

MRTKLIB の紹介

GNSS 測位エコシステムの統合を目指して

塩野 隼人

東京海洋大学 大学院 博士後期課程

測位航法学会 (IPNTJ) 2026 年全国大会 @ TUMSAT · 2026/05/13

本日の講演内容

- 1 なぜ今 MRTKLIB か (12分)**
RTKLIB の功績と、MRTKLIB の取り組み
- 2 MRTKLIB の中核機能 (22分)**
都市走行のベンチマーク、CLAS リアルタイム測位、小型 CLAS 端末の作り方
- 3 MRTKLIB が広げる世界 (6分)**
太陽活動極大期における CLAS の性能評価、CLAS の脅威と対策、GNSS 代替技術との接続
- 4 今後とコミュニティへの招待 (10分)**
今後の開発ロードマップ、参加・貢献の仕方

※ 最後の 10 分は質疑応答に充てたいと思います

0. そもそも MRTKLIB とは何か？

GNSS測位エコシステムの統合を目指して - Modernized RTKLIB

柱 1. ALL-IN-ONE 統合

QZSS をもっと身近に

- MADOCA-PPP と CLAS PPP-RTK を 1 つで解析
- demo5 RTKLIB の都市・移動体向け改善も統合
- パラメータ切替だけで RTK / PPP / PPP-RTK を比較可能

柱 2. メンテナンス性と DX

ユーザ負担の最小化

- TOML 設定 - 可読性・認知負荷を改善
- テスト駆動 (CI/CD) - 60+ 回帰テストで品質保証
- Issue テンプレート - 参加しやすい窓口
- Docker + Web UI - OS・build に依存しない

RTKLIBの偉大なアルゴリズム資産を受け継ぎつつ、現代のソフトウェア・エンジニアリングのベストプラクティスで再構築したOSS

1.1. RTKLIB の功績

GNSS オープン測位ソフトウェアのデファクトスタンダード

Google Scholar 引用数 

1,000+

2026/03 時点

GitHub Stars 

3,000+

Fork 1,800+

初回リリース  17

2006


約20年の歴史

国内測位インフラの基盤

- **CLASLIB** (CAO/QSS) – CLAS PPP-RTK の参考実装
- **MADOCALIB** (CAO/QSS) – MADOCA-PPP の参考実装
- **MALIB** (JAXA/SNU) – MADOCA-PPP の独立実装
- **GSILIB** (国土地理院) – マルチ GNSS 高精度測位解析

日本の主要な高精度測位ライブラリはいずれも RTKLIB 派生

各地域 GNSS の研究基盤として

- **GSDC 2023-2024 (Google):**
Motooka, *ION GNSS+ 2024* – RTKLIB 2.4.3 を拡張した GNSS/INS Tightly-Coupled 統合手法 (1位 )
- **Galileo HAS (欧州)**
Prol et al., *GPS Solutions*, 2024 – HAS のオープン実証 (EU PAULA プロジェクト)
- **BDS PPP-B2b (中国)**
Liu et al., *Meas. Sci. Technol.*, **2026/04** – B2b デコーダを RTKLIB 上に実装

1.2. メンテナンスの現状

本家 RTKLIB は 5 年以上更新が止まっている

客観データ

- 最新リリース: 2.4.3 b34
- 最終更新: 2020-12-30
- 経過期間: 5年以上

2020年末以降、多くの Issue が Open のまま未対応

github.com/tomojitakasu/RTKLIB/releases

MRTKLIB の公開



モダンな次世代RTKLIB「MRTKLIB」の公開

Zenn, 2026/03/12

高須先生より (2026/04)

MRTKLIB 公開について

応援しています。着々と開発が進んでいるみたいで今後が楽しみです。

「メンテナンスの課題と分断されたエコシステム」について

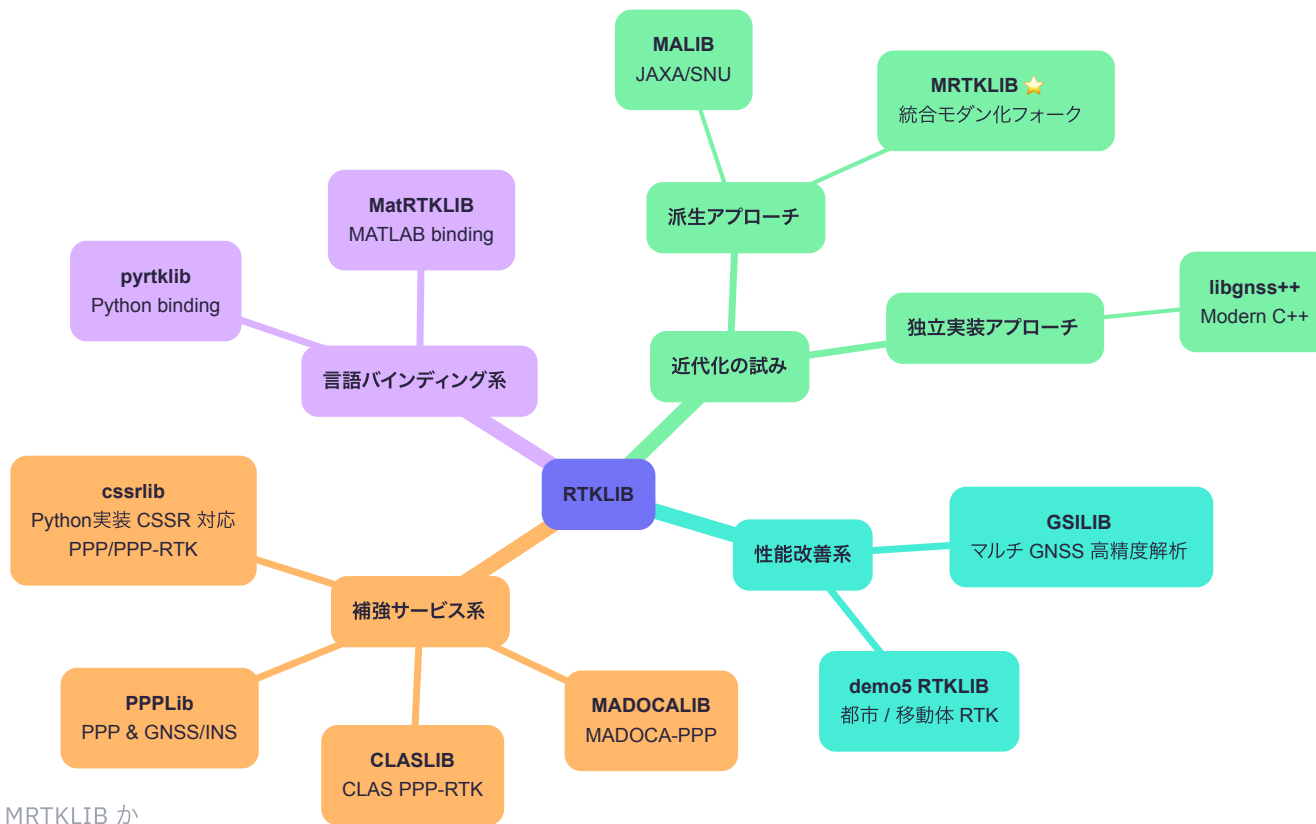
同意。私自身は既に匙を投げています。

ずいぶん前から開発・維持を引き継いで強気にまとめる人を待望しています。

gpspp.sakura.ne.jp/diary202604.htm

1.3 分断されたエコシステム

様々な派生プロジェクトが独立に並立



1.4. 断片化が招く問題

ユーザ視点で見る分断のコスト

シナリオ A

「自分のアプリに最適な測位方式
はどれか?」

RTK / PPP-RTK / PPP を比較したい

今は demo5・CLASLIB・MADOCALIB を個別ビルド、設定形式も CLI もバラバラ → 横比較困難

シナリオ B

「都市部で CLAS を試したい」

demo5 の都市走行改善 + CLASLIB の補正処理、両方を使いたい

別フォーク → 手動マージ / C コード改修が必要 → 実装・検証コストが膨大

シナリオ C

「Python から最新アルゴリズムを叩きたい」

pyrtklib / MatRTKLIB を使って研究したい

本家 (5年前) ベース → demo5 や CLASLIB の改善は含まれない → 言語かアルゴリズムのどちらかを諦める

分断は 開発者の問題 ではなく、ユーザの機会損失 である

1.5. MRTKLIB の設計思想

「強力にまとめる」ための 2 つの柱

柱 1. ALL-IN-ONE 統合

QZSS をもっと身近に

- MADOCA-PPP と CLAS PPP-RTK を 1 つで解析
- `demo5 RTKLIB` の都市・移動体向け改善も統合
- パラメータ切替だけで RTK / PPP / PPP-RTK を比較可能

1.4 のシナリオ A/B を正面から解決

柱 2. メンテナンス性と DX

ユーザ負担の最小化

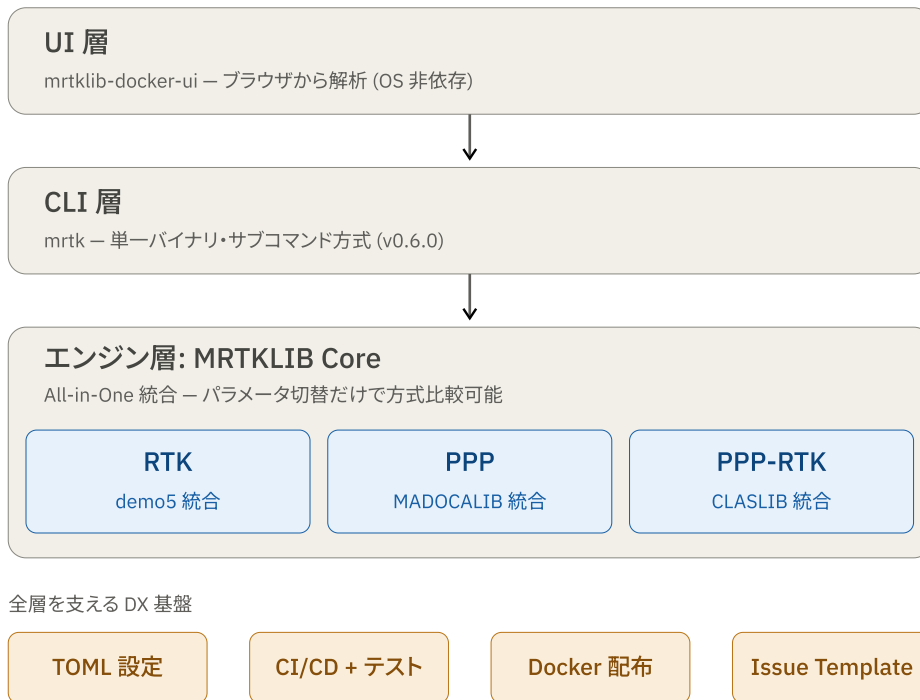
- TOML 設定 — 可読性・認知負荷を改善
- テスト駆動 (CI/CD) — 60+ 回帰テストで品質保証
- Issue テンプレート — 参加しやすい窓口
- Docker + Web UI — OS・build に依存しない

コミュニティ拡大・持続的なメンテナンスのための設計判断

技術基盤としての C11 / POSIX 化・脱 WIN32 がこの 2 本柱を支える

1.6. MRTKLIB のアーキテクチャ

エンジン / CLI / UI の分離 + DX 基盤



1.7. ここまでのまとめ

1

RTKLIB は今も GNSS 研究の基盤

Google Scholar 1,000+ 引用、各地域 GNSS の研究にも採用

2

維持・発展を担う人が求められている

本家 5 年以上更新停止、多数のフォークが並立、ユーザの機会損失

3

MRTKLIB は「強力にまとめる」一つの試み

All-in-One 統合 + DX 追求、3 層アーキテクチャで持続可能性を担保

次のセクションでは

MRTKLIB の特徴・中核機能をご紹介します

Section 2

— 中核機能

MRTKLIB で何ができるのか

2.1

都市走行ベンチマーク
demo5 統合による堅牢化

2.2

CLAS リアルタイム測位
OSS 初のリアルタイム対応

2.3

小型 CLAS 端末構築
コミュニティへの呼びかけ

2.1a. 移動体ベンチマーク環境

測位航法学会「高精度測位チャレンジ 2024」データセットで検証

データセット: PPC-Dataset

- 東京・名古屋で収録した 6 走行 × 約 30 分
- Applanix POS/LV のリファレンス
- GNSS + IMU 両方を収録
(GNSS/INS 検証にも活用可)

github.com/taroz/PPC-Dataset
千葉工大 鈴木太郎先生による公開

簡単に実行可能

- 準備 + 実行の 2 ステップで、
RTK / PPP / PPP-RTK を一括比較

```
# Step 1: L6 ファイルのダウンロード
cd scripts/benchmark
python download_l6.py --mode both
# Step 2: ベンチマークの実行
python run_benchmark.py
```

- バージョン毎に FIX 率・精度を記録・公開



ベンチマーク詳細ページ – 研究者・エンジニアが自分で確かめられる

h-shiono.github.io/MRTKLIB/reference/benchmark/ – 全バージョンの結果・アルゴリズム変更・設定・既知の制限

2.1b. demo5 統合による堅牢化

v0.3.3 (RTKLIB 2.4.3 b34 相当) → v0.4.0 (demo5 統合)

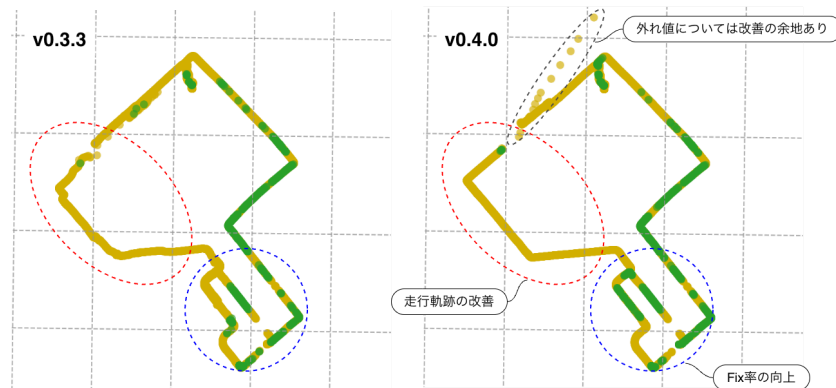
RTK: 2D RMS (FIX 時)

Case	v0.3.3	v0.4.0	備考
tokyo_run1	0.711 m	0.342 m	
tokyo_run2	17.993 m	0.095 m	189× 改善
tokyo_run3	0.293 m	0.219 m	
nagoya_run1	1.536 m	0.418 m	
nagoya_run2	1.081 m	0.596 m	FIX 率も +12 ポイント
nagoya_run3	0.307 m	0.837 m	軽微な劣化

改善の源泉 (DEMO5 由来)

- Partial AR、サイクルスリップ検知
- 全衛星系の重み付け
- ミス FIX 排除

移動体向けの特別チューニングは行っていない (= 初期値での改善)



tokyo_run2 v0.3.3 vs. v0.4.0

DEMO5 知見を PPP-RTK に適用した成果

- CLAS (PPP-RTK) でも改善
 - ・ nagoya_run2: **0.717 m** → **0.461 m** (1σ)
 - ・ tokyo_run1: **0.868 m** → **0.747 m** (RMS)

2.1c. ドキュメントの再現性

「誰でも検証できる」を支える DX 基盤

RTKLIB および派生ライブラリの課題

- PDF マニュアルのみ
- バージョン更新からの遅延
- 網羅性に欠ける項目
- 探したい情報にたどり着きにくい

結果として「コード読めば分かる」が標準に

MRTKLIB のアプローチ

- **GitHub Pages** で常時公開
- リポジトリ更新と同時に **CI/CD** で自動生成
- Doxygen による API リファレンス
- ベンチマーク結果・バージョン履歴も同時掲載

実装の最新状態とドキュメントが常に一致

手順書がメンテされている → **誰でも再現できる** を目指す

h-shiono.github.io/MRTKLIB/

2.1d. 3 方式を一括比較できる価値

Section 1.4 シナリオ A への回答

短基線 

RTK

demo5 統合済み

広域・単独 

PPP

MADOCALIB 統合済み

日本限定・高速収束 

PPP-RTK

CLASLIB 統合済み

ユーザ視点での変化:

TOML 設定の `mode` を変えるだけで 3 方式を横比較できる → **自分のユースケースに最適な方式を選ぶことが可能に**

 QZSS の高精度測位を、もっと身近に

2.2a. OSS 初の CLAS リアルタイム測位

なぜこれが意味を持つのか

従来の状況

- **CLASLIB** (公式): 参考実装のため post-processing のみ
- **商用受信機**: リアルタイム対応だがブラックボックス
- OSS で CLAS リアルタイム測位に対応する実装は見当たらない

CLAS のリアルタイム評価は商用でしか不可能だった

MRTKLIB が初めて実現

- `rtkrvcv` で **CLAS 補正** を直接適用したリアルタイム測位
- 複数の受信機にて検証済み
(u-blox ZED-F9P + NEO-D9C, Septentrio mosaic-G5)
- **FIX 率 97.7%** を達成 (v0.4.3 確認)
- **CLAS dual-channel 配信にも対応** (v0.4.4)

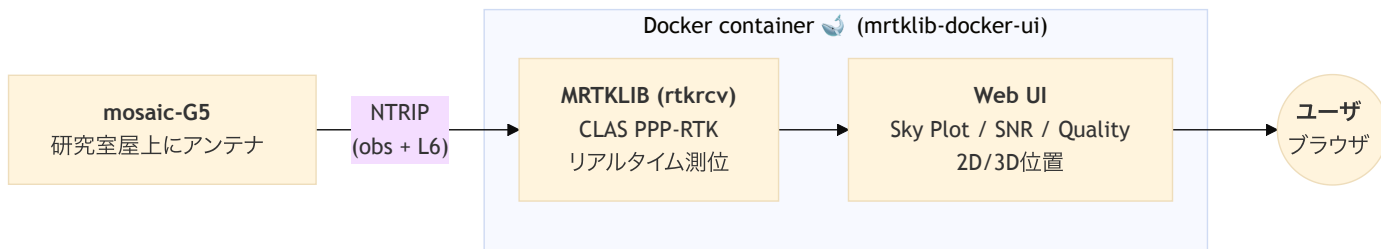
研究者・エンジニアがそのまま運用可能な形で

これが意味するもの:

研究者は自分の環境で CLAS リアルタイム測位を検証可能に。エンジニアは商用受信機に依存せず、プロトタイピングや新たなアプリケーション開発が可能に。

2.2b. デモのシステム構成

ブラウザがあれば OS を問わず操作可能 – OS・build 依存を排した設計






3 ステップで簡単実行 🚀

```
# Step 1: Docker Hub からイメージを取得
docker pull hatognss/mrtklib-docker-ui:0.2.1-alpha
# Step 2: ディレクトリを作成
mkdir -p ./workspace ./data
# Step 3: Docker を起動
docker run -d --name mrtklib-web-ui \
  -p 8080:8000 \
  -v "$(pwd)/workspace:/workspace:rw" \
  -v "$(pwd)/data:/data:ro" \
  hatognss/mrtklib-docker-ui:0.2.1-alpha
```

【ライブデモ】

CLAS リアルタイム測位を Web UI で

-  研究室ストリーム (mosaic-G5)
-  MRTKLIB Real-time (Docker)
-  Web UI (ブラウザ)

2.2d. デモから見たこと

CLAS リアルタイム測位の実力

定量的な結果

FIX 率

97.7%

v0.4.3 検証時

水平精度 (FIX 時)

～ 5 cm

静止時

初期収束時間

数十秒

Open sky

OSS として得られるもの

細かなパラメータチューニングが可能
商用はブラックボックス

任意の受信機との組み合わせ
F9P + D9C で検証、他も試行可能

プロトタイピングと研究評価の加速
CLAS の挙動分析が自由に

次は: これをさらに小型化して持ち運べる端末にしたい

2.3a. 小型 CLAS 対応端末のレシピ

MRTKLIB + 多周波受信機 + Raspberry Pi Zero 2 W

GNSS 受信機

mosaic-G5

L1+L2+L5+L6

計算機

RasPi Zero 2 W

手のひらサイズ

ソフトウェア

MRTKLIB

Docker / arm64

ポイント

- 既製ハードウェア + OSS だけで組み立て可能
- 組み立て後そのまま CLAS リアルタイム測位
- 研究・実証・教育用途に手のひらサイズで



プロトタイプ実機

(Septentrio mosaic-G5 + Raspberry Pi Zero 2 W)

詳細な配線・設定手順は順次公開予定 ⇨ 公開しました🎉 (2026-05-11)



GitHub Pages: "CLAS Positioning with Septentrio mosaic-G5", Zenn: "L6対応受信機を CLAS 測位対応にする - MRTKLIB v0.6.5 と小型端末レシピ", 質問・改善提案も歓迎 🙌

2.3b. 小型 CLAS 対応端末プロトタイプ

実機動作確認の様子

The screenshot displays the MRTKLIB Web UI interface. The top navigation bar includes 'Post Processing', 'Real-Time', 'Stream Server' (selected), 'Conversion', and 'Tools'. A status indicator 'API OK' is visible in the top right.

The main content area is titled 'Multi-Stream Relay CLAS Pipeline'. Under 'CLAS Pipeline', there is a description: 'Wrap mtklib relay and mtklib csst2rtcm3 into a single workflow. Forward raw SBF (with the QZSS L4 CLAS message) from the receiver, decode it to RTCM3 with csst2rtcm3, and send it back to the receiver that run VRS-RTK on its own engine.'

The 'Receiver' is set to 'Septentrio mosaic-G5'. Below this, there are configuration fields for 'Input device', 'Input baud', 'Output device', and 'Output baud', all set to '/dev/serial/by-id/usb-Septentrio_Septentrio_USB_Device_0100012430-002' and '115200' respectively.

The 'Record raw SBF' option is checked, with a note: 'Save the SBF stream to /workspace for post-mortem analysis'. The 'SBF file path' is set to '/workspace/CLAS/sbf_stream_000005_00f'.

A 'Start pipeline' button is visible at the bottom right of the configuration section.

On the right side, there is a performance summary table:

Throughput	SBF blocks/s	Total bytes	Total blocks	Last SBF	Last PVT
11.1 KIB/s	84.8	972.48 MiB	7331901	25 s ago	25 s ago

Below the table is a graph titled 'Position (from SBF PVTGeodetic)'. The graph shows a 2D plot of Position (m) on the Y-axis (ranging from -10 to 10) versus East (m) on the X-axis (ranging from -10 to 10). A small cluster of points is visible near the origin (0,0).

At the bottom right, there is a terminal window showing the output of the 'mrtk relay' process. The output includes a list of log messages with timestamps, process IDs, and status information, such as '2026/09/11 23:02:29 [LCL] mtkrelay # 89488 bps (0) /dev/ttyACM1 (1) 2 clients' and 'mrtk relay mtklib csst2rtcm3'. The process ends with 'mrtklib: Stopping process...' and 'stream server stop'.



10時間以上の安定動作を確認 (CPU使用率 ≈ 8%/core)

今後は測位精度の改善に取り組みます (Issue #97, #98)

2.3c. 2つの動作パターンとコミュニティへの呼びかけ

パターン A

CSSR → RTCM3 変換

受信機内蔵 RTK エンジンで処理

MRTKLIB = 補正変換のみ

パターン B

Raw data → MRTKLIB CLAS

MRTKLIB 側で PPP-RTK 測位

受信機 = 観測のみ

コミュニティへの呼びかけ

さまざまな受信機で試す

F9P / X20P, mosaic-X5, UM980, LG290P, M20 など

バグ報告 (特に Web UI)

mrtklib-docker-ui は alpha — 一番助かります

さまざまな環境で試す

Mac / Windows / Linux, Raspberry Pi 他

機能要望・UI 改善

Issue / Discussion 何でも歓迎

低価格受信機での再現は特に期待 — よりアクセスしやすい CLAS 端末へ

Section 3

— MRTKLIB が広げる世界

中核機能の先にある、研究・産業・未来への接続

3.1

CLAS 性能評価
使ってみる動機を提供

3.2

CSSR Spoofing と CMA
脅威と対策の体系化

3.3

Alt PNT / SoOp
GNSS Denied への備え

3.1. CLAS の本当の性能を明らかにする

Solar Cycle 25 極大期における QZSS CLAS 診断

🤔 CLAS の「実運用での性能」は？ 太陽活動の影響はどの程度あるの？

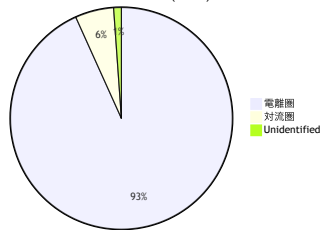
網羅的な診断論文

- Solar Cycle 25 極大期 を含む 4 年以上 (2021~2025) のデータを解析
- GEONETの約 1,300 局を用い、CLAS の真の実力と課題を明らかに

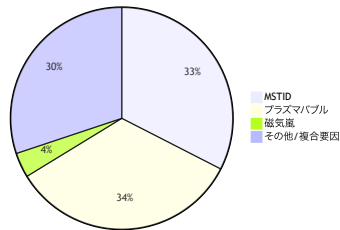
主要な発見

- 93% の精度劣化は電離圏擾乱が原因
- そのうち 37% は補正困難な電離圏擾乱が原因
(プラズマバブルと磁気嵐 - シンチレーションを起こす電離圏擾乱)

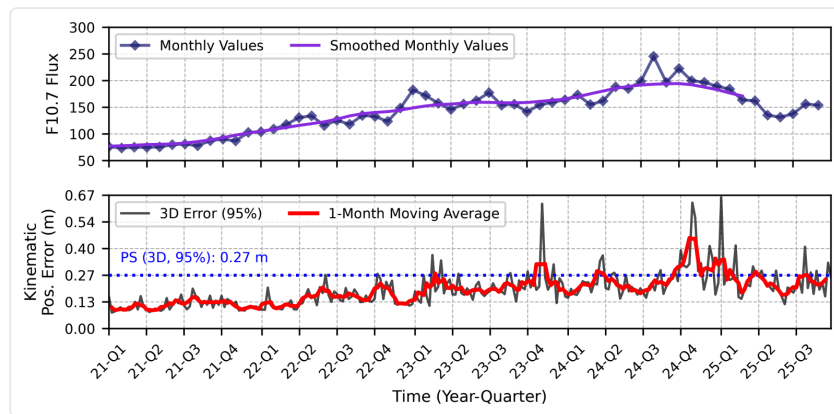
精度劣化要因の分類 (全体)



電離圏擾乱の内訳



精度劣化 VS 太陽活動



太陽フラックス (F10.7) と Kinematic 測位精度 (3D 95%) の対応

Navigating the Storm: A Diagnostic Analysis of QZSS CLAS Performance and Vulnerabilities Through the Solar Cycle 25 Maximum — 共著: 久保信明
ION GNSS+ 2025 → NAVIGATION, DOI: 10.33012/navi.762

3.2. CLAS の脅威と対策

補正情報への攻撃という新しいリスク

🤔 補強情報の安全性は担保されているの？

CLAS STEALTH SPOOFING

- GNSS の欺瞞が社会問題化
- 補強情報への脅威を CLAS を例に 初めて体系的に整理

主要な貢献

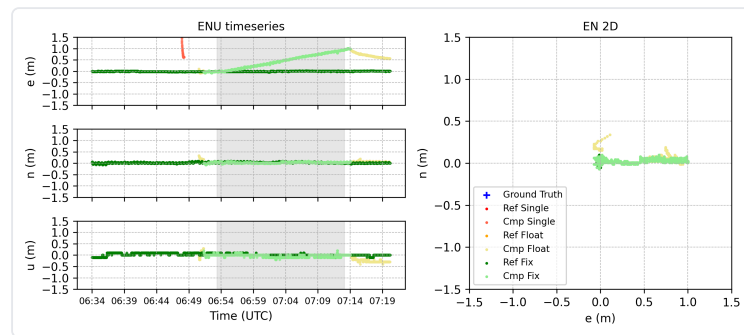
- 脅威モデルの分析と攻撃手法の体系化
- CLASLIB を使った後処理で攻撃方法を検証
- **u-blox ZED-F9P** を使った実機検証で脅威を実証
- **FIX 解を保ったまま = ユーザに気づかれずに、位置を 1 m ずらすことに成功**

有効な対応策

- 補正情報の認証 🛡️ **Correction Message Authentication (CMA)** が計画中

MRTKLIB は CLASLIB の更新に追従して CMA を速やかにサポート予定

U-BLOX ZED-F9P + NEO-D9C を使ったリアルタイム実証



CLAS リファレンス (緑) と CLAS Spoofing (薄緑) の比較

Stealth Spoofing of the CSSR Open Standard: A First Demonstration with QZSS CLAS — 共著: 久保信明
Pacific PNT 2026 — **Best Student Paper Award** 🏆

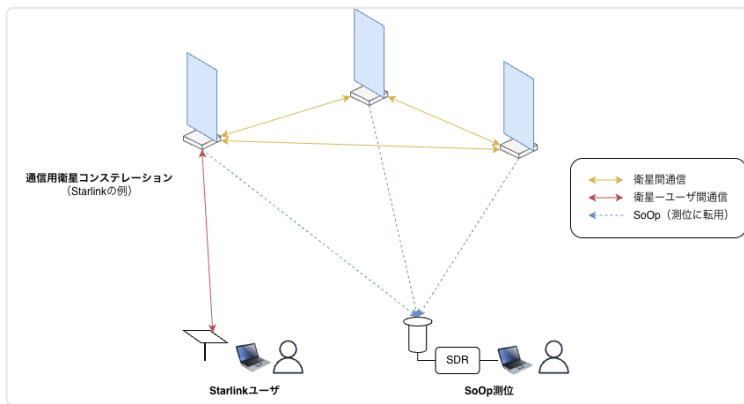
3.3. GNSS が使えない世界への備え

Signals of Opportunity (SoOp) 測位への対応

😞 GNSSが使えない時に、位置を知る手段はあるの？

背景

- GNSS に対する脅威の増大により、GNSS が使えない場合への備えが急務
 - ➡ 代替 PNT (**Alternate PNT**) の研究が加速
- **SoOp** (Signals of Opportunity):
LEO 衛星の通信信号など、GNSS 以外の信号から測位



SOOP における課題

- 信号仕様が非公開なため、リバースエンジニアリングが必要
- 帯域によっては高価な機材が必要
- 各研究機関が各々に取り組んでおり、**共通 SW が不在**

MRTKLIB の役割

- **共通フォーマット**を設計 (SOOPEX v0.1)
- SoOp 観測量を既存 **RINEX** ワークフローに統合
- MRTKLIB の測位エンジンを **SoOp** にも適用

Section 1 で示した 3 層アーキテクチャが、新しい観測量にも対応できる拡張性を担保

MRTKLIB は QZSS だけでなく
これからの PNT エコシステム全体に接続する基盤を目指す

Section 4

— 今後とコミュニティへの招待

一緒に MRTKLIB を育てていきましょう

4.1

ロードマップ
どこへ向かうか

4.2

プロジェクトへの参加・貢献の仕方
どう参加できるか

4.3

最短手順
今すぐ試す

4.1a. MRTKLIB ロードマップ

v0.6.x (現在) → これからの Major リリース



✓ 横断的な改善

- バイナリの自動リリース
- 日本語ドキュメントの整備
- 各種受信機対応拡大

👏 コミュニティへの招待

優先順位は皆さんの声で決まります – [GitHub Issues / Discussions](#) へ

4.1b. mrtdlib-docker-ui ロードマップ

現在 v0.1.0-alpha – ユーザとの接点

直近の優先課題

バグ取り

現在 alpha – 皆さんの報告が最も助かる

Stream Monitor 実装

NMEA 等のリアルタイム可視化

全機能対応

mrtd convert / mrtd dump 等を UI から

中期の計画

Gemma 4 統合

ローカル LLM による自動コンフィグ生成・解析支援

mrtd qc 連携

品質チェック結果の UI 可視化

Image 自動デプロイ強化

latestタグ打ち

Gemma 4 統合について – 現在 Claude API で実装している Config Assistant を、Google の最新オープンモデル Gemma 4 でローカル実行するよう置き換える計画。ユーザのデータとプライバシーを保ちつつ、AI による設定支援を提供します。

4.2. プロジェクトへの参加・貢献の仕方

参加しやすいコミュニティを目指して

現在の準備状況

- ✓ **Issue Template**
バグ報告・機能要望のフォーマット化
- ✓ **CONTRIBUTING.md**
参加・貢献の明文化
- ✓ **Code of Conduct**
安心して参加するための行動規範
- ✓ **CI/CD + 回帰テスト**
PR の品質担保を自動化

これから強化したいこと

- **Good first issue label**
最初の一步にぴったりの題材を明示
- **GitHub Discussion の活性化**
Issue より気軽な質問・相談の場
- **「最初の 30 分」ガイド**
fork から初 PR までの流れ
- **日本語での交流窓口**
敷居を下げる (Discussion 日本語歓迎)

Issue / PR / Discussion — どれも日本語で OK
「こう使いたい」「動かない」「こうしたい」— 何でも歓迎します

4.3. まず触ってみる最短手順

3つの入口

最速

Docker で1 コマンド

```
docker run -p 8080:8000 \  
  hatognss/mrtklib-docker  
-ui:0.2.1-alpha
```

ブラウザで `localhost:8080`

即 Web UI で試せる

手順を読みながら

GitHub Pages

```
h-shiono.github.io/MRTKLIB/  
B/
```

CLAS / MADOCA 導入ガイドあり

API リファレンスも自動更新

日本語で理解したい

Zenn 記事

```
zenn.dev/hatognss
```

MRTKLIB 概要・Web UI 紹介

GNSS エコシステムの背景



github.com/h-shiono/MRTKLIB

試して詰まったら → [GitHub Issue / Discussion](#)へ

うまく動いたら → ぜひ★を!

MRTKLIB を一緒に育てましょう

より使いやすく、持続的な
GNSS 測位エコシステムを目指して

GitHub

github.com/h-shiono/MRTKLIB

Docs

h-shiono.github.io/MRTKLIB

Zenn

zenn.dev/hatognss

ご清聴ありがとうございました

質疑応答へ