公立はこだて未来大学 2025 年度 システム情報科学実習
グループ報告書

Future University Hakodate 2025 Systems Information Science Practice
Group Report

プロジェクト番号/Project No.

11

プロジェクト名/Project Name

Make Brain Project

グループ名

グループ A

Group Name

Group A

プロジェクトリーダー/Project Leader

角脇輝映　Akira Kadowaki

グループリーダー/Group Leader

春原亮太　Ryota Sunohara

グループメンバー/Group Member

石川天翔　Takato Ishikawa

岩渕波空　Haku Iwabuchi

指導教員

香取勇一　栗川知己　島内宏和　佐藤直行

Advisor

Yuichi Katori　Tomoki Kurikawa　Hirokazu Shimauchi　Naoyuki Sato

提出日

2026 年 1 月 21 日

Date of Submission

Jan. 21, 2026

概要

本プロジェクトでは，強化学習を用いて，車両同士のすれ違いが困難な狭い道における自動運転車のゆずりあい行動の獲得を目指す．ここでゆずりあい行動とは，対向車との相対位置や待避所の有無を判断し，自車が一時停止して進路をゆずる，あるいは待避所へ移動して相手車両の通過を待つといった行動と定義する．住宅街などの狭い道では，相手車両の挙動に応じた柔軟な判断が求められるが，従来のルールベース制御ではこうした複雑な状況への対応が困難である．織田・横山・山下・蕨野・大岸・田中（2020）[1] によるラウンドアバウトでの協調行動など，先行研究では様々な形でこの課題にアプローチが行われている．本プロジェクトでは強化学習によるゆずりあい行動の獲得に焦点を当てる．前期には強化学習の理論と基本的な実装について理解することができた．後期に離散空間でのシミュレーションを作成し，実機にてゆずりあい行動を検証した．また，学習の成功には至らなかったが，連続空間におけるシミュレーションを作成した．これらの取り組みは，将来的な自動運転技術の高度化と，安全で円滑な交通の実現に寄与することが期待される．

キーワード　強化学習，マルチエージェント，深層学習, シミュレーション

Abstract

This project aims to use reinforcement learning to enable autonomous vehicles to acquire yielding behavior on narrow roads where passing is difficult. Here, yielding behavior is defined as actions such as determining the relative position and whether a passing space exists for an oncoming vehicle, then either stopping temporarily to yield the right of way or moving to a passing space to wait for the other vehicle to pass. In residential areas and other narrow roads, flexible decision-making based on the behavior of the other vehicle is required. However, conventional rule-based control struggles to handle such complex situations. Previous studies, such as the cooperative behavior in roundabouts by Oda, Yokoyama, Yamashita, Warabino, Okiishi, and Tanaka (2020) [1], have approached this challenge in various ways. This project focuses on acquiring yielding behavior through reinforcement learning. In the first semester, we gained an understanding of reinforcement learning theory and basic implementation. In the latter semester, we created a simulation in discrete space and verified the yielding behavior on an actual vehicle. We also created a simulation in continuous space, though it did not achieve successful learning. In the latter half, we created a simulation in discrete space and verified yielding behavior on actual hardware. Additionally, we created a simulation in continuous space, though learning success was not achieved. These efforts are expected to contribute to the future advancement of autonomous driving technology and the realization of safe and smooth traffic.

Keyword Reinforcement learning, Multi-agent systems, Deep learning, Simulation

目次

第 1 章　はじめに

1.1　背景

近年，自動運転技術は発展しており，高速道路や都市部の幹線道路といった整備された環境では，実用化に向けた実証実験や商用化が進んでいる．一方で，住宅街や山間部の林道などの狭い道では，対向車とのすれ違いをはじめとする複雑な判断が求められ，自動運転の適用には課題が残されている．また，工場や倉庫では無人搬送車や自律走行搬送ロボットが導入され，物流の効率化が進展している．しかし，通路幅が制限された環境下で複数のロボットが同時に稼働する場合，正面衝突を避けるための回避や，一方が道をゆずる協調行動が必須となる．狭い道において対向車とすれ違う場合，人間は交通量や対向車の動き，さらにはゆずりあいの精神といった様々な要素を考慮しながら判断を下している．しかし，これらの判断には相手の速度や意図の曖昧さという観測の不確実性や交通を妨げたくないという気づかいが伴う場合も多く，交通の停滞や事故の原因となっている．こうした現状から，対向車とのすれ違いを自動運転に任せることで，効率的かつ安全な交通の実現が期待される．問題に対する解決方法として強化学習がある．強化学習とは，エージェントが環境との試行錯誤を通じて報酬を得ながら行動を学習していく手法である．そのため，ゆずりあいが必要な場面では対向車の接近タイミングや位置関係に応じて，どのタイミングで待避スペースへ入るかといった協調判断を，強化学習を用いることで柔軟にモデル化できると考えられる．そこで本プロジェクトでは，狭い道において 2 台の実機のすれ違いを対象とし，マルチエージェント強化学習アルゴリズムを用いたゆずりあいの獲得を目的とする．具体的には，1 箇所の待避スペースが存在する直線道路のシミュレーション環境を構築し，学習モデルの実機へ適用し検証する．

（※文責：春原亮太）

1.2　先行研究

織田・横山・山下・蕨野・大岸・田中の研究：ラウンドアバウトにおける協調行動

織田ら [1] は，Raspberry Pi を搭載した RC カー（ラジコンカー）を用いた走行システムを構築し，交差点における交通流の最適化を目的とした PPO という強化学習アルゴリズムの適用を試みている．この研究では，5m × 8m の実験環境にて最大 6 台の車両を同時に走行させ，強化学習によって 1 台の挙動を制御するだけで全体の平均移動距離が増加することを示した．天井の QR コードや赤色マーカーを用いた自己位置推定，および情報共有サーバを介した車両間通信など，実機動作におけるシステム構成を具体的に提示している．本プロジェクトにおいても，Raspberry Pi や ArUco マーカーを用いた位置推定など，類似のハードウェア構成を採用しており，シミュレーションから実環境への適用において参考にした．

（※文責：春原亮太）

1.3 研究動機

2022 年度から 2024 年度にわたる先行プロジェクト [2, 3, 4] では，強化学習によるラジコンカーの制御や画像認識によるレーン・信号検知など，単一の車両における走行能力の獲得・向上に取り組んでいた．一方で，自律して他車両が存在する場合は考慮されていなかった．実社会に即して捉える場合，複数の車両による協調制御は不可欠である．しかし，複数の車両環境では，他者の学習状況によって自身の最適な行動が刻々と変化する非定常性の問題が生じる．狭い道でのゆずりあいにおいては，対向車の意図推定や待避タイミングの同期といった高度な相互作用が必要となるため，学習における探索空間が飛躍的に増大するという技術的困難がある．そこで，マルチエージェント強化学習を導入することにより，プロジェクトとして初めて，複数の車両間での協調行動の獲得に取り組む．

（※文責：石川天翔）

1.4 プロジェクトの目的

本プロジェクトの目的は，マルチエージェント強化学習を用いて，待避スペースのある狭い道における，ゆずりあいを学習するシステムを構築することである．ゆずりあいの成功の定義を以下に示す．

ゆずりあいの成功の定義

1. **衝突の回避**：エージェント同士が接触しないこと．
2. **膠着の回避**：狭い道での対峙による膠着状態を解消するために，一方が待避を選択することができること．
3. **目的地への到着**：2 つのエージェントが目的地とするゴールへたどり着くことができること．

まず離散および連続空間のシミュレーション環境でこれらの定義を満たす学習モデルを構築する．得られた学習モデルを実機へ適用し，物理現象や観測誤差が存在する実環境においても，学習されたゆずりあいが有効に機能するかを検証する．

（※文責：石川天翔）

1.5 プロジェクトの意義

・技術的意義：単一車両の制御から一歩進み，複数台での協調行動を強化学習で実現する点にある．これは，複数エージェントが相互に影響し合う複雑な環境下での意思決定アルゴリズムの実証として重要なステップとなる．

・社会的意義：自動運転の適用範囲を，従来の幹線道路から住宅街や山間部の林道といったゆずりあいが発生する難易度の高い道路へと拡張する鍵となる．これにより，狭い道路における交通停滞の緩和や，工場での配送車両等の自律走行化による物流の効率化が期待される．

（※文責：石川天翔）

第 2 章　理論的背景

本プロジェクト学習に関連する理論的背景に言及し，問題解決に必要なスキルや手法について述べる．また，プロジェクトの遂行に関連性が高い本学の専門科目についても言及する．

2.1　強化学習

強化学習は，エージェントが環境との試行錯誤を通じて報酬をもとに行動を決定する学習手法である．エージェントはある環境において，現在の状態を観測し，それに基づいて行動を選択する．環境はその行動に応じて次の状態へと遷移し，同時にエージェントへ報酬を与える．エージェントはこの一連のプロセスを繰り返しながら，将来にわたって得られる報酬の総和を最大化するような方策を学習することを目標とする．

第 2.1.1 節，第 2.1.2 節では離散空間におけるシミュレーションにおいて使用したアルゴリズムについて説明する．また，以下の 7 つの用語は強化学習で使われることが多いため，簡単な説明とともに言及する．

エージェント　学習の主体であり，環境からの情報に基づき行動を決定する．

環境　エージェントが相互作用する対象であり，エージェントの行動に対して状態の変化と報酬をフィードバックする．

状態と観測　状態とは環境の状況を示す情報である．エージェント自身から得られる情報は観測と呼ばれる．

行動　エージェントが実行可能な動作の集合である．

報酬　行動の結果に対する評価値である．

方策　状態や観測に基づいて行動を決定するためのルールや確率分布である．

（※文責：春原亮太）

2.1.1　Q 学習

Q 学習（Q-learning）[5] は，エージェントがある状態で特定の行動を選択した際の評価を「Q 値（行動価値）」と呼ばれる数値で表現し，環境との試行錯誤を通じてその数値を更新していく学習手法である．Q 値の更新式を式 (2.1) に示す．

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left(r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \right) \tag{2.1}$$

式 (2.1) における各変数は，エージェントの意思決定プロセスを数学的に定義したものである．時刻 t における状態を s_t，選択した行動を a_t とし，その結果として環境から得られる即時報酬を r_{t+1} と定義する．更新される $Q(s_t, a_t)$ は，特定の状態で特定の行動を選択した際に，将来的に得られる報酬の合計を予測した期待値を表している．

また，学習の挙動を制御する重要なパラメータとして，学習率 α と割引率 γ が用いられる．学習率 α は新規情報を既存の Q 値に反映させる割合を示し，高いほど新情報の適応は早まるが，学

習の安定性は損なわれる傾向にある．割引率 γ は将来得られる報酬を現在の価値へと換算する指標であり，値が大きいほど長期的な報酬を重視した戦略が獲得され，値が小さいほど目先の報酬を優先する短絡的な性質となる．

2.1.2 ϵ-greedy 法

強化学習においては，現在の Q 値に基づき最も評価の高い行動を選択する活用と，未知の行動を試行する探索のトレードオフが重要となる．これを制御する代表的な手法が ϵ-greedy 法である．

ϵ-greedy 法では，確率 ϵ でランダムに行動を選択し（探索），残りの確率 $1-\epsilon$ でその時点で最も Q 値が高い行動を選択する（活用）．

$$a_t = \begin{cases} \mathrm{argmax}_a Q(s_t, a) & (\text{確率 } 1-\epsilon \text{ のとき}) \\ \text{random action} & (\text{確率 } \epsilon \text{ のとき}) \end{cases} \tag{2.2}$$

一般に，学習初期段階では ϵ を大きく設定することで環境を広く探索させ，学習が進むにつれて ϵ を徐々に減少させることで，獲得した最適な行動に収束させていく手法が広く用いられている．

（※文責：春原亮太）

2.1.3 独立 Q 学習と非定常性

複数のエージェントが同時に学習を行うマルチエージェント環境において，最も単純なアプローチは独立 Q 学習（Independent Q-Learning: IQL）である．IQL では，各エージェントが他者の存在を環境の一部と見なし，自身の局所的な観測と報酬に基づいて個別に Q 値を更新する．しかし，この手法には非定常性という本質的な課題が存在する．IQL においては各エージェントが同時に学習を進めるため，あるエージェントから見た環境の遷移規則は，他者の学習状況に応じて常に変化し続けることになる．このように，学習が進むにつれて正解となる行動が変動し続けるため，Q 値の更新が安定せず，学習の収束が困難になる．本プロジェクトにおいても，この非定常性は大きな障壁となる．例えば，一方のエージェントが待避所へ入るという行動を学習し始めた途端，他方のエージェントにとってはそのまま直進することが最適な行動へと変化する．

（※文責：石川天翔）

2.2 深層強化学習

深層強化学習とは，強化学習と深層学習を組み合わせたものである．強化学習との大きな違いはニューラルネットワークを取り入れている点である．

2.2.1 Proximal Policy Optimization

連続空間における深層強化学習の代表的なアルゴリズムとして Proximal Policy Optimization（以下 PPO）が挙げられる．各エージェント i の Actor ネットワーク（パラメータ θ_i）は，以下のクリップ付き目的関数を最大化するように更新される．

$$J(\theta_i) = \mathbb{E}_t \left[\min \left(r_t(\theta_i)\hat{A}_t, \mathrm{clip}(r_t(\theta_i), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) + \beta S[\pi_{\theta_i}] \right] \tag{2.3}$$

ここで，$r_t(\theta_i)$ は旧方策と新方策の確率比，ϵ は更新幅を制限するハイパーパラメータである．$\hat{A}_t$ の算出には，Generalized Advantage Estimation（GAE）を用いる．一般に，報酬の和を直接用いる手法は分散が大きく学習が不安定になりやすく，一方で価値関数のみを用いる手法はバイアスが生じやすいという課題がある．GAE は，以下の TD 誤差 δ_t を用いて，これらを調整する役割を担う．

$$\delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t) \tag{2.4}$$

これを用い，GAE はハイパーパラメータ $\lambda \in [0, 1]$ を導入して，次のように定義される．

$$\hat{A}t = \sum k = 0^{\infty} (\gamma\lambda)^k \delta_{t+k} \tag{2.5}$$

ここで γ は割引率である．λ の値を調整することで，バイアスと分散のトレードオフを制御することが可能となる．また，$S[\pi_{\theta_i}]$ は方策のエントロピーを表し，係数 β （エントロピー係数）によって制御される．この項は，方策が決定論的になりすぎるのを防ぎ，エージェントに探索を促すことで局所解への早期収束を抑制する役割を持つ．

一方，Critic ネットワーク（パラメータ ϕ）は，自身の状態を入力とし，以下の損失関数を最小化するように学習を行う．

$$L(\phi) = \mathbb{E}_t \left[\left(V_\phi(s_{global,t}) - R_t \right)^2 \right] \tag{2.6}$$

ここで，R_t は割引報酬和である．このように，Critic が全体情報を活用して正確な評価（価値判断）を行い，それに基づいて各 Actor が分散環境下での最適な行動ルールを学習する．

（※文責：春原亮太）

2.2.2 学習手法：Multi-Agent Proximal Policy Optimization

連続的な行動空間におけるマルチエージェントに対応するため，Multi-Agent Proximal Policy Optimization（以下，MAPPO）[6] を導入した．これは集中学習・分散実行の枠組みに基づいている．集中学習・分散実行とは学習時は全エージェントの情報（グローバル状態）を利用し，実行時は各エージェントの局所観測のみを利用するというものである．具体的には，Critic ネットワークは個々のエージェントの観測ではなく，環境全体のグローバル状態 (全エージェントの位置や速度を含む) を入力として価値関数を学習する．これにより，Critic は他者の行動の影響を含めた正確な価値評価が可能となる． 一方，Actor ネットワークは各エージェントの局所観測のみを入力として行動を決定する．

（※文責：春原亮太）

2.3 Sim2Real

Sim2Real（Simulation to Real）とは，仮想的なシミュレーション環境で学習したアルゴリズムや制御ルールを，現実世界の物理ロボットへ適用するプロセスのことである．現実のロボットに直接学習させるには，主に以下の 2 つの大きな障壁が存在する．

- **物理的なリスク**: 学習中のロボットが予想外の動きをして，自分自身や周囲の機材を損壊させる危険がある．

- **時間の制約**: 実機では時間の流れを速めることはできないが，シミュレーションならコンピュータ上で何倍もの速さで試行錯誤を繰り返すことが可能である．

Sim2Real を実現する上で最も大きな課題は，シミュレータと現実世界の間に存在する物理的なズレ，すなわちリアリティギャップである．シミュレーションがいかに精密であっても，現実の床面の摩擦，モーターの出力特性の個体差，センサーに混入するノイズ，通信のわずかな遅延などを完全に再現することは困難である．

（※文責：石川天翔）

2.4　関連性が高い本学の専門科目

本プロジェクトに関連性の高い専門科目として,「機械学習」,「画像工学」の 2 科目が挙げられる．まず，機械学習では，深層学習，生成モデルについて学んだ．連続空間における強化学習アルゴリズムの実装においては深層学習の知識が役に立った．また，学習率や割引率，バッチサイズの調整においても授業で行ったパラメータの調整が役に立った．次に，画像工学で学んだ座標変換の知識は，カメラを使った位置推定に活用した．ArUco マーカーを認識し，カメラ映像の座標を現実の地図上の座標に変換する射影変換の実装において，授業で学んだ知識が助けとなった．

（※文責：石川天翔）

第 3 章　プロジェクト学習の目標

本プロジェクトの学習目標は，狭い道におけるゆずりあいをエージェントに学習させ，その有効性を実機において実証することである．具体的には，まず問題の単純化を行うため，環境をマス目状に見立てた離散空間におけるシミュレーション環境を構築する．ここでは Q 学習を用いて，エージェントが報酬を手がかりにもう一方のエージェントとのゆずりあいを学習できるか検証する． 次に，より現実的な走行環境に近づけるため，エージェントの位置や速度を連続値として扱う環境へと拡張を行う．連続空間においては状態数が膨大となるため，Q 学習において扱ったテーブル形式の手法に代わり，深層強化学習を導入する．最終的には，これらの手法を用いて学習したモデルを実機で動かし，実環境においても，シミュレーションと同様にゆずりあいが再現されることを確認する．

（※文責：春原亮太）

第 4 章　目的を達成するための手段

4.1　シミュレーションでのソフトウェア環境

シミュレーションで使用したシステム構成を表 4.1 に，ソフトウェア環境を表 4.2 に示す．シミュレーションでは開発言語として Python を用いた．Pygame は物理シミュレーションの描画を担う．PyTorch は深層学習のライブラリであり，強化学習のアルゴリズム（後述）における Actor ネットワークおよび Critic ネットワークの構築，勾配計算，および GPU を用いた推論を担う．Optuna はベイズ最適化アルゴリズムを用い，報酬を最大化するための最適なハイパーパラメータを自動的に探索することを担う．NumPy はエージェントの座標計算，行列演算，および観測データの数値処理を高速に行うため使用する．

表 4.1　シミュレーションのシステム構成

項目	仕様
OS	Ubuntu 24.04.3 LTS
CPU	AMD Ryzen 7 3700X
GPU	NVIDIA GeForce RTX 4070 Ti

表 4.2　ソフトウェア環境

項目	バージョン
Python	3.10.12
Pygame	2.6.1
Pytorch	2.7.1
Matplotlib	3.10.3
Optuna	4.6.0
NumPy	2.2.6

（※文責：春原亮太）

4.2　離散空間におけるシミュレーション

離散空間ではQ学習というアルゴリズムを用いた．

4.2.1　シミュレーション環境

狭い道でのゆずりあいの場面を簡略化した．具体的には道路という環境を離散化し，マス目上で表現した．道路の全長は 5 マスに設定した．これはシミュレーション環境から実環境に向けて接続しやすくするとともに，状態空間の冗長性を排除してゆずりあいという問題の本質的な要素のみを抽出するためである．道路の中央である 3 マス目にのみ，待避所を配置した．この環境では 3 マス

目以外から待避スペースへの移動は不可能とした．待避スペースを中央に配置することで，環境に対称性を持たせた．これは，エージェントの初期位置や進行方向に依存せず，状況に応じたゆずりあい行動が獲得可能であるかを検証することを目的とした．また，中央への配置は 2 つのエージェントが同時に待避所へ到達しやすい状況を生み出す．こうした条件下においても，ゆずりあいによる衝突回避が可能であるかを調査するためである．図 4.1 は作成したシミュレーションである．このシミュレーションでは，道路と見立てた 5 マスとその中央に緑色で塗りつぶされた待避スペースがある．

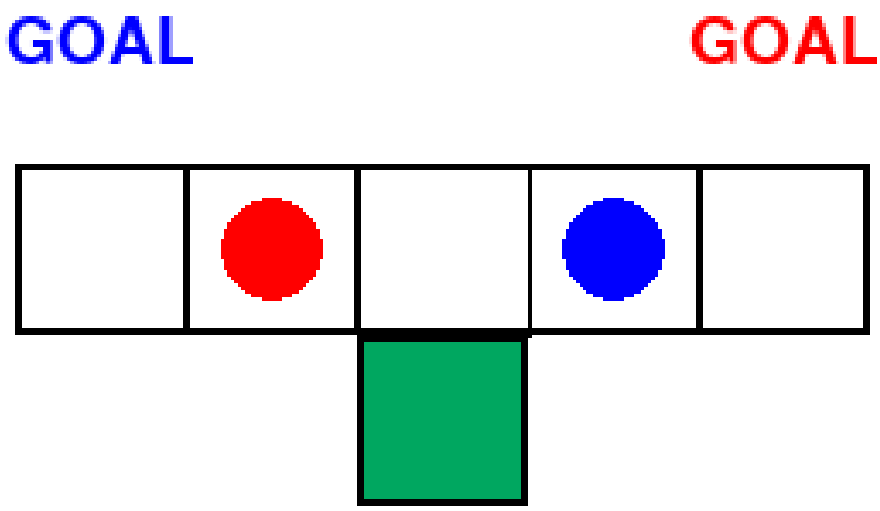

図 4.1　シミュレーション環境

表 4.3　シミュレーション環境設定

カテゴリ	項目	設定値
環境全体	ウィンドウサイズ	400 pixel × 250 pixel
道路	全長	5 グリッド
	車線構成	基本 1 車線＋中央の位置のみ待避可能
	グリッドサイズ	50 pixel × 50 pixel
エージェント	個体数	2 (Agent A, Agent B)
	描画サイズ (半径)	15 pixel (直径 30 pixel)
	初期位置	Agent A: 0, Agent B: 4

（※文責：石川天翔）

4.2.2　エージェント設定

状態空間の規模

：環境は 5×2 の格子状であるが，物理的な制約（待避所が中央 1 マスのみ）により，各エージェントがとり得る有効な位置・車線の組み合わせは 6 通りである．したがって，2 エージェントの総状態数は $6^2 = 36$ となる．

行動空間の規模

エージェントは各ステップにおいて，「前進」「後退」「横移動」「待機」の 4 つの離散行動を選択可能である．

Q テーブルの構成

各エージェントに対し，状態数 × 行動数 の行列として Q テーブルを構成した．状態に基づく要素数はエージェントあたり 144 である

4.2.3 行動のマスキングによる学習の効率化

エージェントが物理的に不可能な行動（壁への衝突や車線外への逸脱）を選択しないよう，行動マスキングを実装した．具体的には，現在の座標と車線に応じて実行可能な行動リストを生成し，それ以外の行動の Q 値が更新されないよう制御した．

（※文責：石川天翔）

4.2.4 学習設定

表 4.4 の通り，学習率，割引率，探索率は組み合わせを網羅的に探索するグリッドサーチを用いた．

表 4.4 グリッドサーチの探索空間

項目	数値
学習率 (α)	0.1, 0.5
割引率 (γ)	0.9, 0.95, 0.99
探索率 (ϵ)	0.1, 0.3

その結果，Q 学習のパラメータは以下の表 4.5 となった．なお，探索率は固定化した．これは，50 エピソード程度で学習が収束し，減衰させる

表 4.5 Q 学習のパラメータ設定

パラメータ	数値
試行回数 (Runs)	300
エピソード数	500
最大ステップ数	50
学習率 (α)	0.1
割引率 (γ)	0.9
探索率 (ϵ)	0.1

（※文責：春原亮太）

4.2.5 報酬設計

報酬関数の設計値を表 4.6 に示す．各数値は，エージェントが自身の効率と全体の安全を比較し，協調行動を発現するように設定した．

表 4.6 報酬設計

項目	数値	設計の意図と役割
片方ゴール	+10	エージェントが目的地へ到達することへの報酬
両者ゴール	+20	システム全体の利益を最大化するための報酬
衝突ペナルティ	−30	衝突しないようにするための報酬

（※文責：石川天翔）

4.3 連続空間におけるシミュレーション

連続空間における強化学習では，離散空間で扱った Q 学習では対応することができない．これは Q 学習が状態と行動の組み合わせをテーブルとして保持しているため，連続空間という実数値のすべての状態と行動をテーブルとして保持することが難しいからである．MAPPO という強化学習アルゴリズムを用いた．

4.3.1 シミュレーション環境

図 4.2 はシミュレーション環境，表 4.7 はシミュレーション環境で使用したパラメータである．図 4.2 では，赤色および青色の円形オブジェクトは各エージェントを表現している．白色の領域は走行可能な道路，緑色の領域は待避スペースをそれぞれ示している．なお，黒色の領域は道路外領域であり，エージェントの進入は制限されている．離散空間でのシミュレーションと同じように，道路の中央に待避スペースを設けた．また，エージェントの大きさに対して道路の幅を小さくした．これは，エージェントが道路の端から端まで蛇行するような無駄な探索を抑制し，目的地へ向かう直進という基本的な行動を早期に獲得させるためである．一方で，中央の待避スペースは離散空間，またエージェントの大きさよりも相対的に広く設計した．これは，連続空間では細かい制御が必要であるため，許容する誤差を大きくするためである．また，エージェントの初期位置に対して配置ノイズを与えている．これは特定の開始位置に依存した固定的な行動パターンの獲得を抑制するためである．

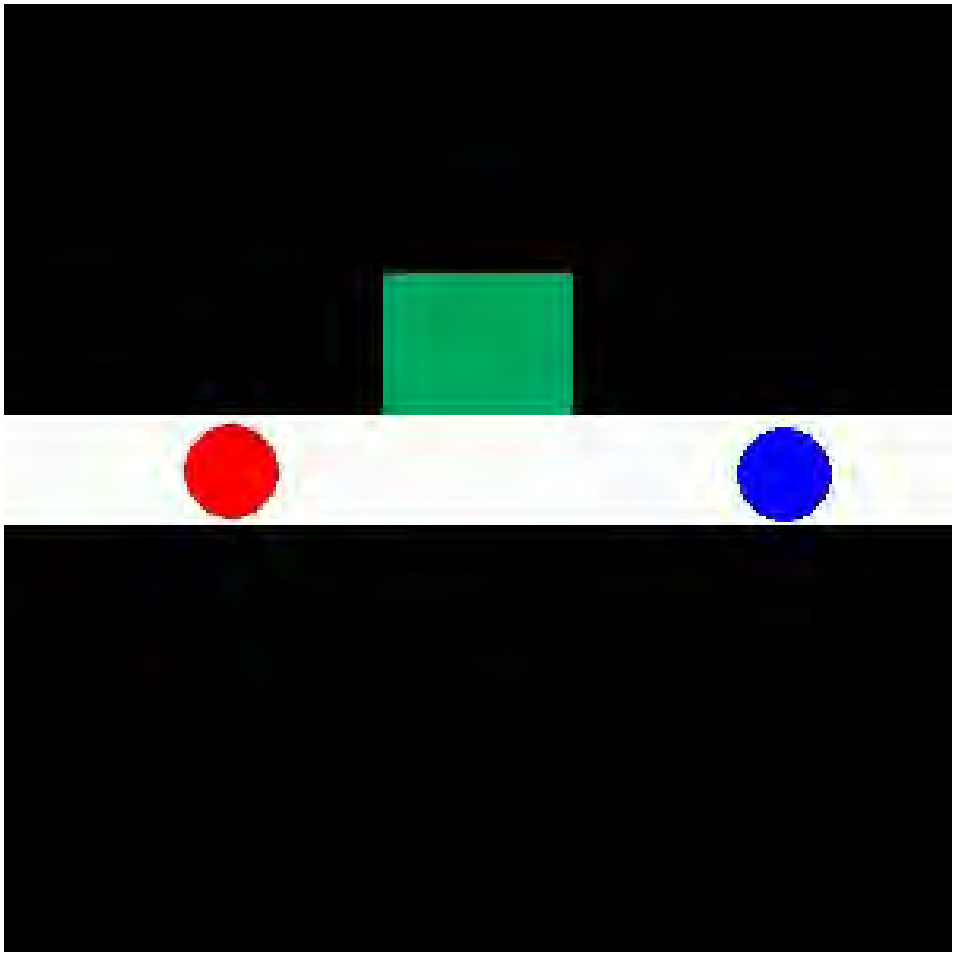

図 4.2　連続空間でのシミュレーション環境

表 4.7　シミュレーション環境

カテゴリ	パラメータ項目	数値
環境全体	フィールドサイズ	300×300 ユニット
	最大ステップ数	600 ステップ
道路	幅（X 方向）	300（全幅）
	高さ（Y 方向）	35（Y 座標: 130 〜 165）
待避所	位置（X 方向）	120 〜 180（中央配置）
	長さ	60
	奥行き	45（Y 座標: 85 〜 130）
エージェント	半径（直径）	15（30）
	最大速度	5.0 / step
初期配置	エージェント A（左）	$X = 50$ 付近
	エージェント B（右）	$X = 250$ 付近
	配置ノイズ	$X : \pm 40, Y : \pm 2.0$

（※文責：春原亮太）

4.3.2 エージェント設計

本シミュレーションにおけるエージェントの行動空間および観測空間の定義を以下に示す．本システムは連続空間上で定義されており，行動および観測はすべて連続値ベクトルとして表現される．

行動空間

エージェントへの入力となる行動は, 2 次元の連続値ベクトルで定義される．各次元は $[-1.0, 1.0]$ の範囲とした．

表 4.8　行動空間

次元	項目	範囲
第 1 次元	X 軸方向の速度	$[-1.0, 1.0]$
第 2 次元	Y 軸方向の速度	$[-1.0, 1.0]$

観測空間

各エージェントは環境から局所的な観測を受け取る．観測ベクトルは 12 次元で構成され，自己の状態だけでなく，相手エージェントや待避所との相対的な関係を含んでいる．すべての値は $[-1, 1]$ 程度に収まるように正規化されている．

表 4.9　観測空間の定義 (12 次元)

次元	項目	説明
0 – 1	自己位置	画面サイズによる正規化
2 – 3	自己速度	最大速度による正規化
4 – 5	相対位置	相手エージェントとの相対ベクトル
6 – 7	相対速度	相手エージェントとの相対速度
8 – 9	待避所相対位置	待避所中心までのベクトル
10	相手待避フラグ	相手が待避所内にいるか
11	エージェント ID	エージェント A: 1.0, エージェント B: -1.0

なお，Critic ネットワークの学習に使用される状態は，両エージェントの位置・速度情報 (各 4 次元) を結合した計 8 次元のベクトルとして定義した．

（※文責：春原亮太）

4.3.3 学習設定

本実験では，強化学習の性能を最大化するために，ベイズ最適化フレームワークである Optuna を用いてハイパーパラメータの探索を行った．20 回の試行を通じて得られた最適なハイパーパラメータおよび，固定パラメータの一覧を 表 4.10 に示す．固定パラメータは OpenAI が作成した強化学習ライブラリである Stable-Baselines3[7] における MAPPO のデフォルト値を参考にした．

表 4.10　設定したハイパーパラメータ

ハイパーパラメータ	数値
Actor 学習率	4.30×10^{-4}
Critic 学習率	3.36×10^{-4}
割引率	0.982
エントロピー係数	0.011
更新エポック数	8
バッチサイズ	128
隠れ層サイズ	64

表 4.11　固定したパラメータ

パラメータ	数値
GAE（一般化アドバンテージ推定）係数	0.95
PPO のクリップ範囲	0.2
勾配クリッピングの閾値	0.5
初期探索分散の対数値	-0.5
行動空間の下限値	-1.0
行動空間の上限値	1.0

（※文責：春原亮太）

4.3.4　報酬設計

以下の表 4.12 は与えた報酬である．報酬を与える際は，－１から１の範囲に正規化した．これは，極端に大きな報酬が入力されると，ネットワークの勾配が爆発し，パラメータが不安定になる可能性があるからである．

表 4.12　報酬関数のパラメータ設定と設計意図

項目	値	設計の意図と役割
個人ゴール到達	+2.0	ゴールへの到達を評価し，目的地への到達を動機付ける．
チームゴール	+5.0	2 つのエージェントのゴールを優先させ，ゆずりあいを促進する．
道路の真ん中を走る報酬	0.01	道路の中央の移動を促す．
衝突	−0.5	エージェントと環境の安全性を優先し，衝突回避を学習させる．
壁接触	−0.05	走行可能領域内での移動を強制し，無駄な探索を減らす．
ステップ罰	−0.0001	待機行動を許容しつつも，最短時間でのゴール達成を促す．
後退罰	−0.005	進行方向を安定させ，不必要な逆走を防ぐ．

（※文責：春原亮太）

4.4 実機

4.4.1 システム構成

システム構成として，実機の制御処理を担うパソコン（以下，PC）と，PC からの命令に基づき物理的な動作を行う 2 台の実機によって構成した． 具体的には，PC で強化学習モデルを実行し，2 台の実機の行動の決定を行い，その行動を実機へ送信する集中制御とした．また，実機の位置情報を取得するセンサとして Web カメラを用い，その画像解析処理も PC 上で実施した．実機には，図 4.3 の Yahboom 社製の「Raspbot v2」[8] を使用した．この実機のコンピュータとして Raspberry Pi 5 を搭載した．表 4.13，表 4.14 では PC，2 台の実機のシステム構成を示す．また，図 4.4 は実機における環境を示したものである．

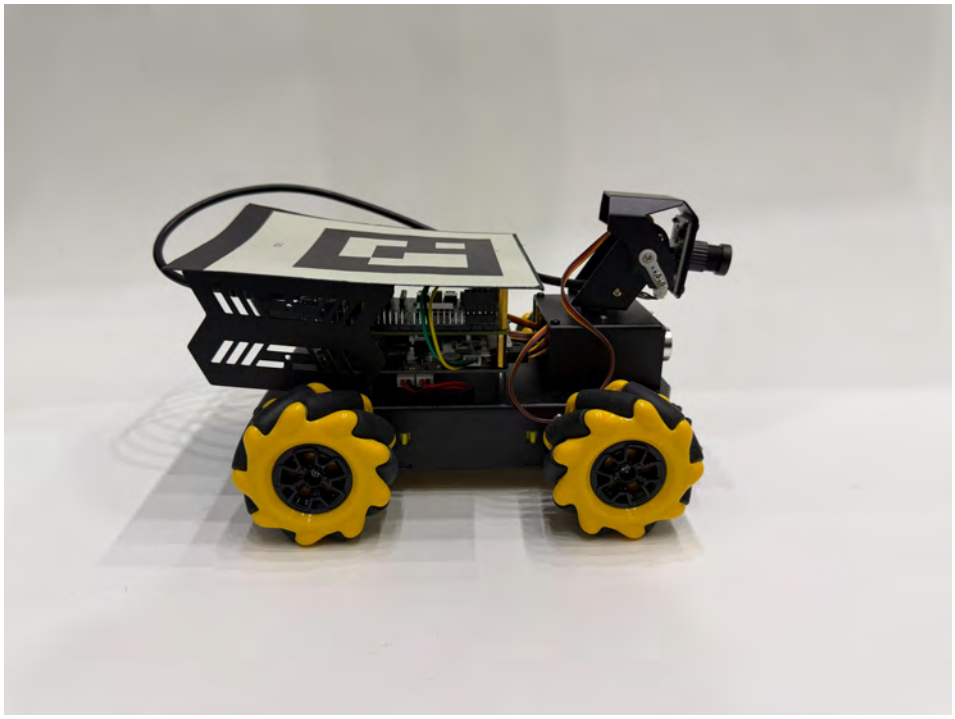

図 4.3　Raspbot v2

表 4.13　PC および使用した実機のハードウェア仕様

項目	仕様
PC OS	Windows 11 64bit
CPU	Intel Core i5-1135G7
RAM	16 GB
Web カメラ	EMEET C960 (640 × 480 px)
実機搭載コンピュータ	Raspberry Pi 5
RAM	8 GB
ストレージ	microSDXC 256 GB
Raspberry Pi OS	12

表 4.14　ソフトウェア環境

項目	PC	Raspberry Pi 5
Python	3.10.12	3.11.2
OpenCV	4.8.0	—
NumPy	1.24.3	—

図 4.4　実機における環境

（※文責：春原亮太）

4.4.2　移動機構と制御方式

実機の移動機構としてメカナムホイールを採用した．これは離散空間でのシミュレーションにおける車線変更という動作を実機においても簡潔に表現するためである．この移動機構では車体の向きを維持したまま横に平行移動することができる．一般的な移動機構であるステアリング方式では回転半径の計算が必要であり，差動駆動方式では隣接レーンへの移動に旋回・直進・逆旋回という 3 段階の動作を要する．そのため，これらの移動機構では制御および学習モデルとの対応が複雑化する．対して，メカナムホイールは車体の向きを維持したまま横に平行移動ができるため，シミュレーション上の行動出力を直接的に利用できる利点がある．その反面，メカナムホイールでは 4 輪すべてを独立に制御する必要があるため，制御の正確さが求められる．

実機の動作を制御するパラメータを表 4.15 に示す．各動作における速度設定値は，0 から 255 までの PWM 制御による出力値とした．PWM 制御とは電圧のオンとオフの時間を周期的に切り替えることで，平均電圧を操作しモータの回転数を調節する方式である．また，速度の制御は Yahboom 社が提供する Raspbot V2 専用ライブラリを用いた．パラメータに関して，平行移動速度についてはメカナムホイール特有の床面との摩擦による速度低下を補うため，直進速度よりも高い値を設定している．また，移動および旋回動作時間を短く設定することで，実機への命令送信ごとの位置誤差の修正を細かくし，実機の動作の正確性を確保した．

表 4.15　実機の動作制御パラメータ

設定項目	数値
直進速度	30
平行移動速度	40
旋回速度	45
移動動作時間	0.5 s
旋回動作時間	0.1 s

（※文責：岩渕波空）

4.4.3　許容誤差と時間制御

確実な実機のマス目間移動を実行するため，表 4.16 に示す許容誤差を導入した．これらの数値は，位置検出における数値の変動による過剰な動作を抑制する役割を持つ．

目標地点との距離許容誤差を 8.0 cm としたのに対し，レーン中央からの許容誤差を 10.0 cm とやや緩やかに設定した．これは，カメラの歪みにより，y 軸方向（車線幅方向）の座標検出誤差が大きくなりやすい傾向を考慮したものである．また，角度許容誤差は実機の現在の進行方向に対し，目標とする道路方向に対してどの程度の誤差を許容できるか設定している．今回の実験では，画像処理による角度検出が実際での傾きに対して，控えめに検出する傾向があった．具体的には，実機が実際には 10° 傾いていても，検出値は 5° 程度にとどまるという約 2 倍の誤差が生じていたからである．これは実機の上面に配置した ArUco マーカー（後述，以下マーカー）の大きさに対して，カメラの高さによる画像上の解像度が不足しており，マーカーの角度変化を正確に捉えきれなかったことが原因と考えられる．

また，実機への命令送信の間隔は実機が移動中であるときのカメラから取得した画像のブレや取得した画像の画像認識までの時間を考慮し，実機の物理的な停止状態と，システム上の認識座標を同期させるためである．

表 4.16　動作完了判定およびシステム制御パラメータ

カテゴリ	項目	数値
許容誤差	目標地点との距離許容誤差	8.0 cm
	レーン中央からの許容誤差	10.0 cm
	角度許容誤差	5.0 °
時間制御	実機への命令送信最小間隔	2.0 s

（※文責：岩渕波空）

4.4.4　実験環境の設定

離散空間における強化学習の状態として扱うため，実機における環境全体をシミュレーション環境と同一のマス目として定義した．物理的な連続座標をいずれかのマス目に対応させることで，連続値を離散状態へと変換した．実験環境の物理的な設定，および自己位置推定のためのカメラ設置

条件を表 4.17 に示す．道路の全長および幅は，使用した実機の全長と全幅を考慮し，2 台がゆずりあいする際に必要な空間を確保した．カメラの設置高さは約 165cm としたが，これは使用した三脚の最大伸長高度である．可能な限り高所から位置情報を取得することで，カメラによる周辺部の歪みを抑制し，環境全体における位置情報の取得精度の均一化を図った．

表 4.17　実験環境の物理パラメータ

項目	数値
実機の全長	24cm
実機の全幅	15cm
道路の全長	150.0 cm
道路の幅	30.0 cm
道路の分割数	5
ArUco マーカーのサイズ	10.0 cm
カメラ設置高さ（床面より垂直方向）	約 165 cm

（※文責：春原亮太）

4.4.5　位置情報の取得

位置情報の取得には，Web カメラと OpenCV ライブラリの ArUco マーカーを用いた．このマーカーはカメラから見た位置や姿勢を検出することに優れている．画像はカメラによる歪みを含むため，フィールドの四隅に基準として配置した 4 点のマーカーをもとに射影変換行列を算出した．この行列を用いることで，画像上のピクセル座標を歪みのない俯瞰視点の実世界座標系へと変換可能とした．位置取得から状態決定までの処理フローを以下に示す．

1. **マーカー検出**: 実機天面のマーカーを検出し，画像上の 4 隅のマーカーのそれぞれの中心から座標を，特定の枠線から進行方向を算出する．
2. **射影変換の適用**: 算出した座標を射影変換行列により物理座標へ変換する．
3. **検出失敗時の処理**: マーカーの検出に失敗した場合は，直前の座標値を保持する．
4. **離散化の実行**: 変換後の座標がどのマス目に属するかを判定し，学習モデルへ渡す現在の状態を決定する．

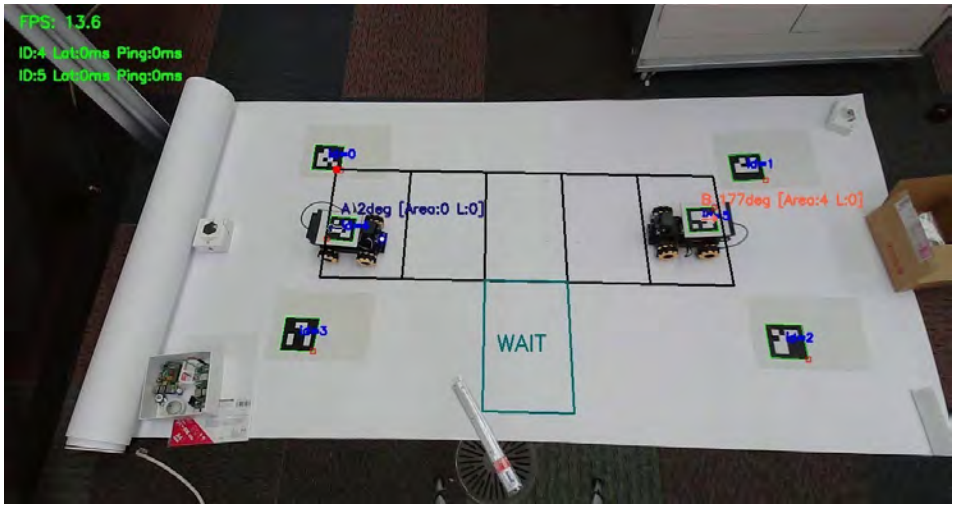

図 4.5　位置情報の検出の様子

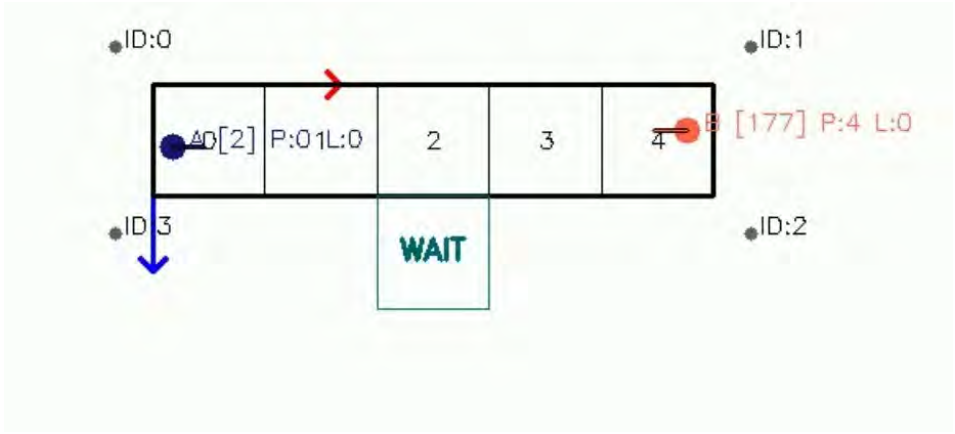

図 4.6　変換後の画像

（※文責：春原亮太）

4.4.6　実機の制御における安定化と同期手法

実機では，床面との摩擦や通信遅延，ハードウェアの個体差といった制約が実機の挙動に大きく影響する．以下に 4 つの課題と対策を整理した．

1. **モーター出力の個体差補正**
   - **課題**: 同一の指令値であっても，モーターの製造上の個体差により直進せず左右へ逸れてしまう．
   - **対策**: 各モーターに対し個別に補正係数を設定した．実験的に導出した係数を制御指令に乗算することで，ハードウェア由来の出力差を吸収し，正確な走行を実現した．
2. **通信バッファによる遅延対策**
   - **課題**: PC から実機への命令送信時，通信遅延により古い位置情報に基づいた命令が滞留・遅延実行される問題が発生した．
   - **対策**: 実機側での処理として最新の命令を優先とし，古い命令を破棄する仕様とした．これにより，常に最新の情報に基づく制御とした．

3. **実行時間と許容誤差による移動判定**
    - **課題**: 物理的な摩擦や慣性により，マス目の中心点へ完全に停止させることは困難である．
    - **対策**: 位置の許容誤差による完了判定と実行時間による制御を併用した．一定時間の経過に加え，座標と角度が許容範囲内（表 4.16）に収まった時点で動作完了とみなすことで，制御ループの待機を防いだ．
4. **ステップ数の同期による実機間同期**
    - **課題**: 2 台の実機のマス目間の移動速度差により，先行した実機が後続を引き離すなど，学習上の同一ステップが物理的にズレる．
    - **対策**: 各エージェントの移動回数を管理するカウンターを導入した．自身のステップ数が他方より先行している場合に待機を選択する仕組みを実装し，交代制の行動として扱うことで 2 台の実機の足並みを揃えた．

（※文責：岩渕波空）

第 5 章　結果

5.1　離散空間におけるシミュレーション

5.1.1　前期におけるシミュレーション

前期では不適切な報酬や環境の設定によりゆずりあうことができなかった．報酬では，待避スペースに毎ステップの正の報酬を与えたが，ゴール報酬に対して相対的に大きすぎたため，ゴールへ到達することよりも待避所に留まり続ける利益が上回る結果となった．また，2 つのエージェントが衝突した際の負の報酬を待避スペースの正の報酬よりも大きくした．その結果，待避スペースがあるにも関わらず車線変更を行わずにお互いが向かい合わせのまま行動しなくなることが発生した．これは，失敗による負の報酬を恐れるあまり，現状維持によって損失を最小化しようとする局所最適解に陥ったためと考えられる．

（※文責：春原亮太）

5.1.2　後期におけるシミュレーション

学習の結果，2 つのエージェントはゆずりあいを行うことができた．具体的には，2 台のエージェントが狭い道で対向した際，一方のエージェントが待避所に移動し，車線を変更して対向車の進路をゆずる挙動を見せた．本実験では，学習の進行を評価するために，各エピソードにおけるエージェントの累積報酬を記録した．強化学習における初期値や探索のランダム性の影響を排除し，結果の再現性と信頼性を担保するため，同一のパラメータ設定の下で独立した試行を実施した．図 5.1 は 500 エピソードのシミュレーションを 300 回繰り返したときの報酬の推移と報酬の分散を平均化したグラフにしたものである．このグラフから 50 エピソード程度から収束が見られる．

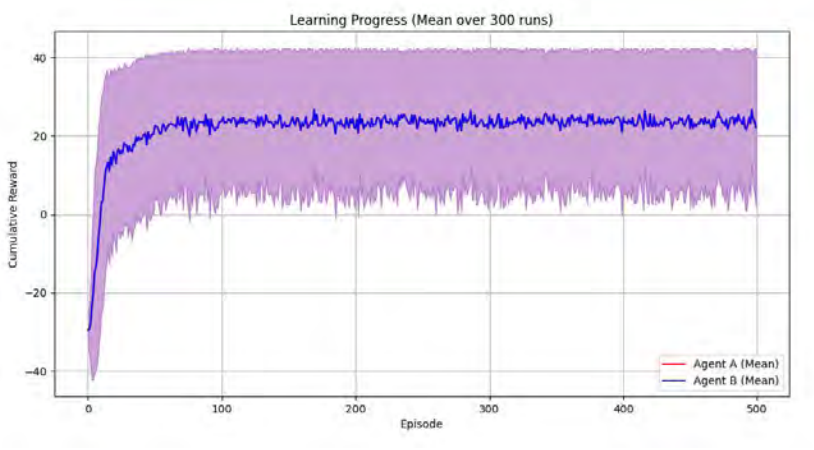

図 5.1　500 エピソードを 300 回繰り返したときの報酬の推移

（※文責：春原亮太）

5.2　連続空間におけるシミュレーション

初期の段階では，待避スペースのある中央へ進む前に壁にぶつかるといった行動が見られた．しかし，学習の経過を観察すると，直進という行動はできていた．これは道路外という壁に負の報酬，道路の中央を移動することに正の報酬を与えたため，早い段階から直進という行動を獲得できていたと考えられる．一方で，待避スペースで，片方のエージェントができる限り，ゴールへ進み，もう片方のエージェントがスタート地点に留まるという傾向が見られた．これは，エージェントが前へ進む行動に対し，報酬を与えたためだと考えられる．

（※文責：春原亮太）

5.3 実機

5.3.1 初期における失敗

初期段階において，通信システムや制御コードにはエラーがないにもかかわらず，実機が想定したゆずりあいの動作を行わない問題が発生した． 具体的には，対向車と対面した局面において，実機はその場で前進と後退を繰り返したり，本来移動すべきではない方向へスライド移動を行い，物理的なマップの外枠を越えて走行を続けたりする挙動が見られた．これらの問題をシミュレーション環境と実機の制御において修正した結果，その場で前進と後退を繰り返す，移動すべきではない方向へスライド移動するといった挙動は解消された．

（※文責：岩渕波空）

5.3.2 ステップ数の管理の有無による比較

実験は以下の 2 つに分けた．分けた理由として，シミュレーションと実環境におけるマス目間の移動速度のズレがあるためである．

ステップ数の管理がない場合

ステップ数の管理がない場合では，10 回試行した結果 7 回はゆずりあいに成功し，2 台の実機はゴールへたどり着くことができた．しかし，3 回は 1 台の実機が待避スペースに移動する際に，直進する 1 台と衝突した．そのため，物理環境ではゆずりあいに失敗した．

ステップ数の管理がある場合

ステップ数の管理がある場合では，10 回試行した結果，10 回ともゆずりあいに成功し，2 台の実機はゴールへたどり着くことができた．2 台の実機が対面した際，片方の実機が待避スペースへとスライド移動を行った．対向車が通過するまで待機し，その後安全を確認して元の車線へ復帰，最終的に双方が衝突することなくゴール地点へ到達することに成功した．

（※文責：春原亮太）

5.3.3 カメラの FPS の測定

ArUco マーカーなどの画像認識を含めたカメラの 1 秒間あたりのフレームレート（以下，FPS）を測定したところ，以下の表 5.1 となった．平均速度として，30FPS となっているため，画像認識の処理速度として問題のない範囲だったと考えられる．

表 5.1　フレームレートの計測結果

項目	数値
最高速度	85.14 fps
平均速度	30.0 fps
最低速度	1.54 fps

（※文責：岩渕波空）

第 6 章　考察

6.1　離散空間におけるシミュレーション

6.1.1　学習モデルの検証

離散空間シミュレーションにおいて，ゆずりあいを実現できた要因は，環境設定と報酬設計にある． 環境設定では，行動空間と状態空間が限られていたことがあげられる．また，行動をマスク処理をしたためにより行動が少なくなり，探索する空間が減少したためだと考えられる．報酬設計においては，衝突時の負の報酬をゴール到達時の報酬と同程度に設定した．一般に，負の報酬が大きすぎると，エージェントは衝突のリスクを恐れるあまりその場に停滞し続けるという消極的な局所解に陥りやすい．この罰則と報酬を同程度に保ったことで，ゆずりあいに成功したと考えられる．

（※文責：春原亮太）

6.1.2　今後の課題

今後の課題として，以下の 2 点が挙げられる．

エージェント数増加への対応

1 つ目は，エージェント数の増加に伴う学習の困難さである．離散空間で用いた Q 学習は，状態と行動の組み合わせを Q テーブルによって保持する．そのため，エージェント数や行動空間が増大するにつれて Q テーブルのサイズが指数関数的に増大し，学習が収束しなくなる．今後は，Qmix[10] などの深層強化学習への移行をする必要がある．

部分観測環境への対応

2 つ目は，部分観測環境への対応である．今回作成したシミュレーションでは，全エージェントが環境全体を把握できる完全観測を前提としているが，現実世界ではセンサーやカメラの視野制限により，観測可能な範囲は限定的である．例えば，曲がり角の先や障害物の背後にいる対向車を即座に認識することは困難であり，通信遅延やセンサー誤差による情報の不確実性も避けられない．

（※文責：春原亮太）

6.2　連続空間におけるシミュレーション

連続空間におけるシミュレーションでは，離散空間と比較して「ゆずりあい」の獲得が困難であった．その主な要因として，探索空間の広大さ，疎な報酬，基本動作の獲得の難しさの 3 点が挙げられる．

第一に，連続空間における探索空間の爆発的増大である．離散的なグリッド環境とは異なり，連続空間では状態と行動の組み合わせが無限に存在するため，有効な方策を見つけ出すための空間が

膨大になる．その結果，学習の収束には膨大な時間を要し，限られたリソース内では十分な試行錯誤が行えなかった．

第二に，疎な報酬の問題である．エージェントが目的地に到達した際にのみ報酬を与える疎な報酬の設定では，連続空間において偶然ゴールに到達する確率が極めて低い．そのため，学習において，行動の改善を掴むことが困難であった

そして第三に，ゆずりあいの前提となる「目的方向へ走行する」という基本的な移動能力の獲得自体が，連続空間では高度な課題となった．これらを踏まえ，学習の構造的な難しさを以下に整理する．

学習の難しさ

ゆずりあいの学習は，単一のタスクではなく，性質の異なる複数の行動を同時に，あるいは段階的に学習する必要がある．ゆずりあいを学習するまでにエージェントが獲得するべき行動は以下の 3 つである．この複雑さが学習までの障壁となっていると考えられる．

1. **基本的な移動動作**
   意図した方向への移動，目標地点への正確な到達，および適切な停止といった基本的な移動能力である．
2. **衝突回避**
   移動経路上に他者が存在する場合に，物理的な接触を避ける能力である．
3. **協調行動**
   単に衝突を避けるだけでなく，互いの目的地への到達効率を最大化するために，一方が進路を譲る，あるいは待機するといった高度な判断を行う能力である．

6.2.1 今後の課題

今後の課題として優先度の高い以下の 2 つを挙げる．

パラメータの最適化

パラメータは，全行程を通して一定の設定で実行した．しかし，学習の難しさで述べた通り，エージェントが段階的に行動を獲得する場合，各フェーズで求められるパラメータの性質は異なる．例えば，第 1 段階の基本的な移動を学習する際は，広範囲を探索するために学習率や探索率を高く設定する必要がある．一方で，第 3 段階の協調行動では，既に獲得した移動能力を損なわないよう，より小さな学習率で精密にポリシーを更新しなければならない．このように，学習の進捗に合わせて動的にパラメータを調整する仕組みの導入が，効率的な学習を実現するための課題となる．

段階的な能力の獲得

まずは基本的な移動操作のみを独立して学習させ，安定した移動能力を獲得したモデルをベースとして，ゆずりあいの動作を追加で学習させる段階的なアプローチが有効であると考えられる．このように，タスクの難易度を段階的に引き上げていくことで，ゆずりあいを効率的に獲得できる可能性がある．

（※文責：春原亮太）

6.3 実機

6.3.1 初期実験における失敗

ここでは初期実験における失敗の原因として以下の 2 つを挙げる．

前後の反復移動：評価対象の誤り

実機において，目標地点付近で前後の反復移動を繰り返す挙動が確認された．この原因は，学習が十分に収束し安定したモデルではなく．学習プロセスの直近の状態である最終エピソードの挙動のみを確認し，成功と誤認していたことにある．その結果，実機では適切ではない学習モデルで動作することとなった．

マップ外への逸脱：環境制約の不足

ロボットがフィールド外へ移動しようとする挙動は，シミュレーション環境における制約が不十分であったことにある．具体的には，フィールドを越える行動に対して負の報酬が設定されていない．もしくは，フィールドを超える行動の選択肢を排除していなかった．この不適切な結果に気づけなかった原因として，学習した結果の Q 値の生データを確認していなかったことにある．そのため，間違ったシミュレーションで学習した結果を再生し，フィールド外移動を有効な選択肢として学習している事実に気づくことができなかった

6.3.2 学習モデルと実機の統合

初期実験における失敗から，強化学習における報酬設計の重要性に気づかされるとともに，シミュレーションの結果を鵜呑みにせず，Q 値の分布や行動ログを詳細に検証することの重要性を再認識した．実機は，単にプログラムを動かす場ではなく，シミュレーションモデルの欠陥をあぶり出す検証としての役割も果たしたと言える．

（※文責：岩渕波空）

6.3.3 ステップ数の管理の有無による比較

シミュレーション環境では，2 台のエージェントは常に完全に同期して意思決定と行動を行う．しかし実機においては，通信遅延やモーターの個体差により，1 マスを移動するのに要する物理的な時間にズレが生じる．ステップ数の管理がない場合，この累積したズレが原因で一方が退避スペースに入る前に，もう一方が通過を試みるというタイミングの不一致が発生し，衝突に至ったと考えられる．

（※文責：春原亮太）

6.3.4 今後の課題

実機制御においては，通信ラグや位置判定の誤動作を防ぐため，1 ステップごとに確実に同期をとる方法を導入した．このアプローチによりシステムの動作安定性は確保されたものの，主に 2 つ

の課題がある．

分散制御の導入

実機の運用において，現在は外部 PC を介して強化学習モデルから次の目的地の座標などの命令を受信し，それに基づいて各機体が動作する構成をとっている．しかし，本来は各ロボットが自身のセンサーから得られる情報に基づき，独立して意思決定を行うことが望ましい．この課題は，MAPPO が採用している集中学習・分散実行の枠組みを導入することで解決が可能である．これは，学習時には全エージェントの情報を集約して利用し，実行時には各エージェントが自身の局所的な観測のみに基づいて独立して意思決定を行う枠組みである．

非同期制御の検討

同期制御では，2 台の実機が同時に動作するため，目標地点までの総移動時間は非同期制御と比較して短縮される．しかし，物理環境においては以下のような不安定さが存在する．

- **衝突リスクの増大：** 互いに向かい合う，ゆずりあいの局面において，同時に移動を開始すると，通信ラグや位置精度の誤差が衝突に直結する．
- **累積誤差の影響：** 双方のステップがズレた際，シミュレーションでは想定されない状態での接触が発生しやすい．

そのため，実機において制御する場合，同期制御よりも非同期制御のほうが安全性が高いと考えられる．

（※文責：春原亮太）

6.4 Sim2Real に向けた制御境界の設計

実機移行の実現において，強化学習が担うべき制御と，実機側の制御が担うべき境界をどこに設定するかは課題である．

6.4.1 離散空間における知見と実装上の課題

離散空間における学習では，制御対象をマス目上の位置という情報のみに限定した．そのため，学習済みモデルを実機に適用する段階においては，位置推定の精度確保や，床面との摩擦，モーターの個体差，および 2 台のロボット間の個体別の速度差といった物理的な不確実性を，すべて実機側の制御において考慮する必要があった．これらの要素に対する試行錯誤的な努力によって最終的な動作は実現されたものの，実装には多大な時間を要し，課題が残る結果となった．

6.4.2 連続空間における現状とリアリティギャップ

現在取り組んでいる連続空間におけるシミュレーションでは，位置に加えて速度を制御対象に含めることで，より実機に近い動的な挙動の獲得を目指している．しかしながら，加速度や摩擦，さらには実機の物理的なサイズといった要素は簡略化されており，シミュレータと現実世界との間には依然として無視できないリアリティギャップが存在する．

6.4.3 実機への適用を最大化するためのアプローチ

この乖離を埋めるためには，シミュレーション内で摩擦などの物理パラメータをランダムに変化させるドメインランダマイゼーションの導入が不可欠である．今後は，これらの物理的制約をどの程度シミュレーションに組み込むべきか検討し，実機への適用を最大化するための制御境界の最適化を図る必要がある．

（※文責：春原亮太）

参考文献

[1] 織田智矢, 横山想一郎, 山下倫央, 蕨野貴之, 大岸智彦, 田中英明,「RC カーを用いた自動運転車両シミュレーション環境の構築」, 情報処理学会研究報告, Vol.2020-ICS-198, No.13, pp.1-6, 2020.

[2] 脳をつくるプロジェクト「World Model Car」, 2022 年度 公立はこだて未来大学システム情報科学実習グループ報告書, https://www.fun.ac.jp/wp-content/uploads/222023/04/document_A.pdf, 2022（アクセス日: 2026 年 1 月 21 日）.

[3] 脳をつくるプロジェクト「画像認識だけで AI カー」, 2023 年度 公立はこだて未来大学システム情報科学実習グループ報告書, https://www.fun.ac.jp/wp-content/uploads/2024/03/document13_B.pdf, 2023（アクセス日: 2026 年 1 月 21 日）.

[4] Make Brain Project「視覚を持つ AI カー」, 2024 年度 公立はこだて未来大学システム情報科学実習グループ報告書, https://www.fun.ac.jp/wp-content/uploads/2025/05/group21C.pdf, 2024（アクセス日: 2026 年 1 月 21 日）.

[5] Watkins, C. J. C. H., and Dayan, P., ”Q-learning,” Machine Learning, vol. 8, pp. 279–292, 1992.

[6] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, Yi Wu, ”The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games,” Advances in Neural Information Processing Systems, vol. 35, pp. 24611–24624, 2022.

[7] Stable-Baselines3, https://stable-baselines3.readthedocs.io/en/master/(アクセス日: 2026 年 1 月 21 日).

[8] Raspbot V2,https://www.yahboom.net/study/RASPBOT-V2 (アクセス日: 2026 年 1 月 21 日).

[9] Pygame Community，https://www.pygame.org/ （アクセス日: 2026 年 1 月 21 日).

[10] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson, ”Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning,” Journal of Machine Learning Research, vol. 21, no. 178, pp. 1–51, 2020.