

2021 年度 卒業論文
3次元測域センサを用いた坂道判定

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 18480390 渡辺由佳

2022 年 2 月

目次

1	はじめに	3
2	関連研究	3
3	自律ロボットとその制御	4
3.1	使用ロボットについて	4
3.2	測域センサ	5
3.3	ロボット制御機能	5
4	中之島チャレンジ	6
5	3次元測域センサによる高低差認識	6
5.1	3次元測域センサからのデータ取得と座標軸の向き及び原点位置の特定	6
5.2	高さ平均出力プログラムの実装と実験	8
5.3	高さ平均出力結果	10
5.4	地形判別プログラム	11
6	まとめ	14
7	謝辞	14

概要

ロボットが目標を持ち、目標達成のために行動する上で、目標物への移動は必要な能力である。我々が従来実験に使用していたロボットは、自律移動に際し、物体を立体的に捉えることができず、段差を回避したり、坂道を壁と区別したりすることが困難であり、そのことが自律走行の安定性に影響していた。そこで本研究では、自律移動ロボットに3次元測域センサを搭載することで、ロボットが段差や坂道を認識できるようにした。具体的には、ロボットに新たに北陽電機の3次元測域センサ YVT-35LX-FK を搭載し、センサが感知した点の高さを見ることで、ロボットの前方の坂や階段を検知できるようにした。それにより、今まで見えていなかった小さな段差に衝突したり、落ちてしまう危険を回避することができ、自律走行の安定性向上が期待できる。

1 はじめに

近年、実世界において人間の立ち合いを伴わなくても、目標を与えられると達成手順を自分で計画し、かつ状況を判断して目標を達成できる自律ロボットの需要が高まり、実現されつつある。自律ロボットにおいては、人間の判断や指揮・命令が伴わないため、人間と同様に実世界における多様な環境変化を感知し、周囲の状況に対応しながら、その都度、最適な目標達成のための行動選択を自ら行わねばならない。我々が目指しているロボットの自律走行においても、安定した走行のためには、段差や坂道など多様な進路の状況に対応する必要がある。しかし、従来我々が用いていたロボットには2次元測域センサしかなく、奈良女子大学内や中之島チャレンジ [1] での試験走行では、段差にぶつかりそうになったり、坂道を壁と認識して登らなかったりなど、多くの問題が起きることがわかった。

そこで我々はロボットに3次元測域センサを追加し、走行中に進路前方の高さの平均を確認しながら進むことで、坂道か段差かを判定し、進むべきか回避すべきかの判断材料をロボット自身が得られるようにした。

2 関連研究

3次元測域センサを用いた段差検出の試みとして、構らの「移動ロボットの障害となる段差・くぼみを記録した地図の生成と経路計画」[2] が挙げられる。この研究では、あらかじめ段差地図を作成し、段差・くぼみ情報を蓄積することで、6cm程度の小さい段差やくぼみを回避している。段差地図の例を図1に示す。

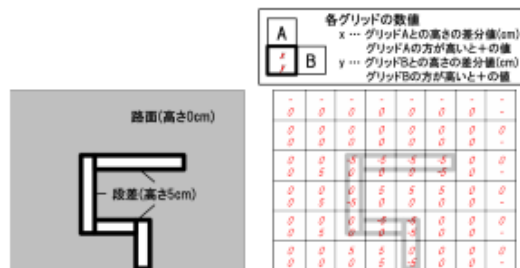


図1: 段差地図のイメージ

段差地図作成の具体的な方法は、まず3次元測域センサを用いてロボット前方の路面を計測し、点群を取得する。その取得した点群データを座標変換し、各グリッドに入れたのち、点群の高さの平均値を算出している。それをもとに隣接グリッド(上グリッドと右グリッド)との高さの差分値を算出し、それを各グリッドに格納する。これらの作業を繰り返すことで、目的地までの経路の段差地図を事前に作成し、段差が検出された

ところは回避するというものである。各グリッドの高さを算出した様子を図 2、隣接グリッドとの高さの差分値を格納したものを図 3 に示す。

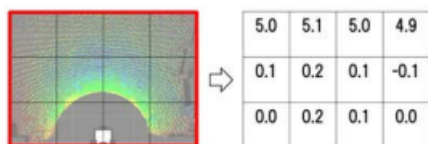


図 2: 各グリッドの高さを算出

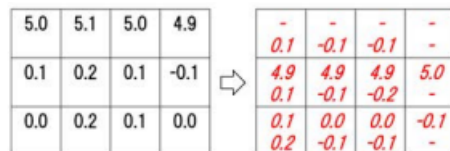


図 3: 隣接グリッドとの高さの差分を出力

しかし、この研究は屋外の小さな段差のみを対象としており、屋内環境に対応できるか、段差と坂の違いが判別できるのか、などについては言及がなかった。そこで、本研究では屋内でも実験を行い、さらに段差と坂の違いなどを検出できることを目標とした。また、この研究では、事前に段差地図を作成したのち走行を試みているが、実際のフィールドで地図を事前に作成することは困難であるため、本研究では走行しながら、随時段差を確認していく手法の実現を目標とした。

3 自律ロボットとその制御

3.1 使用ロボットについて

本研究では、測域センサを搭載した北陽電機製の自律走行ロボット Arno(図 4) を用いた。当初は 2 次元測域センサ URM-40LC-EW を 2 台、前方と後方に搭載していたが、本研究ではさらに 3 次元測域センサ YVT-35LX-FK を 1 台前方に取り付けた。これにより、2 台の 2 次元測域センサによって 360° の周囲を確認することができ、前方に取り付けた 3 次元測域センサによって、ロボットの進行方向の高低差が確認できる。



図 4: 自律走行ロボット Arno

3.2 測域センサ

3.2.1 2次元測域センサ URM-40LC-EW

距離 40m、270° の範囲を計測する屋外用のレーザスキャナである (図 5)[3]。周囲の環境の認識や侵入、障害物を検知できる。Arno では、前後に 2 つ取り付けることで 360° 検知が可能である。



図 5: 2次元測域センサ

3.2.2 3次元測域センサ YVT-35LX-FK

距離 35m、水平 210°、垂直 40° の範囲を 3次元計測するレーザスキャナである (図 6)[4]。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深度を計算する、TOF(Time of Flight)方式 [5] を採用している。本研究では、前方に取り付けることで、ロボットの進行方向を 3次元で計測できるようにした。



図 6: 3次元測域センサ

3.3 ロボット制御機能

3.3.1 ROS

ROS[6]とは、ロボット開発において重要な役割を果たすオープンソースのロボット制御ソフトウェア、およびそれを包括するロボット開発プラットフォーム全体のことで、ロボティクス向けの豊富なライブラリ群を利用可能であること、シミュレーションツールやデバッグツールといった高機能な開発ツールを使えることなどが主な特徴である。ROSは、ノード間をトピックと呼ばれる通信路で接続し、発信側ノードがトピック上にメッセージを発行し、受信側ノードがこのメッセージを受信することで通信を行う。メッセージは受信先を特定せずに流され、受信側ノードは自分が監視するトピックに流れてきたメッセージのみ拾う。受信側ノード

がメッセージを受信すると、ノード内のコールバック関数が呼び出され、その関数の中でメッセージで伝えられた情報の処理を行う。

3.3.2 rviz

rviz[7] とは、ROS アプリケーションの 3 次元可視化ツールであり、ロボットセンサからのセンサ情報をキャプチャして、そのキャプチャしたデータを 3D ビューとして表示するものである。特に、3 次元データは座標値を数値で表示しても形状を把握しにくいいため、そのままではデータを処理するプログラムなどを開発することが難しいが、3D ビューとして可視化すれば、ロボットの位置や姿勢、データの空間的配置などが把握しやすくなり、プログラム開発に有用である。

4 中之島チャレンジ

中之島チャレンジとは、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な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、ロボット開発技術のレベルを向上させることを目的としている。7 月、10 月には、中之島公園の中央公会堂周辺の歩行者天国エリアでの自律走行、11 月、12 月には、八幡屋公園での自律走行を行った。学内とは異なった広いフィールドでの実験により、学内では発見できなかったさまざまな課題が見つかった。例えば、平地の走行は人通りの少ない学内では問題なく行っていたが、人や自転車などの障害物が多すぎると走行できなくなるといった問題が見られた。また、低いところの段差を認識できず、ロボットが落ちそうになったり、上り坂を壁と認識してしまい、進めないなどの問題も起きた。このうち高低差に起因する問題については、2 次元測域センサのみで対処するのではなく、3 次元測域センサで地形の判断をすることで、これらの問題が解決できると考えた。

5 3 次元測域センサによる高低差認識

5.1 3 次元測域センサからのデータ取得と座標軸の向き及び原点位置の特定

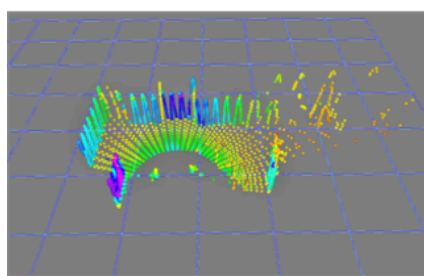
3 次元測域センサが 1 回のセンシングを行うと、ROS が定義する PointCloud2 という型のメッセージが 1 つ、ROS のトピックを通じて配信される。そして、rviz は PointCloud2 型のデータを直接 3 次元のビューとしてプロットすることができる。しかし、PointCloud2 型のメッセージ 1 つには、その時センスした点の 3 次元座標が多数、一括して含まれており、そのままでは点ごとのデータが取り出せない。そこで、3 次元データとして処理するには、まずこれを Numpy 配列の array に変換し、点の配列として個別の点を取り出せる必要がある。以上の考えのもと、センサからの 3 次元データを処理する ROS ノードを作成した。

3 次元測域センサは上側がよく見えるように設計されているが、本研究では主にロボットの進行方向の地面を観測するため、3 次元測域センサを上下逆に取り付けた。そのため、センサの情報を利用するためには座標の向きや原点の位置の確認が必要となる。実際、当初はこの確認を行わずにデータを処理していたため、座標軸の向きに関して正しくない想定でデータを処理しており、正しい結果が得られなかった。そこで、確認に用いるため、まず、センサからのデータのうち、特定の範囲にあるもののみを抽出して再配信するノードを作成した。これは、Numpy の array 型に変換したデータから、Numpy の機能を使って、座標が特定の値をとる点だけを選択し、PointCloud2 型に戻して再配信するノードである。rviz がこのノードからの出力をプロット

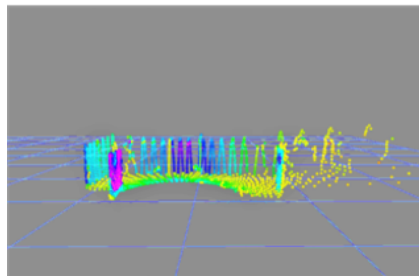
することで、特定の範囲の点だけの表示ができる。再度 PointCloud2 に戻すのは、Numpy の array のままでは rviz が表示できないためである。このノードを用いて、奈良女子大学 G 棟 G403 のドア付近 (図 7) の 3 次元測域センサによるセンシング結果のうち x, y, z 座標がそれぞれ正・負であるもののみを表示したものを図 9～図 11 に示す。比較のため、このノードで処理していない全体のデータを図 8 に示す。



図 7: 実験場所: 研究室のドア付近

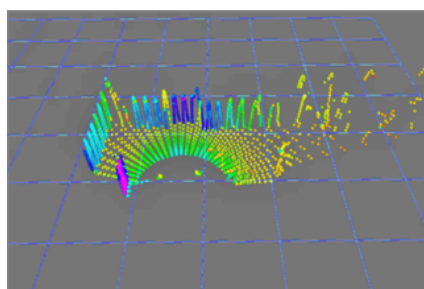


(a) 平面での全体の表示結果

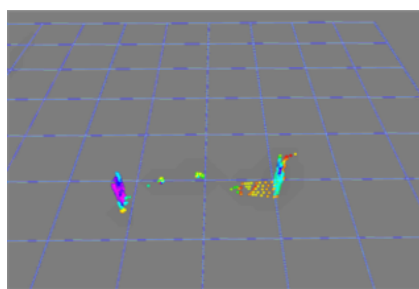


(b) 高さ方向の表示結果

図 8: rviz での全体図表示結果



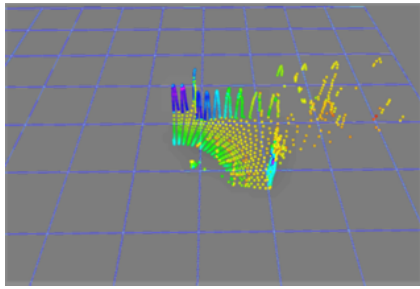
(a) x 座標が正のもの



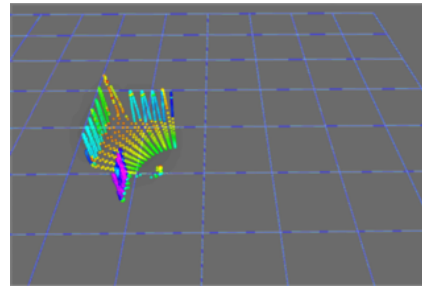
(b) x 座標が負のもの

図 9: x 座標の見え方

これらの結果より、 x 軸は前後方向で前向きが正となり、 y 軸は横方向で右側が正となり、 z 軸は垂直方向で下向きが正であることがわかった。以上の結果を座標軸で表すと図 12 のようになる。

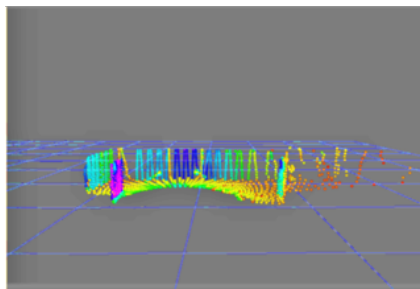


(a) y 座標が正のもの

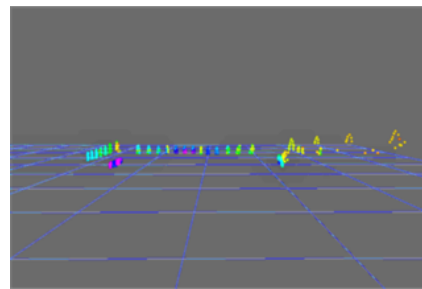


(b) y 座標が負のもの

図 10: y 座標の見え方



(a) z 座標が正のもの



(b) z 座標が負のもの

図 11: z 座標の見え方

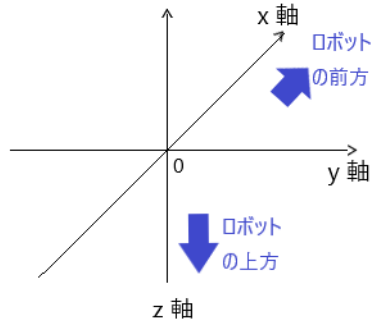


図 12: 3次元測域センサによる座標軸のイメージ

5.2 高さ平均出力プログラムの実装と実験

本研究ではロボットの進路の地形を観測し、その高低差によって平地・坂、段差の判定を行うことを目指した。そこで、前節で判明した座標軸のもと、センサが観測した点の高さ平均を出力することでロボット前方の地形を分析するプログラムを作成した。地形の分析には、Numpy の array に変換したデータを、 x 座標の値によって表 2~9 のような区間に分類し、分類ごとにその分類に属する点の z 座標の平均を求めた。分類の境界を様々な値にして実験を重ねた結果、センサの取り付け位置に起因して、ロボットの前方 50cm は認識できないことがわかった。そこで、ロボットの前方 50cm から 5m までの範囲を 50cm ごとの区間に区切った。ま

た、ロボットの正面のみを重点的に見るため、横方向の観測範囲を -0.5m から 0.5m までとした。そして、奈良女子大学構内の屋内、屋外のそれぞれにつき、平地、上り坂、下り坂、上りの段差のある場所、下りの段差のある場所で実験を行い、各区間ごとに高さの平均と、隣接区間の平均との差分を求めた。表 1 に実験場所の詳細を、図 13 にそれぞれの位置を示す(屋外の下り段差は、ロボットを走行させた範囲内に該当がなかった。また屋外の上り段差については、高さの異なる 2 箇所で行った)。各場所につきそれぞれ 5 回の実験を行った。

表 1: 屋内、屋外での実験場所

	平地	上り坂	下り坂	上りの段差		下りの段差
屋内	① G 棟 4 階 廊下	② G 棟 4 階から C 棟への スロープ	③ C 棟から G 棟 4 階への スロープ	④ G 棟 3 階から G 棟 4 階への 階段		⑤ G 棟 3 階から G 棟 2 階への 階段
屋外	⑥ G 棟横道路	⑦ 体育館から G 棟横への坂	⑧ G 棟横から 体育館への坂	G 棟入口前の 段差		該当なし
				⑨ 6cm	⑩ 13cm	



図 13: 奈良女子大学構内図

5.3 高さ平均出力結果

実験は、表1の各ケースについてそれぞれ5回行ったが、どの回もほぼ同じデータが得られたため、ここにはそのうち1回分のデータを示す。屋内でのそれぞれの実験結果を表2～表6に示す。図12より、位置が高いほど高さの値は小さい。また、ある区分に属する点が1つも検出されなかった場合は平均が算出できないので、その区分での出力はnanとなっている。

表2: G棟4階廊下での実験

区間 (m)	0.5~1	1~1.5	1.5~2	2~2.5	2.5~3	3~3.5	4.5~4	4~4.5	4.5~5
高さ平均 (m)	0.552	0.568	0.585	0.603	0.617	0.625	0.632	0.617	0.653
隣の区間 との差分 (m)		0.016	0.017	0.018	0.014	0.008	0.007	-0.015	0.036

表3: G棟4階からC棟への上りスロープでの実験

区間 (m)	0.5~1	1~1.5	1.5~2	2~2.5	2.5~3	3~3.5	4.5~4	4~4.5	4.5~5
高さ平均 (m)	0.544	0.526	0.488	0.443	0.399	0.352	0.308	0.238	0.219
隣の区間 との差分 (m)		-0.018	-0.038	-0.045	-0.044	-0.047	-0.044	-0.07	-0.019

表4: C棟からG棟4階への下りスロープでの実験

区間 (m)	0.5~1	1~1.5	1.5~2	2~2.5	2.5~3	3~3.5	4.5~4	4~4.5	4.5~5
高さ平均 (m)	0.547	0.590	0.651	0.724	0.784	0.853	0.895	0.973	0.991
隣の区間 との差分 (m)		0.043	0.061	0.073	0.069	0.069	0.042	0.078	0.018

平地ではあまり変化が見られなかった。これは平坦な道であるため、高さ平均がほとんど変わらないためである。上り坂ではロボットから離れるほど高さ平均が少しずつ小さくなり、下り坂では少しずつ大きくなっ

表 5: G 棟 3 階から 4 階への上り階段での実験

区間 (m)	0.5~1	1~1.5	1.5~2	2~2.5	2.5~3	3~3.5	4.5~4	4~4.5	4.5~5
高さ平均 (m)	0.506	0.259	-0.006	-0.179	nan	nan	nan	nan	nan
隣の区間 との差分 (m)		-0.247	-0.265	-0.173	nan	nan	nan	nan	nan

表 6: C 棟 3 階から 2 階への下り階段での実験

区間 (m)	0.5~1	1~1.5	1.5~2	2~2.5	2.5~3	3~3.5	4.5~4	4~4.5	4.5~5
高さ平均 (m)	0.485	0.341	0.746	0.878	1.513	2.064	2.466	2.478	0.888
隣の区間 との差分 (m)		-0.144	0.405	0.132	0.635	0.551	0.402	0.012	-1.59

た。この結果は、上り坂ではセンサと地面との高低差がロボットから離れるほど減り、下り坂では増えることを反映している。また、上り階段では 2.5m までは急激に減り、それ以降は計測できなかった。これはセンサの上向きの角度が 5° のみのため、上り階段の数段目以降は壁と同様となって計測できないことによる。下り階段では急激に大きくなるが、階段を降りきると、高さ平均の値が小さくなった。これは、階段を降りきった先の壁を検出するからである。また、平地と下り坂は屋外でも屋内と全く同様の実験結果が得られたため、ここでは屋内の実験結果のみ示した。上り坂については、屋内のスロープと違い、屋外では緩やかな坂となっており、隣接する高さ平均の値の変化が小さかった。屋外の上り坂の実験結果を表 7 に示す。段差については、屋外の 1 段しかないような小さな段差では階段とは異なるデータが得られるため、表 8、表 9 に結果を示す。高さ 6cm の段差の場所は段差の奥に壁があったため、段差を検出した地点より奥の高さ平均は nan となっている。高さ 13cm の段差の場所は段差の奥が少し下がり、そのさらに奥が壁となる。そのため、高さ平均の値が小さくなった後、少し大きくなり、また小さくなるというデータが得られた。このように、実際の地形をよく反映した実験結果が得られた。

5.4 地形判別プログラム

5.3 節の結果は、平地・坂、段差それぞれの地形に対して明瞭な特徴を示すものであったため、その判別をプログラムで行うことが可能と考え、プログラムを作成した。まず、判別に用いる基準を抽出するため、5.3 節で得られた差分を gnuplot で可視化 (表 14) した結果、平地、坂、段差のいずれについても顕著な傾向があることが見て取れた。これをもとに、判定は各地形の差分が -0.04 以上 0.04 未満、 0.04 以上 0.1 未満、 0.1 以上、 -0.1 以上 -0.04 未満、 -0.1 未満の各段階に属する回数によって行うことにした。差分 8 個のうち、

表 7: 体育館から G 棟への上り坂での実験

区間 (m)	0.5~1	1~1.5	1.5~2	2~2.5	2.5~3	3~3.5	4.5~4	4~4.5	4.5~5
高さ平均 (m)	0.497	0.477	0.468	0.468	0.454	0.443	0.466	0.407	0.462
隣の区間 との差分 (m)		-0.02	-0.009	0	-0.014	-0.011	0.023	-0.059	0.055

表 8: 屋外の高さ 6cm での実験

区間 (m)	0.5~1	1~1.5	1.5~2	2~2.5	2.5~3	3~3.5	4.5~4	4~4.5	4.5~5
高さ平均 (m)	0.578	0.595	0.570	0.573	0.572	0.322	0.026	nan	nan
隣の区間 との差分 (m)		0.017	-0.025	0.003	-0.001	-0.25	-0.296	nan	nan

表 9: 屋外の高さ 13cm での実験

区間 (m)	0.5~1	1~1.5	1.5~2	2~2.5	2.5~3	3~3.5	4.5~4	4~4.5	4.5~5
高さ平均 (m)	0.551	0.580	0.567	0.380	0.195	0.495	0.481	0.094	0.071
隣の区間 との差分 (m)		0.029	-0.013	-0.187	-0.185	0.3	-0.014	-0.387	-0.023

-0.04 以上 0.04 未満の区間に属するものが 6 つ以上あれば平地、そうではなくて 0.04 以上 0.1 未満の区間に属するものが 5 つ以上あれば下り坂、そうではなくて 0.1 以上の区間に属するものが 2 つ以上あれば下りの段差、そうではなくて -0.1 以上 -0.04 未満の区間に属するものが 5 つ以上あれば上り坂、そうではなくて -0.1 未満の区間に属するものが 2 つ以上あれば上りの段差とした。平地では坂や段差のような大きな差分は現れないため、大きな差分が現れた場合はその回数が少なくても坂や段差と判断してよいと考えた。

このプログラムを用いて、正確に判別できるか、屋内、屋外それぞれの平地、坂、段差がある各地形ごとに 5.2 節の実験で取得した 5 回分ずつのデータを使用して確認した。表 10 に判別結果の詳細を示す。屋外の上り坂についてのみ、5 回全てについて平地と誤判定したが、それ以外の地点では正確に判別できた。表 7 に屋外の上り坂の高さ平均出力結果を示す。判定に失敗した奈良女子大学構内の体育館から G 棟への坂は、はじ

めはゆるやかで途中から勾配が少し変化するため、ロボットの前方 5m まででは坂が緩やかであり、そのため平地と判断されたと考える。また、一度ロボットが坂に差し掛かるとロボットにとってその坂が平地となり、そのさらに先で少し勾配が増してもロボットは平地と認識してしまう。本研究の動機の一つはロボットが上り坂を壁と誤認識して進めなくなる問題を 3 次元測域センサを利用して解決することであり、平地と判断するほどの緩い上り坂ではその問題が起きないので、実質的には問題はないと考えられる。また、ロボットは段差以外であれば走行できるため、上り坂を平地と誤認識しても走行に問題はないと考える。

以上より、地形判別プログラムを用いることで、自動的に平地、上り坂、下り坂、上りの段差、下りの段差の判別ができ、ロボットが通行可能なのか、不可能なのかの判定に利用できることがわかった。

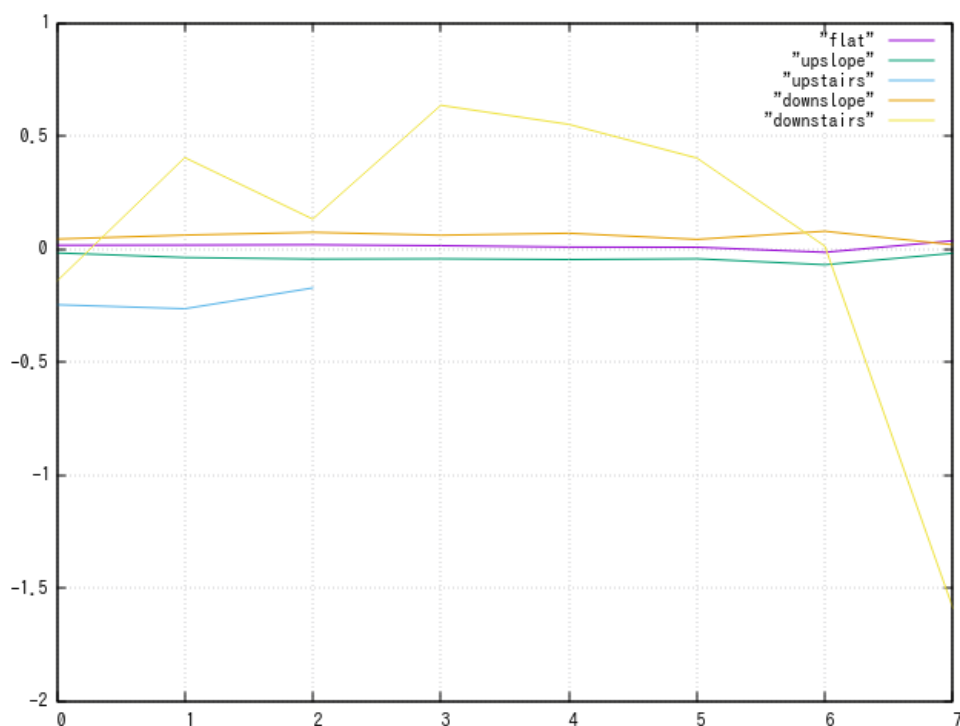


図 14: 1 回分のデータの出力結果

表 10: 各地点での判別結果 (成功回数/失敗回数)

	平地	上り坂	下り坂	上りの段差	下りの段差
屋内	5/0	5/0	5/0	5/0	5/0
屋外	5/0	0/5	5/0	5/0	5/0
					該当なし

6 まとめ

3次元測域センサを利用して、ロボットの前方の地形を判定することができた。また、先行研究と異なり、屋内環境への対応や、坂と段差の区別も可能であることを示し、さらに、あらかじめ計画しておくのではなく、その場での計測で判別可能とした。今後の課題として、今回計測した場所以外の坂や段差の判定がうまく行くかの検討が挙げられる。また、現在はまだ地形の判定と自立走行が連携できていないため、プログラムによる判断結果をロボットが利用して自律走行が行えるようにする必要がある。

7 謝辞

本研究の遂行及び本論文の執筆にあたって、最後まで熱心にご指導頂いた新出尚之准教授、論文に対してアドバイスを下さった嶋浩靖准教授、自律走行ロボット Arno・2次元測域センサ、3次元測域センサを提供していただいた北陽電機株式会社の皆様、特に研究についてのヒントを頂いた北陽電機の嶋地様、ならびに多くのご指摘やヒントを頂いた新出研究室の皆様深く感謝の意を表します。

参考文献

- [1] 中之島ロボットチャレンジ実行委員会. Nakanoshima Robot Challenge 2021. <https://www.nakanoshima-rc.jp>. (2022年2月10日閲覧).
- [2] 構勇海, 萬礼応, 大矢晃久. 移動ロボットの障害となる段差・くぼみを記録した地図の生成と経路計画. 第22回計測自動制御学会 システムインテグレーション部門講演会, 2021年12月.
- [3] 北陽電機株式会社. URM-40LC/LCN-EW. <https://www.hokuyo-aut.co.jp/search/single.php?serial=189>. (2022年2月10日閲覧).
- [4] 北陽電機株式会社. YVT-35LX-F0/FK. <https://www.hokuyo-aut.co.jp/search/single.php?serial=165>. (2022年2月10日閲覧).
- [5] 安富啓太, 川人祥二. Time-of-flight カメラ. 一般社団法人 映像情報メディア学会 映像情報メディア学会誌, Vol. 70, No. 11, pp. 880–885, 2016.
- [6] Open Source Robotics Foundation. ROS. <https://wiki.ros.org/ja>. (2022年2月10日閲覧).
- [7] Open Source Robotics Foundation. ROS. rviz. <https://wiki.ros.org/ja/rviz>. (2022年2月10日閲覧).