

# 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 2 月 3 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 (オンラインでデータセットを収集して訓練する手法の検証)

本論文では、事前に収集した画像と行動を用いて、経路追従行動をオンラインで模倣学習する手法を提案する。また、実ロボットに提案手法を適用することを念頭においているため、実環境を想定した実験において有効であるか検証する。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。従来手法では、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った。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ることが必要で、それが問題となっていた。本論文では、事前に収集した画像と行動を用いて、経路追従行動をオンラインで学習する手法に関して検討する。これにより、従来手法で問題となっていた学習時間の短縮を目指す。シミュレーションを用いた実験により、提案手法の有効性を検証した。結果として、学習時間を大幅に短縮することができ、提案手法が有効であることを確認した。また、実環境を想定した実験では経路追従できないことを確認した。

キーワード: end-to-end 学習, ナビゲーション, オフライン

# abstract

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)

In this paper, we propose an off-line imitation learning method for path-following behavior using pre-collected images and actions. Since we intend to apply the proposed method to a real robot, we verify the effectiveness of the method in experiments under realistic conditions. We have proposed an online imitation method for vision-based path-following behavior by end-to-end learning, and have verified the effectiveness of the proposed method through experiments. Conventional methods require the robot to keep running along the path in order to collect data and perform learning. Therefore, the robot needs to run for a long time to increase the success rate of path-following, which has been a problem. In this paper, we discuss a method for learning path-following behaviors off-line using pre-collected images and behaviors. We aim to reduce the learning time, which has been a problem with conventional methods. We have verified the effectiveness of the proposed method through experiments using simulations. As a result, we confirmed that the proposed method is effective in significantly reducing the learning time. In addition, we confirmed that the proposed method is not able to follow the paths in the experiments under realistic conditions.

keywords: End-to-end learning, Navigation, Offline

# 目次

<b>第 1 章</b>	<b>序論</b>	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
<b>第 2 章</b>	<b>要素技術</b>	6
2.1	ディープラーニング . . . . .	6
2.2	end-to-end 学習 . . . . .	7
2.3	データセット . . . . .	8
2.4	オフライン学習 . . . . .	8
2.5	バッチ学習 . . . . .	8
2.6	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 . . . . .	9
<b>第 3 章</b>	<b>従来手法（オンライン手法）</b>	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
<b>第 4 章</b>	<b>提案手法（オフライン手法）</b>	13
4.1	手法 . . . . .	13
4.2	実験概要 . . . . .	14
4.3	実験装置 . . . . .	14
4.4	実験方法 . . . . .	16

4.5	実験結果と考察 . . . . .	17
4.6	まとめ . . . . .	18
<b>第 5 章</b>	<b>実環境を想定した実験</b>	<b>19</b>
5.1	実験概要 . . . . .	19
5.2	実験装置 . . . . .	19
5.3	実験方法 . . . . .	20
5.4	実験結果と考察 . . . . .	22
5.5	まとめ . . . . .	24
<b>第 6 章</b>	<b>原因の調査</b>	<b>25</b>
6.1	実験概要 . . . . .	25
6.2	実験方法 . . . . .	25
6.2.1	実験結果と考察 . . . . .	27
6.3	まとめ . . . . .	31
<b>第 7 章</b>	<b>結言</b>	<b>32</b>
<b>参考文献</b>		<b>33</b>
<b>参考文献</b>		<b>33</b>
<b>付録</b>		<b>35</b>
<b>謝辞</b>		<b>48</b>

# 図目次

1.1	Path-tracking by end-to-end imitation learning . . . . .	2
1.2	Training the neural network (source: [3]) . . . . .	2
1.3	Framework overview (source: [4]) . . . . .	3
1.4	Robot navigation in a pedestrian environment (source: [4]) . . . . .	4
2.1	Structure of deep learning . . . . .	7
2.2	Structure of general learning . . . . .	7
2.3	Structure of end-to-end learning . . . . .	8
2.4	MNIST dataset (source: [8]) . . . . .	8
2.5	Map based navigation using navigation package . . . . .	9
3.1	System configuration during network training (source: [9]) . . . . .	11
3.2	System configuration after network training (source: [9]) . . . . .	11
3.3	Structure of network . . . . .	12
4.1	Data collected by the simulator in the learning phase (source :[9]) . . .	14
4.2	Experimental environment in simulator . . . . .	15
4.3	Course to collect data . . . . .	15
4.4	Turtlebot3 waffle with a camera . . . . .	15
4.5	Location and orientation of the robot in the experiments (source :[9]) . .	16
4.6	Loss value in the experiment (source :[9]) . . . . .	18
5.1	Turtlebot3 waffle with 3 cameras . . . . .	20

5.2	Add cameras at $\pm 0.2[m]$ along the target path . . . . .	21
5.3	Crop and resize images . . . . .	21
5.4	Example of failed path-tracking . . . . .	23
5.5	Loss value in the experiment . . . . .	23
6.1	Exchange of supervisory data . . . . .	26
6.2	9cameras . . . . .	26
6.3	Example of failed path-tracking for experiment3 . . . . .	28
6.4	Example of failed path-tracking for experiment4 . . . . .	28
6.5	Example of failed path-tracking for experiment5 . . . . .	29
6.6	Example of failed path-tracking for experiment6 . . . . .	29
6.7	Loss value in the experiment3 . . . . .	29
6.8	Loss value in the experiment4 . . . . .	30
6.9	Loss value in the experiment5 . . . . .	30
6.10	Loss value in the experiment6 . . . . .	30

# 表目次

4.1	Number of successes in the batch learning . . . . .	17
5.1	Offsets at each position and orientation . . . . .	20
5.2	Number of successes in the experiments of simulator . . . . .	22
6.1	Number of successes in the experiments of simulator . . . . .	28

# 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 [1][2]. 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.

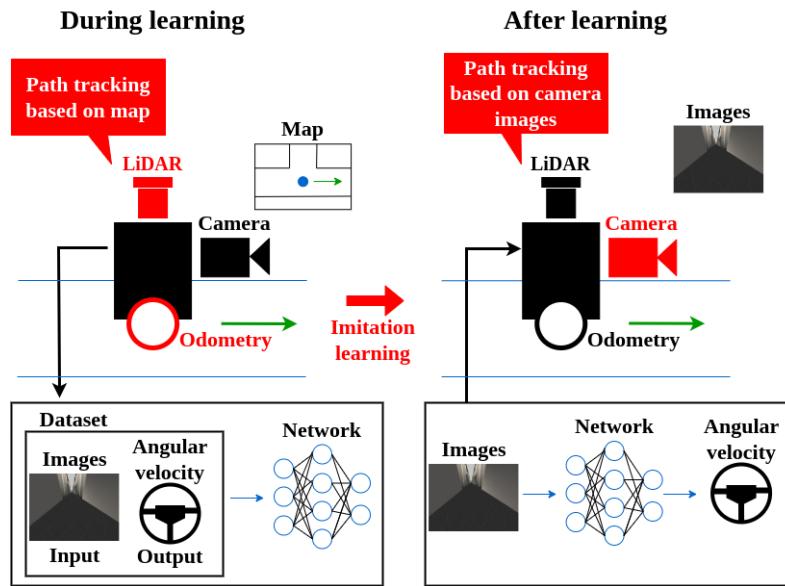


Fig. 1.1: Path-tracking by end-to-end imitation learning

## 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。例えば、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 [3]。

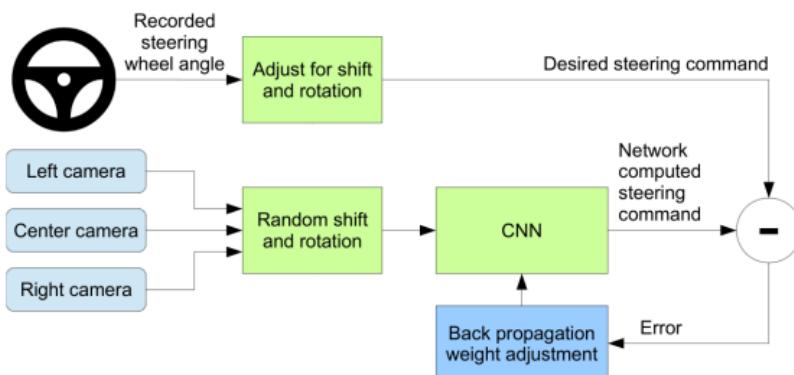


Fig. 1.2: Training the neural network (source: [3])

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, 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 [4].

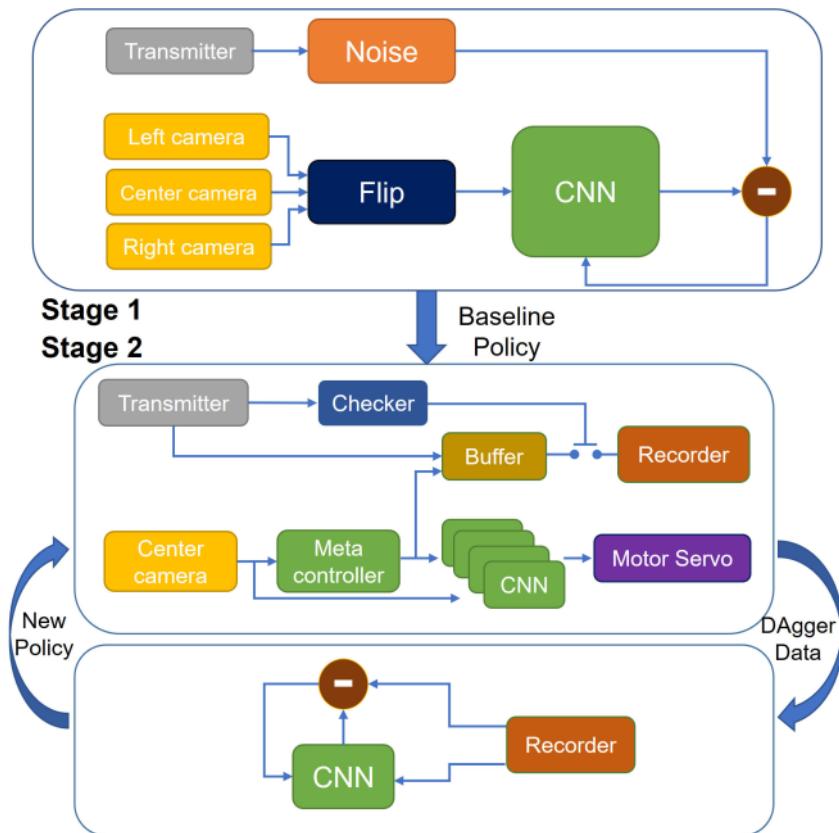


Fig. 1.3: Framework overview (source: [4])



Fig. 1.4: Robot navigation in a pedestrian environment (source: [4])

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、岡田らの手法は自動運転を模倣する点が異なる。しかし、岡田らの手法では、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、岡田らの手法で問題となっていた学習に要する時間を短縮するために、事前に収集した画像と行動を用いて、経路追従行動をオンラインで学習する手法（以後、オンライン手法と呼ぶ）を提案する。これにより、岡田らの手法で問題となっていた学習時間の短縮を目的とする。なお、オン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。また、実ロボットにオンライン手法を適用することを念頭において、経路周辺の視覚情報がどれだけ必要であるかを、シミュレータを用いた実験により明らかにすることも目的とする。

### 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章と第6章では、実験について説明する。そして、第7章では、本研究の結論を述べる。

## 第2章

# 要素技術

### 2.1 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. 2.1 に構造の一例を示す。

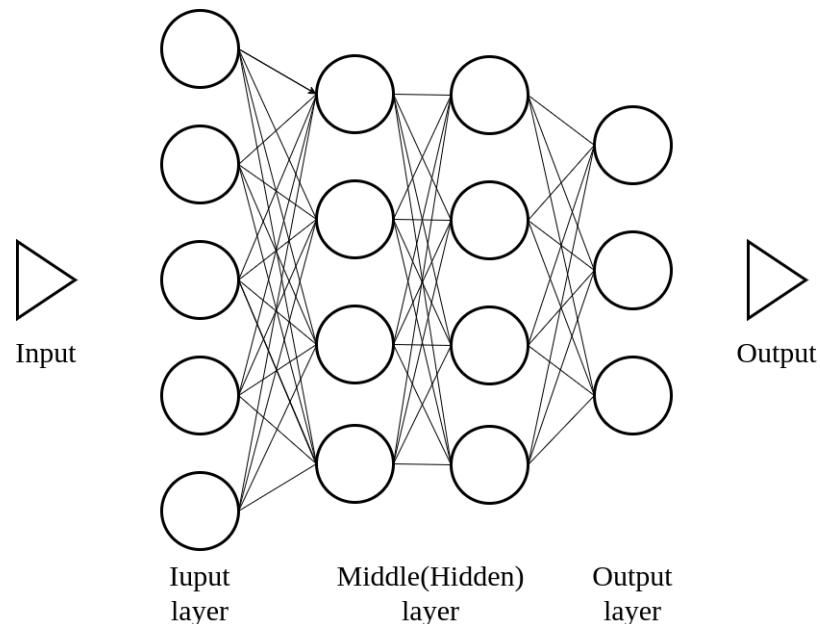


Fig. 2.1: Structure of deep learning

## 2.2 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. 2.2 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. 2.3 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。

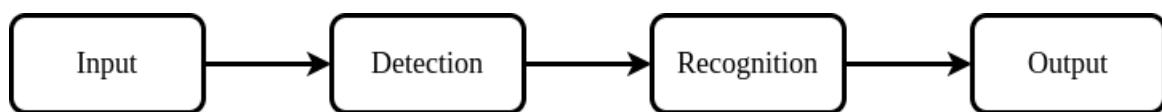


Fig. 2.2: Structure of general learning



Fig. 2.3: Structure of end-to-end learning

## 2.3 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. 2.4 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. 2.4: MNIST dataset (source: [8])

## 2.4 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、先行研究用で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。

## 2.5 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。

## 2.6 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

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、LiDAR やホイールオドメトリなどのセンサデータを使用して作成した占有格子地図を用いて目的地まで移動する。そして、移動ロボットは Fig. 2.5 のように、LiDAR やホイールオドメトリなどのセンサデータを入力として、自己位置推定と経路計画を行い、それらに基づいて自律移動をする。自己位置推定にはパーティクルフィルタを用いたモンテカルロ自己位置推定 (MCL) [6][7]、経路計画とモータ指令には `move_base[5]` を使用している。経路計画では、動的計画法によって壁や障害物を回避するように経路を生成する。モータ指令は、生成された経路に従って速度制御を行っている。



Fig. 2.5: Map based navigation using navigation package

## 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

従来から岡田らにより提案してきた手法（以後、オンライン手法と呼ぶ）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  $64 \times 48$  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  $0.2[m/s]$  で一定とする。

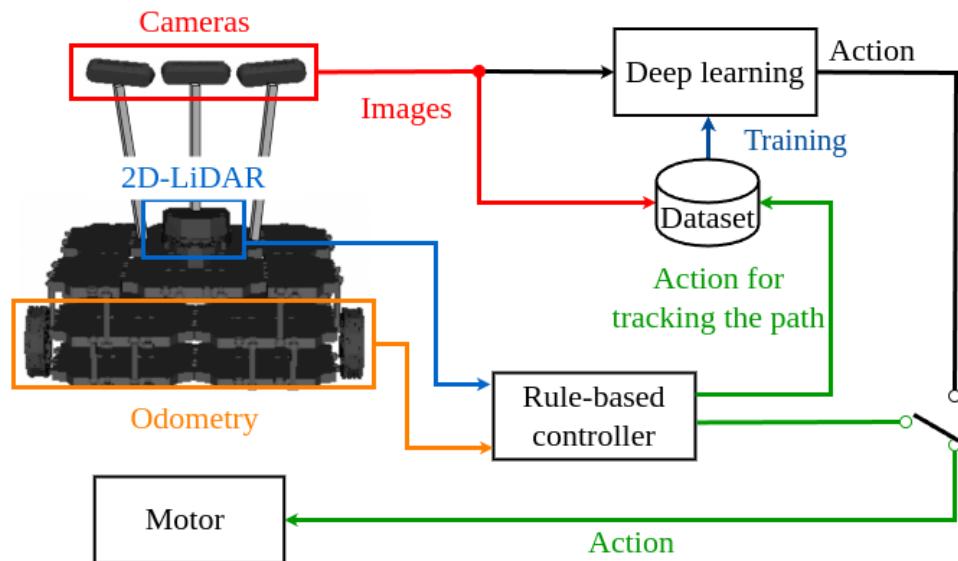


Fig. 3.1: System configuration during network training (source: [9])

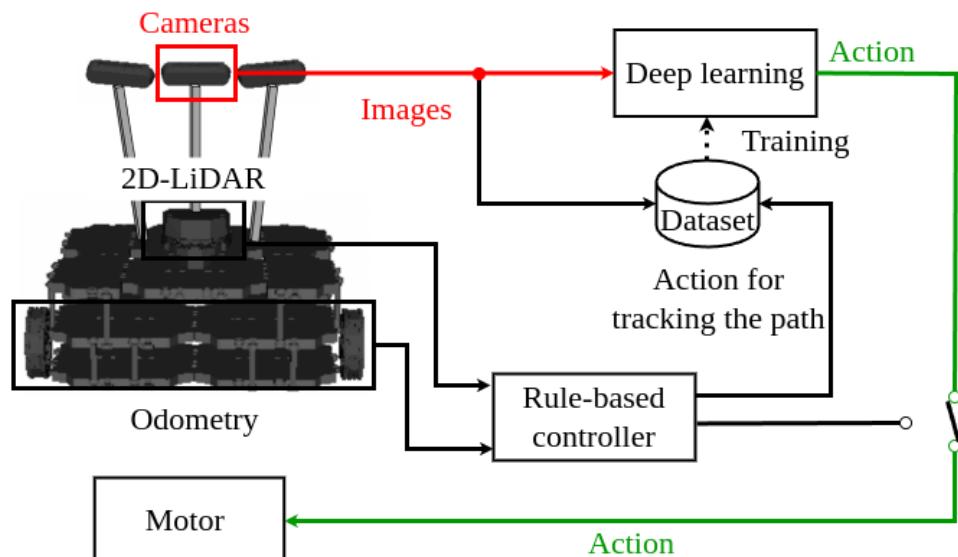


Fig. 3.2: System configuration after network training (source: [9])

## 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。

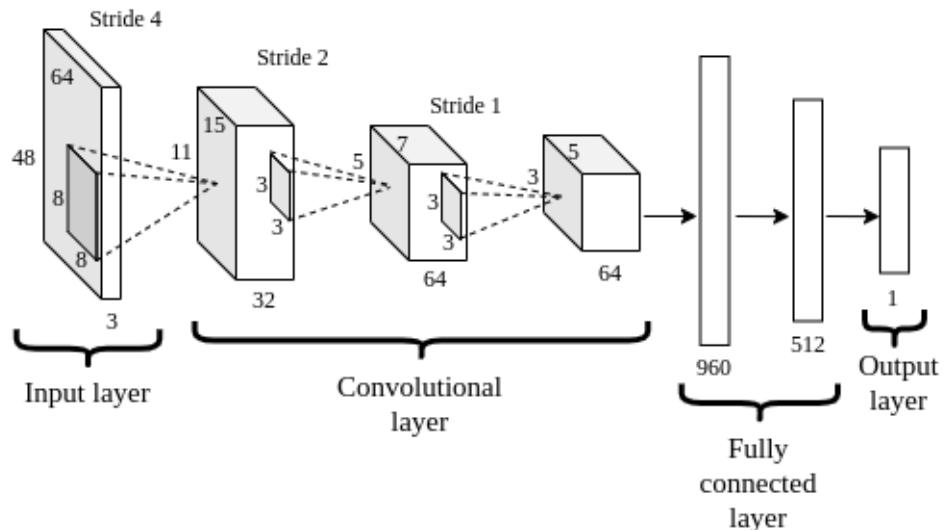


Fig. 3.3: Structure of network

## 第4章

# 提案手法（オフライン手法）

### 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、実ロボットに応用する際には、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、置き直ししなくと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。

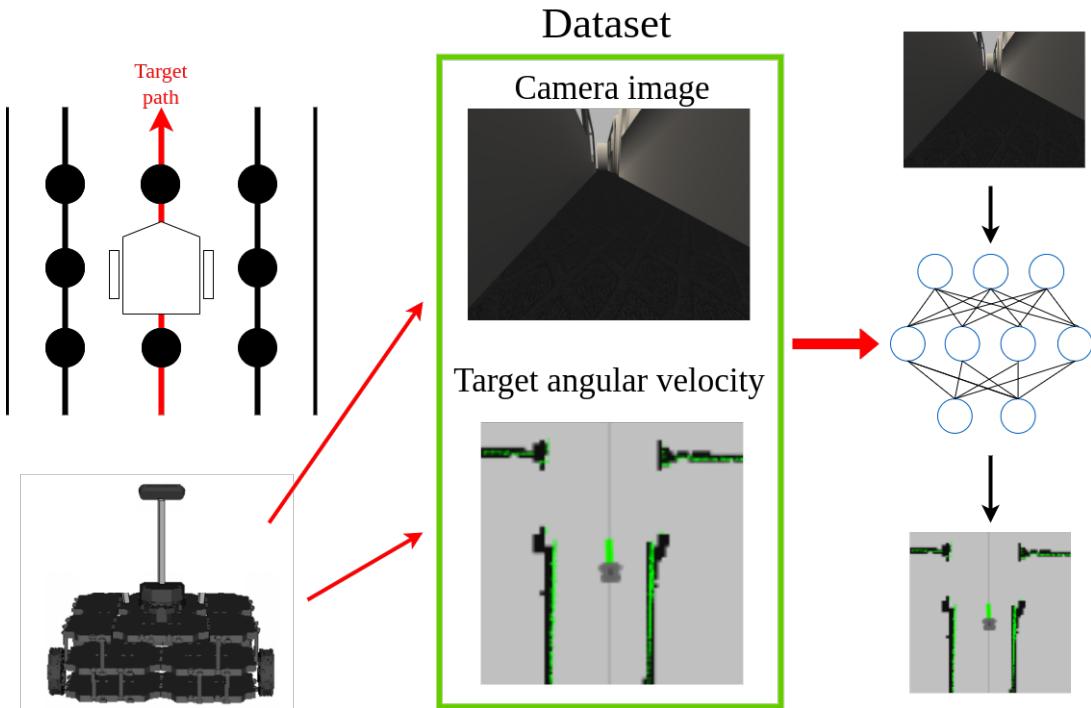


Fig. 4.1: Data collected by the simulator in the learning phase (source :[9])

## 4.2 実験概要

本章では、提案手法の有効性を検証する。そのためにシミュレータを用いた実験を行う。実験はロボットの位置と向きを組み合わせた全8パターンで検証する。それぞれの組み合わせでデータを収集して学習し、成功率を比較する。

## 4.3 実験装置

実験環境は、Fig. 4.2 に示す Gazebo[10] の Willow Garage[11] を使用して、Fig. 4.3 に示す赤線を目標経路として実験する。ロボットモデルには Fig. 4.4 に示すようなカメラを1つ搭載した Turtlebot3[12] を用いた。

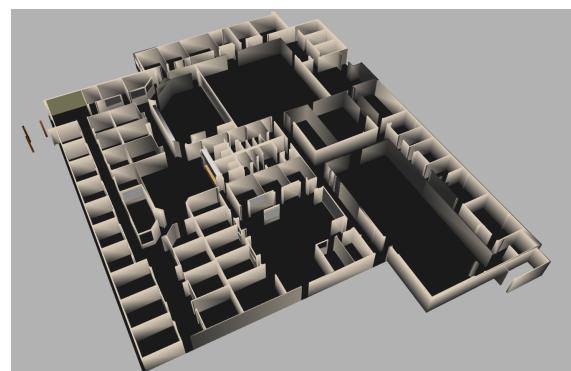


Fig. 4.2: Experimental environment in simulator

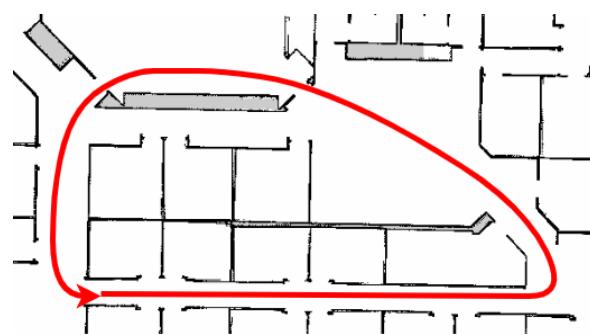


Fig. 4.3: Course to collect data



Fig. 4.4: Turtlebot3 waffle with a camera

## 4.4 実験方法

### 1) データ収集

データの収集方法について述べる。Fig. 4.5 にデータの収集方法を示す。図のようにロボットを目標経路上と目標経路から $\pm 0.1[m]$ ,  $\pm 0.2[m]$ ,  $\pm 0.3[m]$  の位置に配置する。位置に関する実験条件は、以下の4種類となる。

- (a) 目標経路上のみ
- (b) 目標経路上と目標経路から $\pm 0.1[m]$  の位置
- (c) 目標経路上と目標経路から $\pm 0.2[m]$  の位置
- (d) 目標経路上と目標経路から $\pm 0.3[m]$  の位置

また、各位置において、目標経路の向きを基準として、ロボットをヨー方向に $0[deg]$ と $\pm 5[deg]$ 回転させる。角度に関する実験条件は、以下の2種類となる。

- (e) 目標経路の向きと $\pm 5[deg]$ 回転させた向き
- (f) 目標経路の向きのみ

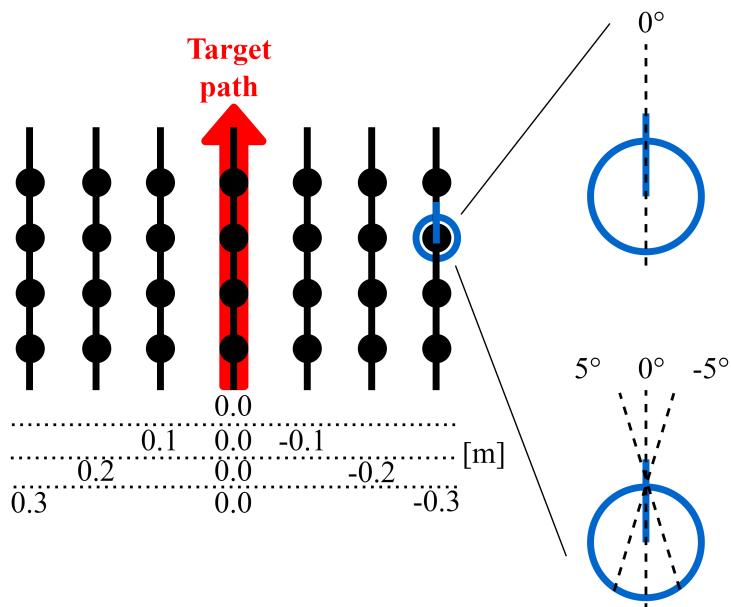


Fig. 4.5: Location and orientation of the robot in the experiments (source :[9])

## 2) 訓練

収集したデータを用いて、バッチ学習を 4000step 行う。なお、オンライン手法では同様の条件で 8000step 行っていた。オフライン手法では、後に述べるように 4000step で loss がほぼ収束するため、4000step を採用した。

## 3) テスト

学習したモデルを用いてロボットを走行させ、Fig. 4.3 に示した目標経路を追従できるかを検証する。ロボットの並進速度 0.2m/s とし、経路を 1 周できた場合を成功とし、壁に激突した場合を失敗とした。上記の 2) 学習と 3) テストを 30 回行い、経路追従の成功回数を求めた。

## 4.5 実験結果と考察

実験結果を Table 4.1 に示す。列は位置の 4 条件 (a) (d) を並べたものであり、行は方向の 2 条件 (e), (f) を並べたものである。分母の 30 は実験回数を示しており、分子の数は成功回数を示している。結果的に、目標経路上及び  $\pm 0.2[m]$  の位置、 $0[deg]$  及び  $\pm 5[deg]$  の向きに、ロボットを配置する条件で 100 % (30 回中 30 回) 成功している。これは、オンライン手法において最も高い成功率 100 % [2] と同じである。オンライン手法が 40 分程度必要なのに対して、オフライン手法での学習時間は 4 分程度であったことから、学習に要する時間を 1/10 に短縮できることを確認できた。ただし、オフライン手法での学習時間 4 分には、実験方法で述べたデータ収集は含まれていない。

Table 4.1: Number of successes in the batch learning

	0[m]	$0, \pm 0.1[m]$	$0, \pm 0.2[m]$	$0, \pm 0.3[m]$
$0, \pm 5[deg]$	1/30	28/30	<b>30/30</b>	27/30
$0[deg]$	0/30	14/30	23/30	20/30

ここで、Fig. 4.6 に、この実験条件で学習したときの loss グラフを示す。図から 4000step で loss がほぼ収束している様子が見られる。

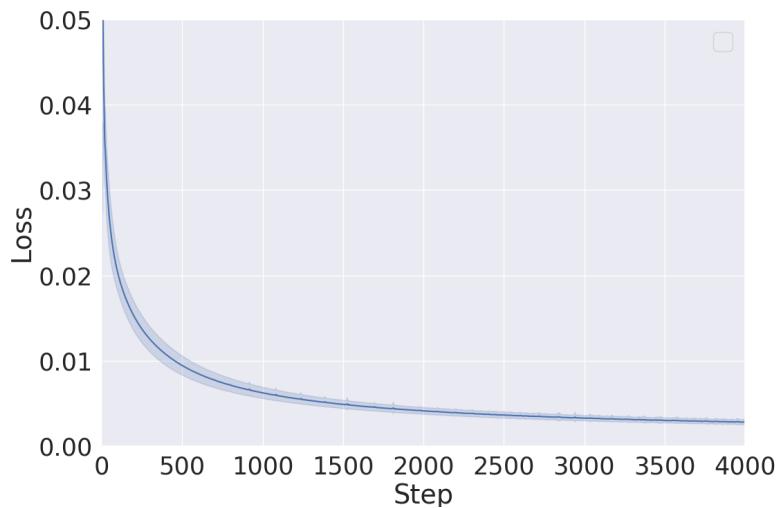


Fig. 4.6: Loss value in the experiment (source :[9])

また, Table 4.1 の他の結果を見ると, 30 回成功した実験条件 (位置 0[m],  $\pm 0.2[m]$ , 向き 0[m],  $\pm 5[\text{deg}]$ ) から離れるほど成功率が低くなっている. 特にカメラの方向が 0[deg] のみの場合は顕著に成功率が低くなっている. また, 目標経路上の画像しか使用しない条件 (位置 0[m]) では, ほとんど成功していない. それ以外の実験条件でも, カメラ間の距離により成功率が変化する様子が見られているが, この原因の究明は今後の課題としたい.

## 4.6 まとめ

本章では, 事前に収集した画像と行動を用いて, end-to-end 学習により経路追従行動をオフラインで模倣学習する手法に関して検討した. シミュレータを用いた実験により, 以下のことを確認した.

- 目標経路上及び  $\pm 0.2[m]$  の位置,  $0[\text{deg}]$  及び  $\pm 5[\text{deg}]$  の向きの視覚情報があれば経路追従できることを確認
- オンライン手法で問題となっていた学習に要する時間を  $1/10$  に短縮できることを確認

## 第 5 章

# 実環境を想定した実験

### 5.1 実験概要

前章では、オフライン手法により end-to-end 学習器を用いて、視覚による経路追従行動を模倣できることを確認した。前章の実験はシミュレータ上での検証であったが、これを実ロボットに適用する場合、まず画角の広いカメラを 3 つ用意する。次に、カメラをロボットの中央及び ± 0.2[m] に配置して、経路に沿って走行する。方向に関しては、画角の広いカメラで得られた画像から、目標経路の方向及び ± 5[deg] 回転させた時に得られる画像を切り抜いて、データセットに加える。各画像に対する目標角速度は、中央のカメラ画像とペアになる角速度にオフセットを加えて得ることができる。これにより、オンライン手法では目標経路を複数回周回しなければならなかったものが、1 周すればよいことになる。この手法が有効であるかをシミュレータ上の実験により検証することを本実験の目的とする。

### 5.2 実験装置

手法の有効性を検証するために、シミュレータを用いた実験を行う。実験環境は 5.2 章と同様とし、ロボットモデルには Fig. 5.1 に示すようなカメラを 3 つ搭載した Turtlebot3[12] を用いた。



Fig. 5.1: Turtlebot3 waffle with 3 cameras

### 5.3 実験方法

本章では、上記でも述べた条件で実験を行い手法の有効性を検証する。

#### 1) データ収集

まず、学習に使用する角速度について述べる。角速度は、収集した角速度に各位置と向きを考慮して、オフセットを加える。オフセットの値は Table 5.1 に示す。なお、この値は move\_base から出力される角速度を参考にしている。

次に、カメラ画像に関して述べる。Fig. 5.2 のように、ロボットの進行方向と垂直な方向に、中央のカメラの位置を基準として、 $\pm 0.2[m]$  の位置にカメラを増やし、それぞれの画角を 120 度から 130 度に広げる。画像サイズは  $640 \times 480$  から  $694 \times 520$  に変更する。また、Fig. 5.3 のように画像の中央及び左右にそれぞれ  $640 \times 480$  で画像を切り抜いて  $64 \times 48$  にリサイズする。

Table 5.1: Offsets at each position and orientation

	0[m]	0.2[m]	-0.2[m]
0[deg]	0	-0.2	0.2
5[deg]	-0.01	-0.25	-0.125
-5[deg]	0.01	0.125	0.25

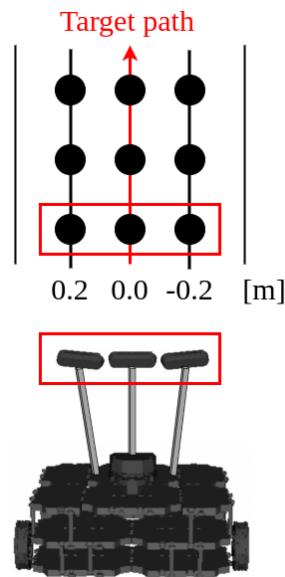


Fig. 5.2: Add cameras at  $\pm 0.2$ [m] along the target path

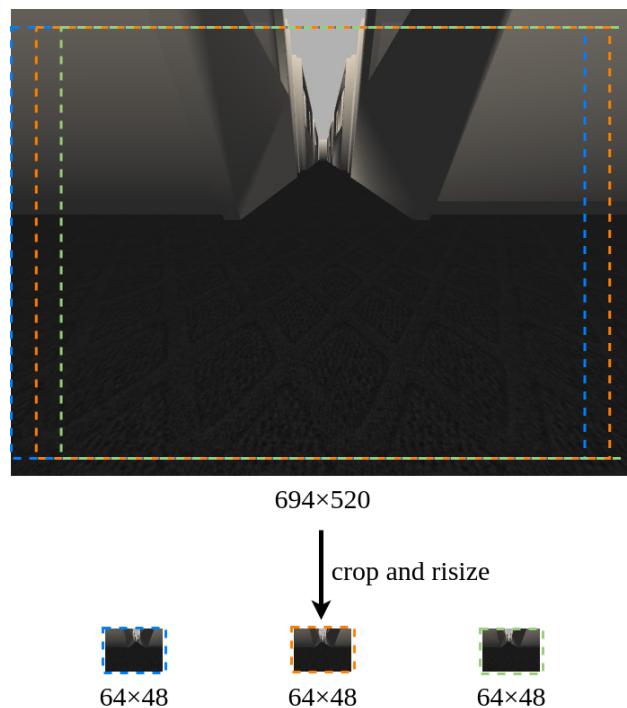


Fig. 5.3: Crop and resize images

## 2) 訓練

収集したデータを用いて、バッチ学習を 4000step 行う。

## 3) テスト

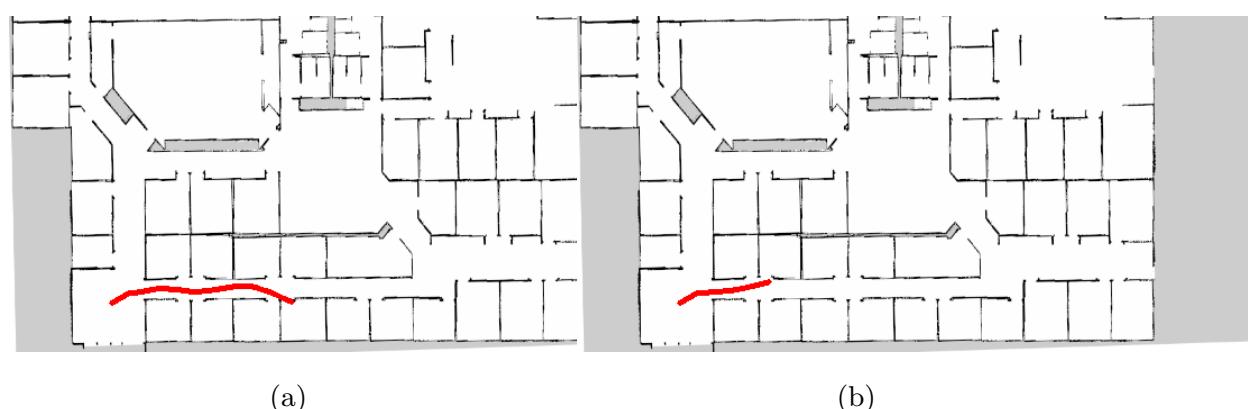
学習したモデルを用いてロボットを走行させ、Fig. 4.3 に示した目標経路を追従できるかを検証する。ロボットの並進速度 0.2m/s とし、経路を 1 周できた場合を成功とし、壁に激突した場合を失敗とした。上記の 2) 学習と 3) テストを 30 回行い、経路追従の成功回数を求めた。

## 5.4 実験結果と考察

実験結果を Table 5.2 に示す。分母の 30 は実験回数を示しており、分子の数は成功回数を示している。結果的に成功回数は 30 回中 0 回となった。テスト時の走行軌跡の一例を Fig. 5.4 に示す。図を見てわかるように、直進時に左右の壁に衝突して失敗している。また、目標経路から外れた際に、復帰するような行動も見られなかった。教師データとして使用している目標角速度とカメラ画像のどちらか、あるいは両方に原因があるため、経路追従を継続できないのではないかと考える。Fig. 5.5 に、この実験条件で学習したときの loss のグラフを示す。図から初期で loss が減少に急激して、その後は loss がほぼ変化していないことがわかる。

Table 5.2: Number of successes in the experiments of simulator

Number of successes	0/30
---------------------	------



(a)

(b)

Fig. 5.4: Example of failed path-tracking

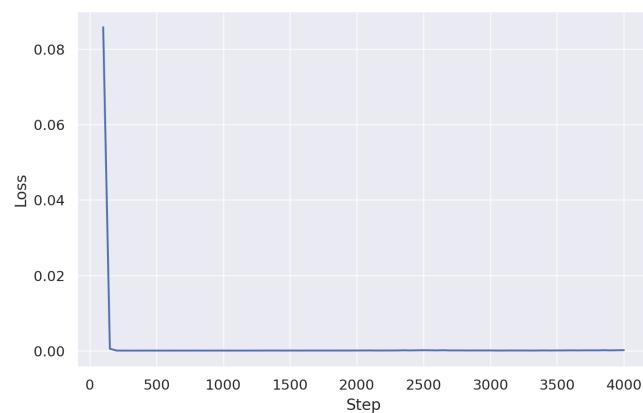


Fig. 5.5: Loss value in the experiment

## 5.5 まとめ

本章では、実環境を想定した実験により、経路追従行動をオンラインで模倣学習する手法を検討した。シミュレータを用いた実験により、以下のことを確認した。

- 本条件では、切り抜いた画像とオフセットを加えた目標角速度を用いた学習では、経路追従を継続できないことを確認
- 目標経路から外れた際に、復帰するような行動が見られないことを確認

## 第 6 章

# 原因の調査

### 6.1 実験概要

本章では、前章の実験で使用した教師データに問題があるかを調査するために、2種類の実験を行う。1つ目は教師データを入れ替えた実験、2つ目はカメラを9個にした実験である。前者の実験で、カメラ画像と目標角速度のどちらに問題があるかを明らかにする。後者の実験では、3つのカメラから得られる画像の切り抜きに問題があるかを確かめる。

### 6.2 実験方法

まず初めに、教師データを入れ替えた実験について説明する。Fig. 6.1 のように、5章で最も成功率の高かった実験（以後、実験1と呼ぶ）で使用した教師データと、前章の実験（以後、実験2と呼ぶ）で使用した教師データを入れ替えて学習を行う。

次に、カメラを9個にした実験について説明する。カメラの取り付け位置は、前章の3つのカメラそれぞれを基準として、ヨー方向に±5[deg]回転させたカメラを追加する。このロボットを使用して、走行させながらデータ収集を行う。目標角速度に関しては、前章のオフセットを用いる。

原因の調査を行うために、シミュレータを用いた実験を行う。実験環境、実験装置は前章と同様のシミュレータ環境を用いる。また、学習器の訓練条件、学習したモデルを用いたテストなどの条件は前章と同様とする。実験条件は以下の通りである。

- 実験 3)** 実験 1 の目標角速度と実験 2 のカメラ画像の組み合わせ  
**実験 4)** 実験 1 のカメラ画像と実験 2 の目標角速度の組み合わせ  
**実験 5)** 実験 1 のカメラ画像と実験 1 の目標角速度の  $0[m]$ ,  $0[deg]$  を  
 実験 2 の目標角速度と入れ替え  
**実験 6)** 9 個のカメラ画像と実験 2 のオフセットのパターン

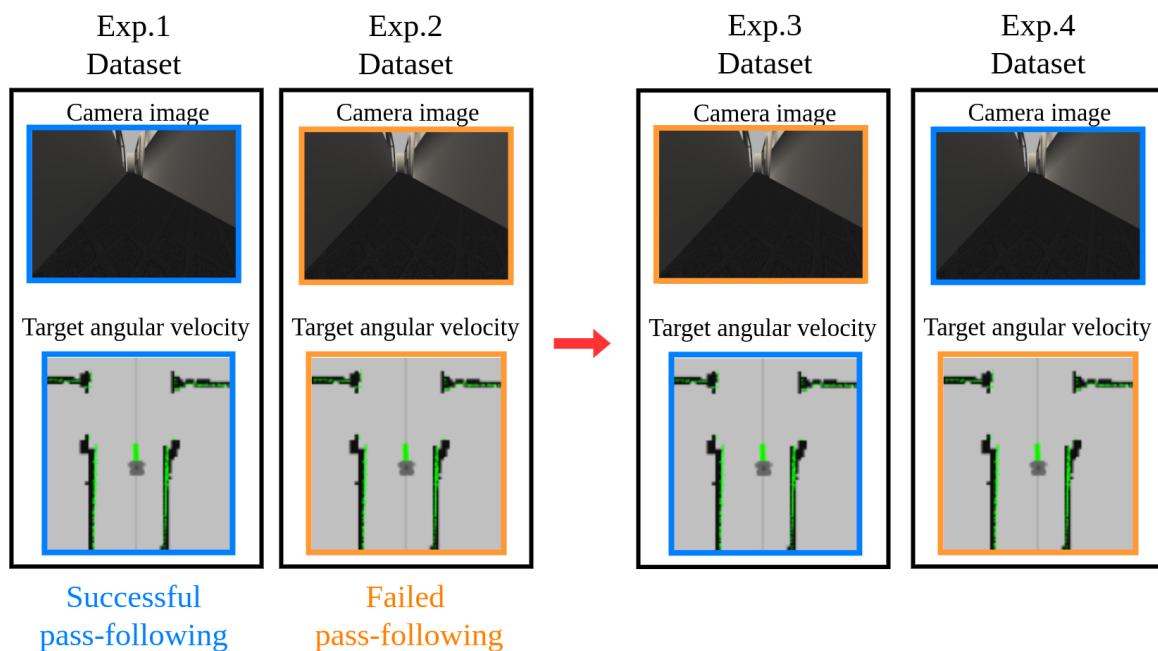


Fig. 6.1: Exchange of supervisory data

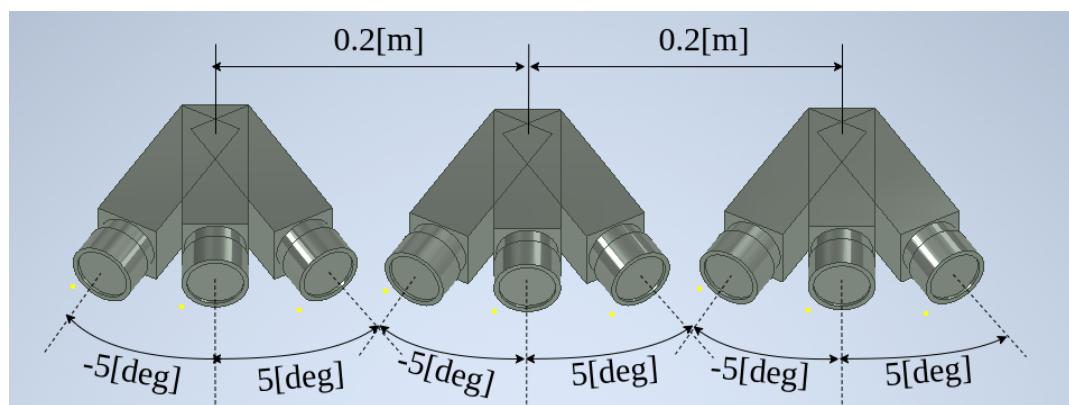


Fig. 6.2: 9cameras

### 6.2.1 実験結果と考察

実験結果を Table 6.1 に示す。分母の 30 は実験回数を示しており、分子の数は成功回数を示している。結果的に、実験 3, 実験 4, 実験 6 の成功回数はいずれも 30 回中 0 回、実験 5 の成功回数は 9/10 となった。それぞれの実験の走行軌跡を Fig. 6.3, Fig. 6.4, Fig. 6.5, Fig. 6.6 に示す。失敗例は、概ね実験 2 と同様に、直進時に目標経路から外れて、壁に衝突して失敗している。Fig. 6.7, Fig. 6.8, Fig. 6.9, Fig. 6.10 に、実験 3, 実験 4, 実験 5, 実験 6 それぞれの条件で学習した際の loss のグラフを示す。図から実験 5 のみ緩やかに loss が減少していることがわかる。それ以外は、初期で loss が減少し急激して、その後は値がほぼ変化していないことが確認できる。

実験 3 の結果から、実験 2 のカメラ画像 1 枚から 3 つの画像を切り抜く画像が、実験 1 の画像と同じであれば、実験 3 は経路追従できるはずである。同様に、実験 4 の結果から、実験 2 の move\_base の出力を参考にオフセットを加えた目標角速度が、実験 1 の目標角速度に近ければ経路追従できるはずである。実験 5 からは、実験 2 で収集した目標角速度には特に問題はなく、経路追従を継続するには、目標経路から  $\pm 0.2[m]$  の位置、ヨー方向に  $\pm 5[deg]$  の目標角速度が重要であると考えられる。

これらのことから、カメラ画像から左右に切り抜く方法では、ロボットをヨー方向に  $\pm 5[deg]$  回転させた際の画像を再現できていないと考えられる。また、目標角速度に関しては、実験 1 ではそれぞれの位置と向きでルールベース制御器の出力を取得していた。しかし、実験 2 ではロボットを走行させながら収集した角速度に、オフセットを加えることで、各位置と向きを考慮した目標角速度を生成している。この方法では、経路追従を継続するのに必要な目標角速度を再現できていないと考えられる。目標経路及び  $\pm 0.2$  の位置、 $0[deg]$  及び  $\pm 5[deg]$  の向きに、ロボットを配置した際に得られるカメラ画像と目標角速度を、どのように再現するかは今後の課題としたい。

Table 6.1: Number of successes in the experiments of simulator

Experiments	Number of successes
Exp. 3	0/30
Exp. 4	0/30
Exp. 5	9/10
Exp. 6	0/30

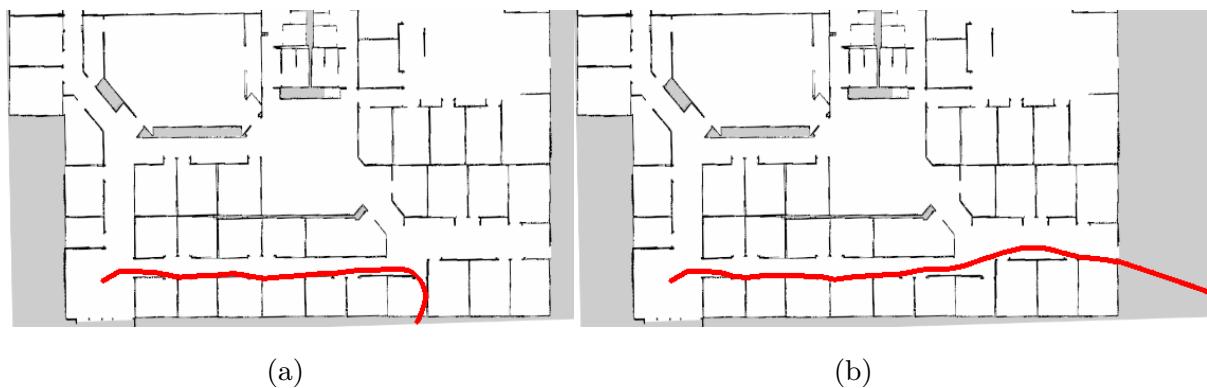


Fig. 6.3: Example of failed path-tracking for experiment3

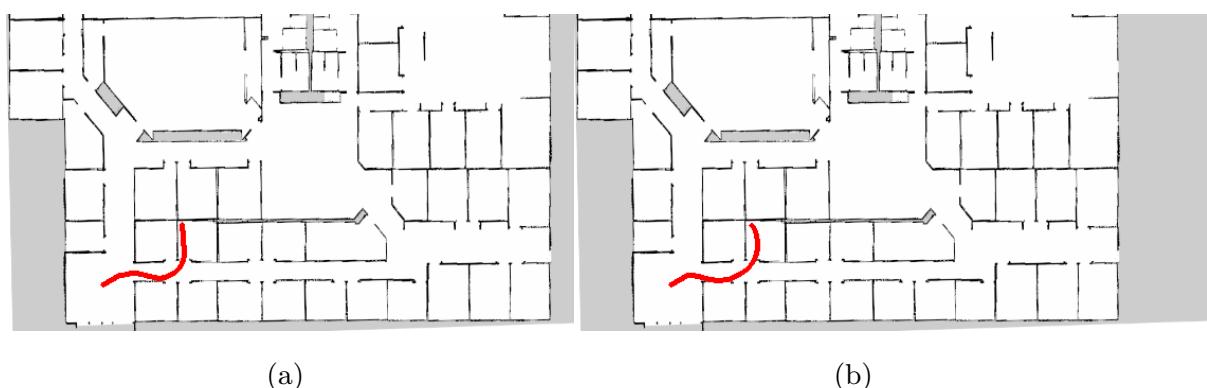


Fig. 6.4: Example of failed path-tracking for experiment4

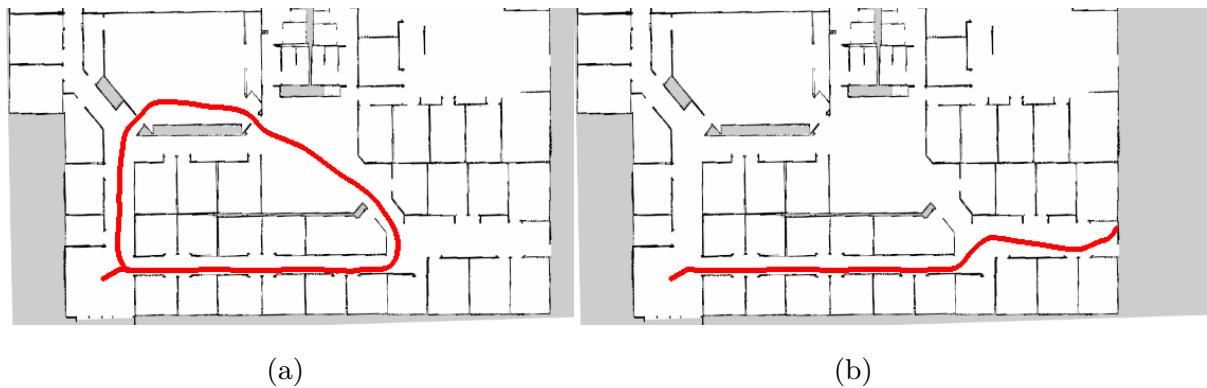


Fig. 6.5: Example of failed path-tracking for experiment5

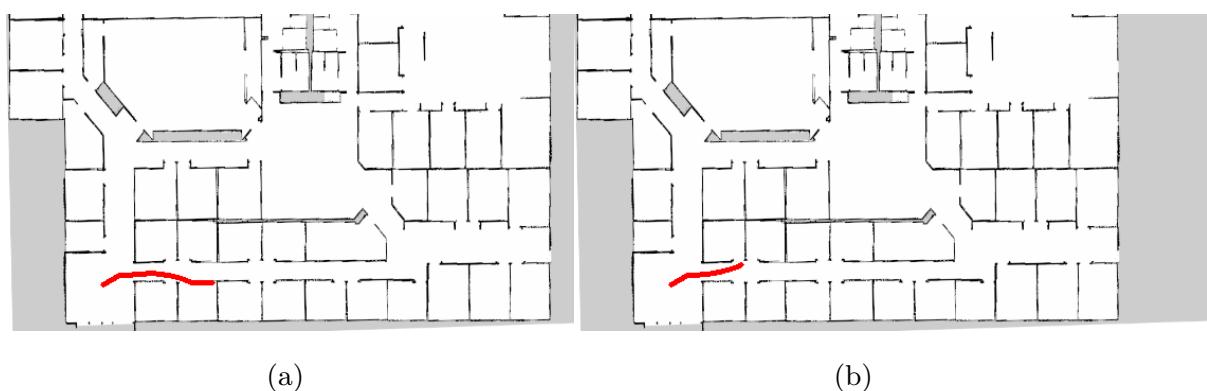


Fig. 6.6: Example of failed path-tracking for experiment6

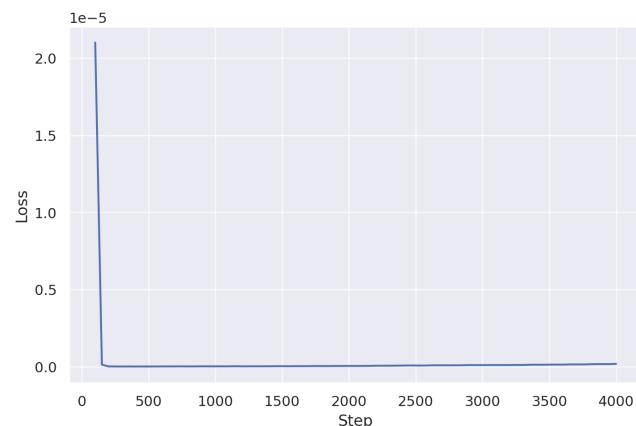


Fig. 6.7: Loss value in the experiment3

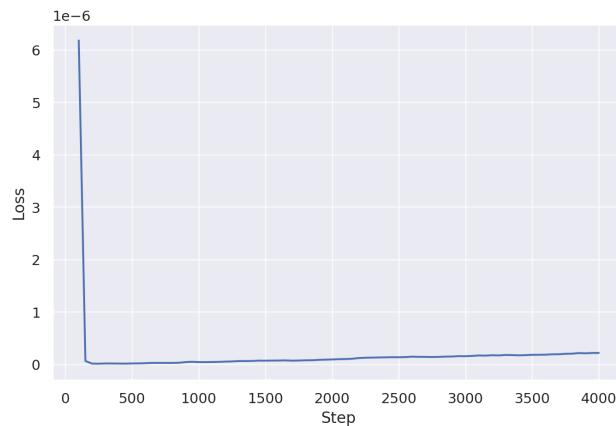


Fig. 6.8: Loss value in the experiment4

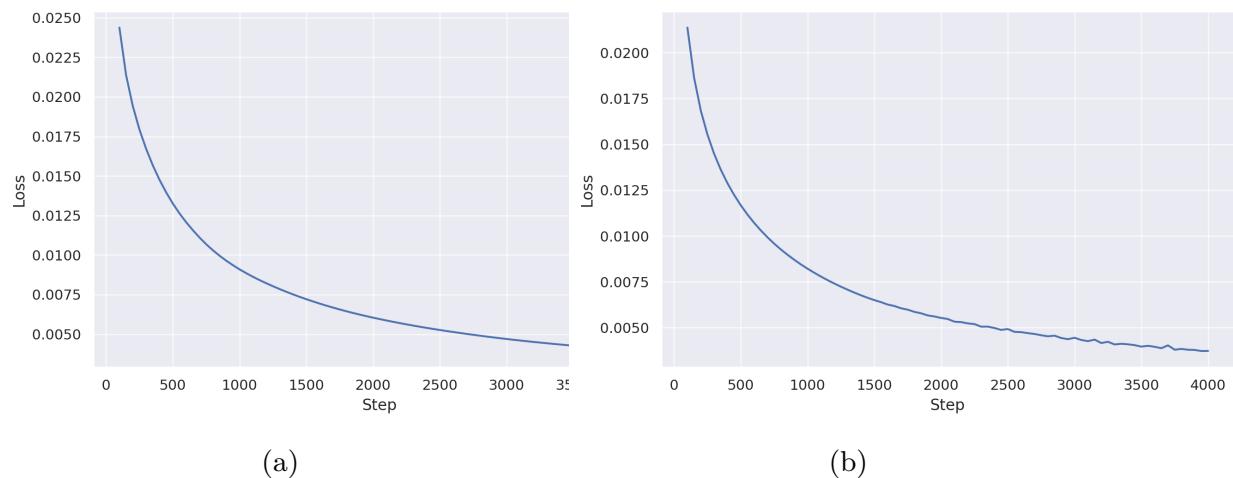


Fig. 6.9: Loss value in the experiment5

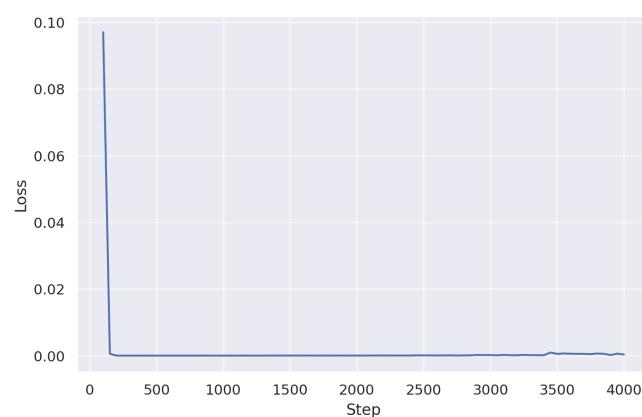


Fig. 6.10: Loss value in the experiment6

### 6.3 まとめ

本章では、前章の実験で使用した教師データに問題があるか調査を行った。シミュレータを用いた実験により、以下のことを確認した。

- 3つのカメラから得られる画像を切り抜く手法では、ロボットをヨー方向に±5[deg]回転させた際の画像を再現できていなく、画像に問題があることを確認
- ロボットを走行させた際に得られる目標角速度に、オフセットを加えた値に問題があることを確認

## 第7章

### 結言

本論文では、事前に収集した画像と行動を用いて、end-to-end 学習により経路追従行動をオフラインで模倣学習する手法に関して検討した。目標経路上及び $\pm 0.2[m]$  の位置、ロボットをヨー方向に  $0[deg]$  及び $\pm 0.5[deg]$  回転させた際の視覚情報と目標角速度があれば、経路追従を継続できることを確認した。実環境を想定した実験では、切り抜き画像とオフセットを加えた目標角速度を用いて学習する手法を検討した。結果として、経路追従を継続できないことを確認した。そこで、画像と目標に問題があるかを教師データを入れ替えた実験と、カメラを 9 個にした実験で調査した。調査の結果、画像と目標角速度の両方に問題があることを明らかにした。

## 参考文献

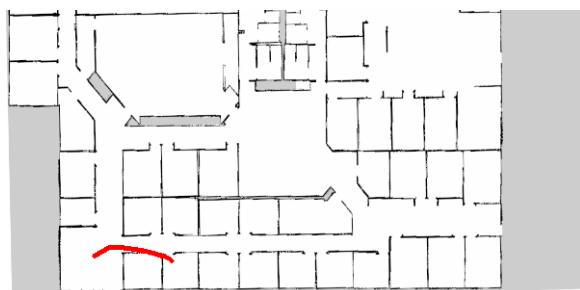
- [1] 岡田 真也, 清岡 優祐, 上田 隆一, 林原 靖男: “視覚と行動の end-to-end 学習により経路追従行動をオンラインで模倣する手法の提案”, 計測自動制御学会 SI 部門講演会 SICE-SI2020 予稿集, pp.1147-1152, 2020.
- [2] 岡田 真也, 清岡 優祐, 春山 健太, 上田 隆一, 林原 靖男: “視覚と行動の end-to-end 学習により経路追従行動をオンラインで模倣する手法の提案 -経路追従行動の修正のためにデータセットを動的に追加する手法の検討”, 計測自動制御学会 SI 部門講演会 SICE-SI2021 予稿集, pp.1066-1070, 2021.
- [3] Mariusz Bojarski et al., “End to End Learning for Self-Driving Cars.”, arXiv: 1604.07316(2016).
- [4] Jing Bi, Tianyou Xiao, Qiuyue Sun, Chenliang Xu., “Navigation by Imitation in a Pedestrian-Rich Environment,” arXiv preprint arXiv:1811.00560, 2018.
- [5] ros-planning, navigation <https://github.com/ros-planning/navigation> (最終閲覧日: 2025 年 2 月 3 日)
- [6] F. Dellaert, Fox D., Burgard W., and S Thrun. “Monte carlo localization for mobile robots “. Proceedings 1999 IEEE international conference on robotics and automation (Cat. No. 99CH36288C), Vol. 2, pp. 1322 – 1328, 1999.
- [7] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. “ Monte carlo localization: Efficient position estimation for mobile robots “. AAAI/IAAI, Vol. 2-2, pp. 343 – 349, 1999.
- [8] The MNIST database of handwritten digits <http://yann.lecun.com/exdb/mnist/> (最終閲覧日: 2025 年 2 月 3 日)
- [9] 高橋 祐樹, 白須 和暉, 藤原 庄, 上田 隆一, 林原 靖男: “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-”,

- 日本機械学会ロボティクス・メカトロニクス講演会 2023 講演論文集, 2P1-G07, 2023.
- [10] gazebo. <http://gazebosim.org/>. (最終閲覧日: 2025 年 2 月 3 日)
  - [11] Koenig and 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 2 月 3 日)
  - [12] Turtlebot3 — robotis emanual.robotis. <https://emanual.robotis.com/docs/> (最終閲覧日: 2025 年 2 月 3 日)

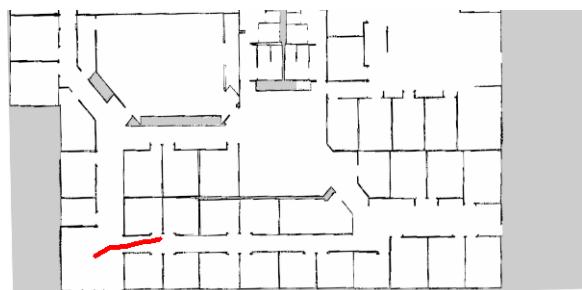
# 付録

## 走行の軌跡

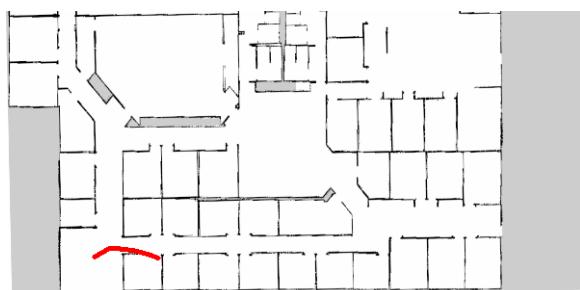
実験 2, 実験 3, 実験 4, 実験 5 の走行軌跡を下記に示す.



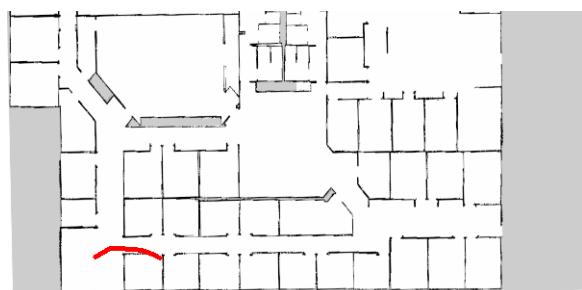
model1



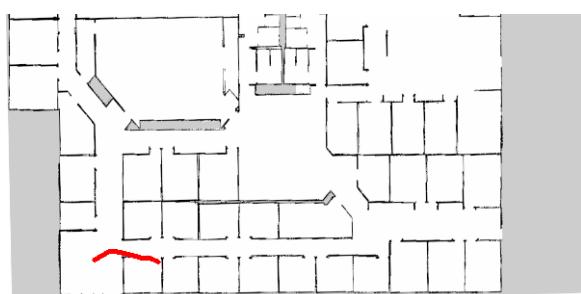
model2



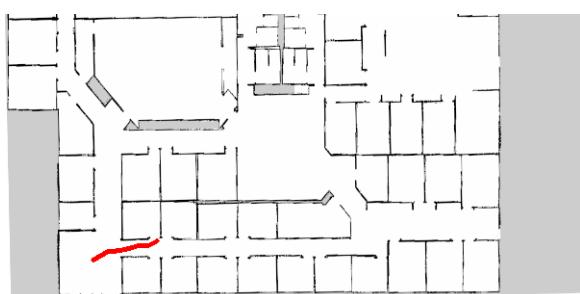
model3



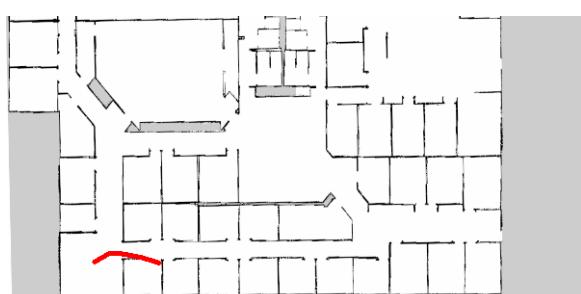
model4



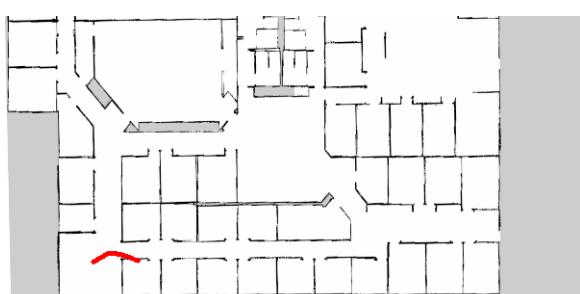
model5



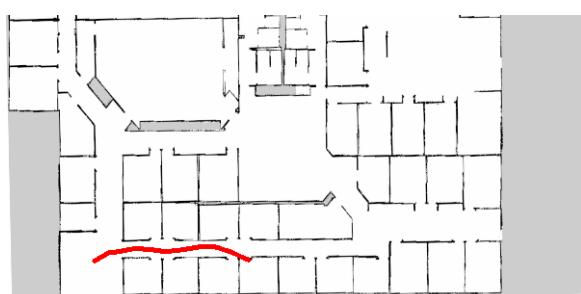
model6



model7



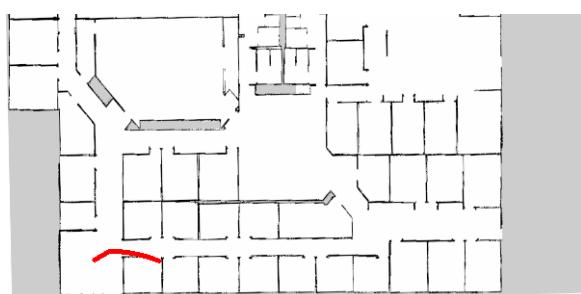
model8



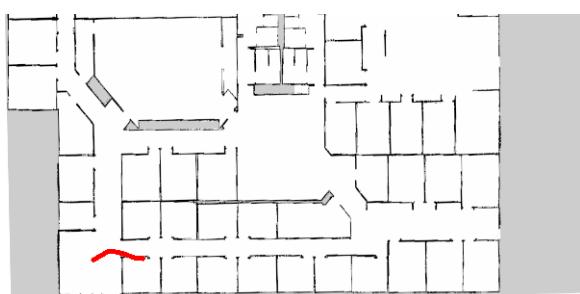
model9



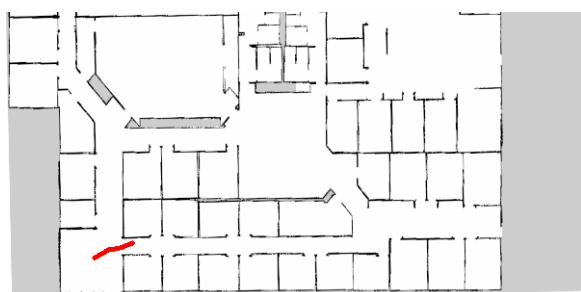
model10



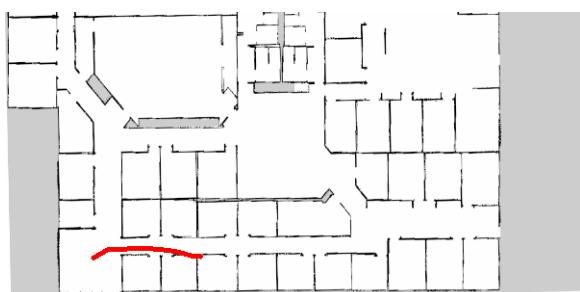
model11



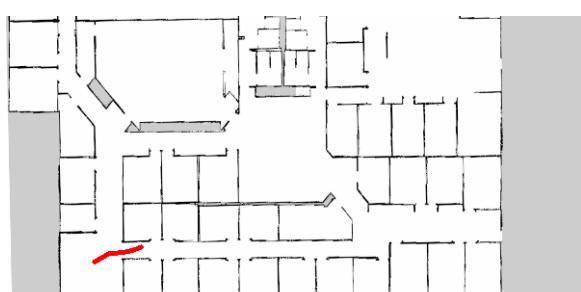
model12



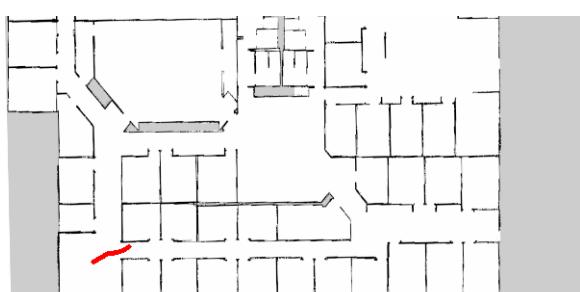
model13



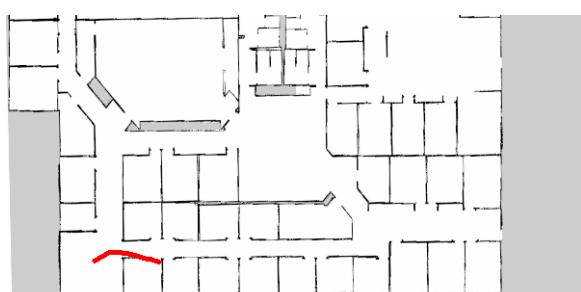
model14



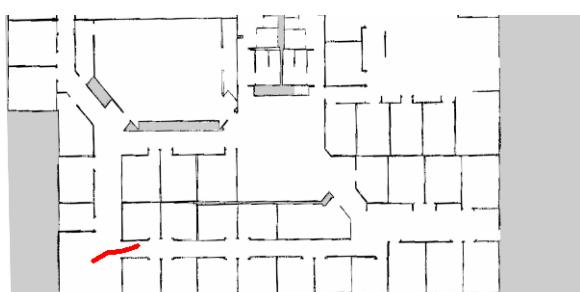
model15



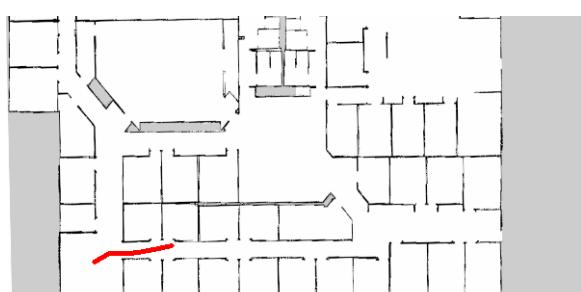
model16



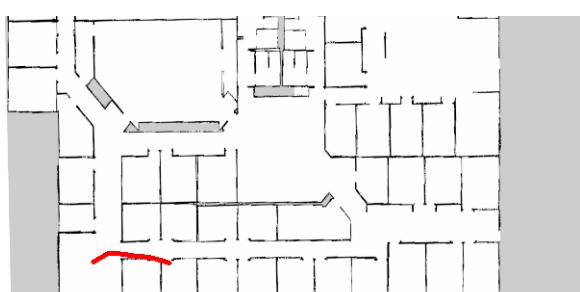
model17



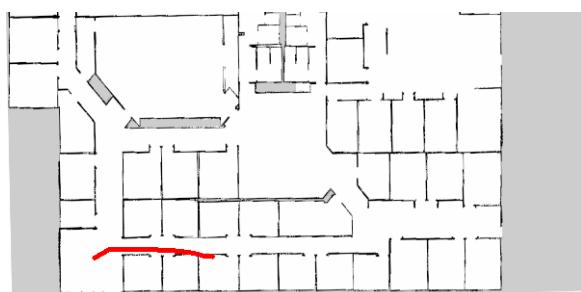
model18



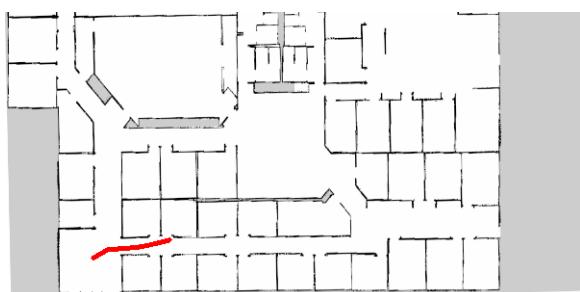
model19



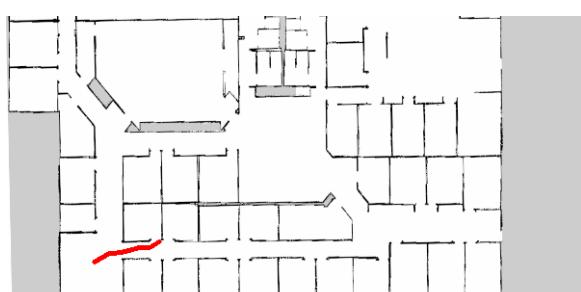
model20



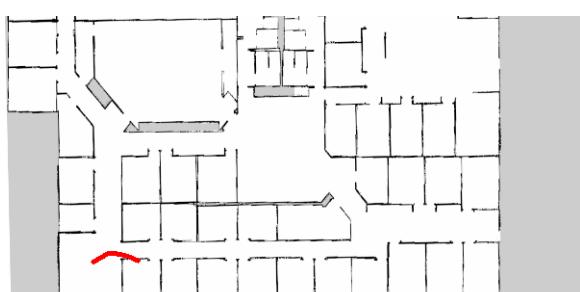
model21



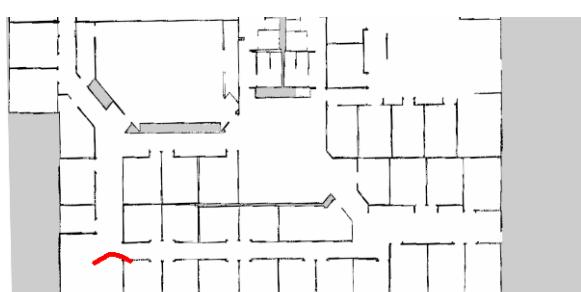
model22



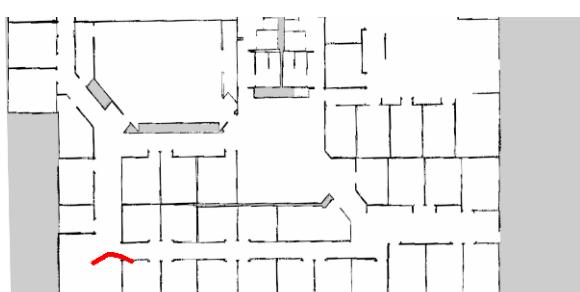
model23



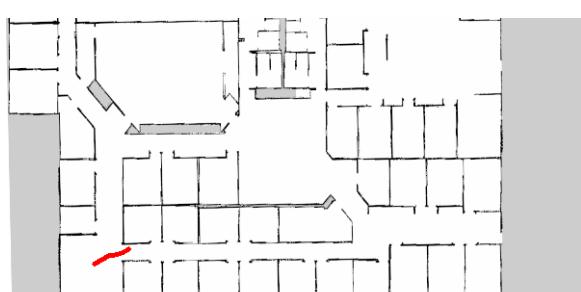
model24



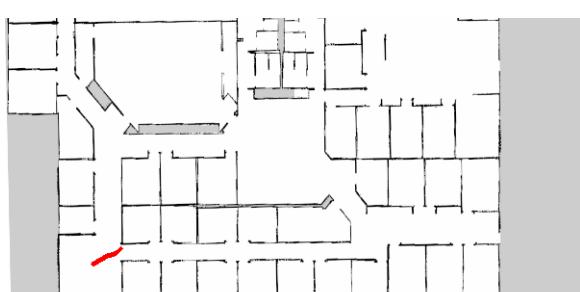
model25



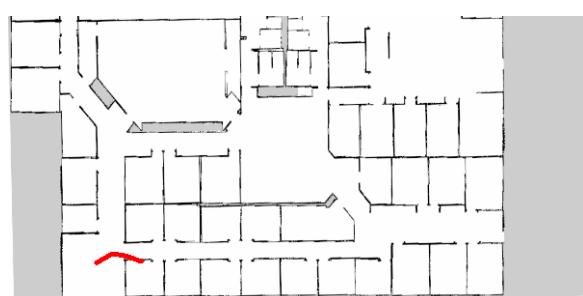
model26



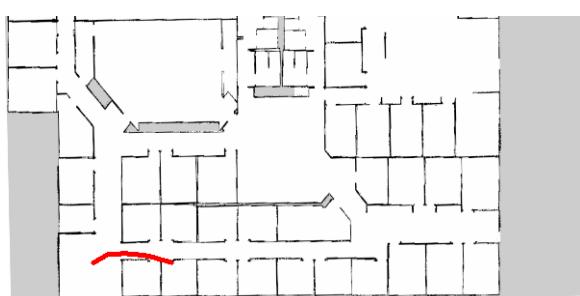
model27



model28

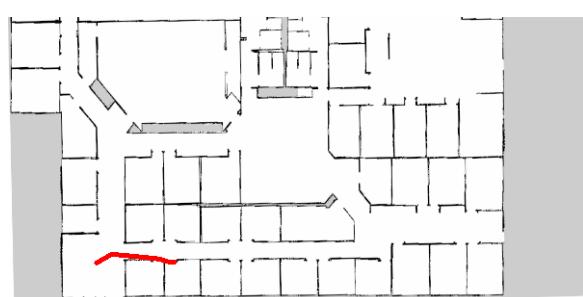


model29

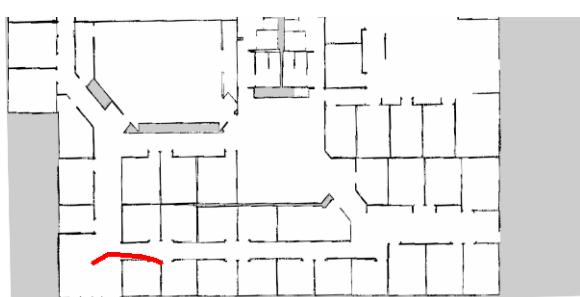


model30

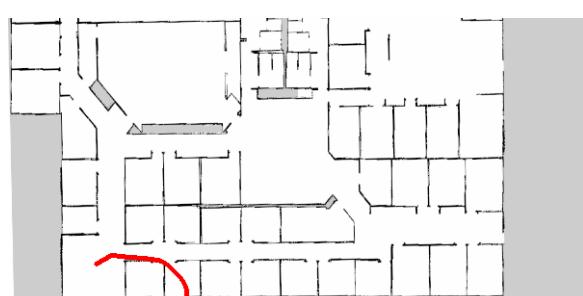
Driving trajectory for Experiment 2



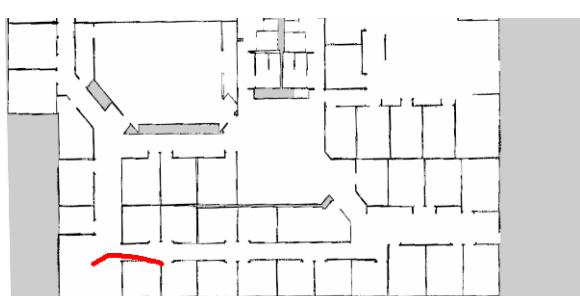
model1



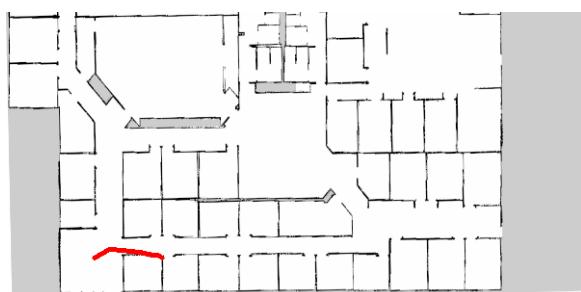
model2



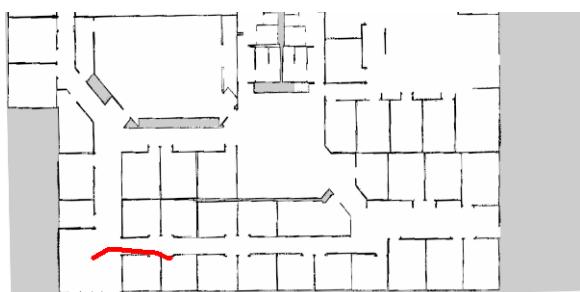
model3



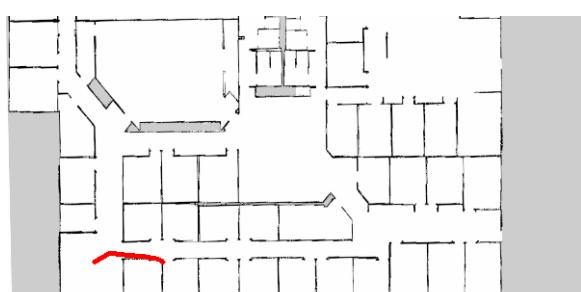
model4



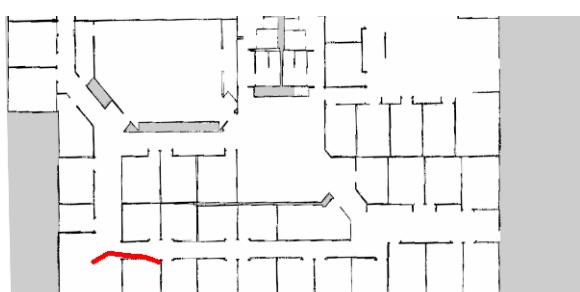
model5



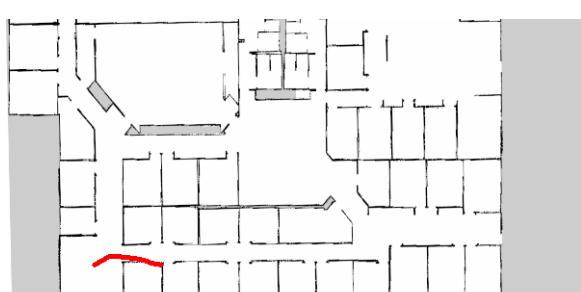
model6



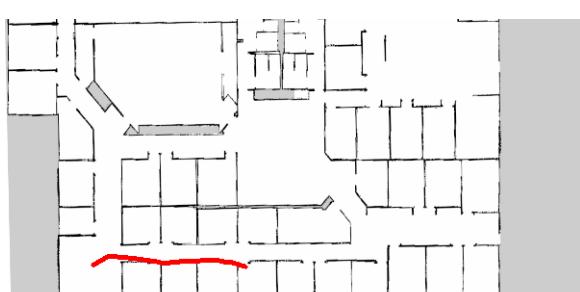
model7



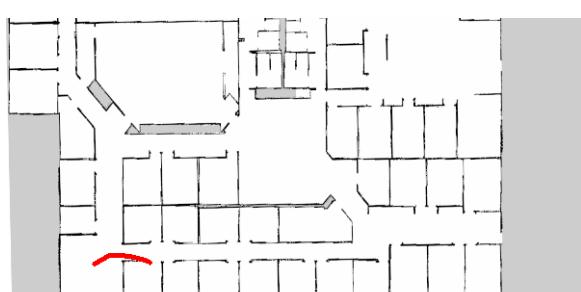
model8



model9



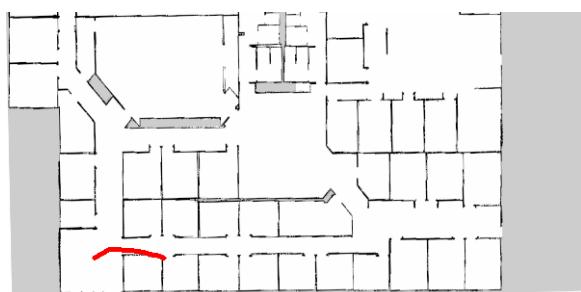
model10



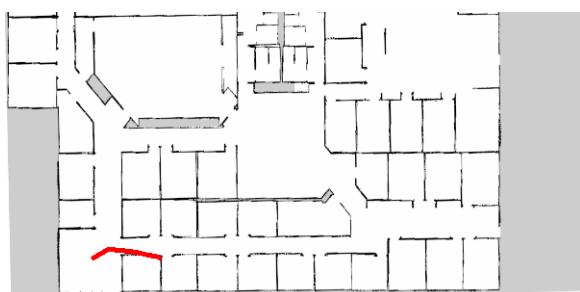
model11



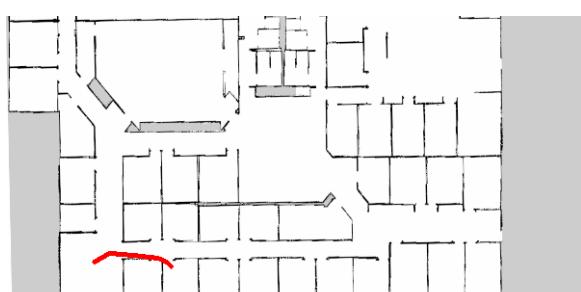
model12



model13



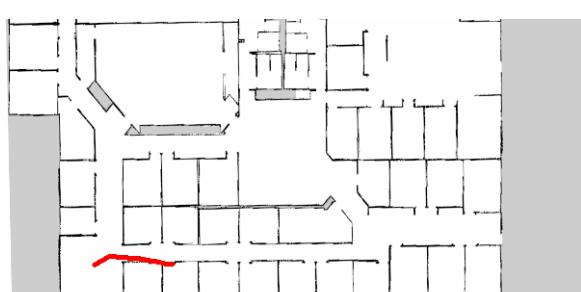
model14



model15



model16



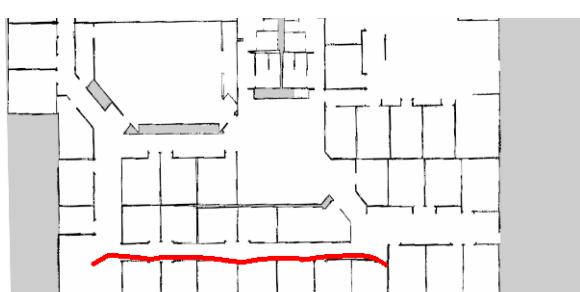
model17



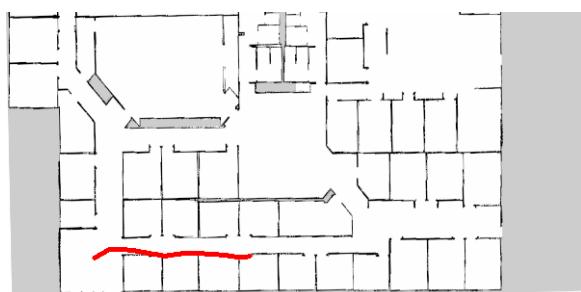
model18



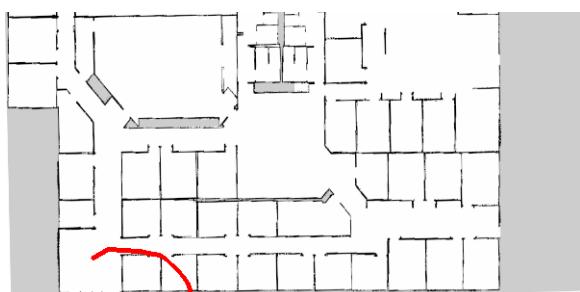
model19



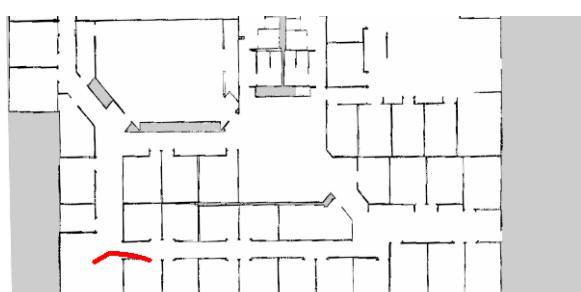
model20



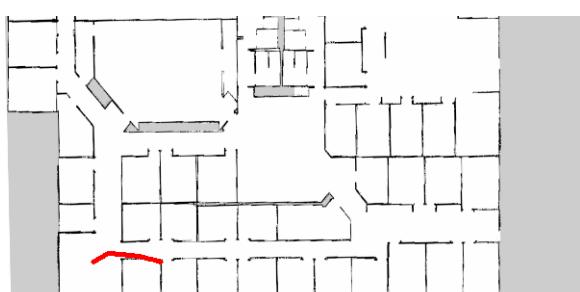
model21



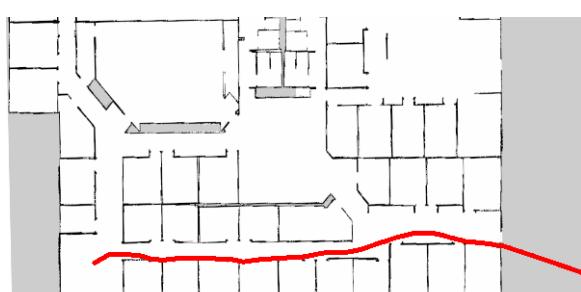
model22



model23



model24



model25



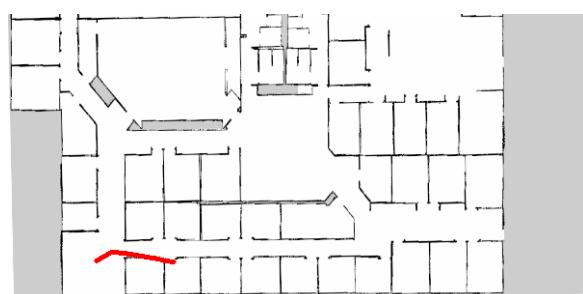
model26



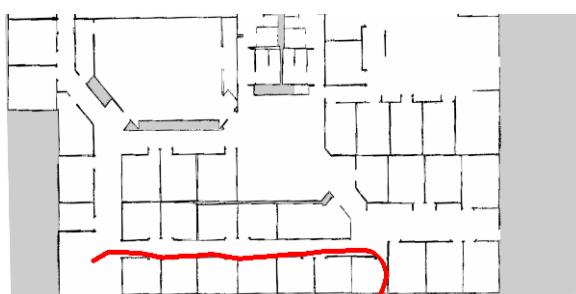
model27



model28

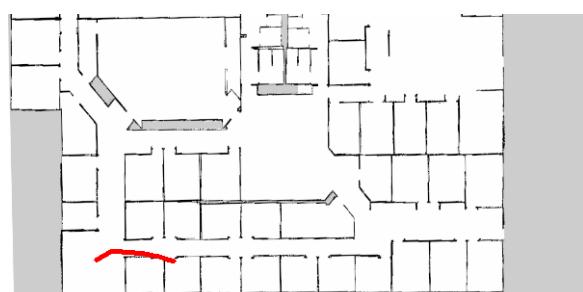


model29

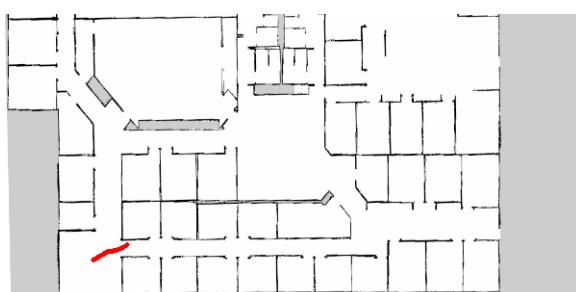


model30

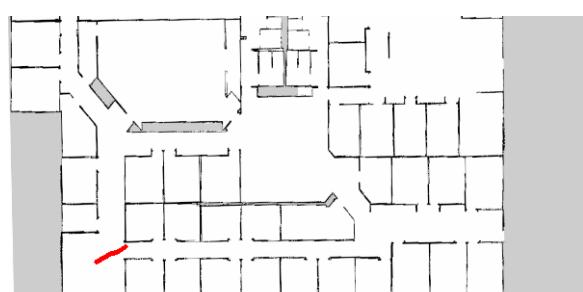
Driving trajectory for Experiment 3



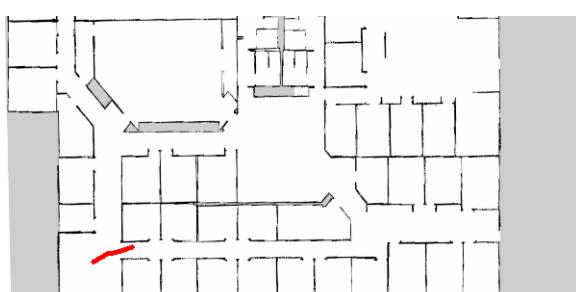
model1



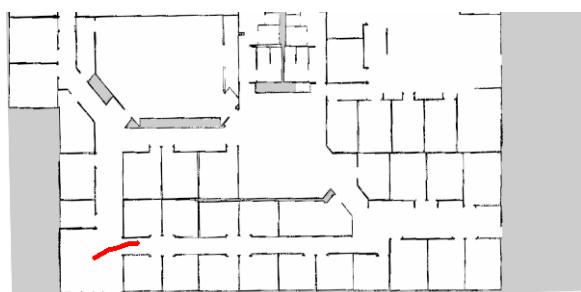
model2



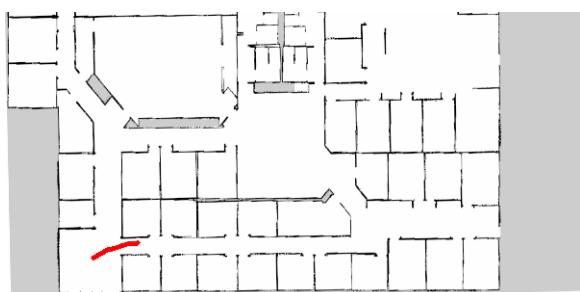
model3



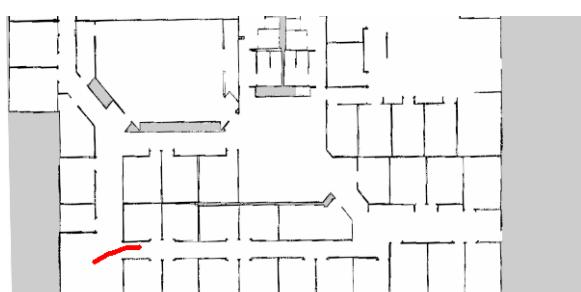
model4



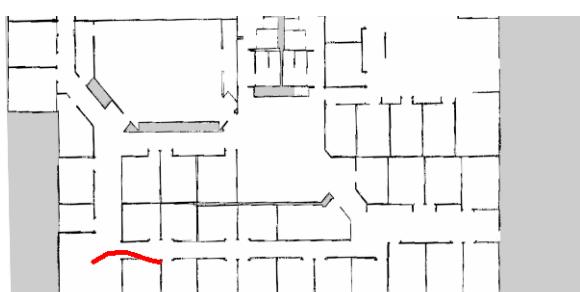
model5



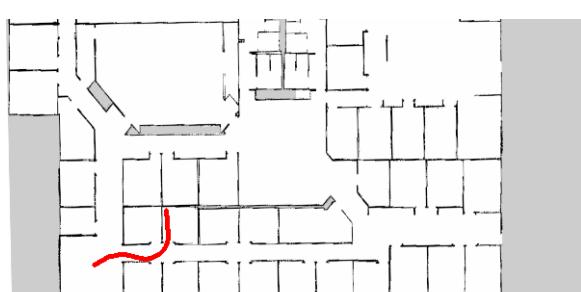
model6



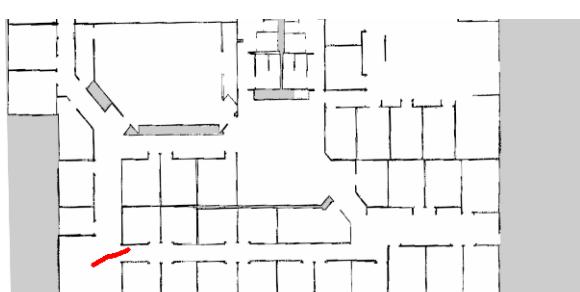
model7



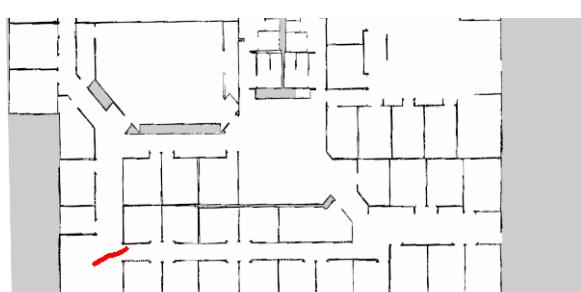
model8



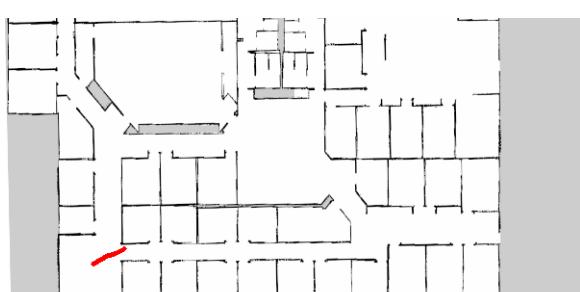
model9



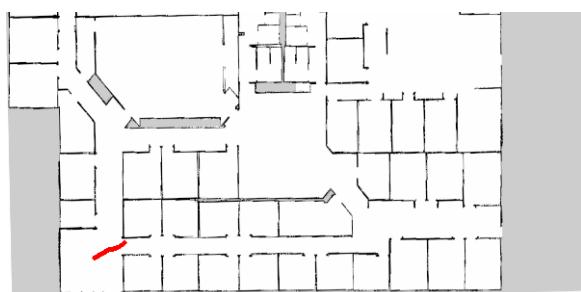
model10



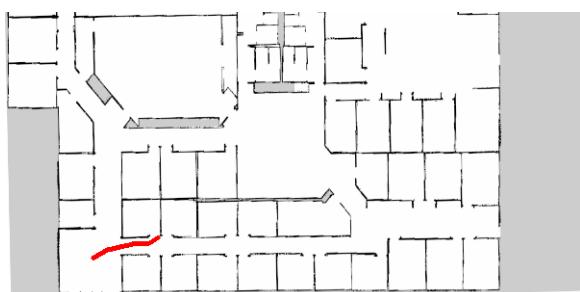
model11



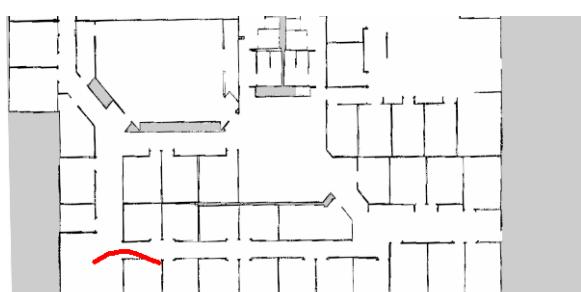
model12



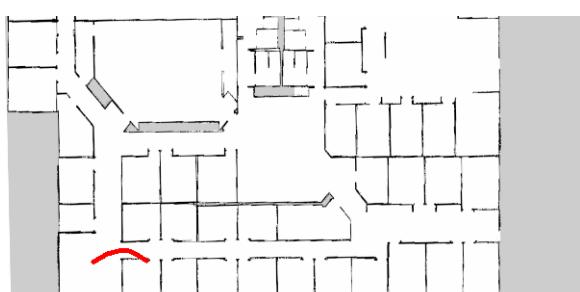
model13



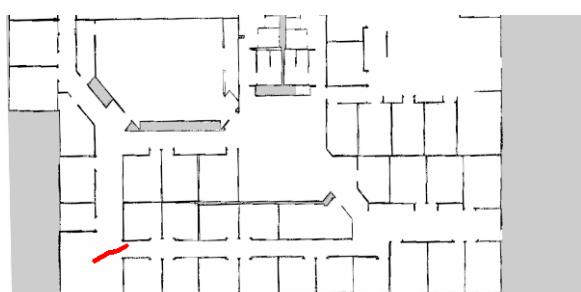
model14



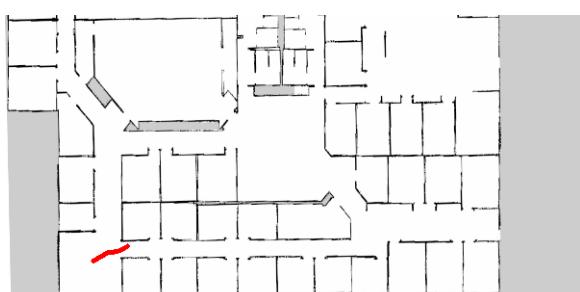
model15



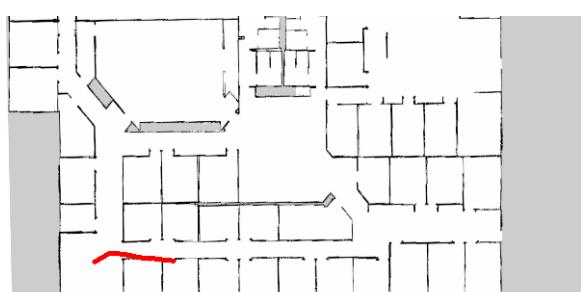
model16



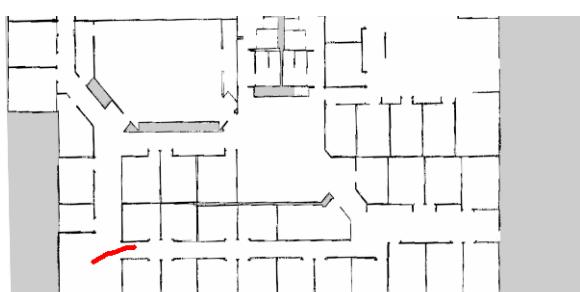
model17



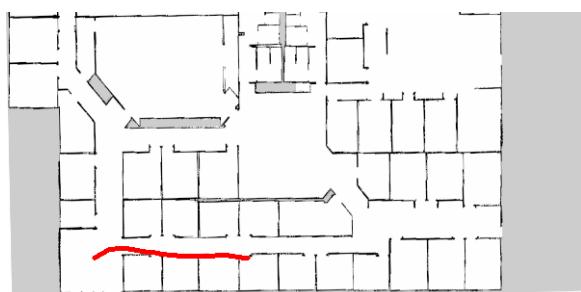
model18



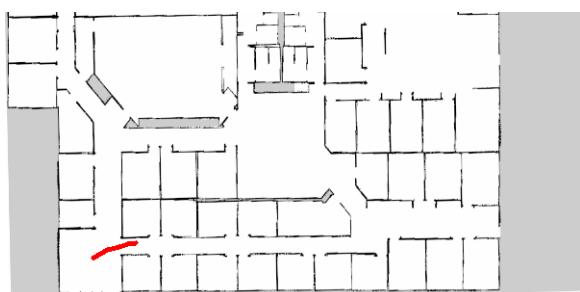
model19



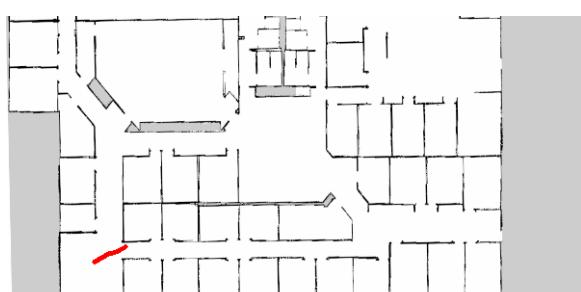
model20



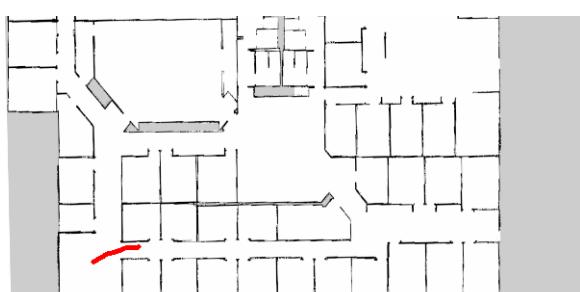
model21



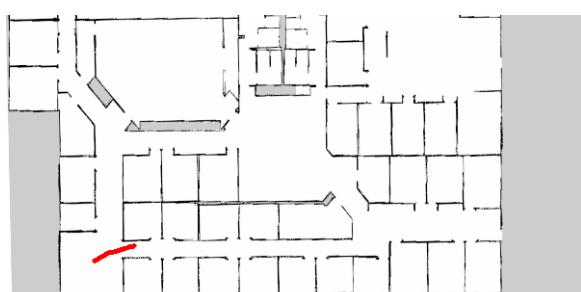
model22



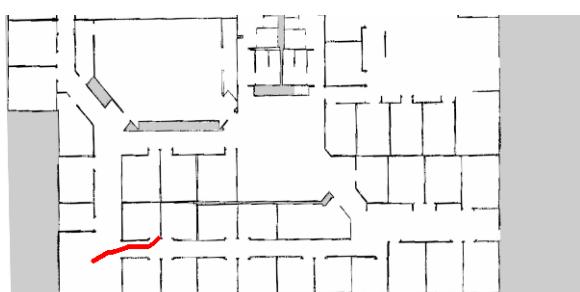
model23



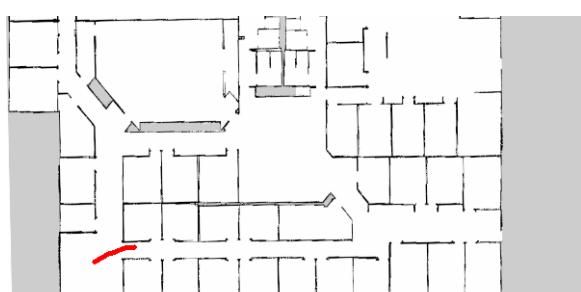
model24



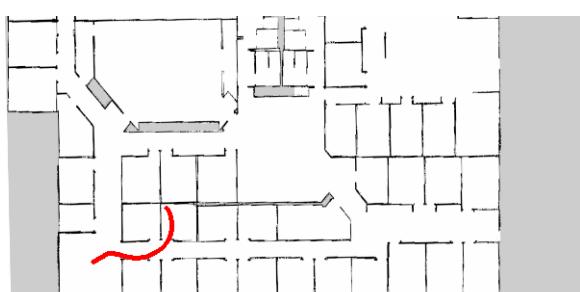
model25



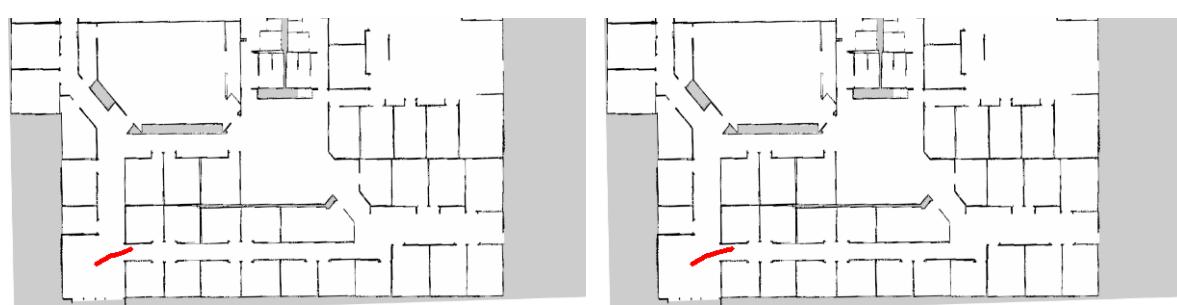
model26



model27



model28



model29

model30

Driving trajectory for Experiment 4

# 謝辞

本研究を進めるにあたり、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、サポートしてくださった同期の藤原柾様、白須和暉様には頭が上がりません。そして、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。