2022（令和４）年度　　修士学位論文

動的環境における生活支援ロボットの障害物回避動作計画

Motion Planning for Obstacle Avoidance of Life Support Robots in Dynamic Environments

2023 年 3 月 3 日

高知工科大学大学院　工学研究科基盤工学専攻
知能機械工学コース

1255011　後藤　遼太

指導教員：王　碩玉，楊　光

目次

第 1 章　　緒言

1-1　研究背景

我が国では少子高齢化が進み，65 歳以上の人口は 3,621 万人．総人口に占める高齢者化率は 28.9[%]となっており（令和 3 年 10 月時点），今後も高齢化は上昇をし続け令和 47 年度には，約 2.6 人に 1 人が 65 歳以上になると推定されている．[1]それゆえ，介護サービスを受ける要介護者の増加に伴い，介護人材の確保が近年問題となっている．図 1.1 に示すグラフは近年の介護サービス利用者数の推移である．[2]このグラフからも読み取れるように近年の利用者数は 500 万人を超え，その中でも「居宅サービス」を利用している人数が半分以上を占めており，令和 2 年度では約 393 万人が利用している．また厚生労働相により推計された「将来，必要になる介護人材数」は，令和 7 年度に約 233 万人，令和 22 年度には約 280 万人となる．[3]これは，令和元年に比べ令和 7 年度には 22 万人，令和 22 年度には約 69 万人もの介護人材を増員しなければならないことを示す．

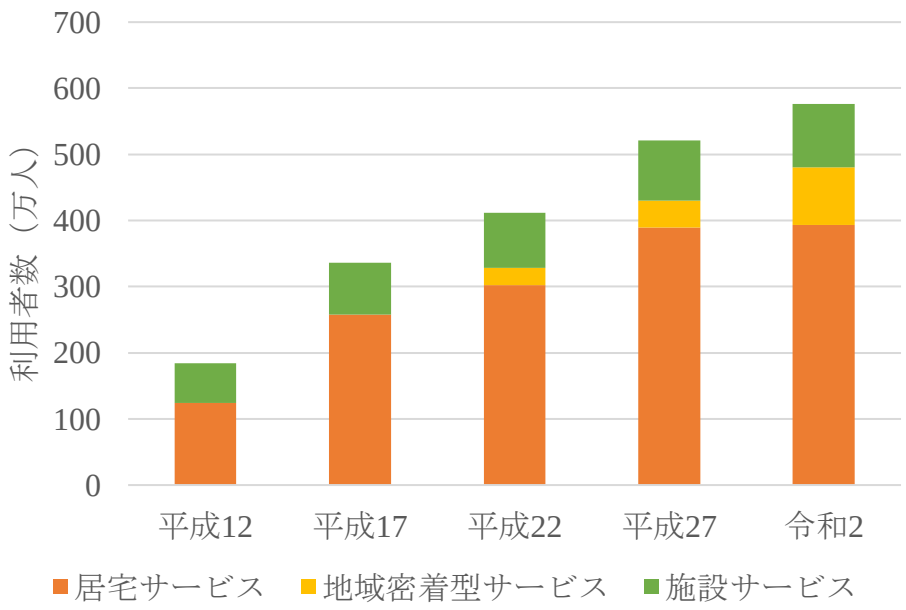


図 1.1　介護サービス利用者の推移

そのため，近年では介護者の代わりに生活支援を行うロボットの開発や製品化が進んでいる．トヨタ自動車株式会社の「HSR（Human Support Robot）」，アイロボット株式会社の「ルンバ（Roomba）」，独立行政法人産業技術総合研究所の「パロ」等が挙げられる（図 1.2）．「HSR」は障がい者や高齢者などの自立生活を支援するために開発されたロボットで，「床の上の物を拾う」，「棚から物を取

ってくる」などの仕事が行える．[4]「ルンバ」は家の間取りを学習してマップを作成するため，部屋全体をくまなく掃除できる．[5]「パロ」は人に楽しみや安らぎなどの精神的な働きかけを目的にしたセラピーロボットである．[6]

(a) HSR　　(b)ルンバ　　(c)パロ

図 1.2　生活支援ロボットの例

我々の研究室では，同居者の介護負担の軽減，介護者の自立支援を目的とした「単体多機能型」の生活支援ロボット「KUT-PCR」を開発している（図 1.3）．食事の支援，掃除，トイレまでの誘導，室内温度の調整等，日常の様々な生活支援に対応することを目的としている．これらの動作を行うには，基礎研究としてロボットが生活環境下で自由自在に自律走行することが必要である．

図 1.3　KUT-PCR

1-2　研究目的

人とロボットが混在する生活環境下において，生活支援ロボットは人の動きを阻害せず安全に自律走行することを本研究の目的とする．生活環境には，居間

や廊下など大小さまざまな空間がある．これらの環境で人とロボットが自由に移動すると，問題となる 2 種類の環境が存在する．（Ⅰ）死角からロボットがいきなり現れロボットと人が衝突する環境．（Ⅱ）ロボットが人の行き先を塞ぎ，お互いに身動きの取れない環境．そこで，環境Ⅰ，Ⅱに対して，有効な局所経路計画を提案する．環境Ⅰの場合は，死角から現れた動的障害物（人）と衝突回避するため，動的障害物の数秒後の位置を予測する局所経路計画を提案する．この手法を提案手法Ⅰとする．また環境Ⅱの場合は，ロボットが人に道を譲るように衝突回避を行うため，動的障害物の速度を予測する局所経路計画を提案する．この手法を提案手法Ⅱとする．

1-3　自律走行の関連研究

自律走行には，ある環境に対して目的地までの経路を計画する大局的経路計画と，ロボットの近傍の障害物認識のみを考慮する局所的経路計画がある．大局的経路計画では，目的地までの最短経路を計画できるダイストラクタ法[7]や A*アルゴリズム[8]がよく知られている．大局的経路計画は，環境全体で経路計画をするため，動的障害物に対して繰り返し処理を行うのは，計算時間の観点から困難な場合がある．そのため，動的障害物には局所的経路計画で衝突回避することが望まれる．局所的経路計画には，人口ポテンシャル法[9]，Vector Field Histogram[10]，DWA アルゴリズム[11]等が挙げられる．しかし，これらの局所的経路計画には，動的障害物が考慮されておらず，環境Ⅱにおいて死角から現れた動的障害物とロボットが衝突する．そこで，動的障害物の数秒後の位置を予測する DWA アルゴリズムを提案する．

1-4　論文構成

メカナムホイールを使用した四輪駆動の全方位方向移動型ロボットをシミュレーション及び実機実験の対象とした．　全方位方向移動型ロボットと搭載している測域センサーについて第 2 章で述べる．本研究は，環境Ⅰ，Ⅱに対してそれぞれ有効な提案手法Ⅰ，Ⅱを提案する．提案手法の実装手法と概要を 3 章で述べる．動的障害物の認識方法を抽出，追従，予測と分けて 4 章で説明する．障害物回避を行う動作計画については 5 章で述べる．また 5 章で，提案手法Ⅰ，Ⅱの衝突回避実験を述べる．提案手法Ⅰでは，シミュレーションにより従来の DWA アルゴリズムと比較し，優位性を示す．提案手法Ⅱは，シミュレーション及び実機実験から有効性を示す. 最後に 6 章で本研究のまとめと今後の課題を述べる．

第 2 章　　全方位方向移動型ロボット

室内環境では，ロボットが狭い領域でも難なく移動することが求められている．本研究では，その場回転や任意方向への平行移動が可能な全方位方向移動型ロボットを使用した．このロボットは，四輪駆動でメカナムホイールという車輪を搭載している（図 2.1）また，このロボットは，車体前方部分に測域センサーを搭載している．測域センサーからのデータにより周囲の環境を認識できる．この測域センサーについては，節 2.2 で説明する．

単純な車輪が搭載されているロボットでは，車輪が回転する方向にしか移動できず，任意方向に移動できない．全方向移動を可能とする車輪としてメカナムホイールやオムニホイールが挙げられる．メカナムホイールは，1978 年，Bengt IIon によって設計された車輪である．[13]メカナムホイールは車輪の円周上に自由に回転するフリーローラが備わっている．このフリーローラは車輪の回転方向に対して 45°傾いた方向に取り付けられている．この構造により，フリーローラの自由に回転方向に対して，90°傾いた向きに駆動力が働く（図 2.2）．これにより駆動輪の回転方向だけでなく，多種多様な動きが可能となる．

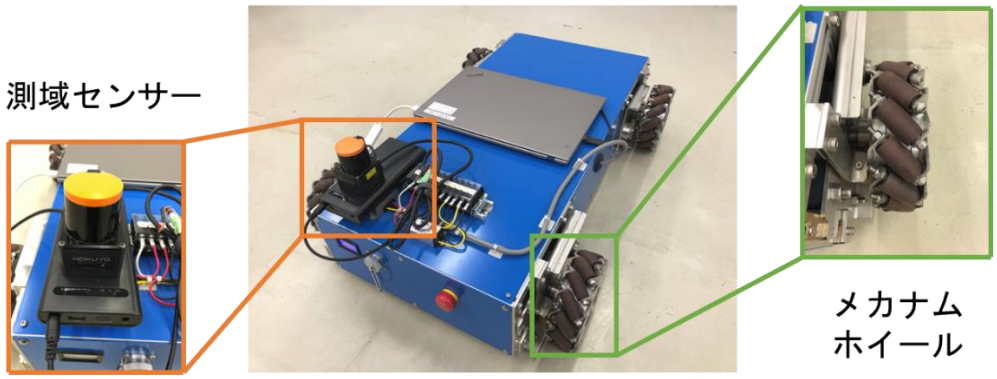


図 2. 1　全方位方向移動型ロボット

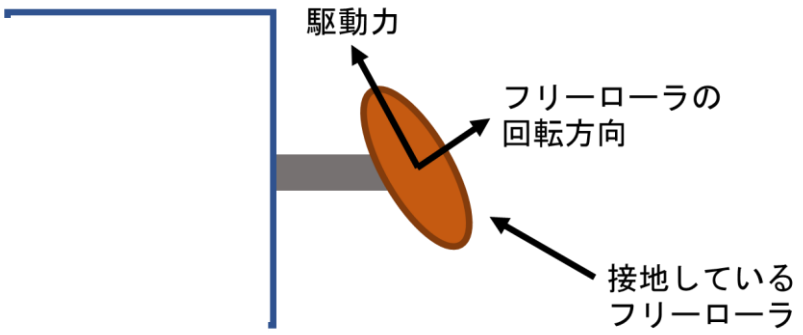
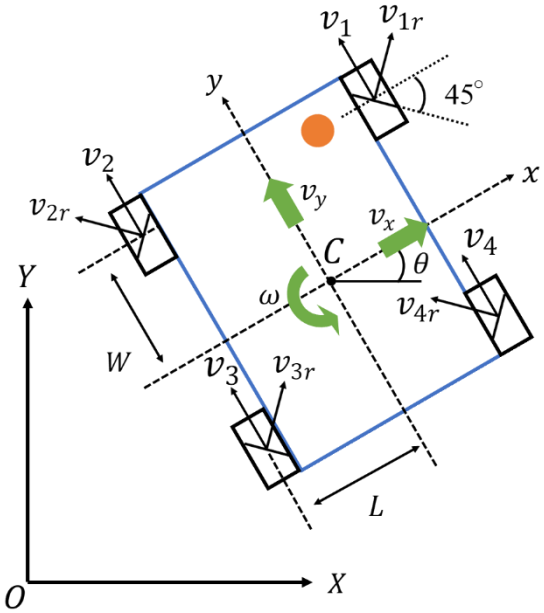


図 2. 2　フリーローラ

2-1　運動学モデル

全方位方向移動型ロボットを制御するために，運動学モデルを導出する．自律走行で用いられる制御入力は車体本体の速度ベクトルである．ここで全方位方向移動型ロボットの運動モデルを導出する．各ホイールと速度及び座標系の関係を示す（図 2.3）．ロボットの中心を相対座標とする．



$\Sigma(X, Y, O)$:絶対座標系
$\Sigma(x, y, C)$:相対座標系
v_i:各ホイールの速度ベクトル
v_{ir}:各フリーローラの
　回転方向の速度ベクトル
v_x:ロボットのx方向の速度
v_y:ロボットのy方向の速度
θ:ロボットの姿勢角
ω:ロボットの角速度

W:ドレッド/2
L:ホイールベース/2

図 2. 3　全方位方向移動型ロボットの運動学モデル

各車輪の速度を$v_i(i = 1,2,3,4)$，各車輪の角速度を$\omega_i(i = 1,2,3,4)$とする．v_{ir}は床に接触している各フリーローラの回転方向の速度である．相対座標$C(x, y)$で各車輪のxy成分$v_{ix}, v_{iy}(i = 1,2,3,4)$は，各車輪とフリーローラの位置関係を考えることで得られる．フリーローラの取り付け角は 45° であるので各車輪の速度成分は

$$
\begin{aligned}
v_{1x} &= \frac{v_{1r}}{\sqrt{2}}, v_{1y} = v_1 + \frac{v_{1r}}{\sqrt{2}} \\
v_{2x} &= -\frac{v_{2r}}{\sqrt{2}}, v_{2y} = v_2 + \frac{v_{2r}}{\sqrt{2}} \\
v_{3x} &= \frac{v_{3r}}{\sqrt{2}}, v_{3y} = v_3 + \frac{v_{3r}}{\sqrt{2}} \\
v_{4x} &= -\frac{v_{4r}}{\sqrt{2}}, v_{4y} = v_4 + \frac{v_{4r}}{\sqrt{2}}
\end{aligned}
\tag{1}
$$

v_x, v_y，ωは絶対座標系におけるロボットのxy成分の速度，及び角速度を示す．図 2.3 より各車輪のxy成分の速度v_{ix}, v_{iy}はv_x, v_yとωで表現される．

$$\begin{aligned} v_{1x} &= v_x - L\omega, v_{1y} = v_y + W\omega \\ v_{2x} &= v_x - L\omega, v_{2y} = v_y - W\omega \\ v_{3x} &= v_x + L\omega, v_{3y} = v_y - W\omega \\ v_{4x} &= v_x + L\omega, v_{4y} = v_y + W\omega \end{aligned} \tag{2}$$

ここで，Lはホイールベースの半分の値，Wはドレットの半分の値を示す．式（1），（2）より，各車輪速度v_iと，ロボット本体の速度v_x, v_yと角速度ωとの関係式が求まる．

$$\begin{aligned} v_1 &= -v_x + v_y + (L+W)\omega \\ v_2 &= v_x + v_y - (L+W)\omega \\ v_3 &= -v_x + v_y - (L+W)\omega \\ v_4 &= v_x + v_y + (L+W)\omega \end{aligned} \tag{3}$$

車輪の半径をRとおくと，車輪の角速度は$v_i = R \times \omega_i$と求めらる．車輪の角速度は式（3）より

$$\begin{aligned} \omega_1 &= \frac{1}{R}\{-v_x + v_y + (L+W)\omega\} \\ \omega_2 &= \frac{1}{R}\{v_x + v_y - (L+W)\omega\} \\ \omega_3 &= \frac{1}{R}\{-v_x + v_y - (L+W)\omega\} \\ \omega_4 &= \frac{1}{R}\{v_x + v_y + (L+W)\omega\} \end{aligned} \tag{4}$$

よって，全方位方向移動型ロボットの逆運動学方程式は，式(4)を行列で書き直し次式となる．

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \frac{1}{R}\begin{bmatrix} -1 & 1 & (L+W) \\ 1 & 1 & -(L+W) \\ -1 & 1 & -(L+W) \\ 1 & 1 & (L+W) \end{bmatrix}\begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \tag{5}$$

続いて，離散系の運動制御の式を導出する．連続系における制御入力後のロボットの位置は

$$\begin{aligned} X(t) &= X_0 + \int_0^t v_X(t)\mathrm{d}t \\ Y(t) &= Y_0 + \int_0^t v_Y(t)\mathrm{d}t \\ \theta(t) &= \theta_0 + \int_0^t \omega(t)\mathrm{d}t \end{aligned} \tag{6}$$

時刻tにおける関数は$f(t)$の形で表記した．X_0, Y_0, θ_{O0}は絶対座標系における初期位置と姿勢を表している．またロボットの初期状態を絶対座標とした場合，

X_0, Y_0, θ_0は零となる．式（6）を両辺tで微分すると

$$\begin{aligned}\frac{\mathrm{d}X}{\mathrm{d}t} &= v_X \\ \frac{\mathrm{d}Y}{\mathrm{d}t} &= v_Y \\ \frac{\mathrm{d}\theta}{\mathrm{d}t} &= \omega\end{aligned} \tag{7}$$

式（7）を離散系で書き換えると

$$\begin{aligned}\frac{X(kT) - X\big((k-1)T\big)}{T} &= v_X(kT) \\ \frac{Y(kT) - Y\big((k-1)T\big)}{T} &= v_Y(kT) \\ \frac{\theta(kT) - \theta\big((k-1)T\big)}{T} &= \omega(kT)\end{aligned} \tag{8}$$

時刻kTにおける関数は$f(kT)$の形で表記した．ここでTはサンプリング周期を表している．また相対座標$C(x, y)$から絶対座標$O(X, Y)$への座標変換式は

$$\begin{bmatrix} v_X \\ v_Y \\ \omega \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \tag{9}$$

v_x, v_y, ωは相対座標系におけるx, y成分の速度，車体の角速度を示す．θはロボット本体の姿勢角である．式(8)，(9）より全方位方向移動型ロボットにおける離散系の運動制御の方程式は

$$\begin{bmatrix} X(kT) \\ Y(kT) \\ \theta(kT) \end{bmatrix} = \begin{bmatrix} X\big((k-1)T\big) \\ Y\big((k-1)T\big) \\ \theta\big((k-1)T\big) \end{bmatrix} + T \begin{bmatrix} \cos\big(\theta(kT)\big) & -\sin\big(\theta(kT)\big) & 0 \\ \sin\big(\theta(kT)\big) & \cos\big(\theta(kT)\big) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x(kT) \\ v_y(kT) \\ \omega(kT) \end{bmatrix} \tag{10}$$

$v_x(kT), v_y(kT), \omega(kT)$は時刻$kT$おける制御入力を表している．

2-2　障害物認識センサー

障害物を認識するセンサーは多種多様にある．3 次元空間を認識したい場合，3D ライダーや深度カメラがあり（図 2.4），2 次元平面を認識したい場合，測域センサーや超音波センサーなどが挙げられる．

測域センサーの原理は，レーザー光を照射し，物体から反射させて認識している．そのレーザー光が移動している時間から，物体との距離を求める．欠点として，光を吸収する黒色の物体や光をすり抜けるガラスなどは光を反射させないため物体を認識することができない．超音波センサーは音の反射を利用しているため，物体によって精度が変化しない．測域センサーのメリットは，超音波センサーに比べて距離精度が良いことである．超音波センサーの場合，検知する物体の姿勢や物体までの距離によって，物体が検知できないことがある．[13]深度カメラは，レンズの歪みなどでキャリブレーションを行うため，計測範囲で精度のずれが生じる．3D ライダーは 3 次元空間を認識できるが，データの扱いが難しく，測域センサーに比べて非常に高価である．

(a)深度カメラ　　(b)3D ライダー

図 2. 4　センサーの種類

本研究では，北陽電気（株）の測域センサー「UST-10LX」を使用した（図 2.5）．この測域センサーは，検知距離が最大 10[m]であり，生活環境下において十分な距離である．また W50×D50×H70 と小型であるため，ロボットに搭載しやすい大きさである．

図 2. 5　　UST-10LX

第 3 章　　実装手法と概要

3-1　　実装手法

本研究での自律走行の構成は，大局的経路計画，局所的経路計画，リカバリー動作，障害物認識の４つに分かれる．これらの関係性を図 3.1 に示す．まず大局的経路計画のプログラムに目的地が与えられ，グローバルコストマップ上で経路が求められる．次に経路が局所的経路計画のプログラムに引き渡され，ローカルコストマップ上で軌道が求められる．軌道から，ロボットに命令される速度ベクトルが求まり，ロボットが移動する．また，大局的経路計画の失敗や局所的経路計画の制御入力が得られないとき，ロボットは停止する．その場合，有効な経路を探索するためにリカバリー動作が用意されている．

ここで，コストマップとは障害物を表現した地図データのことを示しており，詳細は節 3.2 で説明する．大局的経路計画に渡されるコストマップは，グローバルコストマップと呼ばれ，環境全体を表現する．また，局所的経路計画に渡されるコストマップは，ローカルコストマップと呼ばれ，ロボットの近傍のみを表現する．

提案手法 I（動的障害物の位置予測を行う局所的経路計画）は，障害物認識プログラムで予測された数秒後の動的障害物を反映させたローカルコストマップを実装した．そうすることで，動的障害物をも考慮した局所的経路計画が可能である．動的障害者の予測方法は，第 4 章で述べる．

提案手法Ⅱ（動的障害物の速度予測を行う局所的経路計画）は，リカバリー動作として実装した．詳細は，第 5 章の 5.3 節で述べる．

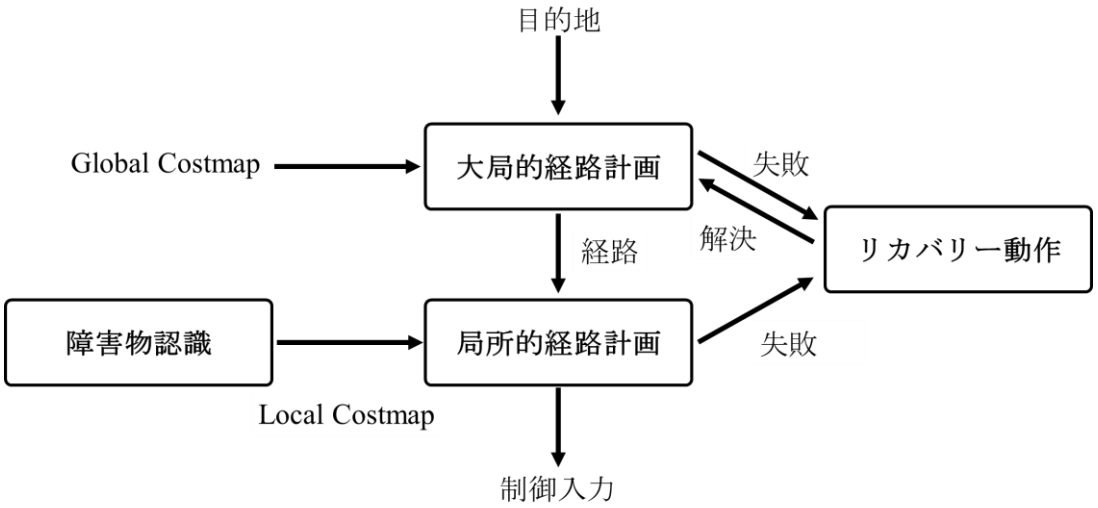


図 3. 1　自律走行の構成

またロボットは，モーターの精度，床の摩擦，ロボットの誘拐などの要因で，制御入力通りに動かないことがある．そのため，自己位置推定と呼ばれる技術が用いられる．ロボットが自身の位置を推定し，制御入力から計算された位置と実際の位置との誤差を修正することが可能である．

上記の実装のために ROS の Navigation Stack を使用した[14]．Navigation Stack には，自己位置推定の amcl，大局的経路計画の global_planner，局所的経路計画の dwa_local_planner 地図データを扱う costmap_2d がパッケージとして用意されている．また，リカバリー動作には，コストマップのある領域のコスト値を消去する clear_costmap_recovery やロボットを 360 度回転させて空きスペースを探す rotate_recovery などがある．大局的経路計画，局所的経路計画，リカバリー動作を図 3.1 のように各データを取り扱いながら統合して動かすために，move_base がある．

Amcl には適応的モンテカルロ位置推定(Adaptive Monte Carlo Localization), global_planner は, ダイストラクタ法や A*アルゴリズムを, dwa_local_planner は, DWA アルゴリズムが実装されている． costmap_2d ではコストマップと呼ばれる地図データが実装できる．

3-2　コストマップ

コストマップはグリッド表現された地図が階層構造になっており，各階層の地図をレイヤーと呼ぶ．（グリッド表現とは，二次元平面を格子状に区切り，各セルにコスト値を割り当てることで障害物を表現できる）各レイヤーは，個別の障害物が表現できる．各レイヤーが呼び出され，マスターレイヤーにコスト値が埋まっていく．Navigation Stack には，Static layer，Obstacle layer，inflation layer の 3 つのレイヤーが用意されている．また本研究では，障害物認識により予測された動的障害物をローカルコストマップに反映させるため，新たな Prediction Obstacle Layer を追加した．各レイヤーを図 3.2 と共に説明する．

Static Layer

事前に生成された地図データ障害物を読み取り，壁など固定された障害物を表現する（図 3.2.a）.

Obstacle Layer

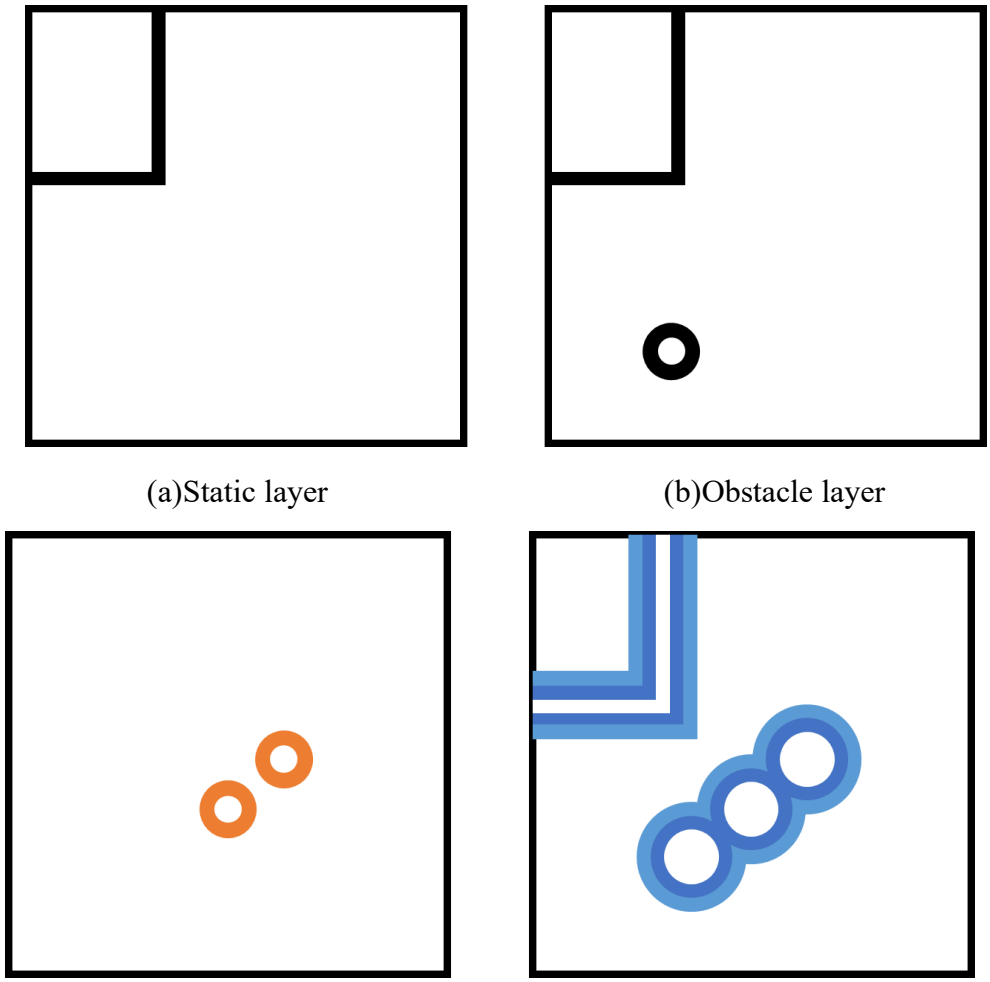
センサーデータによって読み取られた障害物を表現する．固定された障害物だけでなく，突発的に現れた新しい障害物を反映できる（図 3.2.b）.

Prediction Obstacle Layer

障害物認識によって予測された障害物を表現する（図 3.2.c）.

Inflation Layer

動作計画を行うとき，障害物と衝突しないために，ロボット自身の大きさを地図データに含めなければならない．このレイヤーでは，障害物の周りにロボットの大きさを表現した新しい値を追加する（図 3.2.d）．

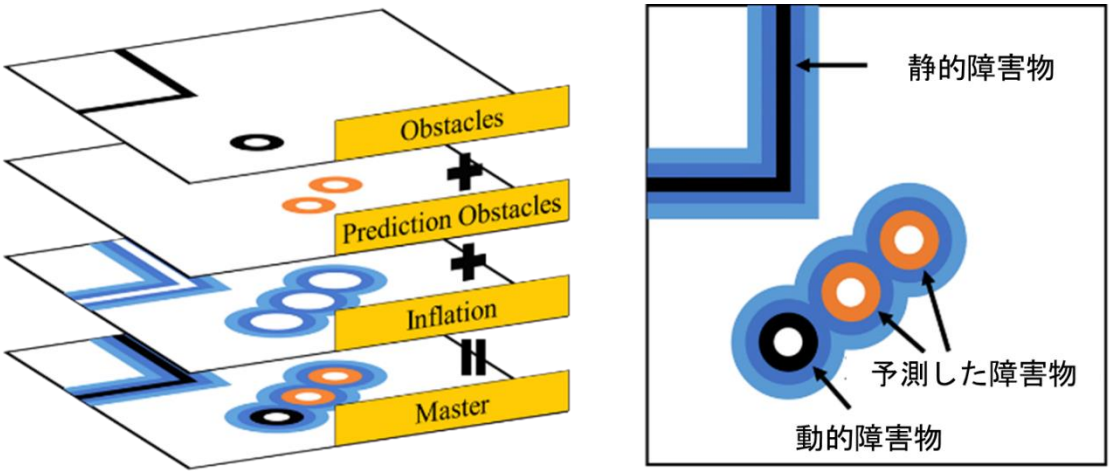


(a)Static layer　(b)Obstacle layer

(c)Static layer　(d)Obstacle layer

図 3.2 Costmap layers

局所的経路計画に提供するローカルコストマップは，Obstacle Layer, Prediction Obstacle Layer, Information Layer の順番で呼び出されるようにした（図 3.2）．Prediction Obstacle Layer を追加したことで，局所的経路計画である DWA アルゴリズムは動的障害物にも衝突しない軌道を選択する．



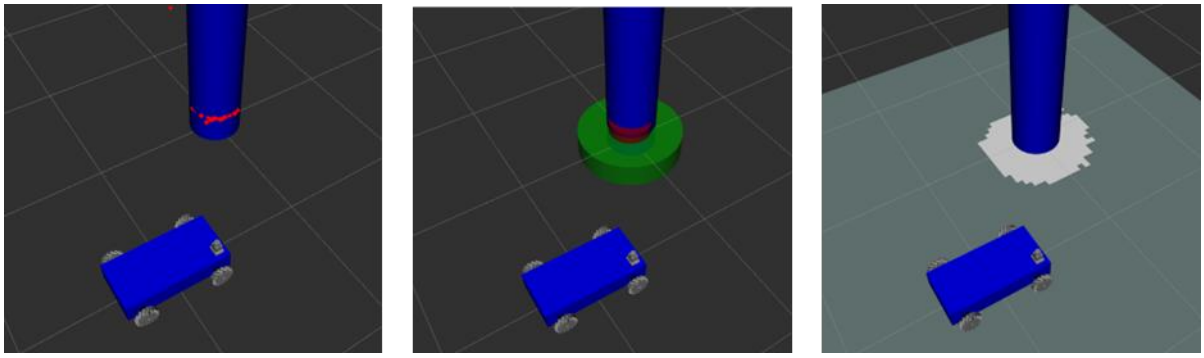
(a)積層された各レイヤー　　(b) 3 種類の障害物

図 3.3　コストマップ

第 4 章　　障害物認識

動的障害物の回避には，動的障害物の位置情報と速度情報を必要とする．位置情報と速度情報から，動的障害物の回避に必要な予測行為が行える．そこで本章では動的障害物の位置情報と速度情報を求める方法について述べる．

本研究では，第 2 章でも述べた通り，障害物を認識するセンサーとして測域センサーを採用した．測域センサーから得られるの生データは点群データであり，局所的経路計画などで使用するには，データとして扱いづらい．そこで，障害物認識では，測域センサーの生データから，動作計画が使用可能なデータに変更する．測域センサーのデータを基に周囲環境を表現する方法は，主に 2 種類に分類でき，グリッド表現とベクトル表現である．グリッド表現は障害物を格子状で区切られたセルに，障害物の有無を表す数値を割り当てる．ベクトル表現では，線分，円，楕円など定義済みの単純な幾何学模様に障害物をモデル化する．本研究では，測域センサーから得られる Laser Scan，ベクトル表現された円形障害物，グリッド表現されたコストマップを使用した(図 4.1).



(a)Laser Scan　　(b)円形障害物　　(c)コストマップ

図 4. 1 障害物データ

障害物認識の流れを示す（図 4.2）．まず，Laser Scan から円形障害物を抽出する．抽出された円形障害物は複数個あり，それらを個別に追従する．各円形障害物に対してカルマンフィルタを適用することで，位置と速度を推定する．障害物は等速モデル(CV model)と仮定し，推定された位置と速度から数秒後の位置を予測する．最後に予測された円形障害物をグリッド表現されたコストマップに変換する．障害物抽出及び障害物追従方法は，M. Przybyła の手法を用いた．[15]

カルマンフィルタは，R. E. Kalman によって提供された推定アルゴリズムの 1 つである．[16]カルマンフィルタは，不確定で不確実な測定値に基づいて，未知の変数を計算する．また，カルマンフィルタは過去の推定値に基づいて将来のシステムの状況を予測することができる．

節 4-1 では障害物検知で使用した障害物追従及び障害物追従の方法について，節 4-2 では障害物予測で使用したカルマンフィルタと等速モデルについて説明する．

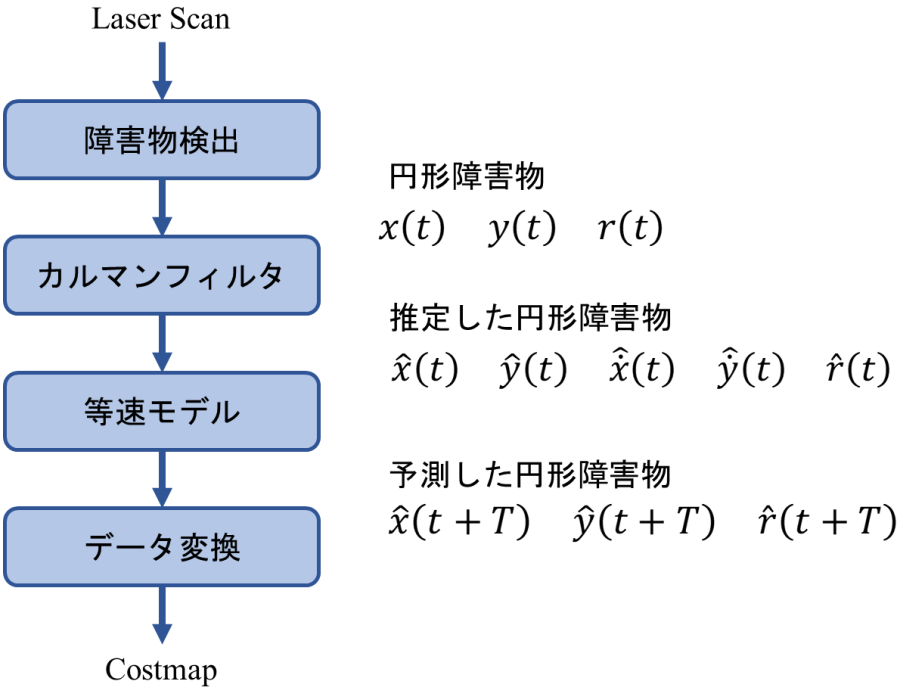


図 4. 2 障害物認識の流れ

4-1　障害物検出

4-3-1　障害物抽出

障害物抽出の目的は，ロボットの移動領域に存在する物体を集合Oで表現し提供することである．通常，測域センサーの生データは量の多さから適切なデータではない．本章では，測域センサーからの障害物を，幾何学モデルである線分または円のどちらかで近似する．生活環境下において存在する壁やテーブルなどの人口的な構造物は線分の近似に，人やペットなど比較的小さい物体においては円で近似することが有効である．ここで測域センサーのデータと幾何学モデルを定義する．

Laser Scan（測域センサーのデータ）

測域センサーのデータを Laser Scan と呼ぶことにする．測域センサーは 0 度から走査角度の最大値まで角度分解能で区切られた全角度における障害物までの距離を取得する．ゆえに Laser Scan は障害物が極座標で記述された集合のデータである（測域センサー位置が原点となる）．

線分 l

2 つの離れた点により表現される．

$$l \triangleq \{\vec{p}_1, \vec{p}_2\} = \{(x_1, y_1), (x_2, y_2)\} \tag{1}$$

円 c

半径と円の中心座標により表現される．

$$c \triangleq \{r, \vec{p}_0\} = \{r, (x_0, y_0)\} \tag{2}$$

結果として得られる障害物の集合Oを次のように定義する．

$$O \triangleq \{L, C\} \tag{3}$$

ここで集合Lは線分障害物の集合，集合Cは円形障害物の集合を表す．以下の節では，抽出処理の詳細について説明する．

1. グループ化

Laser Scan は，隣接する点の距離を調べることでグループ化できる．グループ化の目的は，おそらく別々の障害物を表す部分集合$S_j (j \in \{1, \dots, N_S\})$を提供することである．点$\vec{p}_i (i \in \{2, \dots, N_A\})$は，以下の距離条件を満たす場合，点$\vec{p}_{i-1}$のグループに割り当てられる．

$$d_{i-1}^i < d_{group} + R_i d_p \tag{4}$$

$$d_{i-1}^i \triangleq |\vec{p}_i - \vec{p}_{i-1}| \tag{5}$$

d_{i-1}^iは点$\vec{p}_i$と点$\vec{p}_{i-1}$のユークリッド距離，R_iは点$\vec{p}_i$の距離（図 4），d_{group}はユーザー定義のグループ化の閾値，d_pはユーザー定義の距離割合で，原点から離れた点に対して基準を緩くするものである．式（4）を満たさない場合，新たなグループが作成される．得られたグループは点の部分集合を構成し，さらに分割手続きが行われる．

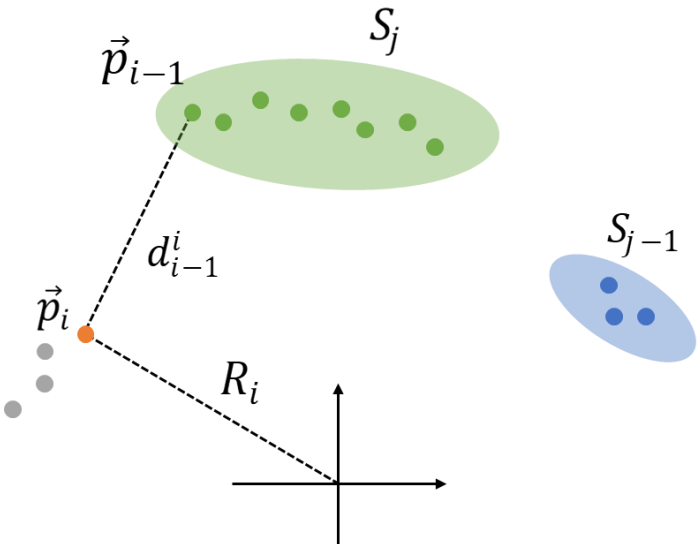


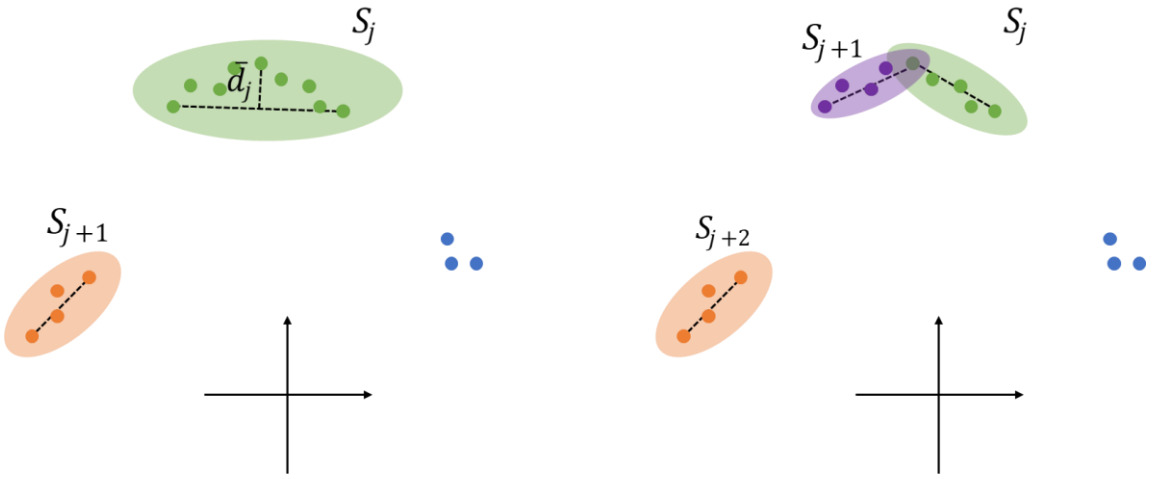
図 4. 3　グループ化の様子

2. 分割化

部分集合S_jは，別々の 2 つの部分集合に分割される可能があるため検査する．簡略化のためN_{min}からなるグループは処理しない．分割の手順は，グループの 2 つの端点に直線を引き，この直線から最も離れているグループ点を探す（図 4.4）．部分集合を 2 つに分割する基準は以下のように定義される．

$$\bar{d}_j > d_{split} + R_j d_p \tag{6}$$

$\bar{d}_j$は直線から部分集合S_jで最も離れている点の距離，d_{split}はユーザーが定義した分割の閾値，R_jは部分集合S_jで最も離れている点の原点からの距離である．分割点は新しい双方の部分集合に割り当てられる．この手順は，分割が行われなくなるまで，それぞれの新しい部分集合に対して再帰的に繰り返される．分割後の新しい部分集合の数N_S^+は，前の数N_Sより小さくなることも，同じになることも，大きくなることもある．



(a) 分割前　　(b) 分割後

図 4. 4　分割化の様子

3. 線分化

すべての部分集合$S_m (m \in \{1, \dots, N_S^+\})$は，線分モデル$l_m$で近似される．近似は 2 つの工程からなる．まず，部分集合に属するN_m個のすべての点に対して，線形回帰により直線を構築する．

$$\begin{bmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_{N_m} & y_{N_m} \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} -C \\ \vdots \\ -C \end{bmatrix} \tag{7}$$

A, B, Cはパラメータ，$x_i, y_i (i \in \{1, \dots, N_m\})$は部分集合に属する各点である．パラメータを計算するために，最小二乗法で求める．この問題の解は無限に存在するので，$C_m \neq 0$と任意に設定し，A, Bについてのみ問題を解くことができる．

$$\begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_{N_m} & y_{N_m} \end{bmatrix}^{\dagger} \begin{bmatrix} -C \\ \vdots \\ -C \end{bmatrix} \tag{8}$$

記号†はムーア・ペンローズ逆行列を表す．次に，部分集合の 2 つの端点が回帰直線に投影され，最終的な線分l_mが形成される（図 9）．抽出された線分は集合Lとなる．

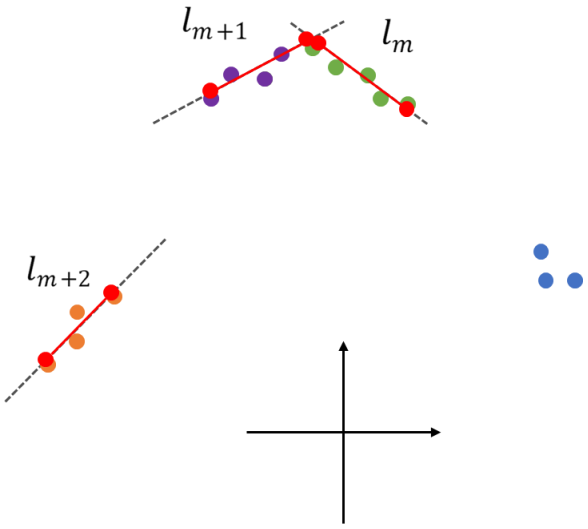


図 4. 5　線分化の様子

4. 円の抽出

線分から直接，円形の障害物を得る．集合Lの各線分ト$l_n (n \in \{1, \dots, N_L\})$に対して，底辺が線分と一致し，中心点が原点から離れた位置にある正三角形を構成する．次に，この三角形を基にして，外接円c_nを構成する（図 4.6）．外接円の半径は次式で与えらる．

$$r_n = \frac{\sqrt{3}}{3} \bar{l}_n \tag{9}$$

$\bar{l}_n$は線分の長さを表す．円の中心点は次式となる（図 4.6）．

$$\vec{p}_{0n} = \frac{1}{2} (\vec{p}_{1n} + \vec{p}_{2n} - r_n \vec{n}_n) \tag{10}$$

$\vec{n}_n$は原点を指す線分の法線ベクトルである．次に半径をユーザーのマージンr_dで拡大する．得られた半径がユーザー定義の閾値r_{max}より小さい場合，その円

は集合Cに追加される．円の許容半径を設定する理由は，抽出された線分が非常に長い場合（例えば，壁），そのような線分から円を抽出すると，移動範囲の大きな領域を被さる物体になってしまうからである．

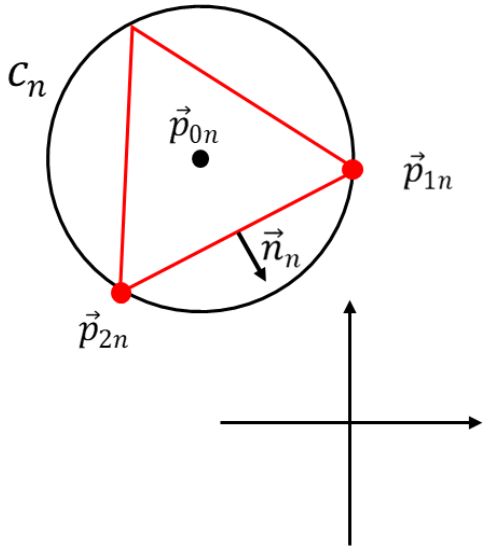


図 4. 6　線分障害物から円形障害物の抽出

4-3-2　障害物追従

本章の冒頭でも述べた通り，障害物追従には対応一致問題がある． Laser Scan は測域センサーから周期的に送られており，抽出される障害物の数は異なる場合がある．さらに，リスト内の出現順序もサンプルごとに異なる可能性がある．対応一致問題は，時刻 k-1 の Laser Scan から抽出された複数の障害物が，時刻 k で抽出されたどの障害物に対応するか見出すことである．ここで，時刻 k-1 で抽出された障害物を「旧障害物」，時刻 k で抽出された障害物を「新障害物」と呼ぶことにする．N_{k-1}，N_kはそれぞれ旧障害物，新障害物の個数を表すとする．任意の 2 つの障害物 $c_i(i \in \{1, \dots, N_k\})$と$c_j(j \in \{1, \dots, N_{k-1}\})$に対して，スカラーコスト関数$c_{ij}$を定義する．

$$c_{ij} \triangleq \sqrt{\left(x_{0i} - x_{0j}\right)^2 + \left(y_{0i} - y_{0j}\right)^2 + \left(r_i - r_j\right)^2} \tag{11}$$

次に，時刻 k-1 と時刻 k の障害物群の差を表すコスト行列$C \in R^{N_k \times N_{k-1}}$を構成する．

$$C = \begin{bmatrix} c_{11} & \cdots & c_{1N_{k-1}} \\ \vdots & \ddots & \vdots \\ c_{N_k 1} & \cdots & c_{N_k N_{k-1}} \end{bmatrix} \tag{12}$$

コスト行列（12）において，行はある新障害物と旧障害物群との関係を表し，列

はある旧障害物と新障害物群との関係を表す.
ここで，関数$\min \mathrm{idx}(A)$を，行列Aの各行に対して最小値を持つ要素の添字を返す関数として定義する. n 番目の行の最小値がユーザー定義のc_{corr}よりも大きい場合，この関数は n 番目の位置に-1 を返す（すべてのc_{ij}は正の値を取るので，-1 と置くと例外扱いになる）.
コスト行列から，次の 2 つのベクトルを作成する.

- 行による最小値の添字を持つベクトル

$$\vec{n}_{rows} \triangleq \left(n_{c1}, \dots, n_{cN_k}\right)^T = \min \mathrm{idx}(C) \tag{13}$$

- 列による最小値の添字を持つベクトル

$$\vec{n}_{cols} \triangleq \left(n_{r1}, \dots, n_{rN_{k-1}}\right)^T = \min \mathrm{idx}(C^T) \tag{14}$$

ある行における最小値の添字が意味することは，ある新障害物に対するコストが最小となる旧障害物の番号を表し，ある新障害物に対応する最も適切な旧障害物である. 行ごとの最小値の添字は, 新障害物とのコストが最小となる旧障害物の添字を格納する（列における最小値を持つベクトルも新と旧を入れ替えることで同様に述べられる）.

$\vec{n}_{rows}$と$\vec{n}_{cols}$を調べることで，いくつかの対応関係を見分けることができる.

- i番目の新障害物は，対応する障害物を持たない．$(n_{ri} = -1)$
- i番目の新障害物は，旧障害物jとただ一つ対応する．$(n_{ri} = j)$
- 新障害物に対応する旧障害物が 2 つ以上ある. $(n_{cj} = i$かつ$n_{ck} = i$ただし$j \neq k)$
- 旧障害物に対応する新障害物が 2 つ以上ある. $(n_{rj} = i$かつ$n_{rk} = i$ただし$j \neq k)$

最初の場合は, データが不足しているため, 更新ステップの実行は不可能である．このような障害物は未追従として扱う．2 番目の場合は，更新ステップの実行は自明である．このような障害物を追従済みとする．3 番目と 4 番目の場合を，障害物融合と障害物分裂と定義する．環境下に動くものがあると，障害物の融合と分裂がよく起こる．障害物の融合（分裂）は，$\vec{n}_{cols}(\vec{n}_{rows})$の重複した添字を調べることで検出できる．同じ新（旧）障害物に対応する旧（新）障害物が 2 つ以上ある場合，それを融合（分裂）と見なす．障害物融合の場合，構成するすべての旧障害物に基づいて新障害物を更新する必要がある．障害物分裂の場合，新障害物は, 元の障害物に基づいて更新されなければならない. 融合または分裂した障害物は，追従済みとして扱われる.

4-2　障害物予測

4-2-1　Kalman Filter

測域センサーによる測定は，精度は高いが正確なものではない．また必ずランダムな誤差を含んでいる．動的障害物の動きは，仮定した運動モデルと一致しない．カルマンフィルタは，モデルの不確かさと観測の不確かさを考慮するために，プロセスノイズw_kと観測ノイズv_kが組み込まれる．プロセスノイズと観測ノイズは，平均値が零の白色ガウスノイズが仮定される．

$$w_k \sim N(0, Q) \tag{15}$$

$$v_k \sim N(0, R) \tag{16}$$

Q, Rはプロセスノイズと観測ノイズの共分散行列であり，定数である．

カルマンフィルタは，初期状態X_0と初期推定誤差行列P_0及びノイズ共分散行列Q，Rを要求し，次の時刻の推定値X_1と，誤差共分散をP_1として更新する．そして，X_1，P_1，Q，Rの値を用いて，次の推定値を更新する，といった処理を繰り返し行う．X_0, P_0が正確でない場合，初期誤差が長い期間で推定値に影響を及ぼすことがある．また，Q，Rが不正確に指定された場合，推定値は偏るかノイズが多くなる．

カルマンフィルタは，モデルによる予測と観測に基づいて，時刻kにおける状態X_kと誤差共分散P_kを推定される．カルマンフィルタの再帰的処理には，「予測ステップ」と「更新ステップ」の 2 つの異なるステップにわかれる．予測ステップでは，1 つ前の状態推定値を用いて，現在の時点の状態推定値が生成される．予測された状態推定値は，現在の時刻の観測値を用いないため，事前状態と呼ばれる．予測ステップでは，2 つの推定値が生成される．

$$\hat{X}_k^- = A\hat{X}_{k-1} + BU_k \tag{17}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{18}$$

$\hat{X}_k^-, P_k^-$は事前状態推定値，事前誤差分散を表す．

更新ステップでは，式(),()で求められた事前状態，事前誤差分散を現在の観測された状態と組み合わせて，状態と誤差共分散を更新する．現在の観測値を用いられて更新された推定値は，事後状態と呼ばれる．以下の式で提供される．

$$K_k = P_k^- C^T (CP_k^- C^T + R)^{-1} \tag{19}$$

$$\hat{X}_k = \hat{X}_k^- + K_k\left(Y_k - C\hat{X}_k^-\right) \tag{20}$$

$$P_k = (I - K_k C)P_k^- \tag{21}$$

$\hat{X}_k, P_k$は事後状態推定値，事後誤差分散を表す．K_kはカルマンゲインと呼ばれる．推定値の不確かさよりも観測の不確かさが大きい場合，カルマンゲインは小さくなる．これは，事後状態推定値が事前状態推定値に近くなることを意味する．（特にカルマンゲインが零行列である場合，事後状態推定値と事前状態推定値

は一致する）推定値の不確かさよりも観測の不確かさが小さい場合，カルマンゲインは大きくなり，事後状態推定値は観測値に近くなることを意味する．

4-2-2　円形障害物予測

離散系で円形障害物の状態方程式を定義する．

$$X\big((k+1)T\big) = AX(kT) + BU(kT) + W(kT) \tag{22}$$

$$Y(kT) = CX(kT) + V(kT) \tag{23}$$

ここで，Tはサンプリング周期を，$W(kT)$，$V(kT)$は，ホワイトガウスノイズを表す．状態変数$X(kT)$を定義する．

$$X(kT) = [x(kT) \quad \dot{x}(kT) \quad y(kT) \quad \dot{y}(kT) \quad r(kT)]^T \tag{24}$$

円形障害物の動作には等速モデルを，半径の大きさは一定であると仮定する．また，動的障害物の制御入力は観測不可能である．各行列A, B, Cは次のようになる．

$$A = \begin{bmatrix} 1 & T & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{25}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{26}$$

$$B = 0 \tag{27}$$

カルマンフィルタにより，状態変数が推定される．推定された速度，位置から等速モデルにより動的障害物の数秒後の位置を予測する．

$$\begin{bmatrix} \hat{x}\big(kT + T_{predict}\big) \\ \hat{y}\big(kT + T_{predict}\big) \\ \hat{r}\big(kT + T_{predict}\big) \end{bmatrix} = \begin{bmatrix} \hat{x}(kT) \\ \hat{y}(kT) \\ \hat{r}(kT) \end{bmatrix} + \begin{bmatrix} \hat{\dot{x}}(kT) \\ \hat{\dot{y}}(kT) \\ 0 \end{bmatrix} T_{predict} \tag{28}$$

$T_{predict}$は予測時間を，$\hat{x}\big(kT + T_{predict}\big), \hat{y}\big(kT + T_{predict}\big), \hat{r}(kT + T_{predict})$は$T_{predict}$秒後の円形障害物の位置，半径を表す．

4-3　シミュレーション

4-3-1　シミュレーション環境

シミュレーション環境は，3 次元シミュレータである Gazebo を用いた．自律走行中のロボットが通る経路やコストマップの可視化には ROS が提供する 3 次元可視化ツールである Rviz を使用した．シミュレーションで使用したロボットと動的障害物を図 4.7，図 4.8 に示す．ロボットは第 2 章で述べた全方位方向移動型ロボットを Gazebo 上で使用できるように著作がモデル化したものである．人を想定した動的障害物は，円柱とした．円柱の半径は 0.2[m]，高さは 1.7[m]とした．

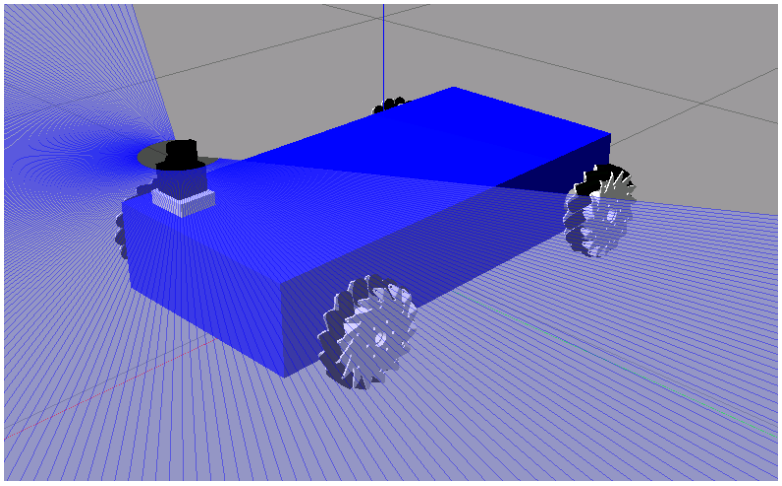


図 4. 7　全方位方向移動型ロボット（Gazebo）

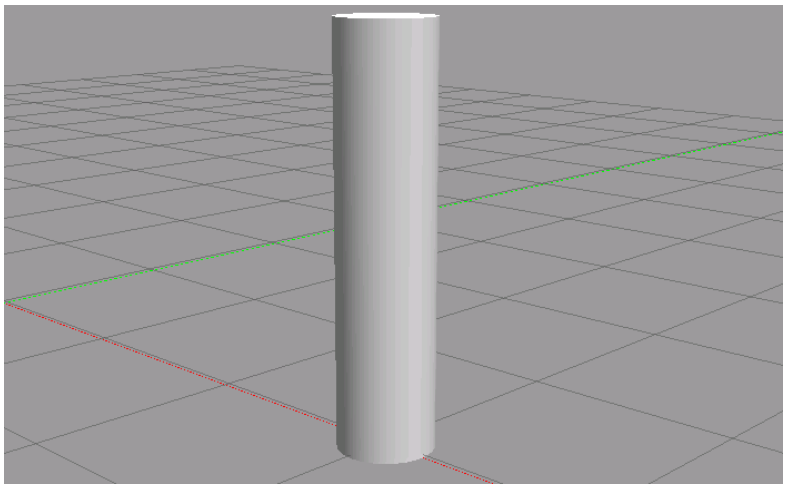


図 4. 8　動的障害物（Gazebo）

4-3-2　障害物認識実験の目的

本研究の提案手法は，動的障害者を回避するために，動的障害者の予測を行う．動的障害物の予測が正確でないとき，適切な局所的経路計画ができず最悪の場合，動的障害物と衝突してしまうことが考えられる．ここでは，Gazebo 上で動的障害物を動かし，そのときの位置と速度が正確に予測できているのか，検証する．予測された動的障害物の軌道と動的障害物が Gazebo 上で移動した真の軌跡を比較する．

4-3-3　障害物認識実験の条件

動的障害物を移動させる軌跡は，四角形と円形とし 2 種類のパターン用意した．ロボットは，動かさず原点に固定した．ロボットに搭載している測域センサーから，軌道を描く動的障害者の位置と速度を推定する．ロボットの中心が絶対座標の原点であり，各点は絶対座標において点（x[m], y[m]）で表す．

（パターン A）

動的障害物に沿わせる四角形の軌跡を図 4.9.a に示す．動的障害物の初期位置は a(1.0, 1.0)とした．各点 a(1.0, 1.0), b(1.0, -1.0), c(3.0, -1.0), d(3.0, 1.0)を順に通り過ぎるようにした．動的障害物の速度は等速として，2.0[m/s]とした．予測時間は 1.0[s]，2.0[s]とした．

（パターン B）

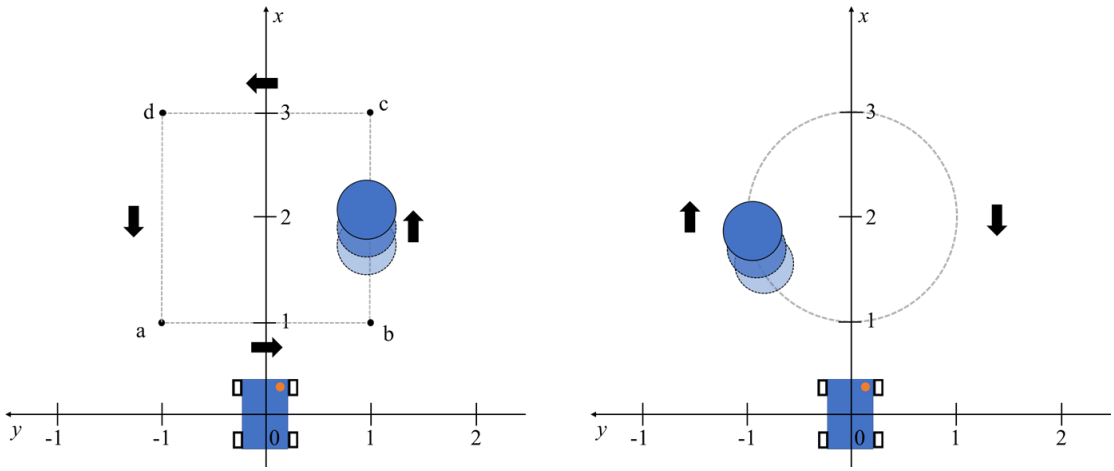
動的障害物に沿わせる円形の軌跡を図 4.9.b に示す．動的障害物の初期位置は（1.0, 0.0）とした．円の中心は(2.0, 0.0)，半径は 1[m]とした．動的障害物の速度は等速，等角速度として，2.0[m/s]とした．予測時間は 1.0[s]，2.0[s]とした．

カルマンフィルタは次のように設計し，パターンＡ，Ｂともに同じ条件にした．初期状態における中心座標と半径は，最初に受け取った円形障害物の中心座標と半径をそのまま適用した．また初期速度は xy 成分ともに 0[m/s]とした．共分散行列P_0, Q, Rは次のように設計した．

$$P_0 = \begin{bmatrix} 500 & 0 & 0 & 0 & 0 \\ 0 & 500 & 0 & 0 & 0 \\ 0 & 0 & 500 & 0 & 0 \\ 0 & 0 & 0 & 500 & 0 \\ 0 & 0 & 0 & 0 & 500 \end{bmatrix}$$

$$Q = \begin{bmatrix} 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0.01 \end{bmatrix}$$

$$R = \begin{bmatrix} 1.0 & 0 & 0 \\ 0 & 1.0 & 0 \\ 0 & 0 & 1.0 \end{bmatrix}$$



(a) 四角形　　(b) 円形

図 4. 9　動的障害物の移動経路

4-3-4　障害物認識実験の結果

パターン A のシミュレーション結果を示す．Laser Scan から抽出された円形障害物の x 成分の値を図 4.10 に，カルマンフィルタにより推定された x 方向の位置と速度の結果を図 4.11，図 4.12 に，等速モデルによる予測値を図 4.13 に示す．またの経路結果を図 4.14 に示す．

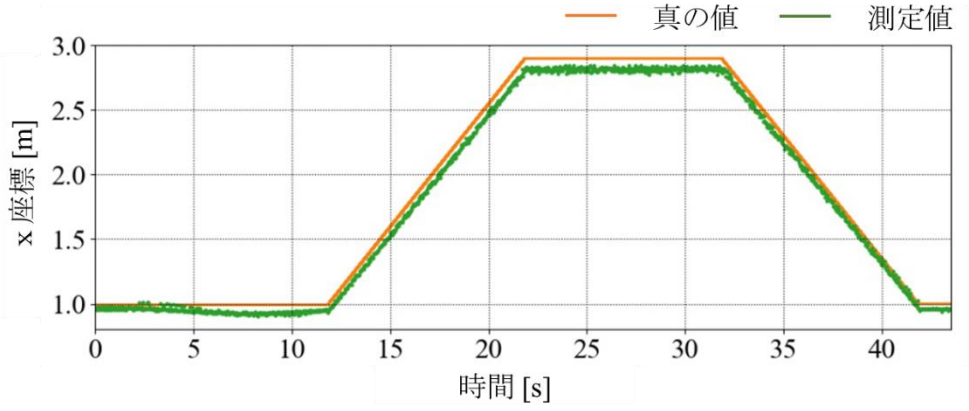


図 4. 10　円形障害物の x 成分（パターン A）

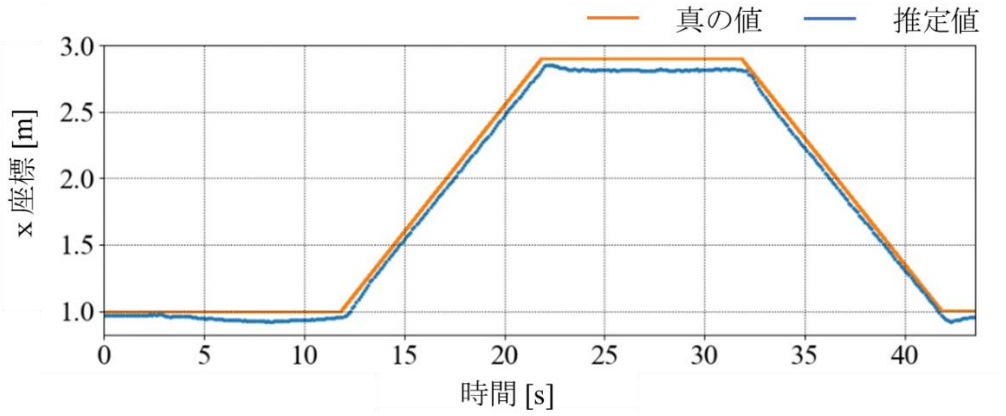


図 4. 11　x 方向の位置推定値（パターン A）

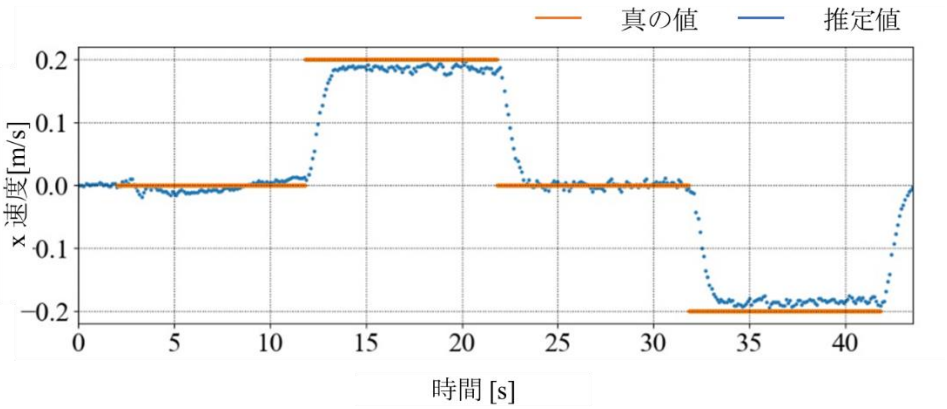


図 4. 12　x 方向の速度推定値（パターン A）

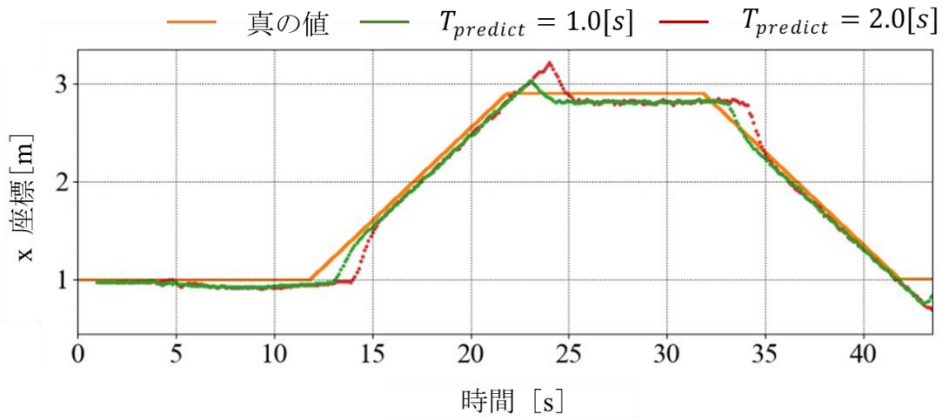


図 4. 13　x 方向の予測位置（パターン A）

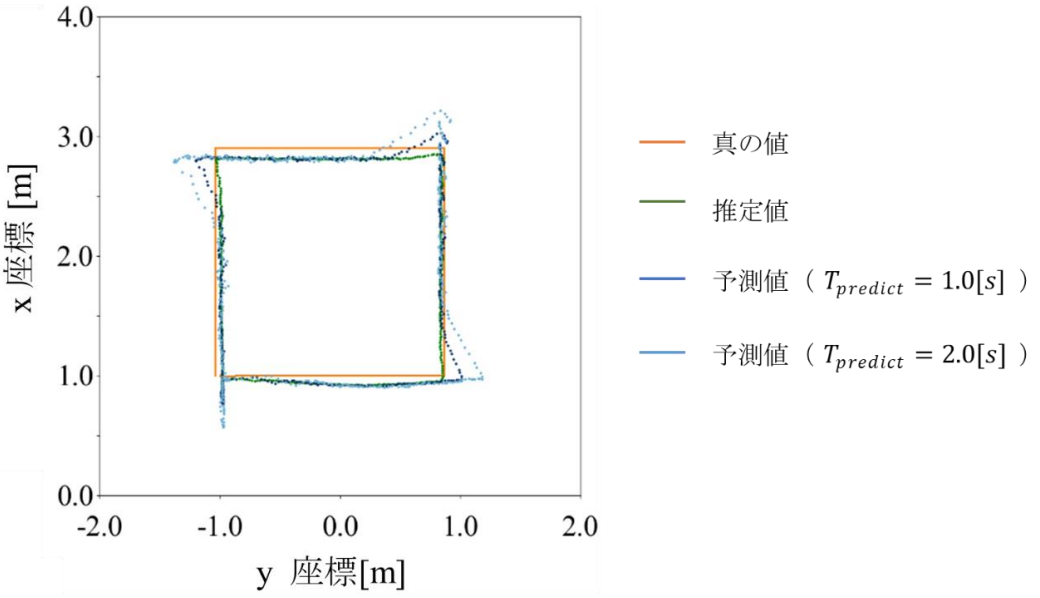


図 4. 14　経路結果（パターン A）

測域センサーから得られる Laser Scan には分散があるため，抽出された円形障害物にも分散が生じる（図 4.10）．円形障害物をカルマンフィルタに通すことで，得られた推定値は分散が小さくなった（図 4.10）．また直線 ac 間と直線 cd 間において，真の軌跡と少しずれている（図 4.10）．これは，Laser Scan から線分障害物を抽出する線分化の過程において，最小二乗法を使用しているが，実際の円形障害物より手前に線分を引いてしまい，円形障害物の中心座標がずれてしまう．この解決策は，円形障害物の半径に値を追加する．半径を余分に大きくすることで，円形障害物は大きくなるが僅かな中心座標のずれを補うことができる．

動的障害物が直線 ab 間や直線 bc 間などの直線区間を移動しているとき，カルマンフィルタから推定された速度は，真の値に十分収束している（図 4.12）．しかし図 4.12 において予測値の経路結果から，点 b，点 c，点 d を通貨したあとの予測値が真の値と大きくずれていることがわかる．また時系列でみると予測位置が 12 秒，22 秒，32 秒の付近で真の値とずれている（図 4.13）．動的障害物の真の値は直感に曲がるため，12 秒付近で 0.2[m/s]から 0[m/s]へと急激に値が変わる．このときの推定速度は 1，2 秒ほど収束していない（図 4.12）．そのため，正しくない推定速度から求められた予測位置は正しく計算されない．動的障害物の運動モデルは等速モデルと仮定したため，等速モデルと大きく離れる挙動を動的障害物が行った場合，予測モデルが機能しなかったため，即応性の悪化繋がったと考えられる．即応性の問題は，動的障害者の移動速度に依存する．動的障害者の速度が速いほど，推定値と真の値の誤差は大きくなる．しかし，図 3.13

からわかるように，予測時間$T_{predict}$を小さくすればするほど，誤差は小さくなる．生活環境下において人が移動する程度の速度であれば，パラメータである予測時間$T_{predict}$を変えることで十分，対応可能であると考えられる．

続いてパターン B のシミュレーション結果を示す．カルマンフィルタにより推定された位置を図 4.3.2.2 に，等速モデルによる予測値を図 4.3.2.2 に，経路結果を図に示す．

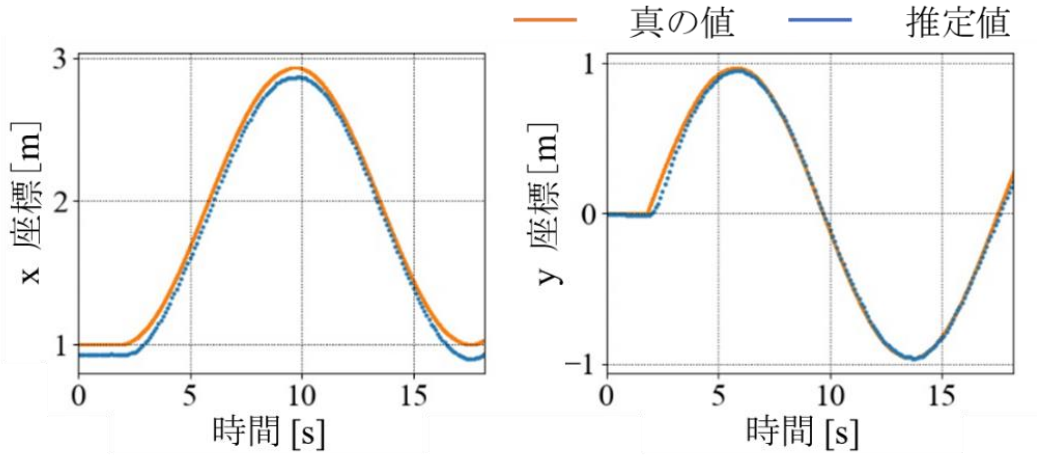


図 4. 15　推定位置（パターン B）

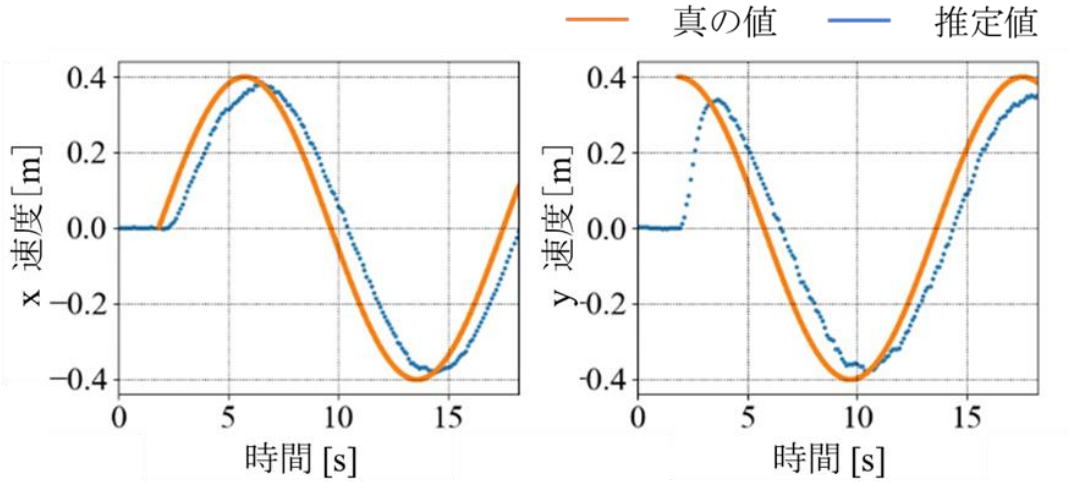


図 4. 16　推定速度（パターン B）

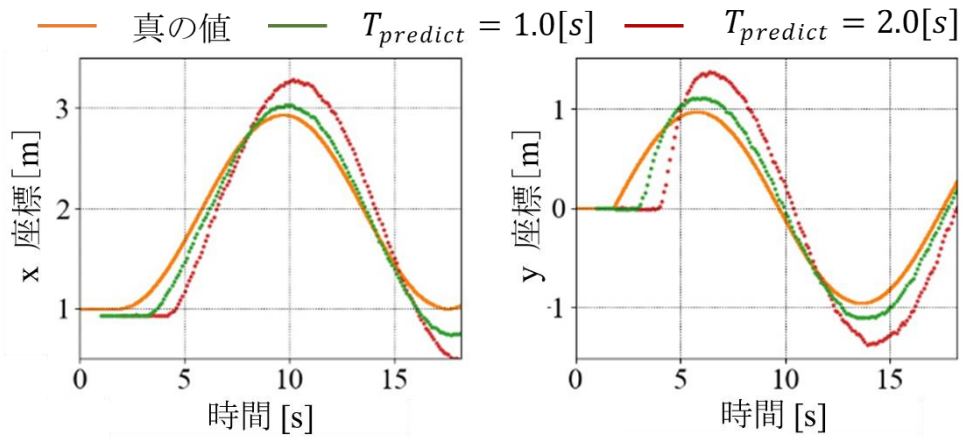


図 4. 17　予測位置（パターン B）

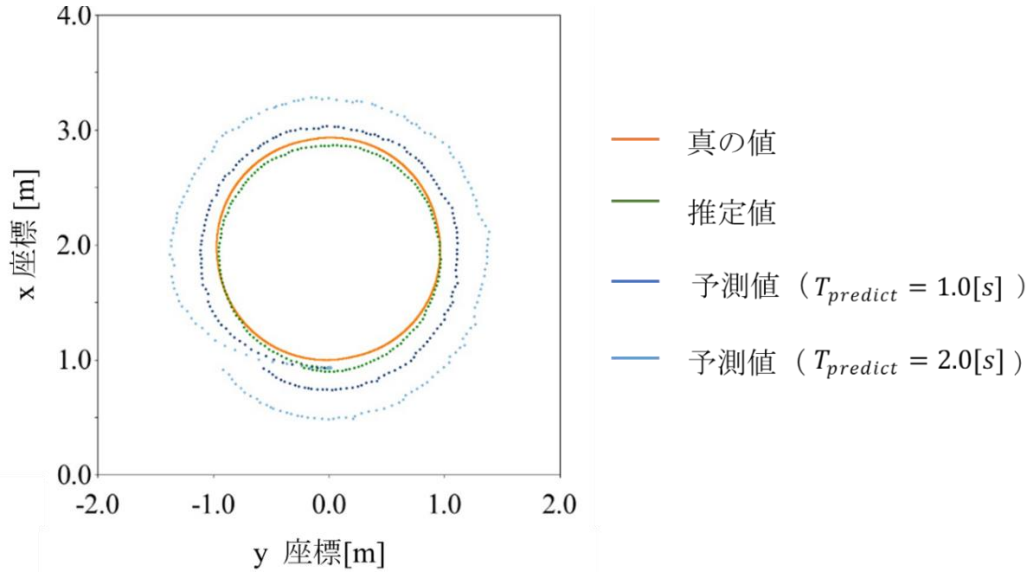


図 4. 18　経路結果（パターン B）

カルマンフィルタから推定された速度は, x 座標, y 座標ともに真の値に近く，推定位置の精度が良い（図 4.15）．しかし，推定された速度は x 軸の場合，真の値より位相が常に遅れており十分に時間が経過してからも収束しなかった（図 4.16）．また y 軸の速度は，位相遅ればかりでなく，カルマンフィルタの初期位置が 0[m/s]であるのに対し，動的障害物の初期速度が 0.4[m/s]であるため，初期誤差が大きく 1 秒程影響を与えていることがわかる．また，予測位置も推定速度の精度に応じて悪化した（図 4.18）．しかし，パラメータ B と同時に真の値と予測値の経路結果は予測時間が小さくなるにつれ, 誤差が小さくなった．予測時間$T_{predict}$を動的環境に応じて，設定することで動的障害者の予測精度は十分であると考えられる．

第 5 章　　動作計画

5-1　動的障害物の位置予測を行う局所的経路計画

局所的経路計画を DWA アルゴリズムとした場合, ロボットは静的障害物と衝突することなく自立走行できる．しかし, 動的障害者とは衝突する場合がある．DWA アルゴリズムは現在の障害物と候補に挙げたロボットの軌道との位置関係を調べ, 衝突しない軌道を選択する．そのため, 現在の障害物に衝突しないように選択された軌道が, 動的障害者の進行方向上であった場合, 動的障害者とロボットは衝突する．第 4 章の手法で動的障害物の$T_{predict}$秒後の位置を予測でき, 第 3 章の実装方法で, その予測された動的障害物を DWA アルゴリズムが認識できる．それにより，現在の動的障害物だけでなく$T_{predict}$秒後の動的障害物をも認識した DWA アルゴリズムは, 動的障害者に対しても衝突しない軌道を選択するため，動的環境であっても安全に自立走行できる．

5-2　Dynamic Window Approach

Dynamic Window Approach（DWA）アルゴリズムの基本的な考えを説明する．

（i）　速度空間（Dynamic Window）を計算する．
（ii）　速度空間の中で，軌道候補群を生成する．
（iii）　軌道候補群の中で，障害物に衝突する軌道候補は破棄する．
（iv）　軌道候補群に対して，評価値関数により最適な軌道を選択する．
（v）　最適な制御入力をロボットに命令する．

以上の処理を繰り返す.

5-2-1　速度空間の計算

制御入力(v_x, v_y, ω)の速度範囲V_s, V_a, V_dを求める．

ロボットの性能，最大速度，最小速度から Vs が求まる．モーターの性能になどによって，これらの値が決まる．

$$V_s = \begin{Bmatrix} v_x \in [v_{x,min},\ v_{x,max}] \wedge \\ v_y \in [v_{y,min},\ v_{y,max}] \wedge \\ \omega \in [\omega_{min},\ \omega_{max}] \end{Bmatrix} \tag{1}$$

障害物の位置とロボットの最大加速度の値から，安全に走行できる制御入力の範囲V_aを求める．

$$V_a = \begin{Bmatrix} v_x \le \sqrt{2dist(v_x, v_y, \omega)\dot{v}_{xb}} \wedge \\ v_y \le \sqrt{2dist(v_x, v_y, \omega)\dot{v}_{yb}} \wedge \\ \omega \le \sqrt{2dist(v_x, v_y, \omega)\dot{\omega}} \end{Bmatrix} \tag{2}$$

ここで，$\dot{v}_{xb}, \dot{v}_{yb}, \dot{\omega}$ はロボットの減速における最大加速度である．$dist(v_x, v_y, \omega)$は，その制御入力の時，障害物と経路の中で最も近い値を出力する関数である．$dist(v_x, v_y, \omega)$の値が 0 の場合，経路と障害物が衝突していることを表す.

モータートルクの限界は，一定時間内に到達可能な最大，最小速度に影響を与える．現在の速度から，次の周期での速度範囲V_dは以下の要件を満たす.

$$V_d = \begin{Bmatrix} v_{xc} - \dot{v}_{xb}\Delta t \le v_x \le v_{xc} + \dot{v}_{xa}\Delta t \wedge \\ v_{yc} - \dot{v}_{xb}\Delta t \le v_y \le v_{yc} + \dot{v}_{ya}\Delta t \wedge \\ \omega_c - \dot{\omega}_b\Delta t \le \omega \le \omega + \dot{\omega}_a\Delta t \end{Bmatrix} \tag{3}$$

v_{xc}, v_{yc}, ω_cは，現在の制御入力を，$\dot{v}_{xa}, \dot{v}_{ya}, \dot{\omega}_a$ロボットの最大値加速度を，$\Delta t$はサンプリング周期を表す.

最終的な速度範囲は，議論された 3 つの速度範囲の交わるところであり，以下のようになる.

$$V_r = V_s \cap V_a \cap V_d \tag{4}$$

5-2-2　軌道候補群の生成

速度範囲の中からサンプリングして，軌道候補を複数生成する．第 2 章から全方位方向移動型ロボットの運動制御の方程式は

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} + \Delta t \begin{bmatrix} cos\theta_{t-1} & -sin\theta_{t-1} & 0 \\ sin\theta_{t-1} & cos\theta_{t-1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{xt} \\ v_{yt} \\ \omega_t \end{bmatrix} \tag{5}$$

v_{tx}, v_{ty}, ω_tは時刻tにおける制御入力である．等速，等角速度を仮定し，軌道候補を生成する.

5-2-3　軌道候補群の評価

軌道候補群を評価する評価関数を定義する.

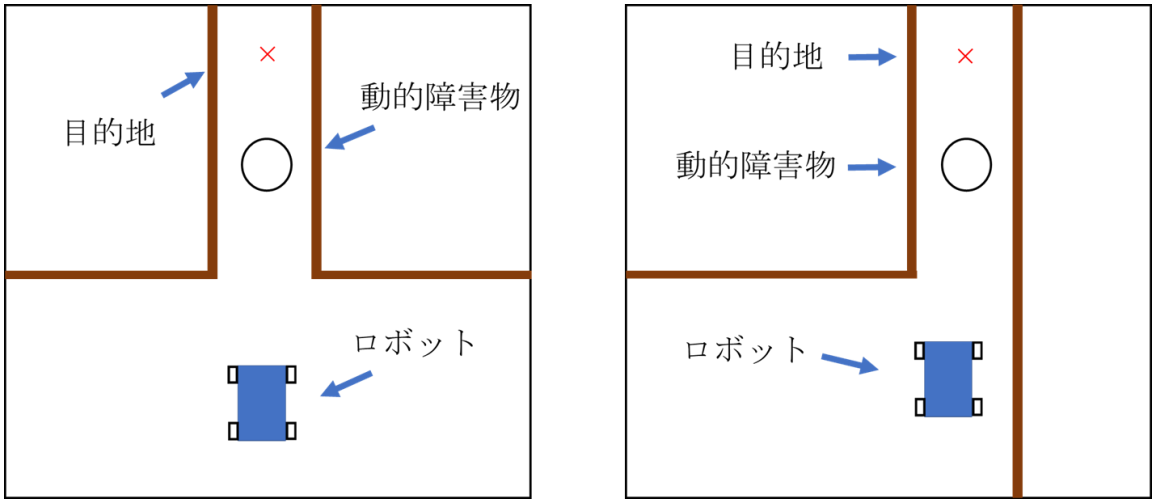
$$G(u_x, u_y, \omega) = \sigma\left(\alpha \times heading(u_x, u_y, \omega) + \beta \times dist(u_x, u_y, \omega) + \gamma \times velocity(u_x, u_y, \omega)\right) \tag{6}$$

$heading(u_x, u_y, \omega)$は，ロボットの方向と目的地の方向の差を表す．目的地に真っ直ぐ向いて場合，評価値が大きくなる．$dist(u_x, u_y, \omega)$は，経路と障害物との最小距離を表す．障害物から遠い程，評価値が大きくなる．$velocity(u_x, u_y, \omega)$は，制御入力の値を使用し，速度が速い制御入力の評価値が大きくなる．α, β, γはそれぞれの重みパラメータである．最後にσはスムージング関数で，制御入力を滑らかにする関数である．

生成された軌道候補群に対して，式（6）で各軌道を評価し，最も評価値が大きい制御入力を最適値として採択する．

5-3　動的障害者の速度予測する局所的経路計画

自律走行は，大局的経路計画，局所的経路計画，リカバリー動作，障害物認識に分かれると 1 章にて述べた．この内，ロボットの挙動を直接決めるのは大局的経路計画と局所的経路計画である．では，環境の要因で大局的経路計画や局所的経路計画が行えない場合，ロボットは適切な軌道を求められずその場停止してしまう．そのとき，リカバリー動作が発動する．ROS のリカバリー動作には，その場回転やコストマップのクリアなどが用意されている．しなしながら動的障害物が行き来する，次の 2 つの環境において対応できない．環境 A，環境 B を図 5.1 に示す．環境 A，B はともに広い空間と狭い空間の 2 つの空間があり，狭い空間は廊下や台所を，広い空間は居間などが想定できる．環境 A は狭い空間と広い空間が繋がっており，動的障害物は狭い空間にいる．このときロボットが目的地まで，移動しようとしたとき，動的障害物が道を塞いでしまい，移動可能な経路は存在しない．一方で，狭い空間から広い空間へ移動しようとする動的障害物にとってもロボットは邪魔であり，ロボットの位置が狭い空間に非常に近い場合は動的障害物もまた移動できない．環境 B においても同様の理由により，動的障害物とロボットはお互いに身動きのとれない位置関係にある．環境 A，B において，ロボットは動的障害物の移動経路を塞いでいるため，道を譲る行為が必要となる．そこで本研究では，動的障害者の速度予測する局所的経路計画を提供する．本提案手法は動的障害物に必ず道を譲ることため，動的障害者の移動経路を確保することを目的にロボットは局所経路計画を行う．そのため本論文では，提案手法 II の局所的経路計画を動線確保動作計画と呼ぶこととする．



(a) 環境 A　　(b)　環境 B

図 5. 1　生活環境

動線確保動作計画の流れを説明する（図 5.2）. 動的障害物の動線を確保するために, 移動する位置に挙げられる位置候補群を用意する. 次に動的障害物の軌道を予測し, 軌道に最も近い位置候補を選択する. 動的障害物が進行する逆の方に, ロボットの軌道を選択する．ロボットが位置候補から十分離れたら静止させる. 位置候補の選択とロボットの軌道選択を繰り返し，動的障害物が位置候補を通り過ぎたときアルゴリズムが終了する.

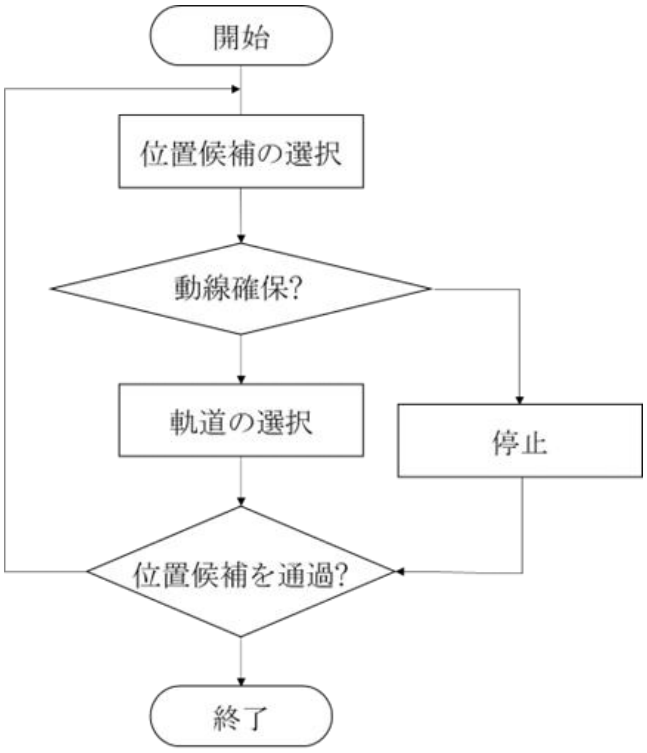
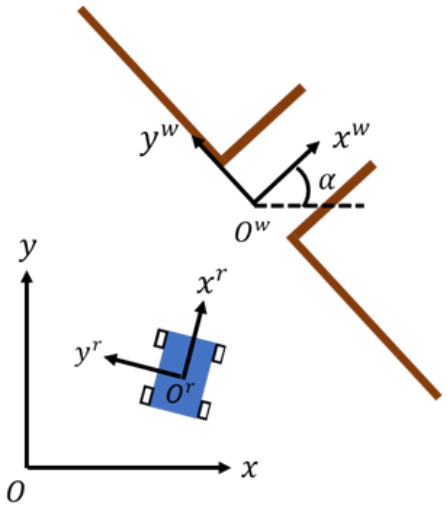


図 5. 2　動線確保動作計画のフローチャート

大局的経路計画に失敗したロボットはリカバリー動作である動線確保動作計画に移る. 動線確保動作計画が終了したとき，リカバリー動作から離れ再度，大局的経路計画と局所的経路計画を行いロボットは目標位置に向かう.

動的障害物の軌道予測, ロボットの軌道選択において, 動的障害物の情報が必要となる．そこで，第 3 章で述べた障害物認識より求められる円形障害物の位置と速度の推定値を使用する.

絶対座標とロボット，壁との座標関係を図 5.3 に示す．壁の座標の原点O^wは出口となる広い空間と狭い空間の境界線上に置く. また, 絶対座標と壁の座標系は静的であるのに対し，ロボットの座標系は動的である.



$\Sigma(x, y, O)$：絶対座標系
$\Sigma(x^r, y^r, O^r)$：ロボットの座標系
$\Sigma(x^w, y^w, O^w)$：壁の座標系
α：壁と絶対座標との角度差

図 5. 3　絶対座標，ロボット，壁の座標関係

5-3-1　位置候補の選択

位置候補点と位置候補を定義する．位置候補点と位置候補を図 5.4 に示す．位置候補点は点O^wを基準とした半径r_{origin}の円周上の点とし，広い空間上に広がるように設定する．また，位置候補は，位置候補点を中心とした半径r_{margin}の円形内部とする．位置候補点と位置候補を表した数式を示す．

$$\mathbb{C}_i = \{x, y | (x - x_i)^2 + (y - y_i)^2 = r_{margin}^2\} \tag{7}$$

$$x_i \triangleq x_o + r_{origin} \cos\theta_i \tag{8}$$

$$y_i \triangleq y_o + r_{origin} \sin\theta_i \tag{9}$$

ここで$\mathbb{C}_i$は位置候補を，(x_i, y_i)は位置候補点のxy座標を，(x_o, y_o)は点 O のxy座標を表す．

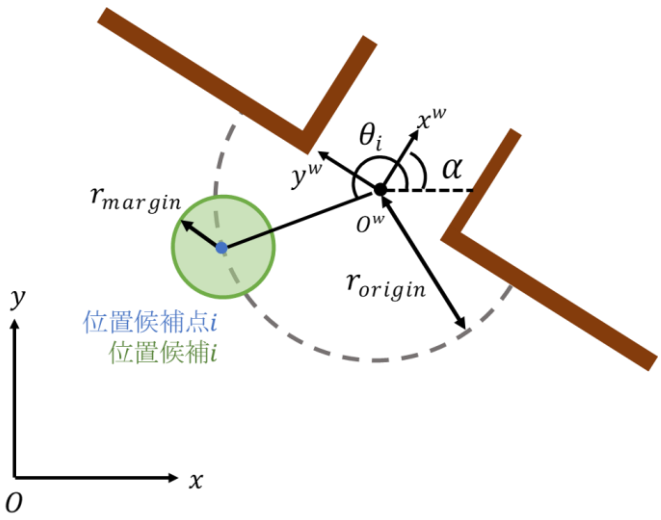


図 5.4　位置候補点と位置候補

第 3 章の障害物認識により，動的障害物の位置と速度がカルマンフィルタによって推定されている．この推定された位置と速度から，動的障害物のΔt秒毎の位置を求め軌道を予測する．動的障害物は等速モデルを仮定しているので軌道は

$$\mathbb{T} = \{\vec{s}(t) | \, k \in (0, 1, \dots, n)\} \tag{10}$$

$$\vec{s}(t) \triangleq \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} + \begin{bmatrix} \hat{v}_x \\ \hat{v}_y \end{bmatrix} k\Delta t \tag{11}$$

$$t \triangleq k\Delta t \tag{12}$$

$$n = \frac{T_{max}}{\Delta t} \tag{13}$$

集合$\mathbb{T}$は予測された動的障害物の軌道を表す．$\hat{x}, \hat{v}_x, \hat{y}, \hat{v}_y$はカルマンフィルタから得られた推定値であり，集合$\mathbb{T}$の要素$\vec{s}(t)$は動的障害物の$k\Delta t$秒後の位置を表す．また$T_{max}$は軌道の予測する最大の時間を決める既定パラメータである．

位置候補群の中から最も軌道に近い位置候補を選択するために，まず各位置候補点と軌道の中で最も近い距離をもとめる．位置候補点と軌道との距離は次式で与えれれる（位置候補点と軌道候補点との距離はDで表記する）．

$$D_{min,i} = \min_{t \in [0, T_{max}]} D_i(t) \tag{14}$$

$$D_k(t) \triangleq \|\vec{s}(t) - \vec{s}_i\| \tag{15}$$

$$\vec{s}_i \triangleq \begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{16}$$

ここで，$D_{min,i}$はi番目の位置候補点と軌跡と最も近い距離を，$D_i(t)$は時刻tにおける軌道上の位置と軌道候補iとの距離を表している．また位置候補に到達する時間を，動的障害物の軌跡上で，距離が最も近くなる時刻をt_eとして以下のように定義する．

$$t_e = \arg \min_{t \in [0, T_{max}]} D_i(t) \tag{17}$$

$D_{min,i}$を表したものを図 5.5 に示す．

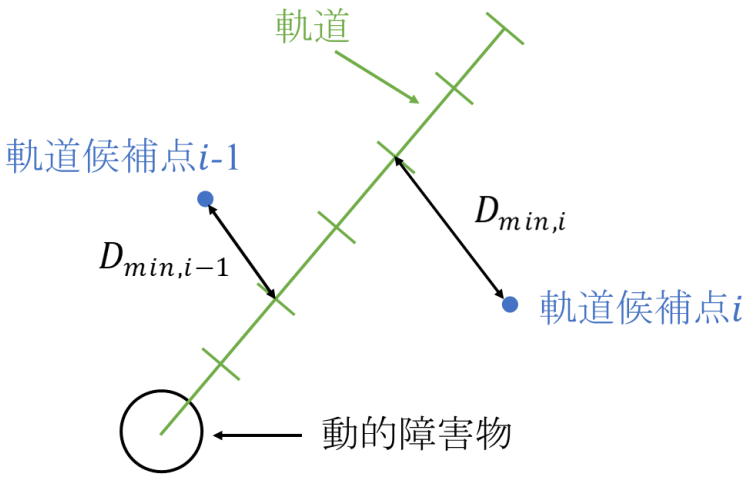


図 5. 5　軌道候補点と$D_{min,i}$

5-3-2　軌道の選択

ロボットに命令する速度ベクトルの速さと向きを定義する．ここでは，命令する速さを定義する．動的障害物とロボットが衝突しないために，動的障害物とロボットの距離が近いとき，ロボットは動的障害物と同じ速さで逃げるように移動する．逆に動的障害物とロボットの距離が遠いとき，動的障害物がどのような経路を選択するのか予測できないため，ゆっくり動くようにする．プロダクションルールより表現すると

$$if\ d \le d_1\ then\ v_u = v_o \tag{18}$$

$$if\ d_1 < d < d_2\ then\ v_u = a(d - d_1) + v_o \tag{19}$$

$$if\ d_2 < d\ then\ v_u = v_{min} \tag{20}$$

$$d_1 \triangleq R_{robot} + R_{obstacle} \tag{21}$$

$$v_o \triangleq \sqrt{\hat{v}_x^2 + \hat{v}_y^2} \tag{22}$$

$$a \triangleq \frac{v_{min} - v_o}{d_2 - d_1} \tag{23}$$

ただし，dはロボットと動的障害物の距離を，v_uは命令する速さを，v_oは動的障害物の速さを表している．R_{robot}，$R_{obstacle}$はロボットの半径，動的障害物の半径を示す．R_{robot}，$R_{obstacle}$からなるd_1は衝突距離を表しており，d_1よりdが小さくなる場合，衝突する．またd_2はロボットが動き始める間隔を意味する規定パラメータである．またv_{min}は制御入力可能なロボットの最低速度である．速さと距離の関係を図 5.6 に示す．

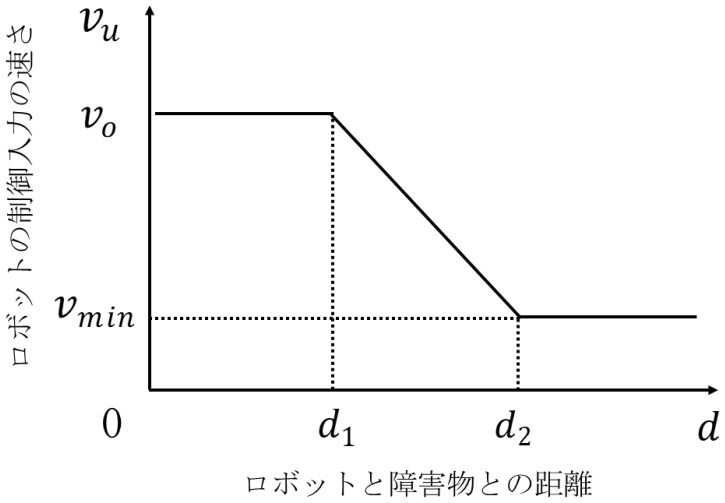


図 5.6　距離dと速さv_uの関係性

続いて命令速度の方向を定義する．壁の座標系であるy^w軸方向が，動的障害物が狭い空間から広い空間に移動する際に左右どちらに移動するのか判別する方向となる．そこで，動的障害物の速度ベクトルを絶対座標から壁の座標系へ座標変換し，y^w軸方向の速度ベクトルv_y^wを求める．

$$v_y^w = -v_x \sin\alpha + v_y \cos\alpha \tag{24}$$

速度v_y^wは壁の座標からの見たy^w軸方向の動的障害物の速度ベクトルである．軌道候補は，右斜め後ろと左斜め後ろの 2 つとした．動的障害物が左に移動すると，ロボットは右斜め後ろへと，動的障害物とは反対方向にロボットが移動するよう制御入力を与える．また動的障害物が直進するとき，動的障害物の位置によって軌道の向きを選択する．軌道の選択をプロダクションルールにより次のように定義される．

$$if\ v_y^w > 0\ then\ \left(v_x^r, v_y^r, \omega^r\right) = (-v_u, -v_u, 0) \tag{25}$$

$$if\ v_y^w < 0\ then\ \left(v_x^r, v_y^r, \omega^r\right) = (-v_u, v_u, 0) \tag{26}$$

$$if\ v_y^w = 0, y_y^w > 0\ then\ \left(v_x^r, v_y^r, \omega^r\right) = (-v_u, -v_u, 0) \tag{27}$$

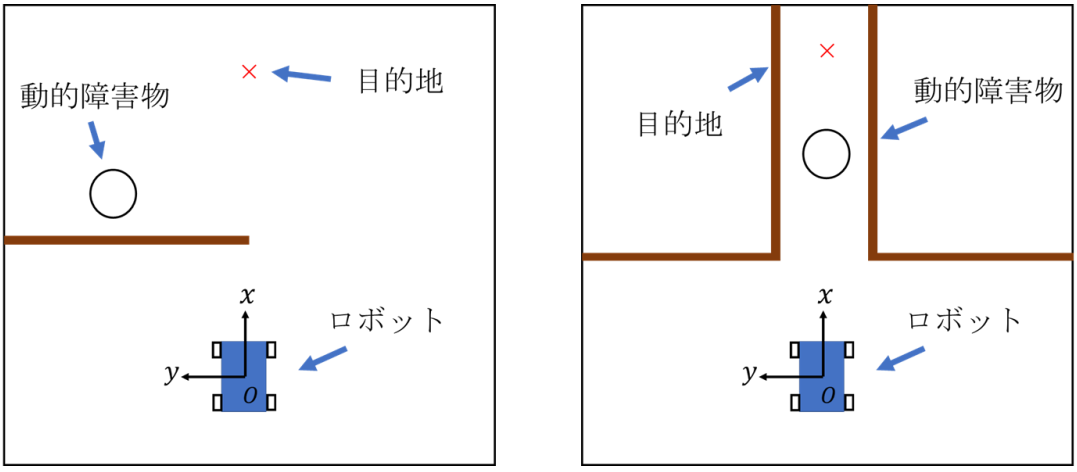
$$if\ v_y^w = 0, y_y^w > 0\ then\ \left(v_x^r, v_y^r, \omega^r\right) = (-v_u, v_u, 0) \tag{28}$$

ここで，$\left(v_x^r, v_y^r, \omega^r\right)$は制御入力を表す．また$v_y^w$は壁の座標系から観測した動的障害物の位置である．

5-4　シミュレーション

5-4-1　衝突回避実験の環境

生活環境は，リビングなどの広い空間と，廊下などの狭い空間がある．また壁などにより，相手の状況が推定できない死角領域がある．提案手法を評価するために，2 つの生活環境を用意した．（Ⅰ）壁や扉により，ロボットと動的障害物が初期状態において目視できない死角領域のある環境，（Ⅱ）廊下とリビングの交差点付近のなど，狭い空間と広い空間の狭間における環境．（Ⅱ）は既に，節 5.2 で述べた環境 A である．シミュレーション環境Ⅰ，Ⅱを図 5.7 に示す．



(a)環境Ⅰ　　(b)環境Ⅱ

図 5.7　衝突回避実験の環境

5-4-2　衝突回避実験の目的

生活環境下において自律走行の難しさは，ロボットと動的障害物の位置関係によって変化する．環境Ⅰにおいて，動的障害物がロボットはお互い死角になっている．そのため，交差点においてロボットと動的障害物が鉢合わせたとき，ロボットは動的障害物の移動する方向を阻害する動きは衝突する可能性があるため危険である．また環境Ⅱにおいては，節 5.2 でも述べた通り，ロボットは目的に動的障害物が邪魔で移動できず，逆に動的障害物はロボットが出口を塞いでいるため身動きが取れない．　生活環境Ⅰにおいて，従来の DWA アルゴリズムと，提案手法Ⅰである動的障害物を予測した DWA アルゴリズムをシミュレーション上で比較実験する．結果を比較し提案手法である動的障害物を予測 DWA アルゴリズムの有効性を示す．また生活環境Ⅱにおいて，提案手法Ⅱの動線確保動作計画によってロボットが動的障害物の道を譲ることを示す．

5-4-3　衝突回避実験 I の条件

死角領域のある環境 I での衝突回避実験を行う．ロボットと動的障害物の軌跡が交わるように，動的障害物の速度と位置を決める．ロボットの最大速度は 0.23[m/s]，動的障害物は等速で 0.5[m/s]とした．ロボットの初期位置が絶対座標の原点とし，動的障害物は$(x, y) = (2.5[m], 4.0[m])$とした．予測時間は，3.2[s]，1.6[s]とした．予測時間を 2 つ用意すると，1 つの動的障害物に対して 2 個，円形障害物が予測される．予測する障害物を増やすことで，ロボットは動的障害物の進路方向に進みにくくなる．

5-4-4　衝突回避実験 I の結果

従来の DWA アルゴリズムと，動的障害物を予測した DWA アルゴリズムをシミュレーションした．動的障害物とロボットが近づいたときの様子を示す（図 5.8，図 5.9）．従来の DWA アルゴリズムでは，動的障害物の進行方向へロボットが進んでしまい，衝突した．一方で提案手法は，動的障害物の進行方向にロボットが進まず，動的障害物が完全に通り過ぎてからロボットが移動した．また動的障害物を予測したときの様子を示す（図 5.10）．青色の円柱は動的障害物を，ピンク色の円形は予測された円形障害物を表している．また，緑色で囲まれた四角形はロボットを，その中心から伸びている緑色の曲線は軌道を表している．動的障害物も考慮した DWA アルゴリズムは，予測された障害物も避けるように，局所的経路計画が行われた．

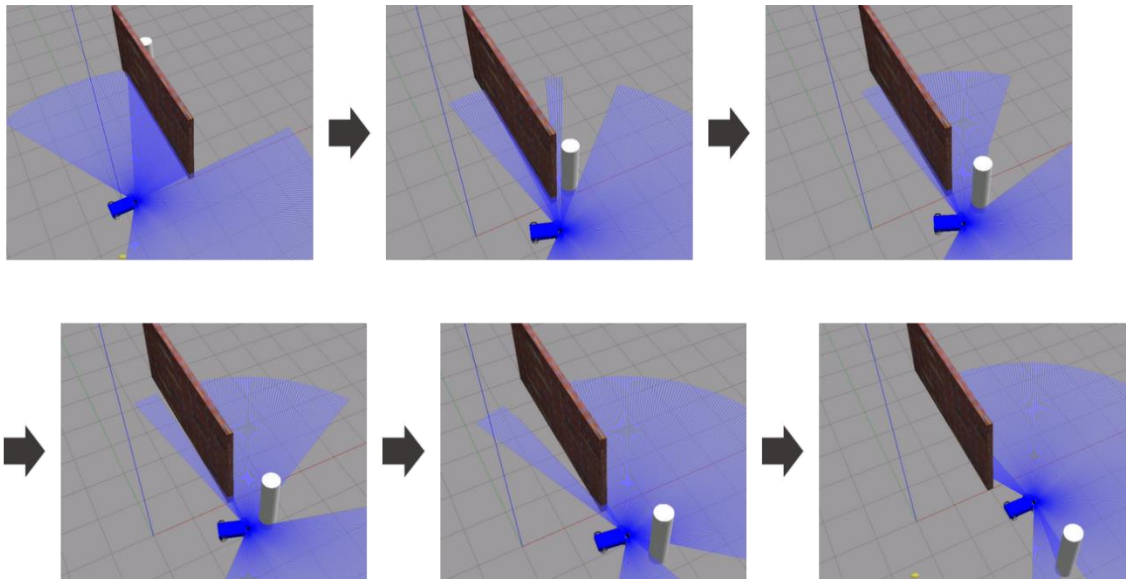


図 5. 8 ロボットが動的障害物を避ける様子（Prediction Obstacles Layer 有り）

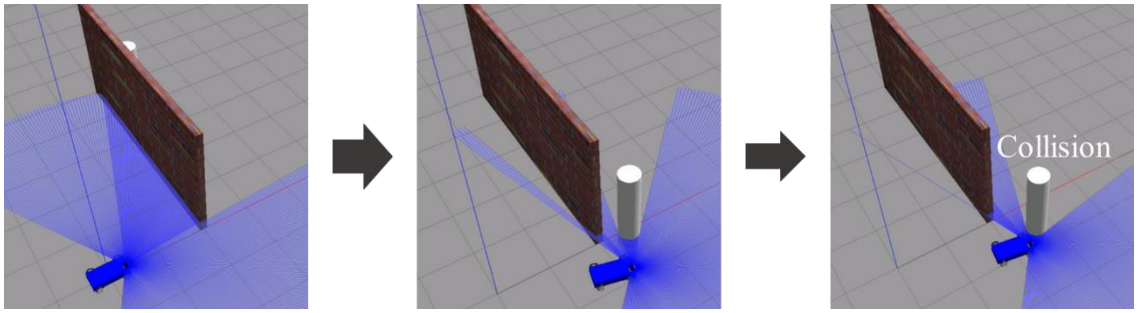


図 5. 9 ロボットが動的障害物と衝突する様子（Prediction Obstacles Layer 無し）

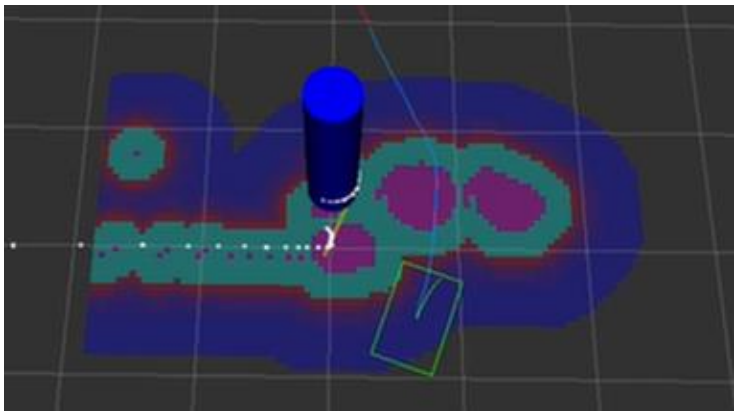
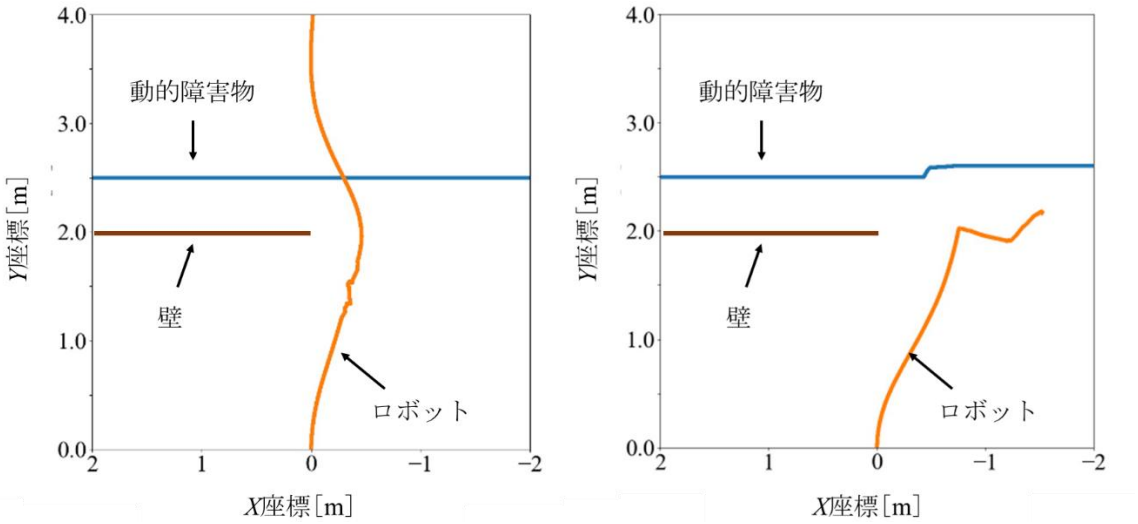


図 5.10　障害物予測の様子（Rviz）

動的障害物とロボットの経路結果を図 5.11 に示す．経路は動的障害物とロボットの中心位置をプロットしたものである．図 5.11.b において，動的障害物とロボットの軌跡は交わっていないが，動的障害物とロボットの大きさがあるため，実際には衝突している．実際，従来の DWA アルゴリズムでは，動的障害物とロボットが衝突し，互いの軌跡が歪んでいることわかる．ロボットは動的障害物に衝突し，そのまま左にずれてしまっている．一方で，提案手法は衝突が起こらなかったため，目的地まで綺麗な曲線を描きながらロボットが進んでいることがわかる（図 5.11.a）．



(a) Prediction Obstacles Layer 有り　　(b) Prediction Obstacles Layer 無し

図 5. 11　ロボットと動的障害物の軌跡

5-4-5　衝突回避実験Ⅱの条件

初期状態でロボットが出口を塞いでいる環境Ⅱにおいて，ロボットに目的地を与え，問題なく自律走行可能かシミュレーションを行う．配置を絶対座標で表現する．ロボットの初期位置は(1.0[m], 0.0[m])，目的地を(4.0[m], 0.0[m])とする．動的障害物の軌道を 3 つ用意する．(v_x^o, v_y^o, ω^o)は動的障害物の制御入力を表す．

（パターンＡ）

動的障害物の初期位置は（3.0[m], 0.0[m]）とする．初期位置から点 a（2.0[m], 0.0）まで$(v_x^o, v_y^o, \omega^o) = (-5.0[m/s], 0.0[m/s], 0.0[rad/s])$で等速直線運動を続ける．点 a からは$(v_x^o, v_y^o, \omega^o) = (-5.0[m/s], 0.0[m/s], -0.1[rad/s])$で等速円運動を続ける．

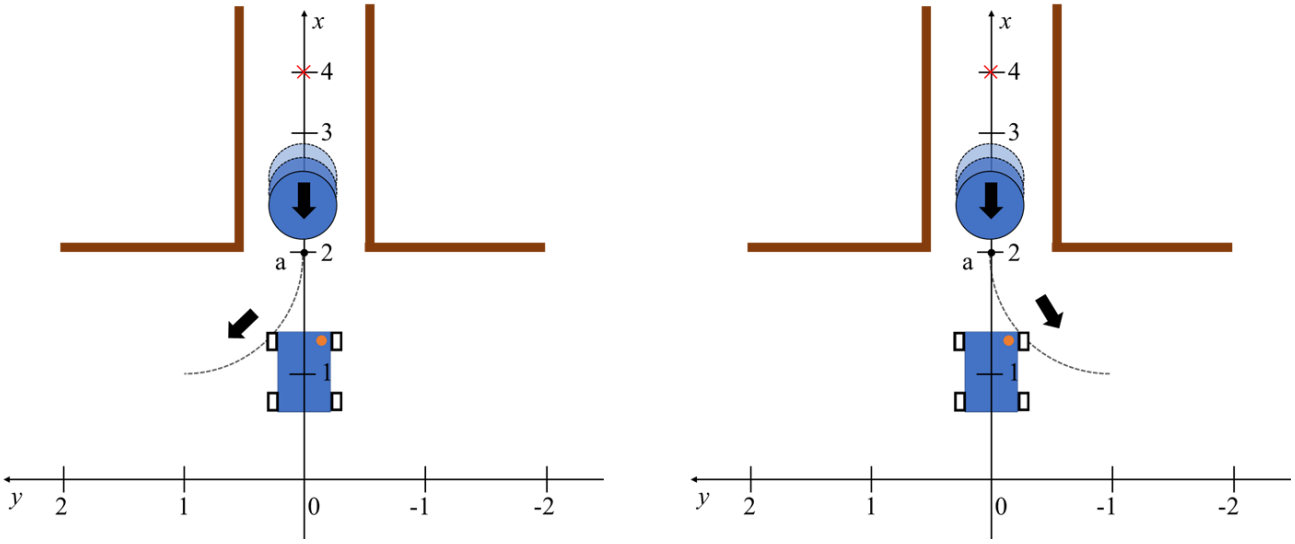
（パターンＢ）

動的障害物の初期位置は（3.0[m], 0.0[m]）とする．初期位置から点 a（2.0[m], 0.0）まで$(v_x^o, v_y^o, \omega^o) = (-5.0[m/s], 0.0[m/s], 0.0[rad/s])$で等速直線運動を続ける．点 a からは$(v_x^o, v_y^o, \omega^o) = (-5.0[m/s], 0.0[m/s], -0.5[rad/s])$で等速円運動を続ける．

（パターンＣ）

動的障害物の初期位置は（3.0[m], 0.0[m]）とする．初期位置から点 a（2.0[m], 0.0）まで$(v_x^o, v_y^o, \omega^o) = (-5.0[m/s], 0.0[m/s], 0.0[rad/s])$で等速直線運動を続ける．点 a からは$(v_x^o, v_y^o, \omega^o) = (-5.0[m/s], 0.0[m/s], 0.5[rad/s])$で等速円運動を続ける．

パターンA，B，C違いは，等速円運動の曲率である．図 5.12 にパターンB，Cの軌跡を示す．



(a)パターン B　　(b)パターン C

図 5. 12　動的障害物の軌跡

動線確保動作計画のパラメータを設定する．

位置候補の大きさを表す半径r_{margin}は 0.4[m]とし，位置候補点の基準となる半径r_{origin}は 1.5[m]とする．位置候補点を定めるためθ_iは以下のようにする．

$$(\theta_0, \theta_1, \theta_2, \theta_3, \theta_4) = \left(\frac{5}{4}\pi[rad], \frac{11}{8}\pi[rad], \pi[rad], \frac{13}{8}\pi[rad], \frac{7}{4}\pi[rad]\right)$$

ロボットの命令速さを調整するd_{min}は 2[m]とする．

全方位方向移動型ロボットの幅は 0.24[m]，長さは 0.37[m]であるので，ロボット半径を示すR_{robot}は 0.441[m]とした（図 5. 13）．

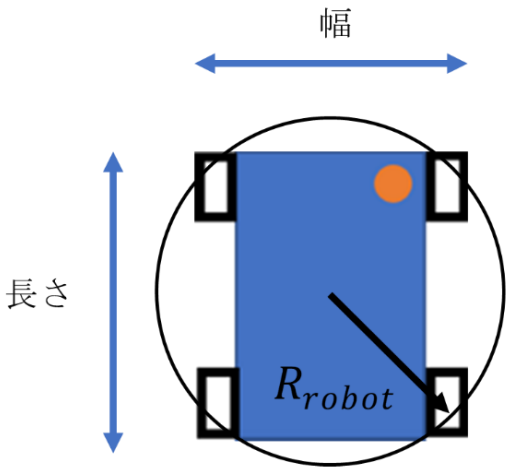


図 5. 13　ロボットの半径

5-4-6　衝突回避実験Ⅱの結果

それぞれの軌道のパターンに対して，3 回ずつ衝突回避実験を行った．

ロボットの中心座標と動的障害物の中心座標との距離dを衝突の指標とする．衝突距離をロボットの半径と動的障害物の半径を足した距離とし，最接近距離から衝突距離を引いた距離を衝突余裕とする．今回，全方位方向移動型ロボットの半径は 0.441[m]，動的障害物の半径は 0.2[m]であるので，衝突距離は 0.641[m]となる．動的障害物とロボットの距離が，衝突距離より下回った場合，動的障害物とロボットはおそらく衝突する．だが，ロボットの向きによっては衝突しない可能があるため，おそらく衝突と記述する．

（パターン A）

パターン A において，3 回ともロボットと動的障害物は衝突しなかった．3 回の衝突回避実験の結果を表 5.1 に示す．

表 5. 1　衝突回避実験結果Ⅱ（パターン A）

	最接近時刻 [s]	最接近距離 [m]	衝突余裕 [m]
1 回目	6.07	1.039	0.398
2 回目	5.82	1.075	0.434
3 回目	6.45	1.048	0.407

パターン A の軌道において，衝突回避実験は 3 回とも衝突余裕は約 0.4[m]程あり，動的障害物から十分離れて，動的障害物の経路を確保し，ロボットが安全に自律走行可能であることがわかる．

3 回の中で最も衝突余裕が小さかった 1 回目の衝突回避実験を図に示す．距離dと時刻を図 5.14 に，ロボットと動的障害物の経路結果を図 5.15 に，軌道選択の基準となるv_y^wを図 5.16 に示す．（図 5.15 において，時刻 4[s]と 7[s]における動的障害物とロボットの位置点を黒色の円で囲った）

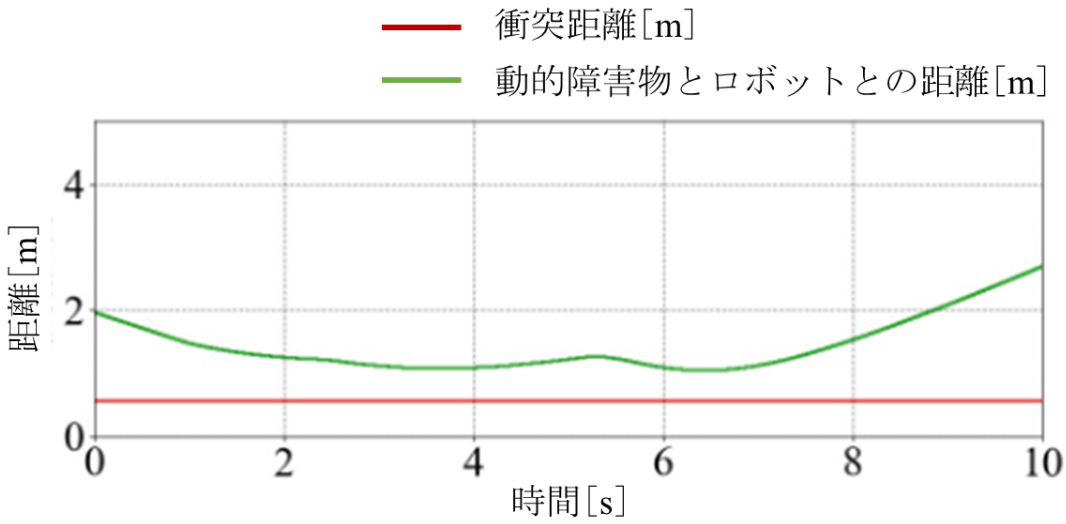


図 5. 14　ロボットと障害物との距離（パターン A）

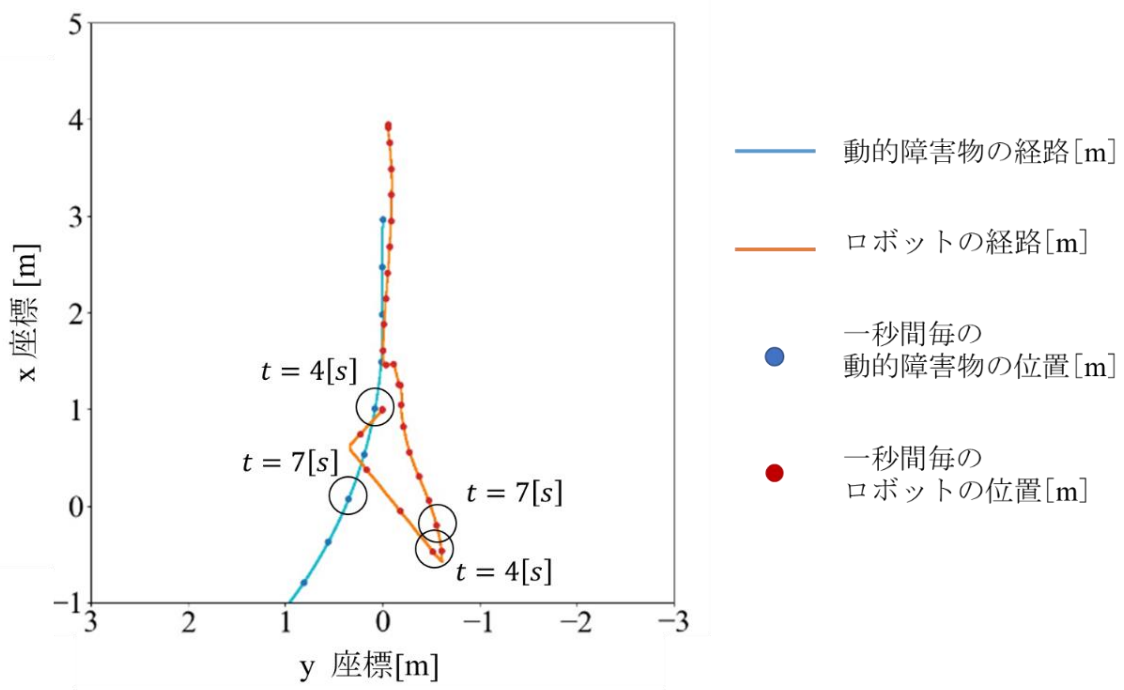


図 5. 15　ロボットと障害物の経路結果（パターン A）

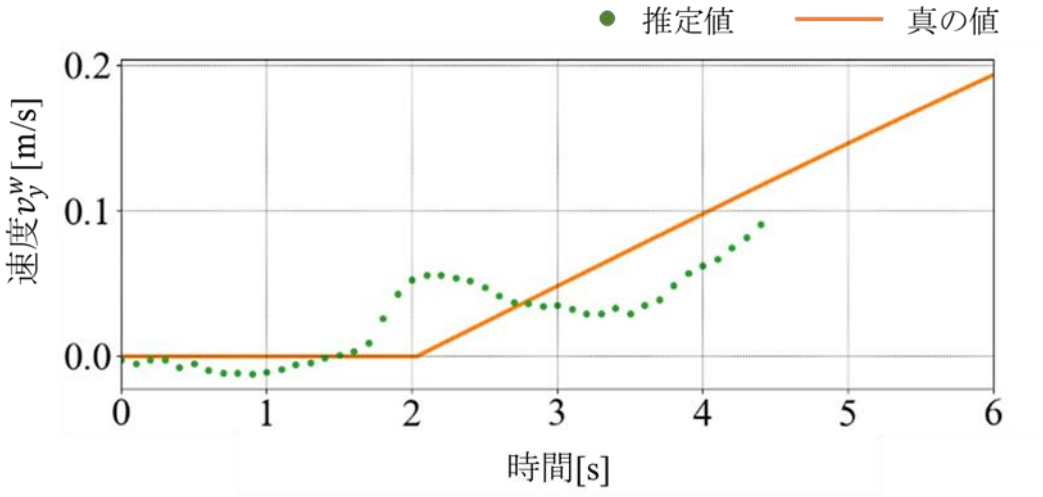


図 5. 16　速度v_y^w（パターン A）

0～1.5 秒間にかけてロボットは左下に移動しているが，1.5～4.5 秒間にかけては，ロボットは右下に移動し動的障害物を避けるように動いている．こののようにロボットが移動したのは，v_y^wの推定値が 0～1.5 秒間にかけて負の値に，1.5～4.5 秒間には正の値になったためである．動的障害物は 0～2.4 秒間まで，直線運動をしており，v_y^wの真の値は 0.0[m/s]である．このとき，ロボットに求められるのは真後ろに動く行為である．v_y^wの真の値が 0.0[m/s]であるとき，推定値は正負を行き来するため，結果的にロボットは真後ろに動く．図 5.16 において，2 秒付近で推定値は真の値に収束していないことがわかるが，v_y^wは正の値であり，動的障害物が移動する方向の推定ができ，逆の方向にロボットが移動することで衝突を避けている．また，図 5.16 において，推定値が 4.5 秒で途切れているのは，動線確保動作計画が終了したことを意味する．図 5.15 の経路結果からわかるように，動線確保動作計画の終了後，ロボットは目的地までの経路が見つけ目的地まで移動できている．

（パターン B）

パターン B において，3 回ともロボットと動的障害物は衝突しなかった．3 回の衝突回避実験の結果を表 5.2 に示す．

表 5. 2　衝突回避実験結果Ⅱ（パターン B）

	最接近時刻 [s]	最接近距離 [m]	衝突余裕 [m]
1 回目	3.25	1.131	0.49
2 回目	3.08	1.113	0.472
3 回目	3.75	1.051	0.41

パターン B の軌道において，衝突回避実験は 3 回とも衝突余裕は約 0.4[m]以上あり，動的障害物から十分離れて，動的障害物の経路を確保し，ロボットが安全に自律走行可能であることがわかる．

3 回の中で最も衝突余裕が小さかった 3 回目の衝突回避実験を図に示す．距離 d と時刻を図 5.17 に，ロボットと動的障害物の経路結果を図 5.18 に，軌道選択の基準となる v_y^w を図 5.19 に示す．

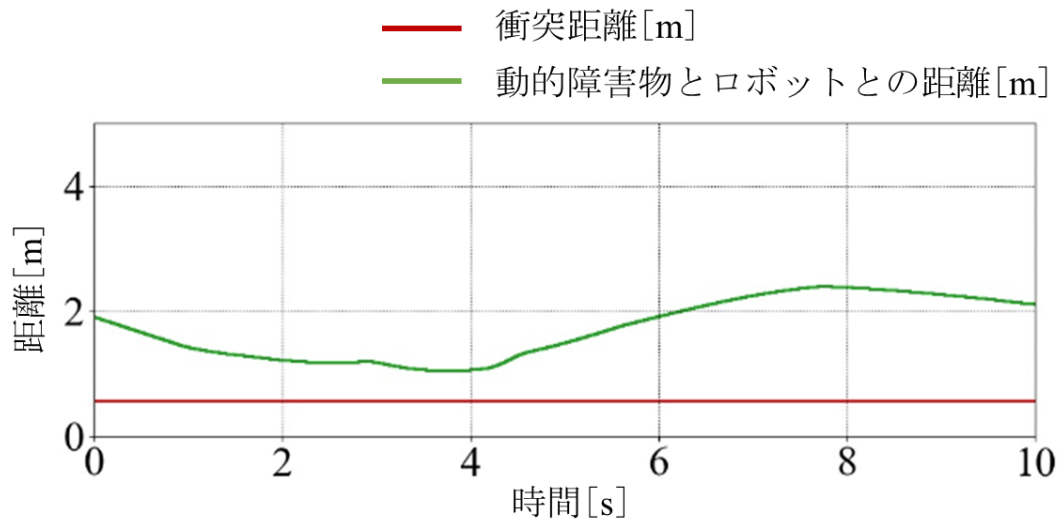


図 5. 17　ロボットと障害物との距離（パターン B）

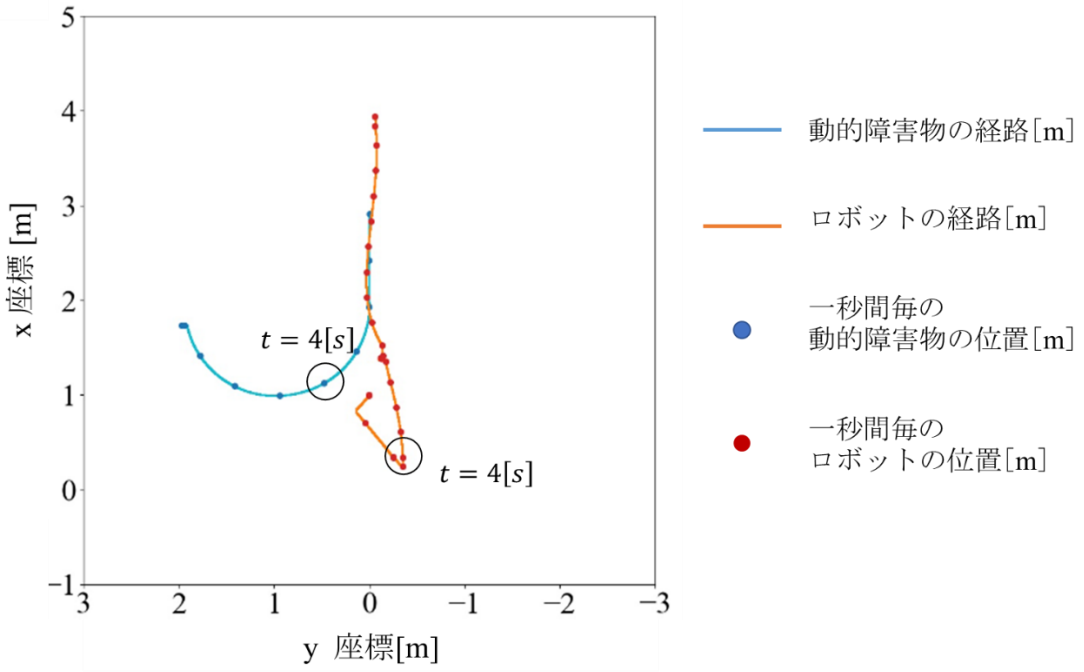


図 5. 18　ロボットと動的障害物の経路結果（パターン B）

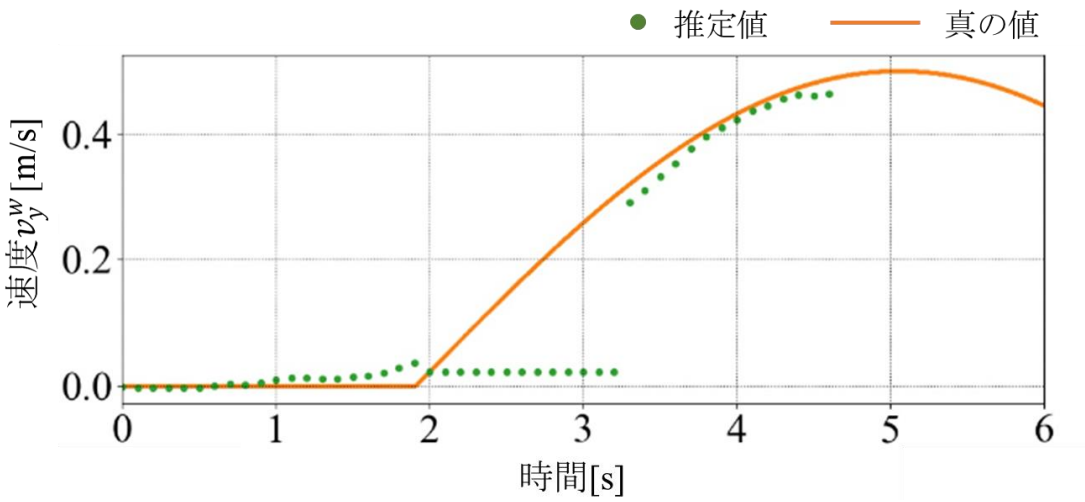


図 5. 19　速度v_y^w（パターン B）

表 5.2 から，3 回目の障害物回避実験は，2 回目と 3 回目に比べて，衝突余裕が 0.7[m]程小さい．これは，速度v_y^wの推定値が正確に計算できないことが原因に挙げられる．2〜3 秒間にかけて，動的障害物の真の値は正の値に加速しているが，推定値は 0[m/s]のままになっている（図 5.19）．環境Ⅱにおいて，動的障害物は 1 つであるため，障害物認識は容易であると想定される．しかし，実験には，壁や壁際などを障害物抽出アルゴリズムが円形障害物で無理やり近似させようとし，環境には複数の動的障害物があるかのように誤認識が起こる（図 5.20）．そのため，別の誤認識された動的障害物と，目的の動的障害物が混ざってしまい，動的障害物を一時的に追従できなかったと考えられる．

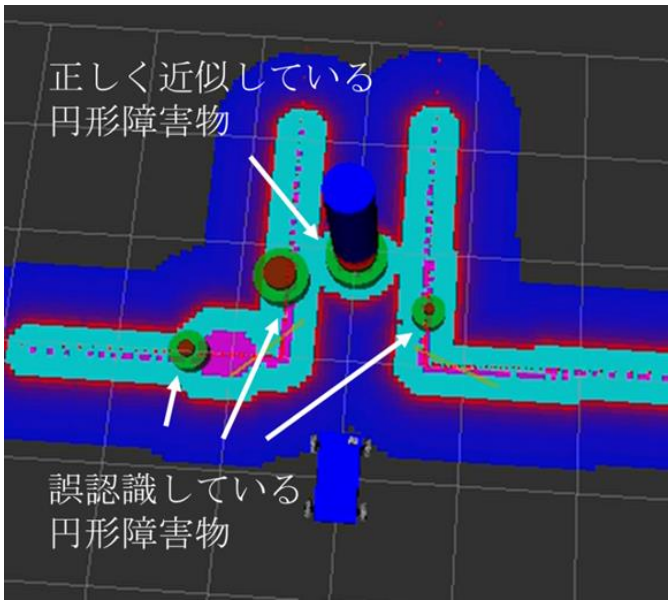


図 5. 20　誤認識した円形障害物

（パターン C）

パターン C において，3 回ともロボットと動的障害物は衝突しなかった．3 回の衝突回避実験の結果を表 5. 3 に示す．

表 5. 3　衝突回避実験結果Ⅱ（パターン C）

	最接近時刻 [s]	最接近距離 [m]	衝突余裕 [m]
1 回目	3.30	1.160	0.519
2 回目	2.76	1.172	0.531
3 回目	3.12	1.470	0.829

パターン B の軌道において，衝突回避実験は 3 回とも衝突余裕は約 0.5[m]以上あり，動的障害物から十分離れて，動的障害物の経路を確保し，ロボットが安全に自律走行可能であることがわかる．

1 回目の衝突回避実験を図に示す．距離dと時刻を図 5. 20 に，ロボットと動的障害物の経路結果を図 5.21 に，軌道選択の基準となるv_y^wを図 5.22 に示す．

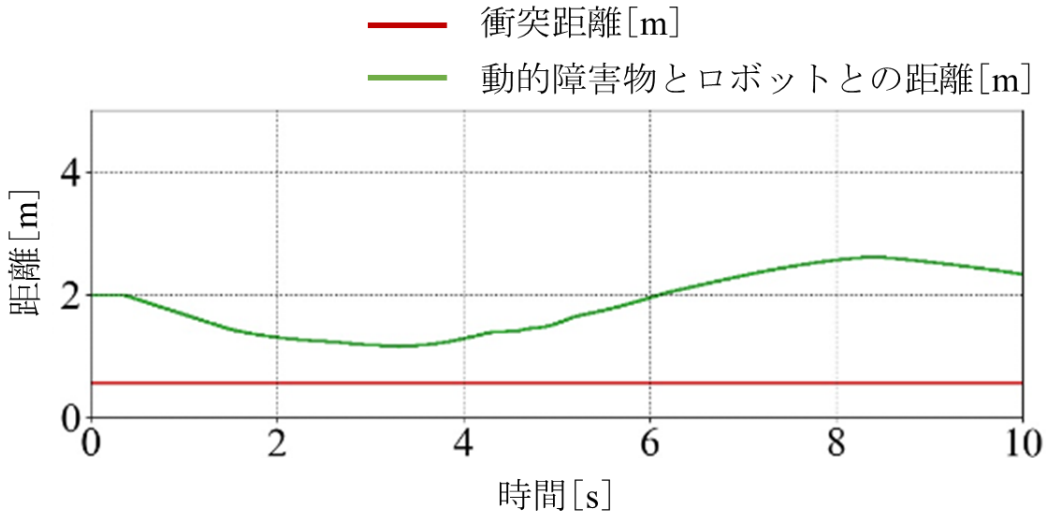


図 5. 21　ロボットと障害物との距離（パターン C）

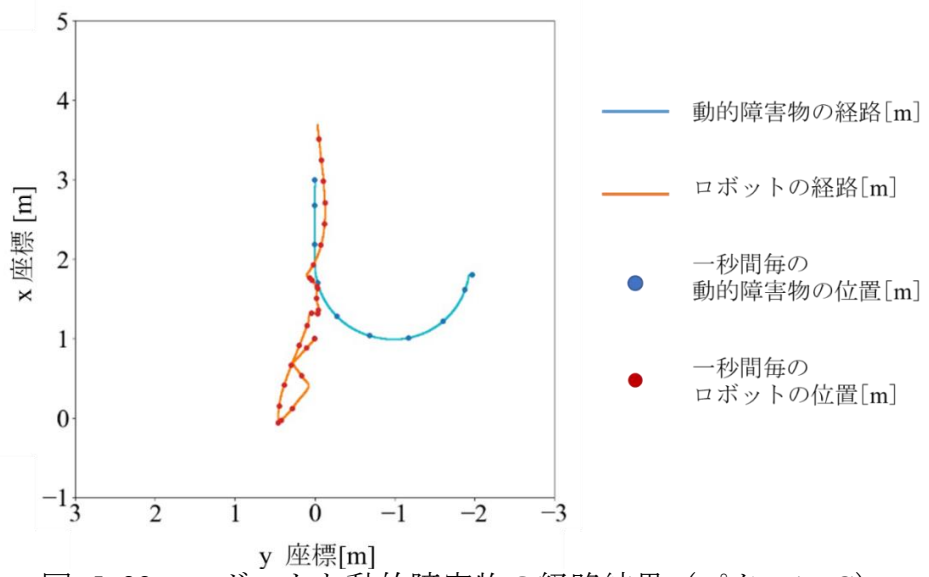


図 5. 22　ロボットと動的障害物の経路結果（パターン C）

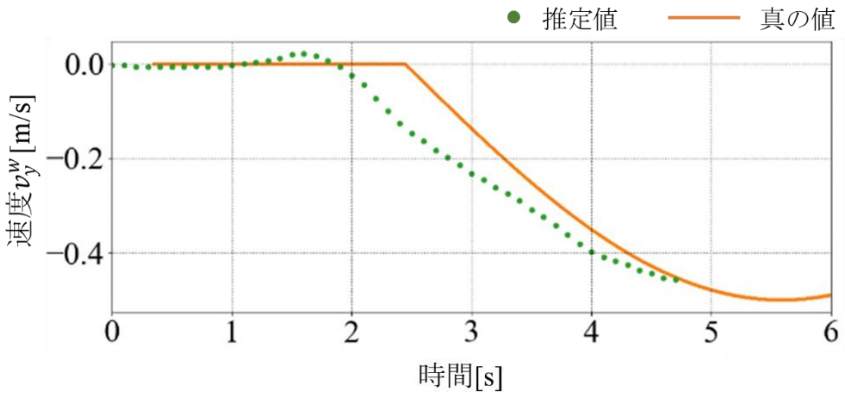


図 5. 23　速度v_y^w（パターン B）

図 5.23 を見ると，動的障害物のv_y^wの推定値は，2～4 秒間にかけて，真の値に収束していない．これは，第 4 章でも述べた通り，動的障害物の予測を等速モデルと仮定しているため，動的障害物の急な速度変化に対応することができない．しかし，1 回目の衝突回避実験は，ロボットは動的障害物に道を譲り，衝突することなく 0.519[m]以上の衝突余裕をもって動線確保動作計画を実行できている動的障害物の速度を正確に精度よく推定するのことは，環境Ⅱにおいて動的障害物がどのような運動をするのか事前に把握できないため困難である．動線確保動作計画は，動的障害物の速度を基準とするのではなく，壁に対して平行なy^w軸方向の速度v_y^wの正負を基準にしたため，精度よく速度を求めることを必要としない．故に，速度推定の値が収束していなくとも，衝突回避をしながら後退することができる．

5-5　実機実験

5-5-1　実験環境

環境 II を再現した実験環境を用意した．図 5.24 にロボットと人の位置関係及び，ロボットの目的地，人の移動経路を示す．壁は段ボールで代用し，安全配慮のため，ビート板を人の前に持って移動してもらった．

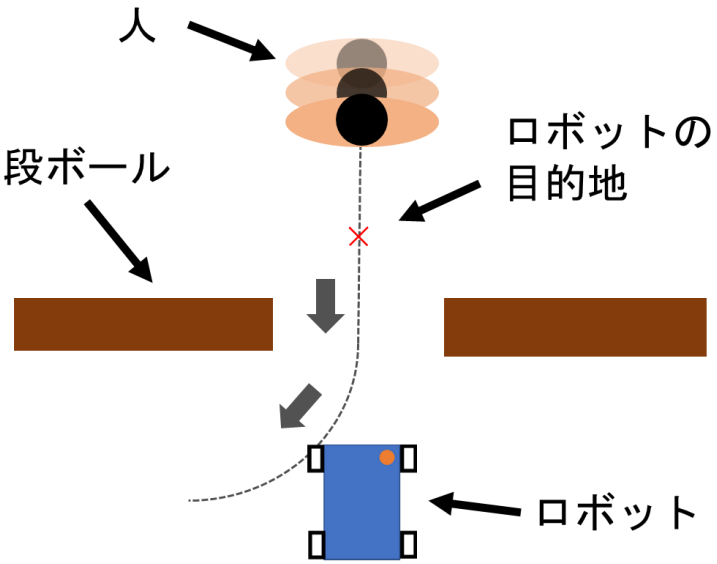


図 5.24　人の移動経路及びロボットの目的地

5-5-2　実験結果

実機実験結果を図 5.25 に示す．初期状態において，ロボットは人が進む道を阻害している．ロボットから見て，人は中央から左斜め後ろ方向に移動している．ロボットは，そのとき右斜め下に移動していることがわかる．シミュレーション環境と同様，動線確保動作計画によってロボットは人に道を譲った．また動線確保動作計画の終了後，ロボットは目的地に向かって走行した．

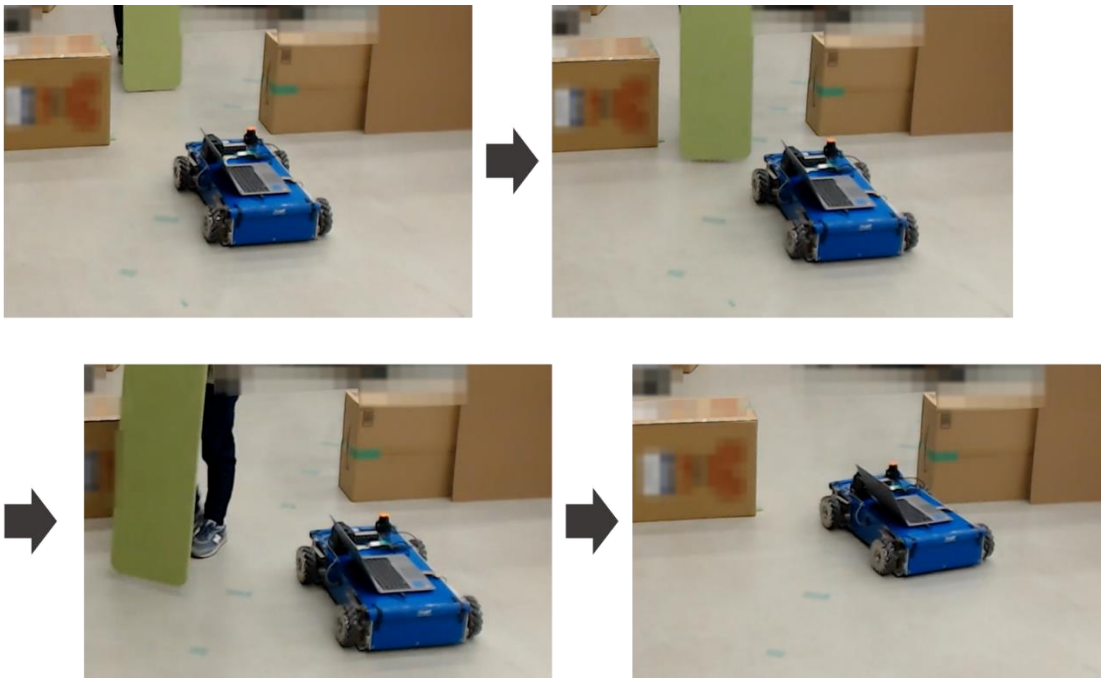


図 5.25　衝突回避実験結果

第 6 章　結言

6-1　まとめ

本研究では，人とロボットが混在する生活環境下において，ロボットが人に危害を与える場合，死角領域からロボットが突然現れ人と衝突する問題，ロボットが人の通り道を塞ぐ問題に対して，それぞれの回避動作計画を提案した．

自律走行の構成は，大局的経路計画，局所経路計画，リカバリー動作，障害物認識に 4 つに分けた．大局的経路計画は目的地までの経路を計画し，局所経路計画は時々刻々のロボットに命令する制御入力を定める．大局的経路計画の失敗や局所的経路計画の制御入力が得られないとき，ロボットは停止する．その場合，コストマップをクリアや，その場回転などを行い，有効な経路を探索するためにリカバリー動作が用意されている．提案手法には，動的障害物の位置と速度を要求する．そこで，ロボットに搭載した測域センサーから送信した Laser Scan より障害物認識にして，動的障害物の位置と速度を推定した．

代表的な局所経路計画として DWA アルゴリズムがあるが，静的障害者を避けるように設計されているため，動的障害物には衝突する場合がある．そのため，純粋な DWA アルゴリズムでは死角領域から突然現れた動的障害物とロボットが衝突する．提案手法は，動的障害物の数秒後の位置を予測し，現在の障害物だけでなく予測された障害物にも衝突しない軌道を選択することで，動的障害物をも回避動作を行える．

廊下などの狭い空間に動的障害物が存在した場合，ロボットは大局的経路計画によって経路を見つけ出せず停止する．また動的障害物にとっても，ロボットが道を塞ぎ身動きが取れない．そこで，リカバリー動作としてロボットが経路を譲る動線確保動作計画を提案した．動線確保動作計画は，動的障害物が動線を確保するまで，後退し続けるアルゴリズムである．動的障害物が左前に移動しようとした場合，ロボットは右斜め下に移動するなどし，動的障害物の進行経路と反対側に移動するようにした．その際，人が左右どちらに移動するか判断する必要がある．障害物認識から得られた動的障害物の速度ベクトルにより，壁に対して平行方向の速度ベクトルを求め，それを指標にした．

障害物認識は，動的障害物の位置と速度を推定する．測域センサーから Laser Scan を受け取り，円形障害物に抽出する．抽出された円形障害物は複数存在し，個別にカルマンフィルタにかけることで動的障害物の位置と速度を推定した．またカルマンフィルタの予測モデルは等速モデルを仮定した．

第 4 章では，四角形と円形の軌道を描く円形の動的障害物をシミュレーション環境上に用意し，障害物認識実験を行った．カルマンフィルタの予測は等速モ

デルを仮定しているため，得られた速度推定値は，四角形の角や円形では，真の値と誤差があった．またその速度推定値から予測された数秒の位置は，予測時間が長い程，誤差が大きくなった．しかし，四角形を描く動的障害物の推定を続けると，次第に誤差が小さくなった．

第 5 章では，死角領域から動的障害物が突然現れる環境 I と，ロボットが動的障害物の経路を塞いでいる環境Ⅱ，それぞれシミュレーション環境を用意し衝突回避実験を行った．環境 I における衝突回避実験は，従来の DWA アルゴリズムと提案手法である動的障害物を予測した DWA アルゴリズムを比較し有効を示した. 従来の DWA アルゴリズムの場合, ロボットは動的障害物に衝突した. 一方で提案手法である動的障害物を予測した DWA アルゴリズムは, 動的障害物の進行方向先にロボットが進まず, 動的障害物が通り過ぎた後に，ロボットが移動したことで, 衝突せずに自律走行した. 環境Ⅱにおける衝突回避実験は， 3 種類の曲線を描く動的障害物を用意し，ロボットに自律走行を行わせた．1 つの軌道パターンに対し各 3 回ずつ衝突回避実験を行った．9 回とも全て，ロボットと動的障害物は衝突することなく自律走行し，動線確保動作計画の有効性が示せた. 動線確保動作計画において, 動的障害物の壁に対して平行方向の速度ベクトルを軌道選択に用いた．カルマンフィルタによる速度推定は等速モデルを仮定しているため, 曲線を描く動的障害物の速度を正確に精度よく求められない．しかし, 動的障害物の進行方向のみが重要となるため, 壁に対して平行方向の速度ベクトルの正負のみ使用した. 故に精度よく速度ベクトルを求めなくても, 動的障害物に経路を譲るように動作計画した.

6-2　今後の課題

第 5 章の環境Ⅱにおいて，動的障害物の速度推定が明らかに行えていない場合があった．これは，複数ある障害物の中から，同じの障害物を選び続ける追従がうまくいかなかったためである．また対象の動的障害物だけでなく, 壁なども円形障害物に強引に近似することがあった. そのため, 壁と壁の隙間をロボットがすり抜けようとする際に誤認識した円形障害物がロボットに邪魔する場合があった．測域センサーの Laser Scan は障害物の位置と形状は正確に精度よく把握することが可能であるが，Laser Scan から近似した線分障害物や円形障害物は形状と位置情報しか情報を持たないため，どの線分障害物が壁を表し，どの円形障害物が人を表しているのか判別できない．また速度で人を判別しようとしても, 誤認識を起こす場合があった. この障害物誤認識の解決策として，Laser Scan から抽出された円形障害物と線分障害物を，Web カメラ等の画像認識によりデータの紐づけを行う手法が挙げられる．そのような障害物認識の手法を使用し

Huixu Don 等は効率的な動的障害物の回避が報告されている．[17]

本研究で提案した動線確保動作計画は人の経路を確保するために後退している．そのとき，障害物が存在するのか確認せずに移動しているため，人や物がロボットの後ろ方向に存在したとき，衝突する．今後の展開として，動線確保動作計画の軌道選択のアルゴリズムにおいて，後ろ方向にある障害物に衝突しないために，軌道を選択するように追加する必要がある．また本研究で使用した測域センサーの走査範囲は 270°と真後ろの障害物を確認することができない．そのため，測域センサーをもう一つ全方位方向移動型ロボットに搭載するか，360°走査可能なセンサーを使用する必要がある．

また，今回提示したロボットが動的障害物の経路を塞いでいる環境Ⅱは，ロボットが道を譲らないと，お互いに身動きのとれない環境であった．しかし，生活環境下においては，ロボットと人が譲り合う行為が求められる．譲り合いの認識は，長期的な予測であるため難しい．伊部らは，譲り合いを確率で表現している．[18]

参考文献

[1]. 内閣府「令和 4 年版高齢社会白書」
https://www8.cao.go.jp/kourei/whitepaper/w-2022/zenbun/04pdf_index.html

[2]. 厚生労働省「介護保険事業状況報告」
https://www.mhlw.go.jp/topics/kaigo/osirase/jigyo/20/dl/r02_point.pdf

[3]. 厚生労働相「第 8 期介護保険事業計画に基づく介護職員の必要数について」
https://www.mhlw.go.jp/content/12004000/000804129.pdf

[4]. トヨタ自動車株式会社，HSR
https://global.toyota/jp/detail/8709536

[5]. アイロボット株式会社，ルンバ
https://www.irobot-jp.com/specialized_robot/

[6]. 独立行政法人産業技術総合研究所，パロ
https://www.aist.go.jp/aist_j/press_release/pr2004/pr20040917_2/pr20040917_2.html

[7]. Dijkstra, E.W., A note on two problems in connexion with graphs, In Numerische Mathematik, Vo.1 (1959), pp. 269–271.

[8]. Korf, R.E., Real-time heuristic search, Artificial Intelligence, Vol. 42, No. 2-3 (1990), pp. 189–211.

[9]. Khatib, O., Real-time obstacle avoidance for manipulators and mobile robots, The International Journal of Robotics Research, Vol. 5, No. 1 (1986), pp. 90–98

[10]. Borenstein, J. and Koren, Y., The vector field histogram — fast obstacle avoidance for mobile robots, IEEE Transactions on Robotics and Automation, Vol. 7, No. 3 (1991), pp. 278–288.

[11]. Fox, D., Burgard, W. and Thrun, S., The dynamic window approach to collision avoidance, IEEE Robotics and Automation Magazine, Vol. 4, No. 1 (1997), pp. 23–33.

[12]. Ilon Bengt Erland, "Rad Fuer Ein Laufstabiles, Selbstfahrendes Fahrzeug", issued 1974-05-16

[13]. 原口雅尚，王碩玉，王義娜,超音波センサと測域センサを併用した障害物認識方法,第 29 回バイオメディカル・ファジィ・システム学会年次大会 講演論文集 (BMFSA2016), pp.79-82,高知, 2016 年 11 月.

[14]. 産業技術総合研究所ロボットイノベーション研究センター，Navigation Stack https://robo-marc.github.io/navigation_documents/navigation_overview.html

[15]. M. Przybyła, “Detection and tracking of 2d geometric obstacles from lrf data,” in 2017 11th International Workshop on Robot Motion and Control (RoMoCo), 2017, pp. 135–14

[16]. Kalman, R. E. 1960. “A New Approach to Linear Filtering and Prediction Problems,” Transaction of the ASME—Journal of Basic Engineering, pp. 35-45 (March 1960).

[17]. H. Dong, C. Y. Weng, C. Guo, H. Yu, and I. M. Chen, “Real-time avoidance strategy of dynamic obstacles via half model-free detection and tracking with 2D lidar for mobile robots,” IEEE/ASME Trans. Mechatronics, early access, Oct. 30, 2022

[18]. 伊部直樹，増山岳人，山下淳，淺間一，歩行者の意図推定に基づくロボットの give-way 行動の生成，日本ロボット学会第 30 回記念学術講演会講演論文集, 2N3-1 (2012), pp.1~4.

謝辞

本論文は著者が高知工科大学大学院工学研究科基盤工学専攻知能機械工学コース在学中に行った研究をまとめたものである．

本研究及び本論文を進めるにあたり多大なるご指導をくださいました，高知工科大学知能ロボティクス研究室 王碩玉教授，楊光助教に対し深く感謝致します．さらに，本研究を行うにあたり実験に協力してくれた宮地君，親切に様々な助言，ご協力をしてくださいました後輩達に深く感謝いたします．

最後に，著者のために学生生活を支えてくださった家族，共に学んだ学友，関係者の方々に心より感謝致します．