

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 2 月 6 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。結果として、提案手法により経路を周回できることを確認した。

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. As a result, it is confirmed that the proposed method is able to go around the path.

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	従来手法	9
3.1	従来手法の概要	9
3.2	従来手法のシステム	12
3.3	ネットワークの構造	13
第 4 章	提案手法	14
4.1	手法	14
第 5 章	実験	16
5.1	実験 1	16

目次	vi
5.1.1 実験目的	16
5.1.2 実験装置	16
5.1.3 実験方法	18
5.1.4 実験結果	19
5.2 実験 2	22
5.2.1 実験方法	22
5.2.2 実験結果	23
5.3 実験 3	26
5.3.1 実験方法	26
5.3.2 実験結果	26
第 6 章 結言	29
参考文献	30
付録	32
謝辞	35

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]	11
3.3	Systems that imitation learning for map-based navigation from [5]	12
3.4	Structure of network	13
4.1	Method of collecting data around the target route	15
5.1	Experimental environment in simulator	16
5.2	Course to collect data	17
5.3	Turtlebot3 waffle with 3 cameras	17
5.4	Method of collecting data around the target route	18

5.5	Failure point of the experiment	19
5.6	Loss value in the experiment1	20
5.7	Loss value in the experiment2	21
5.8	Loss value in the experiment3	21
5.9	Method of collecting data around the target route	22
5.10	Histogram of collected angular velocities in the experiment1 and the ex- periment2	24
5.11	Loss value in the experiment1	24
5.12	Loss value in the experiment2	25
5.13	Loss value in the experiment3	25
5.14	Failure point of the experiment	27
5.15	Loss value in the experiments	27
5.16	Loss value in the experiments	28
5.17	Loss value in the experiments	28

表目次

5.1	Number of successes in the batch learning	19
5.2	Number of failures in the experiment	20
5.3	Number of successes in the experiment	23
5.4	Number of successes in the experiment	26
5.5	Number of failures in the experiment	27

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, Bojarski 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]

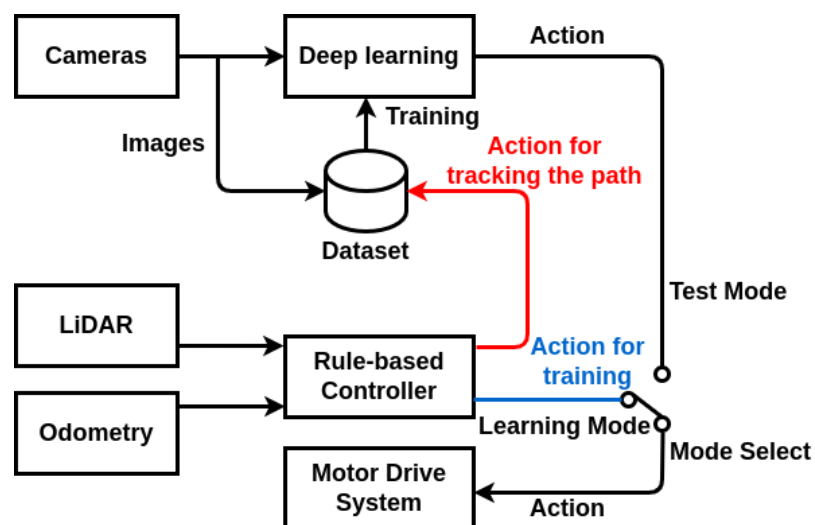


Fig. 1.3: Systems that imitation learning for map-based navigation from [2]

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.

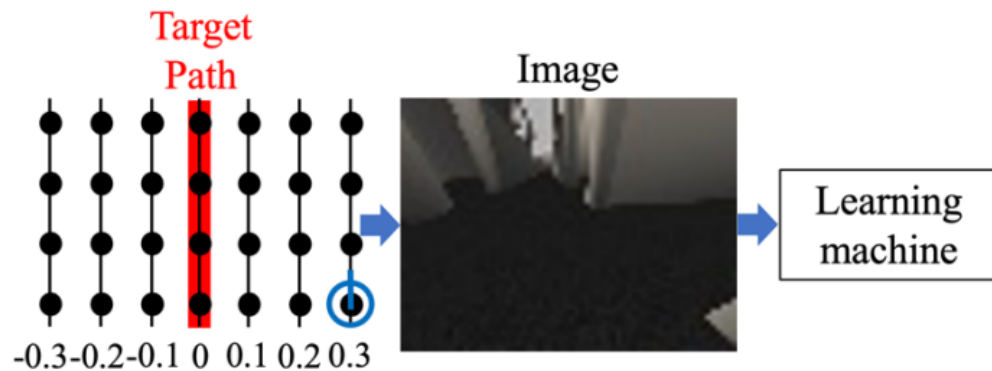


Fig. 1.4: Procedure for visualizing the output of the learning machine from [3]

これらの研究では, 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.

しかし, 岡田らと清岡ら (以下「従来手法」と称する) の提案手法は, カメラ画像を入力とした学習器の出力で自律走行をするのに, 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, 学習の時間が長くなるといった問題点がある. そこで, 本研究では, 従来手法を基に目標経路上及びその周辺のデータを一度に収集してオフラインで訓練する手法を提案する.

1.2 目的

本研究では, 従来手法を基に目標経路上及びその周辺のデータを一度に収集してオフラインで訓練する手法を提案する. さらに訓練後に, カメラ画像を入力とした学習器の出力で自律走行させることで手法の有効性を検証することを目的とする.

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. 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[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。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。

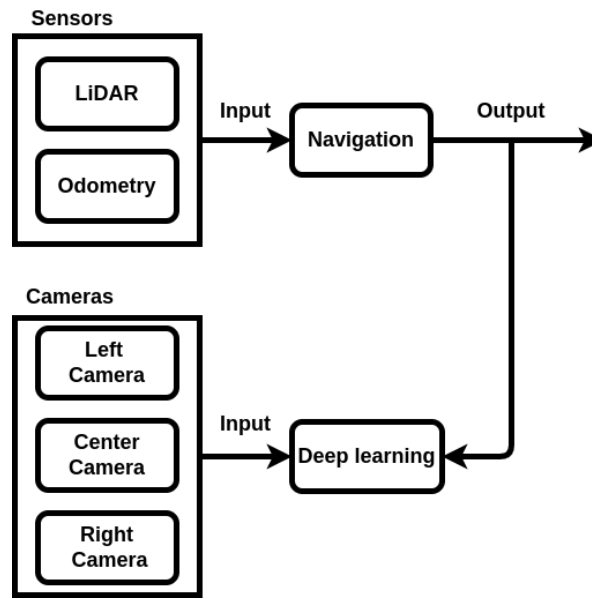
第 3 章

従来手法

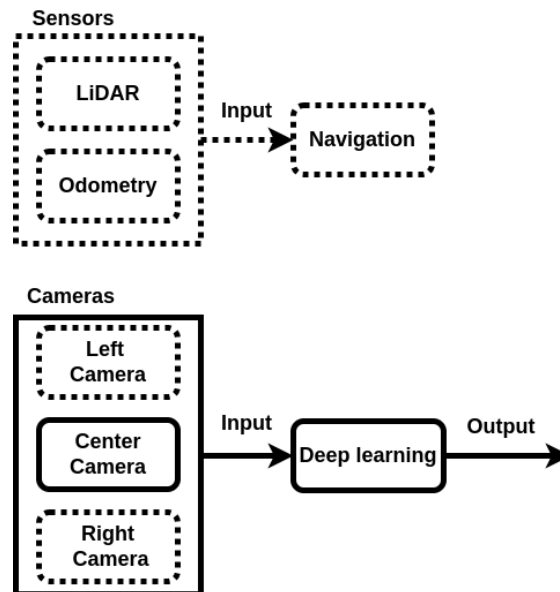
3.1 従来手法の概要

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.



(a) Learning phase



(b) Test phase

Fig. 3.1: Conventional method system

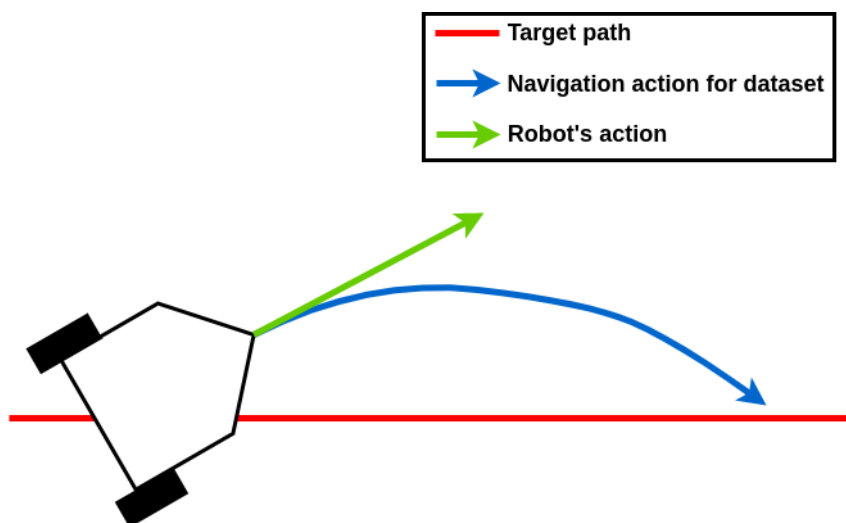


Fig. 3.2: The conventional method collects the navigation actions apart from the robot's actions from [2]

3.2 従来手法のシステム

次に、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 64×48 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。

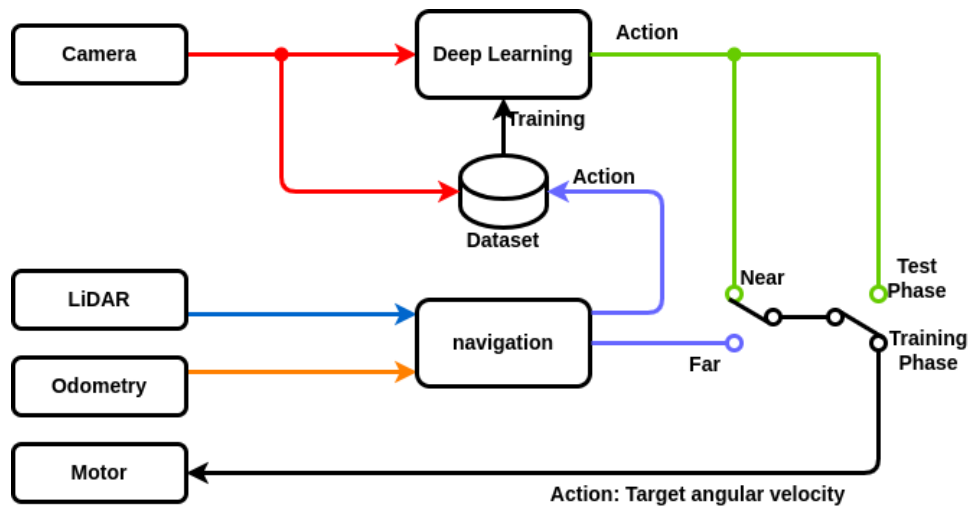


Fig. 3.3: Systems that imitation learning for map-based navigation from [5]

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。

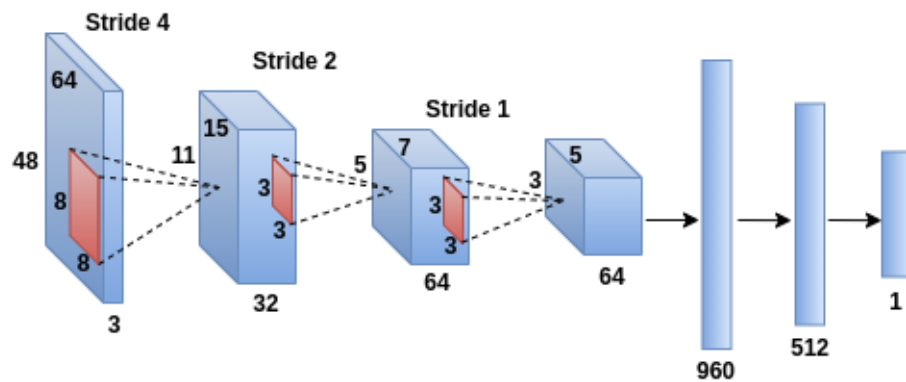


Fig. 3.4: Structure of network

第 4 章

提案手法

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

本研究で提案する手法を述べる。従来手法に対して、提案手法は一度にデータを収集して、オフラインで訓練することが異なる。

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 64×48 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 0m/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、収集したデータを用いてオフラインで学習を行う。また、従来研究ではオンラインで学習を行うため、計算のリソースなどの観点からバッチサイズを 8 にして、全てのデータを利用してなかった。しかし、提案手法ではオフラインで学習を行うため、バッチ学習を用いた訓練を試みる。

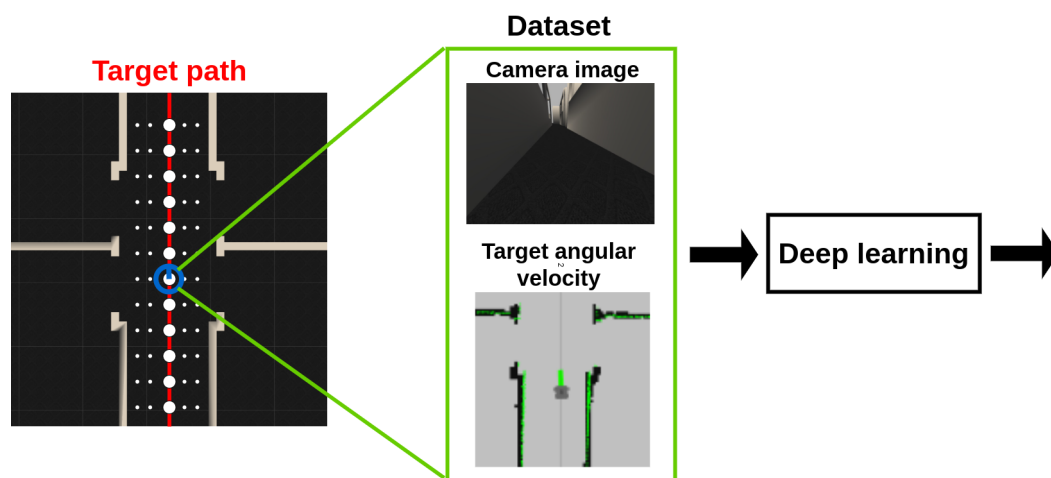


Fig. 4.1: Method of collecting data around the target route

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

aaa 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.

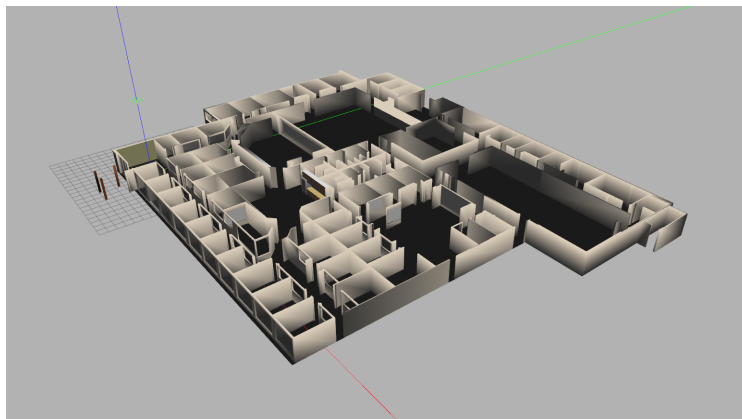


Fig. 5.1: Experimental environment in simulator

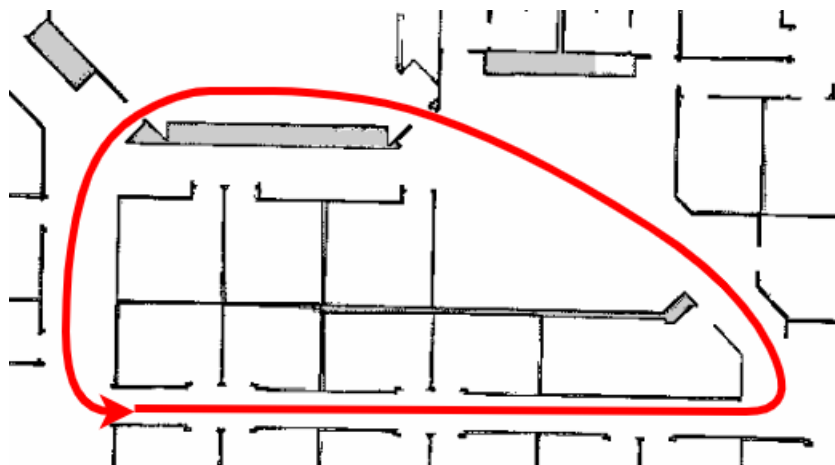


Fig. 5.2: Course to collect data



Fig. 5.3: Turtlebot3 waffle with 3 cameras

5.1.3 実験方法

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.

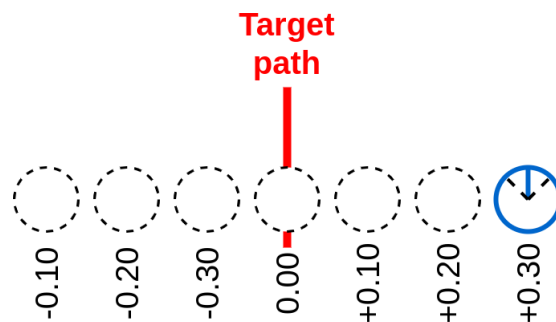


Fig. 5.4: Method of collecting data around the target route

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

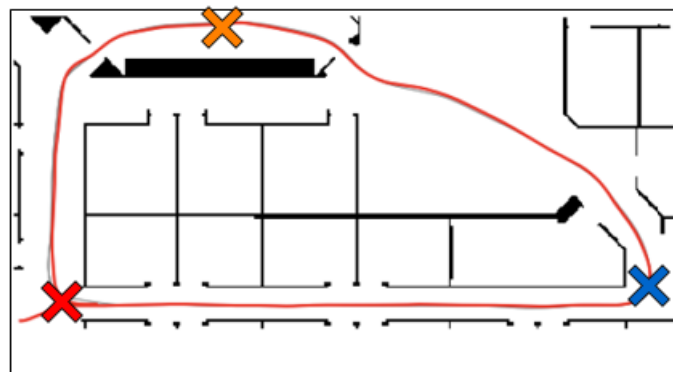


Fig. 5.5: Failure point of the experiment

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

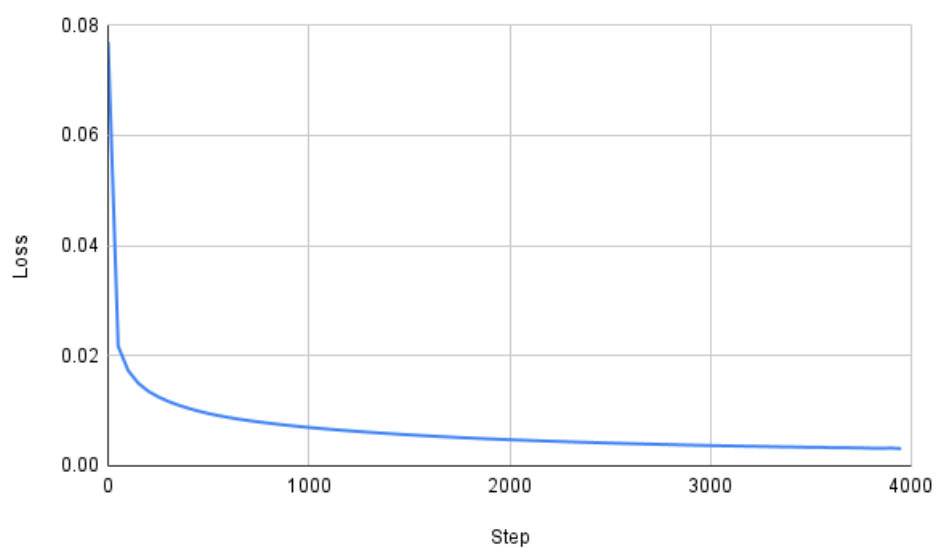


Fig. 5.6: Loss value in the experiment1

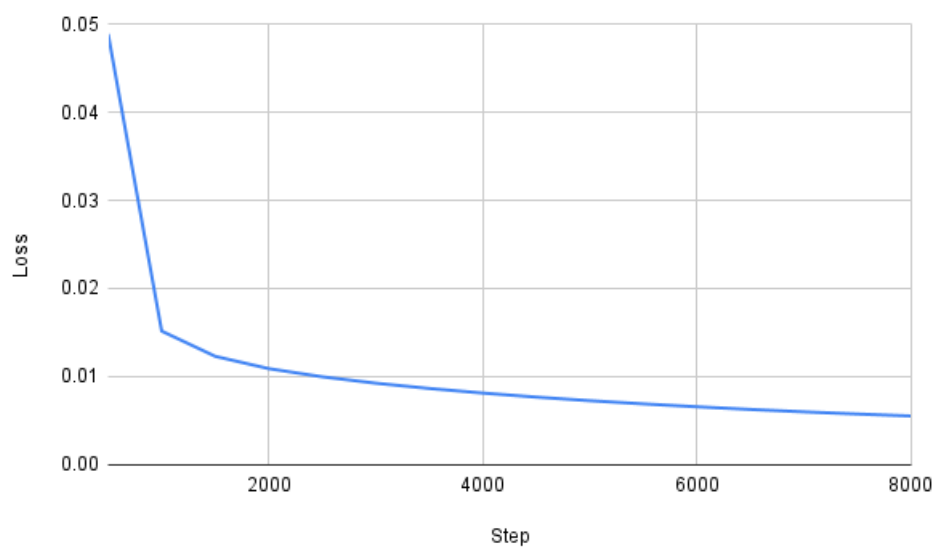


Fig. 5.7: Loss value in the experiment2

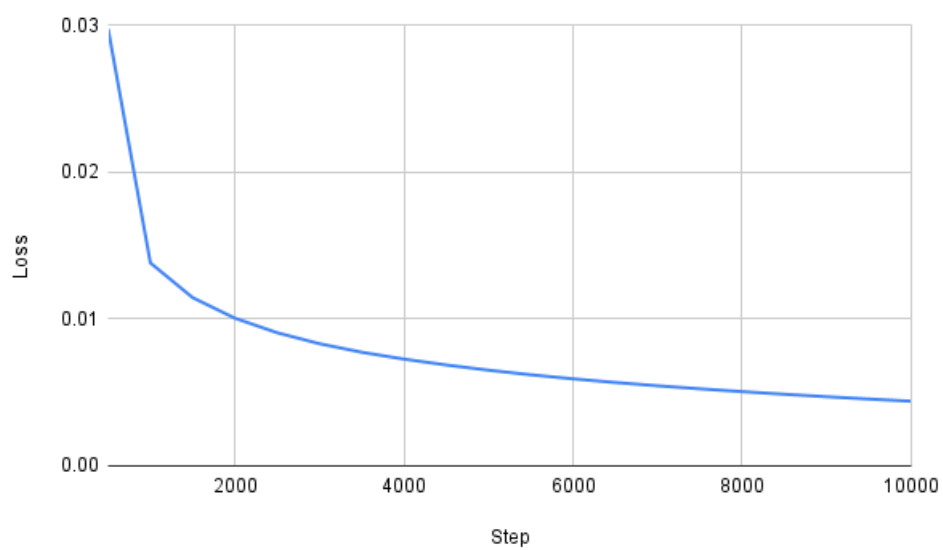


Fig. 5.8: Loss value in the experiment3

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.

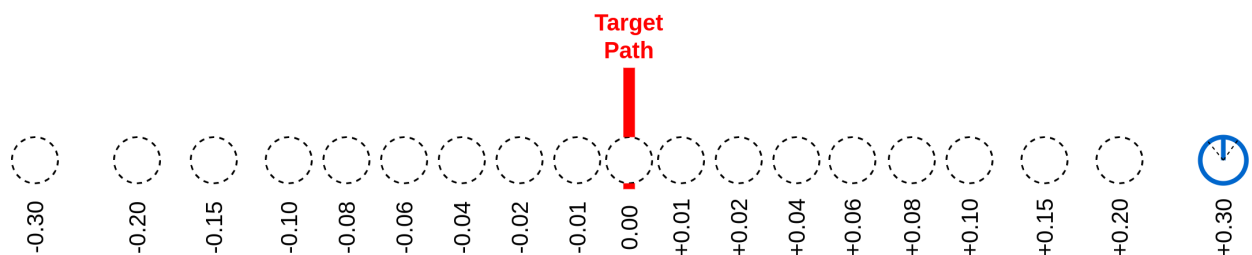


Fig. 5.9: Method of collecting data around the target route

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

実験結果を表 5.3 に示す. 実験 1 と実験 2 で収集した角速度のデータ数の比率を Fig. 5.10 に示す. 実験 2 では, 収集したデータ数全体の数も増えているが, 角を曲がる際の角速度 0.3rad/s 以上のデータも増えていることが分かる. また, 4000step, 8000step, 10000step 全てで成功回数が 10/10 となり, 経路を周回することができた. ここで, 学習の loss を Fig. 5.11, Fig. 5.12, Fig. 5.13 に示す. Fig. 5.11, Fig. 5.12, Fig. 5.13 はステップ数を増やすに連れて, 学習が収束している様子を確認できる. 従って, 目標経路周辺においてロボットの配置間隔を狭め, バッチ学習を用いて訓練することで経路追従できることを確認した.

Table 5.3: Number of successes in the experiment

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

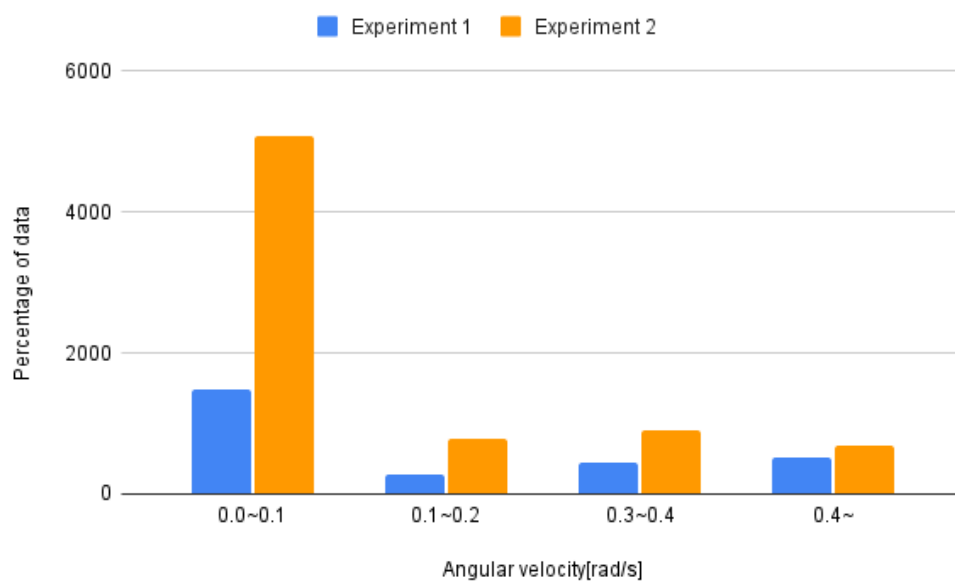


Fig. 5.10: Histogram of collected angular velocities in the experiment1 and the experiment2

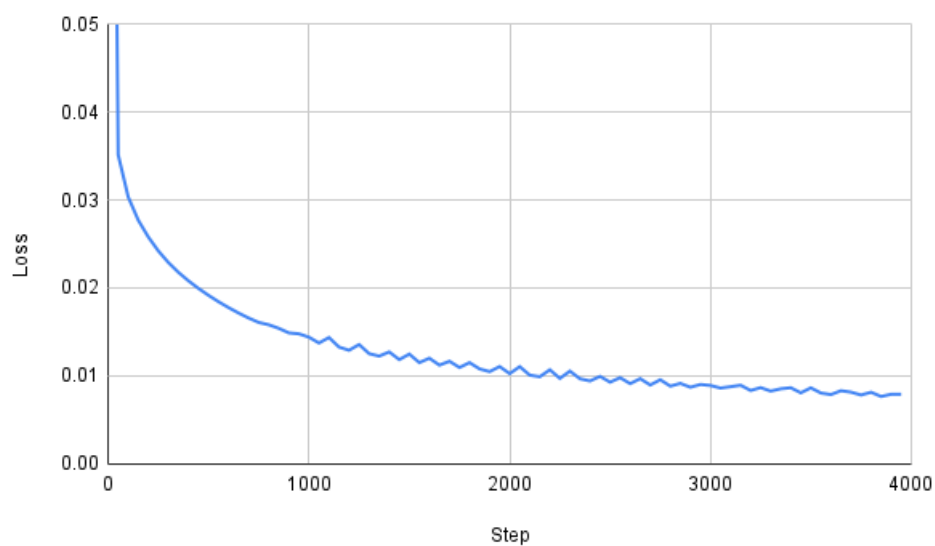


Fig. 5.11: Loss value in the experiment1

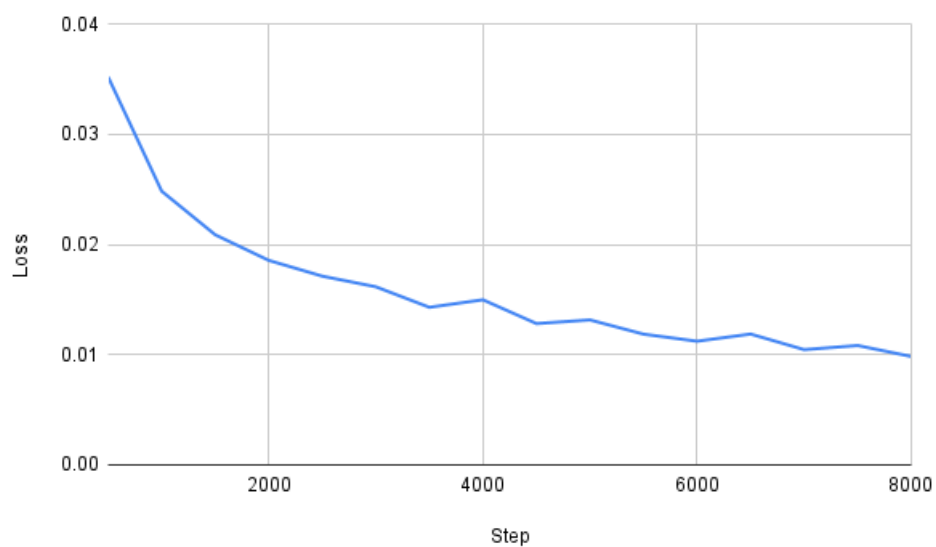


Fig. 5.12: Loss value in the experiment2

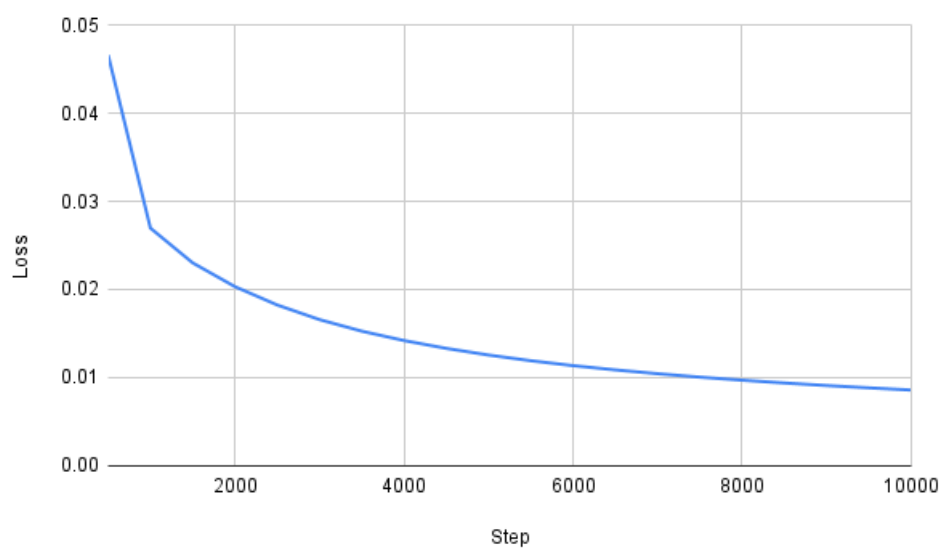


Fig. 5.13: Loss value in the experiment3

5.3 実験 3

ここでは, 実験 2 で成功率が 100% であった 4000step から 1000step ずつステップ数を減らした場合に, 成功率及び訓練時間がどのように変わるか検証する. 実験目的, 実験装置, データ収集フェーズ, テストフェーズは実験 2 と同様である.

5.3.1 実験方法

2. 訓練フェーズ

データ数 7452, バッチ学習で 4000step, 3000step, 2000step, 1000step 学習した.

5.3.2 実験結果

実験結果を表 5.4 を示す. 失敗箇所は Fig. 5.14, 失敗箇所ごとの失敗回数を表 5.5 に示す. 3000step では成功率 90%, 4 分 40 秒で訓練が終了した. 2000step では成功率 80%, 3 分 10 秒で訓練が終了し, 4000step の半分の時間で訓練を終了することができた. また, 1000step では成功率は 50% であったが, 4000step 訓練するのに必要としていた時間を約 25% で削減することができた. ここで, 各ステップ数ごとの loss を Fig. 5.15, Fig. 5.16, Fig. 5.17 に示す. 結果として, 従来手法が訓練に最低約 40 分程度必要であったのに対して, 大幅に時間を短縮できることを確認した.

Table 5.4: Number of successes in the experiment

Experiments	Number of successes	Time required for learning
Exp.1(4000step)	10/10	6min. 20sec.
Exp.2(3000step)	9/10	4min. 40sec.
Exp.3(2000step)	8/10	3min. 10sec.
Exp.4(1000step)	5/10	1min. 34sec.

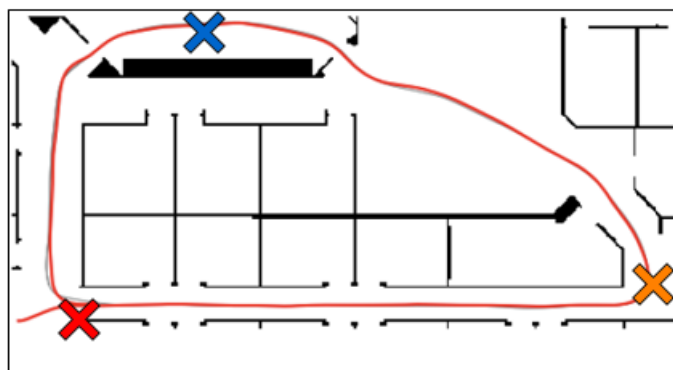


Fig. 5.14: Failure point of the experiment

Table 5.5: Number of failures in the experiment

Experiments	Failures with blue x	Failures with red x	Failures with orange x
Exp.2(3000step)	1	0	0
Exp.3(2000step)	1	1	0
Exp.4(1000step)	4	0	1

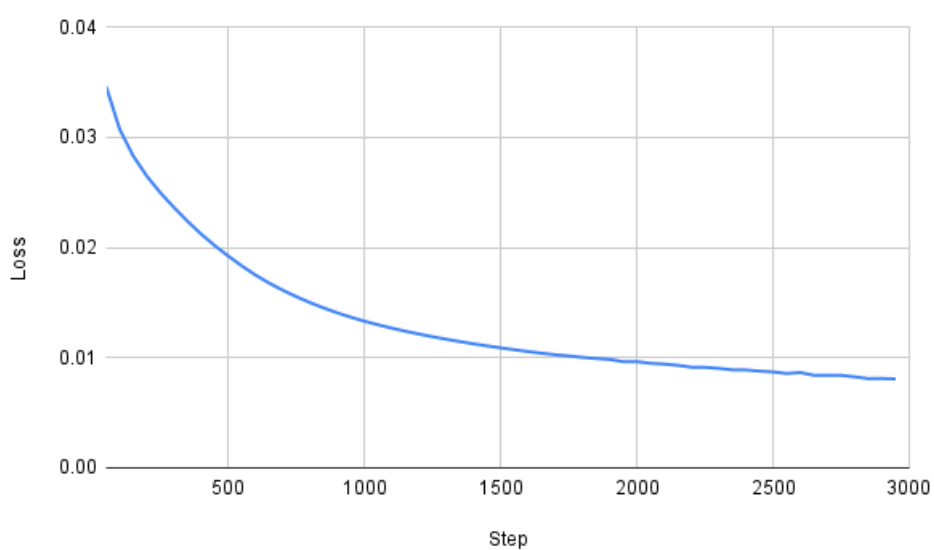


Fig. 5.15: Loss value in the experiments

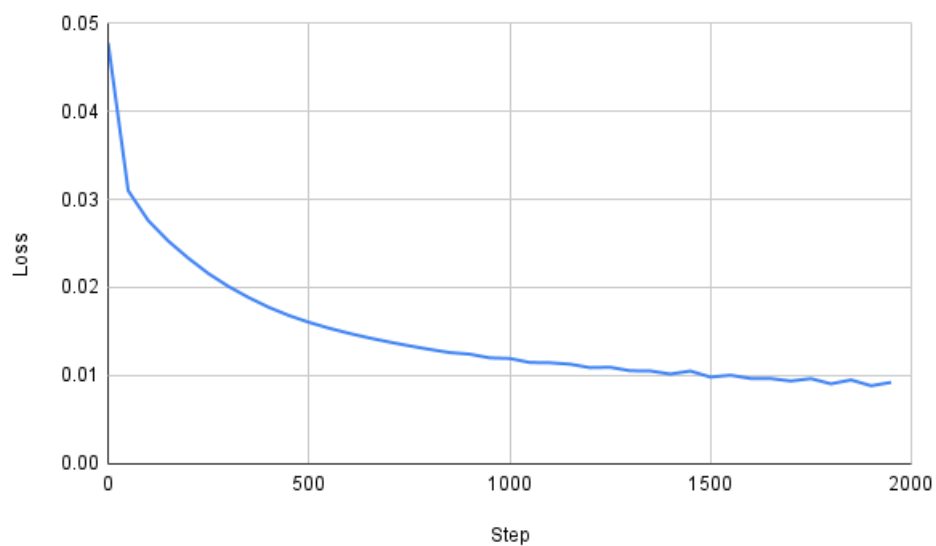


Fig. 5.16: Loss value in the experiments

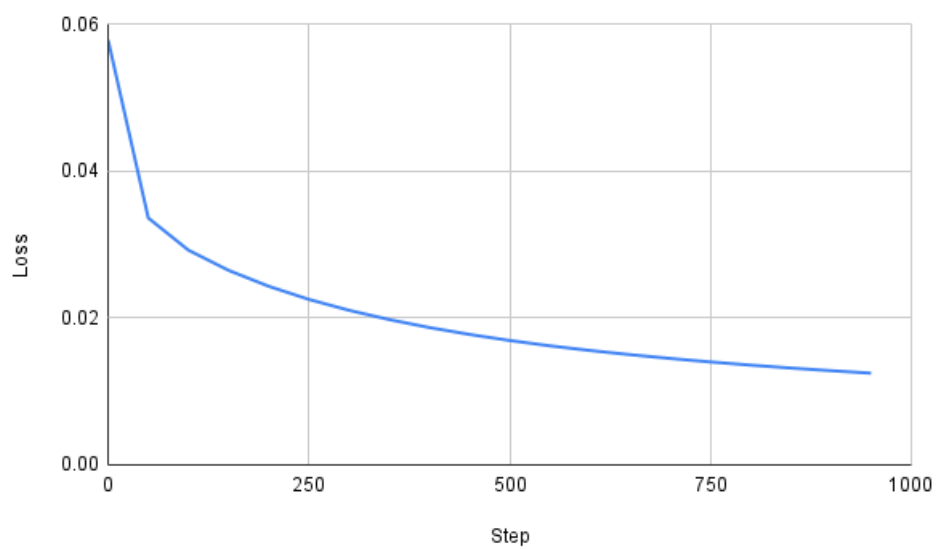


Fig. 5.17: Loss value in the experiments

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: 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 2 月 6 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 2 月 6 日.
- [7] gazebo.
<http://gazebo.org/>.
最終閲覧日 2023 年 2 月 6 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 2 月 6 日.

- [9] Turtlebot3 robotis emanual.robotis.
<https://emanual.robotis.com/docs/>.

最終閲覧日 2023 年 2 月 6 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

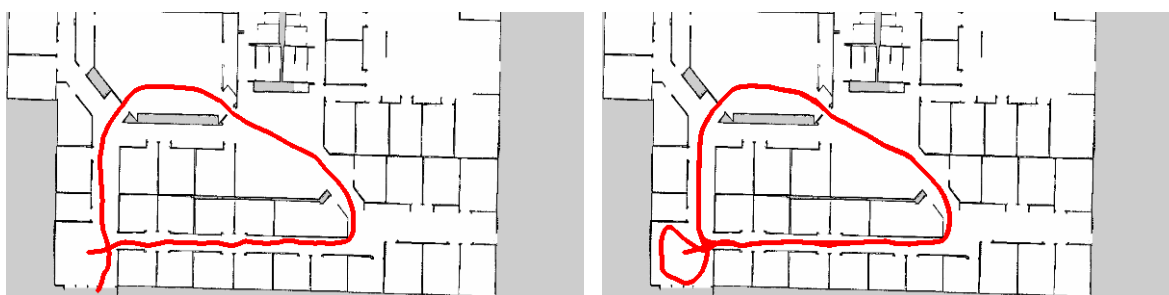
4000step

<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

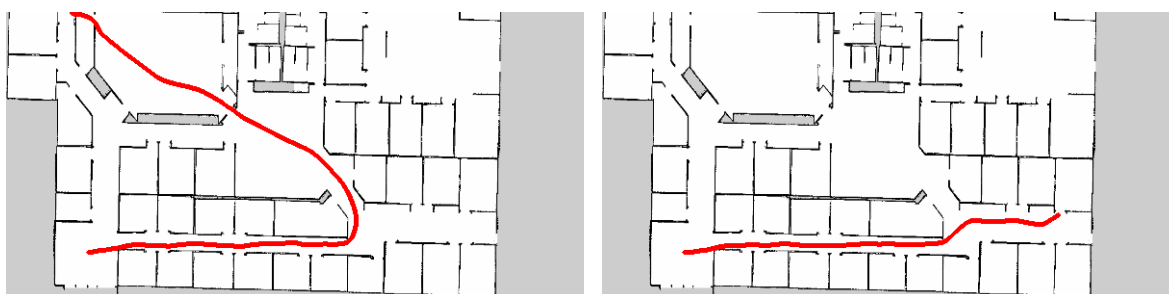


8000step



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。