

## 非線形変換フィルターを用いた GNSS/IMU の融合による リアルタイムかつ高精度な位置推定システムの実現

石智光\* 小高知宏\*\* 黒岩丈介\*\* 諏訪いずみ\*\*\*

### Real-time and highly accurate position estimation system by GNSS/IMU fusion with nonlinear transformation filters

Tomomitsu ISHI\*, Tomohiro ODAKA\*\*, Jousuke KUROIWA\*\*, Izumi SUWA\*\*\*

(Received January 31, 2025)

Global Navigation Satellite Systems (GNSS) are widely used in transportation and autonomous driving, but face challenges like reduced accuracy in urban areas and ionospheric effects. This study integrates GNSS and Inertial Measurement Unit (IMU) data using an Unscented Kalman Filter (UKF) for real-time positioning, implemented on a Raspberry Pi. Experiments on mountain roads showed that the UKF achieved accuracy similar to GNSS in straight sections but degraded in curves due to insufficient orientation updates. The system used GNSS at 1 Hz and IMU at 300 Hz, processed via Python. Improvements in orientation updates and computational efficiency remain key challenges.

**Key words** : GNSS, IMU, UKF, Nonlinear Systems, Real-time Positioning, Data Fusion

## 1. はじめに

近年、全球測位衛星システム (Global Navigation Satellite System; GNSS) のサービスが拡大し、その需要がますます高まっている。<sup>[1]</sup> GNSS は、人工衛星からの電波信号を受信して地球上の任意の地点で高精度な位置、速度、時刻情報を取得するシステムであり、主要なシステムとして米国の GPS、ロシアの GLONASS、欧州連合の Galileo、中国の BeiDou が含まれる。これらは交通・物流、測量・建設、農業、防災、通信インフラ、科学研究、自動運転など、多岐にわたる分野で広く利用されている。GNSS の基本原理は、衛星から送信される信号の到達時間を測定し、それを基に受信機の位置を算出する「三次

元測位」に基づいている。<sup>[2]</sup> 具体的には、少なくとも 4 つの衛星からの信号を受信し、それぞれの衛星との疑似距離を計算する。疑似距離は、衛星と受信機の間で同期された時刻情報を用いて信号の伝播時間に光速を乗じて求められるが、受信機の時計には誤差が含まれるため、位置座標  $(x, y, z)$  と時刻誤差の 4 つの未知数を解く必要がある。このため、4 つの衛星からの情報が必要となる。

高精度な GNSS 技術は非常に便利である反面、都市部や屋内環境、さらには電離層や対流圏の影響により、位置精度が大幅に低下するという課題がある。都市部や屋内では、高層ビルや建物が衛星からの信号を遮蔽したり、信号が壁や窓で反射・屈折してマルチパス干渉<sup>[3]</sup>を引き起こすため、受信機が正確な信号を取得できず、位置情報の誤差が増大する。また、電離層や対流圏では、衛星からの信号がこれらの大気層を通過する際に屈折や遅延が生じ、信号の伝播時間を正確に測定できないことから、位置情報に誤差が生じる。

IMU は加速度計やジャイロスコープを用いて移動体の加速度や角速度を測定し、外部の信号に依存せずに位置や姿勢を推定できる低コストのセンサーである。<sup>[4]</sup> こ

\*1 大学院工学研究科 知識社会基礎工学専攻

\*1 Fundamental Engineering for Knowledge-Based Society, Graduate School of Engineering

\*\* 知能システム工学講座

\*\* Department of Human and Artificial Intelligent Systems

\*\*\* 仁愛女子短期大学生活科学学科

\*\*\* Jin-ai Women's College

れにより、都市部や屋内環境など GNSS の信号が遮蔽されたりマルチパス干渉が発生する環境でも、連続的な位置推定が可能となる。自律移動体（ドローン、ロボット、自動運転車など）では、GNSS と IMU を組み合わせることで、高精度かつ信頼性の高いナビゲーションが実現できる。IMU の長所として、信号遮蔽の影響を受けないことや、高い時間分解能により動的な運動の検出やリアルタイム制御に適していることが挙げられる。一方、短所として、センサーのドリフトやノイズにより時間とともに位置推定の誤差が累積すること、IMU 単独では初期位置からの相対的な変化しかわからず、絶対位置の取得が困難であることがある。<sup>[5]</sup>

そこで、本研究では、まず GNSS と IMU のデータを収集し、それぞれのセンサーの特性を分析した。次に、アンセンテッドカルマンフィルター（UKF）を用いたデータ融合アルゴリズムを設計し、シミュレーション環境でその性能を評価した。具体的には、木が多い山道で、GNSS の信号が遮蔽されやすいシナリオを設定し、IMU のデータと組み合わせることで位置推定の精度向上を図った。さらに、実際の移動体にセンサーを搭載し、実験を行うことで、シミュレーション結果との比較検証を行った。

## 2. GNSS・IMU の融合による位置推定システム

第一章の最後に、本研究の目的について述べた。そして、本章で GNSS と IMU センサーのデータを融合し、アンセンテッドカルマンフィルター（UKF）を用いて位置推定を行うシステムについて述べる。まず、アンセンテッドカルマンフィルターの基礎であるカルマンフィルター（KF）の定義を述べる。次に、アンセンテッドカルマンフィルターと同じく、非線形システムモデルを対象にする拡張カルマンフィルター（EKF）を用いた位置推定の説明やその手法の限界を述べる。最後に、GNSS と IMU センサーを融合したデータで、アンセンテッドカルマンフィルターを用いて位置推定を行う仕組みについて述べる。

### 2.1 カルマンフィルターの概要

カルマンフィルターは、動的システムにおける状態推定を行うためのアルゴリズムで、観測データがノイズを含む場合でも、最も確からしい状態を推定するのに用いられる。具体的には、連続的に変化するシステムの状態（位置や速度など）を、時間の経過とともに更新していくプロセスである。カルマンフィルターは、予測ステップ

と更新ステップの2つの段階から成り立ち、観測データとシステムモデルを組み合わせて推定を行う。

カルマンフィルターは、予測ステップと更新ステップの2つの段階に分かれている。予測ステップでは、まず、システムの現在の状態と制御入力をもとに、次の状態を予測する。次に、状態遷移方程式を使用して、前の状態から新しい状態を算出する。最後に、予測状態とともに、推定誤差の共分散行列も更新され、次の状態の不確かさが反映される。更新ステップでは、まず実際に観測されたデータを取り入れ、予測した状態を修正する。次に、観測データと予測値との差（イノベーション）を計算し、この差を用いて状態を補正する。次に、カルマンゲインという係数を用いて、予測と観測のどちらに重きを置くかを調整する。カルマンゲインは、観測ノイズと予測誤差の共分散を考慮して計算される。最後に、状態推定が修正され、推定誤差の共分散行列も更新されて新たな不確かさが計算される。

カルマンフィルターは、線形ガウス系において最適な状態推定器として機能する。線形ガウス系とは、システムの状態が線形な関数によって変化し、観測およびシステムノイズがガウス分布（正規分布）に従うシステムのことを指す。この条件下では、カルマンフィルターは理論的に最も正確な推定を行うことができる。

### 2.2 拡張カルマンフィルター（EKF）を用いた位置推定

GNSS と IMU のデータを統合し、拡張カルマンフィルター（EKF）を用いて位置推定を行う研究がある。<sup>[6]</sup> これは、両センサーから得られる情報を組み合わせて高精度な位置情報を取得する方法である。IMU は高頻度で加速度や角速度を測定できるが、累積誤差が生じやすい。一方、GNSS は絶対的な位置情報を提供するものの、更新頻度が低く、信号の遮蔽やマルチパス干渉の影響を受けやすい。EKF は非線形なシステムモデルを扱えるため、IMU の高頻度データで状態の予測を行い、GNSS の観測データでその予測を補正する。この予測と更新のプロセスを繰り返すことで、両センサーの長所を活かし短所を補い、リアルタイムで高精度な位置推定が可能となる。

しかし、拡張カルマンフィルター（EKF）を用いた手法にはいくつかの欠点がある。EKF は非線形システムを一次のテイラー展開で線形近似するため、高度な非線形性を持つシステムでは近似誤差が大きくなり、推定精度が低下する可能性がある。また、システムモデルや観測モデルのヤコビアン行列を導出する必要があり、モデル

が複雑になると計算量が増大し、リアルタイム処理が難しくなる。

## 2.3 アンセンテッドカルマンフィルタ（UKF）を用いた位置推定

EKFを用いた位置推定に対して、アンセンテッドカルマンフィルタ（UKF）はこれらの欠点を補うことができる。UKFはシグマポイントと呼ばれるサンプル点を用いて、非線形関数の統計的特性を高精度に推定するため、線形化による近似誤差を回避できる。ヤコビアン行列の計算が不要であり、モデルの複雑さに関わらず実装が容易である。さらに、高度な非線形性を持つシステムでも安定して高精度な推定が可能となり、GNSSとIMUのデータ融合においてEKFの限界を克服できる。この章では、まずアンセンテッドカルマンフィルタの概念について述べる。次に、アンセンテッドカルマンフィルタのアルゴリズムそして特性を述べる。

従来のカルマンフィルタは線形システムに対して最適なフィルタである。しかし、多くの実世界のシステムは非線形である。そのため、非線形システムにカルマンフィルタを適用するには、拡張カルマンフィルタ（EKF）またはアンセンテッドカルマンフィルタ（UKF）が使われる。

UKFはアンセンテッド変換を使用して、EKFよりも精度が高く非線形性を正確に扱うことができる。確率分布の非線形変換を正確に計算するのは困難である。EKFは平均値を非線形関数に通すだけの単純化を行うため、精度が低下する場合がある。

そこで、アンセンテッド変換は3ステップに分ける。まずは、シグマポイントを作成する。次に、各シグマポイントに対して非線形変換を適用する。最後に、変換後のシグマポイントを用いて、非線形変換の分布の平均と分散を計算する。

シグマポイントとは、確率分布を効率的に表現し、非線形システムでの推定を行うために使用される代表的な点の集合である。非線形システムにおいて、状態分布を非線形関数に通すと、結果として得られる分布を正確に計算するのは困難である。シグマポイントを用いることで、確率分布をより正確に表現し、非線形変換後の分布の平均値や分散を計算する際の精度を向上させる。シグマポイントは、ガウス分布などの状態分布を効率的にサンプリングするために選ばれる点の集合である。これらの点は、分布の平均や分布の分散、高次の情報という特徴をもつ。それぞれは、シグマポイントの重み付き平均が元の分布の平均値に一致する、シグマポイントの重み付き分散が元の分布の共分散に一致する、状態分布の

非線形性を適切に表現するため少ない点数で高次モーメントを近似する。

シグマポイントの生成手順として、まずは状態平均値と共分散行列を取得する。次に、スケールパラメータを設定して、シグマポイントを計算する。最後に、シグマポイントの重みを設定する。

UKFのアルゴリズムについて述べる。UKFは予測ステップと更新ステップの2ステップで構成される。

まずは、システムモデルについて述べる。UKFは以下の非線形状態空間モデルを仮定する。

- 状態遷移方程式

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1}$$

- $\mathbf{x}_k$ : 現在の状態ベクトル
- $f()$ : 状態遷移関数
- $\mathbf{w}_{k-1}$ : プロセスノイズ

- 観測方程式

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k$$

- $\mathbf{z}_k$ : 現在の観測値
- $h()$ : 観測関数
- $\mathbf{v}_{k-1}$ : 観測ノイズ

UKFアルゴリズムの流れとして、まず、状態分布と共分散の初期値を設定する。次に、シグマポイントを生成する。次に、状態遷移関数での次のシグマポイントを予測し、平均値と共分散を計算する。最後に、観測値を用いてカルマンゲインを計算し、状態と共分散を更新する。そして、この流れを繰り返す。

このようにして、GNSSとIMUセンサーを融合したデータで、アンセンテッドカルマンフィルタを用いて、GNSS単独な位置精度を上回るような位置推定を行う。

## 3. システムの実装

第二章ではGNSSとIMUセンサーを融合したデータで、アンセンテッドカルマンフィルタを用いて位置推定を行うシステムについて説明した。この章ではUKFを用いてGNSSとIMUによる位置推定システムの実装を説明する。まず、GNSSとIMUのデータを集計するシステムを説明する、それをハードウェアの構成とソフトウェアフローの二つに分けて説明する。次に、GNSSとIMUのデータ取得と処理・データ同期について述べる。最後に、集計したデータをUKFを用いて位置推定を行うための環境を述べる。

### 3.1 システム概要

#### 3.1.1 ハードウェア構成

まずはシステムのハードウェア構成について説明する。本研究で位置情報や加速度情報、角速度情報を同期的に受信し、保存するために使われたデバイスはラズベリーパイである。ラズベリーパイは小さくて持ち出しやすいため、実験する際に車に配置しやすい。そして、ラズベリーパイの計算力は一般のコンピュータよりも低い、計算コストもコンピュータよりも低いので、低計算コストの場合での同期的に情報を受信して保存することを目指している。

位置情報を取得するために使った GNSS デバイスは SIM7600A-H モジュールを搭載したデバイスである。この GNSS デバイスはラズベリーパイ専用のデバイスであり、物理的に接続して簡単に利用することができる。そして、使用したアンテナは 2.7~5.0V 範囲のラズベリーパイに適用するアンテナである。SIM7600A-H モジュールはラズベリーパイと GPIO そして USB ポート接続している (図 1)。



図 1: データ収集デバイス

加速度情報と角速度情報を取得するために使った IMU デバイスは ICM20948 である。この IMU センサーは I2C と SPI 通信を対応している。本研究では、IMU センサーから加速度と角速度というシンプルな情報を受信するため、より少ないピン接続かつシンプルな通信に適している I2C 通信を使用し、ラズベリーパイと GPIO 接続を行う。

プログラムを操作するためにディスプレイモニターとキーボードを使った。モニターはラズベリーパイ用の 800\*480 解像度の LCD ディスプレーモニターで、キーボードは Bluetooth 接続可能なキーボードを使用した。そして、システム全体の電源として、PHILIPS の 5V=2A

の 20000mAh のモバイルバッテリーを使用する。

このシステムのハードウェアはラズベリーパイ・GNSS アンテナ・GNSS モジュール・IMU センサー・モニター・キーボードで構成されている。このようにして、実験する際に簡単に車に配置し操作することができる。そして、位置情報や加速度情報、角速度情報を正確に取得し、位置推定に使われるために保存することができる。

表 1: 使用ハードウェアの詳細

デバイス	モデル	詳細
ラズベリーパイ	Raspberry Pi 4	小型コンピュータ, 低計算コスト
GNSS モジュール	SIM7600A-H	GNSS デバイス, GPIO および USB 接続
GNSS アンテナ	-	2.7 5.0V 範囲対応
IMU センサー	ICM20948	加速度・角速度センサー, I2C 通信
ディスプレイモニター	-	800*480 解像度 LCD
キーボード	-	Bluetooth 接続
モバイルバッテリー	PHILIPS	5V=2A, 20000mAh

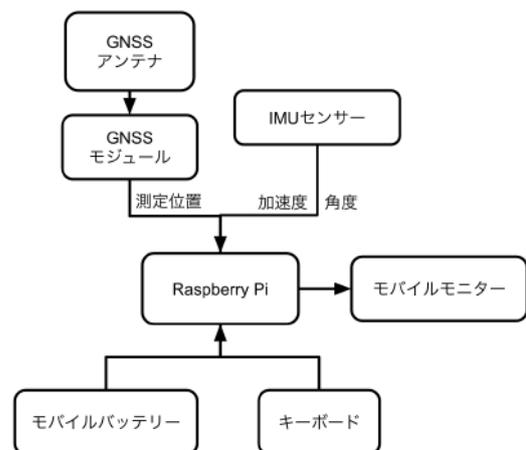


図 2: ハードウェア構成図

### 3.1.2 ソフトウェアフロー

この節では、ハードウェア間の接続方法とソフトウェアフローを説明する。

GNSS コンテナから受信した信号を位置情報に変換する機器である GNSS モジュールはラズベリーパイとシリアル通信を介して AT コマンドを用いてデータの通信をしている。USB ポートを仮想 COM ポートとして通信する。これにより、ラズベリーパイと GNSS モジュールとの間で、AT コマンドを用いてデータの送受信が可能となる。

AT コマンドは、モデムや通信モジュールなどのデバイスを制御・設定するためのテキストベースのコマンドセットであり、主にシリアル通信 (UART や USB を経由した仮想 COM ポートなど) を通じて送受信される。「AT」は「Attention」の略で、すべてのコマンドはこの接頭辞で始まる。AT コマンドを用いることで、デバイスに対してダイヤル発信 (ATD)、通話の応答 (ATA)、接続の切断 (ATH)、ネットワーク設定の変更、SMS の送受信、データ通信の開始・終了など、多岐にわたる操作を行うことができる。これらのコマンドは国際標準化団体 (ITU-T) によって標準化されており、多くの通信デバイスで互換性がある。AT コマンドは人間が直接入力して操作することも可能であり、組み込みシステムやアプリケーションから自動的に送信することで、デバイスをプログラマティックに制御することもできる。この柔軟性と汎用性から、AT コマンドは通信デバイスの設定や制御において広く利用されている。

加速度情報と角速度情報は IMU センサーから受信し、処理され、I2C 通信 (Inter-Integrated Circuit) 接続でラズベリーパイへ送信される。I2C 通信は、マイクロコントローラやセンサーなどのデバイス間でデータを送受信するためのシリアル通信プロトコルである。主に 2 本の信号線、SDA (データ線) と SCL (クロック線) を使用し、マスター・スレーブ方式で通信を行う。マスターはクロック信号を生成し、スレーブデバイスを個別のアドレスで識別してデータを送信または受信する。I2C はシンプルなハードウェア構成と柔軟なデバイス接続が可能であり、低速から中速の通信速度 (通常 100kHz から 400kHz) で複数のデバイスを同一バス上で制御できるため、組み込みシステムや電子機器で広く利用されている。

本研究では、ラズベリーパイを用いて位置情報、加速度情報、角速度情報を受信し、保存する。情報の受信には Python で作成したプログラムを使用しており、位置情報の取得には serial ライブラリを、加速度および角速度情報の取得には IMU センサー ICM20948 専用の

icm20948 ライブラリを用いている。これら二つのライブラリを活用することで、短いコードで情報を簡便かつ正確に取得することが可能となった。

### 3.1.3 データ取得と処理の流れ

この節では、センサーから取得するデータの流れる方と処理方法というデータ処理プロセスを説明する。

位置情報は GNSS アンテナから GNSS モジュールへ送信し、処理され、ラズベリーパイへ送信される。まず、衛星から発信した信号を GNSS アンテナで受信する、そして、受信した信号を GNSS モジュールが受信し、それを人間が読めるようなデジタル情報に変換する。次に、変換されたデジタル情報の中に含まれる位置情報をラズベリーパイが受信する。ラズベリーパイが 1 秒間に 1 回取得した位置情報を取得して、CSV ファイルに保存する。

GNSS から取得した位置情報のサンプリングレートは 1 秒間 1 回である。GNSS 衛星は、自身の位置や時刻情報を 1 秒ごとに地上に送信しており、受信機はこれらの情報を基に位置を算出する。また、1 秒間隔のデータは多くの用途で十分な精度とリアルタイム性を提供し、電力消費やデータ処理量とのバランスも取れている。高いサンプリングレートを使用するとデータ量や消費電力が増加するため、一般的には 1 秒に 1 回のサンプリングが最適とされている。

加速度情報と角速度情報は IMU センサーから 1 秒間 200~400 回取得する。IMU センサーのサンプリングレートは一般的に 200~400Hz である。それは移動体や機械の動的な挙動を正確に捉えるためである。この範囲のサンプリングレートは、人間の動作や車両の運動など、多くのアプリケーションで必要とされる動きの周波数成分を十分に対応することができる。例えば、ナビゲーションやモーションキャプチャ、安定化制御などでは、対象の加速度や角速度が高速で変化することがあるため、高いサンプリングレートが求められる。一方で、サンプリングレートを過度に高くすると、データ量が増大し、処理負荷や消費電力が増えるため、200~400Hz という範囲が精度と効率のバランスが取れた最適な値となる。

### 3.1.4 データ同期の手法

GNSS と IMU という複数のセンサーからデータを取得する際、各データが異なるタイミングで取得される可能性がある。異なるタイミングのデータをそのまま使用すると、データ間の不整合が生じ、正確な位置推定が

困難になる。データの同期を行うことで、各センサーからのデータを同一の時間基準で扱い、精度の高いデータ融合を可能にする。

本研究で使われるデータは位置座標、加速度、角速度である。GNSS から取得する位置座標は経度と緯度で、サンプリングレートは 1 Hz。そして、IMU から取得する加速度と角速度はロール、ピッチ、ヨーで、サンプリングレートは 300Hz。

## 3.2 UKF を用いた位置推定環境

GNSS・IMU データによる UKF を用いた位置推定を行う環境は Google Colaboratory<sup>[7]</sup> というブラウザから Python を実行できるサービス環境である。Google Colaboratory を使うメリットとして、Python の実行環境をしなくて良い、そして、UKF で必要な外部ライブラリは簡単にインストールすることができる。

GNSS と IMU から集計したデータを用いて、Google Colaboratory で位置推定を行う。そして、推定した位置と GNSS のみでの位置を地図にプロットし、HTML ファイルに保存する。その HTML ファイルをブラウザで開くと推定位置と GNSS のみでの位置が表示され、比較することができる。

## 4. UKF を用いた位置推定の実験

この章では UKF を用いた位置推定の実験について説明する。まずは、実験の準備として環境と UKF モデル設定を述べる。次に、使用機器を述べる。次に、評価指標と評価方法を述べる。最後に、実験結果を述べる。

### 4.1 実験環境

今回の実験環境は福井大学付近の足羽山の山道（図 3）を選定した。この山道は曲がりくねっており、かつ木々が茂っているため、UKF を用いて推定された座標と GNSS によって提供される座標を比較するのに非常に適している。山道の曲がり具合は、IMU センサーの角速度性能および UKF アルゴリズムにおけるその性能を検証するために重要である。また、木々が茂っていることで、GNSS の座標が影響を受けやすくなり、精度が低下する。そのため、IMU センサーの加速度性能および UKF アルゴリズムにおけるその性能を試験する条件としても最適である。



図 3: 足羽山の道路図、黒線は今回の実験で車が移動した軌跡

今回の実験では、データ収集時の天候として晴天を選択した。これは、GNSS の座標が木々の遮蔽だけでなく、曇り空の影響も受けやすいためである。特に、曇りによる遮蔽が原因で GNSS 信号が受信できなくなる可能性がある。そのため、本実験では晴天を選択し、GNSS から受信した座標と UKF を用いて推定した座標をより正確に比較できる条件を整えた。

今回の実験では、デバイスを車両に設置し、IMU デバイスの x 軸方向を車両の進行方向と一致させた。山の麓から山頂まで走行し、その間に GNSS で受信した座標データと、IMU センサーで取得した加速度および角速度のデータを記録した。GNSS 受信機の制約により、1 秒ごとに 1 セットの座標データ（経度、緯度、高度、タイムスタンプなど）が収集された。一方、IMU センサーからは、1 秒間に 300 セットの加速度 ( $\mathbf{a} = (a_x, a_y, a_z)$ ) および角速度 ( $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$ ) のデータが収集された。IMU データの各秒の最初のセットは、その秒の GNSS データのタイムスタンプと一致している。これらのデータは Python コードを用いて収集され、GNSS データと IMU データは Python の並列処理を利用して同時に収集された。

### 4.2 UKF モデル設定

本実験で使用した UKF モデルは、2.4 節で定義した状態方程式と観測方程式を基に構築された。以下に状態方程式と観測方程式を示す。

- 状態遷移方程式

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{a}_{k-1}) + \mathbf{w}_{k-1} \quad (1)$$

状態遷移方程式において、 $\mathbf{x}_k$  は現在の状態ベクトルを表し、 $f()$  は前時刻の状態ベクトル  $\mathbf{x}_{k-1}$  と加速度  $\mathbf{a}_{k-1}$  を用いて現在の状態を計算する遷移関数である。 $\mathbf{w}_{k-1}$  はプロセスノイズを示しており、システム内部の不確実性を含めるためのランダムな要素である。

- 観測方程式

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \quad (2)$$

観測方程式では、 $\mathbf{z}_k$  は現在の観測値を表し、 $h()$  は現在の状態ベクトル  $\mathbf{x}_k$  に基づいて観測値を推定する関数である。 $\mathbf{v}_k$  は観測ノイズであり、観測データに含まれる不確実性を考慮するための要素である。

次に、変数の定義を行う。本実験で使用する UKF モデルでは、データ収集装置から得られた座標  $(x, y)$ 、加速度  $(a_x, a_y)$ 、および角速度  $(\omega_z)$  を利用する。また、離散サンプリングによって、すなわち 1/300 秒ごとに記録された加速度を用い、数値積分の方法で速度を近似計算する。速度計算の離散形式の公式は以下の通りである：

$$v_k = v_{k-1} + a_k \cdot \Delta t \quad (3)$$

ここで、 $v_k$  は時刻  $k$  における速度を表す。 $v_{k-1}$  は時刻  $(k-1)$  における速度であり、現在の速度は直前の速度に加速度を時間間隔  $\Delta t$  で乗じた値を加えることで算出される。加速度  $a_k$  は時刻  $k$  における値を指し、今回の実験では、IMU センサーから得られるデータを用いている。時間間隔  $\Delta t$  は、IMU データの記録間隔に対応し、ここでは 1/300 秒と設定されている。

状態変数  $\mathbf{x}$  の定義は以下の通り：

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ v_x \\ v_y \\ a_x \\ a_y \\ \omega_z \end{bmatrix} \quad (4)$$

このベクトルに含まれる各要素の意味は以下の通りである。 $x$  と  $y$  は車両の位置座標を示す。 $v_x$  と  $v_y$  はそれぞれの軸方向の速度を表し、 $a_x$  と  $a_y$  は対応する軸方向の加速度を示す。また、 $\omega_z$  は車両の回転運動を表す角速度を指す。これらの変数を組み合わせることで、車両の運動状態を包括的に表現できる。

プロセスノイズは共分散行列  $Q$  で表され、システム内部の不確実性を反映する。加速度と角速度には独立したノイズがあり、 $Q$  の対角要素は IMU センサーの精度パラメータに基づいて設定される。一方、観測ノイズは共分散行列  $R$  で定義され、観測データに含まれる不確実性を反映する。GNSS で取得される位置座標のノイズは、GNSS 受信機の位置精度誤差に基づいて設定される。

次に、初期値を設定する。初期状態ベクトル  $\mathbf{x}_0$  の座標は、GNSS から取得した最初の座標データを利用する。実験開始時点では車両が静止しているため、速度ベクトルは  $\mathbf{0}$  に設定し、加速度と角速度もそれぞれ  $\mathbf{0}$  に設定する。また、初期共分散行列  $P_0$  は、使用する IMU センサーの誤差に基づいて設定する。加速度の誤差は  $\pm 0.4 \text{ m/s}^2$ 、角速度の誤差は  $\pm 0.071 \text{ rad/s}$  とする。観測ノイズ  $R_0$  は、GNSS 受信機の誤差が約 2 メートルであることに基づき、2 メートルと設定する。

次に、シグマポイントの生成について述べる。UKF の核心は、Unscented 変換を使用して非線形システムの特徴を正確に捉える点にある。Sigma 点の数は通常  $2n+1$  ( $n$  は状態の次元数) に設定される。Sigma 点を生成する式は以下の通りである：

$$\chi = \mathbf{x} \pm \sqrt{(n+\lambda)\mathbf{P}} \quad (5)$$

ここで、 $\lambda$  はスケールパラメータで、Sigma 点の分布範囲を調整するために用いる。

最後に、UKF の再帰的な計算式について述べる。計算は主に 2 つの段階に分かれる。最初の段階は予測ステップであり、状態方程式を用いて次の時刻 (1/300 秒後) の状態  $\mathbf{x}$  と共分散  $Q$  を予測する。次の段階は更新ステップであり、観測方程式を使用して予測した状態を修正する。この 2 つのステップを繰り返すことで、UKF はリアルタイムでの高精度な状態推定を実現する。

### 4.3 評価指標

今回の実験の評価指標は、Folium ライブラリを用いて生成した地図上で推定した座標と GNSS 座標の精度を比較することである。図 3 は Folium ライブラリを用いて生成した足羽山の地図であり、今回の実験ではこの図中の黒い線で示されたルートに沿ってデータを記録した。このルートは道路が曲がりくねり、木々が茂っているため、UKF による推定座標と GNSS 座標の比較に非常に適している。

木々が GNSS 信号を遮蔽するため、車両が白いルートに沿って走行していても、GNSS の元データの座標はルートから外れてしまう可能性が高い。そのため、UKF で推定した座標と GNSS 元データの座標を地図上で比較し、それらが道路中心からどれだけ離れているかを評価する。この評価基準に基づき、もし GNSS 元データの座標が推定座標よりも道路中心に近い場合、今回の UKF モデルは実験の理想的な結果を達成していないと判断する。一方で、推定座標が道路中心に近い場合、UKF を用いた推定座標の方が GNSS 元データよりも高精度であることを示し、実験の目的を達成したといえる。

## 4.4 UKF を用いた位置推定の実験結果

GNSS の元データの座標は図 4 の青い線に示されており、UKF を用いて推定した座標は図 4 の赤い線に示されている。図から明らかなように、UKF を用いて推定した座標は GNSS の元データの座標よりも道路の中心からずれており、推定座標の精度は元データの座標よりも低いことが分かる。



図 4: 実験結果

そして、図表で表すと、GNSS のみでの位置が道路中心からの平均距離は約 5 メートルで、UKF を用いて推定した位置は道路中心からの平均距離は約 8 メートルとなることがわかる。

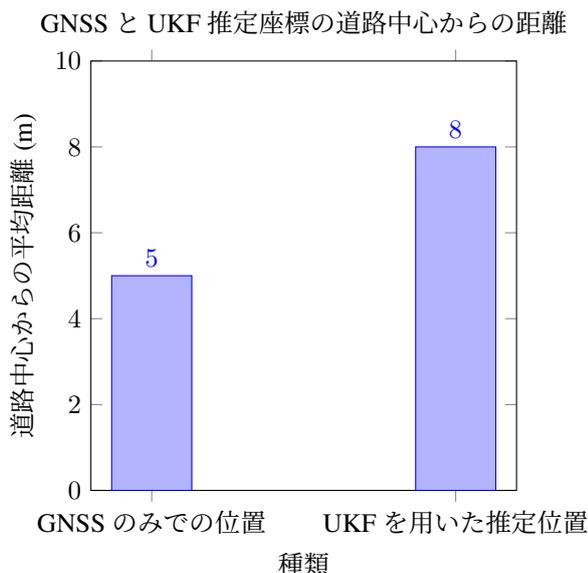


図 5: GNSS と UKF 推定座標の道路中心からの距離の比較

表 2: GNSS と UKF 推定位置の道路中心からの平均距離

種類	道路中心からの平均距離 (m)
GNSS のみでの位置	5
UKF を用いた推定位置	8

## 5. 考察

この実験結果から、GNSS と IMU を融合して推定した位置の経路（赤色の線）が、GNSS のみの観測位置（青色の線）よりも精度が低下した要因として、予測ステップにおける車両の向き（角度）の更新が適切に行われていないことが考えられる。図 6 の緑色の線は、GNSS と IMU による 1 秒間の予測位置を示しており、各ステップで車両の向きがほとんど変化していないことが確認できる。

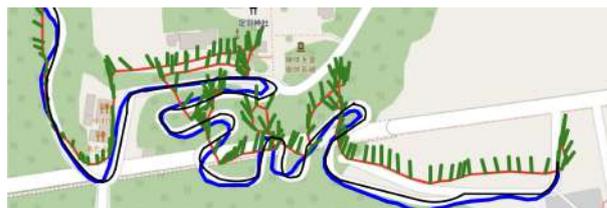


図 6: 実験結果考察

これは UKF の予測更新が不十分であり、実際の進行方向の変化を正確に反映できなかったことを示している。つまり、式 1 状態遷移方程式にある  $f()$  は正しく定義されていなかったことが分かった。具体的に正しく定義されていなかったのは、角速度の更新が行われなかったというところである。現在時刻の位置推定を行うためには、前の時刻の向き（角度）から角速度の差分を追加する必要があったが、 $f()$  関数では前の時刻の向きを参照しなかった。この向き更新の欠如は、特にカーブの多い道での経路のずれを引き起こし、位置推定精度の低下に直結する。したがって、今後は向きの更新に関するモデルの改善が必要であり、例えば、IMU の角速度データをより効果的に活用するアルゴリズムの導入が考えられる。

## 6. まとめと今後の課題

本研究では、GNSS と IMU を融合させた非線形変換フィルターを用い、リアルタイムかつ高精度な位置推定システムの実現を目指した。実験の結果、直線区間においては、GNSS と IMU を統合した UKF モデルが GNSS 単独とほぼ同等の精度を達成することが確認された。しかし、カーブ区間では UKF モデルの精度が大幅に低下し、場合によっては GNSS 単独の観測を下回ることも観測された。この精度低下の原因の一つとして、車両の向き（角度）の更新が不十分であることが挙げられる。この結果から、GNSS と IMU を統合した UKF モデルは直線区間で一定の有効性を示した一方で、カーブ区間での精度向上には改善が必要であると結論付けられる。

今後の課題として、いくつかのポイントが挙げられる。まず、車両の向き（角度）を適切に更新するために、角速度データをより効果的に活用する方法を検討する必要がある。また、アルゴリズムの計算効率を向上させることも重要である。さらに、GNSS 単独を上回る精度が求められる条件を補足し、それに基づく改善を進める必要がある。加えて、センサーデータの前処理方法の見直しや、タイムラグを補正する仕組みを導入することも重要な課題である。最後に、実験条件を多様化し、さらに詳細な検証を行うことで、モデルの信頼性と適用範囲を拡大することが求められる。

## 参考文献

- [1] Veton Hamza, Bojan Stopar, Oskar Sterle, and Polona Pavlovčič-Prešeren. Observations and positioning quality of low-cost gnss receivers: a review. *GPS solutions*, Vol. 28, No. 3, p. 149, 2024.
- [2] Elliott D Kaplan and Christopher Hegarty. *Understanding GPS/GNSS: principles and applications*. Artech house, 2017.
- [3] Qiuzhao Zhang, Longqiang Zhang, Ao Sun, Xiaolin Meng, Dongsheng Zhao, and Craig Hancock. Gnss carrier-phase multipath modeling and correction: A review and prospect of data processing methods. *Remote Sensing*, Vol. 16, No. 1, p. 189, 2024.
- [4] Norhafizan Ahmad, Raja Ariffin Raja Ghazilla, Nazirah M Khairi, and Vijayabaskar Kasi. Reviews on various inertial measurement unit (imu) sensor applications. *International Journal of Signal Processing Systems*, Vol. 1, No. 2, pp. 256–262, 2013.
- [5] Oliver J Woodman. An introduction to inertial navigation. Technical report, University of Cambridge, Computer Laboratory, 2007.
- [6] Marcos R Fernandes, Giorgio M Magalhães, Yusef Rafael Cáceres Zúñiga, and João BR do Val. Gnss/mems-ins integration for drone navigation using ekf on lie groups. *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 59, No. 6, pp. 7395–7408, 2023.
- [7] Google colab. <https://colab.google/>. Accessed: 2024-11-22.