2022 年度 卒業論文
3 次元測域センサを用いたロボットの自律走行の改善

奈良女子大学 生活環境学部 情報衣環境学科
生活情報通信科学コース 4 回生 新出研究室
19480353 西中裕梨果

2023 年 2 月

目次

概要

ロボットが自律走行する際に、周りの状況を把握し、様々な状況に応じた行動を取り、安全・安定に走行する技術が求められている。これまで我々が実験時に使用しているロボットでは、2 次元測域センサのみの情報をもとに自律走行をさせていた。しかしながら、2 次元測域センサでは坂道を壁と誤って認識する、上り段差を検知できず自律走行を続けようとする、ロボットよりも低い位置にある障害物・下り段差を見落とし転落するといった問題点があった。そこで、ロボットに搭載されていたが、実用化できていなかった 3 次元測域センサによる地形判定を自律走行時に利用できるようにした。具体的には、3 次元測域センサから点群データを得て、そのデータをもとに判定を行い、判定結果をトピックに配信することで、ロボットの自律走行時に用いるノードがそのメッセージを購読できるようにした。これにより、今まで出来ていなかった段差や壁を検知して直前でロボットが停止することと、壁と区別することが困難であった坂道を「坂道である」と検知して自律走行を続けることが可能になった。また、3 次元測域センサから得た点群データがない場合の補完処理や判定基準の見直しも行い、より精度の高い検知ができるようにした。

1　はじめに

近年、本学並びに様々な大学・企業において自律走行ロボットの研究や大会が盛んに行われている。ロボットが自律走行する際には、周りの状況を逐一把握し、様々な環境に応じた適切な行動をロボット自ら考え選択する技術が求められている。我々が目指すロボットの自律走行においても、段差や坂道を検知し、状況に応じた行動をロボット自身が考え、実行することが必要不可欠である。しかしながら、我々が使用しているロボットでは従来、2 次元測域センサのみの情報をもとに自律走行を行っていたため、段差や坂道の検知ができず、段差から転落するなどの危険性があり、目標地への到達が安定して行えなかった。そこで、従来作られていた 3 次元測域センサによる地形判定プログラムの問題点を改善するとともに、実用化できていなかったこのプログラムによる段差や坂道の検知、段差または壁があればロボットは停止する処理をできるようにした。また、センサがデータを十分に得られなかった時は、数値の補完処理を行い、判定できるようにした。さらに、様々な場所でも正確に判定できるよう判定基準の見直しも行った。

2　先行研究

渡辺 [1] は、奈良女子大学の学内において、3 次元測域センサのデータから段差や坂道を判定するところまでを実現していたが、実際の自律走行時にその判定を利用できる段階に達していなかった。また、センサがデータを取れない領域がある場合の処理も行っていなかったため、ロボットの前方に壁や段差があることにより、データを取れない領域がある場合、判定不可となっていた。さらに、その判定基準が学内での実験に特化しており、坂道や段差である箇所が平地であると判定されてしまうなど、他の場所では正確な判定が行えなかった。また、先行研究で用いられたプログラムは、プログラムの保守性も良いとは言えなかった。中之島エクストラチャレンジ (5.2 節参照) の開催場所である ATC(大阪市住之江区) の実験で例で挙げると、図 1 のような下り階段や、図 2 のような上りスロープといった場所では正確な判定ができなかった。

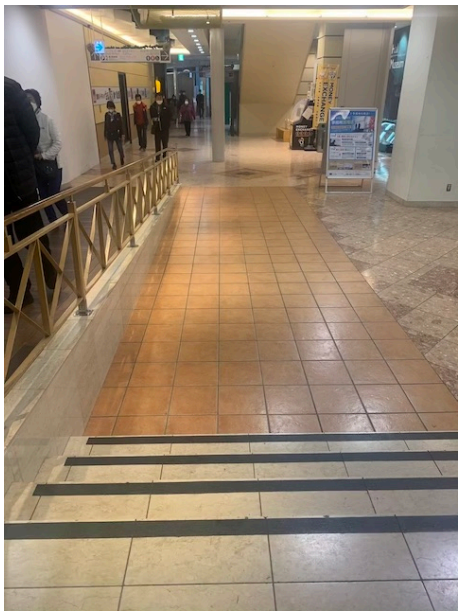

図 1　ATC 内の下り階段

図 2　ATC 周辺の上りスロープ

3　使用したロボットについて

3.1　Arno

本研究では北陽電機制作の自律走行ロボット「Arno」(図 3) を使用した。前方に 3 次元測域センサを 1 台、前方・後方に 2 次元測域センサを 1 台ずつ搭載している。これにより、2 台の 2 次元測域センサで周囲 360°、3 次元測域センサでロボットの進行方向の高低差を認識できる。センサや走行の制御は、ROS を用いて行っ

ている。

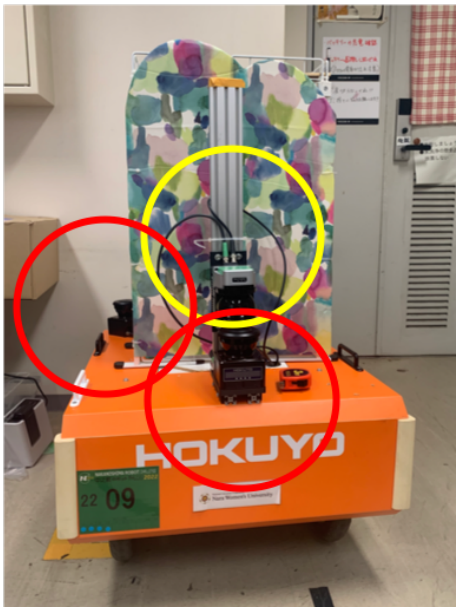

図 3　Arno (赤：2 次元測域センサ 黄：3 次元測域センサ)

3.2　測域センサ

3.2.1　2 次元測域センサ URM-40LC-EW

距離 40m、水平 270° の範囲を計測可能な屋外用レーザスキャナ (図 4)[2] である。周囲の環境や障害物を検知できる。

図 4　2 次元測域センサ

3.2.2　3 次元測域センサ YVT-35LX-FX

距離 35m、水平 210°、垂直 40°の範囲を計測可能なレーザスキャナ (図 5)[3] である。また、測距原理には光源からパルス光を照射して、物体に反射させ、反射したパルス光が戻ってくるまでの時間を計測することで物体の震度を計算するという TOF(Time of Flight) 方式 [4] を採用している。

図 5　3 次元測域センサ

4　ROS

ROS(Robot Operating System)[5] とは、ロボット開発用のソフトウェアプラットフォームであり、ロボットの制御を行うソフトウェアフレームワークである。

4.1　トピック

ROS では、ノードと呼ばれるプログラム同士が相互に通信し、情報やデータの交換をすることで、ロボットの制御を行う。このようなノード間での通信を行う方法の 1 つがトピックである。トピックは、配信/購読 (publish/subscribe) 型の通信メカニズムを実装している。

5　実験による課題発見

我々は、学内のみならず、中之島ロボットチャレンジ・エクストラチャレンジに参加し、実験を行った。

5.1　中之島ロボットチャレンジ

中之島ロボットチャレンジ [6] とは、人々の往来する実環境において自律移動ロボットが問題無く行動できる技術開発の公開実験を大学や企業向けに提供する大会である。様々な研究開発機関が参加する技術交流の場を設けることで、ロボットの開発技術のレベルを向上させることを目的としている。この大会に参加したことで長いコースや屋外での実験ができ、学内では発見できなかった課題が見つかった。

我々は、7 月に中之島公園 (大阪市北区) の中央公会堂周辺の歩行者天国エリアの定められたコース (図 6) で自律走行実験を行ったところ、ロボットの 2 次元測域センサよりも低い位置にある障害物・下り段差を 2 次元測域センサでは検知できず、障害物に衝突しそうになったり、段差を踏み外しそうになるといった危険な走行があった。この問題を解決するためには、3 次元測域センサの実用化を実現することが必要であると考えた。

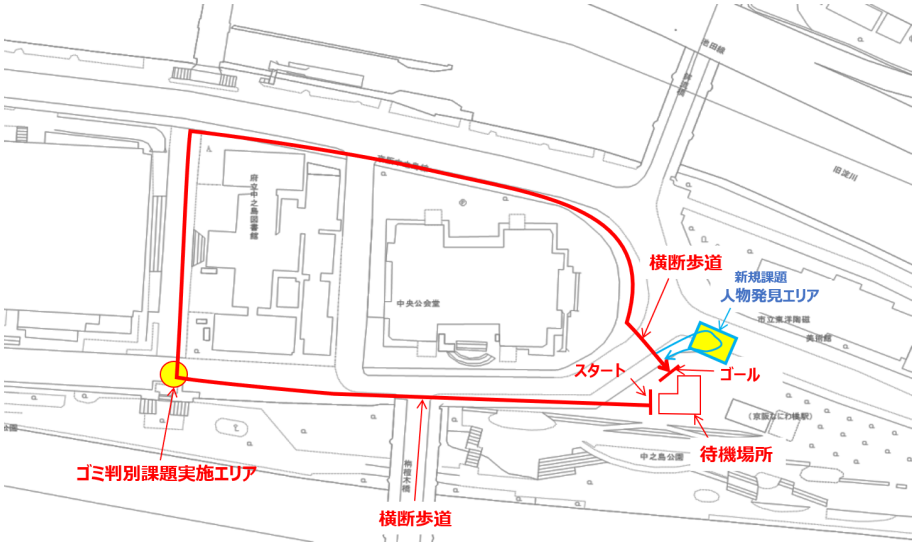

図 6　中之島チャレンジコース

5.2　中之島エクストラチャレンジ

11 月に ATC O's 棟の屋内および周囲の屋外の定められたコース (図 7) で自律走行実験を行った。ここでは、まず従来の 3 次元測域センサによる地形判定プログラムを用いて地形判定の実験を行った。しかしながら、正確に判定できない箇所が見つかった。これらの問題については、判定基準とデータ処理の見直しを行うことで解決できると考えた。

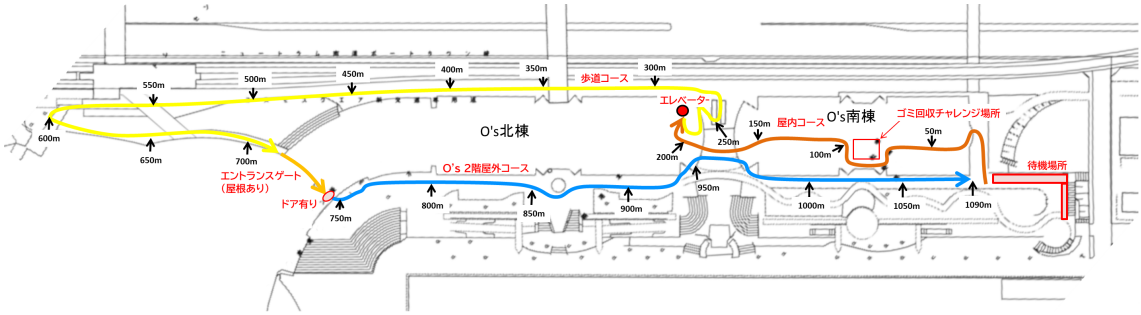

図 7　中之島エクストラチャレンジコース

6　実装

本節では、本研究で、従来より正確な地形判定を行うために作成したプログラムについて述べる。得られたデータの区分の仕方は、従来の 3 次元測域センサによる地形判定プログラムをもとにしており、判定の閾値の変更や点群データが取れない領域がある場合の処理の追加を行っている。

6.1　得られたデータの処理

トピックから配信された 3 次元測域センサの点群データを購読し、そのデータを Numpy の array に変換する。そのうちロボットの前方 50cm から 2m までの範囲を抽出し、30cm ごとに区切る。また、横方向はロボットの左右 50cm の範囲を抽出している。それぞれの範囲ごとに得られた点の z 座標の値を配列に格納し、それらの平均値を求める。そして、その配列の隣り合う要素の差分を取ることで地形判定を行う。

区切った範囲内に、3 次元測域センサが点群データを取れない領域がある場合、得られた配列に None が含まれる。そこで、処理は以下のような 3 パターンに分けて行った。

1. None の個数が 0 の場合
   得られた点群データの値をそのまま判定に用いる。
2. None の個数が、得られた点群データの個数の半分以上である場合
   壁または段差であると判定する。
   これは、学内や ATC での実験により、None が多数含まれるのは、ロボットの前方に壁や下り段差があるためにその範囲に点が検出できないという理由によると判明したためである。
3. None の個数が得られた点群データの個数の半分より少ない場合
   以下のような補完処理を行なった。
   - 配列の先頭と最後尾が None ではない場合
     データがある両端の点から平均を出して None のところにデータを入れる。
   - 配列の先頭が None である場合
     データがある右側 2 つの点から平均を出して None のところにデータを入れる。
   - 配列の最後尾が None である場合
     データがある左側 2 つの点から平均を出して None のところにデータを入れる。

6.2　判定

屋内で実験を行った上で定めた判定基準が以下の通りである。また、ロボットには 3 次元測域センサを上下逆に取り付けているため、z 軸は垂直方向で下向きが正となっている。本研究では、ロボットの前方 50cm から 2m までの範囲を抽出し、30cm ごとに区切るため、得られる配列の要素 (データの数値) の個数は 5 個となる。そして、その配列の隣り合う要素の差分を取り、その結果から判定を行うため、データの最終的な個数は、4 個となる。また、段差とスロープの判定基準は、バリアフリー法 [7] で定められた勾配 1/12 以下をもとにして、$30 \div 12 = 2.5$ の結果 2.5cm としている。平地の判定基準に関しては、人間の目で「平地」と判断できる場所における 3 次元測域センサのデータを参考にし、平地と判断できる範囲を -1.5cm から 1.5cm としている。

- 下り段差
  隣り合うデータの z 軸の数値の差が 2.5cm より大きい箇所が 1 つ以上ある場合、下り段差であると判定し、downstep と出力する。
- 上り段差
  隣り合うデータの z 軸の数値の差が −2.5cm より小さい箇所が 1 つ以上ある場合、上り段差であると判定し、upstep と出力する。
- 下りスロープ
  隣り合うデータの z 軸の数値の差が 1.5cm 以上 2.5cm 以下の箇所が 2 つ以上ある場合、下りスロープであると判定し、downslope と出力する。
- 上りスロープ
  隣り合うデータの z 軸の数値の差が −2.5cm 以上 −1.5cm 以下の箇所が 2 つ以上ある場合、上りスロープであると判定し、upslope と出力する。
- 平地
  隣り合うデータの z 軸の数値の差が −1.5cm より大きく 1.5cm より小さい箇所が 2 つ以上ある場合、平地であると判定し、flat と出力する。
- 上記以外
  unknown と出力する。

6.3 停止処理

判定が壁または段差・上り段差・下り段差である場合、ロボットを停止させる処理は、判定結果をトピックに配信して、ロボットの自律走行時に用いるノードがこれを購読し、段差であるならば、自律走行の動作を行う while 文を抜けることによって実現させている。また、while 文を抜ける際に、「stop!」と出力させることにより、壁・段差を検知したためにロボットが停止したと確認できるようにした。

7　検証

実装したプログラムで実験を行った。実験の方法としては、学内・中之島エクストラチャレンジ (5.2 節参照) で実験を行った際に、トピックに流れているメッセージを保存する機能である rosbag を用いて走行時のデータを記録し、そのデータを使うことで自律走行時と同じ環境を仮想的に再現して、作成したプログラムでの判定結果を検証した。ここでは、学内の G 棟 4 階下り階段 (図 8) のデータで仮想的な実験を行った際の様子を例として示す。実験のときに図 9 のように 3 つのターミナルを開け、プログラムの各部分の処理を観察した。それぞれのターミナルに出力されているものは図 10、11、12 の通りである。

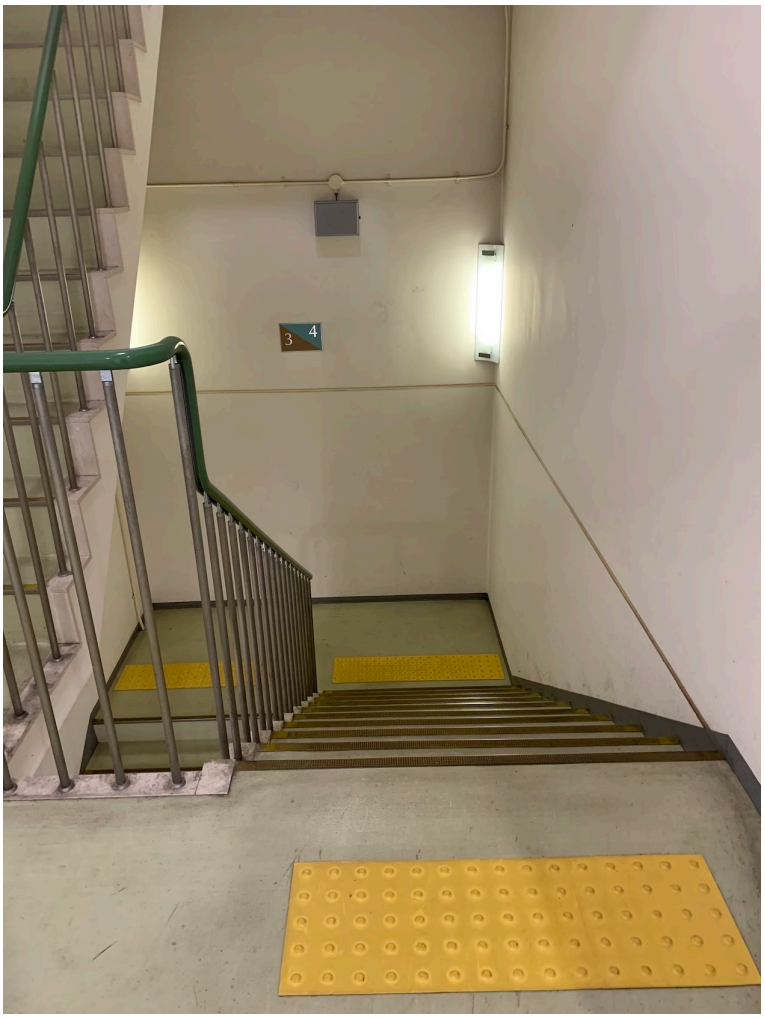

図 8　G 棟 4 階下り階段

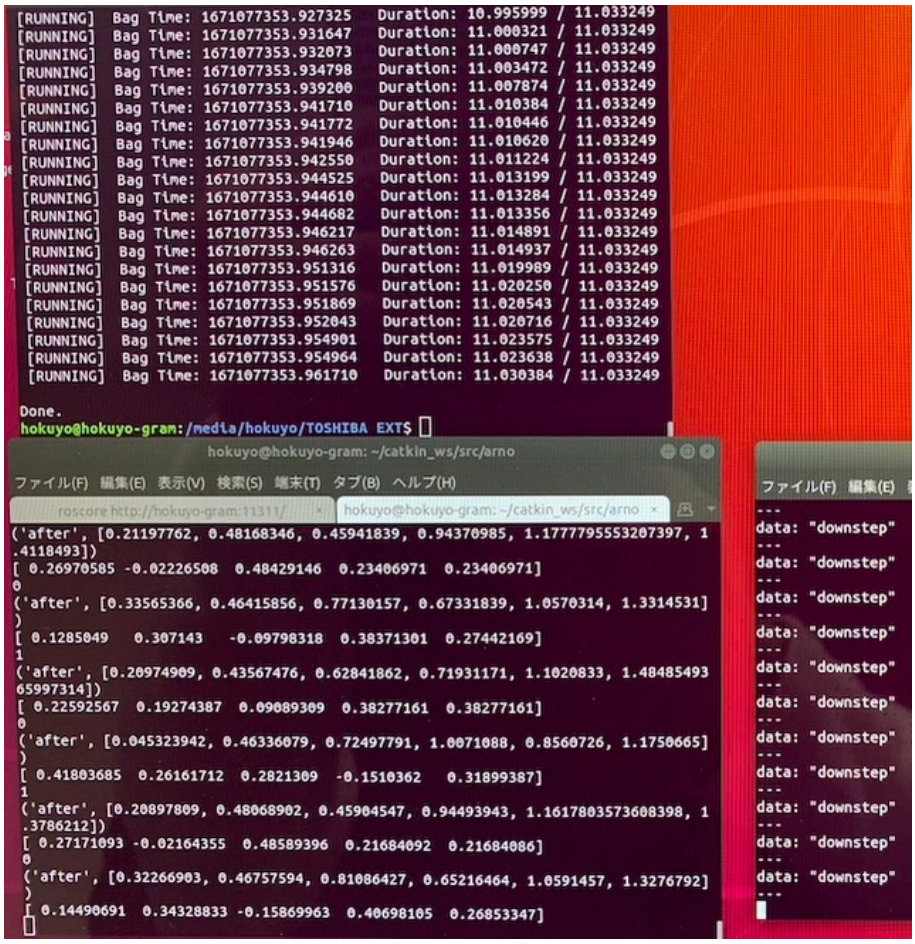

図 9　実験の様子

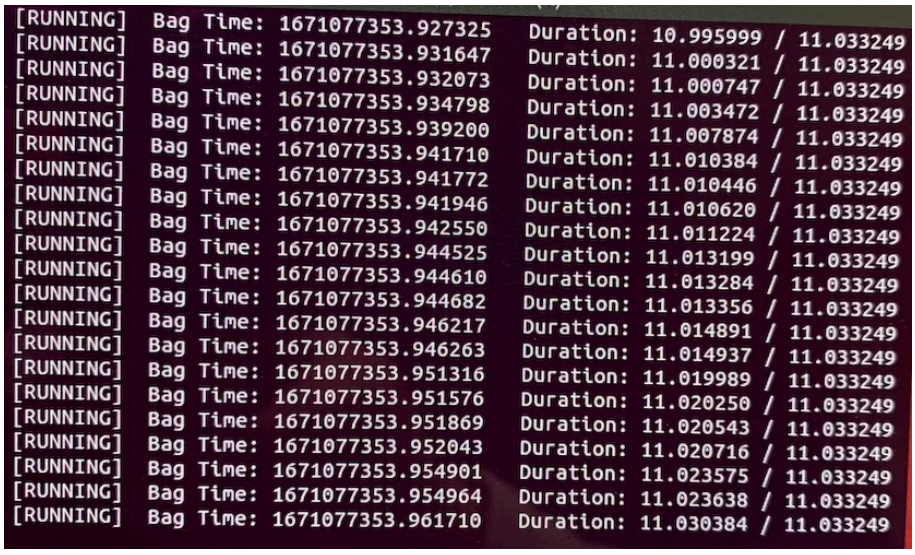

図 10　rosbag で記録したデータを再生

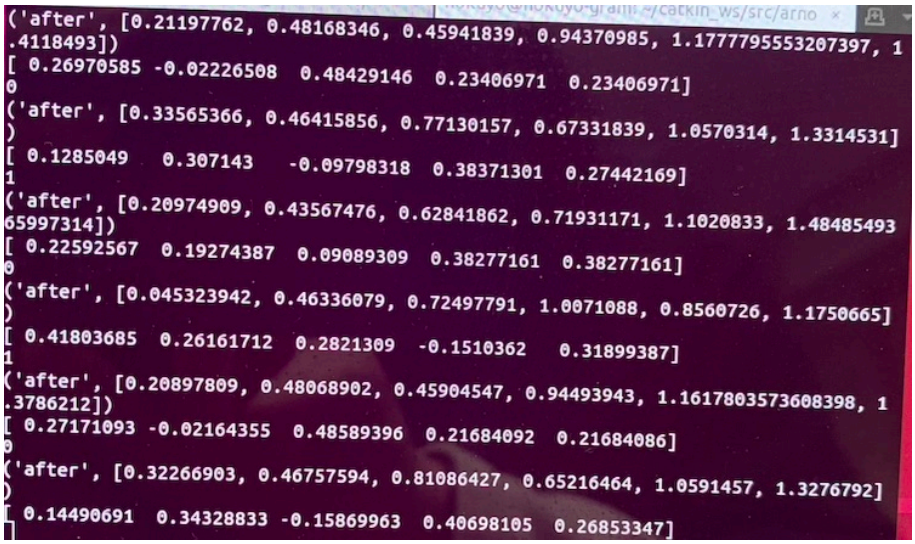

図 11　データを処理

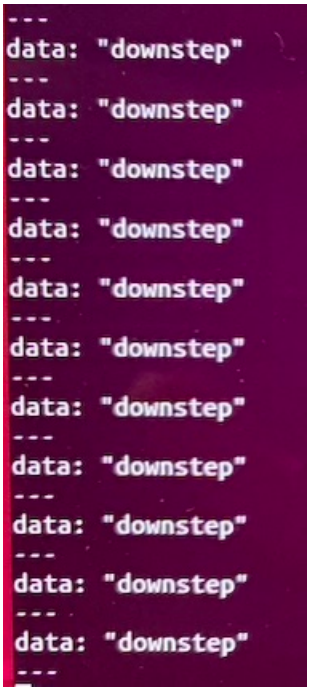

図 12　判定結果をトピックに配信

8　結果

点群データが取れない領域がある場合の処理を実装したことにより、従来のプログラムでは正確に判定が行えなかった場所が判定できるようになった。また、判定基準の見直しを行ったことにより、2 節で挙げた場所を含め、屋内で実験を行った場所全てに対応する汎用性の高いものができた。実験に成功した場所 (抜粋) は

図 13、14 の通りである。

図 13　ATC 周辺の下りスロープ

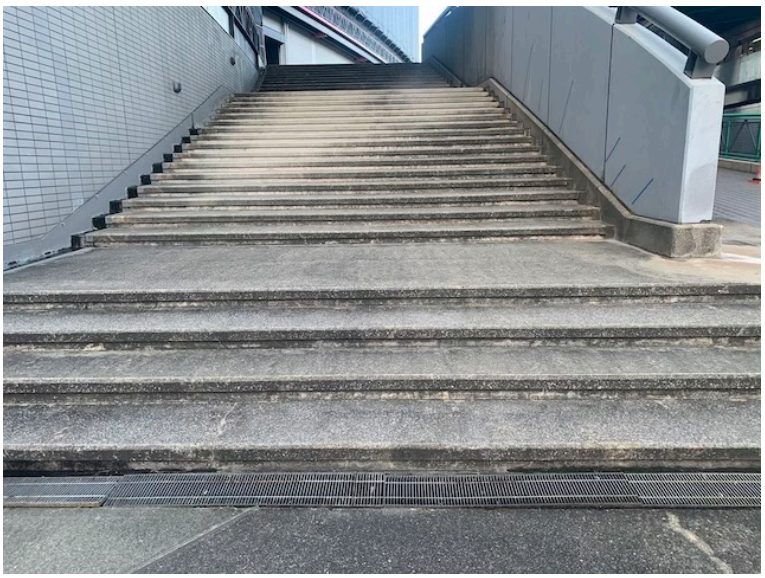

図 14　ATC 周辺の上り階段

9　まとめ

3 次元測域センサのデータをロボットの自律走行に活かすことができた。また、本学での昨年の研究と異なり、点群データが得られなかった場合の補完処理を行い、判定させることと判定基準を見直したことで、プログラムの保守性が向上し、より精度と汎用性の高い判定が行えるようになった。また、判定結果を利用してロボットを停止させることも可能になった。今後の課題としては、3 点挙げられる。表 1 の通りである。

表 1　課題一覧

課題	考えられる原因
屋内スロープ用に設定されている坂道の判定基準を屋外の自然形成された坂道に適用可能か	現在の判定基準だと勾配のきつい屋外の自然形成された坂道を段差と判定されてしまう可能性があるため
スロープの高低差よりも低い段差を見極められるか	本研究では、スロープの高低差よりも低い段差は判定対象としていないため
ロボットが段差を検知して停止した後、迂回ルートを探して自律走行を続けられるか	6.3 節より、while 文を抜けることで停止処理を実現しているため、自律走行を続けられないため

また、今後、新たな場所でロボットの自律走行実験を行った場合、本研究で定めた判定基準を再び変更する必要に迫られるか、という問題も考えられる。これについては、ニューラルネットなどを用いた機械学習によって、多数の場所に対応できる判定器を構築することも考えられる。しかしながら、機械学習には多量の教師データが必要であるため、これを実現しようとすると、非常に多くの場所で実験データを取る必要があり、大きな手間を要する点を如何に解決するのかが課題となってくる。

10　謝辞

本研究にあたり、丁寧かつ細やかなご指導をしてくださった新出尚之准教授、ロボットの提供・中之島ロボットチャレンジにご協力頂いた北陽電機の皆様、並びにロボットの使い方のご指導や中之島ロボットチャレンジに共にご参加頂いた新出研究室の先輩方に心から感謝申し上げます。

参考文献

[1] 渡辺由佳. 3 次元測域センサを用いた坂道判定. 2021 年度卒業論文, 奈良女子大学生活環境学部情報衣環境学科生活情報通信科学コース.

[2] 北陽電機株式会社. URM-40LC/LCN-EW. https://www.hokuyo-aut.co.jp/search/single.php?serial=189.

[3] 北陽電機株式会社. YVT-35LX-F0/FK. https://www.hokuyo-aut.co.jp/search/single.php?serial=165.

[4] 安富啓太, 川人祥二. Time-of-flight カメラ. https://www.jstage.jst.go.jp/article/itej/70/11/70_880/_pdf.

[5] M. Quigley, B. Gerkey, and W. D. Smart. プログラミング ROS—Python によるロボットアプリケーション開発—. オライリー・ジャパン, 2017. (河田卓志 (監訳)).
[6] 中之島ロボットチャレンジ実行委員会. 中之島ロボットチャレンジ 2022. https://www.nakanoshima-rc.jp.
[7] 国土交通省. バリアフリー法. https://www.mlit.go.jp/sogoseisaku/barrierfree/content/001379342.pdf, 令和 3 年 4 月 1 日施行.