卒 業 論 文

屋外自律高速モビリティを対象とした経路追従ソフトウェアの開発

Development of path-following software for outdoor autonomous high-speed mobility

2025 年 1 月 16 日 提出

指導教員　林原 靖男 教授

千葉工業大学 先進工学部 未来ロボティクス学科

21C1135 山下響

概要

屋外自律高速モビリティを対象とした
経路追従ソフトウェアの開発

本論文では屋外自律高速モビリティを対象とした経路追従ソフトウェアを開発する．また, 開発した経路追従ソフトウェアを実環境で検証を行った結果について述べる．屋外自律高速モビリティを使用してスピードと知能を競うリアルワールドレースを行う AI-Formula という技術チャレンジがある．AI-Formula では, ベースとなるハードウェアが貸与される．技術チャレンジという性質上, ルールが設けられていてハードウェアは支給されたモータやタイヤを使用するなどの制約がある．そのため, 開発要素としてはソフトウェア開発が主となる．AI-Formula ではベースソフトウェアが用意されているが, これらはモビリティを制御するための基盤となるソフトウェア構成となっている．提供されるベースソフトウェアではモビリティを制御するための最低限の機能に留まっているため, 経路追従などのコースを周回するソフトウェアは各チームで開発することが必要となる．そのため, 本論文では AI-Formula で使用されるモビリティを対象とした経路追従ソフトウェアを開発する．また, 作成した経路追従ソフトウェアを実環境で検証する．結果として, 用意した目標経路に対して追従する様子を確認した．

キーワード: AI-Formula, 経路追従, Pure pursuit

abstract

Development of path-following software for outdoor autonomous high-speed mobility

In this paper, I develop a path-following software for outdoor autonomous high-speed mobility. I also present the results of the verification of the developed path-following software in a real environment. There is a technical challenge called AI-Formula, which is a real-world race for speed and intelligence using outdoor autonomous high-speed mobility vehicles. One of the objectives of AI-Formula is to provide opportunities to acquire many skills necessary for next-generation mobility and to foster engineers who will develop next-generation mobility systems. In AI-Formula, the base hardware is loaned to the participants. Due to the nature of the technical challenge, there are some restrictions such as the use of supplied motors and tires for the hardware. Therefore, the main development element is software development. AI-Formula provides a base software, which is the software structure that serves as the basis for controlling mobility. Since the provided base software has only the minimum functions to control mobility, each team needs to develop its own software to follow a path around the course. Therefore, in this paper, we develop a path-following software for mobility used in AI-Formula. In addition, the developed path-following package is verified in a real environment. As a result, we confirmed that the software follows the prepared target paths.

keywords: AI-Formula, path-following, Pure pursuit

目次

第 1 章　序論　1

1.1　背景 1

1.1.1　ハードウェア 1

1.1.2　コース 3

1.2　目的 3

1.3　論文の構成 3

第 2 章　要素技術　4

2.1　ROS 2 4

2.1.1　RViz2 4

2.2　GNSS 5

2.2.1　UTM 座標系 5

2.2.2　ECEF 座標系 6

2.2.3　測地座標系 6

2.2.4　マルチパス 7

2.3　IMU 7

2.4　PID 制御 7

2.4.1　P 制御 8

2.4.2　I 制御 8

2.4.3　D 制御 8

第 3 章　経路追従のシステム　9

3.1 ハードウェア 9
3.1.1 センサ構成 10
3.1.2 従動輪 10
3.1.3 ブレーキ機構 11
3.2 システム構成 11
3.3 目標経路の作成 13

第 4 章 経路追従に用いるアルゴリズム 14
4.1 経路追従 14
4.1.1 PurePursuit 14
4.2 経路生成 15
4.2.1 3 次スプライン補間 15

第 5 章 シミュレータ環境での実験 17
5.1 シミュレータ 17
5.2 実験の概要 20
5.3 実験環境 20
5.4 実験内容 21
5.5 実験結果 22

第 6 章 実環境での実験 24
6.1 実験の概要 24
6.2 実験装置 25
6.3 実験内容 27
6.4 実験結果と考察 29
6.4.1 実験 1 29
6.4.2 実験 2 31

第 7 章 結論 34

参考文献 35

付録 36

謝辞 37

図目次

1.1 Exterior view of the mobility platform 2
1.2 Mobile Power Pack 2
1.3 AI-Formula Course 3

2.1 RViz2 view 5
2.2 UTM Coordinate System（source : [7] 6
2.3 Multipath 7

3.1 Robot appearance 9
3.2 VectorNav VN-200 (source : [8]) 10
3.3 Mechanism for driving the caster orientation 11
3.4 Path following system 12
3.5 Predefined target trajectory 13

4.1 PurePursuit Algorithm 15
4.2 Spline path 16

5.1 Appearance of simulator 18
5.2 AI-Formula and racing course 18
5.3 Configuration of ROS 2 nodes 19
5.4 Simulator world top view 20
5.5 Simulator course 22
5.6 The AI-Formula path-following behavior on the simulator 23

6.1 Aerial view and mobility perspective of the course 25
6.2 Path-following system 26
6.3 Aerial view and mobility perspective of the course 27
6.4 Experiment path 28
6.5 Trajectory of travel route and target route 30
6.6 Transition of error from the target path 31
6.7 Trajectory of travel route and target route 32
6.8 Transition of error from the target path 33

表目次

5.1 Sensors and actuators to be simulated 19
5.2 experimental device 21

6.1 evaluation item 24
6.2 experimental device 26
6.3 experiment1 parameters 29
6.4 evaluation item 31
6.5 experiment1 parameters 32
6.6 evaluation item 33

第 1 章

序論

1.1 背景

AI-Formula は計測自動制御学会と自動車技術会が主催し, 本田技術研究所がサポートする自律移動モビリティの競技会で, 正式な競技会は 2025 年から始まる. [?] AI-Formula はスピードと知能を競うリアルワールドレースであり, 次世代モビリティに必要な技術を身につける機会を提供し, 自動化システムのエンジニアを育成することを一つの目的としている. 実環境でモビリティを自律走行させてレースを行うため, ハードウェアとソフトウェアの両方が要求される. しかし, AI-Formula ではハードウェアは貸与の形式で与えらているため, 現時点での開発はソフトウェアの開発が主となる. ハードウェアは経路追従するために必要なパーツが全て揃っているが, ソフトウェアはモビリティを制御するためのシステムの基盤となるベースソフトウェアが用意されているのみである. [?] そのため, 経路追従などのソフトウェアは各チームで開発することが必要となる. [1] [2]

1.1.1 ハードウェア

AI-Formula では詳細なルールは検討中であるとのことで, 2025 年 3 月に開催される予定のプレ大会以降に詳細なルールが決定される予定となっている. Fig.1.1 に 2025 年のプレ大会で使用される予定のモビリティを示す. AI-Formula では, モビリティは本田技術研究所から貸与される. 貸与されたモビリティは差動二輪と一輪のキャスタ（以下, 従動輪と呼ぶ）で構成された三輪モデルである. 貸与されたモビリティには, ステレオカメラ (ZED-X stereolabs) や

GNSS+IMU センサ (VN-200 VectorNav) が搭載されている．バッテリーには MPP（Mobile Power Pack）が使用される．MPP は持ち運び可能な交換式バッテリーとなっている．Fig.1.2 に MPP を示す．プレ大会では指定されたコースを 3 周する時間で競う予定となっている．

Fig. 1.1: Exterior view of the mobility platform

Fig. 1.2: Mobile Power Pack

1.1.2　コース

AI-Formula のコンテストは, 茨城県常総市に位置する AI モビリティパーク紫峰において実施される. AI モビリティパーク紫峰はアスファルト舗装された路面や車線などの路面情報が整備されているコースである. Fig.1.3 に AI モビリティパーク紫峰を示す.

Fig. 1.3: AI-Formula Course

1.2　目的

本研究では, 屋外自律高速モビリティを対象とした経路追従ソフトウェアを開発して, 実環境で検証することを目的とする.

1.3　論文の構成

本論文は以下のように構成される.

まず, 2 章で本研究で使用される要素技術について述べる.

3 章では開発した経路追従ソフトウェアに要求されるシステムについてまとめる.

4 章では経路追従する際に使用するアルゴリズムについて述べる.

5 章ではシミュレータ環境で経路追従の実験を行い, アルゴリズムの有効性を確認する.

6 章では実環境で開発した経路追従ソフトウェアの実験を行う.

7 章では本研究の結論をまとめる.

第 2 章

要素技術

本章では, 本研究に関連する要素技術を述べる.

2.1 ROS 2

ROS 2（Robot Operating System versition 2）[3] は, オープンソースのロボットソフトウェアフレームワークであり, ロボットアプリケーションの開発や実行をサポートするミドルウェアである. 異なるバージョンが存在しているが, 本研究では ROS 2 foxy を主に使用している.

2.1.1 RViz2

RViz2（ROS VIsualization 2）[4] は ROS 2 で提供される三次元ビジュアライゼーションツールであり, 数値で表されるロボットの座標や各センサのデータを視覚的に表すことができる.

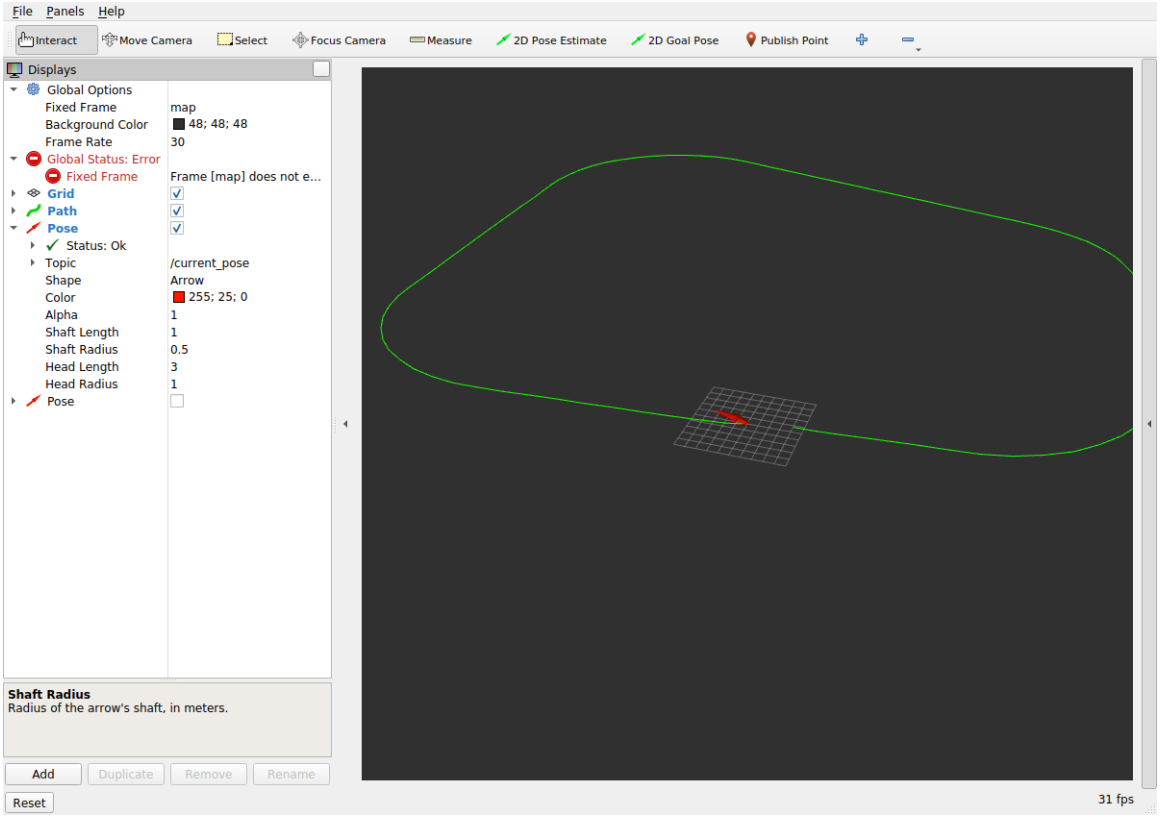


Fig. 2.1: RViz2 view

2.2　GNSS

GNSS（Global Navigation Satelite System）は, 人工衛星を利用して地上の現在位置を計測するためのシステムであり, アメリカの GPS, ロシアの GLONASS, ヨーロッパの Galileo, 日本の QGSS などを総称した衛生測位システムを指す.

計測する値は衛生と受信機（アンテナ）間の距離であり, 衛生位置を既知として受信機の三次元座標と受信機時計の誤差を未知数として最低 4 つの観測値から座標計算される. 測定された距離にはさまざまな誤差要因が含まれる.

2.2.1　UTM 座標系

UTM（Universal Transverse Mercator）座標系 [5] とは, 全世界を経度 6 度ごとのゾーンに分けて東回りに番号を付けて規格化したものである. 世界的にも大・中縮尺の図法として採用され, 日本では国土地理院の地形図や地勢図で採用されている.

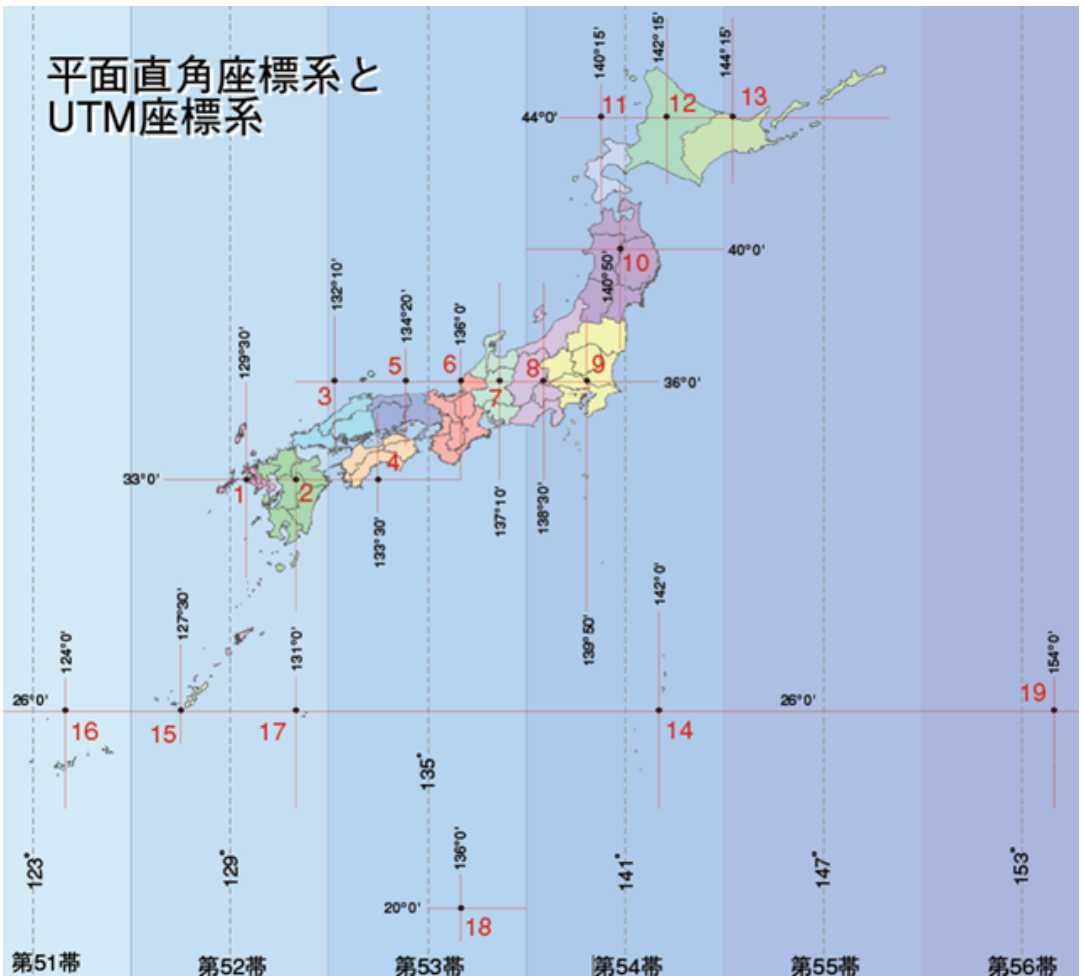


Fig. 2.2: UTM Coordinate System（source : [7]

2.2.2　ECEF 座標系

ECEF（Earth-Centered, Earth-Fixed）座標系とは, 地理的・直交的な座標系であり, 地球の自転と同期して常時回転している座標系である.

2.2.3　測地座標系

測地座標系とは, 地球上の位置を緯度, 経度, および回転楕円体からの高さで表す座標系である.

2.2.4　マルチパス

マルチパスとは, 電波がまっすぐに届くだけでなくビルなどの高層建造物に反射して複数のルートを通って伝播することである. 反射した電波は到達するまでにわずかな遅れを生じ, 遅れの時間の分だけ距離が遠いと計測されてしまうため, 正確な測位を乱す要因の一つとなっている.

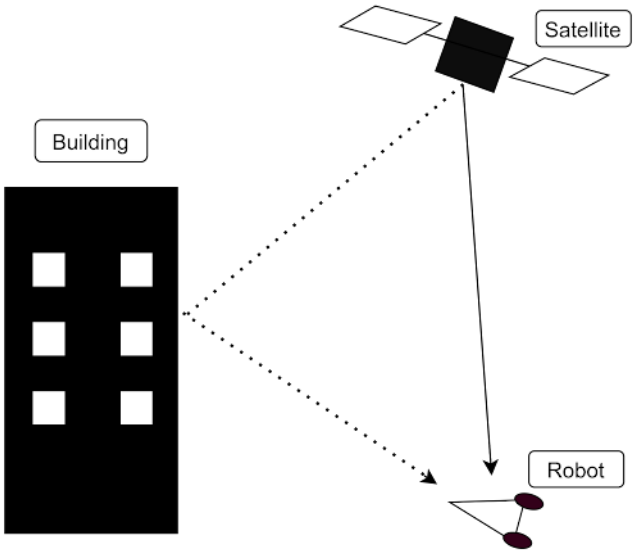


Fig. 2.3: Multipath

2.3　IMU

IMU（Inertial Measurement Unit）は, 3 次元の慣性運動を検出する装置である. 加速度センサーにより並進運動を検出し, ジャイロセンサーにより回転運動を検出することができる.

2.4　PID 制御

PID 制御（Proportional-Integral-Differential Controller）は, 制御工学におけるフィードバック制御の一種である. 出力値と目標値との偏差, その積分, および微分の 3 つの要素によって, 入力値の制御を行う方法である.

2.4.1　P 制御

基本的なフィードバック制御として比例制御（P 制御）がある．これは操作量を制御量と目標値の偏差の一次関数として制御するものである．PID 制御では, この偏差に比例して操作量を変化させる動作を比例動作, または P 動作といい, その時の定数は比例ゲイン, P ゲインと呼ばれる．

2.4.2　I 制御

P 制御において, 周囲の環境が変わるたびに残留偏差をなくすように比例ゲインを決定しなおすことが難しい．そこで, 残留偏差が存在する場合, その偏差の時間積分に比例して入力値を変化させる動作をする項を追加する．追加した項が持つ役割が I 制御である．偏差のある状態が長い時間続けばそれだけ入力値の変化を大きくして目標値に近づけようとする役目を果たす．ここで, 定数は積分ゲイン, I ゲインと呼ばれる．

2.4.3　D 制御

PI 制御の問題点として, 周囲の環境が変化したり制御対象に撹乱が加わったりすることで出力値が急に変動することがある．この問題を解決するために, 急激な出力値の変化が起こった場合, その変化の大きさに比例した入力を行うことでその変化に抵抗する役割を持つ項を追加する．追加した項が持つ役割が D 制御である．定数は微分ゲイン, D ゲインと呼ばれる．

第 3 章

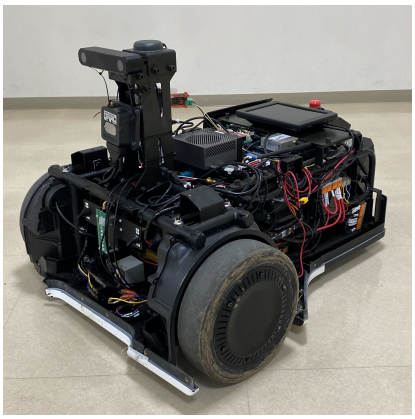
経路追従のシステム

ここでは, 本論文のベースとなるシステムについて述べる.

3.1 ハードウェア

本研究は AI-Formula で貸与された屋外自律高速モビリティを使用する. モビリティは差動二輪と従動輪で構成された三輪モデルであり, 駆動モータの動作に追従するような従動輪機構となっている. Fig.4.1 に外装を取り付けたモビリティと外装を取り外したモビリティのハードウェアの外観を示す.

(a) Robot with exterior



(b) Robot without exterior

Fig. 3.1: Robot appearance

3.1.1　センサ構成

モビリティに搭載されているセンサ構成について述べる．貸与されたモビリティにはステレオカメラ, GNSS+IMU, ホイールエンコーダなどのセンサが搭載されている．開発するソフトウェアでは GNSS+IMU センサ (VN-200 VectorNav) を使用する．

VN-200

VN-200[6] は GNSS と IMU を内包したセンサーであり, 3 軸ジャイロ, 加速度計, 磁力系, GNSS 受信機を用いて位置, 速度, 姿勢を推定することができる小型で高性能な GNSS 支援慣性航法システムである．IMU を内包しているため, 位置だけでなく方位も取得できるセンサとなっている．VectorNav の方位は 2 種類の情報を手がかりに補正されるようになっており, 起動時は地磁気を手がかりに, 移動後は GNSS の履歴を手がかりに補正される．ROS 2 のラッパーとして, vectornav というパッケージが公開されている.[7] Fig.4.2 に VN-200 の外観を示す.

Fig. 3.2: VectorNav VN-200 (source : [8])

3.1.2　従動輪

貸与されたモビリティには従動輪の向きが望む方向に変化しない問題が発生していたため, 従動輪の向きをモータで駆動することで対処した．これを実装したモビリティは結果的に, 高い操縦性により望む方向に移動できるようになった．Fig.4.3 に機構を示す.

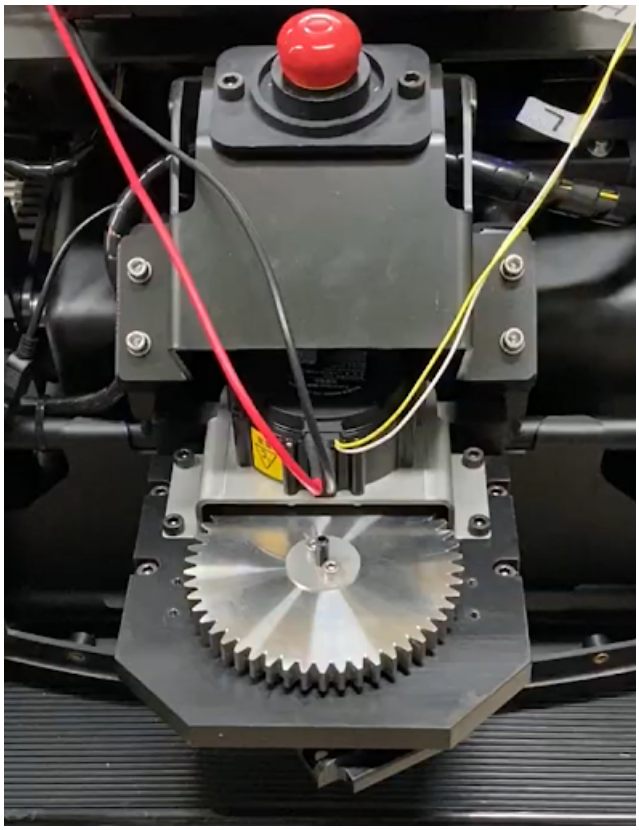


Fig. 3.3: Mechanism for driving the caster orientation

3.1.3　ブレーキ機構

ブレーキ機構にはディスクブレーキと回生ブレーキが使用されている．ディスクブレーキ機構はステッピングモータでディスクブレーキのワイヤを引っ張ることでブレーキがかかる機構となっている．

3.2　システム構成

ここで, 経路追従に用いることができるセンサはステレオカメラに加えて GNSS がある．2 つのセンサを比べたとき, 扱いの用意さという点では GNSS に軍配があがる．ただし, マルチパスなどによる経緯度の誤差は数 10m におよぶことがあるため, 必ずしも経路追従に利用できるとは限らない．幸いにして, AIFormula の会場である AI モビリティパークの周囲には高い建物が無く, マルチパスの影響が少ないと考えられた．そのため, GNSS に基づいた経路追

従を行うシステムを開発した．Fig.4.4 に経路追従で使用するシステム構成を示す．

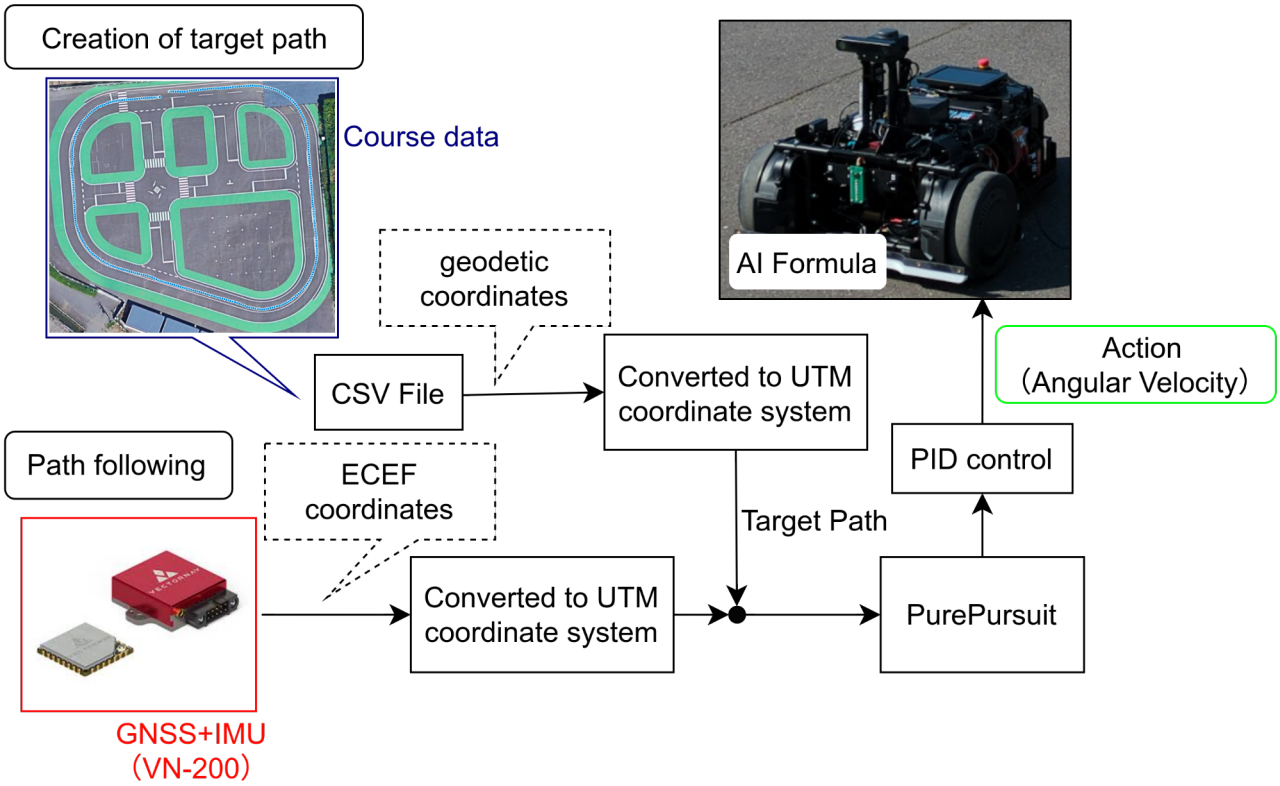


Fig. 3.4: Path following system

GNSS を用いて経路追従を行うためには, 位置に加えて方向のデータも必要となる．モビリティに搭載されている GNSS+IMU（VectorNav VN-200）は方位も計測できる仕様であるため, それを利用することが開発の多重化を防ぐ意味で望ましいと判断した．GNSS+IMU（VectorNav VN-200）は vectornav[7] というパッケージが公開されているため, そちらを使用して Vectornav のセンサデータを ROS 2 topic として扱っている．作成した経路追従のソフトウェアは, 事前にシュミレータ環境や手動走行をさせた時の GNSS から得た座標データを CSV ファイルに経路情報としてまとめている．用意した経路情報は測地座標系でまとめられていて, プロセスを起動したときに UTM 座標系に変換されて目標経路の情報を含んだ ROS 2 topic として publish される．経路追従を行うに際して, 自己位置は GNSS から取得した座標データをそのまま自己位置として扱っている．取得した座標系は ECEF 座標系で取得されるため, 目標経路の座標系と合わせるために UTM 座標系に変換する．UTM 座標系に合わされた目標経路の座標と自己位置の座標を利用して PurePursuit による経路追従を行う．

PurePursuit によって目標地点との角度差を取得し, その角度差に対して PID 制御を行うことでモビリティの角速度を決定する. 並進速度は一定として, 求めた角速度に基づいてモビリティを制御する.

3.3　目標経路の作成

経路追従をする際には追従するための目標経路が必要となる. 本研究では, 以下の図に示すように予め取得した経緯度のウェイポイントをつなぐライン状を移動する目標位置を PurePursuit で求め, 求めた目標地点に向かって進む制御を採用した. 経路情報はシミュレータ環境や事前に手動走行させたときの GNSS から得た情報を参照している. Fig.4.5 に作成した目標経路の例を示す.

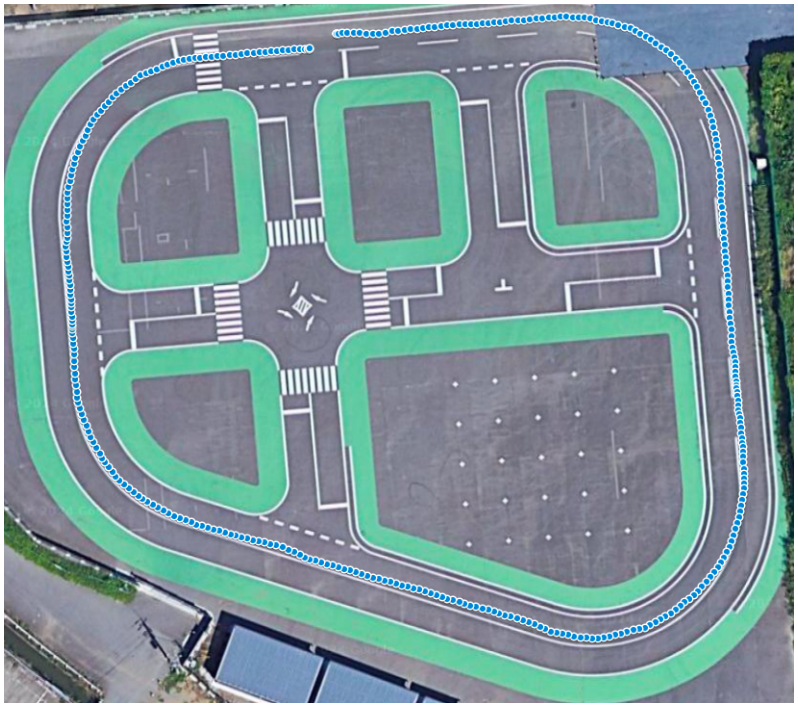


Fig. 3.5: Predefined target trajectory

第 4 章

経路追従に用いるアルゴリズム

本章では, 経路追従に用いるアルゴリズムについて述べる.

4.1 経路追従

経路追従（Path following）とは, 経路計画（Path planning）で引いた経路に対してモビリティが経路を追従できるようにモビリティを制御することである. 基本的にはモビリティのモデルと制御アルゴリズムを利用することで, 最終的にモビリティの入力値（ステアリング角度や並進速度）を計算することが目的となる.

4.1.1 PurePursuit

PurePursuit アルゴリズムは, 経路追従アルゴリズムの中で最も基礎的だが, 非常に広く使われているアルゴリズムである.

PurePursuit は Fig.3.1 に示すように目標経路上（Path）に対して一定距離先の点を目標点（Look Ahead）とし, その点に到達するようなステアリング制御を行う. 目標点に対してモビリティが前進することで, より先の目標点にたどり着くように制御を行うため, 結果的に目標経路に追従する形となる.

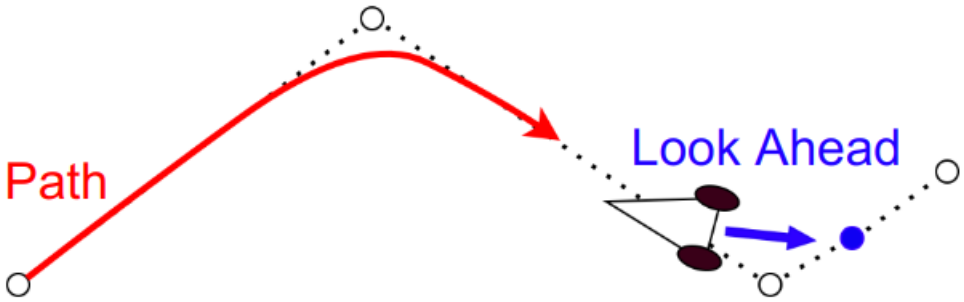


Fig. 4.1: PurePursuit Algorithm

4.2　経路生成

経路追従を行うためには, 追従するための目標経路（Path）を設定する必要がある. 自律移動モビリティの開発に置いて目標経路の探索を経路計画と呼び, 重要な問題の一つとされている. 本研究では, 目標経路は予め取得した測地座標系の経緯度で用意されたウェイポイントを結ぶ線形経路として設定している.

4.2.1　3 次スプライン補間

本研究で取得する目標経路は, ウェイポイントを結ぶ線形経路として設定している. 設定したウェイポイントが疎らである場合に, 離散的で不連続なウェイポイントから, 滑らかで連続的な経路を生成するために 3 次スプライン補間が使用される.

3 次スプライン補間とは, 複数のサンプル点が与えられた時に, そのサンプル点の間を 3 次の多項式で近似し, 滑らかに保管する手法である. Fig.3.2 にその様子を示す.

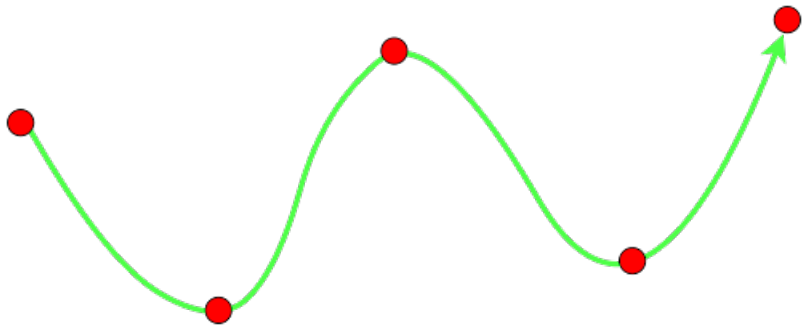


Fig. 4.2: Spline path

第 5 章

シミュレータ環境での実験

本章では, 開発した経路追従ソフトウェアをシミュレータ環境で使用することで, ソフトウェアの初期検証を行う.

5.1 シミュレータ

AI-Formula のような人の歩行速度以上で移動するロボットは検証する場所を確保することが容易ではないため, アルゴリズムの初期検証などはシミュレータ環境で検証・改善を行うことが望ましい. AI-Formula では基本的な機能が実装されたシミュレータ環境が運営から提供されている. 提供されたシミュレータ環境はモビリティと AI モビリティパーク紫峰の外周コースをシミュレートしたものになっている. Fig.5.1 に提供されたシミュレータ環境の外観を示す. Fig.5.2 にシミュレートするモビリティとコースの外観を示す.

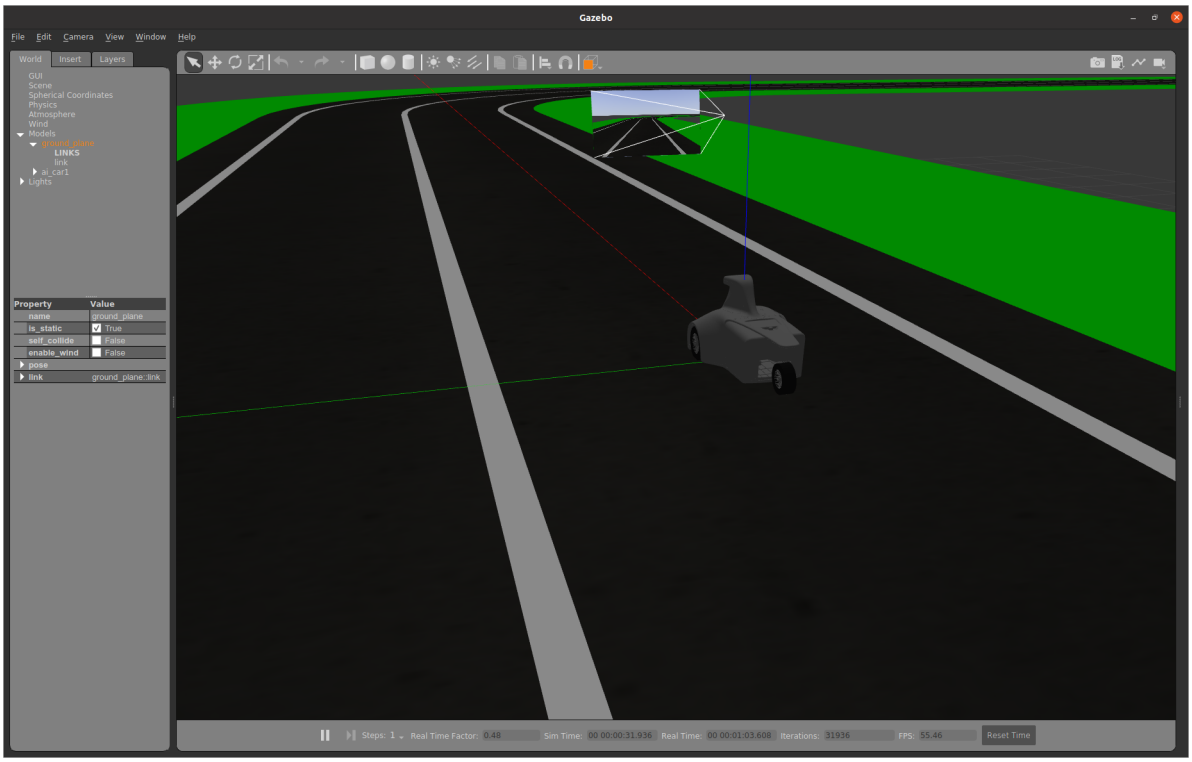


Fig. 5.1: Appearance of simulator

Fig. 5.2: AI-Formula and racing course

提供されたシミュレータ環境では基本的な機能が実装されているが, GNSS+IMU センサ（VectorNav VN-200）の出力を再現できていない. そのため, GNSS+IMU センサ（VectorNav VN-200）の出力を再現する機能を追加して, 提供されたシミュレータを拡張して使用している.[8]

シミュレータは ROS 2 と Gazebo 上で実装されており, Fig.5.3 に示すようなノードの構成になっている.

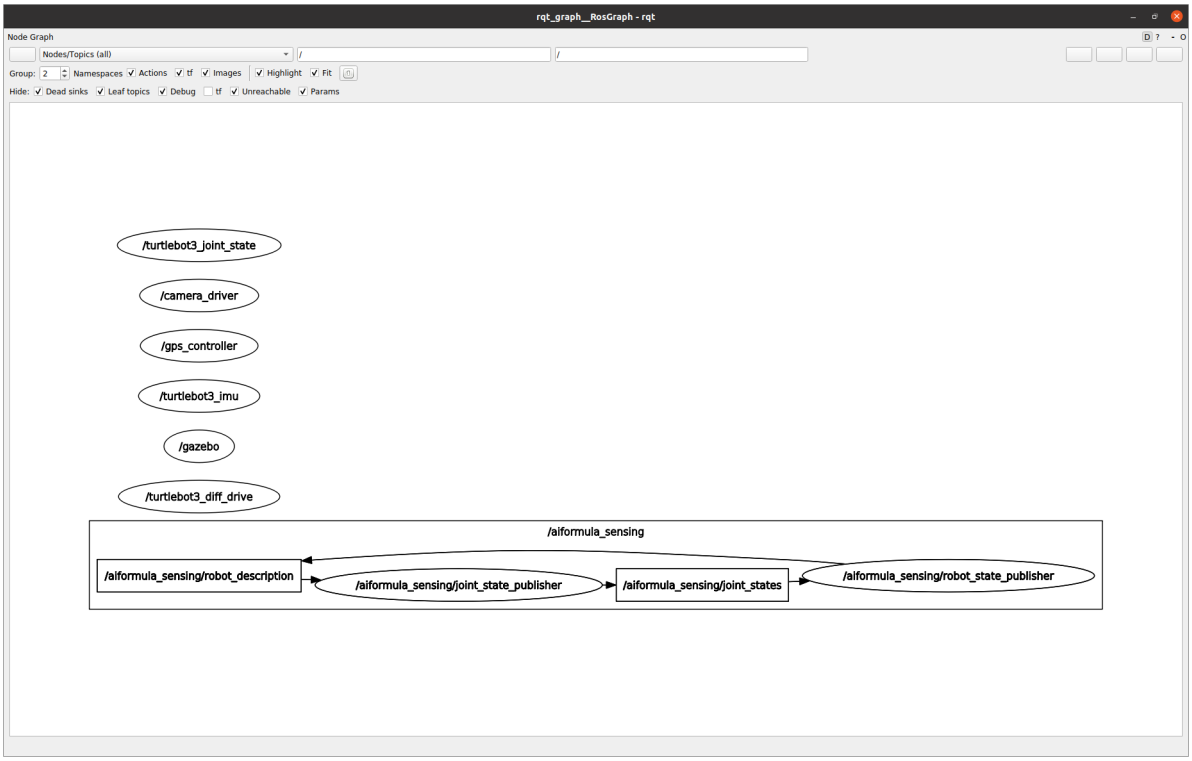


Fig. 5.3: Configuration of ROS 2 nodes

Table.5.1 にシミュレーションする主なセンサとアクチュエータを示す.

Table 5.1: Sensors and actuators to be simulated

Sensor	Camera IMU（Inertial Measurement Devices） GNSS
Actuator	Drive motor

5.2　実験の概要

開発した経路追従ソフトウェアをシミュレータ環境で検証する.

5.3　実験環境

前述の提供されたシミュレータ環境に GNSS+IMU センサ（VectorNav VN-200）の出力を追加したシミュレータ環境を使用する. ソフトウェアには ROS 2 foxy を使用し, シミュレータ環境には Gazebo Classic を使用している. Fig.5.4 にシミュレータで扱うコースの全体図を示す. Table.5.2 にシミュレータに使用する PC の仕様を示す.

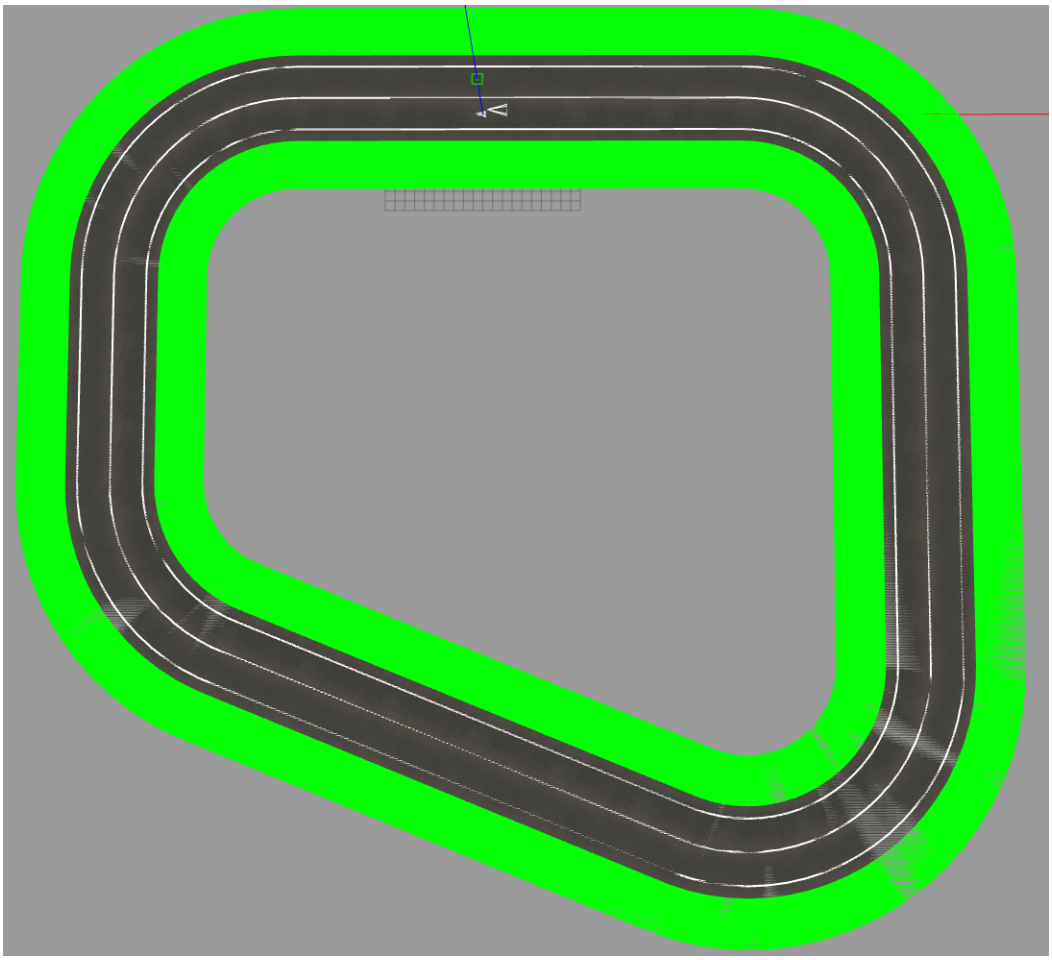


Fig. 5.4: Simulator world top view

Table 5.2: experimental device

Computer	Let‘s note CF-SV
OS	Ubuntu20.04 LTS
ROS 2	Foxy
simulator	Gazebo Classic

5.4　実験内容

Fig5.5 の丸で示された位置から矢印に沿うルートでシミュレータ環境のコースを一周させる. 制御パラメータは以下とし, 目標速度は一定値とする.

目標速度　:　6 [m/s]

制御周期　:　20 [Hz]

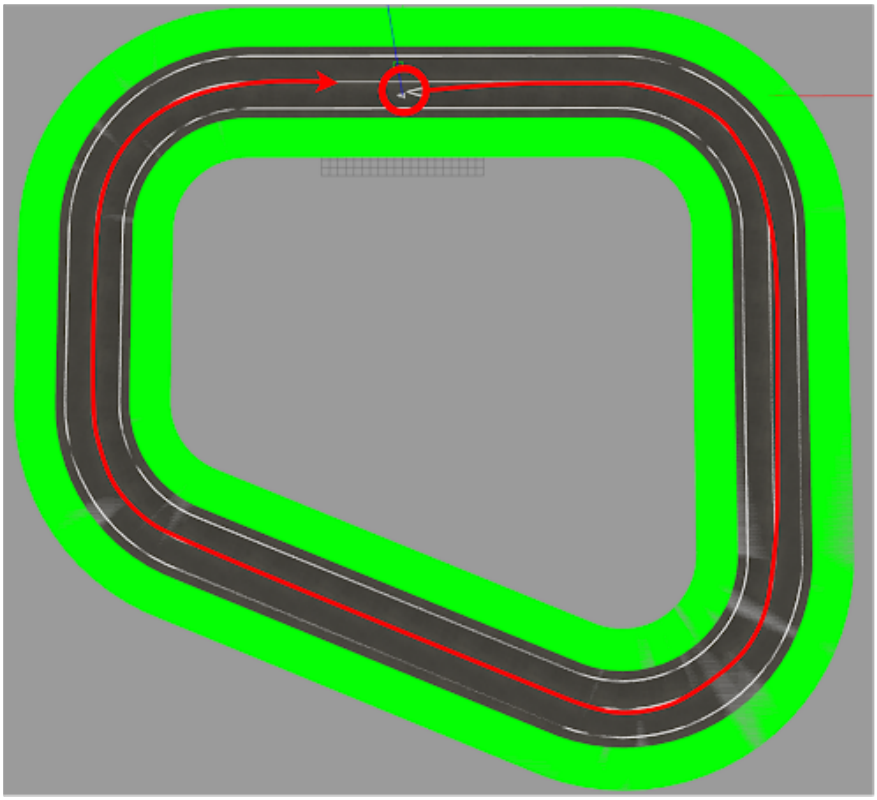


Fig. 5.5: Simulator course

5.5　実験結果

シミュレータ環境で作成した経路追従ソフトウェアを使用することで実験を行った.

シミュレータで自律走行したときのロボットの軌跡を Fig.5.6 に示す. 結果として, シミュレータ環境のコースを周回することができた.

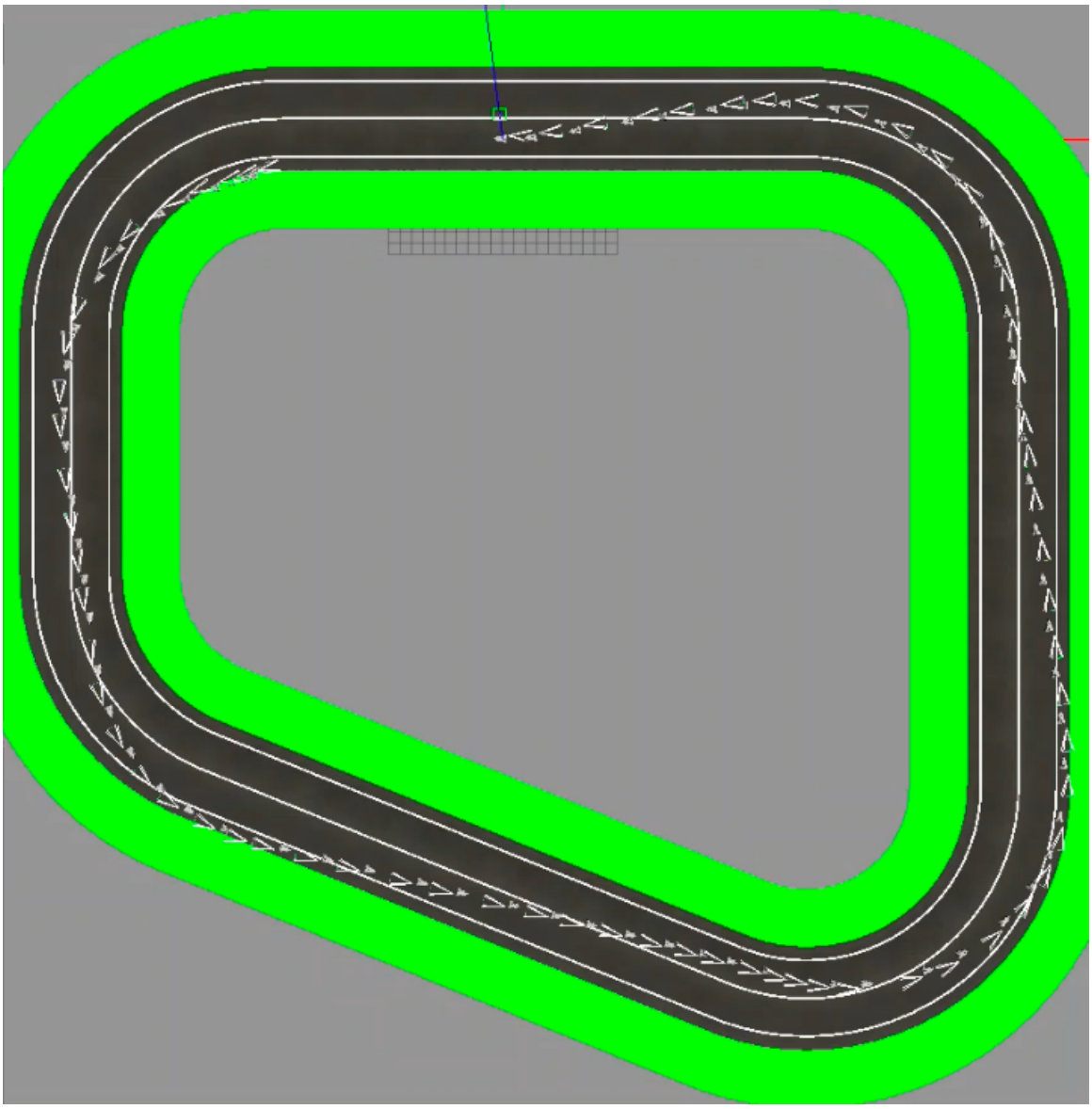


Fig. 5.6: The AI-Formula path-following behavior on the simulator

第 6 章

実環境での実験

本章では, シミュレータ環境で動作が確認された経路追従ソフトウェアを実環境で検証する. また, その結果を考察する.

6.1 実験の概要

開発した経路追従ソフトウェアを実環境で検証する. AI-Formula の会場となっている AI モビリティパーク紫峰で実験を行う. Fig.6.1 に AI モビリティパーク紫峰を示す. 検証をするにあたって, いくつかの評価項目を設ける. 評価項目を Table.6.1 に示す. 追従誤差は目標経路の全ての経路点と GNSS から得たモビリティの座標データの距離から最小の値を取り出すことで求める. 求めた追従誤差を積分することで, 追従誤差の累計を求める.

Table 6.1: evaluation item

評価項目
目標速度（並進速度） [m/s]
ゴールするまでの時間 [s]
追従誤差の累計 [m]
追従誤差の最大値 [m]

Fig. 6.1: Aerial view and mobility perspective of the course

6.2　実験装置

実験装置を Fig6.2 に示す．センサ類は GNSS+IMU センサ (VectorNav VN-200) を使用する．また, 実験の様子を空撮するためにドローン (DJI Air3) を使用した．

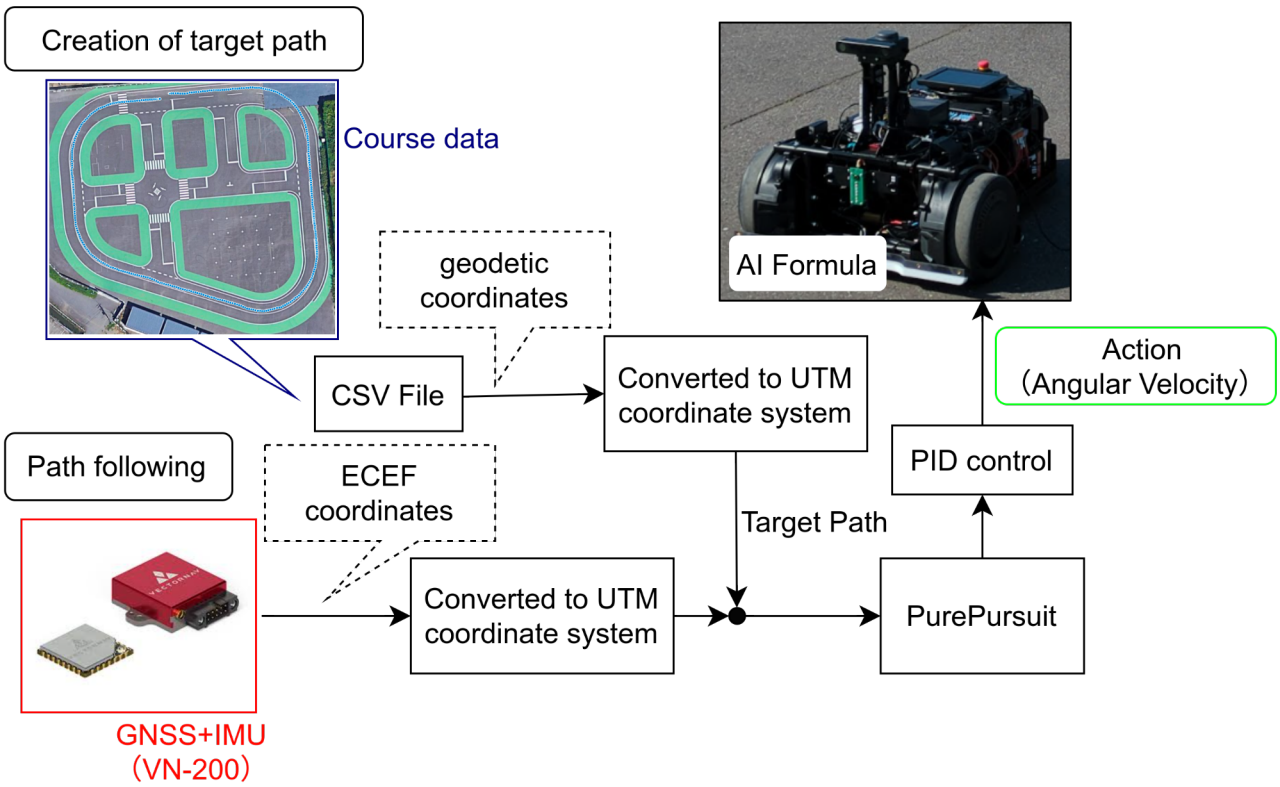


Fig. 6.2: Path-following system

Table 6.2: experimental device

computer	Jetson AGX Orin
OS	Jetpack 5.1.2
ROS 2	foxy

本研究で開発した経路追従ソフトウェアは事前に用意した目標経路に沿って追従する動作を行う．本実験で使用する目標経路は実験当日に AI モビリティパーク紫峰で中央の白線に沿う

ように手動走行を行い, GNSS から得たデータを基に作成した. Fig.6.3 に作成した目標経路を Google Map 上にプロットした図を示す.

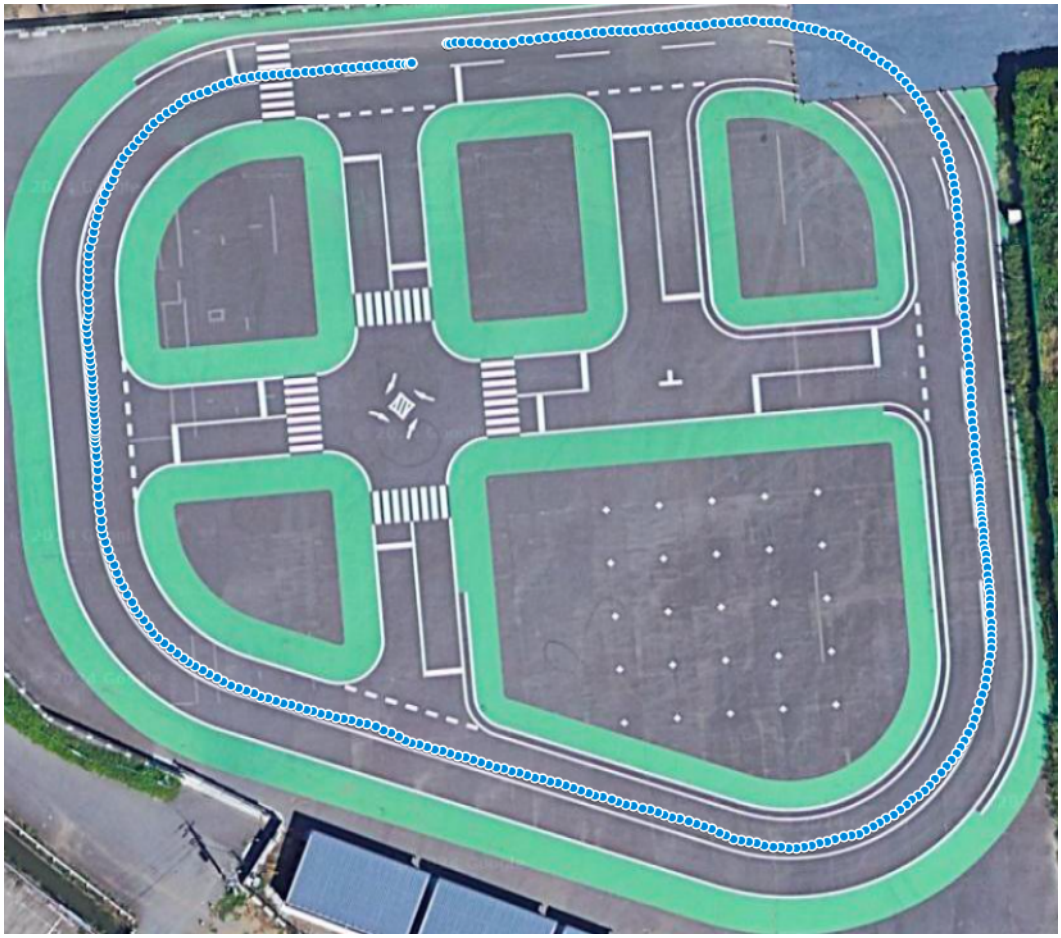


Fig. 6.3: Aerial view and mobility perspective of the course

6.3　実験内容

Fig.6.4 の丸で示された位置から矢印に沿うルートで AI モビリティパーク紫峰の外周を一周させる. 制御パラメータは以下とし, 目標速度は一定とする.

Fig. 6.4: Experiment path

$linear_vel$	:	目標速度 [m/s]
$interval_ms$	:	制御周期 [ms]
p_gain	:	p 制御のゲイン
i_gain	:	i 制御のゲイン
d_gain	:	d 制御のゲイン

6.4　実験結果と考察

6.4.1　実験 1

走行時のパラメータを Table.6.3 に示す.

Table 6.3: experiment1 parameters

linear_vel	3.0
interval_ms	50
p_gain	0.8
i_gain	0.0
d_gain	0.0

実験の結果, 目標並進速度 3m/s で経路から外れること無くコースを一周することができた. 実験時の走行経路と目標経路の軌跡を Fig.6.5 に示す. 走行経路は走行時の GNSS データを基準に求めている. また, 目標経路との誤差の遷移を Fig.6.6 に示す. 最後に評価項目の結果を Table.6.4 に示す.

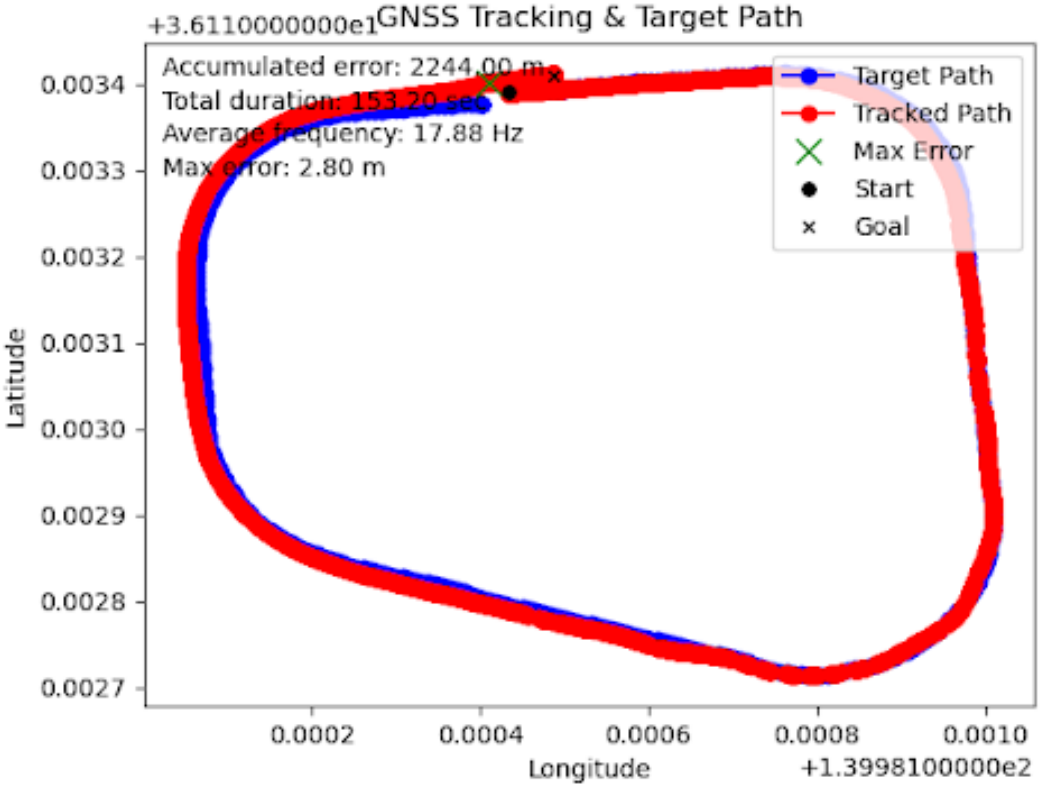


Fig. 6.5: Trajectory of travel route and target route

Fig. 6.6: Transition of error from the target path

Table 6.4: evaluation item

ゴールするまでの時間 [s]	153
追従誤差の累計 [m]	2244.0
追従誤差の最大値 [m]	2.8

6.4.2　実験 2

走行時のパラメータを Table.6.5 に示す.

Table 6.5: experiment1 parameters

linear_vel	6.0
interval_ms	50
p_gain	1.0
i_gain	0.0
d_gain	0.15

実験の結果, 目標並進速度 6m/s で経路から外れること無くコースを一周することができた. コースの初めのカーブ後の道で蛇行する動きを見せたものの, 経路から外れることなく周回することができた. 実験時の走行経路と目標経路の軌跡を Fig.6.7 に示す. また, 目標経路との誤差の遷移を Fig.6.8 に示す. 最後に評価項目の結果を Table.6.6 に示す.

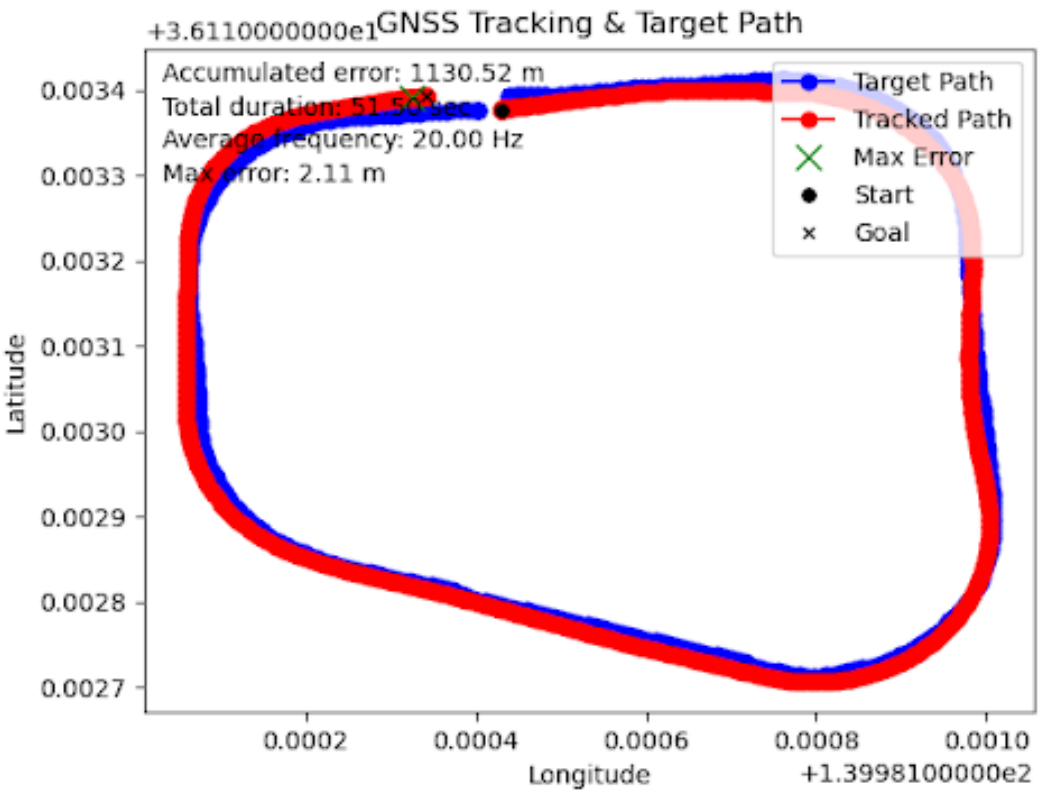


Fig. 6.7: Trajectory of travel route and target route

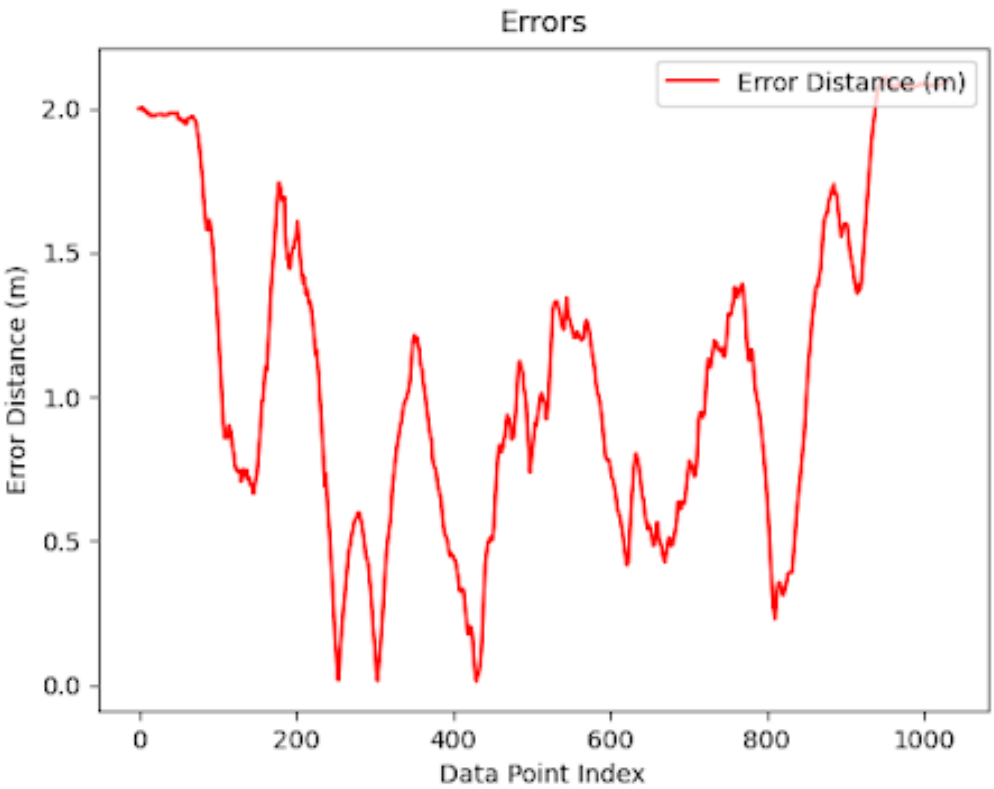


Fig. 6.8: Transition of error from the target path

Table 6.6: evaluation item

ゴールするまでの時間 [s]	51
追従誤差の累計 [m]	1130.0
追従誤差の最大値 [m]	2.11

第 7 章

結論

本研究では, AI-Formula で使用する屋外自律高速モビリティを対象として経路追従行動を行うためのソフトウェア開発を行った. また, 開発した経路追従ソフトウェアをシミュレータ環境と実環境で検証した. 検証の結果, 予め与えられた目標経路に対して経路追従する動作を確認した.

参考文献

[1] 藤野一真，山下響，林原靖男，"千葉工業大学における AI-Formula の開発と学び" 計測自動制御学会 SI2024, (2024).

[2] aiformula GitHub repository. https://github.com/open-rdc/aiformula. (Accessed on 12/20/2024).

[3] ROS 2 Documentation. ROS 2 Documentation. https://docs.ros.org/en/foxy/index.html.

[4] ros2. ros2/rviz GitHub repository. https://github.com/ros2/rviz.

[5] esri ジャパン. 日本で使用される座標系. https://www.esrij.com/gis-guide/coordinate-and-spatial/coordinate-system-japan/.

[6] VN-200. https://www.vectornav.com/products/detail/vn-200.

[7] vectornav GitHub repository. https://github.com/dawonn/vectornav/tree/ros2.

[8] 藤野一真, 林原靖男, “AI-Formula 用シミュレータの拡張とそれを用いた自動化ソフトウェアの開発”, 計測自動制御学会 SI2024, (2024).

[9] 岡田眞也, 秋元優輝, 加藤敦, 安井裕司, “AI フォーミュラ”, 自動車技術会 2024 年春季大会, (2024).

[10] aiformula-support GitHub repository. https://github.com/aiformula-support/aiformula. (Accessed on 12/20/2024).

付録

実験動画を Youtube で公開している．また, 本研究で開発したパッケージや AIFormula 千葉工業大学チームが開発したリポジトリは github で公開している.

- 実験動画のリンク https://youtu.be/nZ-q6UWQDuA
- github のリンク https://github.com/open-rdc/aiformula

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, チームの活動を通して知識や技術を教えていただきました, AIFormula 千葉工業大学チームの皆様に感謝いたします. 最後に, 日常生活の多くを支えてくださった両親と祖父母に感謝いたします.