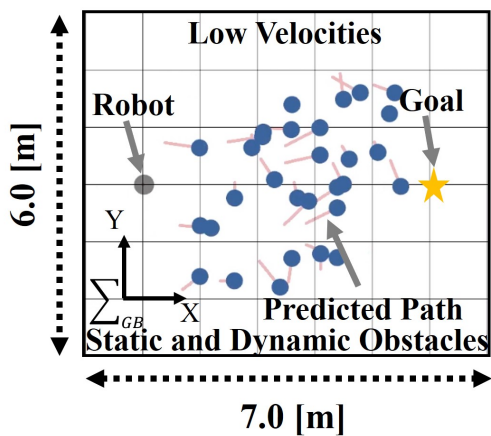
Case	Obstacle	Initial Position	Obstacle Velocity
Case S1	Static Obstacles	Constant	0
Case S2	Dynamic Obstacles	Constant	Constant
Case S3	Static / Dynamic Obstacles	Random	Random (Low)
Case S4	Static / Dynamic Obstacles	Random	Random (High)



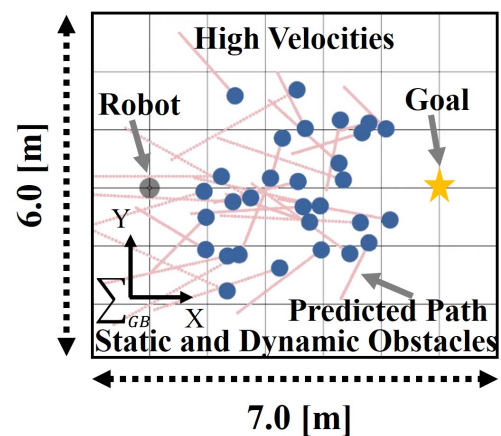
(a) Case S1



(b) Case S2



(c) Case S3



(d) Case S4

図 4.13 シミュレーション環境

表 4.2 が示すように 4 パターンのシミュレーションを行った。図 4.13 は *Case S1* から *Case S4* のシミュレーション環境を示した。図 4.13 の灰色と青色の円はロボットと障害物である。図 4.13(c)-(d) ピンク色の線は障害物の移動方向である。全てのケースでのスタート位置とゴール位置は $(^{GB}x^{start}, ^{GB}y^{start}) = (0.0, 0.0)$ と $(^{GB}x^{goal}, ^{GB}y^{goal}) = (5.0, 0.0)$ とした。ロボットの位置とゴール位置が 0.3 [m] 以下になれば、ゴールに到達したとみなした。*Case S1* から *Case S4* の概要を次に示す。

- *Case S1*: 静的障害物

図 4.13(a) が示すように、10 個の静的障害物を設置した。

- *Case S2*: 動的障害物

図 4.13(b) が示すように、10 個の動的障害物を設置した。障害物は 0.3 [m/s] (緑) と 0.25[m/s] (ピンク) の速度で移動することとした。

- *Case S3*: 静的/動的障害物 (低速)

図 4.13(c) が示すように、30 個の静的/動的障害物を設置した。障害物の位置と速度はシミュレーション開始時にランダムに設定することとした。障害物の速度はロボットの最高速度より小さい 0.0 [m/s] から 0.2 [m/s] の範囲でランダムで選択された。

- *Case S4*: 静的/動的障害物 (高速)

図 4.13(d) が示すように、30 個の静的/動的障害物を設置した。障害物の位置と速度はシミュレーション開始時にランダムに設定することとした。障害物の速度はロボットの最高速度より大きい 0.0 [m/s] から 0.6 [m/s] の範囲でランダムで選択された。

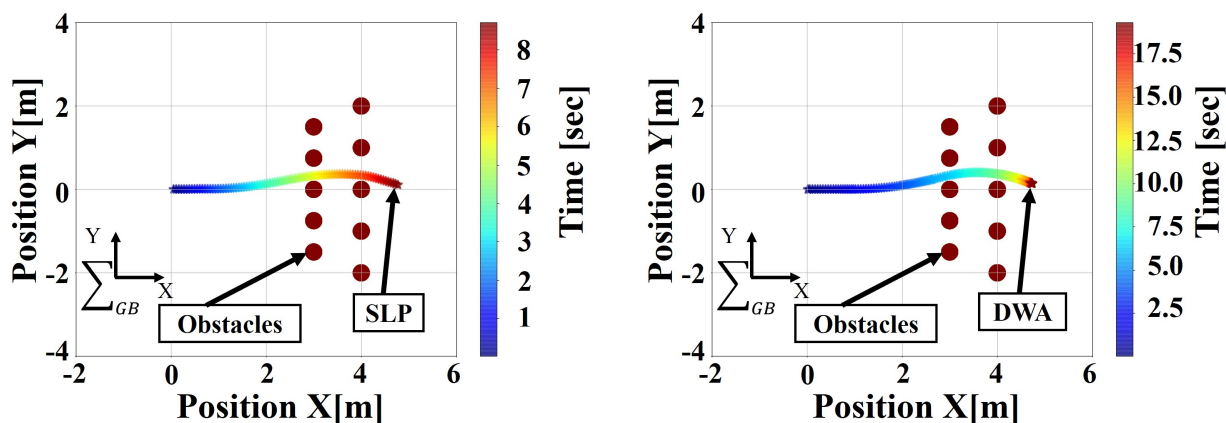
Case S1 と *Case S2* ではシミュレーション回数を 1 回行った。*Case S3* と *Case S4* ではシミュレーション回数を 100 回行った。

4.5.3 シミュレーション結果

表 4.3 シミュレーション結果

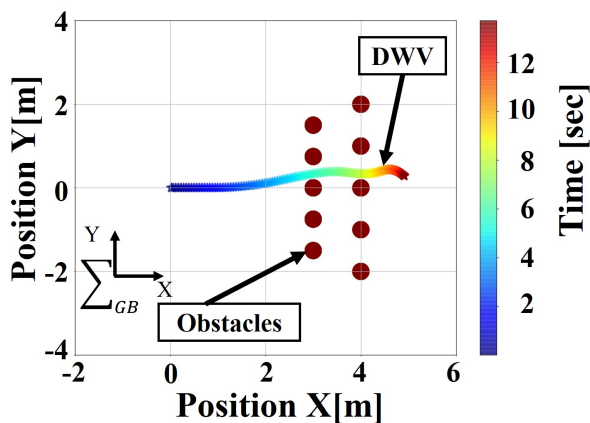
Case	Method	Success [%]	Time [sec]	TL [m]	PD [rad]
<i>Case S1</i> 1 time	SLP	100	8.708	4.835	1.324
	DWA	100	19.301	4.814	0.950
	DWV	100	13.711	5.009	2.362
<i>Case S2</i> 1 time	SLP	0	-	-	-
	DWA	0	-	-	-
	DWV	100	16.711	5.490	5.681
<i>Case S3</i> 100 times	SLP	14	18.223	6.091	2.874
	DWA	32	42.556	9.678	4.398
	DWV	85	20.919	7.562	7.392
<i>Case S4</i> 100 times	SLP	6	12.959	5.071	2.096
	DWA	4	23.425	5.765	2.422
	DWV	70	22.135	8.003	7.877

表 4.3 にシミュレーション結果を示す。表 4.3 にはゴールに到達した時の成功率、平均到達時間、平均移動距離 (TL)、平均姿勢変位 (PD) を示した。図 4.14-4.15 に *Case S1-Case S2* でのロボットと障害物の軌跡を示す。図 4.16-4.17 に *Case S3-Case S4* において 100 回シミュレーションを行ったロボットの軌跡を示す。



(a) SLP

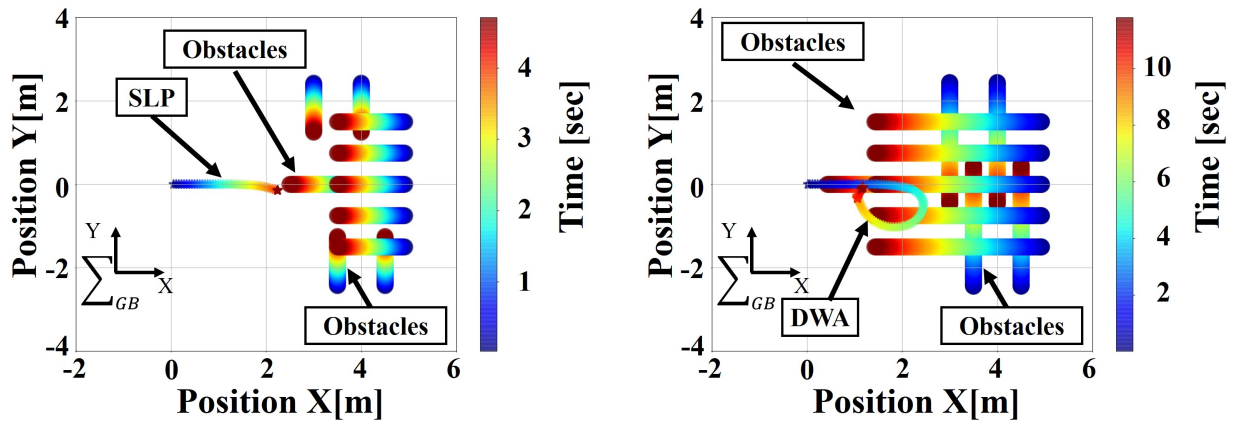
(b) DWA



(c) DWV

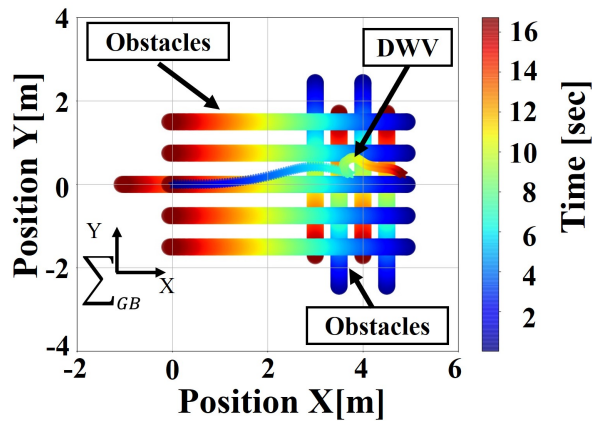
図 4.14 シミュレーション結果 (*Case S1*)

図 4.14 に *Case S1* のシミュレーション結果を示す。図 4.14(a)-(c) には SLP、DWA、DWV 使用時のロボットと障害物の軌跡を時間毎に描画した。*Case S1* の静的環境下では、図 4.14 が示すように、全ての手法がゴールへ到達した。到達時間はそれぞれ、SLP が約 8.7 秒、DWA が約 19.3 秒、DWV が約 13.7 秒であった。表 4.3 と図 4.14(a) より、*Case S1* で SLP は DWA や DWV よりもゴール地点に早く到達した。



(a) SLP

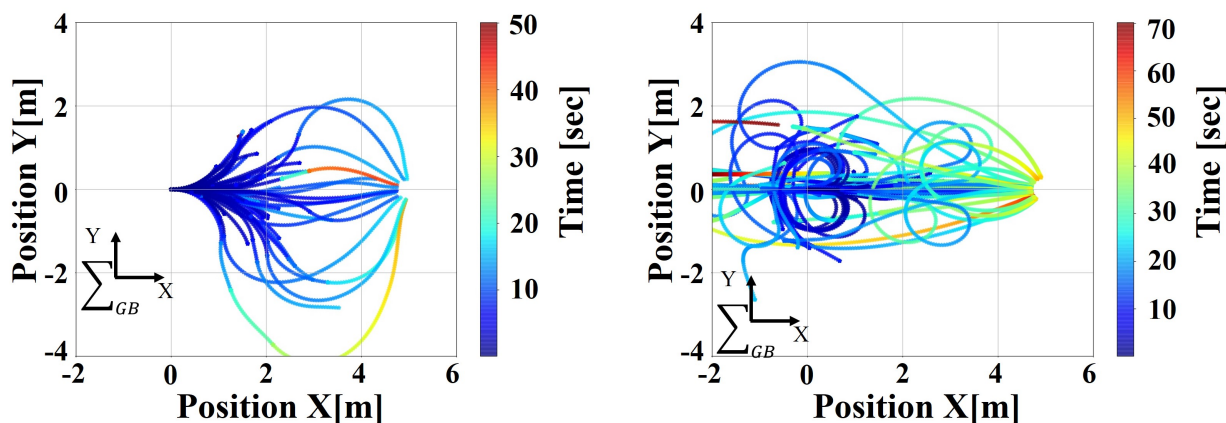
(b) DWA



(c) DWV

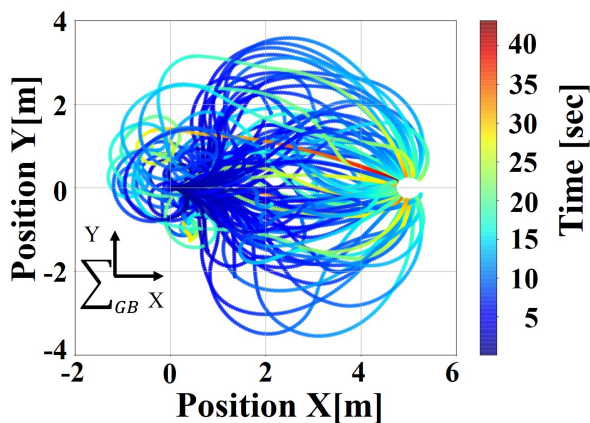
図 4.15 シミュレーション結果 (*Case S2*)

*Case S2*の動的環境下では図 4.15 が示すように、DWV のみがゴールへ到達した。図 4.15 から、SLP と DWA はゴール位置に到達していないことが分かった。SLP と DWA は、経路候補から最適な経路を選択する際に障害物を考慮しているが、経路候補を生成する時点で障害物を考慮しないため、SLP と DWA は動的障害物に衝突することとなった。一方、DWV は動的障害物を考慮した経路候補を生成したため、DWV はゴール地点に到達することが可能であった。表 4.3 と図 4.15(c) のように、DWV は *Case S2* で最も良い結果を得た。



(a) SLP

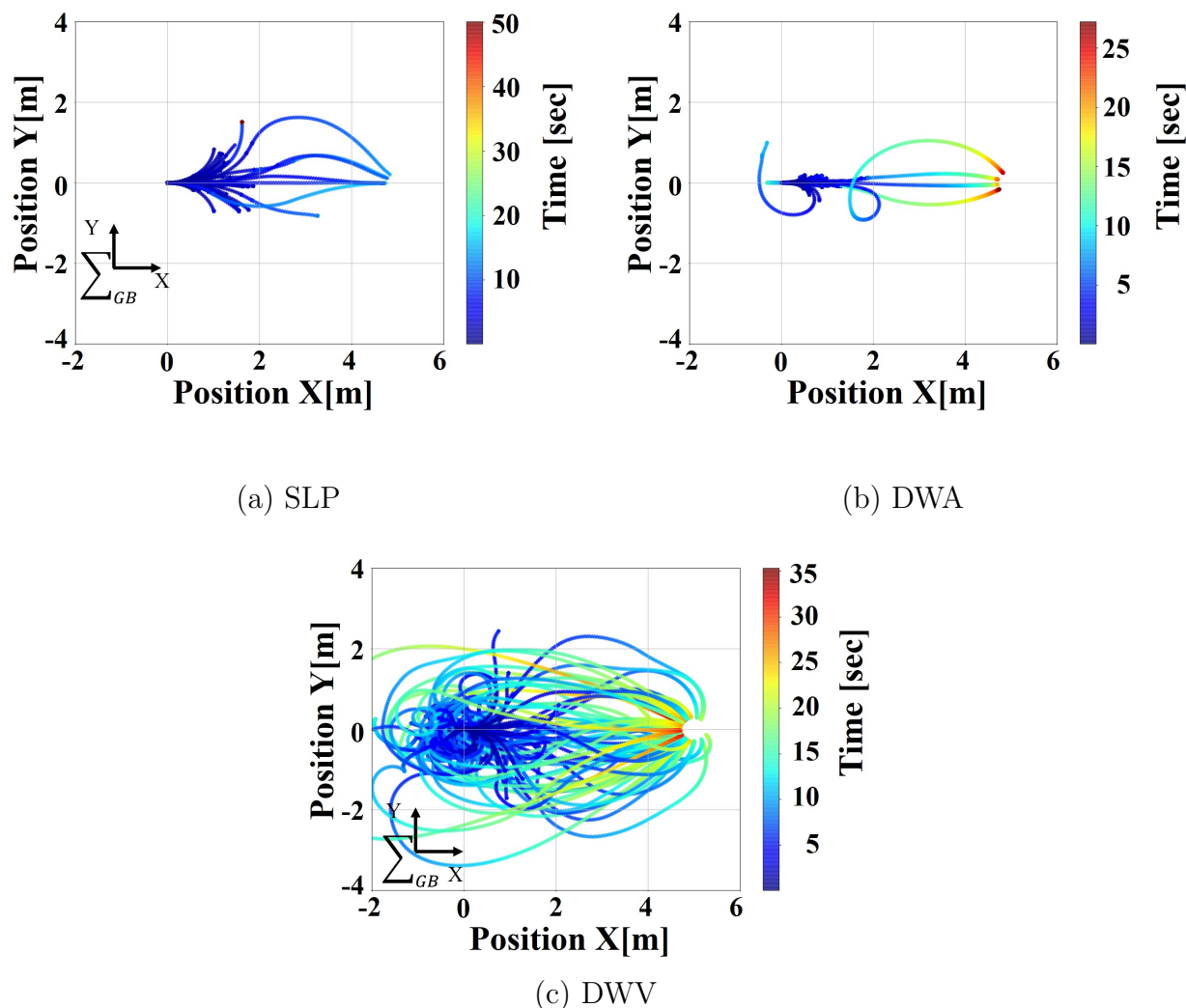
(b) DWA



(c) DWV

図 4.16 シミュレーション結果 (*Case S3*)

*Case S3*では、図 4.16 と表 4.3 が示すようにゴール到達率は SLP が 14%、DWA 32%、DWV が 85%であった。図 4.16(a) は SLP の軌跡結果を示し、ゴールへの軌道は直線に近い傾向であった。障害物の位置と速度がランダムで与えられているために、100 回のシミュレーションの中で、障害物がロボットの方向へ向かわない簡単なシチュエーションが生じたためと考えられる。図 4.16(b) は DWA の軌跡結果を示している。DWA が障害物と衝突せずにゴールへ到達した時、DWA は後進の行動を生成したため、他の手法と比べてゴールへの到達平均時間が増加した。表 4.3 と図 4.16(c) のように、DWV がもっともゴール到達率 85% と高い結果になった。よって *Case S3* においてもっとも良い結果を示した手法は DWV であった。

図 4.17 シミュレーション結果 (*Case S4*)

*Case S4*では、図 4.17と表 4.3が示すようにゴール到達率はSLPが6%、DWA4%、DWVが70%であった。図 4.17(a)(b)はSLPとDWAの軌跡結果を示しており、ゴールへの軌道は直線に近い傾向があった。障害物の位置と速度がランダムで与えられているために、100回のシミュレーションの中で、障害物がロボットへ接近しない簡単なシチュエーションが生じたためと考えられる。表 4.3と図 4.17(c)のように、DWVがもっともゴール到達率70%と高い結果になった。よって *Case S4*においてもっとも良い結果を示した手法はDWVであった。以上より、シミュレーション結果から提案手法の有効性が確認された。

シミュレーション結果よりDWVの課題や今後の方針等について述べる。*Case S1*から*Case S4*のシミュレーションにより、DWVが他の手法と比べて優れた結果を得たが、表 4.3

が示すように *Case S3* や *Case S4* ではロボットと障害物が衝突する場合が生じた。*Case S3* や *Case S4* では障害物の位置や速度をランダムに生成するために、障害物の初期位置によって障害物回避が困難な設定であった場合も生じた。DWV では経路候補の生成に予測時間を設定し、その予測時間内で衝突しない経路と速度指令値を生成することで、障害物と衝突しない自律移動を達成する手法である。つまり速度指令値をロボットに与える周期内に、予測時間を考慮した経路候補の生成と選択を行わなければならない。本シミュレーションでは予測時間を 4 秒とし、速度指令値の周期は 0.1 秒とした。予測時間が長い程、長期的な環境下を考慮した経路計画が可能になるが、障害物の予測軌跡が正しくなければ、障害物と衝突しない経路を生成しても、障害物と衝突してしまう可能性がある。また予測時間が短い程、計算コストは減少するが、短期的な動的障害物の動作しか考慮できず、ロボットが考慮できる障害物数が減少することとなり、障害物と衝突する可能性が上昇する。またロボットの動作速度より速い障害物や、全方向に動作が可能な障害物が存在する場合には、本シミュレーションで採用している 2 輪駆動型ロボットは真横に移動不可能なために、環境下に衝突しない空間があったとしても障害物回避可能な動作を生成できるとは限らない。以上のことから、ロボットと障害物が衝突する場合が生じた。

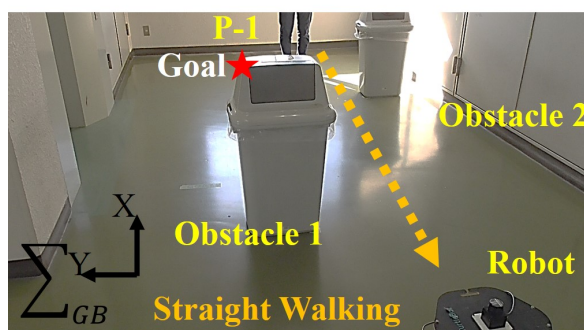
今後は評価関数の再設計、パラメータの設計方法の構築、新たな動作生成の提案等を行うことで、障害物と衝突する頻度を軽減し、より安全な自律移動の実現を目指す。

4.6.2 実験条件

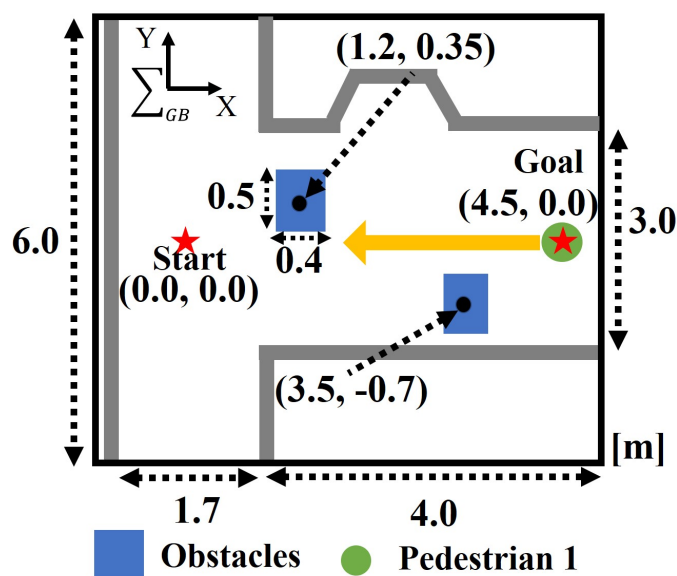
実験では2つの環境を使用した。図 4.20-4.21 が示すように、スタートとゴール位置はそれぞれ $(^{GB}x^{start}, ^{GB}y^{start}) = (0.0, 0.0)$ と $(^{GB}x^{goal}, ^{GB}y^{goal}) = (4.5, 0.0)$ とした。

Case E1: Straight Walking

図 4.20 が示すように、2つの静的障害物と歩行者 (P-1) を1名配置した。P-1 は実験環境で直線移動し、ロボットが動作すると同時に移動を開始することとした。



(a) 環境

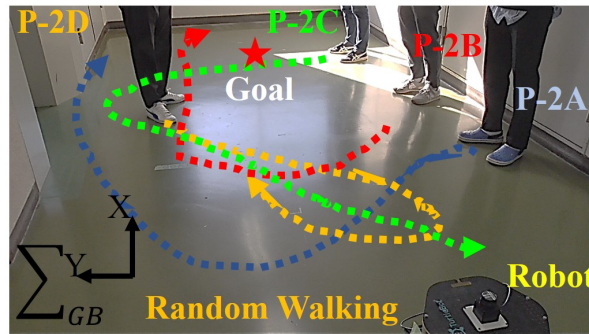


(b) 概略図

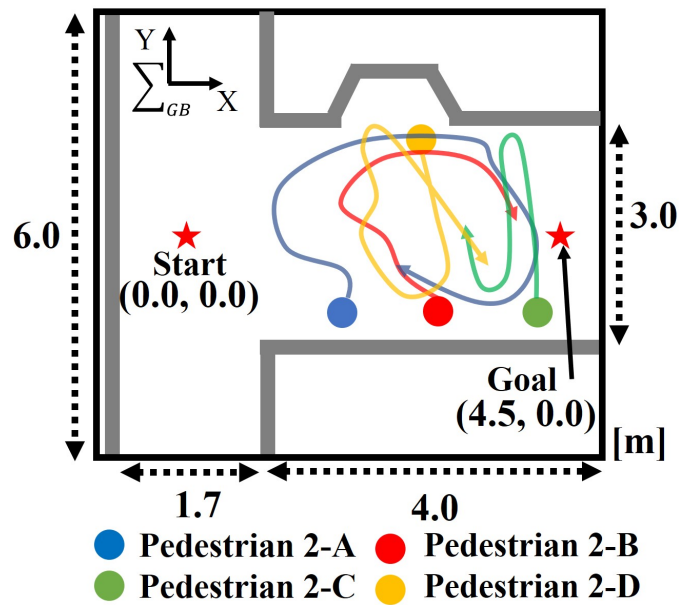
図 4.20 実験環境 (Case E1)

Case E2: Random Walking

図 4.21 が示すように、歩行者 (P-2A ~ 2D) を 4 名配置した。4 名の歩行者は環境内をランダムに歩行し、ロボットが動作すると同時に移動を開始することとした。



(a) 環境



(b) 概略図

図 4.21 実験環境 (Case E2)

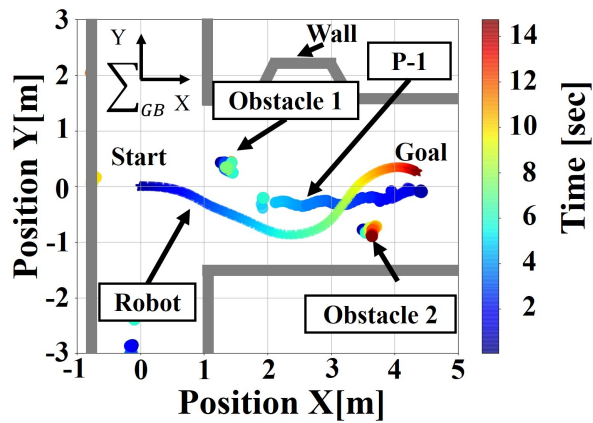
4.6.3 実験結果

図 4.22-4.23 は *Case E1* と *Case E2* の実験結果を示す。図 4.22(a) と図 4.23(a) はロボットと歩行者の軌跡を示す。図 4.22(b)-(e) と図 4.23(b)-(e) は任意の時間での経路候補生成や最適経路選択の様子を示す。黒円はロボットを示し、青円は障害物を示す。黒線と緑線は VM、黄線と赤線は経路候補と最適な経路、ピンク線は歩行者の移動方向、オレンジの点は測域センサから取得した点群である。

Case E1 では 2 つの静的障害物と 1 人の歩行者を設置した。図 4.22(a) が示すように、ロボットは衝突せずにゴールまで到達した。ロボットの動作様子を図 4.22(b)-(e) を用いて説明する。図 4.22(b) のように、スタートから 2 秒後時点でロボットは LRF を用いて Obstacle 1、Obstacle 2、P-1 を認識し、DWV により Obstacle 1 等との衝突回避を考慮した黄色の経路候補を生成した。図 4.22(c) のように、スタートから 4 秒後時点でロボットは Obstacle 1 を回避した。生成された経路候補に注目すると P-1 の進行方向には経路候補を生成せず、P-1 の背後に経路候補を生成した。また経路候補から選択された赤線の最適な経路は、Obstacle 2 と P-1 を回避する経路を選択していることから、静的障害物である Obstacle 2 と動的障害物である P-1 との衝突回避を考慮した行動を選択した。図 4.22(d) のように、スタートから 5 秒後時点でロボットは Obstacle 2 と P-1 を考慮した経路候補を生成した。生成された経路候補に注目すると P-1 の進行方向には経路候補を生成せず、P-1 の背後や Obstacle 2 を回避する経路候補を生成した。図 4.22(e) のように、スタートから 7 秒後時点では、ロボットは P-1 を回避した後に Obstacle 2 を回避する経路候補を生成した。以上より、*Case E1* では静的・動的障害物と衝突することなく、ゴール地点へ到達した。

Case E2 では 4 人の歩行者を設置した。図 4.23(a) が示すようにロボットは衝突せずにゴールまで到達した。図 4.23(b) のように、スタートから 3 秒後時点でロボットは LRF を用いて P-2A~ 2D を認識し、DWV により P-2A を考慮した経路候補を生成し、ロボットは後進した。図 4.23(c) のように、スタートから 6 秒後時点でロボットは LRF を用いて P-2A~ 2D を認識しており、ロボットは P-2A を回避し、P-2A~ 2D を考慮した経路候補を生成した。図 4.23(d) のように、スタートから 12 秒後時点でロボットは LRF を用いて P-2A~ 2D を認識しており、DWV により P-2A と P-2C を回避し、P-2A、P-2B、P-2D を考慮した経路候補を生成した。図 4.23(e) のように、スタートから 16 秒後時点でロボットは LRF を用いて P-2B を認識しており、ロボットは P-2B を考慮した経路候補を生成した。以上より、*Case E2* では障害物と衝突することなく、ゴール地点へ到達した。

以上、*Case E1* と *Case E2* の実験結果より提案手法の有効性が確認された。



(a) Trajectory Results

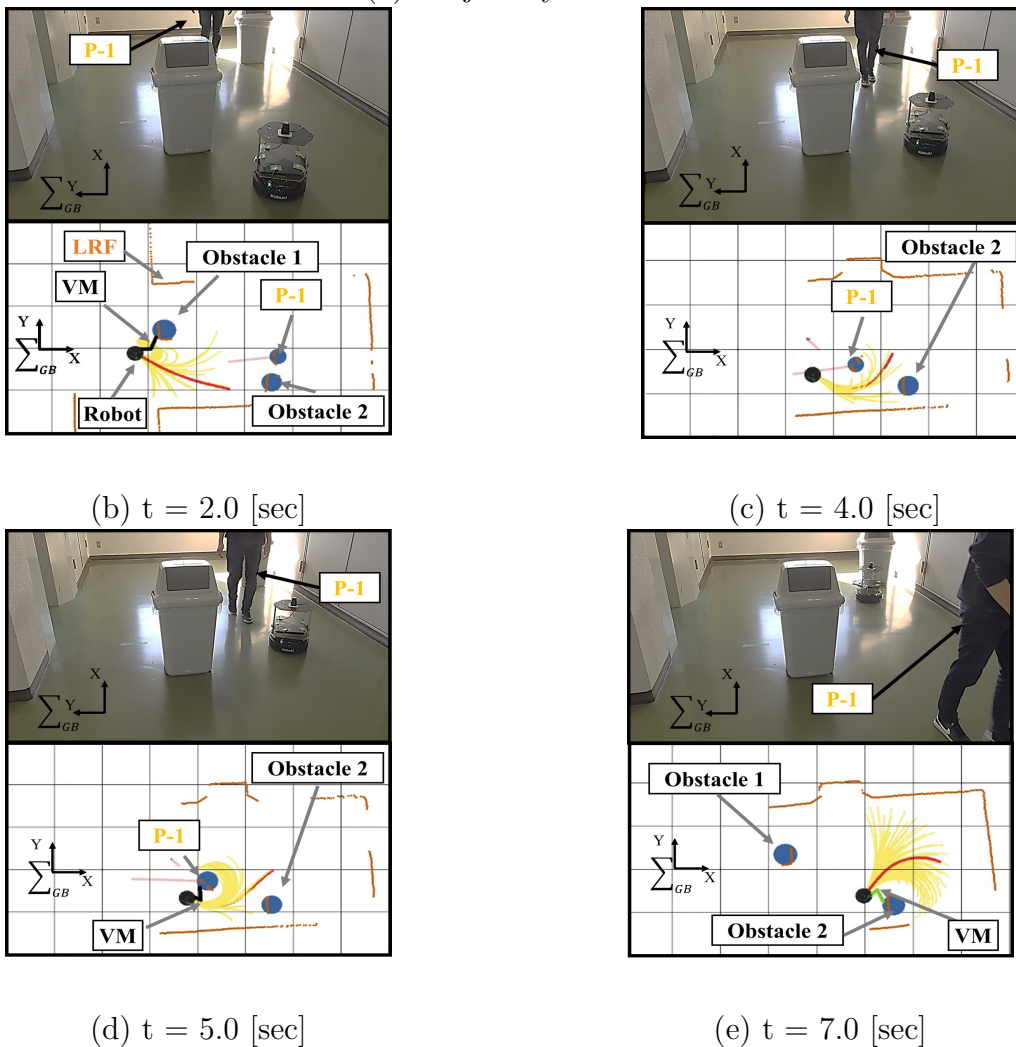
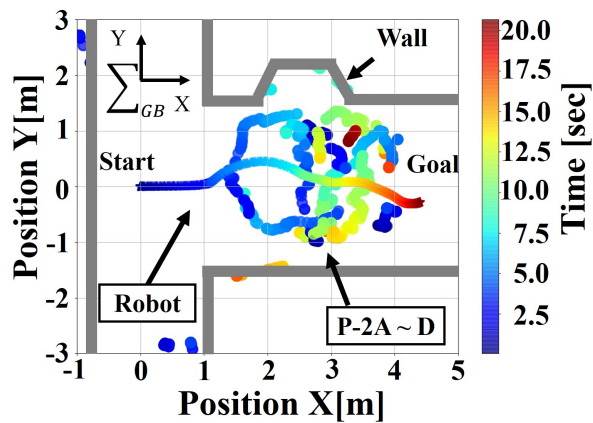
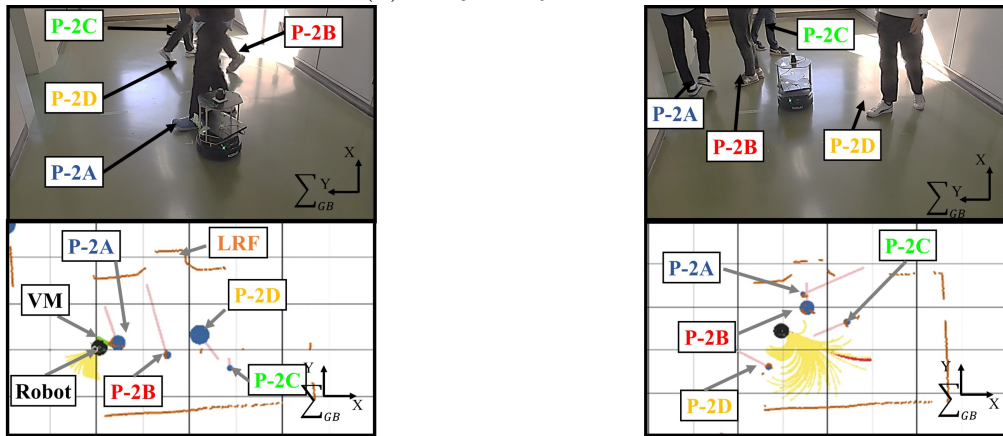


図 4.22 実験結果 (Case E1)

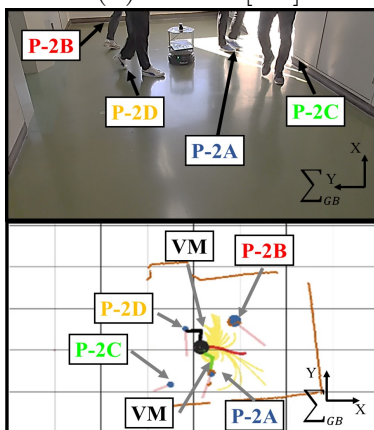


(a) Trajectory Results

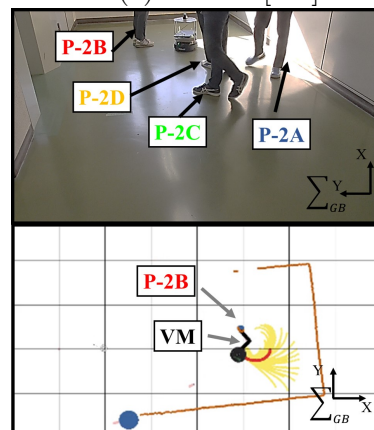


(b) $t = 3.0$ [sec]

(c) $t = 6.0$ [sec]



(d) $t = 12.0$ [sec]



(e) $t = 16.0$ [sec]

図 4.23 実験結果 (Case E2)

4.7 まとめ

少子高齢社会の影響によりサービスロボットによる労働力創出が希求されており、ロボットが人と共存する空間で移動する場面が多々求められる。ロボットが人と共存する空間で自律移動を実現するためには、時々刻々と変化する動的な環境下に対応する必要があるため、動的障害物を考慮した経路計画について述べた。

本章では、Virtual Manipulator (VM) と DWA に基づく局所経路計画手法である Dynamic Window Approach with Virtual Manipulators (DWV) を提案した。DWV は、VM と静的・動的障害物の予測位置によって修正された可変速度によって非直線・非円弧経路を含む経路候補を生成する。DWV では障害物の予測位置さえ推定できれば、障害物の予測経路が線形もしくは非線形でも対応可能である。そのため静的・動的な障害物が存在する環境においても、非直線経路や非円弧経路を含む障害物回避可能な経路を生成可能となった。様々なシミュレーションにより従来手法と比較と実環境での実機実験により、提案手法の有効性を示した。以上より今後、人が混在する動的環境下での自律移動実現の躍進となる。

第 5 章 結論

5.1 本稿の結論

本稿では人との共存を考慮した自律移動手法について論じた。人との共存を考慮した自律移動の課題として、「死角領域からの人の飛び出しが衝突の可能性を高めること」、「人等の動的障害物が混在するような時々刻々と変化する環境下ではロボットと障害物との衝突の可能性が高いこと」の 2 点を課題としてとりあげた。

第 2 章では、自律移動システムについて述べた。自律移動の実現には、自己位置推定、地図生成、認識、経路計画等が必要となる。自己位置推定では、測域センサや車輪のエンコーダ情報等から地図上でのロボットの位置を推定する。認識では、センサ等を用いてロボット周囲の環境情報を計測する。経路計画では、取得した環境情報とロボットの自己位置等の情報をもとに、障害物との衝突を回避する経路計画を行う。これらの処理を実時間で行うことで自律移動が実現されている。本稿では自律移動システムの実現に Robot Operating System (ROS) を採用した。

第 3 章では、死角領域を考慮した自律移動手法について論じた。提案したシステムは ROS Navigation Stack を基盤のシステムとして採用しており、死角領域に潜む人間を推定する認識部と死角領域を考慮した経路計画部をそれぞれ並列動作させることで実現した。死角領域を考慮した認識部では、LRF や RGB-D カメラから取得した環境情報から死角境界座標を算出し、死角領域において人が存在する可能性がある位置を推定する。推定位置を危険領域の中心座標とし、人が全方向に移動することを想定して、コストをコストマップに円形に伝播させることで危険領域を表現する。経路計画部においては、ロボットの運動制約を考慮した局所経路計画法である DWA (Dynamic Window Approach) の経路評価時に危険領域のコストを考慮することで、死角領域とロボットの運動制約を考慮した経路計画をリアルタイムで実行することが可能となった。シミュレーションにより、提案手法の有効性を確認した。

第 4 章では、人等の動的障害物が混在する環境下での自律移動手法について論じ、仮想マニピュレータと DWA に基づく局所経路探索手法である Dynamic Window Approach with

Virtual Manipulators (DWV) を提案した。特に経路候補生成に着眼点を置き、DWV は仮想マニピュレータと静的・動的障害物の予測位置によって修正された可変速度を用いて非直線・非円弧経路を含む経路候補生成した。そのため静的・動的な障害物が存在する環境においても非直線経路や非円弧経路を含む障害物回避可能な経路を生成可能とした。シミュレーション及び実験による提案手法の有効性を確認した。

以上より、本研究は人との共存を考慮したロボットの自律移動を実現し、労働力創出に貢献できると考える。

5.2 今後の展望

人との共存を考慮した自律移動手法の実現へ向け、「死角領域を考慮した自律移動手法」と「動的障害物を考慮した自律移動手法」の今後の展望について述べる。

死角領域を考慮した自律移動手法では、LRF や RGB-D カメラを用いて死角領域を考慮したコストマップを生成することで死角領域に対応した。今後は実際の環境下において評価実験を行いたい。また病院やスーパーマーケット等の屋内施設では、人等の動的障害物や商品や設備等の静的障害物等が多々存在するため、より複雑な環境に対してロバストな認識手法が求められるため、認識技術についても検討していきたい。具体的には機械学習によるアプローチ等、人が移動する際の認識や思考等を参考にしていきたい。

動的障害物を考慮した自律移動手法である DWV は、障害物の将来の予測軌道とロボットの相互作用を用いて、ロボットの障害物回避可能な経路候補を予め生成し、障害物回避可能な経路候補から最も評価関数を満たす経路を選択することにより、障害物と衝突せずに自律移動を実現可能とする手法である。本稿では DWV を局所的経路計画手法のみで評価を行ったため、大域的経路計画と組み合わせたシステムを構築していきたい。また本研究成果である DWV は障害物の予測軌道が線形・非線形に関わらず、障害物を考慮した経路候補を生成可能な手法であるため、LRF や RGB-D カメラ等のセンサを用いた認識システムを構築し、評価を行いたい。

上述の両者の手法は主にロボットと障害物との衝突回避に重点を置いた手法であるため、ロボットが人との共存を実現するにあたり、ロボットへの恐怖感、ぎこちなさ等を感じさせない移動手法についても検討していきたい。具体的には上述の問題を解決する評価関数の再設計、パラメータ設計法の構築、機械学習を用いた手法に取り組んでいきたい。

以上のように、本稿で得られた成果をさらに発展・展開させることで、ロボットの自律移動を実現し、労働力創出に貢献できると考える。

参考文献

- [1] “統計からみた我が国の高齢者 - 「敬老の日」にちなんで -”
<https://www.stat.go.jp/data/topics/topi1291.html>.
- [2] 櫻井 良: “ソフトロボット学への産業界からの期待 (特集 ソフトロボット学)”, 日本機械学会誌, Vol. 122, No. 1205, pp. 10-11, 2019.
- [3] 山崎 公俊, 渡辺 哲陽, 原田 研介, 横小路 泰義: “ロボット技術による食材加工の支援に向けて”, 日本ロボット学会誌, Vol. 39, No. 2, pp. 132-136, 2021.
- [4] P. Le and H. Kang: “Robot Manipulator Calibration Using a Model Based Identification Technique and a Neural Network With the Teaching Learning-Based Optimization”, *IEEE Access*, Vol. 8, pp. 105447-105454, 2020.
- [5] 後藤 聡, 岩永 卓也, 久良 修郭, 中村 政俊: “産業用多関節ロボットアームのスプライン補間教示信号を用いたジャーク抑制高速高精度制御”, 電気学会論文誌D, Vol. 123, No. 9, pp. 1071-1080, 2003.
- [6] I. Enebuse, M. Foo, B.S.K.K. Ibrahim, H. Ahmed, F. Supmak and O.S. Eyobu: “A Comparative Review of Hand-Eye Calibration Techniques for Vision Guided Robots”, *IEEE Access*, Vol. 9, pp. 113143-113155, 2021.
- [7] 原田 研介: “産業用ロボットによる組み立て作業の自動化に関する研究動向”, 精密工学会誌, Vol. 84, No. 4, pp. 299-302, 2018.
- [8] 武居 直行, 関口 叡範: “協働・自動化システムとキネマティクス”, 日本ロボット学会誌, Vol. 37, No. 10, pp. 919-922, 2019.
- [9] D. Strazdas, J. Hintz, A.-M. Felßberg and A. Al-Hamadi: “Robots and Wizards: An Investigation Into Natural Human – Robot Interaction”, *IEEE Access*, Vol. 8, pp. 207635-207642, 2020.

-
- [10] S. Robla-Gómez, V.M. Becerra, J.R. Llata, E. González-Sarabia, C. Torre-Ferrero and J. Pérez-Oria: “Working Together: A Review on Safe Human-Robot Collaboration in Industrial Environments”, *IEEE Access*, Vol. 5, pp. 26754-26773, 2017.
- [11] J. Liu, F. Zhou, L. Yin and Y. Wang: “A Novel Cloud Platform for Service Robots”, *IEEE Access*, Vol. 7, pp. 182951-182961, 2019.
- [12] Y. Dobrev, P. Gulden and M. Vossiek: “An Indoor Positioning System Based on Wireless Range and Angle Measurements Assisted by Multi-Modal Sensor Fusion for Service Robot Applications”, *IEEE Access*, Vol. 6, pp. 69036-69052, 2018.
- [13] S. Dutta, T.K. Maiti, M. Miura-Mattausch, Y. Ochi, N. Yorino and H.J. Mattausch: “Analysis of Sensor-Based Real-Time Balancing of Humanoid Robots on Inclined Surfaces”, *IEEE Access*, Vol. 8, pp. 212327-212338, 2020.
- [14] F. Dayoub, T. Morris and P. Corke: “Rubbing Shoulders With Mobile Service Robots”, *IEEE Access*, Vol. 3, pp. 333-342, 2015.
- [15] D. Lee, G. Kang, B. Kim and D.H. Shim: “Assistive Delivery Robot Application for Real-World Postal Services”, *IEEE Access*, Vol. 9, pp. 141981-141998, 2021.
- [16] K. Hirota and T. Murakami: “IMU Sensor based Human Motion Detection and Its Application to Braking Control of Electric Wheeled Walker for Fall-prevention”, *IEEE Journal of Industry Applications*, Vol. 5, No. 4, pp. 347-354, 2016.
- [17] M. Faisal, M. Algabri, B.M. Abdelkader, H. Dhahri and M.M. Al Rahhal: “Human Expertise in Mobile Robot Navigation”, *IEEE Access*, Vol. 6, pp. 1694-1705, 2018.
- [18] L. Zhu, M. Menon, M. Santillo and G. Linkowski: “Occlusion Handling for Industrial Robots”, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020.
- [19] A. Rangesh and M.M. Trivedi: “No Blind Spots: Full-Surround Multi-Object Tracking for Autonomous Vehicles Using Cameras and LiDARs”, *IEEE Transactions on Intelligent Vehicles*, Vol. 4, No. 4, pp. 588-599, 2019.
-

-
- [20] X. Zhao, X. Chen, Y. He, H. Cao and T. Chen: “Varying Speed Rate Controller for Human – Robot Teleoperation Based on Muscle Electrical Signals”, *IEEE Access*, Vol. 7, pp. 143563-143572, 2019.
- [21] M. Ito, K. Kawatsu, and M. Shibata: “Kinematic Control of Redundant Manipulators for Admitting Joint Range of Motion Maximally”, *IEEJ Journal of Industry Applications*, Vol. 6, No. 4, pp. 278-285, 2017.
- [22] P. De La Puente, D. Fischinger, M. Bajones, D. Wolf and M. Vincze: “Grasping Objects From the Floor in Assistive Robotics: Real World Implications and Lessons Learned”, *IEEE Access*, Vol. 7, pp. 123725-123735, 2019.
- [23] N. Motoi and R. Kubo: “Human-Machine Cooperative Grasping/ Manipulating System Using Force-based Compliance Controller with Force Threshold,” *IEEJ Journal of Industry Applications*, Vol. 5, No. 2, pp. 39-46, 2016.
- [24] “Amazon Scout”, <https://blog.aboutamazon.com/transportation/meet-scout>.
- [25] “Moxi”, <http://diligentrobots.com/moxi/>.
- [26] “Roomba”, <https://www.irobot-jp.com/roomba/>.
- [27] “Servi”, <https://www.softbankrobotics.com/jp/product/servi/>.
- [28] S.G. Tzafestas: “Mobile Robot Control and Navigation: A Global Overview”, *Journal of Intelligent and Robotic Systems*, Vol. 91, pp 35 – 58, 2018.
- [29] L. Chenping, L. Baoquan, W. Runhua and Z. Xuebo: “A survey on visual servoing for wheeled mobile robots”, *International Journal of Intelligent Robotics and Applications*, pp. 203 – 218, 2021.
- [30] T. Feng, Y. Yu, L. Wu, Y. Bai, Z. Xiao and Z. Lu: “A Human-Tracking Robot Using Ultra Wideband Technology”, *IEEE Access*, Vol. 6, pp. 42541-42550, 2018.
- [31] M. Tsunoda and C. Premachandra: “Remote Control of a Wheeled Robot by Visible Light for Support in Infectious Disease Hospitals”, *IEEE Access*, Vol. 9, pp. 124165-124175, 2021.
-

-
- [32] W. Li, N. Yang, J. Wang and D. Ren: “Kinematic Teleoperation of Wheeled Mobile Robot With Slippage Compensation on Soft Terrains”, *IEEE Access*, Vol. 7, pp. 110982-110991, 2019.
- [33] F.A.X. Da Mota, M.X. Rocha, J.J.P.C. Rodrigues, V.H.C. De Albuquerque and A.R. De Alexandria: “Localization and Navigation for Autonomous Mobile Robots Using Petri Nets in Indoor Environments”, *IEEE Access*, Vol. 6, pp. 31665-31676, 2018.
- [34] U. Orozco-Rosas, K. Picos and O. Montiel: “Hybrid Path Planning Algorithm Based on Membrane Pseudo-Bacterial Potential Field for Autonomous Mobile Robots”, *IEEE Access*, Vol. 7, pp. 156787-156803, 2019.
- [35] M.N.A. Wahab, C.M. Lee, M.F. Akbar and F.H. Hassan: “Path Planning for Mobile Robot Navigation in Unknown Indoor Environments Using Hybrid PSOFS Algorithm”, *IEEE Access*, Vol. 8, pp. 161805-161815, 2020.
- [36] K. Schlegel, P. Weissig and P. Protzel: “A blind-spot-aware optimization-based planner for safe robot navigation”, *Proceedings of European Conference on Mobile Robots*, pp. 1-8, 2021.
- [37] 横田 篤紀, 本田 空, 山福 佳, 西田 健, 池永 全志, 森 直樹, 松永 彰, 丸山 翔, 吉田 享広, 長田 真太郎: “産業用ロボット制御への第5世代移動通信システムの適用”, *電気学会論文誌D*, Vol. 140, No. 4, pp. 314-326, 2020.
- [38] D. Portugal, P. Alvito, E. Christodoulou, G. Samaras , and J. Dias: “A Study on the Deployment of a Service Robot in an Elderly Care Center”, *International Journal of Social Robotics*, Vol. 11, No. 2, pp. 317-341, 2019.
- [39] X. Zhang, X. Mu, H. Xu, A.B. Alhassan and H.K. Kadry: “Vibration Characteristics Analysis of Human- Robot Coupled System for Walking Posture of Elderly-Assistant Robot”, *IEEE Access*, Vol. 9, pp. 44217-44235, 2021.
- [40] Q. Yan, J. Huang, C. Xiong, Z. Yang and Z. Yang: “Data-Driven Human-Robot Coordination Based Walking State Monitoring With Cane-Type Robot”, *IEEE Access*, Vol. 6, pp. 8896-8908, 2018.
-

-
- [41] R. Pinillos, S. Marcos, R. Feliz, E. Zalama, and J. Gómez-García-Bermejo: “Long-term assessment of a service robot in a hotel environment”, *Robotics and Autonomous Systems*, Vol. 79, No. C, pp. 40-57, 2016.
- [42] “2035 年に向けたロボット産業の将来市場予測”, <https://www.nedo.go.jp/content/100080673.pdf>
- [43] B. Rouzier, M. Hazaz, T. Murakami, and W. Xu: “Application of Active Driving Assist to Remotely Controlled Car in Collision Avoidance”, *IEEJ Journal Industry Applications*, Vol. 7, No. 4, pp. 289-297, 2018.
- [44] Y. Hasegawa, and Y. Fujimoto: “Experimental Verification of Path Planning with SLAM”, *IEEJ Journal Industry Applications*, Vol. 5, No. 3, pp. 253-260, 2016.
- [45] D. Fox, W. Burgard, and S. Thrun: “The Dynamic Window Approach to Collision Avoidance”, *Proceedings of IEEE International Conference on Robotics & Automation Magazine*, Vol. 4, pp. 23-33, 1997.
- [46] 内海 佑斗, 北川 晋吾, 矢野倉 伊織, 岡田 慧, 稲葉 雅幸: “CNNによる深度予測に基づく鏡を活用した死角の物体知覚”, 人工知能学会全国大会論文集, 1L2-J-11-02, pp. 1-4, 2019.
- [47] 田村 雄介, 濱崎 峻資, 山下 淳, 浅間 一: “環境に応じた人間の移動予測に基づく移動ロボットの人物回避”, 日本機械学会論文集 C 編, Vol. 79, No. 799, pp. 617-628, 2013.
- [48] W. Chung, S. Kim, M. Choi, J. Choi, H. Kim, C. Moon, and J. Song: “Safe Navigation of a Mobile Robot Considering Visibility of Environment”, *IEEE Transactions on Industrial Electronics*, Vol. 56, No. 10, pp. 3941-3950, 2009.
- [49] 関 道人, 羽田 靖史: “センサ計測可能範囲を考慮したダイクストラ法による移動ロボットのオフライン経路計画”, ロボティクス・メカトロニクス講演会講演概要集, 2A2-F16, 2018.
- [50] 萬 礼応, 森口智規, 高橋正樹: “自律移動ロボットの環境地図とセンサ情報に基づく隠れ領域を考慮したリアルタイム速度制御”, 人工知能学会 AIチャレンジ研究会論文集, pp. 48-52, 2012.
-

-
- [51] 黒坂 翼, 金子 正秀: “センサの死角を考慮して安全な速度制御と最適移動経路の選択を行う自律移動ロボット”, 電気学会論文誌C, Vol. 4, No. 4, pp. 356-364, 2015.
- [52] 刈谷啓伸, 小林一行: “障害物のオクルージョン領域を考慮した自律移動ロボットの経路計画法”, 法政大学大学院理工学・工学研究科紀要, Vol. 56, pp. 1-6, 2015.
- [53] 小林 聖人, 元井 直樹: “車輪型移動ロボットにおける死角領域を考慮したコストマップに基づく局所的経路計画手法,” 電気学会論文誌D, Vol. 141, No. 8, pp. 598-605, 2021.
- [54] M. Kobayashi, and N. Motoi: “Path Planning Method Considering Blind Spots Based on ROS Navigation Stack and Dynamic Window Approach for Wheeled Mobile Robot,” *Proceedings of International Power Electronics Conference*, pp. 274-279, 2022.
- [55] “ROS Wiki DWA Local Planner”, http://wiki.ros.org/dwa_local_planner.
- [56] “ROS Wiki costmap 2d”, http://wiki.ros.org/costmap_2d.
- [57] G. Grisetti, C. Stachniss and W. Burgard: “Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling”, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2432-2437, 2005.
- [58] C. Nam, S. Lee, J. Lee, S.H. Cheong, D.H. Kim, C. Kim, I. Kim and S. Park: “A Software Architecture for Service Robots Manipulating Objects in Human Environments”, *IEEE Access*, Vol. 8, pp. 117900-117920, 2020.
- [59] M.A.V.J. Muthugala and A.G.B.P. Jayasekara: “A Review of Service Robots Coping With Uncertain Information in Natural Language Instructions”, *IEEE Access*, Vol. 6, pp. 12913-12928, 2018.
- [60] L. Lestingi, M. Askarpour, M.M. Bersani and M. Rossi: “A Deployment Framework for Formally Verified Human-Robot Interactions”, *IEEE Access*, Vol. 9, pp. 136616-136635, 2021.
- [61] Y. Fuse and M. Tokumaru: “Social Influence of Group Norms Developed by Human-Robot Groups”, *IEEE Access*, Vol. 8, pp. 56081-56091, 2020.
-

-
- [62] C. Sirithunge, A.G.B.P. Jayasekara and D.P. Chandima: “Proactive Robots With the Perception of Nonverbal Human Behavior: A Review”, *IEEE Access*, Vol. 7, pp. 77308-77327, 2019.
- [63] R. Parween, M. Vega Heredia, M.M. Rayguru, R. Enjikalayil Abdulkader and M.R. Elara: “Autonomous Self-Reconfigurable Floor Cleaning Robot”, *IEEE Access*, Vol. 8, pp. 114433-114442, 2020.
- [64] M.A.K. Niloy, A. Shama, R.K. Chakraborty, M.J. Ryan, F.R. Badal, Z. Tasneem, M.H. Ahamed, S.I. Moyeen, S.K. Das, M.F. Ali, M.R. Islam and D.K. Saha, “Critical Design and Control Issues of Indoor Autonomous Mobile Robots: A Review”, *IEEE Access*, Vol. 9, pp. 35338-35370, 2021.
- [65] A. Motroni, A. Buffi and P. Nepa: “A Survey on Indoor Vehicle Localization Through RFID Technology”, *IEEE Access*, Vol. 9, pp. 17921-17942, 2021.
- [66] Y. Zheng, S. Chen and H. Cheng: “Real-Time Cloud Visual Simultaneous Localization and Mapping for Indoor Service Robots”, *IEEE Access*, Vol. 8, pp. 16816-16829, 2020.
- [67] M.B. Alatise and G.P. Hancke: “A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods”, *IEEE Access*, Vol. 8, pp. 39830-39846, 2020.
- [68] C. Chen, J. Jiang, N. Lv and S. Li: “An Intelligent Path Planning Scheme of Autonomous Vehicles Platoon Using Deep Reinforcement Learning on Network Edge”, *IEEE Access*, Vol. 8, pp. 99059-99069, 2020.
- [69] F. Li, X. Fan and Z. Hou: “A Firefly Algorithm With Self-Adaptive Population Size for Global Path Planning of Mobile Robot”, *IEEE Access*, Vol. 8, pp. 168951-168964, 2020.
- [70] M. Luo, X. Hou and J. Yang: “Surface Optimal Path Planning Using an Extended Dijkstra Algorithm,” *IEEE Access*, Vol. 8, pp. 147827-147838, 2020.
- [71] Z. Liu, H. Liu, Z. Lu and Q. Zeng: “A Dynamic Fusion Pathfinding Algorithm Using Delaunay Triangulation and Improved A-Star for Mobile Robots,” *IEEE Access*, Vol. 9, pp. 20602-20621, 2021.
-

-
- [72] Q. Jin, C. Tang and W. Cai: “Research on Dynamic Path Planning Based on the Fusion Algorithm of Improved Ant Colony Optimization and Dynamic Window Method”, *IEEE Access*, Vol. 10, pp. 28322-28332, 2022.
- [73] U. Patel, N.K.S. Kumar, A.J. Sathyamoorthy and D. Manocha: “DWA-RL: Dynamically Feasible Deep Reinforcement Learning Policy for Robot Navigation among Mobile Obstacles”, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 6057-6063, 2021.
- [74] D. González, J. Pérez, V. Milanés and F. Nashashibi: “A Review of Motion Planning Techniques for Automated Vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, Vol. 17, No. 4, pp. 1135-1145, 2016.
- [75] K. Cai, C. Wang, J. Cheng, C.W. De Silva, and M.Q. Meng: “Mobile Robot Path Planning in Dynamic Environments: A Survey”, *Instrumentation*, Vol. 6, No. 2, pp. 92-102, 2019.
- [76] P. Fiorini and Z. Shiller: “Motion Planning in Dynamic Environments using Velocity Obstacles”, *The International Journal of Robotics Research*, Vol. 17, No. 7, pp. 760-772, 1998.
- [77] J. van den Berg, Ming Lin and D. Manocha: “Reciprocal Velocity Obstacles for Real-Time Multi-Agent Navigation”, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1928-1935, 2008.
- [78] T. Xu, S. Zhang, Z. Jiang, Z. Liu and H. Cheng: “Collision Avoidance of High-Speed Obstacles for Mobile Robots via Maximum-Speed Aware Velocity Obstacle Method”, *IEEE Access*, Vol. 8, pp. 138493-138507, 2020.
- [79] J. Snape, J.v.d. Berg, S.J. Guy and D. Manocha: “The Hybrid Reciprocal Velocity Obstacle”, *IEEE Transactions on Robotics*, Vol. 27, No. 4, pp. 696-706, 2011.
- [80] D. Fox, W. Burgard, and S. Thrun: “The Dynamic Window Approach to Collision Avoidance”, *IEEE Robotics & Automation Magazine*, Vol. 4, pp. 23-33, 1997.
-

-
- [81] M. Dobrevski and D. Skocaj: “Adaptive Dynamic Window Approach for Local Navigation”, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 6930-6936, 2020.
- [82] L. Liu, J. Yao, D. He, J. Chen, J. Huang, H. Xu, B. Wang and J. Guo: “Global Dynamic Path Planning Fusion Algorithm Combining Jump-A* Algorithm and Dynamic Window Approach”, *IEEE Access*, Vol. 9, pp. 19632-19638, 2021.
- [83] T.M. Howard and A. Kelly: “Optimal rough terrain trajectory generation for wheeled mobile robots”, *The International Journal of Robotics Research*, Vol. 26, No. 2, pp. 141-166, 2007.
- [84] T.M. Howard, C.J. Green, A. Kelly, and D. Ferguson: “State Space Sampling of Feasible Motions for High-Performance Mobile Robot Navigation in Complex Environments”, *Journal of Field Robotics*, Vol. 25, No. 6-7, pp. 325-345, 2008.
- [85] M. Kobayashi and N. Motoi: “Local Path Planning Method Based on Virtual Manipulators and Dynamic Window Approach for a Wheeled Mobile Robot”, *Proceedings of IEEE/SICE International Symposium on System Integration*, pp. 499-504, 2021.
- [86] M. Kobayashi, N. Motoi: “Local Path Planning: Dynamic Window Approach with Virtual Manipulators Considering Dynamic Obstacles”, *IEEE Access*, Vol. 10, pp. 17018-17029, 2022.
- [87] K. Yamazaki and M. Inaba: “Trajectory Control of Wheeled Mobile Robots Based on Virtual Manipulators”, *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, pp. 2974-2978, 2009.
- [88] 山崎 公俊, 稲葉 雅幸: “仮想マニピュレータを用いた移動ロボットの反射的動作生成法”, *日本ロボット学会誌*, 29 巻, 2 号, pp. 163-171, 2011.
- [89] H. Zhang, C. Zhang, W. Yang and C.-Y. Chen: “Localization and navigation using QR code for mobile robot in indoor environment”, *Proceedings of IEEE International Conference on Robotics and Biomimetics*, pp. 2501-2506, 2015.
-

- [90] J. Park, R. Delgado and B.W. Choi: “Real-Time Characteristics of ROS 2.0 in Multi-agent Robot Systems: An Empirical Study”, *IEEE Access*, Vol. 8, pp. 154637-154651, 2020.
- [91] H. Lee, H. Seo and H. Kim: “Trajectory Optimization and Replanning Framework for a Micro Air Vehicle in Cluttered Environments”, *IEEE Access*, Vol. 8, pp. 135406-135415, 2020.
- [92] D. Jin, Z. Fang and J. Zeng: “A Robust Autonomous Following Method for Mobile Robots in Dynamic Environments”, *IEEE Access*, Vol. 8, pp. 150311-150325, 2020.
- [93] “Turtlebot2”, <https://www.turtlebot.com/turtlebot2/>.
- [94] “URG-04LX-UG01”, <https://www.hokuyo-aut.jp/search/single.php?serial=166>.
- [95] “obstacle_detector”, https://github.com/tysik/obstacle_detector.

謝辞

私は2016年より神戸大学 元井研究室に所属しました。2019年3月に修士課程を修了後、セイコーエプソン株式会社に入社し、ロボットや自動化技術に携わる研究・技術開発に従事しました。2019年10月より博士課程後期課程に入学し、再び元井研究室での研究活動を社会人生活と並行しながら開始しました。本稿を執筆できるのも、数多くの人々のサポートがあったからです。誠に僭越ながら、皆様に感謝申し上げたいと思います。

本稿をまとめるにあたり、常日頃からご指導・ご鞭撻を賜り、有意義な研究生活や環境を提供して頂いた神戸大学 元井直樹准教授に心より感謝を申し上げます。国内外の学会参加等、とても充実した日々を過ごすことができました。

本稿の審査過程において、数々の助言とご指導を承りました、神戸大学 佐保博章教授、堀口知也教授、山本茂広准教授に深く感謝申し上げます。議論した時間はとても有意義な一時であり、今後の研究活動等においても活かしていこうと思います。

学部在籍時より様々なサポート等を頂きました、神戸大学 海事科学部(海洋政策科学部)、海事科学研究科の皆様にご感謝申し上げます。学内プログラムでオーストラリアのタスマニア大学への短期留学を行えたのはとても良い経験となりました。

国立研究開発法人科学技術振興機構(JST)の次世代研究者挑戦的研究プログラム(SPRING)事業である「異分野共創による次世代卓越博士人材育成プロジェクト」や「梅木信子奨学金」において、多大なるご支援を頂きました。研究活動等とても充実した日々を過ごせることができました。関係者の皆様に深く感謝申し上げます。

国内外の学会等では、芝浦工業大学 清水創太教授、東京電機大学 五十嵐洋教授、並びに様々な先生方や学生の皆様からご助言を頂きました。深く感謝申し上げます。

セイコーエプソン株式会社籍時には、関係会社様を含め、様々な方々にお世話になりました。社会人としての過ごし方、仕事の楽しさ、やりがい等とても学ぶことが多く、長野県や山形県等にて充実した日々を過ごすことができました。博士課程に対しての暖かい応援、業務等で関わって下さいました社員の皆様や、信州大学 山崎公俊教授に深く感謝申し上げます。

オムロンサイニックス株式会社 田中一敏さま、濱屋政志さま、並びに社員の皆様、オックスフォード大学 山田潤さまにはロボティクスと機械学習を用いた研究において数々のご指導やサポートを承り、深く感謝申し上げます。とても刺激的な日々を過ごすことができました。

本稿の研究を遂行するにあたり、元井研究室の諸先輩方、同期、後輩の皆様にはとてもお世話になりました。研究室生活がとても有意義な時間となり、感謝申し上げます。特に眞崎峻くんとは日々の活発な研究議論やオーストラリア タスマニア大学への留学等、大変お世話になりました。研究活動等においては、移動ロボット関係では戸田瑤子さん、眞崎峻くん、高谷拓伸くん、花岡勇吾くん、稲原大翔くん、池野聡司くん、上田洋輔くん、中村友亮くん、モーションコントロール関係では中村昇暉くん、椿崇裕くん、山口瞭くん、岡田大河くん、元木碧波くんには改めて深く感謝申し上げます。

研究活動等を応援して下さった友人にも改めて感謝申し上げます。

日々の生活において、生協の皆様との会話は心が和むものでした。生協の皆様、ありがとうございました。

これまで家族の大きな支えや協力があり、研究生生活に専念できました。私が博士課程へ進学することへの理解や応援、また様々なアドバイスも頂きました。家族と過ごす日々はとても心が和むものでした。父、母、姉、祖父母、親戚一同に深く感謝申し上げます。

最後になりましたが、皆様のおかげでとても充実した研究生生活を送ることができ、博士論文執筆までに至りました。今一度、深く感謝申し上げます。

業績

学術雑誌 (査読有り)

1. M. Kobayashi, and N. Motoi: “Local Path Planning: Dynamic Window Approach with Virtual Manipulators Considering Dynamic Obstacles”, *IEEE Access*, Vol. 10, pp. 17018-17029, 2022.
2. R. Masaki, M. Kobayashi, and N. Motoi: “Remote Control Method with Force and Visual Assists Based on Time to Collision for Mobile Robot”, *Applied Sciences*, Vol. 12, No. 8: 3727, 2022.
3. 小林 聖人, 元井 直樹: “車輪型移動ロボットにおける死角領域を考慮したコストマップに基づく局所的経路計画手法”, *電気学会産業応用部門誌*, Vol. 141, No. 8, pp. 598-605, 2021.
4. M. Kobayashi, and N. Motoi: “Hybrid Control of Trajectory Tracking and Collision Avoidance Based on Optimization Problem for Wheeled Mobile Robot”, *IEEJ Journal of Industry Applications*, Vol. 9, No. 4, pp. 331-340, 2020.
5. N. Motoi, M. Kobayashi, and R. Masaki: “Remote Control Method for Mobile Robot by using Force Feedback Based on Collision Prediction Map”, *IEEJ Journal of Industry Applications*, Vol. 8, No. 4, pp. 727-735, 2019.

国際会議論文 (査読有り)

1. M. Kobayashi, and N. Motoi: “Path Planning Method Considering Blind Spots Based on ROS Navigation Stack and Dynamic Window Approach for Wheeled Mobile Robot”, *Proceedings of the International Power Electronics Conference*, pp. 274-279, 2022.
2. M. Kobayashi, A. Yamaguchi, and N. Motoi: “Motion-Reproduction Method Based on Iterative Learning Control for Variable Reproduction Speed”, *Proceedings of the IEEJ International Workshop on Sensing, Actuation, and Motion Control*, pp. 419-424, 2022.

-
3. H. Takatani, M. Kobayashi, and N. Motoi: “Human Tracking Control by Using Model Predictive Control with Human Trajectory Model for Mobile Robot”, *Proceedings of the IEEJ International Workshop on Sensing, Actuation, and Motion Control*, SS3-3, pp. 128-133, 2021.
 4. M. Kobayashi, and N. Motoi: “Local Path Planning Method Based on Virtual Manipulators and Dynamic Window Approach for a Wheeled Mobile Robot”, *Proceedings of the IEEE/SICE International Symposium on System Integrations*, pp. 499-504, 2021.
 5. N. Motoi, R. Masaki, and M. Kobayashi: “Remote Control Method with Force Assist Based on Collision Prediction Calculated from Each Turning Radius in Mobile Robot”, *Proceedings of IEEE 2019 International Conference on Mechatronics*, pp. 477-482, 2019.
 6. M. Kobayashi, and N. Motoi: “Velocity Command Generation Considering Trajectory Tracking and Collision Avoidance for Mobile Robot”, *Proceedings of the IEEJ International Workshop on Sensing, Actuation, Motion Control and Optimization*, II10, pp. 1-6, 2019.
 7. M. Kobayashi, and N. Motoi: “Tracking Control Method Considering Obstacle Avoidance by Reflective Motion for Mobile Robot”, *Proceedings of the Annual Conference of the IEEE Industrial Electronics Society*, pp. 5493-5498, 2018.
 8. N. Motoi, M. Kobayashi, and R. Masaki: “Remote Control Method for Mobile Robot by using Force Feedback Based on Collision Prediction Map”, *Proceedings of the IEEJ International Workshop on Sensing, Actuation, and Motion Control*, IS3-1, pp. 1-6, 2018.
 9. N. Motoi, H. Kimura, and M. Kobayashi: “Experimental Operability Evaluation of Remote Control with Force Feedback for Mobile Robot”, *Proceedings of IEEE International Conference on Industrial Technology*, pp. 159-164, 2018.

国内会議論文 (査読なし)

1. 山口 瞭, 小林 聖人, 元井 直樹: “再現速度変更型モーションコピーシステムの反復学習制御による精度向上”, 電気学会産業計測制御研究会論文集, IIC-21-037, pp. 35-40, 2021.
2. 小林 聖人, 元井 直樹: “車輪型移動ロボットにおける衝突回避と軌道追従を考慮した速度生成手法の検討”, 電気学会産業計測制御研究会論文集, IIC-18-023, pp. 27-32, 2018.
3. 眞崎 峻, 小林 聖人, 元井 直樹: “状況に応じて選択された仮想的な力指令値に基づく力覚操作アシスト機能を有する移動ロボットのための遠隔操作制御手法の研究”, 電気学会産業応用部門大会, No. Y-96, 2018.
4. 小林 聖人, 元井 直樹: “移動ロボットにおけるモデル予測制御に基づく衝突回避動作に関する研究”, 電気学会産業応用部門大会, No. Y-78, 2017.

受賞

1. M. Kobayashi, “Excellent Presentation Award”, *IEEJ Industry Applications Society*, 2022.
2. N. Motoi, R. Masaki, M. Kobayashi, “IES Best Conference Paper Award”, *IEEE Industrial Electronics Society*, 2020.
3. M. Kobayashi, “Excellent Presentation Award”, *IEEJ Industry Applications Society*, 2019.
4. N. Motoi, R. Masaki, M. Kobayashi, “Best Regular Paper Award”, *IEEE International Conference on Mechatronics*, 2019.
5. 小林 聖人, “優秀論文発表賞”, 電気学会産業応用部門, 2019.
6. 小林 聖人, “優秀論文発表賞”, 電気学会産業計測制御技術委員会, 2018.