

# 卒業論文

## 視覚と行動の end-to-end 学習により経路追従行動を オンラインで模倣する手法の提案 (オフラインでデータセットを収集して訓練する手法の検証)

A proposal for an online imitation method of path-tracking  
behavior by end-to-end learning of vision and action  
(Validation of a method to collect and train datasets offline)

2023 年 1 月 9 日 提出

指導教員 林原 靖男 教授

千葉工業大学 先進工学部 未来ロボティクス学科

19C1068 高橋祐樹



# 概要

## 視覚と行動の end-to-end 学習により経路追従行動を オンラインで模倣する手法の提案 (オフラインでデータセットを収集して訓練する手法の検証)

近年、自律移動ロボットの研究が盛んに行われている。本研究室においても、2D-LiDAR を用いた自律移動システムの出力を教師信号として与えることで、ロボットの経路追従行動をオンラインで模倣する手法を提案し、実験を行うことで有効性を示してきた。本研究では、従来手法を基に、目標とする経路上及び周辺のデータを一度に収集し、オフラインで訓練する手法を提案する。提案手法では、経路上にロボットを配置し、カメラ画像と教師データとなる目標角速度を収集する。それらのデータを基にオフラインで学習を行い、学習後はカメラ画像を入力とした学習出力により自律移動する。

キーワード: end-to-end 学習, Navigation, オフライン

# abstract

A proposal for an online imitation method of path-tracking  
behavior by end-to-end learning of vision and action  
(Validation of a method to collect and train dataset offline)

Recently, autonomous mobile robots have been studied extensively. In our laboratory, we have proposed an online imitation method of a robot's path-following behavior by providing the output of a 2D-LiDAR-based autonomous mobile system as a teacher signal, and have demonstrated the effectiveness of the proposed method through experiments. In this study, we propose an off-line training method based on the conventional method by collecting data on and around the target path at a time. In the proposed method, a robot is placed on the path and collects camera images and target angular velocity as teacher data. The robot is trained off-line based on these data, and after training, the robot moves autonomously using the training output from the camera images as input.

keywords: End-to-End Learning, Navigation, Offline

# 目次

第 1 章	序論	1
1.1	背景 . . . . .	1
1.2	目的 . . . . .	4
1.3	論文構成 . . . . .	4
第 2 章	要素技術	5
2.1	地図を用いたルールベース制御器によるナビゲーション . . . . .	5
2.2	ディープラーニング . . . . .	6
2.3	end-to-end 学習 . . . . .	7
2.4	データセット . . . . .	8
2.5	オフライン学習 . . . . .	8
2.6	バッチ学習 . . . . .	8
第 3 章	従来研究	9
3.1	従来手法の概要 . . . . .	9
3.2	従来手法のシステム概要 . . . . .	11
3.3	ネットワークの構造 . . . . .	12
第 4 章	提案手法	13
4.1	手法 . . . . .	13
第 5 章	実験	15
5.1	実験条件 1 . . . . .	15

目次	vi
5.1.1 実験目的 . . . . .	15
5.1.2 実験装置 . . . . .	15
5.1.3 実験方法 . . . . .	16
5.1.4 実験結果 . . . . .	17
5.1.5 考察 . . . . .	18
5.2 実験条件 2 . . . . .	20
5.2.1 実験方法 . . . . .	20
5.2.2 実験結果 . . . . .	21
5.3 実験条件 3 . . . . .	24
5.3.1 実験結果 . . . . .	24
5.4 実験条件 4 . . . . .	27
5.4.1 実験結果 . . . . .	28
第 6 章 結論	29
参考文献	30
付録	32
謝辞	33

# 目次

1.1	Training the neural network from [1] . . . . .	1
1.2	Map based navigation using navigation indoors from [2] . . . . .	2
1.3	Systems that imitation learning for map-based navigation from [2] . . . .	2
1.4	Procedure for visualizing the output of the learning machine from [3] . .	3
2.1	Map based navigation using navigation package . . . . .	5
2.2	Structure of Deep Learning . . . . .	6
2.3	Structure of general Learning . . . . .	7
2.4	Structure of end-to-end Learning . . . . .	7
2.5	MNIST dataset from [4] . . . . .	8
3.1	Conventional method system . . . . .	10
3.2	The conventional method collects the navigation actions apart from the robot's actions from [2] . . . . .	10
3.3	Systems that imitation learning for map-based navigation from [5] . . . .	11
3.4	Structure of network . . . . .	12
4.1	Method of collecting data around the target route . . . . .	14
5.1	Experimental environment in simulator . . . . .	15
5.2	Course to collect data . . . . .	16
5.3	Turtlebot3 waffle with 3 cameras . . . . .	16
5.4	Method of collecting data around the target route . . . . .	17

5.5	Failure point of the experiment . . . . .	18
5.6	Loss value in the experiment1 . . . . .	19
5.7	Loss value in the experiment2 . . . . .	19
5.8	Loss value in the experiment3 . . . . .	19
5.9	Histogram of collected angular velocities in the experiment1 . . . . .	20
5.10	Method of collecting data around the target route . . . . .	20
5.11	Failure point of the experiment2-1 . . . . .	21
5.12	Histogram of collected angular velocities in the experiment2-1 . . . . .	22
5.13	Loss value in the experiment1 . . . . .	23
5.14	Loss value in the experiment2 . . . . .	23
5.15	Loss value in the experiment3 . . . . .	23
5.16	Failure point of the experiment2-2 . . . . .	24
5.17	Loss value in the experiment1 . . . . .	26
5.18	Loss value in the experiment2 . . . . .	26
5.19	Loss value in the experiment3 . . . . .	26
5.20	Histogram of collected angular velocities in the experimental condition 3 . . . . .	27



# 表目次

5.1	Number of successes in the experiment1 . . . . .	17
5.2	Number of failures in the experiment . . . . .	18
5.3	Number of successes in the experiment . . . . .	21
5.4	Number of failures in the experiment . . . . .	21
5.5	Number of successes in the experiment2-2 . . . . .	24
5.6	Number of failures in the experiment . . . . .	25
5.7	Number of successes in the experiment . . . . .	28

# 第 1 章

## 序論

### 1.1 背景

近年, 様々なセンサを用いた自律移動に関する研究が活発に行われており, その中で視覚を入力とした end-to-end 学習により自律走行した例もある. 例えば, Bojaski らは Fig. 1.1 に示すシステムでカメラ画像と人が操作するステアリングの角度を end-to-end 学習することで, 自律走行する手法を提案した [1].



Fig. 1.1: Training the neural network from [1]

岡田らは Fig. 1.2 のように地図ベースのナビゲーションによる出力を模倣することで、経路追従行動を獲得した [2]. Fig. 1.3 に示すような, LiDAR, オドメトリを入力としたナビゲーションの出力を end-to-end で模倣学習し, 学習後はカメラ画像を入力とした学習器の出力により, 一定の経路において周回が可能であることが確認された.



Fig. 1.2: Map based navigation using navigation indoors from [2]

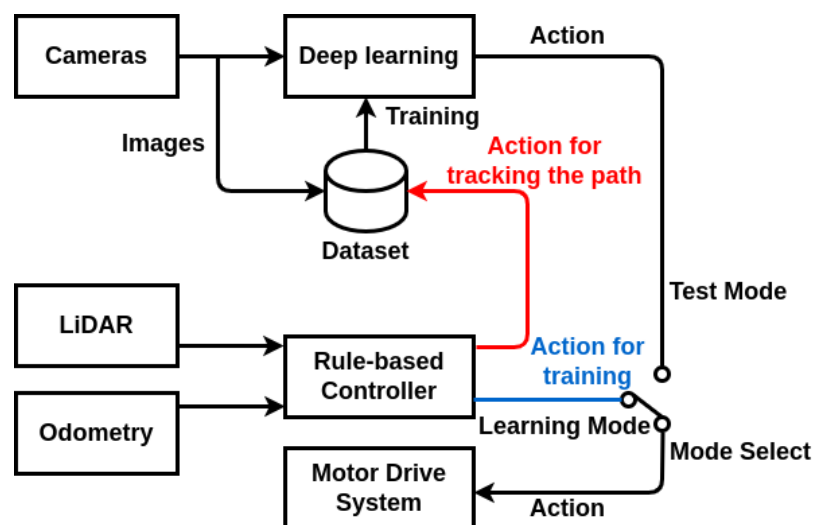


Fig. 1.3: Systems that imitation learning for map-based navigation from [2]

また, 清岡ら [3] により, Fig. 1.4 に示すような手法を用いて, 経路上だけでなく経路から離れた状態も学習することが, 経路追従行動を模倣する上で有効であることが示された.

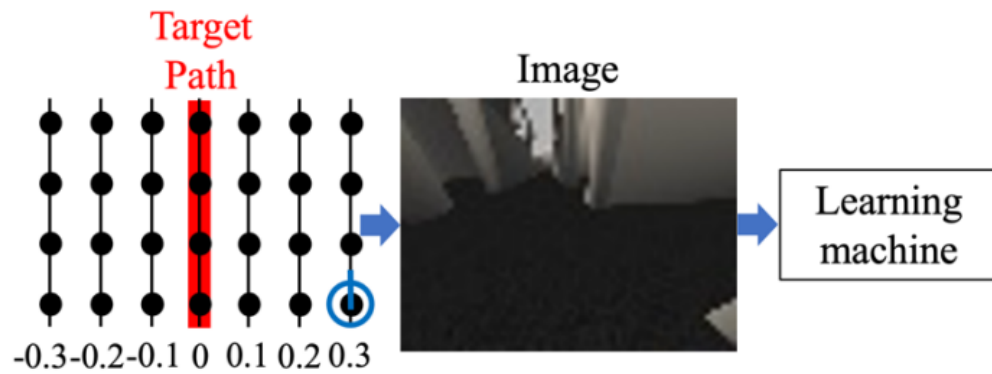


Fig. 1.4: Procedure for visualizing the output of the learning machine from [3]

以上で述べたように, カメラ画像を入力とした学習器の出力により, ロボットが学習した経路を周回可能であることが示されている.

次に, 岡田らと清岡ら (以下「従来研究」と称する) の提案手法を基に, 新たなデータセットの収集方法を提案する.

## 1.2 目的

本研究では, 先行研究を基にオフラインでデータを収集して学習を行う手法を提案する. これにより, 経路追従が可能であるかをシミュレータを用いた実験を通して検証する.

## 1.3 論文構成

本論文の構成は以下に述べる通りである. 第 1 章では, 研究を行う背景や目的を述べた. 第 2 章では, 研究に関連する要素技術, 第 3 章では, 従来手法について説明する. そして, 第 4 章では, 提案手法について説明し, 第 5 章では, 本研究の結論を述べる.

## 第 2 章

# 要素技術

### 2.1 地図を用いたルールベース制御器によるナビゲーション

教師信号としている地図を用いたルールベース制御器によるナビゲーションについて説明する。このナビゲーションには、ROS のパッケージである navigation[6] を使用している。移動ロボットは、Fig. 2.1 のように LiDAR のスキャンデータやオドメトリを入力として自己位置推定と経路計画を行い、これらに基づいて自律走行をする。また、自己位置推定には、amcl(Adaptive Monte Carlo Localization)、経路計画とモータ指令には move\_base[6] を使用している。



Fig. 2.1: Map based navigation using navigation package

## 2.2 ディープラーニング

ディープラーニングとは、人間の神経細胞を模したネットワーク構造のことである。主に、入力層と出力層、その間に中間層（隠れ層）という構成である。中間層を多層化することで、複雑な入力情報を処理し、パターンを認識することや、ルールを読み解くことができる。近年では、画像や物体認識、自然言語処理などで活用されている。Fig. 2.2 に構造の一例を示す。



Fig. 2.2: Structure of Deep Learning

## 2.3 end-to-end 学習

end-to-end 学習とは、入力から出力までの流れを一括に学習することができる手法である。例として、画像中からの文字認識を行う処理を挙げる。一般的な処理では、Fig. 2.3 のように画像から文字検出を行い、その後に文字分割、最終的に文字認識をする。しかし、end-to-end 学習では、Fig. 2.4 に示すような入力から出力までの流れを一括して学習することができる。

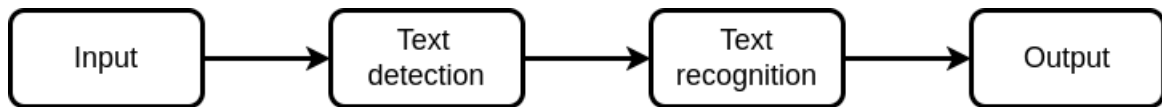


Fig. 2.3: Structure of general Learning



Fig. 2.4: Structure of end-to-end Learning



## 2.4 データセット

データセットとは、学習に使用する学習（訓練）データの集合のことである。例として、Fig. 2.5 に示すような 0 から 9 の手書きで書かれた数字の画像セットである MNIST が挙げられる。機械学習や画像認識において多く利用されており、訓練画像 6000 枚とテスト画像 1000 枚で構成されている。



Fig. 2.5: MNIST dataset from [4]

## 2.5 オフライン学習

オフライン学習とは、あらかじめ用意したデータセットを使用して学習を行うことである。これに対して、先行研究用に用いたオンライン学習とは、タスクを行いながらデータ収集をし、そのデータを使用して学習することを指す。

## 2.6 バッチ学習

バッチ学習とは、訓練データを一括で処理する学習方法である。特徴として、一度に大量のデータを扱うことができるため学習の進行が安定しやすく、訓練データに異常データが混じっていても受ける影響が小さくて済むなどが挙げられる。

## 第 3 章

# 従来研究

### 3.1 従来手法の概要

従来手法 [2] では, 地図を用いたルールベース制御器によるナビゲーションの走行を模倣し, 視覚に基づく経路追従行動を獲得した. 従来手法のシステム概要を Fig. 3.1 に示す. 学習時, 移動ロボットは Fig. 3.1(a) に示すように LiDAR とオドメトリを入力とする地図を用いたルールベース制御器によるナビゲーションで走行する. 同時に, 学習器はカメラ画像とナビゲーションの出力であるロボットの目標角速度を end-to-end 学習する. 学習後は, Fig. 3.1(b) のようにカメラ画像のみを入力とした学習器の出力により走行する.

また, 1.1 章でも述べたように, 目標経路より離れた位置から経路に戻る学習をすることが経路追従をする上で有効である. そのためには, 経路から一度外れる必要がある. しかし, それでは経路から外れる行動も学習してしまう. そこで, 従来手法では, 学習のデータセットに利用する行動と, 学習時にロボットを制御する行動を別々に扱うことができる. これにより, Fig. 3.2 に示すように経路から離れた位置から経路に戻る行動を学習することができる.

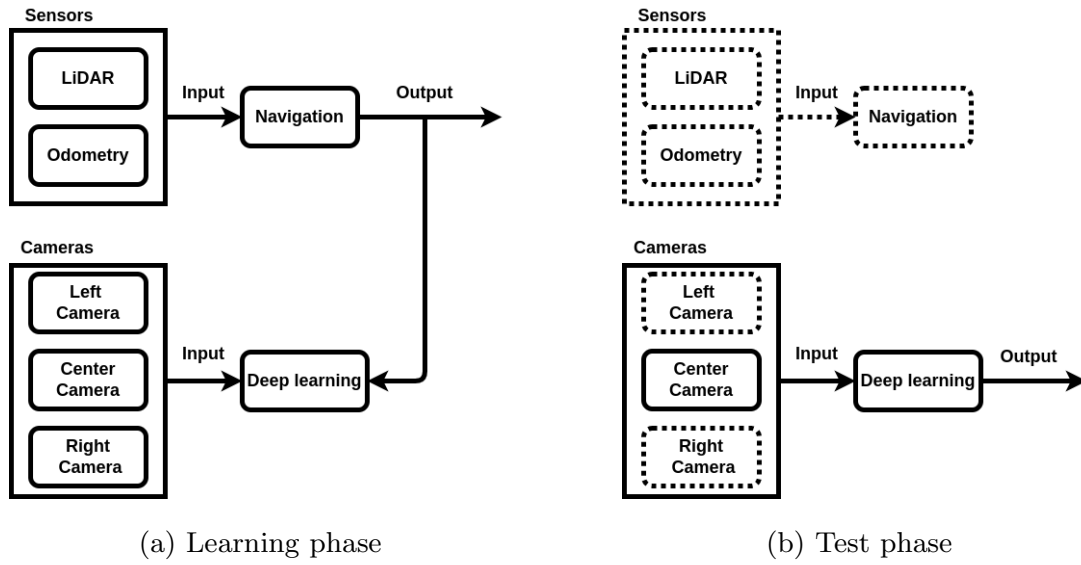


Fig. 3.1: Conventional method system

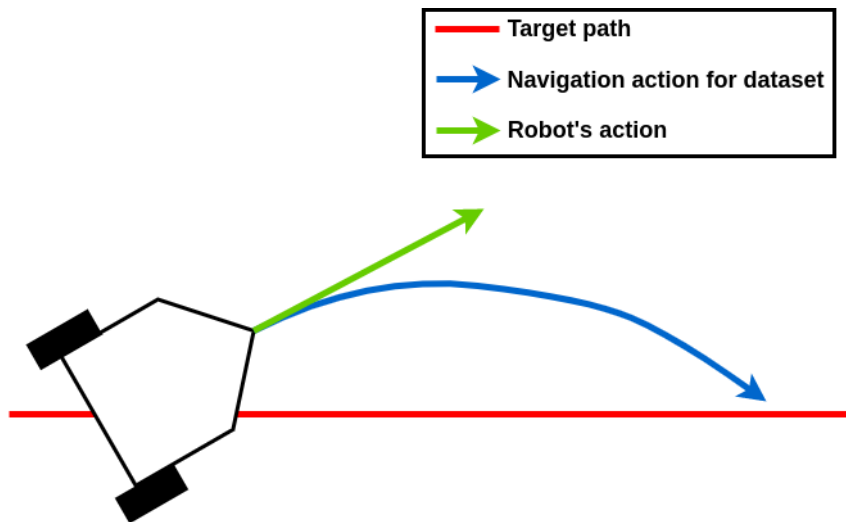


Fig. 3.2: The conventional method collects the navigation actions apart from the robot's actions from [2]

## 3.2 従来手法のシステム概要

次に, 3.1 章を基に構築された従来手法のシステムを Fig. 3.3 に示す. システムでは, LiDAR, オドメトリを入力としたナビゲーションの出力である角速度を学習器とモータ駆動系に与える. ナビゲーションの角速度は, ROS のパッケージである navigation[6] により計算される. また, 学習器には, カメラ画像を  $64 \times 48$  にリサイズした画像を入力し, ナビゲーションの角速度を出力して, 0.2s の周期で end-to-end 学習する.

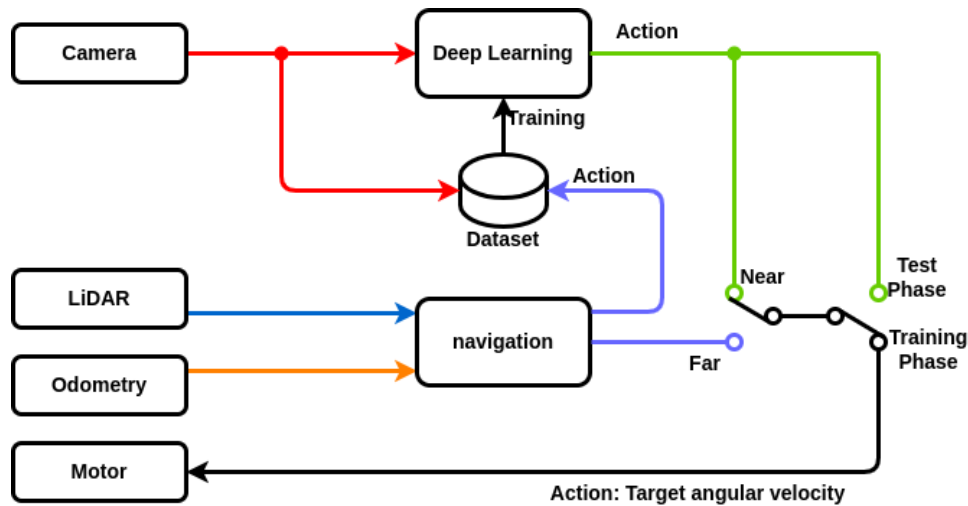


Fig. 3.3: Systems that imitation learning for map-based navigation from [5]

### 3.3 ネットワークの構造

Fig. 3.4 に従来研究で用いたネットワークの構造を示す．構造は、入力層 1, 畳み込み層 3, 全結合層 2, 出力層 1 の計 7 層から構成されている．

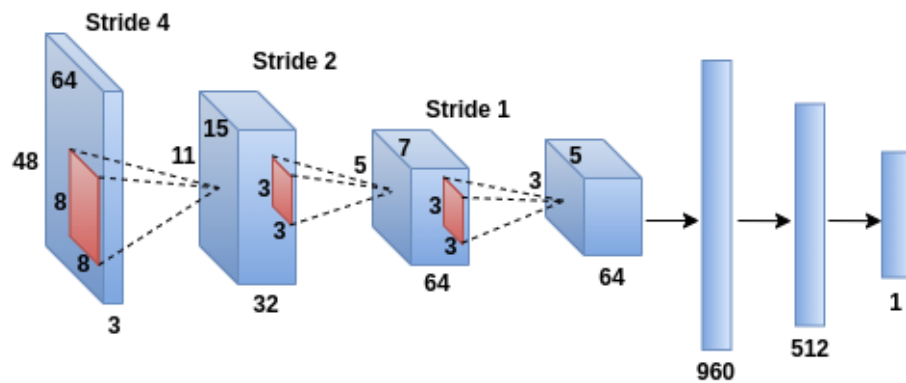


Fig. 3.4: Structure of network

以上のようにして、従来研究ではオンラインでデータを収集し学習を行っていた．しかし、学習器の出力で自律走行をするためには、何周もロボットを走行させて学習する必要がある．これでは、走行中にコースアウトしないか監視する必要や、時間がかかるといった問題点がある．

これらを踏まえて、本研究では、オフラインでデータセットを収集して訓練する手法を試みる．

## 第 4 章

# 提案手法

本章では, 先行研究を基にしたオフラインでデータを収集し訓練する手法を提案する.

### 4.1 手法

Fig. 4.1 にデータの収集方法を示す. 赤色の線である目標経路から平行に (例:  $\pm 0.10$ ,  $\pm 0.20$ ,  $\pm 0.30\text{m}$ ) 離れた座標にロボットを配置する. そして, その座標ごとに目標経路に沿った向きを基準として  $\pm 5$  度傾けて,  $64 \times 48$  のカメラ画像 (RGB 画像) とルールベース制御器によるナビゲーションの出力である角速度を Fig. 4.1 のように収集する. ロボットの進行方向に対する並進速度は  $0\text{m/s}$  であるが, データセットにはナビゲーションの出力である角速度がロボットに与えられる.

このように, ロボットを走行させることなく, 目標経路上及び周辺に配置することで, 一度に大量のデータを収集することができる. その後, 収集したデータを使用してオフラインで学習を行う.

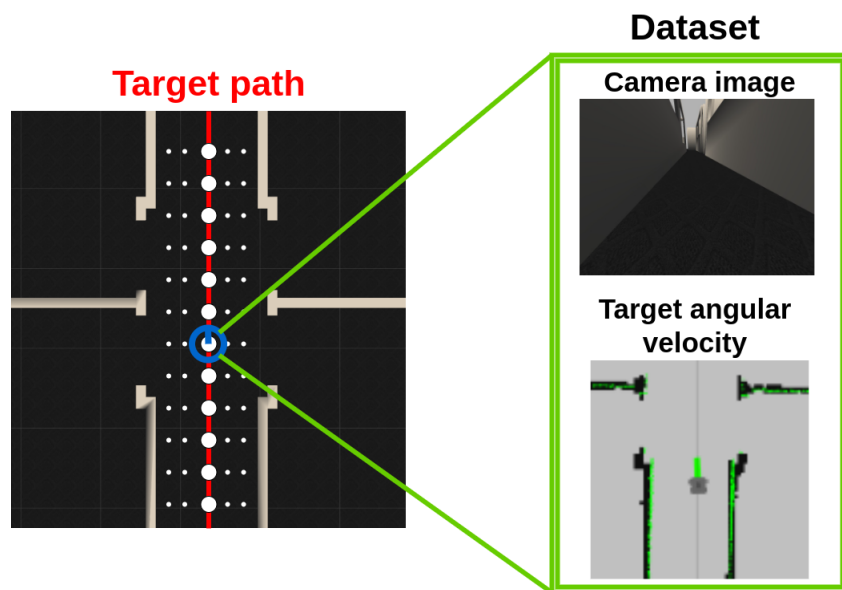


Fig. 4.1: Method of collecting data around the target route

## 第 5 章

# 実験

### 5.1 実験条件 1

#### 5.1.1 実験目的

シミュレータ上で実験を行い, 提案手法の有効性を検証する.

#### 5.1.2 実験装置

実験は, Fig. 5.1 に示す Gazebo[7] の Willow Garage[8] で Fig. 5.2 に示すコースで一周行う. また, ロボットモデルには Fig. 5.3 に示すようなカメラを 3 つ搭載した Turtlebot3[9] を用いた.

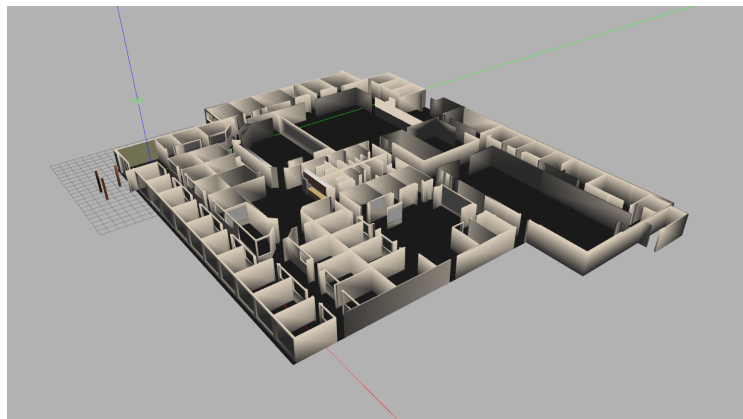


Fig. 5.1: Experimental environment in simulator



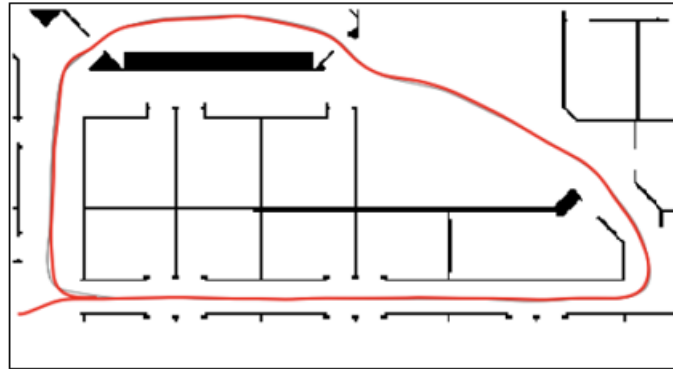


Fig. 5.2: Course to collect data



Fig. 5.3: Turtlebot3 waffle with 3 cameras

### 5.1.3 実験方法

データ収集フェーズ データの収集方法について述べる. Fig. 5.4 にデータの収集方法を示す. 赤色の線である目標経路から平行に  $\pm 0.10$ ,  $\pm 0.20$ ,  $\pm 0.30\text{m}$  離れた座標にロボットを配置する. そして, その座標ごとに目標経路に沿った向きを基準として  $\pm 5$  度傾けて, カメラ画像とナビゲーションの出力である角速度を収集する. これを Fig. 5.2 に示した経路で実験を行う.

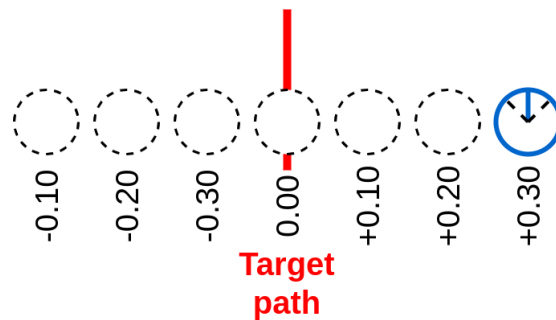


Fig. 5.4: Method of collecting data around the target route

訓練フェーズ データ量 2658, 先行研究に倣ってバッチ数 8, 実験 1 では 4000step(従来研究のシミュレータ上での学習時より), 実験 2 では 8000step, 実験 3 では 10000step(従来研究の実環境での学習時より) 学習した.

テストフェーズ Fig. 5.2 に示すコースで, 10 回走行させる. 壁に衝突せずに一周できた場合を成功とし, 壁に激突したり, コースアウトして経路に復帰できなかった場合を失敗とした.

#### 5.1.4 実験結果

実験結果は, 5.1 のようになった. また, 失敗箇所は Fig. 5.5, 実験ごとの失敗箇所における失敗回数は 5.2 のようになった.

Experiments	Number of successes
Exp.1	0/10
Exp.2	0/10
Exp.3	0/10

Table 5.1: Number of successes in the experiment1

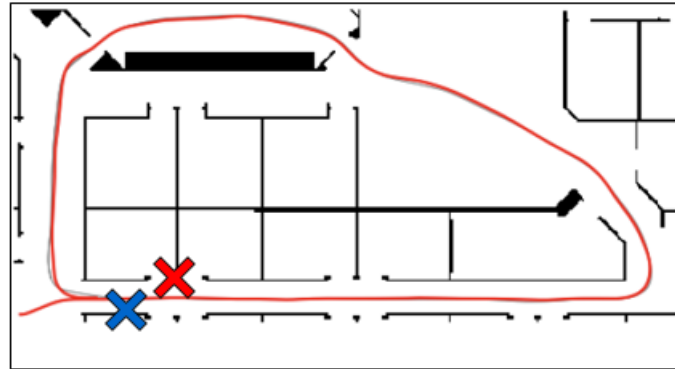


Fig. 5.5: Failure point of the experiment

Experiments	Number of failures with red x	Number of failures with blue x
Exp.1	1	9
Exp.2	0	10
Exp.3	0	10

Table 5.2: Number of failures in the experiment

### 5.1.5 考察

ステップ数を増やしても成功回数は増えなかった。また、目標経路から離れた際に戻る挙動や、壁に近づきすぎた際に避ける挙動も見られなかった。ここで、訓練させた際の各実験ごとの loss を Fig. 5.6, Fig. 5.7, Fig. 5.8 に示す。実験 1 から実験 3 全てにおいて、オーバーシュートしていると考えられる。そこで、収集した角速度を Fig. 5.9 のようにヒストグラムにした。これより、経路上及び経路周辺のデータである  $0\text{rad/s}$  から  $0.1\text{rad/s}$  が全体の 4 割程度しかないことが分かる。以上を踏まえて、経路上及び経路周辺のデータが多い方が、経路追従の成功回数が増えるのではないかと考えた。

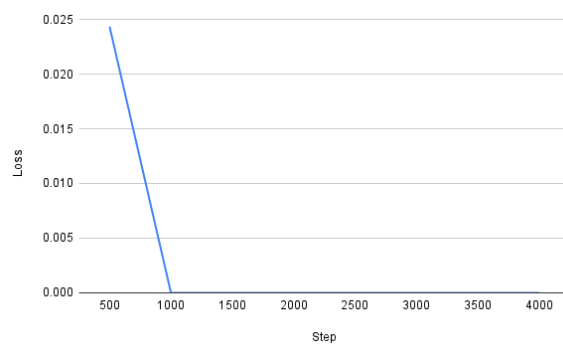


Fig. 5.6: Loss value in the experiment1

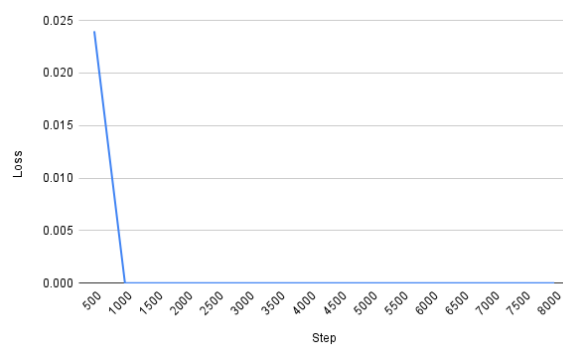


Fig. 5.7: Loss value in the experiment2

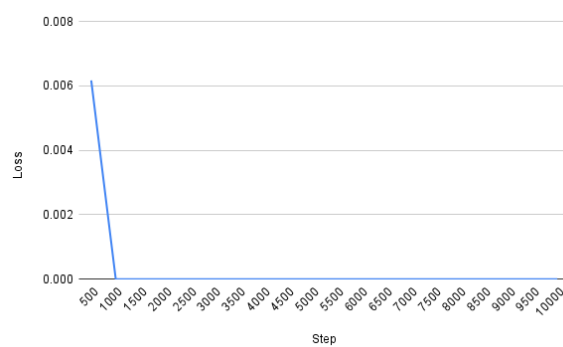


Fig. 5.8: Loss value in the experiment3

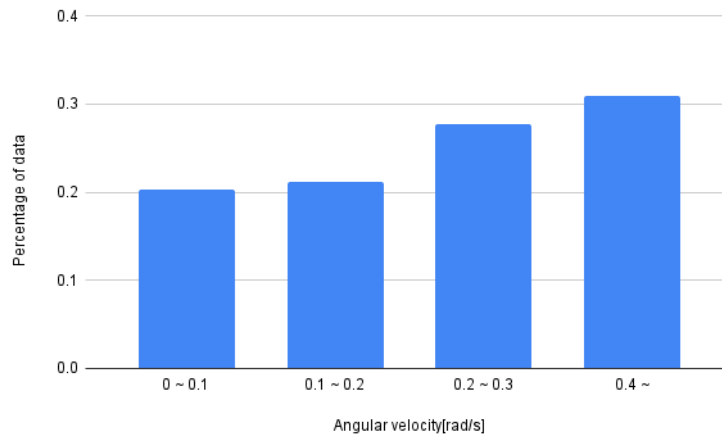


Fig. 5.9: Histogram of collected angular velocities in the experiment1

## 5.2 実験条件 2

実験目的, 実験装置, テストフェーズは実験条件 1 と同様である.

### 5.2.1 実験方法

データ収集フェーズ 実験条件 1 を踏まえて, 経路周辺のデータを多く取得する手法を試みる. Fig. 5.10 にデータの収集方法を示す. 赤色の線である目標経路から平行に  $\pm 0.01$ ,  $\pm 0.02$ ,  $\pm 0.04$ ,  $\pm 0.06$ ,  $\pm 0.08$ ,  $\pm 0.10$ ,  $\pm 0.15$ ,  $\pm 0.20$ ,  $\pm 0.30$ m 離れた座標にロボットを配置する. そして, 手法 1 と同様にロボットを傾けて画像と角速度を Fig. 4.1 のように収集する. これを Fig. 5.2 に示すコースで一周行う.

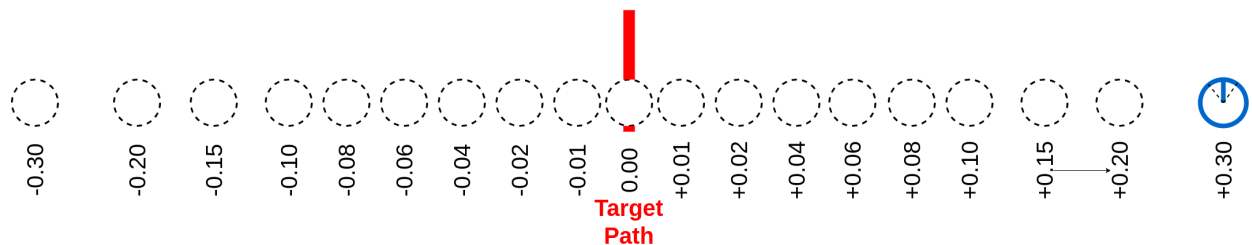


Fig. 5.10: Method of collecting data around the target route

訓練フェーズ データ量 7242, バッチ数 8, ステップ数は実験条件 1 と同様に 4000step, 8000step, 10000step 学習した.

### 5.2.2 実験結果

実験結果は, 5.3 のようになった. また, 失敗箇所は Fig. 5.11, 実験ごとの失敗箇所における失敗回数は 5.4 であった.

Experiments	Number of successes
Exp.1	0/10
Exp.2	0/10
Exp.3	0/10

Table 5.3: Number of successes in the experiment

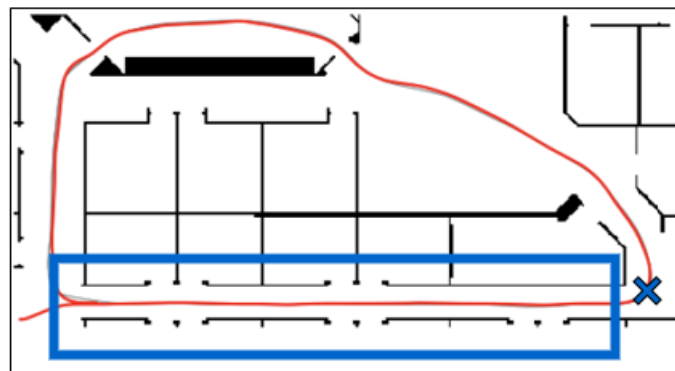


Fig. 5.11: Failure point of the experiment2-1

Experiments	Number of failures with blue x
Exp.1	10
Exp.2	10
Exp.3	10

Table 5.4: Number of failures in the experiment

考察 収集した角速度を Fig. 5.12 のようにヒストグラムにした。経路上及び経路周辺のデータが Fig. 5.9 と比べて多くなっている。これにより, Fig. 5.11 の青枠の区間において, 壁に衝突することなく走行できていることが分かる。しかし, 経路上や経路周辺のデータを増やしたり, ステップ数を増やしたりしても成功回数は増えなかった。ここで, 訓練させた際の各実験ごとの loss を Fig. 5.13, Fig. 5.14, Fig. 5.15 に示す。実験条件 1 と同様に実験 1 から実験 3 全てにおいて, オーバーシュートしていると考えられる。そこで, 先行研究のオンライン学習では計算のリソースなどの観点からバッチ数を 8 にしていたが, 提案手法ではオフラインで学習を行うため, バッチ学習に変更する。これにより, 一度に大量のデータを扱えるため最適解に辿り着くことができ, 成功回数が増えるのではないかと考えた。

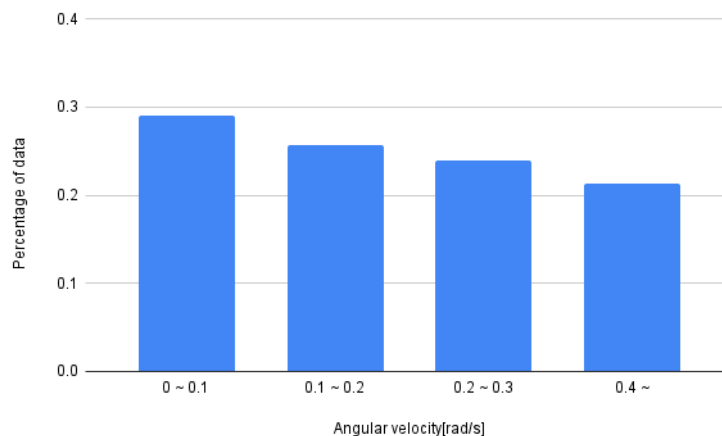


Fig. 5.12: Histogram of collected angular velocities in the experiment2-1

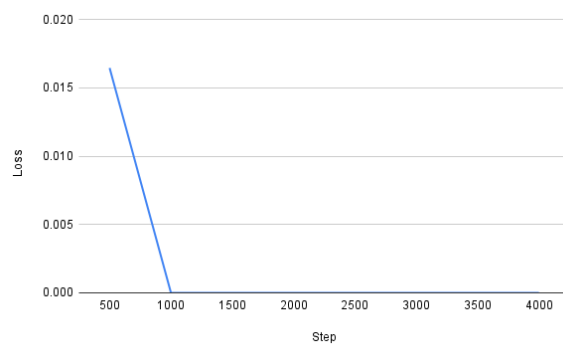


Fig. 5.13: Loss value in the experiment1

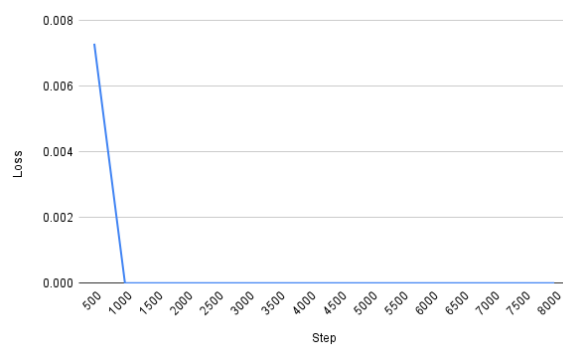


Fig. 5.14: Loss value in the experiment2

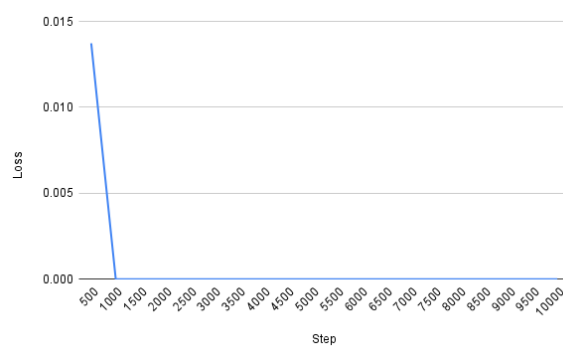


Fig. 5.15: Loss value in the experiment3



### 5.3 実験条件 3

オフラインで学習を行うメリットを活かして, ここではバッチ学習を用いることで成功回数が増えるか検証する. 実験目的, 実験装置, データ収集フェーズは実験条件 2 と同様である.

訓練フェーズ バッチサイズはデータ量と同じ 7242, バッチ学習で実験条件 1, 2 と同様に 4000step, 8000step, 10000step 学習した.

#### 5.3.1 実験結果

実験結果は, 5.5, 失敗箇所 Fig. 5.16, 実験ごとの失敗箇所における失敗回数は 5.6 のようになった.

Experiments	Number of successes
Exp.1	0/10
Exp.2	1/10
Exp.3	1/10

Table 5.5: Number of successes in the experiment2-2

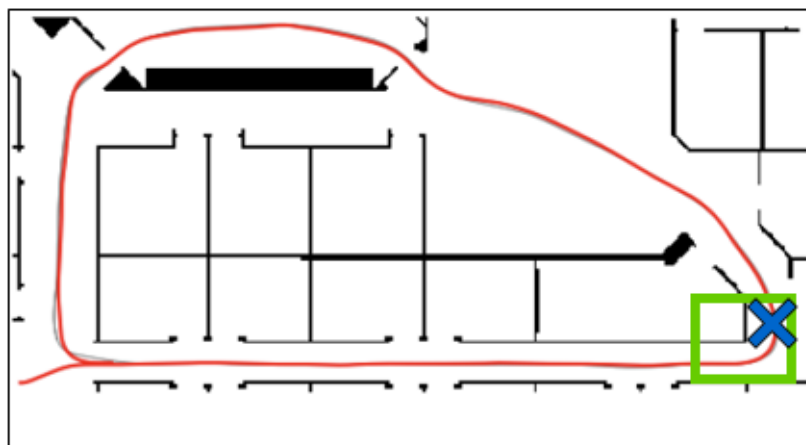


Fig. 5.16: Failure point of the experiment2-2

Experiments	Number of failures with blue x
Exp.1	10
Exp.2	9
Exp.3	9

Table 5.6: Number of failures in the experiment

考察 4000step では成功回数が 0/10 だったが, 8000step, 10000step にすることで成功回数が 1/10 になり, 目標経路を一周することができた. このことから, 訓練する際に異常データから受ける影響が少なく済むバッチ学習を用いて, ステップ数を増やすことで成功回数が増えることを示せた. また, loss は Fig. 5.17, Fig. 5.18, Fig. 5.19 のようになった. 失敗箇所は Fig. 5.16 の緑枠内の角で 4000step, 8000step, 10000step 全てで曲がりきれず, 青×に示す箇所でコースアウトした. しかし, 実験条件 2 と比べて角で曲がる挙動が見られた. 以上から, 成功回数は増えたが, 実験条件 3 では成功回数が十分であるとは言えず, ほとんどで角を曲がり切れていない. そこで, Fig. 5.16 の緑枠内のデータを増やすことで角を曲がり切ることができ, 成功回数が増えるのではないかと考えた.

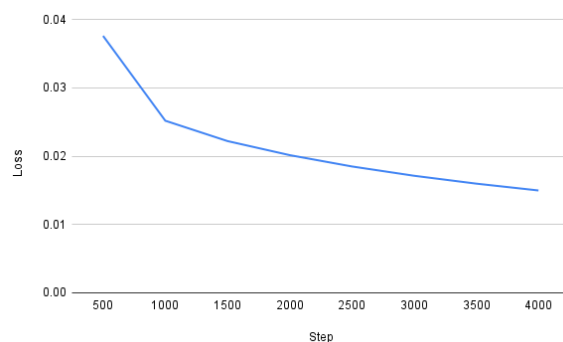


Fig. 5.17: Loss value in the experiment1

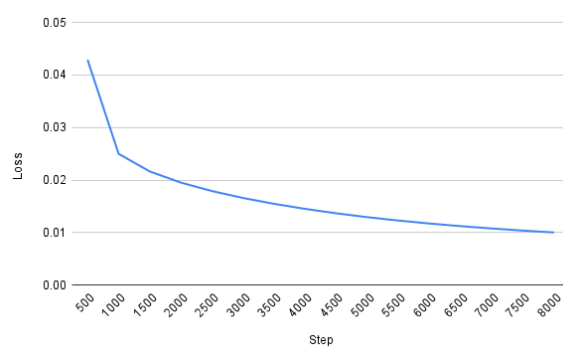


Fig. 5.18: Loss value in the experiment2

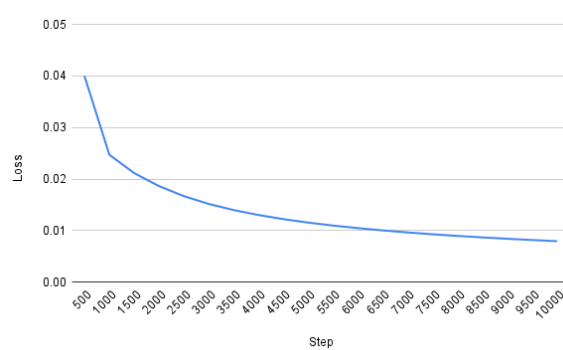


Fig. 5.19: Loss value in the experiment3

次の添削までに、角において並進方法の配置密度を狭めてデータ量を増やすことで成功回数が増えるか検証します。

## 5.4 実験条件 4

ここでは、角のデータを増やすことで成功回数が増えるかどうかを検証する。実験条件 3 で用いた角速度のデータを Fig. 5.20 に示す。直進に近いデータ ( $-0.2\text{rad/s}$  から  $0.2\text{rad/s}$ ) の合計は 4227、角を左折する際の角速度のデータ ( $0.3\text{rad/s}$  以上) の合計は 1280 である。直進は失敗していないことから、左折する際のデータも直進と同量であれば曲がれるのではないかと考えた。よって、左折する際のデータ量を 4 倍して実験を行う。

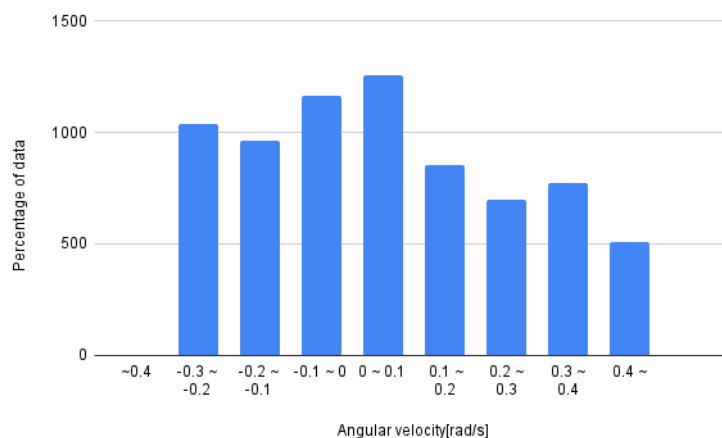


Fig. 5.20: Histogram of collected angular velocities in the experimental condition 3

**データ収集フェーズ** データの収集方法は、目標経路に対して平行な方向は実験条件 2, 3 と同様であるが、実験条件 4 では目標経路に沿った向きのロボットの配置間隔を変更する。実験条件 1 から実験条件 3 では目標経路に沿った向きで  $0.5\text{m}$  間隔でロボットを配置していた。しかし、実験条件 3 の Fig. 5.16 でも示したように、緑枠内において目標経路に沿った向きのロボットの配置間隔を  $0.25\text{m}$  に変更する。これにより、角においてのデータ量が増えるため、角を曲がることができ、成功回数が増えることが期待できる。

**訓練フェーズ** バッチサイズはデータ量と同じ?個、バッチ学習で 4000step, 8000stp, 10000step 学習した。

### 5.4.1 実験結果

実験結果は, 5.7 のようになった.

Experiments	Number of successes
Exp.1	/10
Exp.1	/10
Exp.1	/10

Table 5.7: Number of successes in the experiment

訓練 2-3, 実験 2-3 の結果, 考察は実験が終わり次第記入

## 第 6 章

# 結論

本研究では, 経路追従行動をカメラ画像を入力とした end-to-end 学習で模倣する岡田ら [2] と清岡ら [3] の手法を基に, 新たなデータセットの収集方法と収集したデータ量を増やすことで, 経路追従行動を獲得できる手法を提案した.

## 参考文献

- [1] Mariusz Bojarski et al. ””end to end learning for self-driving cars.””. arXiv: 1604.07316(2016).
- [2] 岡田眞也, 清岡優祐, 上田隆一, 林原靖男. ”視覚と行動の end-to-end 学習により経路追従行動をオンラインで模倣する手法の提案”. 計測自動制御学会 SI 部門講演会 SICE-SI2020 予稿集, pp.1147-1152(2020).
- [3] 清岡優祐, 岡田眞也, 岩井一輝, 上田隆一, 林原靖男. ”視覚と行動の end-to-end 学習により経路追従行動をオンラインで模倣する手法の提案- データセットと生成された経路追従行動の解析”. 計測自動制御学会 SI 部門講演会 SICE-SI2021 予稿集, pp.1072-1075(2021).
- [4] The mnist database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>. 最終閲覧日 2023 年 1 月 9 日.
- [5] 岡田眞也, 清岡優祐, 春山健太, 上田隆一, 林原靖男. ”視覚と行動の end-to-end 学習により経路追従行動をオンラインで模倣する手法の提案- “ 経路追従行動の修正のためにデータセットを動的に追加する手法の検討”. 計測自動制御学会 SI 部門講演会 SICE-SI2021 予稿集, pp.1066-1070(2021).
- [6] ros-planning, navigation. <https://github.com/ros-planning/navigation>. 最終閲覧日 2023 年 1 月 9 日.
- [7] gazebo. <http://gazebo-sim.org/>. 最終閲覧日 2023 年 1 月 9 日.
- [8] Koenig, Nathan, and Andrew Howard. ”design and use paradigms for gazebo, an open-source multi-robot simulator.”. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566). Vol. 3. IEEE, pp.2149-2154(2004). 最終閲覧日 2023 年 1 月 9 日.
- [9] Turtlebot3 robotis emanual.robotis. <https://emanual.robotis.com/docs/>. 最

終閲覧日 2023 年 1 月 9 日.

- [10] Jan-Aike Bolte and Andreas Bar, Daniel Lipinski, and Tim Fingscheidt. "towards corner case detection for autonomous driving". 2019 IEEE Intelligent Vehicles Symposium (IV) Paris, France. June 9-12, 2019.



# 付録

成功や失敗した際のロボットの走行の軌跡を掲載 (一例)

# 謝辞

本研究を進めるにあたり，1年に渡り，熱心にご指導を頂いた林原靖男教授に深く感謝いたします。また，日頃から研究へのアドバイス，指導，サポートしてくださった清岡優祐様，春山健太様，藤原柁様，白須和暉様，並びにロボット設計制御研究室の皆様には，心から深く感謝を申し上げます。