

NAIST-IS-MT1651109

修士論文

移動ロボットのための安全かつ効率的な局所的経路計画

村瀬 由基美

2018 年 3 月 15 日

奈良先端科学技術大学院大学
情報科学研究科 情報科学専攻

本論文は奈良先端科学技術大学院大学情報科学研究科に
修士（工学）授与の要件として提出した修士論文である。

村瀬 由基美

審査委員：

小笠原 司 教授 （主指導教員）

萩田 紀博 教授 （副指導教員）

高松 淳 准教授 （副指導教員）

丁 明 助教 （副指導教員）

移動ロボットのための安全かつ効率的な局所的経路計画*

村瀬 由基美

内容梗概

多くの人間が行き来する動的環境下でのロボットの移動に着目した際、人間との衝突を回避し安全を保証しながら移動することは不可欠である。そのため、現在だけでなく将来の人間の状態を予測し事前に危険を認識して衝突を回避する必要がある。しかし人間との衝突回避にのみ専念すると、ロボットはあらゆる地点で衝突の可能性があると判断し、目的地へ到達するまでに多くの時間を費やしてしまう。ロボットは人間の安全を保証することに加えて、いかに時間や距離の無駄なく目的地まで移動することができるかという移動効率も考慮するべきである。

本研究では、人間の安全とロボットの移動効率を両立するナビゲーションの実現のため、衝突回避を担う局所的経路計画における人間の動きを考慮した新たな経路計画手法を提案する。人間の将来の状態から、将来ロボットと人間の間で衝突可能性のある領域を予測し、その情報を局所的経路計画における軌道評価に用いることで衝突を回避するための経路を生成する。さらにロボットの移動効率向上のため、予測された衝突領域における人間の移動方向の情報を扱う要素を新たに軌道評価関数に加え、衝突を回避しつつも無駄のない効率的な経路を生成できるようにする。

シミュレータ実験ではロボットと人間が交差する際の軌道の交差角度によって分けられた複数の状況において、Robot Operating System で提供されている従来のナビゲーションとの比較により提案手法によるナビゲーションの性能を評価した。

キーワード

衝突回避, 局所的経路計画, 動的ナビゲーション, コスト関数, 自律移動ロボット

*奈良先端科学技術大学院大学 情報科学研究科 情報科学専攻 修士論文, NAIST-IS-MT1651109, 2018 年 3 月 15 日.

Human-Safe and Robot-Efficient Local Path Planner for Mobile Robot in Dynamic Environments*

Yukimi Murase

Abstract

For robot navigation in dynamic environments where many people are moving around the robot, it is essential to ensure human safety by avoiding collisions between humans and the robot during navigation. In order to do this, it is necessary to estimate the danger of collision in advance by not only considering the current human state but predicting future human states. However, focusing exclusively on collision avoidance can cause the robot to move too slowly to stop entirely. Therefore, in addition to ensuring human safety, the robot should also consider the movement efficiency, i.e., how can the robot reach the goal without wasting time and with the shortest path.

In this thesis, we propose a new path planning method that considers human movement in a local path planner and achieves human-safe and robot-efficient navigation in dynamic environments while avoiding collisions. To generate a safe path, we predict regions with possible collisions between the robot and the human from the future human state and use this information to evaluate a trajectory in the local path planner. Furthermore, in order to improve the movement efficiency of the robot, we add the direction of the human movement of the predicted collision area to the proposed trajectory evaluation function, which allows the robot to plan an evasive maneuver around the human. We evaluated the navigation performance using the proposed method by simulating

*Master's Thesis, Department of Information Science, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT1651109, March 15, 2018.

multiple situations where the robot crosses paths with the human at different angles, and compared it to state-of-the-art methods and the static navigation provided by default in the Robot Operating System.

Keywords:

collision avoidance, local path planning, dynamic navigation, cost function, automobile robots

目次

図目次		v
表目次		vi
第 1 章	はじめに	1
1.1	背景	1
1.2	ロボットナビゲーション	1
1.3	関連研究	3
1.4	目的	4
1.5	本論文の構成	5
第 2 章	提案手法	6
2.1	コストマップ	6
2.1.1	Prediction Layer	8
2.1.2	Direction layer	10
2.2	2 つのコストマップを用いた局所的経路計画	14
2.2.1	Direction cost	14
第 3 章	シミュレータ実験	21
3.1	実験環境	21
3.2	実験条件の設定	23
3.3	実験結果	26
3.4	考察	33
3.4.1	問題点	38
第 4 章	おわりに	42
4.1	本論文のまとめ	42
4.2	今後の課題	43
謝辞		44

図目次

2.1	Structure of a layered costmap in proposed system.	7
2.2	Transition of covariance of Eq. 2.1.2.	9
2.3	An example of inflation case	11
2.4	Direction layer.	12
2.5	Division of distance ranges with the $D_{\min}$ and $D_{\max}$ thresholds [24].	17
2.6	An example of inflation case [24]	18
3.1	Overview of our local path-planner.	22
3.2	Gazebo Pioneer3DX model that is used in the simulator experiments	22
3.3	Three crossing situations differing by the angle between trajectories.	25
3.4	Results in case of $\theta = 90^\circ$	28
3.5	Results in case of $\theta = 45^\circ$	30
3.6	Results in case of $\theta = 135^\circ$	32
3.7	Trajectory comparison for the case of human start position $x = -0.5$ in case of $\theta = 90^\circ$	34
3.8	Trajectory comparison for the case of human start position $x = -2.0$ in case of $\theta = 45^\circ$	35
3.9	Trajectory comparison for the case of human start position $x = -2.0$ in case of $\theta = 135^\circ$	37
3.10	Trajectory generated by proposed approach in the case of human start position $x = -2.5$ in case of $\theta = 90^\circ$	38
3.11	Trajectory comparison for the case of human start position $x = -2.0$ in case of $\theta = 135^\circ$	40

表目次

3.1	Simulation results in case of $\theta = 90^\circ$	27
3.2	Simulation results in case of $\theta = 45^\circ$	29
3.3	Simulation results in case of $\theta = 135^\circ$	31

第 1 章 はじめに

1.1 背景

近年サービスロボットの普及が進み、産業界だけでなくショッピングモールや病院、ホテル、空港などのサービス業界においても幅広く活躍の場を広げている [1, 2, 3]. それに伴い、ロボットは人間と物理的空間を共有することが求められ、多くの人が行き来する環境下で人間との衝突を回避し、人間の安全を保証しながら移動することが不可欠となる.

移動ロボットのナビゲーションは長きにわたり研究されているトピックである. その中でも、人間のような動的物体が存在しない静的環境下でのナビゲーションは技術としてほとんど完成しており、既に問題なく実用化されるまでに達している [4]. 一方、動的環境でのナビゲーションは人間や動的物体を扱うための特別なシステムがなければ、通常ロボットは環境内の物体を全て静的物体として処理するため、静的環境でのナビゲーションに比べて非常に困難である. 動的環境でのナビゲーションを実現するために、位置空間座標だけでなく時空間座標も考慮して動的物体の動きを予測した結果を用いる経路計画手法 [5, 6] や、様々な場面における人間の行動を学習してロボットの行動に反映させる経路計画手法 [7] が提案されている.

ロボットが人間の安全を担保するためには、現在だけでなく将来の人間の位置や速度を予測し事前に危険を回避する必要がある. しかし、その過度な回避行動によりロボットはあらゆる地点で衝突の可能性があると判断し、目的地へ到達するまでに多くの時間を要してしまう. そのため、動的環境でのナビゲーションには人間の安全を保証することと同時に、いかに時間や距離の無駄なく目的地まで移動することができるかという、ロボットの移動効率も考慮する必要がある.

1.2 ロボットナビゲーション

ここで、ロボットナビゲーションについて簡単に説明を述べる. ナビゲーションのフレームワークを構成する数ある要素のうち、主要要素として環境地図生成、自

己位置推定，経路計画が上げられる．特に，ナビゲーション中の衝突回避の役割を含む経路計画について詳しく述べる．

経路計画では，障害物との衝突を回避するために環境の障害物情報を含むコストマップを用いて目的地までの障害物を回避した経路計画を行う手法がある．コストマップは占有格子地図 [8] から派生したもので，マップは格子状に分割されている．占有格子地図では，各セルに格納される情報は占有されているかいないかもしくはは未知であるという情報のみであるのに対し，コストマップでは各セルが障害物によって占有されているかどうかの確率的な推定値が格納される．障害物によって占有されているセルだけでなく，その周辺のセルに対してもロボットが障害物に接近しすぎないようにするための制約がコストとして格納されるため，経路計画において極端に障害物に接近することのないような経路を生成するための環境の情報を表現することができる．

環境地図やセンサデータなどからロボットが目的地までナビゲーションを行う際に 2 つの経路計画が実行される．1 つは大域的経路計画と言われ，ナビゲーションを開始する時に現在の位置からゴールまで，ロボットにとって最適かつ衝突の無い経路を生成する．生成される経路は大域的経路と呼ばれ，大域的経路計画はナビゲーション開始時だけでなく，ロボットがナビゲーション中に大域的経路から大きく外れてしまったり，一時的に障害物に囲まれて身動きが取れなくなってしまう場合に再び実行される．大域的経路計画には Rapidly-Exploring Random Trees (RRT) [9] や，A*アルゴリズム [10]，ダイクストラ法 [11] がよく用いられている．これらを拡張して様々な手法が提案されている [12, 13]．

一方，大域的経路計画が効率的に処理することのできない動的物体との衝突を回避するために実行されるのが局所的経路計画である．大域的経路計画と同様に様々な手法を用いた経路計画が提案されているが，中でもよく知られているのは Trajectory Rollout [14] と Dynamic Window Approach (DWA) [15] を用いた経路計画である．これら 2 つの基本的な経路計画手法は，決められた時間ステップにおけるロボットの制御可能範囲である Dynamic Window を計算し，その範囲において制御入力であるロボットの速度や角速度をある一定値ごとにサンプリングする．そして，サンプリングされた制御入力に対して以下の軌道評価関数 G を用いて評価する．

$$G(v, \omega) = \delta(\alpha \cdot \text{heading}(v, \omega) + \beta \cdot \text{distance}(v, \omega) + \gamma \cdot \text{velocity}(v, \omega)) \quad (1.2.1)$$

一つ目の項の $\text{heading}(v, \omega)$ は目的地に対するロボットの方位、二つ目の項の $\text{distance}(v, \omega)$ は最も近い障害物との距離、三つ目の項の $\text{velocity}(v, \omega)$ はロボットの速度の大きさである。サンプリングされた各制御入力を式 (1.2.1) を用いて評価し、最も評価の高い制御入力を選択することでロボットが障害物との衝突を回避するような経路の生成を可能にする。この局所的経路計画をロボットは移動中常に繰り返すことで、大域的経路計画時に存在しなかった障害物に対する衝突回避を可能にする。

1.3 関連研究

本研究の関連研究として、動的環境における人間の動きを考慮した経路計画に関する研究について紹介する。

人間との衝突を回避し、かつ移動効率も考慮した経路を生成するためには、人間の動きを予測する必要がある。その予測に用いるモデルとして線形モデルを用いる手法 [16, 17, 18] がある。これらの手法は簡単に実装することが可能であるが、モデルが完全な人間の動きを再現しているわけではなく、実際の人間の動きはもっと複雑なものになる。

より複雑な人間の動きを予測可能にするために Social Force Model (SFM) [19] や Human Motion Behaviour Model (HMBM) [20] を用いて実際の人間の動きに近い動作モデルを定義する手法がある。これらは一般的に人間が遭遇する様々なシナリオにおいて、人間の行動に近い行動決定を行うことができるモデルをロボットの経路計画に用いることで、人間が取るような経路をロボットが選択できるようにする。しかし、多くのパラメータ調整が必要であったり、明確な事前環境や考えられる全てのシナリオを定義する必要があり、最適なモデルを生成するまでに多くの時間と労力が費やされる。それにも関わらず想定外な環境やシナリオに遭遇した際にモデルがその環境において最もふさわしい行動を選択するとは限らない。

そのため、事前の情報収集を避けてなるべく簡単な予測モデルや情報のみを経路計画に用いて安全と効率を両立した経路を生成することが望ましい。これには予測

モデルの結果が静的物体であるという情報だけではなく、動的物体かどうかという情報も付加するべきである。

また、Okal ら [21] はシミュレータを用いて、不特定多数の人間が移動している環境で人間の動きの流れを把握し、ロボットと同様の姿勢で移動する人間に追従する方法で目的地までのナビゲーションを可能にしている。しかしリアルタイム性を考慮すると、ロボットが自身のセンサデータのみで処理できる人間の数は限られるため、環境内の人間の数によっては人混みの流れを把握することが困難となる。ロボットは人混みを移動する場合、多くの人間の動作を処理するのではなく、人間に危害を与えない程度にロボット周辺の最小限の処理数でナビゲーションを行う必要がある。

1.4 目的

動的環境下において人間の安全とロボットの移動効率を両立するナビゲーションの実現のため、本研究では衝突回避の役割を担う局所的経路計画に着目する。

ロボットが人間との衝突を回避し安全を保証するためには、現在の状況における将来衝突が発生するかどうかの潜在的な危険を認識し、経路計画に反映させることが重要となる。そこで本研究では、現在の人間の位置情報だけでなく、将来人間との間で起こりうる衝突の可能性がある領域の情報を経路計画で用いられる障害物情報を扱うためのコストマップによって表現する。複雑な人間の行動に対応した予測モデルを定義することは多くの時間を要するため、人間の将来の状態は簡単な予測モデルを用いて推定され、その結果を元に将来ロボットと人間との間で起こりうる衝突を予測する。

予測された将来衝突の可能性がある領域の情報によって、ロボットは人間との間に存在する潜在的な衝突を回避する経路を生成することが可能になるが、衝突回避に専念してしまうと目的地まで到達するのにかなりの時間を要してしまう。また、生成される経路の方向が人間の前方を通るような経路であると、ロボットの移動効率が低下してしまうことも考えられる。そこで、人間の移動方向に関する情報を経路計画で扱うことができるようにするため、障害物情報を扱うコストマップとは別に、人間の移動方向に関する情報を持つコストマップを提案する。

将来の人間の状態から衝突可能性領域の情報を格納したコストマップと、人間の移動方向に関する情報を格納したコストマップから得られる 2 つの情報を、局所的経路計画で経路を生成する際にサンプリングした軌道を評価するための要素として軌道評価関数に用いることで新たな局所的経路計画手法を提案し、人間の安全とロボットの移動効率を両立させるナビゲーションを実現する。

1.5 本論文の構成

2 章では提案手法について説明し、3 章では提案手法をシミュレータを用いて従来のナビゲーションの性能と比較し、評価した結果を述べる。最後に 4 章では本研究のまとめと今後の課題を述べる。

第 2 章 提案手法

本章では，経路計画でよく用いられるコストマップを改良することにより，人間の安全とロボットの移動効率を両立した局所的経路計画を実現するため，将来の人間の情報を含む障害物情報を扱うコストマップと，人間の移動方向に関する情報を扱うコストマップを局所的経路計画に用いることを提案し，実際にこれらの 2 つのコストマップを用いた局所的経路計画の方法について述べる．

2.1 コストマップ

本システムでは，局所的経路計画に用いるコストマップとして階層的コストマップ [22] を用いる．階層的コストマップでは，コストマップデータの処理を特定の機能ごとにレイヤとして分割しているため，動的環境のようにロボットの周囲に様々な状況が点在する環境でナビゲーションを行うための複雑なコストマップを表現することが可能である．

階層的コストマップは最初に設定された処理順序に従って各レイヤのコストマップを更新し，最終的にマスターコストマップと呼ばれるコストマップにコストデータを統合して経路計画に利用する．階層的コストマップには標準として 3 つのレイヤのコストマップが順に設定されている．これらのレイヤは静的レイヤ，障害物レイヤ，インフレーションレイヤの順に処理が行われ，各レイヤのコストはマスターコストマップに蓄積される．

静的レイヤ

ナビゲーションを実行する前に取得した環境の静的情報を格納する．

障害物レイヤ

レーザーセンサや RGB-D センサから得られる障害物データを格納する．

インフレーションレイヤ

ロボットが障害物に衝突する，あるいは接近しすぎることのないように，ロボットの大きさや形状を元にロボットと障害物との間に安全な距離を確保する．

マスターコストマップは2つのステップによって更新される。1つ目のステップでは、マスターコストマップを更新する範囲を取得するために全てのレイヤを参照して、更新する全てのセルを含む最小の矩形を取得する。次のステップでは、1つ目のステップで取得した更新範囲内のコストを初期化し、レイヤの順序に従って更新範囲内のセルのコストを更新する。これらのステップを一定の周期で行うことにより、ロボットはコストマップを効率的に経路計画で 사용할ことが可能となる。

本システムにおける階層的コストマップの構成を Fig. 2.1 に示す。事前に取得した地図からの情報とロボットのセンサから取得した障害物情報のみを扱う3つの標準レイヤに加えて、本システムでは、将来の人間の状態を扱う層である Prediction layer と、人間の移動方向に関する情報を扱う層である Direction layer を新たにレイヤとして追加した階層的コストマップを提案する。

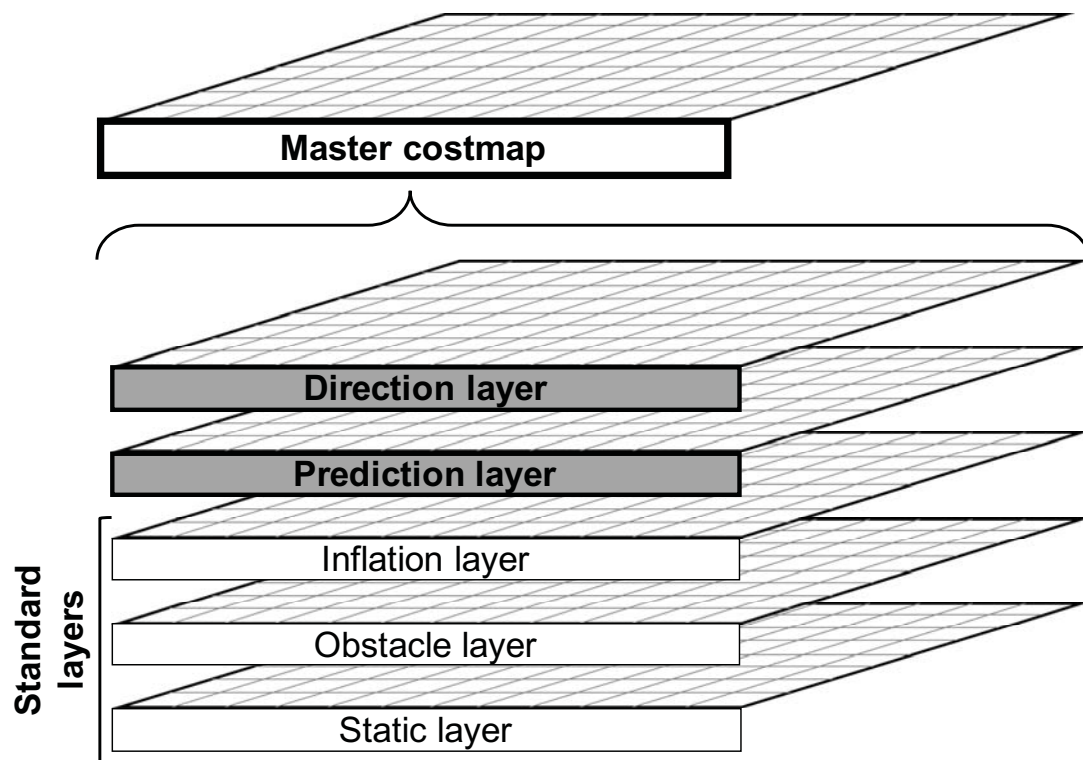


Fig. 2.1: Structure of a layered costmap in proposed system.

2.1.1 Prediction Layer

ロボットが人間との衝突を回避し安全を保証するためには，現在の人間の位置情報を知ることに加えて将来どこで衝突が起きる可能性があるかという危険を事前に予測することも重要となる．そこで本研究では，将来のロボットと人間の状態（位置と速度）によって予測される衝突可能性が高い領域の情報を持つコストマップを Prediction layer として提案する．Prediction layer では簡単な予測モデルを用いて将来のロボットと人間の状態を推定し，その結果から予測される衝突可能性が高い領域にコストを格納する．将来衝突の可能性のある領域の情報を障害物情報としてコストマップに追加することで，局所的経路計画において人間との衝突を事前に回避するような経路生成が可能となる．

Prediction layer では将来の人間の状態を表現するために Encounter time [23] という概念を用いる．Encounter time は予測モデルによって算出されるロボットと人間の軌道上で，両者間の距離が最も近くなる時刻を t_e として以下の式 (2.1.1) のように定義される．

$$t_e = \arg \min_t D(s_r(t), s_h(t)) \quad (2.1.1)$$

ここで， $s_r(t)$ および $s_h(t)$ はそれぞれ時刻 t におけるロボットと人間の状態を表し， $D(*)$ はロボットと人間の間の距離を測定するための関数である．本システムではロボット，人間共に等速運動で移動すると仮定して状態 $s(t)$ を計算する．

時刻 t_e でロボットは人間に最も接近すると考えられるため，その瞬間はロボットにとって最も衝突の危険が高い瞬間とみなすことができる．Prediction layer では等速直線運動モデルによって算出される時刻 t_e における人間の位置周辺のセルに，人間が存在する確率をコストとして格納する．時刻 t_e における各セル c に人間が存在する確率は式 (2.1.2) の確率密度関数によって計算される．

$$P_h(c|t_e) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{(c - s(t_e))^2}{2\sigma^2} \right\} \quad (2.1.2)$$

ここで，ロボットと人間の距離が近づくにつれて人間の状態はより確かなものであると考えることができるため，式 (2.1.2) で表される確率分布の共分散 σ^2 をロ

ボットと人間の距離に応じて変化させる．そのため，Fig. 2.2 のようにロボットと人間の間の距離 d が最大距離 $D_{\max}$ と最小距離 $D_{\min}$ の時に， d が大きくなるにつれて比例的に σ^2 の値も大きくなるように変化させる．

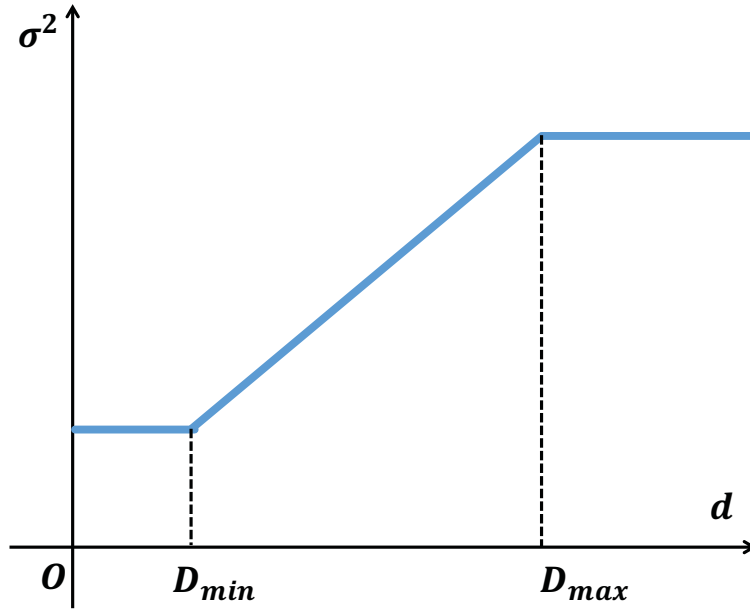


Fig. 2.2: Transition of covariance of Eq. 2.1.2.

アルゴリズム

本システムにおける Prediction layer のアルゴリズムを Algorithm 1 に示す．

Algorithm 1 Prediction layer

Require: s_r, s_h

Ensure: Obstacle cost

- 1: $t_e \leftarrow \arg \min_t D(s_r(t), s_h(t))$
 - 2: **for** each cell c **do**
 - 3: Obstacle cost $\leftarrow P_h(c|t_e)$
 - 4: **end for**
-

2.1.2 Direction layer

Prediction layer によって、人間との衝突を回避するためにロボットが最も回避すべき場所の情報がコストマップとして与えられるため、その情報を用いて局所的経路計画を行うことで衝突可能性のある領域を避けるような経路を生成することができる。しかし、局所的経路計画では障害物情報のみを用いて軌道の評価を行うわけではなく、その軌道と目的地までの距離や、大域的経路計画で生成した経路からどれほど離れているか等の複数の情報を総合的に評価して局所的経路を生成する。そのため、Prediction layer によって与えられた領域を回避する局所的経路であっても、その経路が人間の前方を通る経路であった場合、ロボットの進行方向と同じ方に Prediction layer による領域も移動するため、再度回避するための経路を生成し直す必要があり移動効率が低下するという問題がある。

上記のような移動効率が低くなる状況の例を Fig. 2.3 に示す。図中の色付けされたセルは Prediction layer によるコストが格納されており、青色から赤色に変化するにつれてセルの Prediction layer における確率が大きくなることを示す。また、Fig. 2.3(a) から Fig. 2.3(d) は時系列順に変化する状況を表している。Fig. 2.3(a) においてロボットは Prediction layer による衝突可能性のある領域を右へ回避するような局所的経路を生成したが、生成した経路の方向と人間の移動方向が同じであるため Prediction layer による領域は人間の進行方向へ移動し、Fig. 2.3(b), Fig. 2.3(c) において今度は Fig. 2.3(a) で生成した経路とは反対の方向に経路を生成することで、ようやく Fig. 2.3(d) のように衝突を回避することができる。このような状況では移動時間や移動距離に無駄を生じ、効率の低下につながる。

この問題を解決するために、人間の移動方向の情報を含むコストマップとして新たに Direction layer を提案し、局所的経路計画における経路生成に利用する。Direction layer では Fig. 2.4 に示すように、Prediction layer において式 (2.1.2) の確率密度関数によって求められるコストが格納されている全てのセルに対して人間の移動方向に関する情報を格納する。人間の移動方向に関する情報には、世界座標系における人間の速度 $\vec{v}_h$ の角度 $\alpha_h \in [0, 2\pi]$ を用いる。

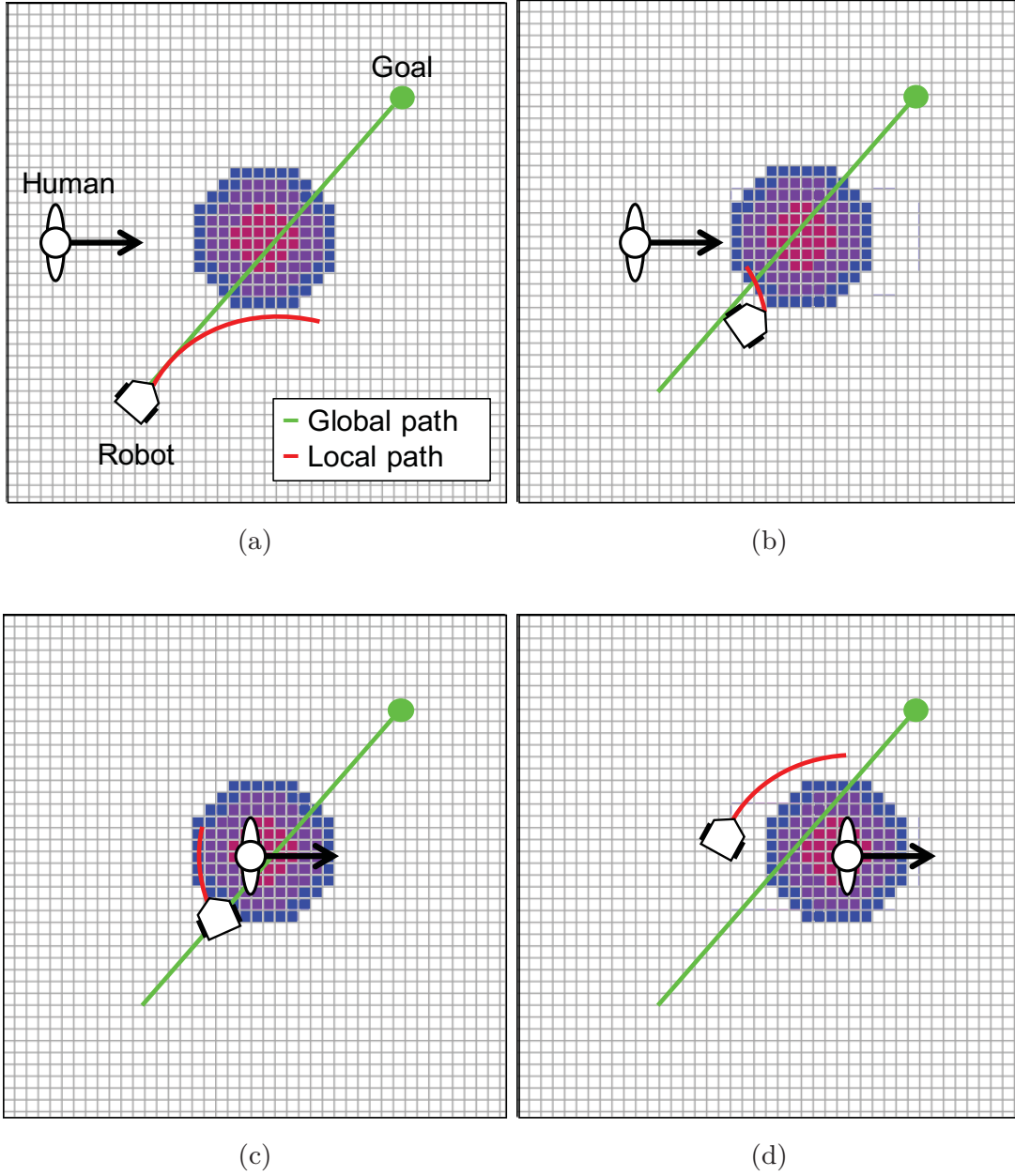
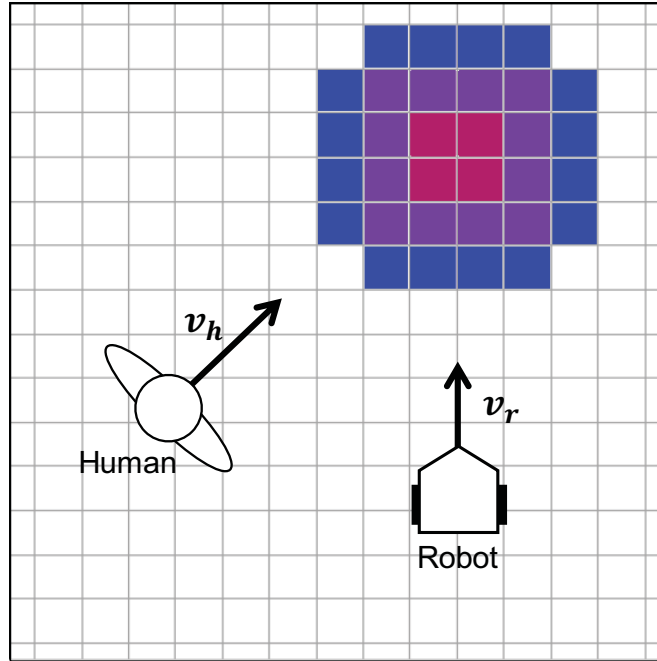
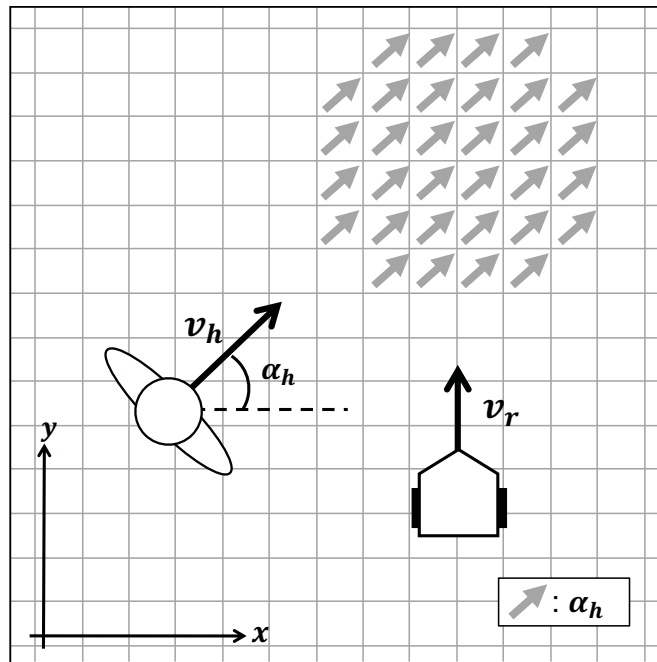


Fig. 2.3: An example of inflation case. The colored cell represents Cost distribution generated by the Prediction layer, the cost increases from blue to red. Fig. 2.3(a) to (d) show the result of a simulation. If the robot had the information about the direction of the human movement, it would not need to replan.



(a) Cost distribution generated by the Prediction layer.



(b) The direction of human movement is stored in cells where there is a cost from the Prediction layer.

Fig. 2.4: Direction layer.

アルゴリズム

本システムにおける Direction layer のアルゴリズムを Algorithm 2 に示す.

C_{MAX} は各セルが格納できる値の最大値であり, $\alpha_h \in [0, 2\pi]$ は各セルが格納できる値の範囲である整数 $[0, 255]$ に変換した値で Direction layer に格納される.

Algorithm 2 Direction layer

Require: s_r, s_h, v_h

Ensure: Human direction

```
1:  $t_e \leftarrow \arg \min_t D(s_r(t), s_h(t))$ 
2:  $\alpha_h \leftarrow \tan^{-1}(\frac{v_{hy}}{v_{hx}})$ 
3: if  $\alpha_h < 0$  then
4:    $\alpha_h \leftarrow \alpha_h + 2\pi$ 
5: end if
6: for each cell  $c$  do
7:   if  $P_h(c|t_e) \neq 0$  then
8:     Human direction  $\leftarrow \alpha_h * \frac{C_{\text{MAX}}}{2\pi}$ 
9:   end if
10: end for
```

2.2 2つのコストマップを用いた局所的経路計画

提案した2つのコストマップの情報を局所的経路計画に利用するため、経路を生成する際に用いる軌道評価関数を新たに定義する。局所的経路計画では、離散的にサンプリングされた速度と角速度が、一定の短い期間適用された場合の各軌道に対して軌道評価関数を用いて軌道の評価値である C_{traj} を計算し、評価値が最も小さくなる軌道を局所的経路として生成する。本システムでは、将来の人間の状態に関する情報と人間の移動方向に関する情報を軌道評価の要素として利用するために、式 (2.2.1) に示すような軌道評価関数を定義する。

$$C_{traj} = scale_1 \cdot D_p + scale_2 \cdot D_g + scale_3 \cdot C_o + scale_4 \cdot C_d \quad (2.2.1)$$

式 (2.2.1) において、 D_p , D_g はそれぞれ軌道の終点から大域的経路までの距離と軌道の終点から目的地までの距離を表し、生成される軌道が大域的経路に沿った軌道であるか、または目的地へ向かう軌道となっているかを評価する要素である。 C_o は Prediction layer を含む障害物情報を扱うコストマップにおいて、軌道が通過するセルのうち障害物コストが最も高くなる時の値を表し、その軌道が障害物に衝突する危険があるかを評価する要素である。 C_d は軌道が通過するセルのうち人間の移動方向に対する軌道の方向のコストである Direction cost が最も高くなる時の値を表し、人間の移動方向に対して軌道が効率的であるかどうかを評価する要素である。また、 $scale_1$, $scale_2$, $scale_3$, $scale_4$ は評価関数の各要素の重みを調整するためのスケールである。

2.2.1 Direction cost

人間の移動方向に対してロボットの軌道が人間の後方を通るような効率的な経路であるかどうかを評価するために Direction cost を定義する。Direction cost は Direction layer から得られる人間の移動方向 α_h と、評価する軌道の終点におけるロボットの速度 $\vec{v}_r$ の角度 α_r を用いて計算される。

人間の移動方向によって軌道进行评估するためには、その軌道の方向と人間の移動方向に応じて値が変化するような指標が必要となる。また、局所的経路計画はロ

ロボットが環境中の動的物体に対し素早く反応できるようにするため、動的物体の動きを時間空間も考慮して先読みする評価方法より、計算コストが少ない現在の状況を直感的に評価する方法がふさわしいと考える。Garcia ら [24] は人間の安全とロボットの動作効率を両立するために、人間の危険概念に基づくヒューリスティックなアプローチによって速度を制限する手法として Asymmetric Velocity Moderation (AVM) を提案している。この手法は人間とヒューマノイドロボットが近接して作業するような環境において、人間とロボットとの距離と相対速度に応じてロボットの速度を非対称的に制限することによって、人間の安全とロボットの動作効率を両立している。本システムにおける Direction cost の計算に AVM の概念を用いることによって、人間の移動方向を考慮した経路を生成できるようにする。

AVM では、本来ロボットと人間との距離 d が Fig. 2.5 に示すような 3 つの範囲のうち、どの範囲にあてはまるかによって速度制限を計算するために用いる補助円の大きさを Fig. 2.6 のように線形に変化させる。

Fig. 2.6 において、速度制限のための値である f_{AVM} は O' から補助円上の点までの長さに対応しており、 O' は d に応じて移動する。 f_{AVM} は、余弦定理を用いて Fig. 2.6 の r と r' から式 (2.2.2) のように計算される。

$$f_{AVM} = r' \cos(\pi - \theta) + \sqrt{r'^2 \cos^2(\pi - \theta) + r^2 - r'^2} \quad (2.2.2)$$

ここで、 f_{AVM} は以下の条件を満たすように設計されている。

$d \geq D_{\max}$ の場合

f_{AVM} は θ の値に関係なく 1 の値をとる。このとき $r = 1$, $r' = 0$ となる。

$d \leq D_{\min}$ の場合

f_{AVM} は $\theta = 0$, すなわち人間に向かって移動するときの値を 0 とし、 $\theta = \pi$ のとき $2r$ の値をとる。

これらの条件を満たす r と r' はロボットと人間の間の距離 d に応じて式 (2.2.3) のように表される。

$$r = \left(1 - \frac{F_{\text{rec}}}{2}\right) \frac{d - D_{\min}}{D_{\max} - D_{\min}} + \frac{F_{\text{rec}}}{2}$$

$$r' = \frac{F_{\text{rec}}}{2} \frac{D_{\text{max}} - d}{D_{\text{max}} - D_{\text{min}}} \quad (2.2.3)$$

式 (2.2.3) における F_{rec} は、ロボットが人間からゆっくりと離れて距離を回復するための係数である． F_{rec} には補助円の最小直径が定義される．

本システムでは各セルでの AVM による速度制限の値を Direction cost とし，Fig. 2.5 の範囲のうち “Too close” の場合，すなわち $d \leq D_{\text{min}}$ の場合の補助円に基づいて計算する．

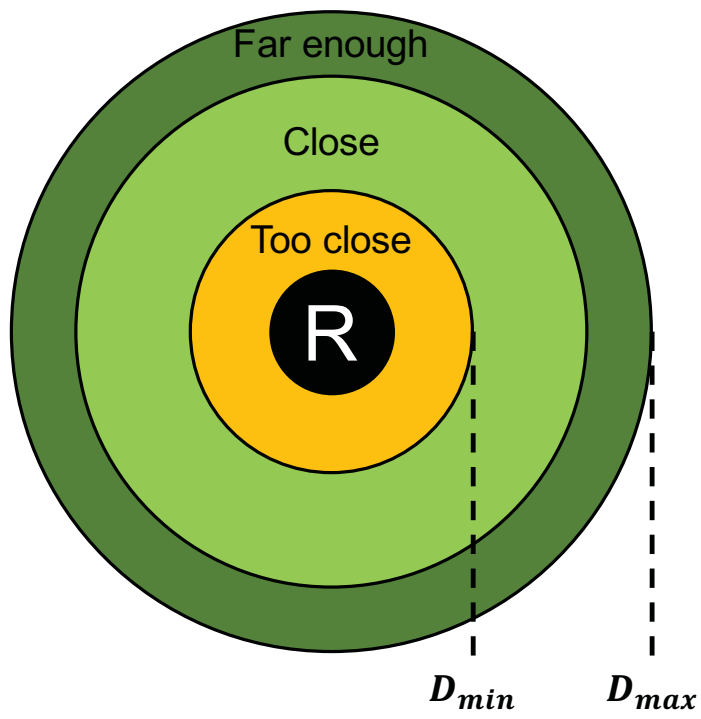


Fig. 2.5: Division of distance ranges with the D_{min} and D_{max} thresholds [24].

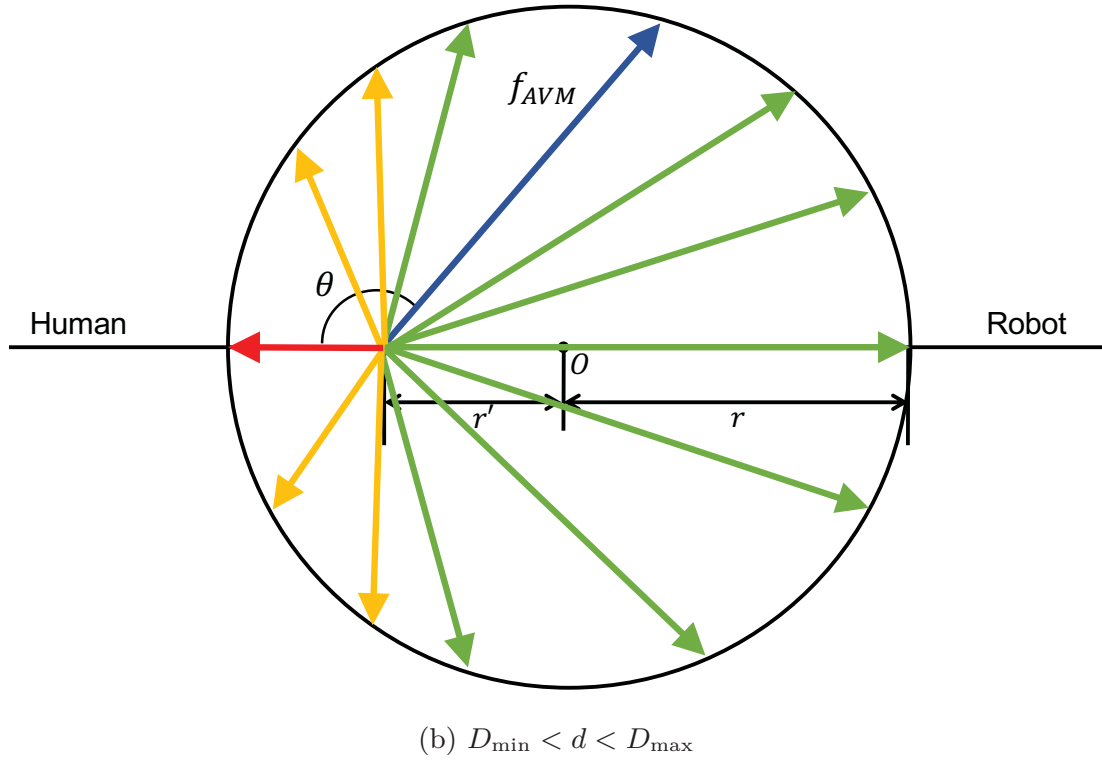
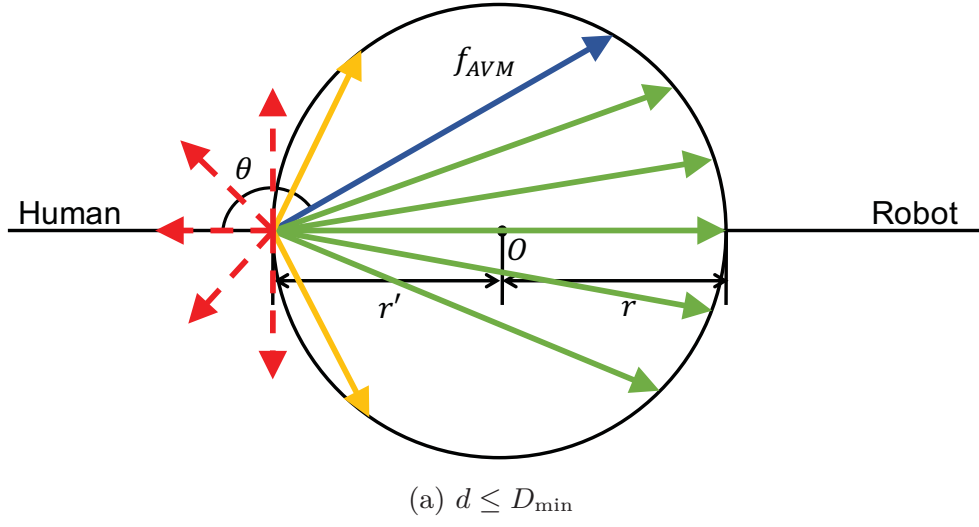


Fig. 2.6: Speed restrictions for any direction of the motion using the auxiliary circle [24].

Direction cost は、局所的経路計画における軌道評価時に各セルのデータを参照する際、Direction layer によって与えられる人間の移動方向に関する情報が格納されている場合に AVM を用いて計算される。

$d \leq D_{\min}$ の場合、式 (2.2.3) より r と r' は式 (2.2.4) のようになる。

$$r = r' = \frac{F_{\text{rec}}}{2} \quad (2.2.4)$$

また、式 (2.2.2) の f_{AVM} は式 (2.2.5) のように表すことができる。

$$f_{\text{AVM}} = r' \cos(\pi - \theta) + \sqrt{r'^2 \cos^2(\pi - \theta)} \quad (2.2.5)$$

本システムにおいて θ はロボットから見た人間の速度の角度とし、世界座標系におけるロボットの速度 v_r の角度 $\alpha_r \in [0, 2\pi]$ を用いて、 $\theta = \alpha_h - \alpha_r$ と表される。

以上から求まる f_{AVM} と、隣り合うセルの急激な値の変化を防ぐため、軌道終点の障害物コストを正規化した値でスケールリングすることにより、Direction cost C_d は式 (2.2.6) で定義される。

$$C_d = (F_{\text{rec}} - f_{\text{AVM}}) \cdot \frac{C_o}{C_{o\text{MAX}}} \quad (2.2.6)$$

アルゴリズム

本システムにおいて、Direction cost の計算アルゴリズムを Algorithm 3、局所的経路計画に用いる軌道評価関数のアルゴリズムを Algorithm 4 に示す。

Algorithm 3 Direction cost

Require: α_h, α_r, C_o **Ensure:** C_d

- 1: $\theta \leftarrow \alpha_h - \alpha_r$
 - 2: $r \leftarrow \frac{F_{\text{rec}}}{2}$
 - 3: $f_{\text{AVM}} \leftarrow r \cos(\pi - \theta) + \sqrt{r^2 \cos^2(\pi - \theta)}$
 - 4: $C_d \leftarrow (F_{\text{rec}} - f_{\text{AVM}}) \cdot \frac{C_o}{C_{o\text{MAX}}}$
-

Algorithm 4 Trajectory cost function

Require: $\Delta t, s_r, a_r, v_{\text{sample}}, v_{\text{theta_sample}}, \text{Master costmap, Direction layer}$ **Ensure:** C_{traj}

- 1: $\text{steps} \leftarrow \frac{\text{simtime} * |v_r|}{\text{granularity}}$
 - 2: $D_p \leftarrow 0, D_g \leftarrow 0, C_o \leftarrow 0, C_d \leftarrow 0$
 - 3: **for** $\text{step}_i = 0$ to steps **do**
 - 4: $t \leftarrow \Delta t * \text{step}_i$
 - 5: $vr \leftarrow \text{calculateState}(s_r, v_{\text{sample}}, v_{\text{theta_sample}}, a_r, t)$
 - 6: $D_p \leftarrow \text{get distance between cell } c \text{ to Global path}$
 - 7: $D_g \leftarrow \text{get distance between cell } c \text{ to goal}$
 - 8: $C_o \leftarrow \text{get cost of cell } c \text{ from Master costmap}$
 - 9: $\alpha_h \leftarrow \text{get cost of cell } c \text{ from Direction layer}$
 - 10: $\alpha_h \leftarrow \alpha_h * \frac{2\pi}{C_{\text{max}}}$
 - 11: $\alpha_r \leftarrow \tan^{-1}\left(\frac{v_{ry}}{v_{rx}}\right)$
 - 12: **if** $\alpha_r < 0$ **then**
 - 13: $\alpha_r \leftarrow \alpha_r + 2\pi$
 - 14: **end if**
 - 15: $C_d' \leftarrow \text{calculate } C_d$
 - 16: $C_d \leftarrow \min(C_d, C_d')$
 - 17: **end for**
 - 18: $C_{\text{traj}} \leftarrow \text{scale}_1 \cdot D_p + \text{scale}_2 \cdot D_g + \text{scale}_3 \cdot C_o + \text{scale}_4 \cdot C_d$
-

第 3 章 シミュレータ実験

提案手法によるナビゲーションの性能を評価するために，シミュレータ上に構築した動的環境において，ROS で提供されている従来のナビゲーションシステムと提案手法によるナビゲーションとの比較を行った．

3.1 実験環境

実験環境として Ubuntu 14.04 をインストールした PC とロボットソフトウェアプラットフォームの一つである ROS Indigo [25] を使用し，3 次元物理シミュレータである Gazebo [26] を用いて実験を行った．人間が歩行している環境でロボットがナビゲーションを行うといった状況を Gazebo 上で再現した．ナビゲーション中のロボットが取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．Fig. 3.1 に可視化されたナビゲーション実行時の様子を示す．ロボットから出る緑色と赤色の線はそれぞれ大域的経路と局所的経路であり，地面に描画されている円形などの様々な形はコストマップのコストが分布している箇所を表している．

シミュレータ実験で用いるロボットは Fig. 3.2 に示す Pioneer3DX の Gazebo モデルを用いた．ロボット前方にはレーザレンジファインダとして SICK 社製 LMS100 の Gazebo モデルが搭載されており，障害物検出や自己位置推定のためにセンサデータを利用した．

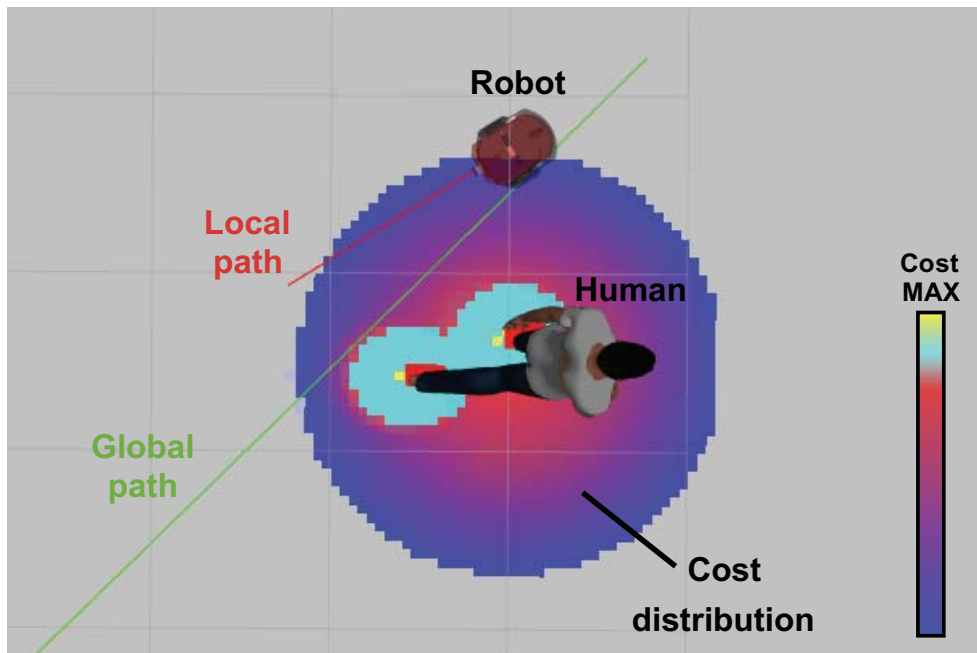


Fig. 3.1: Overview of our local path-planner.

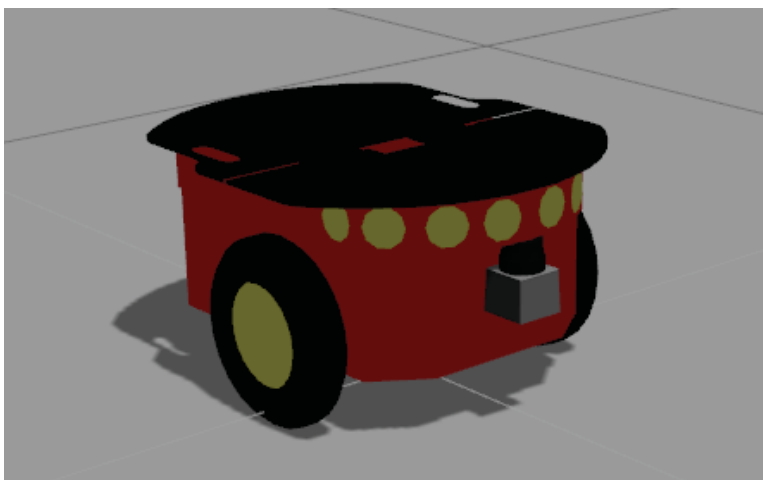


Fig. 3.2: Gazebo Pioneer3DX model that is used in the simulator experiments

3.2 実験条件の設定

本システムを評価するために、ロボットの目的地までの軌道と一人の人間が歩行する軌道が交差するというシナリオを用意し、軌道の交差角度によって3つのパターンに分けて実験を行った。人間の軌道と、ロボットの移動開始地点と目的地を直線で結んだ軌道について、各パターンごとに Fig. 3.3 に示す。各地点は世界座標系において点 $(x[m], y[m])$ で表される。

1つ目のパターンは Fig. 3.3(a) に示すように、それぞれの軌道がなす角度が 90° すなわち直交する場合を想定している。世界座標系においてロボットの移動開始地点は点 $(3.0, 3.0)$ で、点 $(3.0, -3.0)$ が目的地としてロボットに指示される。

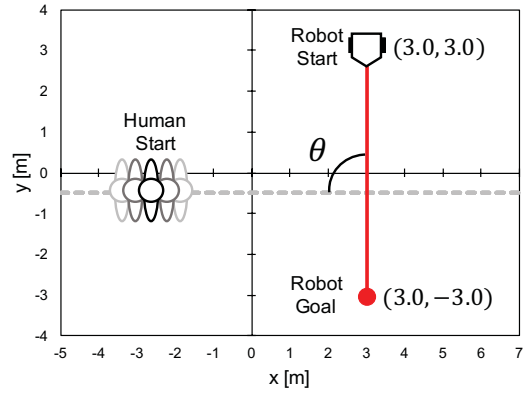
2つ目のパターンは Fig. 3.3(b) に示すように、それぞれの軌道がなす角度が 45° の場合を想定している。世界座標系においてロボットの移動開始地点は点 $(-0.5, 3.0)$ で、点 $(5.5, -3.0)$ が目的地としてロボットに指示される。

3つ目のパターンは Fig. 3.3(c) に示すように、それぞれの軌道がなす角度が 135° の場合を想定している。世界座標系においてロボットの移動開始地点は点 $(6.5, 3.0)$ で、点 $(0.5, -3.0)$ が目的地としてロボットに指示される。

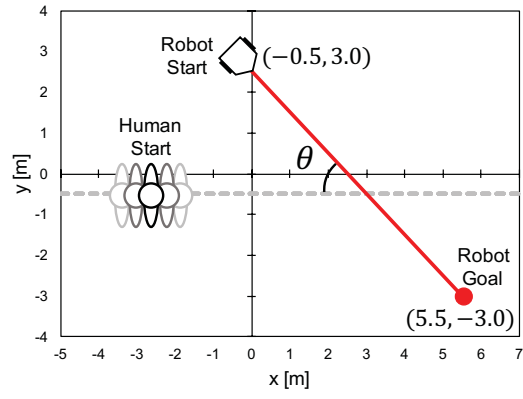
ロボットは環境内の人の位置と速度の情報を常に取得できると仮定し、ロボットの最大速度は $0.5[m/s]$ に設定した。全てのパターンにおいて人間の軌道は同一で、世界座標系において $y = -0.5$ の直線に沿って速度 $v_h = (0.5, 0)$ で移動する。全てのパターンにおいてロボットと人間は同時に移動を開始し、人間の移動開始地点を $(-5.0, -0.5)$ から $(3.0, -0.5)$ の間で x 座標位置を $0.5[m]$ ずつ変化させて計17回のシミュレーションを行った。ロボットがナビゲーション中に人間と遭遇した際の行動を解析し、ゴールまでの到達時間とナビゲーション中の人間とロボットとの最小距離の2つを評価指数とした。

人間とロボットとの間の最小距離は衝突を回避出来ているかの指標であり、人間の安全を確保出来ているかを確認することができる。人間とロボットの距離は、それぞれの中心位置によって計算する。そのため衝突の有無の判定には、シミュレータにおける人間とロボットのモデルの最大半径の和を衝突の距離基準として用いて、最小距離が距離基準を下回れば衝突したと見なす。シミュレータ実験に用いた人間とロボットのモデルの大きさから、本実験での距離基準は全てのパターンにおいて $0.625[m]$ とした。

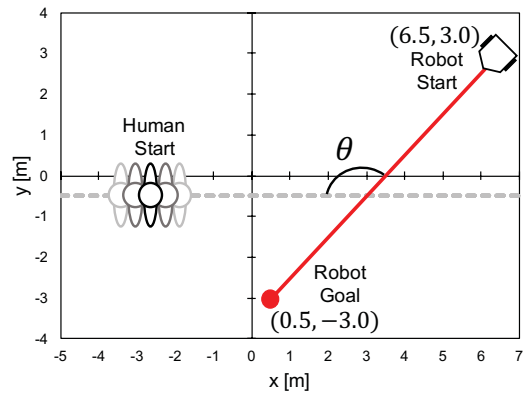
実行時間はどれだけ早く目的地に到達できるかの指標であり，ロボットの移動が効率的であるかを確認することができる．各パターンにおいて移動開始地点から目的地まで人間と遭遇しない場合のナビゲーション，すなわち直線距離を移動した時のナビゲーションを最も効率の良い経路の時間基準とするため，移動開始地点と目的地を直線で結んだ経路の移動を 10 回実行した際の平均実行時間を計測した．移動開始地点から目的地までの直線距離によって実行時間も異なるため，平均実行時間は各パターンごとに計測した．



(a) $\theta = 90^\circ$



(b) $\theta = 45^\circ$



(c) $\theta = 135^\circ$

Fig. 3.3: Three crossing situations differing by the angle between trajectories.

3.3 実験結果

各パターンにおいて，提案手法によるナビゲーションと従来の ROS ナビゲーションで性能比較を行った．また，提案手法によって生成される軌道が人間の移動方向に対して後方を通るような効率的な経路であるかを確認するために，軌道評価関数に Direction cost C_d を用いる場合と用いない場合のナビゲーションに分けて比較を行った．評価のために，ナビゲーション中の人間とロボットとの間の最小距離，ロボットが指示を受け取ってから目的地に到達するまでの実行時間を計測した．以下に各パターンごとのシミュレーション結果を示す．

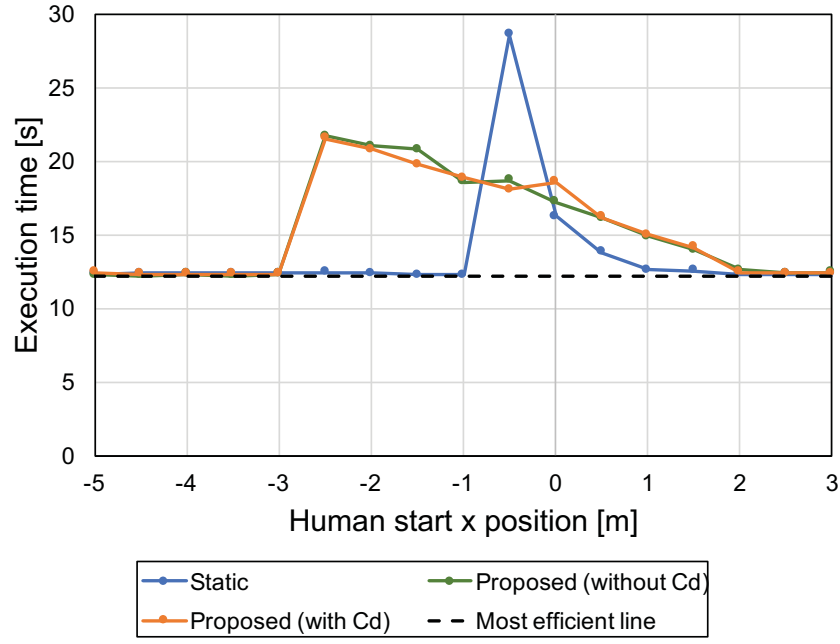
$\theta = 90^\circ$ の場合

ロボットと人間の軌道の交差角度が 90° の場合における，提案手法によるナビゲーションと従来の ROS ナビゲーションによるシミュレーション結果を Table 3.1 と Fig. 3.4 に示す．Fig. 3.4 はそれぞれナビゲーション開始時の人間の x 座標位置ごとに実行時間を計測した結果と，ナビゲーション中の人間とロボットとの最小距離を計測した結果を表している． $\theta = 90^\circ$ の場合における時間基準は 12.19[s] であり，Fig. 3.4(a) において点線で示す．また，Fig. 3.4(b) において衝突判定の距離基準を点線で示す．

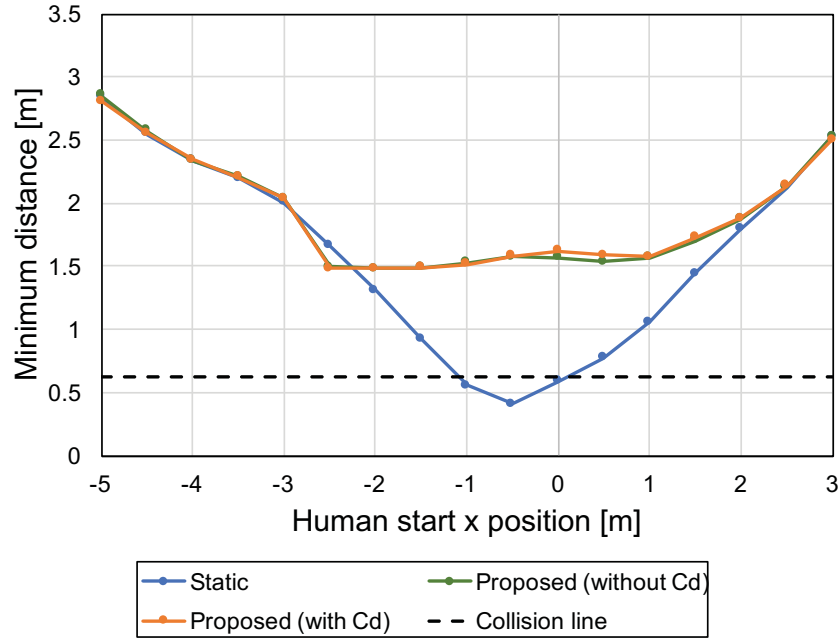
Table 3.1 より，従来の ROS ナビゲーションに比べて，提案手法によるナビゲーションは C_d を用いる場合と用いない場合共に実行時間が長くなるという結果であったが，従来の ROS ナビゲーションでは Fig. 3.4(b) から分かるように，距離基準により 17 回の試行の内 3 回の試行で衝突があったと判定することができ，これは人間との衝突回避の点において安全なナビゲーションであったとは言えない．一方，提案手法によるナビゲーションは C_d を用いる場合と用いない場合共に衝突がなく，人間と一定の距離を保ちながら衝突を回避できていた．従来の ROS ナビゲーションと比較すると，その安全性は明らかである．しかし Fig. 3.4(a) から分かるように，人間の移動開始地点の x 座標が -2.5 の試行において，従来の ROS ナビゲーションにおいても衝突がなく実行時間もほぼ時間基準と同じであったにも関わらず，提案手法によるナビゲーションでは最も実行時間が高くなった．

Table 3.1: Simulation results in case of $\theta = 90^\circ$

	Static navigation		Proposed approach without C_d		Proposed approach with C_d	
	Execution time [s]	Minimum distance [m]	Execution time [s]	Minimum distance [m]	Execution time [s]	Minimum distance [m]
average	13.64	1.59	15.38	1.91	15.39	1.91
min	12.27	0.41	12.18	1.49	12.30	1.48
max	28.57	2.84	21.67	2.86	21.51	2.81



(a) Execution time.



(b) Minimum distance between human and robot while navigating.

Fig. 3.4: Results in case of $\theta = 90^\circ$.

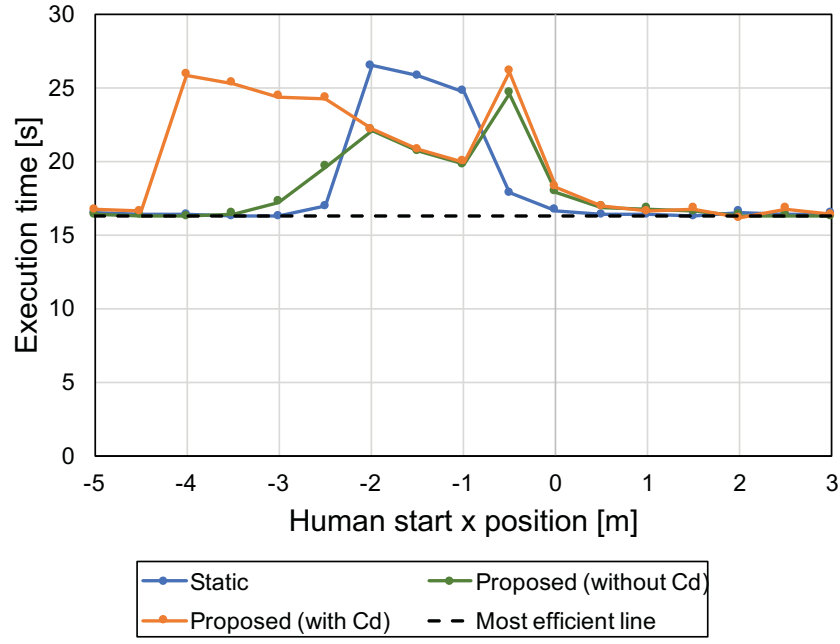
$\theta = 45^\circ$ の場合

ロボットと人間の軌道の交差角度が 45° の場合における，提案手法によるナビゲーションと従来の ROS ナビゲーションによるシミュレーション結果を Table 3.2 と Fig. 3.5 に示す．Fig. 3.5 はそれぞれナビゲーション開始時の人間の x 座標位置ごとに実行時間を計測した結果と，ナビゲーション中の人間とロボットとの最小距離を計測した結果を表している． $\theta = 45^\circ$ の場合における時間基準は 16.30[s] であり，Fig. 3.5(a) において点線で示す．また，Fig. 3.5(b) において衝突判定の距離基準を点線で示す．

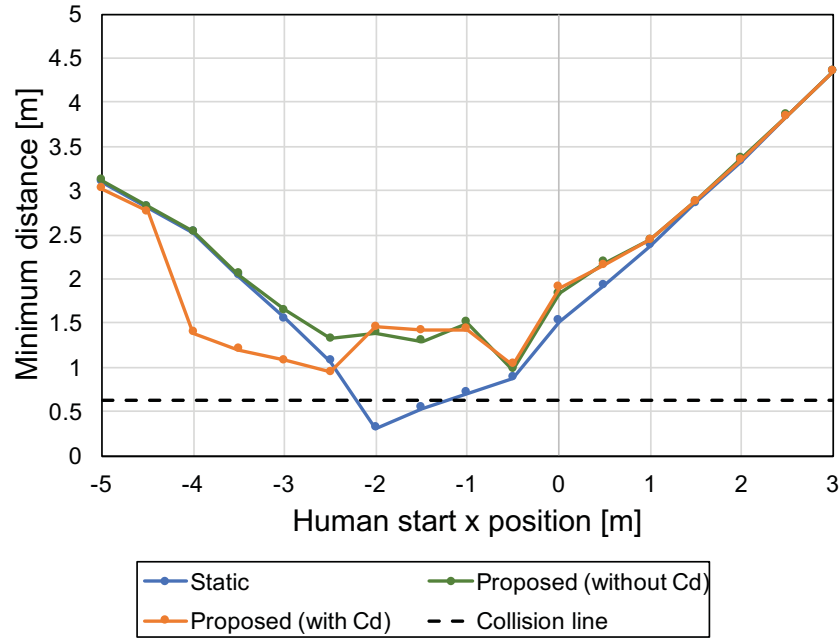
Table 3.2 より，従来の ROS ナビゲーションに比べて，提案手法によるナビゲーションは C_d を用いる場合に実行時間が長くなるという結果であったが，従来の ROS ナビゲーションでは Fig. 3.5(b) から分かるように，距離基準により 17 回の試行の内 2 回の試行で衝突があったと判定することができ，これは人間との衝突回避の点において安全なナビゲーションであったとは言えない．一方，提案手法によるナビゲーションは C_d を用いる場合と用いない場合共に衝突がなく，人間と一定の距離を保ちながら衝突を回避できていた．従来の ROS ナビゲーションと比較すると，その安全性は明らかである．しかし Fig. 3.5(a) から分かるように，人間の移動開始地点の x 座標が -4.0 の試行において，従来の ROS ナビゲーションにおいても衝突がなく実行時間もほぼ時間基準と同じであったにも関わらず， C_d を用いる場合の提案手法によるナビゲーションでは他の試行に比べて実行時間が高くなった．

Table 3.2: Simulation results in case of $\theta = 45^\circ$

	Static navigation		Proposed approach without C_d		Proposed approach with C_d	
	Execution time [s]	Minimum distance [m]	Execution time [s]	Minimum distance [m]	Execution time [s]	Minimum distance [m]
average	18.12	2.10	18.00	2.33	19.97	2.15
min	16.21	0.3109	16.23	0.98	16.12	0.95
max	26.44	4.342	24.57	4.34	26.03	4.34



(a) Execution time.



(b) Minimum distance between human and robot while navigating.

Fig. 3.5: Results in case of $\theta = 45^\circ$.

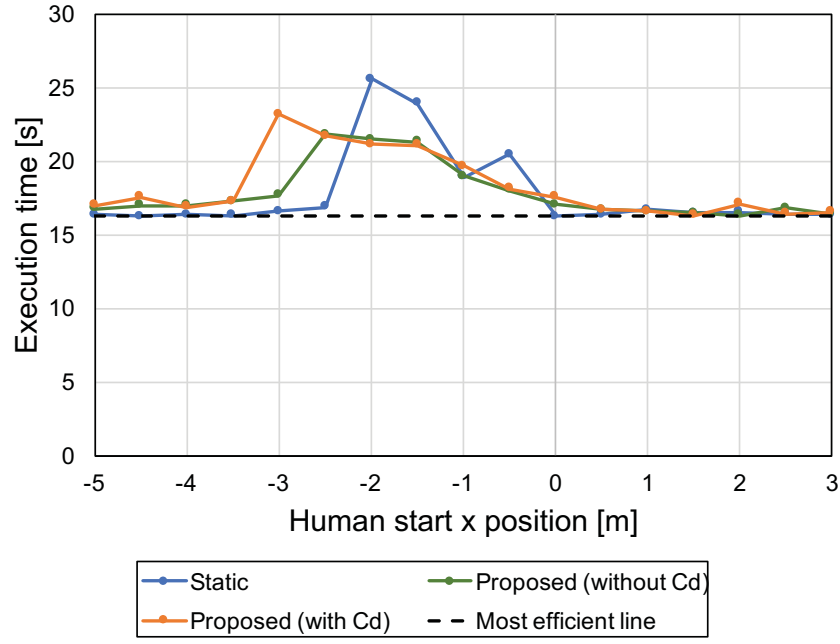
$\theta = 135^\circ$ の場合

ロボットと人間の軌道の交差角度が 135° の場合における，提案手法によるナビゲーションと従来の ROS ナビゲーションによるシミュレーション結果を Table 3.3 と Fig. 3.6 に示す．Fig. 3.6 はそれぞれナビゲーション開始時の人間の x 座標位置ごとに実行時間を計測した結果と，ナビゲーション中の人間とロボットとの最小距離を計測した結果を表している． $\theta = 135^\circ$ の場合におけるでの時間基準は 16.27[s] であり，Fig. 3.6(a) において点線で示す．また，Fig. 3.6(b) において衝突判定の距離基準を点線で示す．

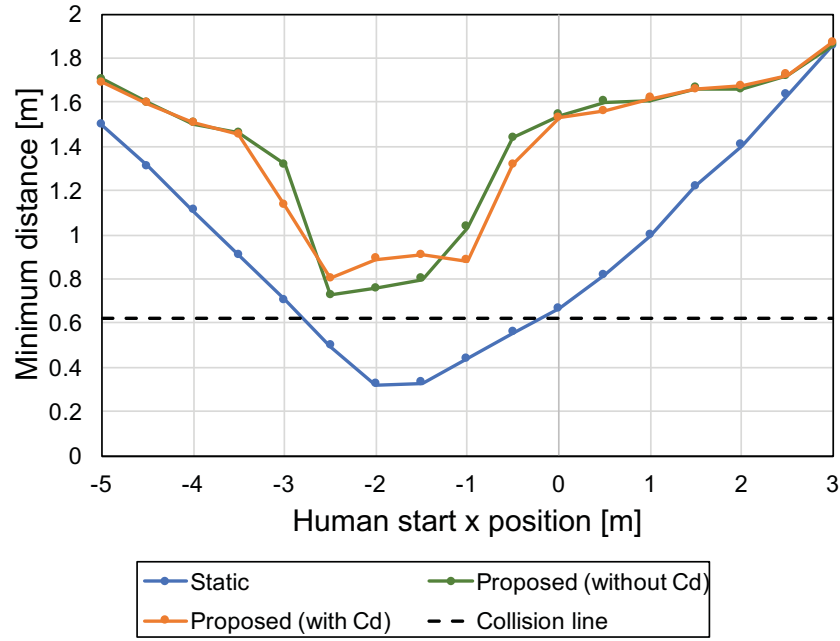
Fig. 3.6(b) から分かるように，従来の ROS ナビゲーションに比べて，提案手法によるナビゲーションは C_d を用いる場合と用いない場合共に実行時間が長くなるという結果であったが，従来の ROS ナビゲーションでは Fig. 3.6(b) から分かるように，距離基準により 17 回の試行の内 5 回の試行で衝突があったと判定することができ，これは人間との衝突回避の点において安全なナビゲーションであったとは言えない．一方，提案手法によるナビゲーションは C_d を用いる場合と用いない場合共に衝突がなく，人間と一定の距離を保ちながら衝突を回避できていた．従来の ROS ナビゲーションと比較すると，その安全性は明らかである．しかし Fig. 3.6(a) から分かるように，人間の移動開始地点の x 座標が -3.0 の試行において，従来の ROS ナビゲーションにおいても衝突がなく実行時間もほぼ時間基準と同じであったにも関わらず， C_d を用いる場合の提案手法によるナビゲーションでは最も実行時間が高くなった．

Table 3.3: Simulation results in case of $\theta = 135^\circ$

	Static navigation		Proposed approach without C_d		Proposed approach with C_d	
	Execution time [s]	Minimum distance [m]	Execution time [s]	Minimum distance [m]	Execution time [s]	Minimum distance [m]
average	17.79	0.96	17.83	1.41	18.26	1.40
min	16.21	0.32	16.27	0.73	16.26	0.80
max	25.57	1.85	21.80	1.86	23.18	1.87



(a) Execution time.



(b) Minimum distance between human and robot while navigating.

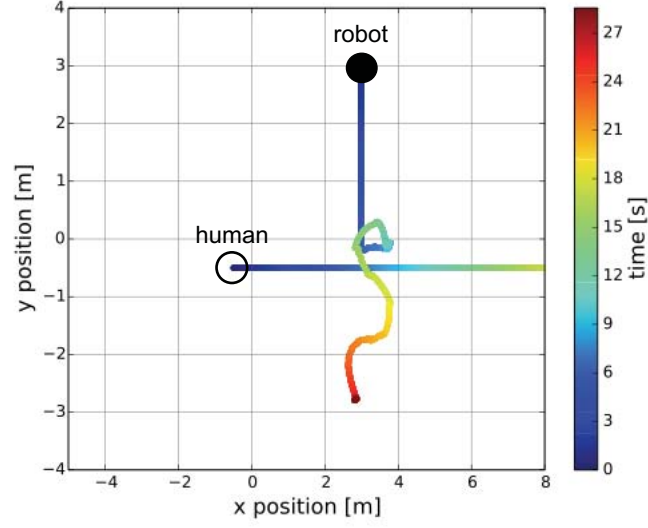
Fig. 3.6: Results in case of $\theta = 135^\circ$.

3.4 考察

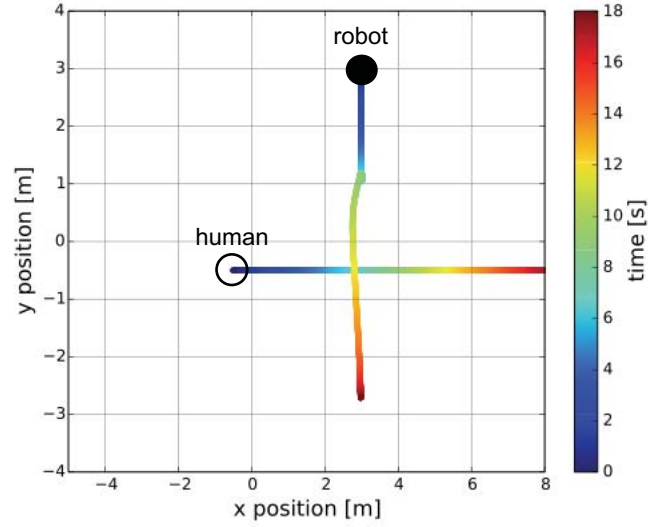
ロボットと人間の軌道の全交差角度のパターンにおいて提案手法のナビゲーションに衝突は無く，衝突可能性が最も高くなる試行においても人間と一定の距離を保ちながらナビゲーションを行うことができていた．以下で各パターンごとの試行を例に考察する．

$\theta = 90^\circ$ の場合において人間の移動開始地点の x 座標が-0.5 の試行を安全なナビゲーションの例として挙げる．Fig. 3.3(a) より，ロボットと人間が2つの軌道の交点にほぼ同時に到達するのは，双方の速度を考慮すると人間の移動開始地点の x 座標が-0.5 付近の試行であると推測でき，最も衝突の危険があると考えられる．この試行における従来の ROS ナビゲーションと， C_d を用いる場合の提案手法によるナビゲーションでのロボットと人間の軌道を時間経過によって色付けしたものを Fig. 3.7 に示す．Fig. 3.7(a) より，従来の ROS ナビゲーションではロボットは人間を静的障害物として処理するため，目的地までの経路上に人間が移動するまで回避行動をとらず，結果としてロボットは目的地まで直進し衝突が発生していることがわかる．一方 C_d を用いる場合の提案手法によるナビゲーションでは Fig. 3.7(b) より，ロボットは人間に接近しすぎる前に減速し，人間が通り過ぎるのを待つことで衝突を回避していることがわかる．これは Prediction layer にによって将来衝突の可能性のある地点を事前に認識することができるため，ロボットは危険領域に近づくと減速し，衝突のないナビゲーションを可能にしていると考えられる．

また， $\theta = 45^\circ$ の場合においても $\theta = 90^\circ$ の場合と同様の回避行動を確認することができた． $\theta = 45^\circ$ の場合において人間の移動開始地点の x 座標が-2.0 の試行を安全なナビゲーションの例として挙げる．Fig. 3.3(b) より，ロボットと人間が2つの軌道の交点にほぼ同時に到達するのは，人間の移動開始地点の x 座標が-2.0 付近の試行であると推測でき，最も衝突の危険があると考えられる．この試行における従来の ROS ナビゲーションと， C_d を用いる場合の提案手法によるナビゲーションでのロボットと人間の軌道を時間経過によって色付けしたものを Fig. 3.8 に示す．Fig. 3.8(a) より，従来の ROS ナビゲーションではロボットは事前に衝突回避するための行動をとることができず衝突していることがわかる．一方 C_d を用いる場合の提案手法によるナビゲーションでは Fig. 3.8(b) より， $\theta = 90^\circ$ の場合と同様に，人間が通り過ぎるのを待つことで衝突を回避していることがわかる．

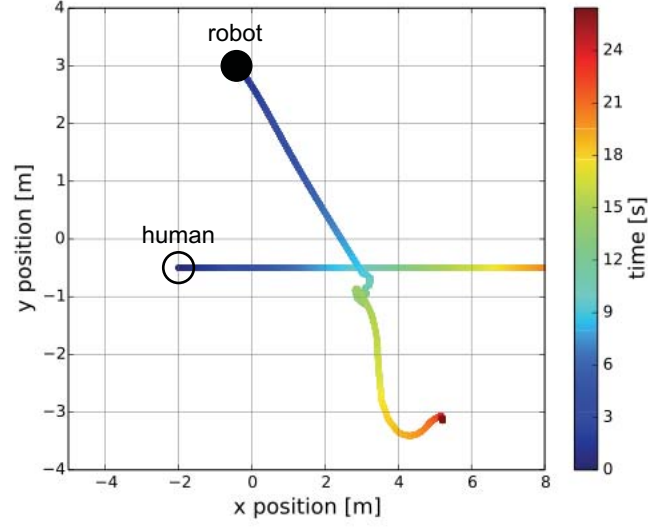


(a) Static approach.

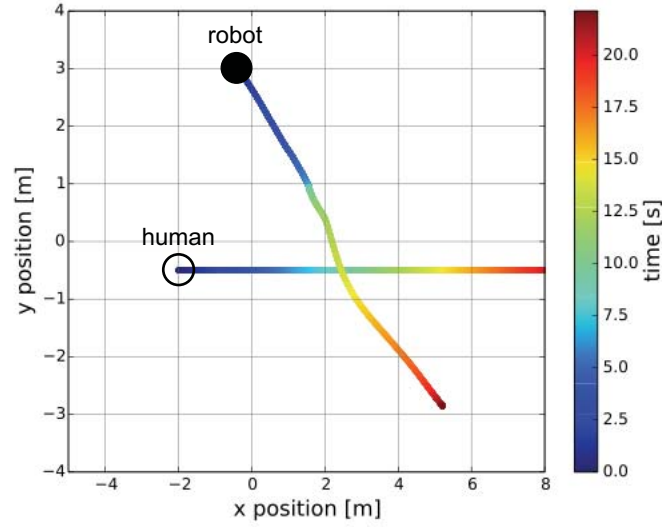


(b) Proposed approach with C_d .

Fig. 3.7: Trajectory comparison of static approach and proposed approach in case of $\theta = 90^\circ$. The human starts at point $(-0.5, -0.5)$. Trajectories are colored to visualize the time dimension.



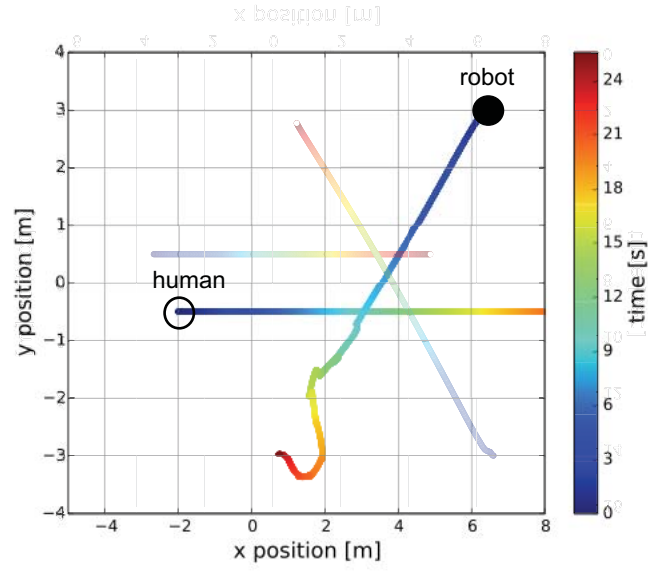
(a) Static approach.



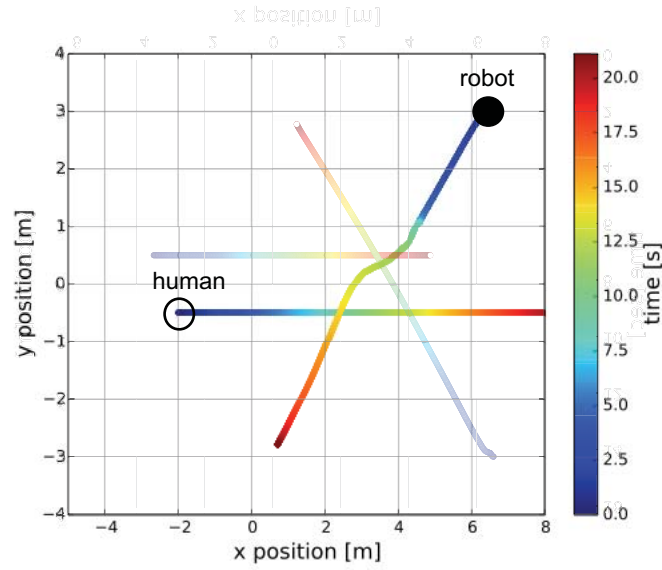
(b) Proposed approach with C_d .

Fig. 3.8: Trajectory comparison of static approach and proposed approach in case of $\theta = 45^\circ$. The human starts at point $(-2.0, -0.5)$. Trajectories are colored to visualize the time dimension.

次に $\theta = 135^\circ$ の場合の安全なナビゲーションの例を挙げる．Fig. 3.3(c) より，ロボットと人間が2つの軌道の交点にほぼ同時に到達するのは，双方の速度を考慮すると人間の移動開始地点の x 座標が-2.0 付近の試行であると推測でき，最も衝突の危険があると考えられる．この試行における従来の ROS ナビゲーションと， C_d を用いる場合の提案手法によるナビゲーションでのロボットと人間の軌道を時間経過によって色付けしたものを Fig. 3.9 に示す．Fig. 3.9(a) より，従来の ROS ナビゲーションではロボットは事前に衝突回避するための行動をとることができず衝突が発生していることがわかる．一方 C_d を用いる場合の提案手法によるナビゲーションでは Fig. 3.9(b) より，ロボットは人間に接近しすぎる前に減速し，人間の移動方向に対して後方を通る経路によって衝突を回避していることがわかる．人間の移動方向と同じ方向に移動して衝突を回避しようとすることは，現在の衝突危険性を回避できたとしても将来の衝突危険性を回避したとは言えず，一定の距離を移動し，人間にある程度接近してから後方を通る経路を再計画する必要があるため，移動効率の低下に繋がると考える．ロボットは Prediction layer によって得られる衝突危険領域を認識した際に，人間の移動方向と同じ方向へは進まずに後方を通る経路で移動していることから提案手法によるナビゲーションが効率的な経路を生成可能であると言える．



(a) Static approach.



(b) Proposed approach with C_d .

Fig. 3.9: Trajectory comparison of static approach and proposed approach in case of $\theta = 135^\circ$. The human starts at point $(-2.0, -0.5)$. Trajectories are colored to visualize the time dimension.

3.4.1 問題点

ロボットと人間の軌道の全交差角度のパターンにおいて実行時間の変化を人間の移動開始地点の x 座標が-5.0 の試行から順に見ると、ある地点で突然実行時間が大きく変化していることが確認できる．Fig. 3.4(a) より、 $\theta = 90^\circ$ の場合の人間の移動開始地点の x 座標が-2.5 の試行の軌道を Fig. 3.10 に示す．ロボットが Prediction layer によるコストが分布する領域に接近した際に、人間との十分な距離があったにもかかわらず減速し、人間が通り過ぎるまで速度を落とした状態であったことが分かる．これは、Prediction layer によって予測された将来の衝突危険領域を認識し、本来ならばその領域を避けるための経路を生成するべきであるが、 C_d の影響を強く受けることによって、人間の進行方向と逆の経路が優先され Prediction layer による危険領域の後方を通る経路を選択しても速度を上げることができず、また Prediction layer による危険領域の位置はロボットの進行方向によっても変化するため、結果として人間が通り過ぎてから再び速度を上げ始めるといった経路が生成されている．

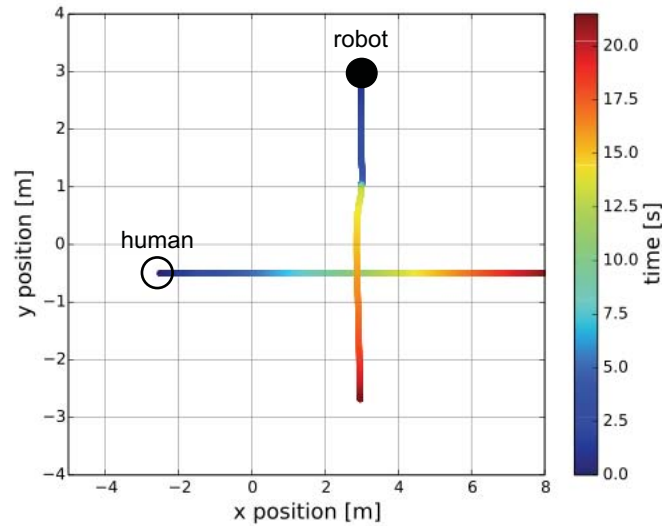
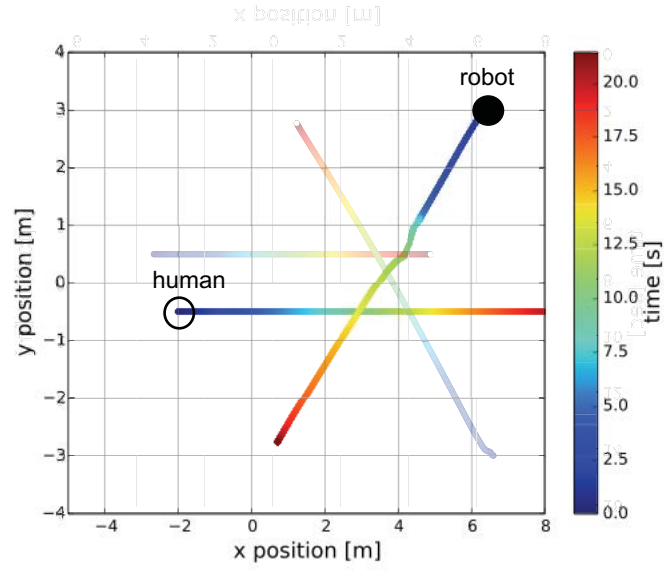
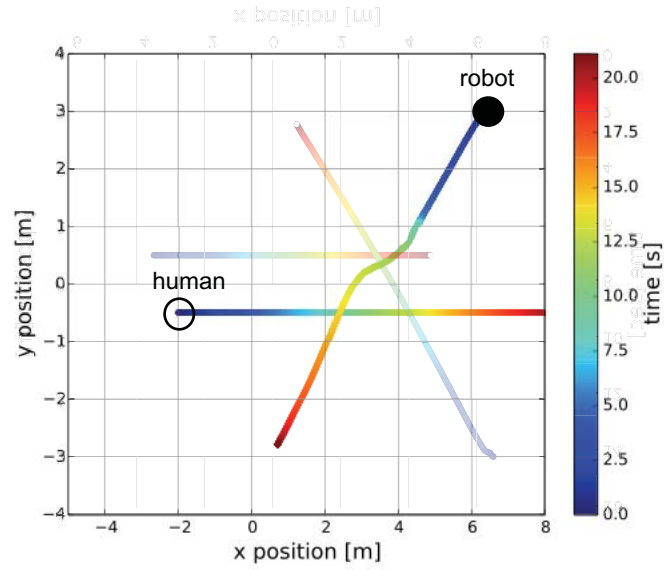


Fig. 3.10: Trajectory generated by proposed approach in case of $\theta = 90^\circ$. The human starts at point $(-2.5, -0.5)$. Trajectory is colored to visualize the time dimension.

加えて、 $\theta = 135^\circ$ の場合の提案手法によるナビゲーションの結果が C_d を用いる場合と用いない場合で変化が見られなかったことが問題点として挙げられる．2つの間に変化が見られなかった試行の軌道を時間経過によって色付けしたものを Fig. 3.11 に示す． C_d を用いる場合と用いない場合共に，Prediction layer による危険領域に接近すると減速し，人間の移動方向に対して後方を通る経路で衝突を回避している．本来 C_d を用いない場合は，軌道評価時に C_d の影響がないため，危険領域に接近した際に減速しても C_d を用いる場合に比べて人間の前方を通る経路を生成する．この問題の原因として，式 (2.2.1) の局所的経路計画に用いる軌道評価関数における各要素のスケールパラメータの影響によるものだと考えられる．これは，最初に述べた提案手法によるナビゲーションにおいてある試行で突然実行時間が大きく変化するという問題の原因でもあると言える．



(a) Proposed approach without C_d .



(b) Proposed approach with C_d .

Fig. 3.11: Trajectory comparison of static approach and proposed approach in case of $\theta = 135^\circ$. The human starts at point $(-2.0, -0.5)$. Trajectories are colored to visualize the time dimension.

パラメータ調整によって、軌道評価に用いる各要素の影響を変化させることができ、それぞれの要素のスケーリングのバランスによってロボットの行動は大きく変化する．今回、実験で用いた式 (2.2.1) の各パラメータの値は、それぞれ $scale_1 = 0.2$, $scale_2 = 0.4$, $scale_3 = 0.1$, $scale_4 = 1.0$ であるが、Prediction layer による危険領域や C_d の影響を左右するパラメータ $scale_3$, $scale_4$ の値を変更することで、危険領域に接近しても減速もしくは停止し続けるのではなく、ある程度柔軟な経路を生成することも可能となる．しかしこの場合、ロボットの移動効率は向上するが、目的地に到達することを優先してしまい人間との十分な距離を保てず衝突危険性が高くなる可能性がある．サービスロボットのように人間と密接に関わる環境で動作するロボットは、効率よりも人間の安全を最優先すべきである．

本研究において、ロボットは人間を車両や他のロボットのような動的物体と同様の障害物として処理を行っているが、人間の安全を第一に考えるならば、人間と他の動的物体は別の物体として扱うべきである．提案手法における衝突回避はロボットにとって人間などの障害物が危険であるかどうかの評価要素のみ用いて経路計画を行っているため、生成する経路が人間にとって安全であるかを評価する要素も局所的経路計画における軌道評価関数に加えることで、より人間の安全を確保した経路の生成が可能になると考える．また、将来の衝突可能性を予測するにあたり本研究では人間の行動は等速運動とみなして衝突危険領域を予測しているが、複雑な人間の行動をより詳細に表現するために Social Force Model (SFM) [19] などのモデルを予測に用いることで、安全の確保だけでなく人間の行動特性を考慮した経路を生成することができ、移動効率の向上にもつながると考えられる．

第 4 章 おわりに

4.1 本論文のまとめ

本論文では，人間の安全とロボットの移動効率を両立する動的環境下でのナビゲーションを実現するため，衝突回避を担う局所的経路計画における人間の動きを考慮した新たな経路計画手法を提案した．提案手法では，ロボットが危険を事前に認識し衝突を回避する経路を生成できるようにするために，経路計画で障害物回避のために用いる階層的コストマップを用いて，将来ロボットと人間との間で衝突が起こりうる領域の情報も障害物情報として与えた．衝突可能性のある領域は，現在のロボットと人間の位置と速度を含む状態から将来最も接近する時刻を推定し，その時刻における人間の位置を中心に人間が存在する確率をコストマップの各セルに障害物コストとして格納する．さらにロボットの移動効率向上のため，経路生成時の軌道評価関数に障害物情報だけでなく，人間の移動方向の情報を扱う要素を追加した．予測された衝突可能性のある領域における人間の移動方向の情報を，障害物コストを格納するコストマップとは別のコストマップとして新たに定義し，その情報を元にして軌道評価関数において，人間の移動方向によって軌道の優先度を決定する．

シミュレータ実験ではロボットと人間が交差する際の軌道の交差角度によって 3 つのパターンに分けてシミュレーションを行い，ROS で提供されている従来のナビゲーションと提案手法によるナビゲーションを比較し性能を評価した．また，提案手法において Direction cost を用いて軌道評価を行う場合と用いない場合の比較も行った．評価指標にはロボットが目的地に到達するまでの実行時間と，ナビゲーション中に人間と最も接近した際の距離を用いた．結果として，提案手法によるナビゲーションでは Direction cost を用いる場合と用いない場合共に，従来の ROS ナビゲーションでは衝突が発生した試行を含む全ての試行において衝突を回避することができたことから，提案した局所的経路計画手法の安全性を確認することができた．また，提案手法によるナビゲーションでは生成される経路は人間が通り過ぎるまで減速するもしくは，人間の後ろを通るような経路であったことから，提案手法が人間の移動方向を考慮した経路を生成可能にしていることがわかった．

4.2 今後の課題

今後の課題として、提案手法における軌道評価関数の見直しが挙げられる。シミュレータ実験におけるいくつかの試行において、実行時間が大きく変化してしまうことがあった。また、Direction cost を用いる場合と用いない場合のナビゲーションによるロボットの行動に変化が見られなかった。これは、局所的経路計画に用いる軌道評価関数における各要素のスケールパラメータの影響が原因であると考えられる。これらの問題はパラメータを調整することで改善することが可能であるが、人間の安全を確保できなくなる可能性が生じてしまう。そのため、軌道評価関数にその軌道が人間にとって安全であるかどうかを評価する要素を加えることで、パラメータの影響だけではなく、本質的に人間にとって安全な経路を生成する必要がある。また、衝突危険領域の予測に用いる人間の行動モデルに SFM などのより複雑な行動を表現可能なモデルを用いることで移動効率を向上させることができ、問題の改善、解決につながると考える。

本論文では環境内に単独の人間しか存在していない状況でのみ評価しているため、より実環境に近い不特定多数の人間が環境内を移動している状況において提案手法の安全や効率が保証されるかを検証し、シミュレータ上だけでなく実環境において実機を用いた検証実験を行うことで、提案した局所的経路計画を実環境におけるナビゲーションへの利用に発展させることができると考える。

謝辞

本研究は、奈良先端科学技術大学院大学 情報科学研究 情報科学専攻 ロボティクス研究室 小笠原司教授のご指導の下で行いました。本研究の遂行にあたり、多岐にわたるご指導やご助言を賜りました小笠原司教授に深く感謝いたします。

論文執筆にあたり適切な御教示や御助言を頂きました本学情報科学研究科の萩田紀博教授に深く感謝いたします。

研究方針や論文執筆にあたり、専門分野から様々な御指導や御助言を頂きました同研究科の高松淳准教授に深く感謝いたします。

同研究科の丁明助教には本研究室の研究会などで御指導や御助言を頂きました。深く感謝いたします。

同研究科の Gustavo Alfonso Garcia Ricardez 博士研究員には研究内容のより細かな部分まで多くの御指導や御助言を頂きました。深く感謝いたします。

同研究科秘書の大脇美千代氏、池田美輝氏には研究室での生活において様々な面でサポートをして頂きました。深く感謝いたします。

本研究に対し、厳しくも的確な御意見・御指摘を頂きました本研究室博士後期課程の築地原里樹氏に深く感謝いたします。

本研究に対し、多くの御指導や御助言、的確な御意見・御指摘を頂きました本研究室博士後期課程の Felix von Drigalski 氏に深く感謝いたします。

本研究室博士後期課程の袴田有哉氏、湯口彰重氏にはサービスロボットグループのメンバーとして様々な形でご支援頂きました。深く感謝いたします。

本研究室博士後期課程の山崎亘氏、趙崇貴氏には研究や研究生活について多くのご助言をいただきました。深く感謝いたします。

先輩、同輩ならびに後輩たちには研究に対してのご助言や様々な形での支援を頂きました。深く感謝いたします。

最後に、長い学生生活を支えていただいた両親、家族並びに友人たちに心から感謝いたします。

参考文献

- [1] 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.
- [2] E. Broadbent, R. Stafford, and B. MacDonald, “Acceptance of Healthcare Robots for the Older Population: Review and Future Directions,” *International Journal of Social Robotics*, vol. 1, no. 4, pp. 319–330, 2009.
- [3] L. Marchionni, J. Pages, J. Adell, J. R. Capriles, and H. Tomé, “REEM Service Robot: How May I Help You? ,” in *Proceedings of the International Work-Conference on the Interplay Between Natural and Artificial Computation (IWINAC), Natural and Artificial Models in Computation and Biology*, pp. 121–130, 2013.
- [4] M. Shneier and R. Bostelman, “Literature review of mobile robots for manufacturing,” *National Institute of Standards and Technology (NIST), U.S. Department of Commerce*, 2015.
- [5] I. Nishitani, T. Matsumura, M. Ozawa, A. Yorozu, and M. Takahashi, “Human-centered X–Y–T space path planning for mobile robot in dynamic environments,” *Robotics and Autonomous Systems*, vol. 66, no. C, pp. 18–26, 2015.
- [6] M. Kollmitz, K. Hsiao, J. Gaa, and W. Burgard, “Time dependent planning on a layered social cost map for human-aware robot navigation,” in *Proceedings of the European Conference on Mobile Robots (ECMR)*, pp. 1–6, 2015.
- [7] N. Pérez-Higueras, R. Ramón-Vigo, F. Caballero, and L. Merino, “Robot local navigation with learned social cost functions,” in *Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, vol. 2, pp. 618–625, 2014.
- [8] H. Moravec, “Sensor Fusion in Certainty Grids for Mobile Robots,” *AI Magazine*, vol. 9, no. 2, pp. 61–74, 1988.

- [9] S. M. LaValle and J. J. Kuffner, “Randomized Kinodynamic Planning,” *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.
- [10] P. E. Hart, N. J. Nilsson, and B. Raphael, “A Formal Basis for the Heuristic Determination of Minimum Cost Paths,” *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
- [11] T. J. Misa and P. L. Frana, “An Interview with Edsger W. Dijkstra,” *Communications of the ACM*, vol. 53, no. 8, pp. 41–47, 2010.
- [12] A. Stentz, “The Focussed D* Algorithm for Real-time Replanning,” in *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1652–1659, 1995.
- [13] S. Koenig and M. Likhachev, “Fast replanning for navigation in unknown terrain,” *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 354–363, 2005.
- [14] B. P. Gerkey and K. Konolige, “Planning and control in unstructured terrain,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Workshop on Path Planning on Costmaps*, 2008.
- [15] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” *IEEE Robotics Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.
- [16] D. Zhang, Z. Xie, P. Li, J. Yu, and X. Chen, “Real-time navigation in dynamic human environments using optimal reciprocal collision avoidance,” in *Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 2232–2237, 2015.
- [17] H. Hisahara, Y. Ishii, M. Ota, T. Ogitsu, H. Takemura, and H. Mizoguchi, “Human Avoidance Function for Robotic Vacuum Cleaner through Use of Environmental Sensors: Roomba; Making Way for Humans,” in *Proceedings of the 5th International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, pp. 64–67, 2014.
- [18] J. Guzzi, A. Giusti, L. M. Gambardella, G. Theraulaz, and G. A. D. Caro, “Human-friendly robot navigation in dynamic environments,” in *Proceed-*

- ings of the *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 423–430, 2013.
- [19] D. Helbing and P. Molnár, “Social force model for pedestrian dynamics,” *Physical Review E*, vol. 51, no. 5, pp. 4282–4286, 1995.
 - [20] S. Oli, B. L’Esperance, and K. Gupta, “Human Motion Behaviour Aware Planner (HMBAP) for path planning in dynamic human environments,” in *Proceedings of the 16th International Conference on Advanced Robotics (ICAR)*, pp. 1–7, 2013.
 - [21] B. Okal and K. O. Arras, “Learning socially normative robot navigation behaviors with bayesian inverse reinforcement learning,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2889–2895, May 2016.
 - [22] D. V. Lu, D. Hershberger, and W. D. Smart, “Layered costmaps for context-sensitive navigation,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 709–715, 2014.
 - [23] X. Han, Y. Leng, H. Luo, and W. Zhou, “A novel navigation scheme in dynamic environment using layered costmap,” in *Proceedings of the 29th Chinese Control And Decision Conference (CCDC)*, pp. 7123–7128, 2017.
 - [24] G. A. G. Ricardez, A. Yamaguchi, J. Takamatsu, and T. Ogasawara, “Asymmetric Velocity Moderation for human-safe robot control,” *Advanced Robotics*, vol. 29, no. 17, pp. 1111–1125, 2015.
 - [25] “ROS Indigo Igloo,” <http://wiki.ros.org/indigo>.
 - [26] “Gazebo,” <http://gazebo.org/>.