

2-D/3-DOF シームレス大域-局所的動作計画 ROS パッケージの開発

パッケージの紹介と石油プラント点検ロボットへの適用試験結果

○渡辺 敦志 (東北大学) 遠藤 大輔 (東北大学) 山内 元貴 (東北大学) 永谷 圭司 (東北大学)

Abstract— This paper introduces an algorithm and implementation of a new navigation package for ROS. The package realizes a 2-D/3-DOF seamless global-local planner. In this package, collision avoidance frequency can be faster than the conventional one and planned path is always taking global goal into account. The new navigation package is released as an open source software on GitHub. In this paper, the field test results in the representative offshore oil plant, which has narrow corridors with projecting pieces of equipment are shown.

1. はじめに

ROS (Robot Operating System) を用いて移動ロボットの自律走行を実現するため、Willow Garage 社によって開発された、navigation メタパッケージがよく用いられる。navigation メタパッケージは、屋内のオフィスにおける長時間、長距離のナビゲーションが可能であることが報告されている [1]。

navigation メタパッケージで提供される経路計画アルゴリズムは、大域的動作計画として主にダイクストラ法、局所的動作計画として主に Dynamic Window Approach [2] および Trajectory Rollout [3] をベースとしている。また、経路計画における地図として、階層化された 2D のコストマップの枠組みが提供されている。

産業用のプラントや乱雑な屋内環境など、狭く複雑な環境においては、navigation メタパッケージを用いた自律走行には、下記に示すような問題が発生する。

- 大域的動作計画と局所的動作計画が独立に動作
 - 局所的動作計画の範囲内でのみ障害物を迂回
 - 移動に伴い大域的動作がチャタリング
- リカバリ動作内で動作計画系の機能が利用不可
 - リカバリ動作が周辺環境の情報を利用しづらい構成のため、狭い複雑な環境で衝突を引き起こす

そこで筆者らは、これらの問題を解決するため、2-D/3-DOF シームレス大域-局所的動作計画 ROS パッケージを開発している。このパッケージは、GitHub 上でオープンソースソフトウェアとして公開している*1

他方で、筆者らは海洋石油プラントの自律点検ロボットシステムを開発する国際コンペティション ARGOS (Autonomous Robot for Gas and Oil Sites) Challenge [4] に参加している。このコンペティションは、総合石油会社の Total が主催し、Total が消防訓練等に用いている模擬プラント上で行われる。この模擬プラントは実プラントと同様の設備を持つ狭く複雑な環境であり、これを通

*1 <https://github.com/at-wat/neonavigation/>

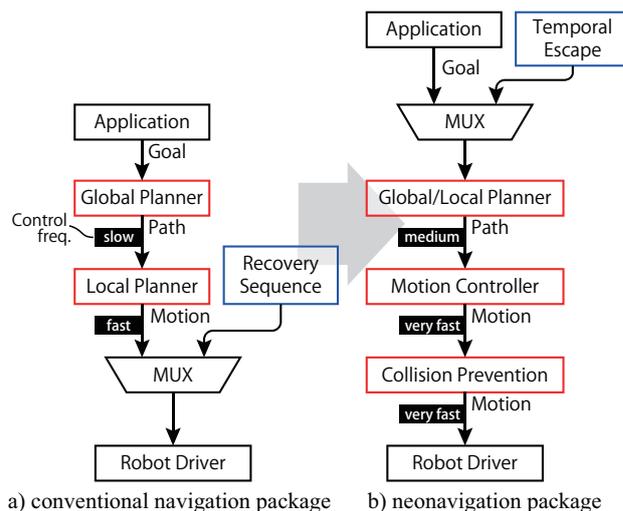


Fig. 1 System structures of the conventional navigation meta-package and the new navigation meta-package, “neonavigation”.

り抜けられる自律移動システムが要求される。

本稿では、筆者らが開発している新しいナビゲーションメタパッケージのアルゴリズムと実装について説明する。また、本パッケージを ARGOS challenge において適用した、フィールド試験の結果を示す。

2. 制御システムの構成と戦略

Fig. 1 に、従来の navigation メタパッケージと、提案する新しいナビゲーション用パッケージ“neonavigation”メタパッケージのシステム構成の差異を示す。

2.1 従来の navigation メタパッケージ

従来の navigation メタパッケージを用いた自律移動系は、move_base パッケージが提供する枠組み内で、大域的動作計画と局所的動作計画が独立の実行周期で動作する。大域的動作計画と局所的動作計画の動作周期はそれぞれ、1 Hz、10 Hz オーダーの値とすることが多い。

このシステム構造において、局所的動作計画は、ロボットのキネマティクスを考慮した動作計画と、障害物との衝突回避・防止の 2 つの機能を持つ。衝突回避の役割を持つため、局所的動作計画は、ロボットの速度に応

じて十分に早い周期で実行される必要がある。

move_base の枠組みでは、局所的動作計画は、大域的なゴールの情報を持たないため、大域的動作計画が意図する経路と異なる動作を生成し、結果として経路のチャタリングを引き起こす場合がある。

ロボットがスタックした際のリカバリ動作の枠組みが提供されるが、大域・局所的動作計画とは独立に動作してロボットの動作を出力する構成を持つ。そのため、リカバリ動作は周辺環境等を考慮しない動作となり、複雑で狭い環境では容易に衝突を引き起こす。

2.2 neonavigation メタパッケージ

開発した neonavigation メタパッケージでは、ロボットは文献 [5] の経路追従手法を用いて、経路計画アルゴリズムが生成した経路を追従する。また、衝突防止の機能は経路計画からは分離されているため、経路計画とは独立により高速な周期で実行できる。例えば、大域的・局所的動作計画は 5 Hz 程度で動作させ、経路追従制御と衝突防止は 50 Hz 程度で動作させることが可能である。

また、大域的な動作計画と局所的な動作計画はシームレスに同時に行われ、大域的動作計画が常に考慮された動作計画を実現する。

neonavigation メタパッケージにおいては、リカバリ動作は、経路計画系に与えるゴールを一時的に書き換える構成となっているため、リカバリ動作中も常に、障害物への衝突回避・防止が働く状態が維持される。

3. neonavigation メタパッケージ

開発した neonavigation メタパッケージには、後述の 5 つのパッケージが含まれる。自律移動には、これらのうち、4 つのパッケージを用いる。neonavigation_launch パッケージは、他のパッケージを組み合わせるための使用例を launch ファイルとして提供している。

Fig. 2 に、これらのパッケージを用いた、自律移動系のデータフローとシステム構成を示す。図中のそれぞれのノードは以下に示すように動作する。

3.1 costmap_3d

costmap_3d ノードは、従来から用いられている 2-D の占有格子地図から、ロボットの外形を用いて、Fig. 3 に示すような 2-D/3-DOF コンフィギュレーション空間の占有格子地図を作成する。図中では高さ軸に Yaw 方向の次元を割り当て、占有されているグリッドをポリゴンとして表示しており、ロボットの姿勢によって壁に接近できる距離が変化していることが確認できる。

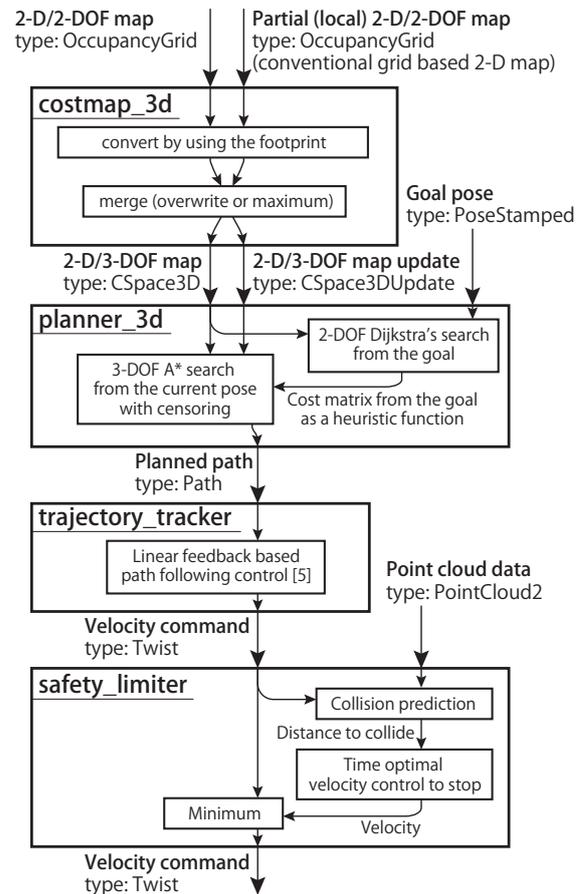


Fig. 2 Data flow diagram of the navigation using neonavigation meta-package.

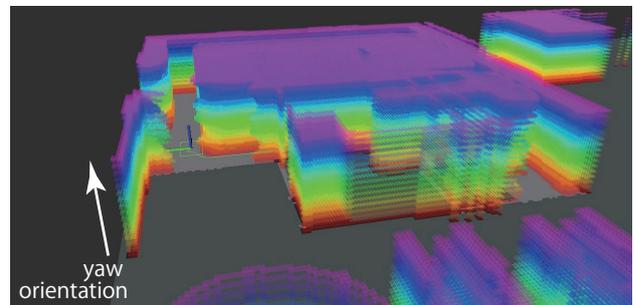


Fig. 3 Visualized 3-DOF costmap in (x, y, yaw) configuration space.

3.2 planner_3d

planner_3d ノードは、ヒューリスティクス関数に、ゴールから逆方向に探索した各地点への最短移動コストを用いた、グリッドベースの A* アルゴリズムにより、主にノンホロミックな移動体用の経路計画を行う。このとき、ゴールから各地点への最短移動コストは、あらかじめダイクストラ法により計算することができ、環境変化またはゴールが変化した際、必要な範囲のみ再計算することで高速な繰り返し計算が可能となる。動作計画は常に大域的に行い、探索を設定された距離で打ち切ることで、計算量を抑える。この距離が、従来のナビゲーター

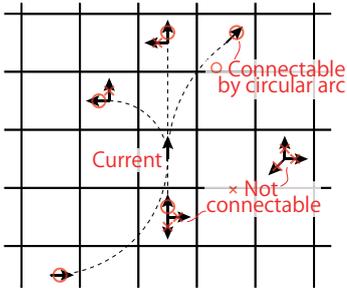


Fig. 4 Grid search with kinematic constraints.

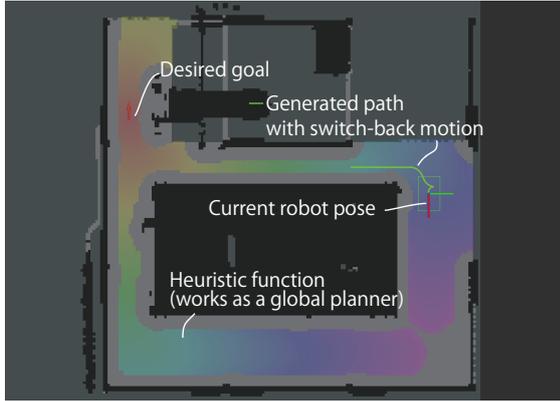


Fig. 5 Grid search with kinematic constraints.

ションで用いられる、局所的動作計画の探索範囲に相当する。

2-D/3-DOF 占有格子地図中での状態遷移は、周辺の設定された x-y 空間距離内の全ての yaw 方位について探索される。ただし、Fig. 4 に示すように、現在の状態から次の状態を円弧で結ぶことができない場合には、ノンホロミックな系では到達できないため、探索を行わない。

また、直前の動作計画で得られた経路の付近を選択するとコストが下がるコスト関数を導入しており、得られる経路にヒステリシスを持たせることができる。

Fig. 5 に例を示すように、狭い場所などでは切り返しを含む経路を得ることができる。planner_3d の実装には、CPU のキャッシュヒット率を向上するようにデータのメモリ配置を工夫しており、Core i7 プロセッサの 1 コアを 50% 程度使用して、約 5 Hz の周期で動作計画を実行できる。

より複雑な環境では、動作の計画が難しくなり、planner_3d の実行周期が遅くなる場合がある。その際には、ウォッチドッグタイマーを用いて、動作計画が完了するまで動作を停止させることで安全を確保する。

3.3 safety_limiter

safety_limiter ノードは、ロボットが現在の速度・角速度指令値で走行したときの衝突を予測し、衝突の前に停止するように指令値を制限する。予測は、Fig. 6 に示すように、現在の状態と指令値から線形外挿により求める。

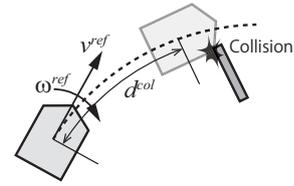


Fig. 6 Controlled variables of the collision prevention.

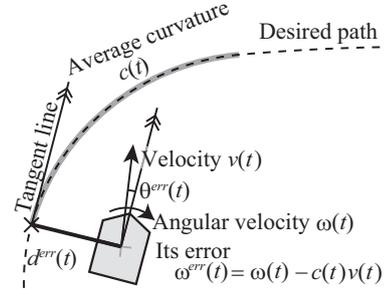


Fig. 7 Controlled variables of the path following control based on [5].

指令値の制限は、予測される衝突までの距離 d^{col} と、距離のマージン d^{margin} から、次式に示す、最短時間制御により行う。

$$v^{lim} = \text{sgn}(d^{col} - d^{margin}) \text{sgn}(v^{ref}) \sqrt{2\alpha |d^{col} - d^{margin}|}$$

$$\omega^{lim} = \omega \frac{v^{lim}}{v^{ref}} \quad (1)$$

ここで、 α は、ロボットが許容する並進加速度である。

safety_limiter は常に、入力された速度・角速度指令値と、出力の旋回半径を同一にするように制御を行う。したがって、safety_limiter が用いられても、ロボットの動作は、動作計画のノードが意図した経路から外れにくい。

3.4 trajectory_tracker

trajectory_tracker ノードは、文献 [5] で提案された移動ロボットの直線追従制御系を自由曲線に拡張した経路追従制御 [6] を行う。曲線に対する距離、角度、角速度偏差 (Fig. 7) をロボットの角加速度に線形フィードバックすることで、曲線上を曲線に沿って走行する。

4. 模擬海上石油プラントにおけるフィールド試験

開発した neonavigation メタパッケージの有効性を、ARGOS Challenge のフィールドにおいて確認した。以下の試験は、2016 年 4 月 4 日から 8 日に開催された、ARGOS Challenge 2nd コンペティションにおいて実施した。試験環境は、フランスの Total 社の保有する訓練用の模擬海上石油プラント “UMAD” を用いた。

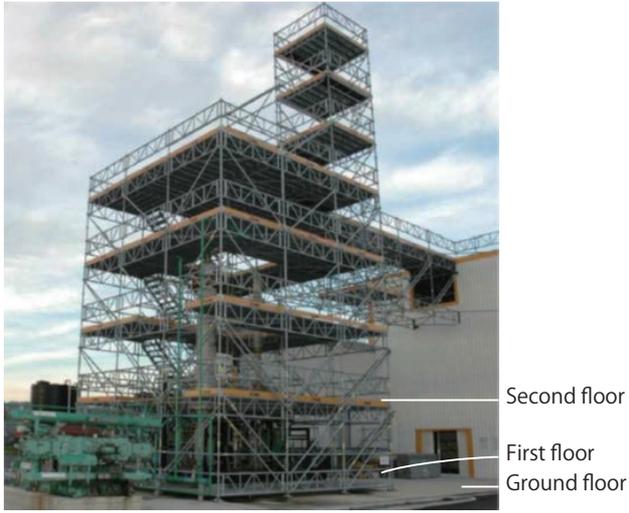


Fig. 8 The representative offshore oil plant “UMAD” in the ARGOS (Autonomous Robot for Gas and Oil Sites) Challenge competition held by Total.

4.1 試験のシナリオ

Fig. 8 に、UMAD の外形を示す。試験を行ったコンペティションでは、地上階、1 階、2 階を自律移動し、設置された圧力計、バルブ、水位計の値、機器の温度の測定が要求される。全ての動作は自律で行われ、ガス漏れや計器異常などを検出した場合のみ、オペレータによる動作指示が許されている。いくつかのミッションでは、ネットワークの遮断、非常警報、ガス漏れ、高温の熱源が現れる場合がある。

Table 1 に、与えられたミッションの概要を示す。ミッションの指示は、試験開始 2 時間前に印刷物として渡され、筆者らがそれを実現するための目的地や計測対象を記述したスクリプトを作成した。

Table 1 Summary of the missions

mission	start	goal	points to be inspected	special
#1	Starting area	Starting area	Measure the pressure gauge CP1, and the valve CP14 on the second floor	
#3	Safe area	Starting area	Visit VP1, VP2, and VP3	An unknown obstacle exists in the UMAD
#4	Starting area	Starting area	Look around and find heat source	
#5	Starting area	Starting area	Measure the valve CP5 and the pressure gauge CP7, CP1, and CP3	
#6	Starting area	Starting area	Measure the valve CP4 and the water level gauge CP9	Several abnormal state and additional mission will be given during the mission

* #2 was a free mission to be defined by each participant.

4.2 neonavigation メタパッケージを用いたシステム構成

Fig. 9 に、ARGOS challenge2nd コンペティションで用いたロボットシステム “AIR-K 1.5” のハードウェア構成を示す。AIR-K 1.5 は、2+4-DOF クローラ移動ロボット “kenaf” [7] をベースとして構築した。

AIR-K 1.5 は、2 台の 3-D 測域センサ 北陽電機 YVT-X002 [8] および IMU を自律走行のために、4 台のカメラ、熱画像カメラ、超音波マイクを計器の読み取り、非常事態の認識のために搭載した。ロボットは無線 LAN でオペレータールームと接続されており、ミッションのレポートとロボットの状態を送信し、ミッションの更新などを受け取る。

Fig. 10 に、AIR-K 1.5 のソフトウェア構成を示す。mission_manager ノードはシステム全体を統括し、与えられたミッションに基づいて、動作計画ノードに指令を与える。planner_3d の他に、段差や階段を踏破するための step_traverse、計器を測定するために姿勢を変更する cv_measure が動作計画ノードとして動作し、ミッションの状況に応じてその出力を選択してロボットのドライバに与える。3-D 測域センサのポイントクラウドは hokuyo3d ノードから取得し、travelable_area ノードにより、2-D 占有格子地図に変換され、costmap_3d ノードに与えられる。静的な地図は、環境の階毎に保持され、select_map ノードにより現在の階の地図が costmap_3d ノードに与えられる。

自己位置推定には、3-D のポイントクラウドから生成した 2-D のデータを用い、従来の navigation メタパッケージが提供する amcl ノードを用いた。

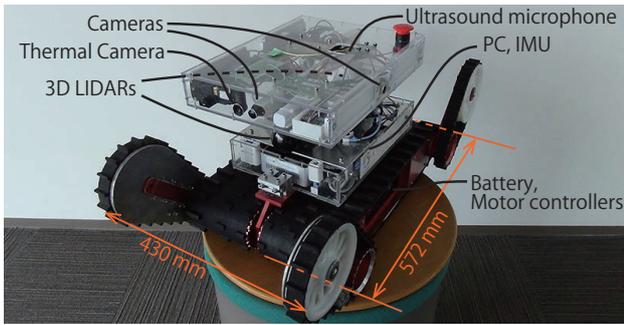


Fig. 9 The robot system, called AIR-K 1.5, for the test in the ARGOS challenge, based on the 6-crawler mobile robot “kenaf” [7]

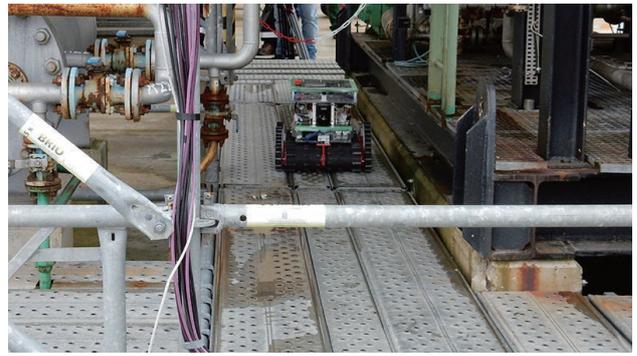


Fig. 11 Picture of the robot on UMAD

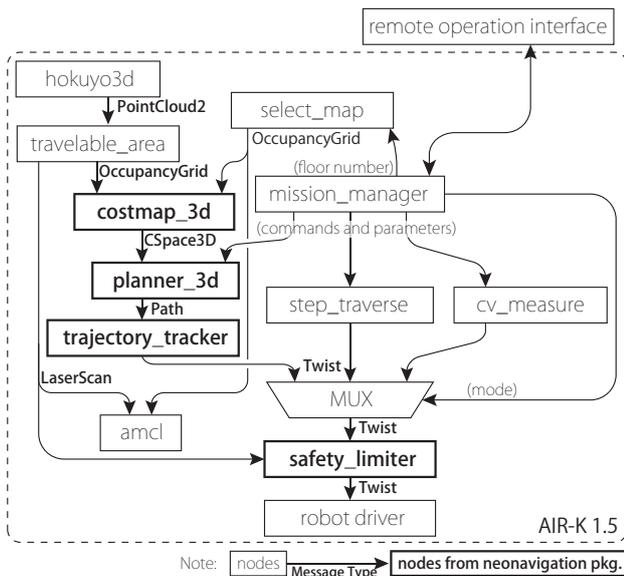


Fig. 10 Software system structure of the AIR-K 1.5 accepting measurement system

4.3 フィールド試験結果

試験では、ARGOS Challenge で与えられたミッション#1, #3, #4, #5, #6 を実行した。Fig. 11 に、UMAD 上で動作するロボットの概観を示す。床は突起と穴のある仮設足場で構築されており、ロボットの動作時に大きな外乱を発生させる。

Fig. 12-14 に、自律移動の結果の一部を示す。狭い通路においても、障害物に衝突することなく、動作していることが確認できる。

以下に、これらのミッションを実行した9回の試行の結果を示す。neavigation メタパッケージに起因する問題を下線で示す。

mission #1 try-1 (中断)

- 1) 自己位置推定の失敗 (スタート地点から再開)

mission #1 try-2 (成功)

mission #3 (成功)

- 1) 狭い通路を通り抜ける際、前後に振動 (自律的

に解決)

- 2) 環境に軽度の接触 (続行)

- 3) 自己位置推定の失敗 (手動で修正し続行)

mission #4 try-1 (成功)

mission #5 try-1 (中断)

- 1) mission_manager のバグにより、階段踏破後にスタック (スタート地点から再開)

mission #5 try-2 (成功)

- 1) サブクローラの脱落 (続行)

mission #4 try-2 (中断)

- 1) サブクローラの脱落 (スタート地点から再開)

mission #4 try-3 (成功)

- 1) 環境に軽度の接触 (続行)

mission #6 (中断)

- 1) 自己位置推定の失敗 (手動で修正し続行)
- 2) サブクローラの脱落 (続行)
- 3) バグにより階段降坂失敗 (中断)

5. 考察

試験の結果から、開発した neavigation メタパッケージは、狭い通路をもつプラント環境で適切に動作することが確認できた。今回の試験では、従来の navigation メタパッケージとの定量的な比較は行っておらず、今後の課題として残されている。

試験においては、致命的な問題は主に、自己位置推定の失敗とハードウェアの異常により引き起こされた。環境中の通路の端は手すりや鉄管で構築されているため、3-D ポイントクラウドから生成した 2-D のデータを用いた自己位置推定を用いたことで、誤差が大きくなったと考えられる。3-D の自己位置推定を用いることで、改善が可能と考えられる。

一方、neavigation メタパッケージにも、軽微な問題が複数あることが明らかになった。いくつかの状況では、狭い通路を通過する際に、経路が振動的になること

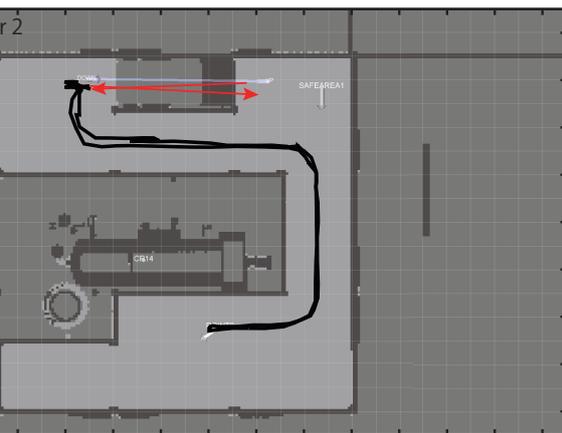
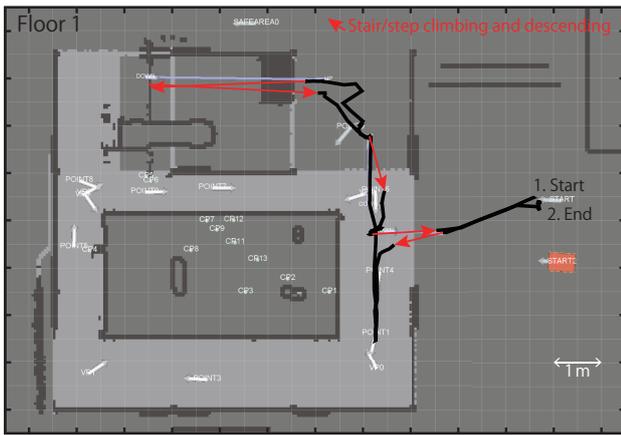


Fig. 12 Navigation result of the mission #1 (try-2)

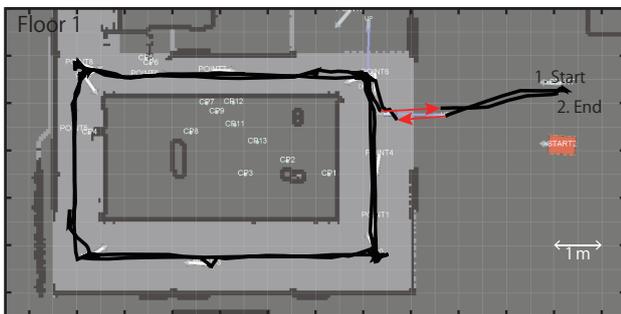


Fig. 13 Navigation result of the mission #4 (try-3)

があった。これは、移動に伴う視点の変化で障害物の見え方が変化したことに起因すると考えられる。また、環境への軽度の接触が発生したが、これは、センシングの精度の限界と、床面から受ける外乱の影響によると考えられる。

6. おわりに

本稿では、筆者らが開発している、ROS用の新しいナビゲーションメタパッケージのアルゴリズムと実装について説明した。本メタパッケージは、狭く複雑な環境で



Fig. 14 Navigation result of the mission #5 (try-2)

の自律移動を可能にする、2-D/3-DOFのシームレス大域-局所的動作計画を実現した。このパッケージは、GitHub上でオープンソースソフトウェアとして公開している。

また、本パッケージを ARGOS challenge において適用した、フィールド試験の結果を示した。試験の結果から、開発した *neonavigation* メタパッケージは、狭い通路をもつプラント環境で適切に動作することが確認できた。今回の試験では、従来の *navigation* メタパッケージとの定量的な比較は行っておらず、今後の課題として残されている。

参考文献

- [1] E. Marder-Eppstein, et al.: “The office marathon: Robust navigation in an indoor office environment”, IEEE ICRA 2010, 2010.
- [2] 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.
- [3] B. P. Gerkey and K. Konolige: “Planning and control in unstructured terrain”, IEEE ICRA 2008, Workshop on Path Planning on Costmaps, 2008.
- [4] K. Kydd, et al.: “Autonomous robot for gas and oil sites”, SPE Offshore Europe Conference and Exhibition Society of Petroleum Engineers, 2015.
- [5] S. Iida and S. Yuta: “Vehicle command system and trajectory control for autonomous mobile robots”, IEEE/RSJ IROS '91, Intelligence for Mechanical Systems, pp. 212–217 vol.1, 1991.
- [6] Y. Morales, et al.: “Including human factors for planning comfortable paths”, IEEE ICRA 2015, pp. 6153–6159, 2015.
- [7] T. Yoshida, et al.: “A high mobility 6-crawler mobile robot 'kenaf' ”, 4th International Workshop on Synthetic Simulation and Robotics to Mitigate Earthquake Disaster (SRMED2007), p. 38, 2007.
- [8] K. Kimoto, et al.: “Development of small size 3d lidar”, IEEE ICRA 2014 IEEE, pp. 4620–4626, 2014.