

修 士 論 文

視覚と行動の end-to-end 学習による経路追従行動の模倣 (データセットを収集してオフラインで訓練する手法の検討)

Imitation of path-tracking behavior by end-to-end learning of vision and action
(Investigation of a method to collect datasets and train them offline)

2025 年 1 月 16 日 提出

指導教員 林原 靖男 教授

千葉工業大学 先進工学研究科 未来ロボティクス専攻

23S1022 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ロボットに与えて学習させることで、経路追従行動をオンラインで模倣する手法を提案し、実験によりその有効性を確認してきた。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器の出力により自律移動させることで、手法の有効性を検証する。

キーワード: end-to-end 学習, ナビゲーション, 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 path-following behavior by training a robot with the output of a 2D-LiDAR-based autonomous mobile system as a teacher signal, and have confirmed 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, the robot is placed on the path, and camera images and target angular velocity are collected as teacher data. The effectiveness of the proposed method is verified by training the robot off-line based on these data, and after training, the robot moves autonomously by using the output of the trainer with 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	関連研究	2
1.3	目的	5
1.4	論文構成	5
第 2 章	要素技術	6
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	6
2.2	ディープラーニング	7
2.3	end-to-end 学習	8
2.4	データセット	9
2.5	オフライン学習	9
2.6	バッチ学習	9
第 3 章	従来手法 (オンライン手法)	10
3.1	オンライン手法の概要	10
3.2	ネットワークの構造	12
第 4 章	提案手法 (オフライン手法)	13
4.1	手法	13
第 5 章	実験	15
5.1	実験 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	17
5.1.4 実験結果	18
5.2 実験 2	21
5.2.1 実験方法	21
5.2.2 実験結果	22
第 6 章 結言	23
参考文献	24
付録	26
謝辞	29

1.1	Path-tracking by end-to-end imitation learning	2
1.2	Training the neural network from [1]	2
1.3	Framework overview from [1]	3
1.4	Robot navigation in a pedestrian environment from [1]	4
2.1	Map based navigation using navigation package	6
2.2	Structure of deep learning	7
2.3	Structure of general learning	8
2.4	Structure of end-to-end learning	8
2.5	MNIST dataset from [2]	9
3.1	System configuration during network training	11
3.2	System configuration after network training	11
3.3	Structure of network	12
4.1	Data collected by the simulator in the learning phase	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	20
5.8	Loss value in the experiment3	20
5.9	Method of collecting data around the target route	21

表目次

5.1	Number of successes in the batch learning	18
5.2	Number of failures in the experiment	19

第 1 章

序論

1.1 背景

近年, 配膳ロボットや警備案内ロボットなどの需要が高まり, 自律的に移動できることが求められている. これらのロボットは LiDAR, ホイールオドメトリ, IMU などの様々なセンサから得られるデータに基づいて作成された占有格子地図を用いてナビゲーションを行う.

また, 本研究グループでは, end-to-end 学習により, 視覚に基づく経路追従行動をオンラインで模倣する手法を提案し, その有効性を実験により検証してきた. 岡田らが提案した手法(以後, オンライン手法と呼ぶ)では Fig. 1.1 に示すような LiDAR やホイールオドメトリなどを入力として生成したナビゲーションの行動を, カメラ画像を入力とする行動に模倣する [3][4]. これにより, 地図を用いたナビゲーションと視覚に基づく経路追従行動の 2 つのナビゲーション手段が得られるため, 状況に応じて高い信頼性が見込まれる方を選択することで, 経路追従を継続できる可能性が高まる.

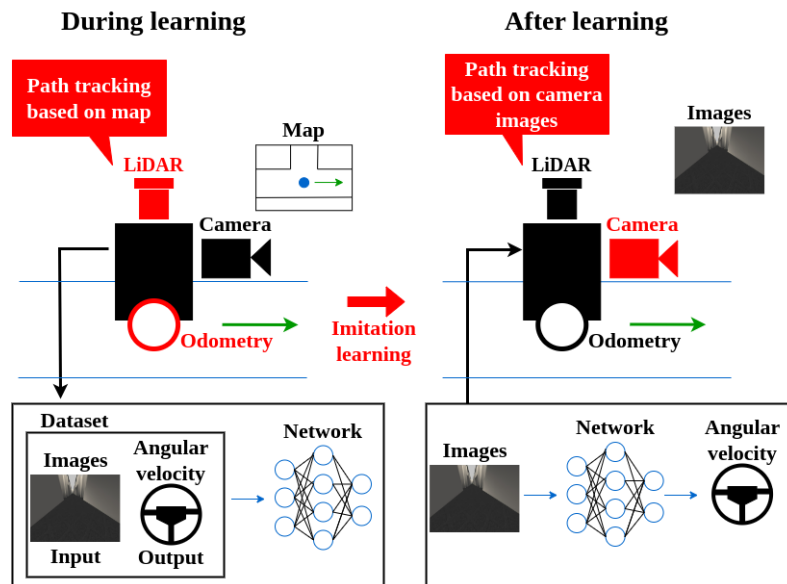


Fig. 1.1: Path-tracking by end-to-end imitation learning

1.2 関連研究

視覚を入力として, end-to-end 学習により経路追従を模倣する手法をいくつか紹介する.

Bojarski らは人が操作したステアリングの角度を end-to-end 学習することで経路追従する手法を提案した. Fig. 1.2 に示すように学習器に, 3 つのカメラ (左・中央・右) の画像と対応するステアリングの角度を入力する. 学習後は出力されたステアリングの角度のみで自律走行することが確認された [1].

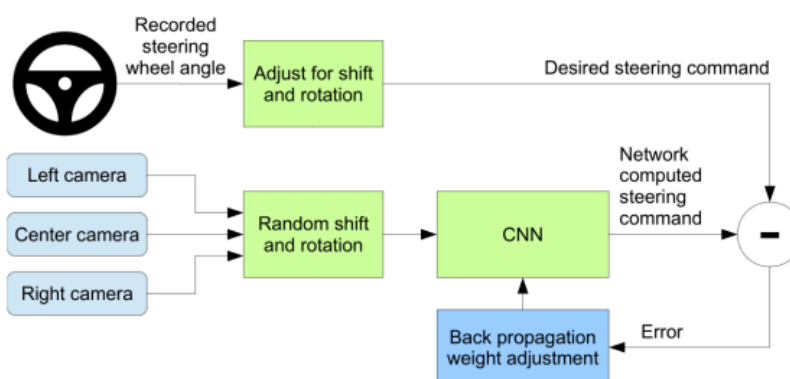


Fig. 1.2: Training the neural network from [1]

また, Jing Bi らは, 歩行者の多い環境における経路追従行動の模倣に取り組んだ. 具体的には, Fig. 1.3 に示すようなフレームワークを用いて, Fig. 1.4 のような歩行者のいる状況に合わせて, 人が操作しながらデータ収集を行う. ステージ 1 では「どの状況にいるのか」を判断する. ステージ 2 では分類結果に基づいて「具体的にどう動くか」を決定する. 経路から外れたり, 人と衝突したりなどのエラーに遭遇しそうな場合には人が介入を行う. そして, その介入の前後を含めてポリシーを更新する. これにより, 歩行者がいる場合でも経路追従できることが確認された.

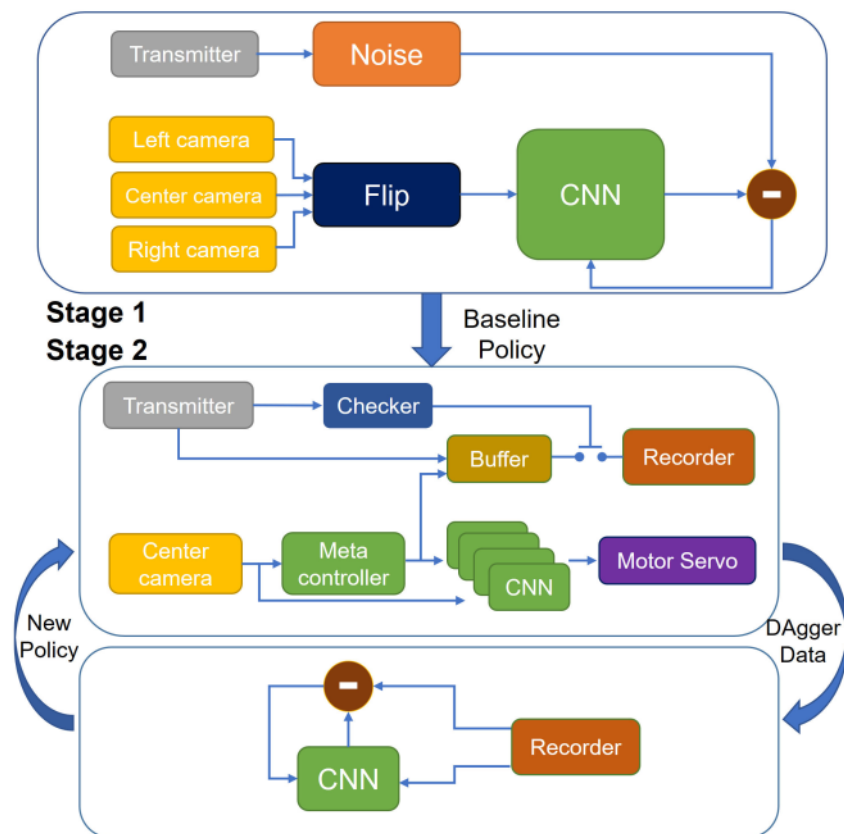


Fig. 1.3: Framework overview from [1]



Fig. 1.4: Robot navigation in a pedestrian environment from [1]

これらの手法が人の手動操作を模倣しているのに対して、オンライン手法は自動運転を模倣する点異なる。しかし、オンライン手法では、データ収集及び学習を行うために、ロボットを経路に沿って走行させ続けることが必要である。そのため、経路追従の成功率を上げるためにロボットを長時間走行させ続ける必要があり、それが問題となっていた。

1.3 目的

本論文では、事前に収集した画像と行動を用いて、経路追従行動をオフラインで学習する手法（以後、オフライン手法と呼ぶ）を提案する。これにより、オンライン手法で問題となっていた学習時間の短縮を目指す。なお、オフラインで模倣学習することは、他の研究でも行われていることであるが、そのデータセットを自動で収集する点が本手法の特徴である。

1.4 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。第 4 章では、提案手法について説明し、第 5 章では、実験について説明する。そして、第 6 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[5] 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[5] を使用している。

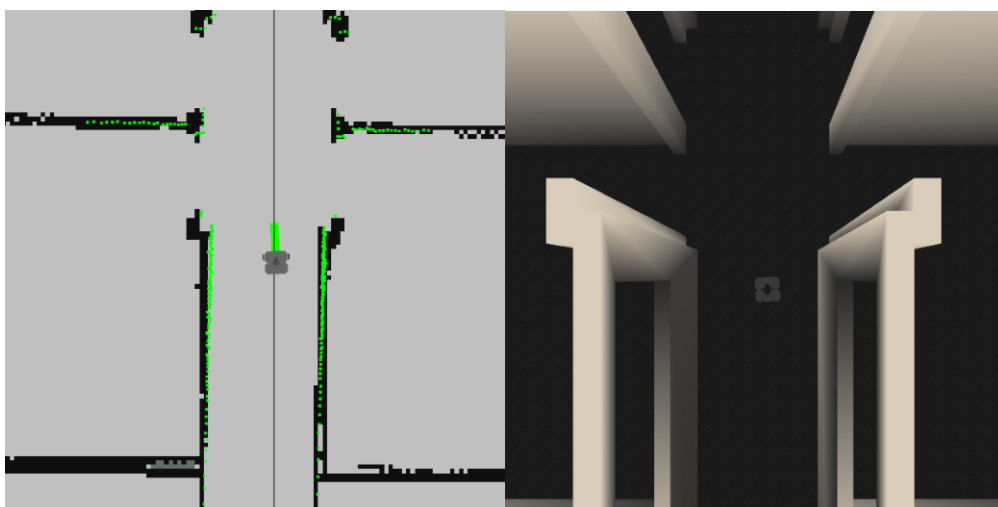


Fig. 2.1: Map based navigation using navigation package

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。

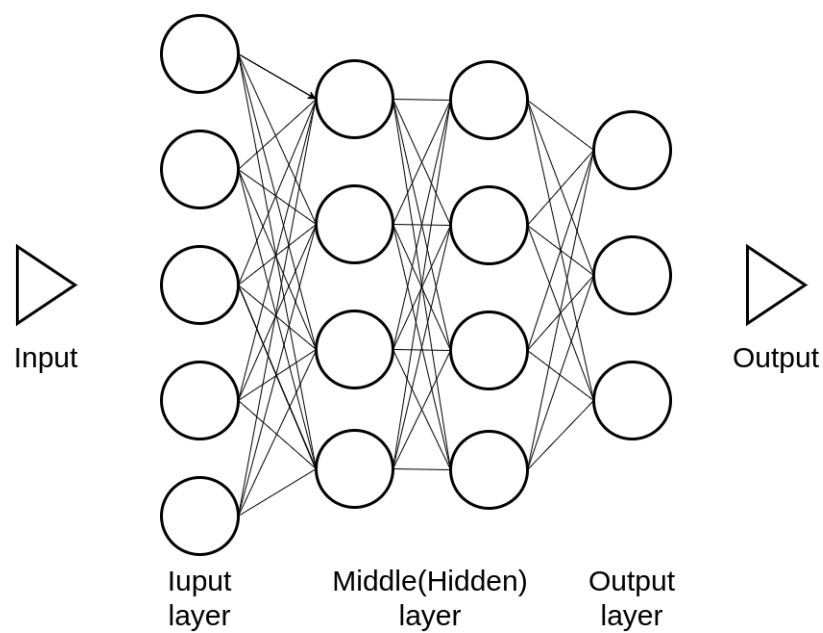


Fig. 2.2: Structure of deep learning

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。

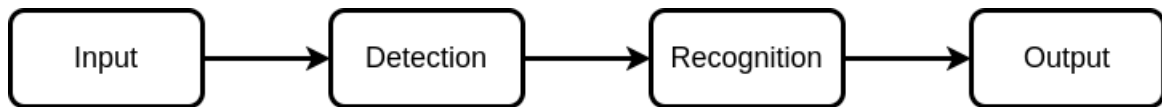


Fig. 2.3: Structure of general learning

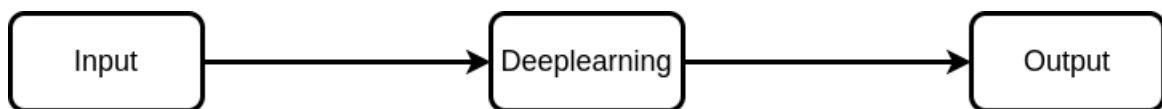


Fig. 2.4: Structure of end-to-end learning

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。

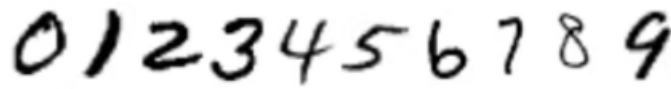


Fig. 2.5: MNIST dataset from [2]

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。また、バッチ学習ではバッチサイズがデータ数となることが多い。

第 3 章

従来手法（オンライン手法）

3.1 オンライン手法の概要

従来から提案してきたオンライン手法に関して述べる。オンライン手法では、地図を用いたルールベース制御器の出力を模倣して、経路追従行動を獲得する。Fig. 3.1 にシステム概要を示す。学習時は、LiDAR とオドメトリを入力として、ROS の navigation[7] により、目標角速度を求めて、ロボットを経路追従させる。同時に 64×48 にリサイズした 3 つのカメラ画像 (RGB 画像) を入力、目標角速度を出力とするデータを、データセットに加える。そのデータをランダムにピックアップしてリアルタイムに学習する。左右のカメラ画像に対する目標角速度には、それぞれ経路に戻るためのオフセットを加える。これらは経路から外れた場合に、経路に戻る行動が選択されるように加えている。

一定ステップ数の学習を行った後、学習したモデルで経路追従できるか確認する。Fig. 3.2 にシステム概要を示す。中央のカメラ画像を深層学習器に入力して、出力された目標角速度を用いてロボットを制御する。なお、目標の並進速度は $0.2[\text{m/s}]$ で一定とする。

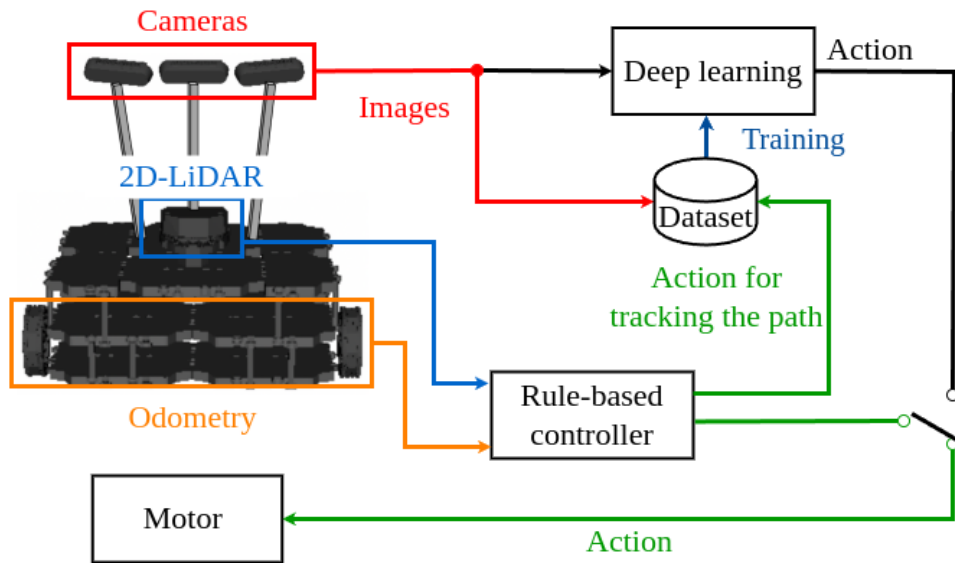


Fig. 3.1: System configuration during network training

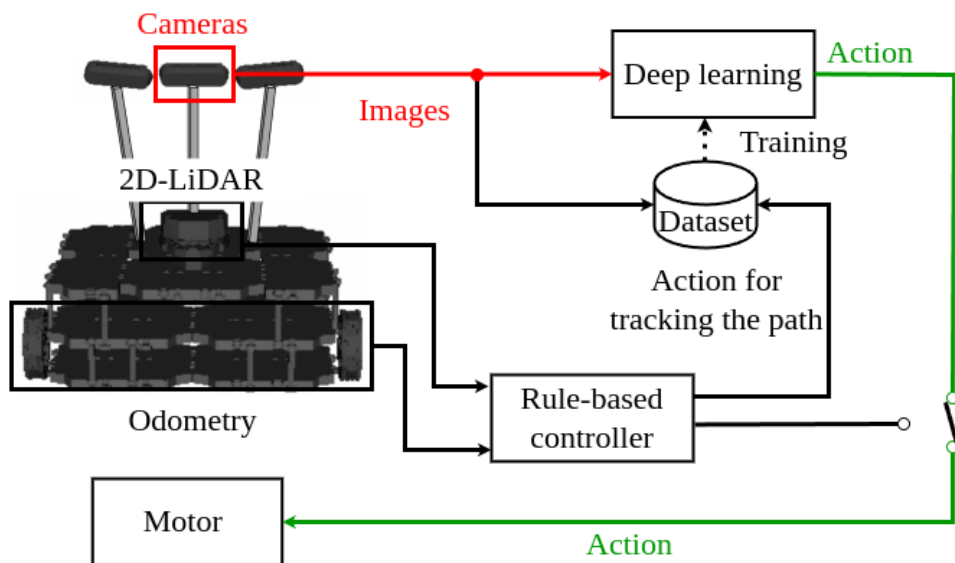


Fig. 3.2: System configuration after network training

3.2 ネットワークの構造

Fig. 3.3 に従来手法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。また、オンラインで学習が行えるように、ネットワークは畳み込みニューラルネットワーク (CNN) を基にしている。

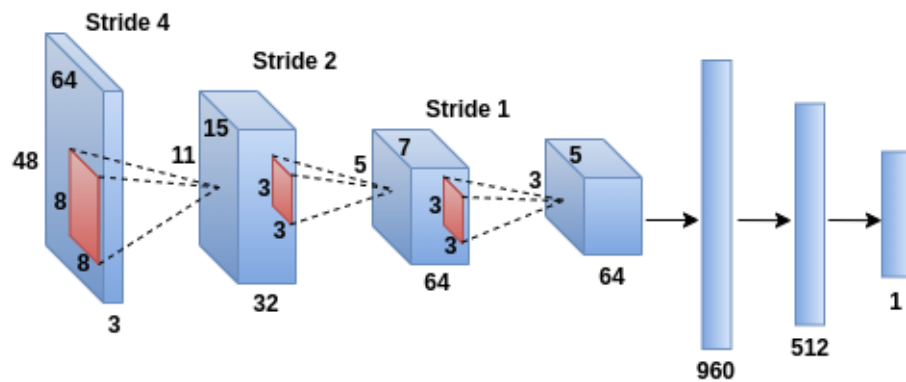


Fig. 3.3: Structure of network

第 4 章

提案手法（オフライン手法）

本章では、従来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

本研究で検証するオフライン手法に関して述べる。オンライン手法と比べて、オフライン手法は画像と目標角速度のデータを事前に収集して、学習するところが異なる。Fig. 4.1 にシミュレータを用いて収集するデータを示す。目標経路（赤線）から一定距離の位置にロボットを配置して、さらに目標経路の方向を基準として、ヨー方向に一定量回転する。その時の中央のカメラの画像と、目標角速度を収集してデータセットに加える。ちなみに、本手法でデータを収集するためには、非常に多くのロボットの置き直しをしなければならない。これを、実ロボットに应用する際には、1 台のロボットに複数のカメラを搭載して、経路から一定距離離れた画像を収集する。そうすることで、置き直ししなくても経路を走行すればデータを集められる。そのため、実ロボットへの応用も可能であると考える。このように収集したデータセットを用いてオフラインで学習する。なお、リアルタイム性に配慮して、オンライン手法ではバッチサイズ 8 のミニバッチ学習を行っていたが、オフライン学習ではリアルタイム性は必要ないため、バッチ学習を行う。

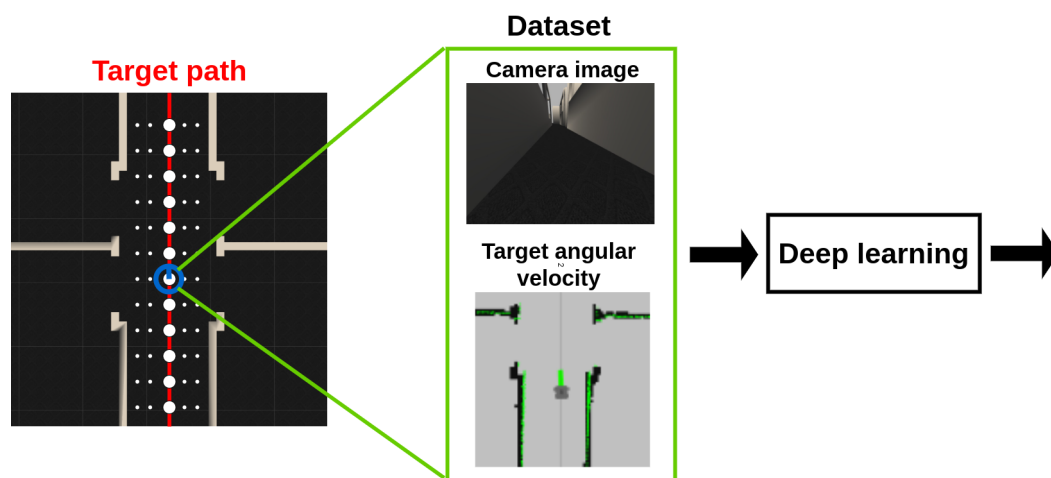


Fig. 4.1: Data collected by the simulator in the learning phase

第 5 章

実験

5.1 実験 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[6] の Willow Garage[7] 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[8] を用いた.

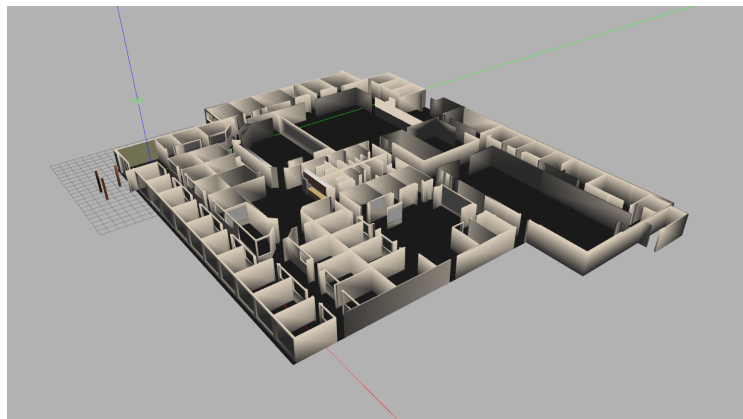


Fig. 5.1: Experimental environment in simulator

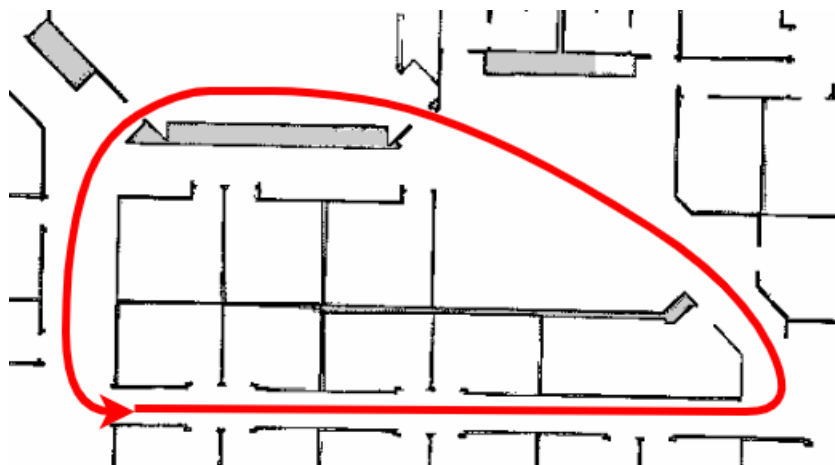


Fig. 5.2: Course to collect data



Fig. 5.3: Turtlebot3 waffle with 3 cameras

5.1.3 実験方法

1. 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 ± 0.10 , ± 0.20 , $\pm 0.30\text{m}$, また, ロボットの進行方向に対しては 0.5m 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 ± 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.

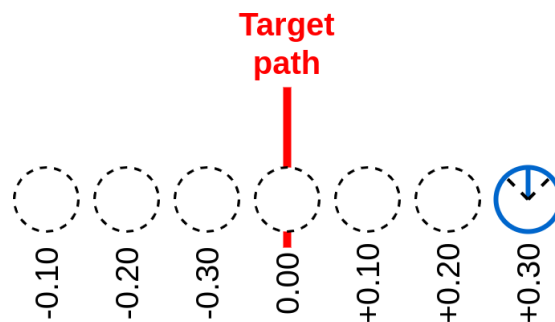


Fig. 5.4: Method of collecting data around the target route

2. 訓練フェーズ

データ収集フェーズで収集したデータ 2748 個を用いて, バッチ学習で 4000step, 8000step, 10000step 学習した. なお, 4000step は従来手法において, シミュレータの実験に用いられてきたステップ数であり, 10000step は従来手法において, 実ロボットの試験に用いられていたステップ数である.

3. テストフェーズ

Fig. 5.2 に示すコースで 10 個の学習済みモデルを使用して走行させる. ロボットの並進速度 0.2m/s とし, 経路を 3 周できた場合を成功, 壁に激突したり, 経路から 10m 離れたりした場合を失敗とした.

5.1.4 実験結果

実験結果を表 5.1 に示す。また, 失敗箇所は Fig. 5.5, 失敗箇所ごとの失敗回数は表 5.2 のようになった。

Fig. 5.5 の×の箇所で曲がり切ることができずにコースアウトしてしまった。訓練時の loss を Fig. 5.6, Fig. 5.7, Fig. 5.8 に示す。図では, 学習が収束している様子が確認できる。ここで, 角を曲がりきれなかった要因の一つとして, コースアウトした箇所付近の目標経路周辺のデータが足りなかったためだと考えられる。また, ステップ数を増やして成功回数が減ったのは, 直進のデータを多く学習しすぎて過学習を起こしている可能性もある。これを踏まえて, 次に目標経路と平行な方向のロボットの配置間隔を狭めて, データ数を増やすことで成功回数が増えるか検証する。

Table 5.1: Number of successes in the batch learning

Experiments	Number of successes
Exp.1(4000step)	4/10
Exp.2(8000step)	2/10
Exp.3(10000step)	2/10

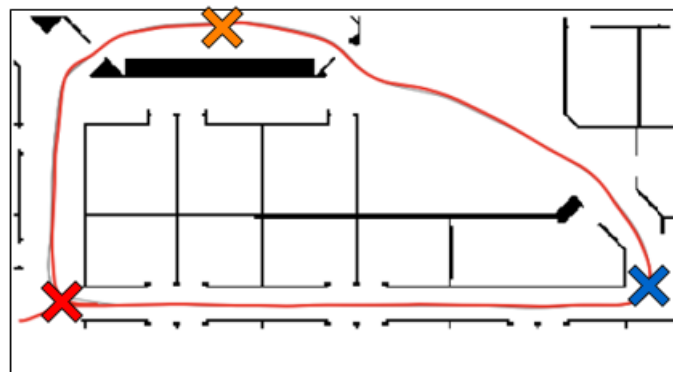


Fig. 5.5: Failure point of the experiment

Table 5.2: Number of failures in the experiment

Experiments	Failures with blue x	Failures with red x	Failures with orange x
Exp.1(4000step)	1	5	0
Exp.2(8000step)	1	7	0
Exp.3(10000step)	1	5	2

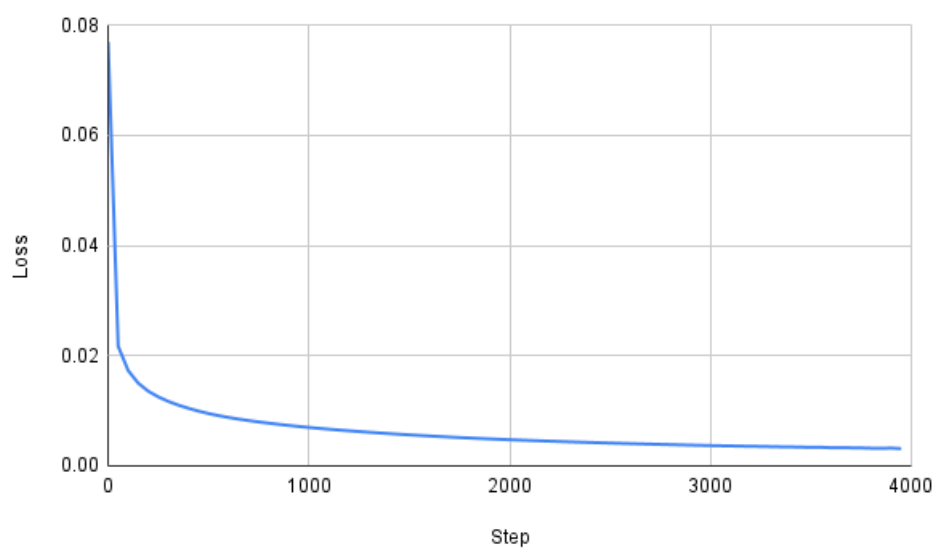


Fig. 5.6: Loss value in the experiment1

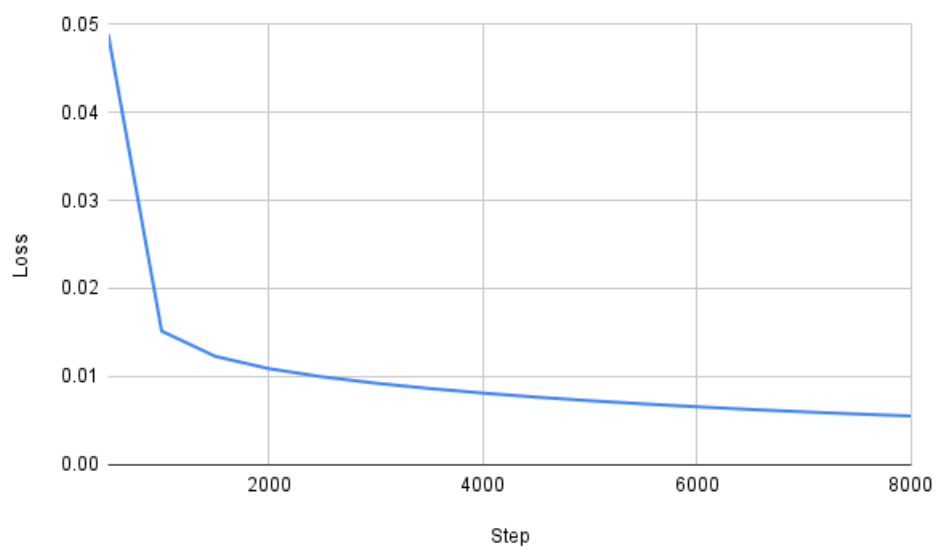


Fig. 5.7: Loss value in the experiment2

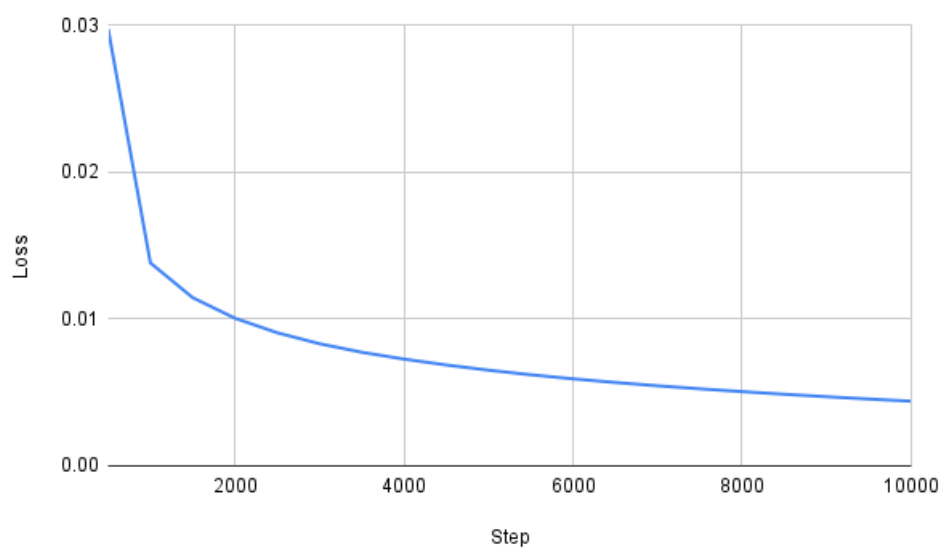


Fig. 5.8: Loss value in the experiment3

5.2 実験 2

実験目的, 実験装置, テストフェーズは実験 1 と同様である.

5.2.1 実験方法

1. データ収集フェーズ

実験 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. 5.9 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 $\pm 0.01, \pm 0.02, \pm 0.04, \pm 0.06, \pm 0.08, \pm 0.10, \pm 0.15, \pm 0.20, \pm 0.30\text{m}$ 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.

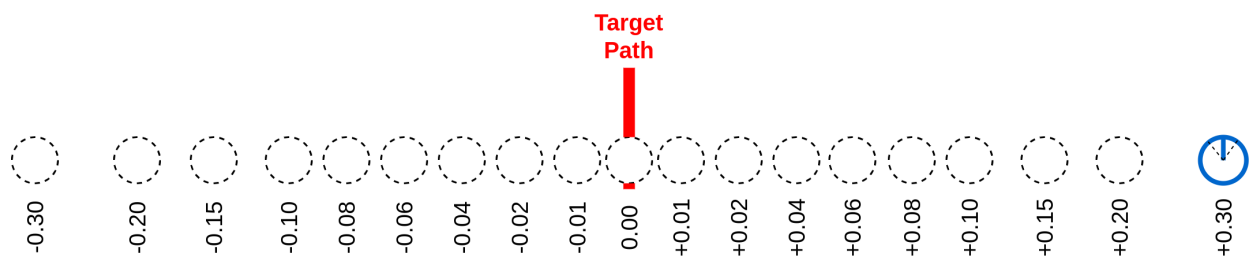


Fig. 5.9: Method of collecting data around the target route

2. 訓練フェーズ

データ収集フェーズで収集したデータ 7452 個を用いて, バッチ学習で 4000step, 8000step, 10000step 学習した.

5.2.2 実験結果

これから追記します

第 6 章

結言

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 学習で模倣する従来手法を基に、目標経路上及びその周辺でデータを収集してオフラインで訓練する手法を提案した。実験では、経路周辺のデータを多く収集し、バッチ学習を用いて訓練を行った。これにより、テストフェーズでは成功率が 100% となり、手法の有効性を示すことができた。また、従来手法では訓練時間が最低でも 40 分必要であったのに対して、提案手法を用いることで 6 分程度で訓練を終了することができた。結果として、訓練時間を 85% 削減できることを確認した。

参考文献

- [1] Mariusz Bojarski et al. ””end to end learning for self-driving cars.””. arXiv, 2016. arXiv:1604.07316.
- [2] The mnist database of handwritten digits.
<http://yann.lecun.com/exdb/mnist/>.
最終閲覧日 2025 年 1 月 16 日.
- [3] 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, 2020. pp.1147-1152.
- [4] 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, 2021. pp.1066-1070.
- [5] ros-planning, navigation.
<https://github.com/ros-planning/navigation>.
最終閲覧日 2025 年 1 月 16 日.
- [6] gazebo.
<http://gazebo-sim.org/>.
最終閲覧日 2025 年 1 月 16 日
.
- [7] 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 2025 年 1 月 16 日.

[8] Turtlebot3 robotis emanual.robotis.

<https://emanual.robotis.com/docs/>.

最終閲覧日 2025 年 1 月 16 日.

付録

動画

実験 2 で経路を周回する様子を動画に記録した. 下記に URL を掲載する.

- 実験 2 で経路を周回する様子

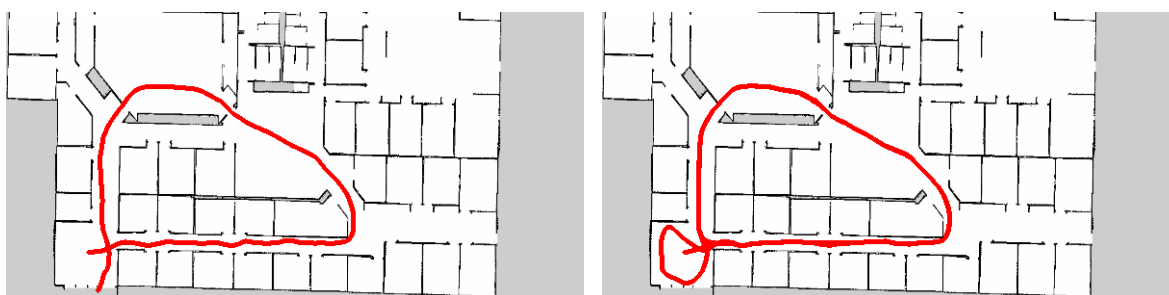
4000step

<https://youtu.be/bVp--mFVrZ0>

走行の軌跡

実験 1 と実験 3 のテストフェーズの失敗例を下記に示す.

実験 1



4000step

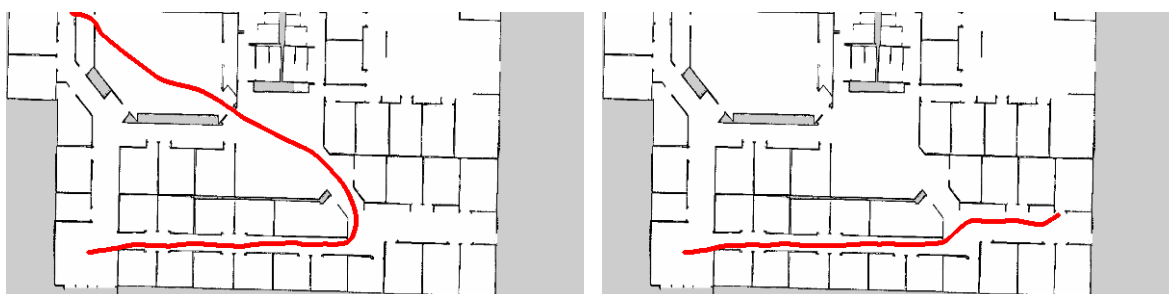


8000step



10000step

実験 3



1000step

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。

そして何より，生活面で私を支えてくださった両親に感謝します。