RBProceedings 文書クラス サンプル文書

佐藤 ** ¹, 鈴木 ** ¹, 高橋 ** ², 田中 ** ³ 伊藤 ** ^{1,3}, 渡辺 ** ^{1,4}

 1 ○○大学, 2 △△大学, 3 ××株式会社, 4 □□研究所

概要

概要の例文. 概要の例文.

1 はじめに

近年, 労働人口の減少や働き方改革により現場 での作業の効率化が求められている. こうした背景 でロボットには工場や倉庫での運搬作業やビルな どの警備、屋外では工事現場の清掃や農薬の散布な どの作業が自動化されることが期待されている.特 に, [1][2] のような AGV や AMR は大量生産を行う 工場や大規模な物流倉庫での運搬の自動化を進め, [3][4] のような警備ロボットでは実際のビルの警備 を少ない人数で運用することに成功している. これ らのロボットでは、[5][6]のような手法が用いられ、 場所は限定されているが今日でも現場の効率化に絶 大な効果を与えている. しかし、発電所の除草作 業や工事現場の運搬作業など,屋外の現場において は大半が自動化されておらず、ラジコン操作や遠隔 映像による操作のロボットで実証実験を行うところ までしか進んでいないという現状がある. 以上から まだまだロボットによる仕事の効率化や協業、代替 に対しては発展途上であると言える. CuboRex で は、一輪車を電動化した "E-cat kit2" や走破性の高 いクローラユニット "CuGo" シリーズをリリース し、"現場のツラいをロボティクスで改善する"を 掲げ屋外での現場効率化を進める活動をしている. この "CuGo" シリーズにロボット制御ミドルウェ アである "ROS" でのアプリケーション開発ができ る環境を整えることで、[7][8] のようなロボットの 活動範囲を広げることができ, ロボットによる自動 化がより一層発展だろう. 殊に, つくばチャレンジ では、リアルワールドでの自律走行技術レベルを向 上すること目標としていることから、つくばチャレ

ンジでの課題を "CuGo" で実現することが "現場の ツラいをロボティクスで改善する"の近道となると 考えた. 本年度のつくばチャレンジの取り組みで は、"ROS 開発キット CuGo V3"をベースに走行用 ロボットを作成し、GNSS と MAP の位置推定を切り 替えることができるナビゲーションシステムを開発 した. これにより、本年度から追加された衛星測位 を遮る屋根のある区間でも安定して走行することが できるロボットとなった. 本走行では、確認走行 区間を抜けた先のパイロン地帯でパイロンの前に停 止して断念した. 計算資源が足りなかったことと, 設定した経路が未熟であったため、もう少し距離 を伸ばすことができると考えられる. 本年度では, 2DLidar, GNSS, ホイールオドメトリのみを使った シンプルなナビゲーションシステムで幅広い環境で 走行できるシステムを構築することができた.

2 ハードウェア

2.1 ハードウェア概要

本年度の実験に使用したロボットの外観を以下の図1に示す. ハードウェアの構成を表1に示す. つくばチャレンジ走行用ロボットには, CuboRex 製自律走行ロボットの開発用プラットフォームである, "ROS 開発キット CuGo V3"をベースに課題に必要な装備を追加した.

2.2 ROS 開発キット CuGo V3

"ROS 開発キット"は、汎用クローラユニット である CuGo V3 を正確に制御するマイクロコント ローラ、ROS を実行する Linux コンピュータ、それ を駆動するバッテリーや電源システムがオールイン ワンとなったパッケージ製品である. これに必要に 応じて 3D LiDAR や GNSS, カメラなどの装備を加 えることで, 最小限の労力で屋外を走行するクロー ラロボットの開発を開始することができる.

2.3 つくばチャレンジ仕様

本年度のつくばチャレンジでは、"ROS 開発キット"に GNSS、デバッグ用モニター、モニター確認用のひさし、機体拡張のため押しやすい位置に非常停止スイッチを再設置する変更を行った。後述するが、ナビゲーションシステムに使用したセンサは 2DLidar、GNSS、ホイールオドメトリのみという非常にシンプルな構成となっている。計算資源としても Jetson Orin NX というシングルボードコンピュータ 1 台とモータ制御用の Arduino UNO R3 のコントローラで構成されており、比較的安価で製造しやすい屋外走行ロボットとなった。

3 ナビゲーションシステム

3.1 ナビゲーション概要

本年度のつくばチャレンジのナビゲーションシ ステム構成図を図 X に示す. このシステムでは、基 本的にみちびきのセンチメーター級測位補強信号を 使って補正した GNSS の位置を使用して目的地を目 指す. 2023 年のコースから GNSS の信号を遮る屋 根や建物の直下を通るようになったため、対策が必 要である. このエリアを通過するとき、図 X のよう に GNSS の測位精度の悪化を確認できたため、この エリアに限って地図を作成し地図に対して自己位置 推定を行った. GNSS を使用しないエリアを通過し た後、ふたたび GNSS の自己位置推定に切り替えて 走行することにより、大規模な地図を作成すること なく長距離の自律走行を実現した. この章では、以 下の順番でナビゲーションシステムの詳細について 説明する. GNSS による自己位置推定 2D MAP によ る自己位置推定センサフュージョン GNSS と MAP の位置推定の切り替え Waypoint の設定経路計画障 害物回避その他開発支援ツール

3.2 GNSS による自己位置推定

3.3 2D MAP による自己位置推定

ロボット下部に取り付けられた 2DLidar を使ってつくば市庁舎周辺の地図を作成した. 2DLidar は地面から 120mm の位置に取付てあり、周囲 360 度、最大 30m までの距離を得ることができる. SLAMは Cartographer[9]を利用した. 確認走行区間のルートをを 1.5 周するセンサデータを使い、パラメータを調整して地図を作成した. 作成した地図は図 X に示す. 実際のナビゲーションでは、北の屋根があるエリアからピロティを抜けた区間まではこの地図の位置を使用した. (場合によってカットする)地図を作成する際に、ループクローズが発生するが直線の精度が最も高くなるように調整した. この調整をしないと、ループクローズしたときの誤差の勾配がちょうど北エリアで詳細するため、GNSS と地図の位置にずれが発生する.

3.4 センサフュージョン

このナビゲーションシステムでは、図 X のよう にロボットの位置推定にホイールオドメトリ, IMU, GNSS, MAP Localization を入力している. IMU は 9 軸で地磁気を利用して絶対確を取得していたが、市 庁舎北エリアの急速充電器付近で非常に強いノイズ を受け屋根ありエリアでの姿勢角の精度悪化が無 視できないため共分散を無限大に設定していて、実 質使用していない. ホイールオドメトリの値は、 作動二輪モデルから nav msgs Odometry の値を算出 して利用している. GNSS の姿勢は前項の通り、直 前の連続した位置からロボットの向いている方向 を推定し nav msgs Odometry に格納している. MAP Localization は nav2 AMCL を利用している. AMCL の自己位置推定結果は tf での出力は無効化して、 nav msgs Odometry に変換して利用している. つの nav msgs Odometry を robot localization[10] パッ ケージを利用して EKF 処理をし、ロボットの自己 位置としてナビゲーションに利用している.

3.5 GNSS と MAP の位置推定の切り替え

ナビゲーション構成図の Pose Selector が示す部分で上記 3.2 と 3.3 の各自己位置推定をした値を排他的に選択し、EKF に送る. こうすることで、センサフュージョンをする側のノードでは、現在の状態が GNSS であるか MAP であるかを意識することなく機械的に処理することができる. それぞれの状態は次に向かう Waypoint のデータ構造にフラグとして内包しており、Waypoint ごとにどちらの位置を使用するかを記録することで切り替えを実現している.

3.6 Waypont の設定

Waypoint の設定はロボットの自己位置をそのまま記憶するツールを作成した. GNSS と MAP Localization のどちらの状態であっても, 先述の通り, ロボットが認識する自己位置は変わらないため, 現在認識している自己位置をそのままキーボード操作をトリガーに csv ファイルに記憶する. 実験走行では, 当日の午前にロボットをコントローラでマニュアル操作し, 走行させたい経路の 2m おきのWaypoint を記憶させた. 午後に検定を実施し, 作成した Waypoint 列の csv ファイルをナビゲーションシステムに読み込ませて自律走行を行った.

3.7 経路計画

ロボットの自己位置から Waypoint までの走行経路の生成は、ROS 2 標準の Navigation2[11] をそのまま利用した。Waypoint までの経路は A*アルゴリズム [12] で算出し、その経路を Regulated Pure Persuit [13] で追従した。

3.8 障害物回避

3.9 その他開発支援ツール

現場でのロボットのデバッグは、さまざまな条 件の中で何が起きたかを素早く発見し、迅速に修正 することが大切である. ロボットの状態確認とし ては、1 次の目標の Waypoint の ID 2 フラグ管理で正 常でない組み合わせが発生したことの通知 3) セン サや中間ノードからトピックが来なくなった時の通 知、を読み上げツールによってロボットにしゃべら せた. これにより、異常なゴール判定、想定できな かった状態遷移、各種機能の失陥をいち早く知るこ とができ、対策を本走行当日までに適用することが できた. 読み上げツールは OpenJTalk[12] を使用し, 文字列のトピックを受け取るとそのまま読み上げる アプリケーションを作成し、ロボットに搭載された ディスプレイのスピーカーから音声を再生すること で実現した.

4 性能評価

4.1 ハードウェア走行性能

走行ハードウェアとしては、CuboRex 製のクローラユニットである CuGo V3 を使用した. 研究学園前公園の階段の乗降はできないが、必須課題の走行コースで問題になる段差はなかった. 715 の実験走行では、炎天下の中必須課題の走行コースをマニュアル走行で4周したが、ハードウェアでのトラブルはなかった.

4.2 GNSS 位置推定

4.3 2D MAP 位置推定

図 X に示すつくば市庁舎北エリアの区間は AMCL による自己位置推定を使用している。図 X で示した作成した地図では、1) 図 X の位置に商用バンが止まっていた時に著しく位置精度が低下した 2)

図 X の位置で1度だけ点対象の位置に誤マッチングした,ことがあった.いずれも確率は低かったが,あらゆるシーンで走行可能にするには,3D にすることやもっと遠くの地形を見ることなどの工夫が必要であると感じた.

4.4 障害物回避

5 本走行の結果

本走行では,確認走行区間を超えた先のパイロ ン地帯でパイロンを避けずに停止してしまったた め走行を断念した. パイロンを避けきれなかった原 因としては、以下の原因が考えられる. 経路計画の 動作周波数が 1Hz であり、パイロンを回避する経 路が引かれる前にロボットが走行を続けた. もと より、GNSS のみでどこまで走行できるかを主眼と して開発を進めていた時期があり、GNSSと MAP Localization が同居することを想定していなかった. 本番直前にシステムを大幅に肥大化させたことによ り、最適化が進まず計算資源不足に陥ってしまった ため、さまざまな制御周期を下げる対応をした. こ れにより、経路計画の動作周波数が 1Hz となってし まい、回避行動をとる前に障害物を認識する前の行 動を続けてしまった.また、周囲の障害物を避ける コストマップの範囲も大幅に小さく設定したため, ロボットが障害物を発見するのにより障害物に近づ いた状態でないと障害物に気づけない状況になっ ていた. ロボットが他のロボットなどの障害物を回 避する手段として、積極的に回避することをせずに その場でとどまることを選択してしまった. 回避行 動をとっていればとどまることはないが、回避行動 をとることができず、この仕様によってデッドロッ クしてしまった. Waypoint を疎に打ちすぎていた. 確認走行区間では、周囲のロボット以外の障害物は 明確に知ることができており、Waypoint を 1.5m お きに密に設置し走行経路を明確に定義することで 安定した走行を実現していた. しかし, パイロン区

間は日によってパイロンの設置位置が変わるため、 Waypoint の密度を 10m 以上の疎な設定していた. これによりパイロンから積極的に離れる経路にはな らず、繰り返しパイロンに向かって走行する挙動を してしまった. 当日の GNSS の値が最大 0.5m ほど ずれていた. GNSS の値は以前より比較的高精度に 位置推定をすることができるようになったが、衛星 の位置によっては誤差が大きい時間帯が存在する. ロボットが停止してしまった区間は理想的なオー プンスカイな環境ではあったが、もとから設定した Waypoint の位置より 0.5m 程度北にずれた位置を走 行していた. パイロン地帯でも比較的設置されにく い点字ブロックに沿って走行をする経路を設定した が、そこから北にずれた経路を走行していたため積 極的にパイロンが多い経路を走行し、パイロンを避 けきれないケースが発生した. パイロンを避けら れなかった直接的な原因は以上の4点が挙げられる が、確認走行区間の先の対応をしたのが実験走行の 最終日の午後のみで試行回数が圧倒的に少なかっ た. この先のほとんどの区間がオープンスカイな理 想的な環境であり、運用次第では 2DLidar+GNSS+ホ イールオドメトリのみという非常にシンプルな構成 でもっと距離を伸ばせたと考える. この先の区間で しか現れない問題を収集できなかった点が今回の課 題である.

6 結言

本年度のつくばチャレンジの取り組みとして、つくばの市街地で走行できるロボットハードウェアを作成し、GNSSと MAP Localization の位置推定を任意のタイミングで切り替えることができるナビゲーションシステムを作成し検討した. 走行実験では、確認走行区間の走行を実験走行最終日の検定と本走行を連続で達成した.しかし、その先の走行区間の調整や設定まで至れなかったため、パイロン区間で停止した.シンプルな構成で自律走行するシステムを実現したが、計算能力の限界や動的物体の障害物回避などに課題が残る.そのため、多くの現場で走行できるシステムにするためには最適化や計算資源の見直し、より広い範囲を認識することができるセンサ追加など、必要に応じて資源を増設できるシステムになるように開発を続けたい.

文献



図1 図の例

表 1	表の例
7 57	77 (7) 1VIII

日本語 Japanese ほげほげ ふげふげ 英語 English hogehoge fugefuge

7 はじめに

RBProceedings 文書クラスは W3C により策定されている『日本語組版の要件』[?] に準拠することを目指す jlreq クラスをベースにしている。ただし、本文書クラスでは紙面スペースの都合上、多くの余白値をかなり詰めるように設定しており、例えば行間は外国人参政権のようにルビを振れる最小限の余白に設定してある。

論文では、単純なテキストのみならず、しばしば 数式

$$P(B \mid A) = \frac{P(A \mid B)P(B)}{P(A)} \tag{1}$$

や箇条書き

- 第1の項目
- 第2の項目

といった構造も用いられるが、これらもよく知られた文書クラス(例えば jsarticle 等)と同様のシンタックスで利用できる.

8 図表の挿入

図表についても通常の M_{EX} と同じ方法を用いることができる.

8.1 図について

図の挿入は、通常 graphicx パッケージによって行う(図 1). クラスオプションにワークフロー (dvipdfmx 等) を指定していれば、各パッケージを読み込む際に何度も同じオプションを指定する必要はない.

8.2 表について

表の挿入は, \begin{table}...\end{table} 環境を使う(表 1).

9 参考文献

参考文献の参照例.

- 論文誌の参照例 [?]
- 本の参照例 [?]
- ・国際会議の参照例 [?]
- ・技術報告の参照例 [?]
- Web ページの参照例 [?]

10 Writing in English

This paragraph shows an English sample. There is no problem with writing your manuscript in English. If you write in LaTeX, please use the distributed document class with the english option:

\documentclass[

platex,dvipdfmx,english]{rbproceedings}