

# GNSSとIMUセンサを利用した小 型船の姿勢推定

---

東京海洋大学 情報通信工学研究室

発表者： 水谷伝

指導教員： 久保信明

# 目次

- はじめに
- 統合方法
  - IMU角速度から得られる水平姿勢
  - IMU加速度から得られる水平姿勢
  - IMU角速度、加速度のカルマンフィルタを用いた統合
  - IMUとGNSSとの相補フィルタを用いた統合
- まとめ

# はじめに

- GNSS コンパスと IMUの角速度、加速度 を利用して小型船舶の姿勢を推定する。
- LiDARによる3Dマップの作成には計測した船舶のできる限り正確な姿勢の情報が必要不可欠である。
- IMUだけを用いた姿勢、方位の推定は時間経過とともに誤差が蓄積される。
- GNSSコンパスを用いた姿勢、方位は長期的には正しいが細かい誤差が多い。
- IMUとGNSSコンパスのいいとこどりをを行う。

芝浦工業大学の研究室にてLiDARを用い作成した動画

# はじめに

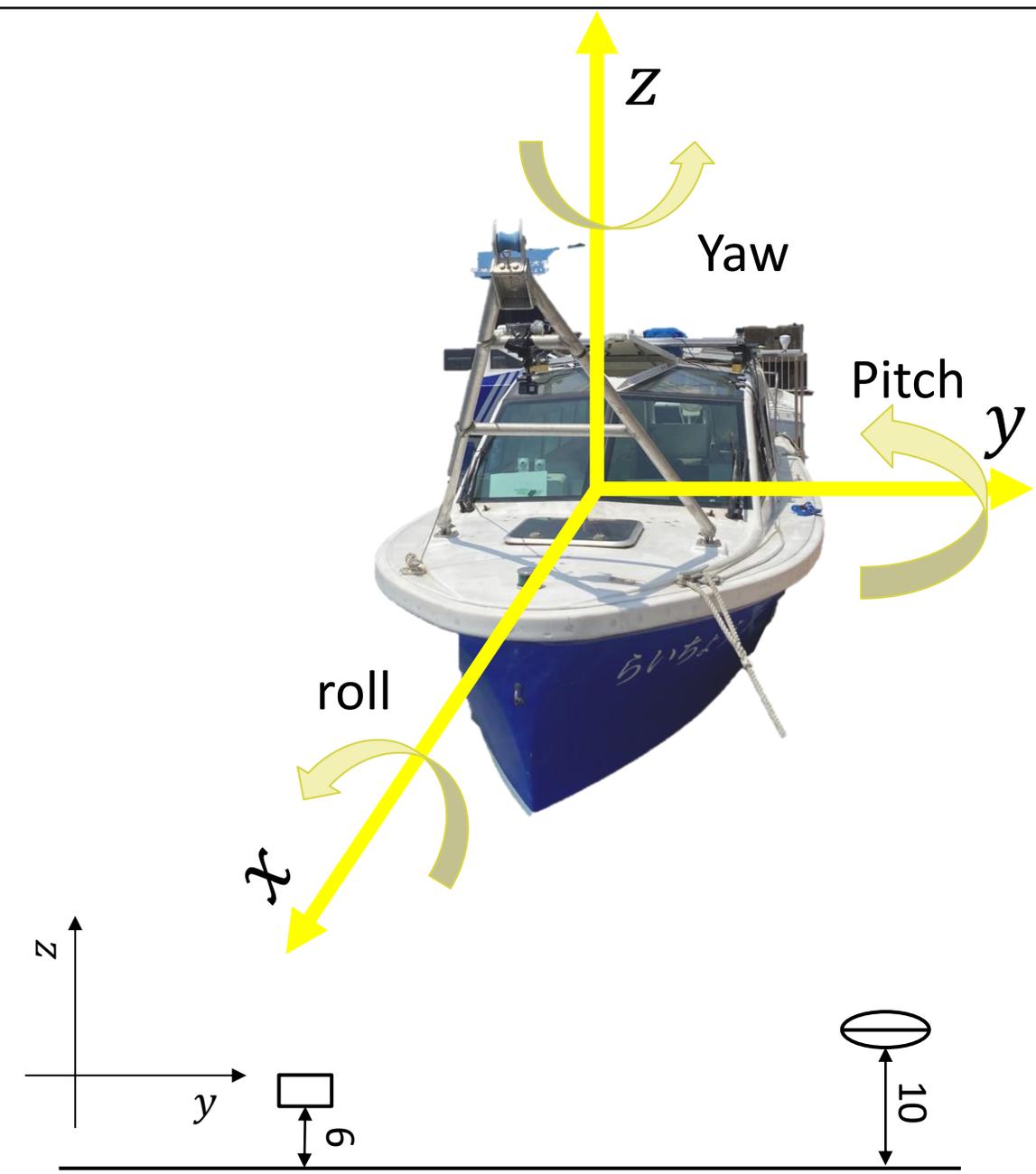
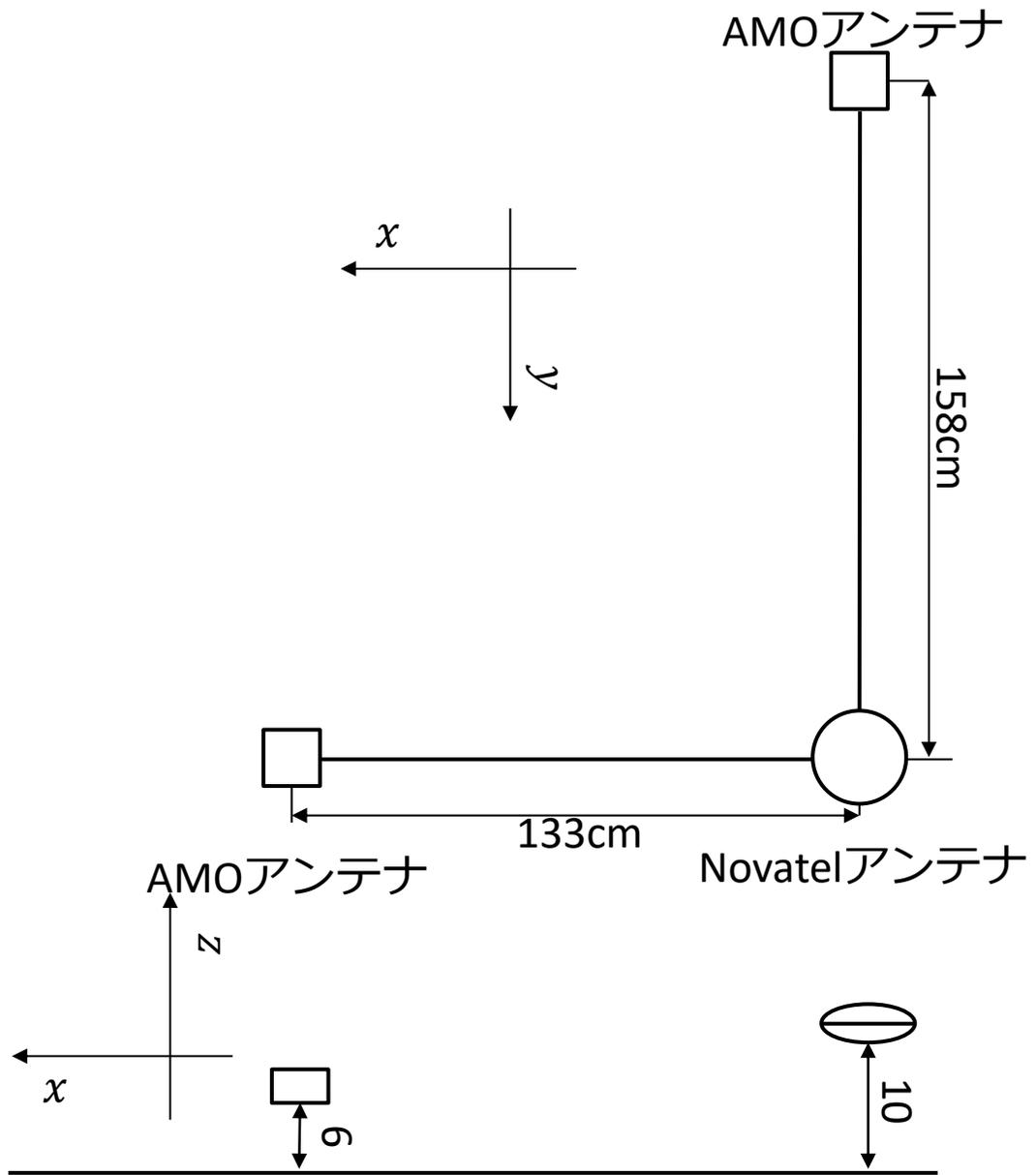
航行ルート



○にアンテナを設置



今回のデータは7月19日に本学越中島キャンパスのポンド⇒レインボーブリッジ⇒晴海ふ頭⇒本学越中島キャンパスを航行したものである。



# IMU角速度 算出方法

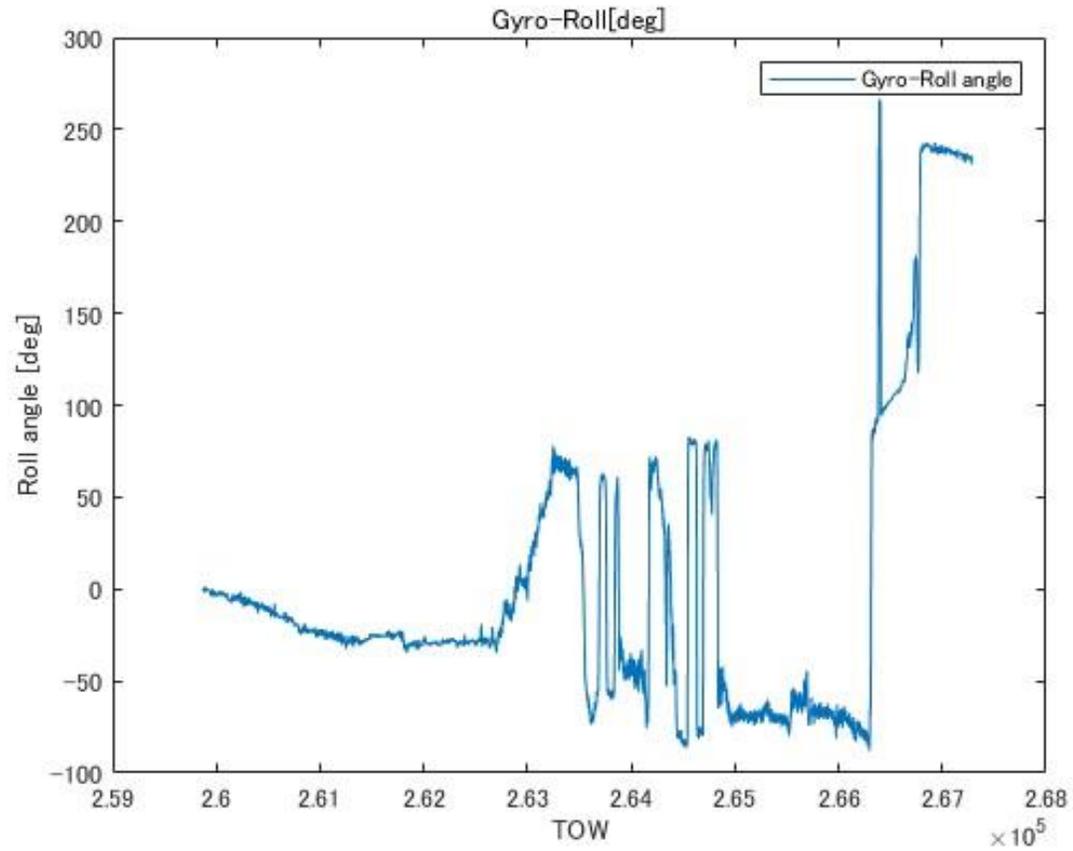
- IMU角速度センサから測定された角速度( $p, q, r$ )はオイラー角( $\varphi, \theta, \psi$ )ではないため、測定値をオイラー角の変化率( $\dot{\varphi}, \dot{\theta}, \dot{\psi}$ )に下記の式(1.1)を用いて変換しそれらを積分することで求める。

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\varphi \tan\theta & \cos\varphi \tan\theta \\ 0 & \cos\theta & -\sin\varphi \\ 0 & \sin\varphi / \cos\theta & \cos\varphi / \cos\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (1.1)$$

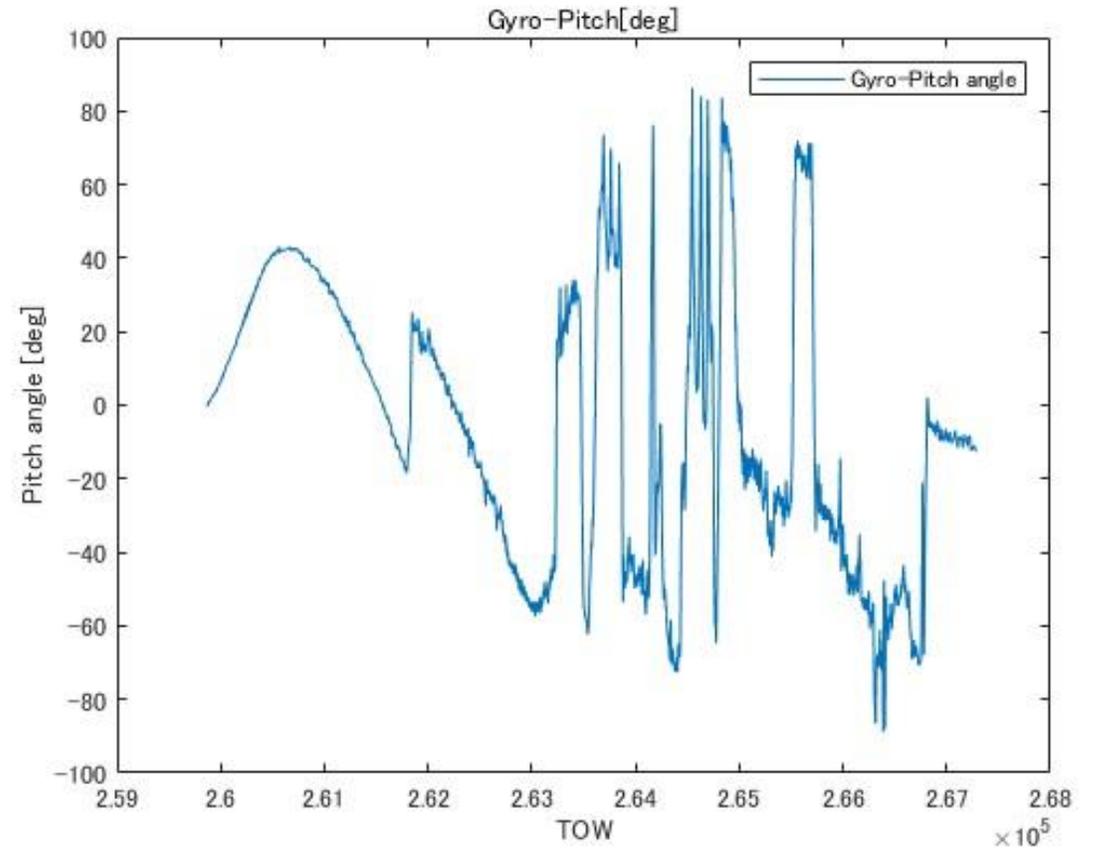
- 角速度からの測定値には誤差があり、積分することにより蓄積される。
- この方法は非常に正確なセンサを用いるか、動作時間が非常に短い場合を除き姿勢を求めるに適さない。

# 角速度

## ロール



## ピッチ



# 加速度 算出方法 (1)

- 船舶が静止または等加速で運動している場合、IMU加速度センサで測定される加速度( $f_x, f_y, f_z$ ) は次のように表せる。 $g$ は重力加速度。

$$\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = g \begin{bmatrix} \sin\theta \\ -\cos\theta\sin\varphi \\ -\cos\theta\cos\varphi \end{bmatrix} \quad (1.2)$$

- 式(1.2)よりロール角(1.3)、ピッチ角(1.4)を算出する。

$$\varphi = \sin^{-1} \left( \frac{-f_y}{g\cos\theta} \right) \quad (1.3)$$

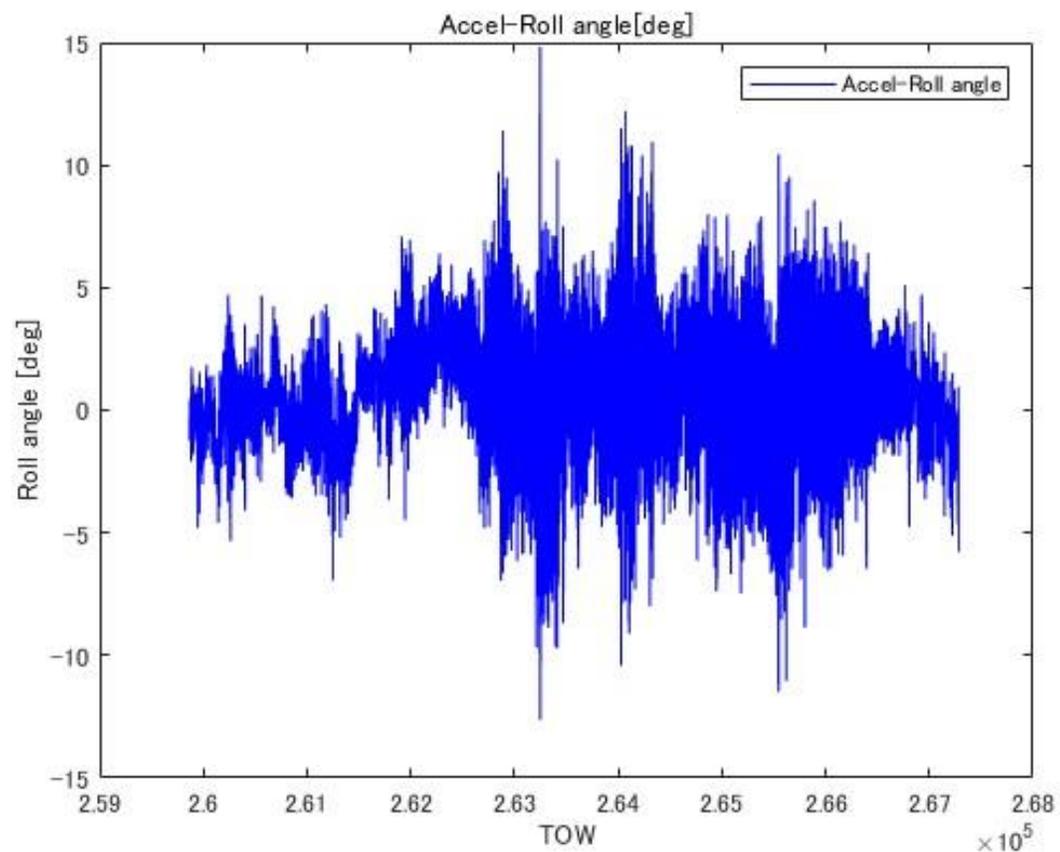
$$\theta = \sin^{-1} \left( \frac{f_x}{g} \right) \quad (1.4)$$

## 加速度 算出方法 (2)

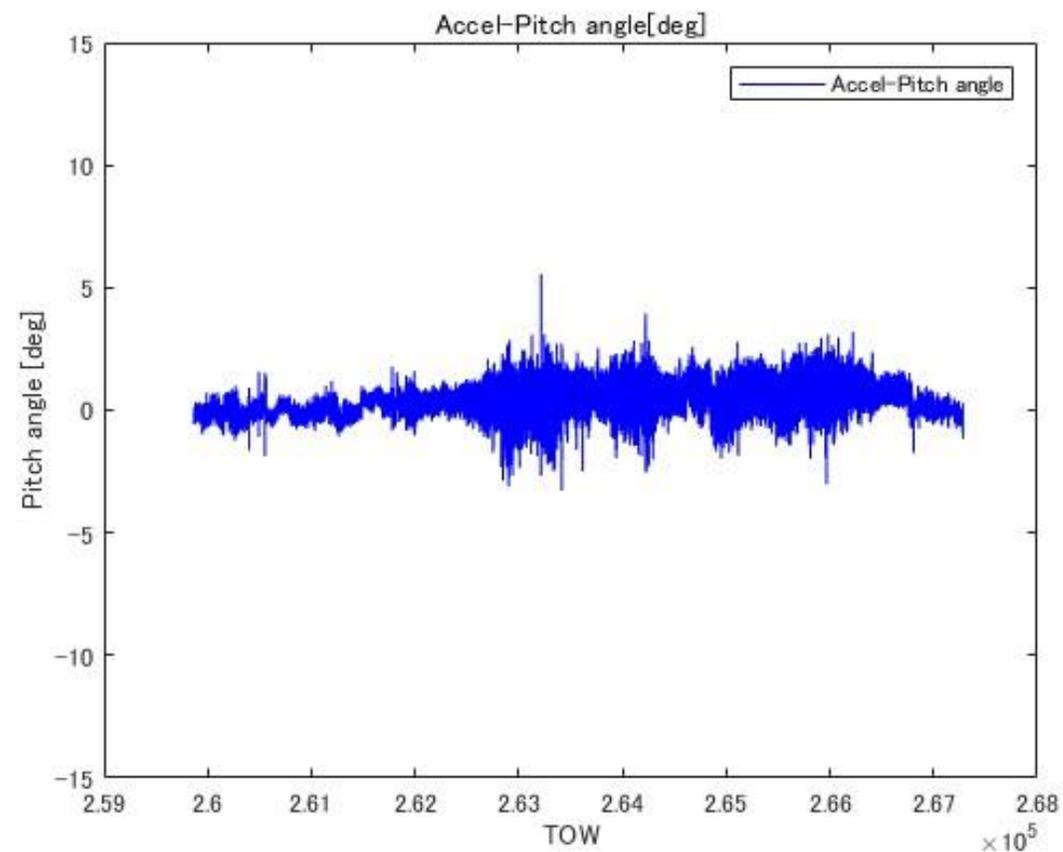
- 加速度センサによる姿勢推定は数値積分を行っていないため一定時間経過しても誤差を発散させることなく限られた範囲に保つことができる。
- しかし、船舶が高い角速度で回転している場合、また急激な速度変化がある場合では使用できず、加速度または角速度が十分に小さい場合にのみ有効な近似であるため精度があまりよくない。

# 加速度

ロール



ピッチ



# 角速度、加速度KF統合 算出方法 (1)

- 角速度、加速度センサはそれぞれ単独では欠点がある。
  - 角速度：短期的には正確であるが誤差が蓄積する。
  - 加速度：長期的には正確であるが瞬間的な誤差が大きい。
- この2つのセンサをカルマンフィルタで統合し、それぞれの長所を活かし短所を補完することによって正確な姿勢推定を行う。
- 加速度センサから得られた姿勢をカルマンフィルタの測定値として扱う。
- 式(1.1)をシステムモデルとして使用したいが次の形式(1.5)で表すことができない。

$$x_{k+1} = Ax_k + w_k(1.5)$$

# 角速度、加速度KF統合 算出方法 (2)

- 行列からオイラー角を抽出する方法がないため、代わりに線形カルマンフィルタを利用するため状態変数として四元数を使用する。
- 下記の式(1.6)を用いて角速度を四元数の変化率に変換する。

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \quad (1.6)$$

- 式(1.6)を離散的な表現すると次のようなシステムモデルが得られる。

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix}_{k+1} = \left( I + \Delta t \cdot \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \right) \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}_k \quad (1.7)$$

システムモデルの行列Aとする

# 角速度、加速度KF統合 算出方法 (3)

- 行列Aは定数ではなく、角速度に応じて変化する変数となる。
- 状態変数は四元数であるため加速度から得たオイラー角を四元数に変換する必要がある。オイラー角から四元数への変換式は次式(1.8)で表される。

$$z = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} \cos \frac{\varphi}{2} \cos \frac{\theta}{2} \cos \frac{\psi}{2} + \sin \frac{\varphi}{2} \sin \frac{\theta}{2} \sin \frac{\psi}{2} \\ \sin \frac{\varphi}{2} \cos \frac{\theta}{2} \cos \frac{\psi}{2} - \cos \frac{\varphi}{2} \sin \frac{\theta}{2} \sin \frac{\psi}{2} \\ \cos \frac{\varphi}{2} \sin \frac{\theta}{2} \cos \frac{\psi}{2} + \sin \frac{\varphi}{2} \cos \frac{\theta}{2} \sin \frac{\psi}{2} \\ \cos \frac{\varphi}{2} \cos \frac{\theta}{2} \sin \frac{\psi}{2} - \sin \frac{\varphi}{2} \sin \frac{\theta}{2} \cos \frac{\psi}{2} \end{bmatrix} \quad (1.8)$$

# 角速度、加速度KF統合 算出方法 (4)

- 加速度計から決定された姿勢( $\varphi, \theta, \psi$ )を先ほどの式(1.8)に代入することにより四元数の形式でカルマンフィルタの測定値( $Z_k$ )が得られる。
- 観測行列H、プロセスノイズQ、観測ノイズRを下記に設定する。

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad Q = \begin{bmatrix} 0.0001 & 0 & 0 & 0 \\ 0 & 0.0001 & 0 & 0 \\ 0 & 0 & 0.0001 & 0 \\ 0 & 0 & 0 & 0.0001 \end{bmatrix}, \quad R = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}$$

- 状態変数と誤差共分散行列の初期値を次に設定する。

$$\hat{x}_0^- = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad P_0^- = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

# 角速度、加速度KF統合 算出方法 (5)

- 時刻n-1における状態ベクトル(予測値)

$$x_{n-1} = Ax_n$$

- 共分散遷移式

$$P_{n-1} = AP_nA^T + Q$$

- カルマンゲイン

$$K = P_{n-1}H^T(HP_{n-1}H^T + R)^{-1}$$

- 状態更新式

$$x_n = Ax_n + K(z - Hx_{n-1})$$

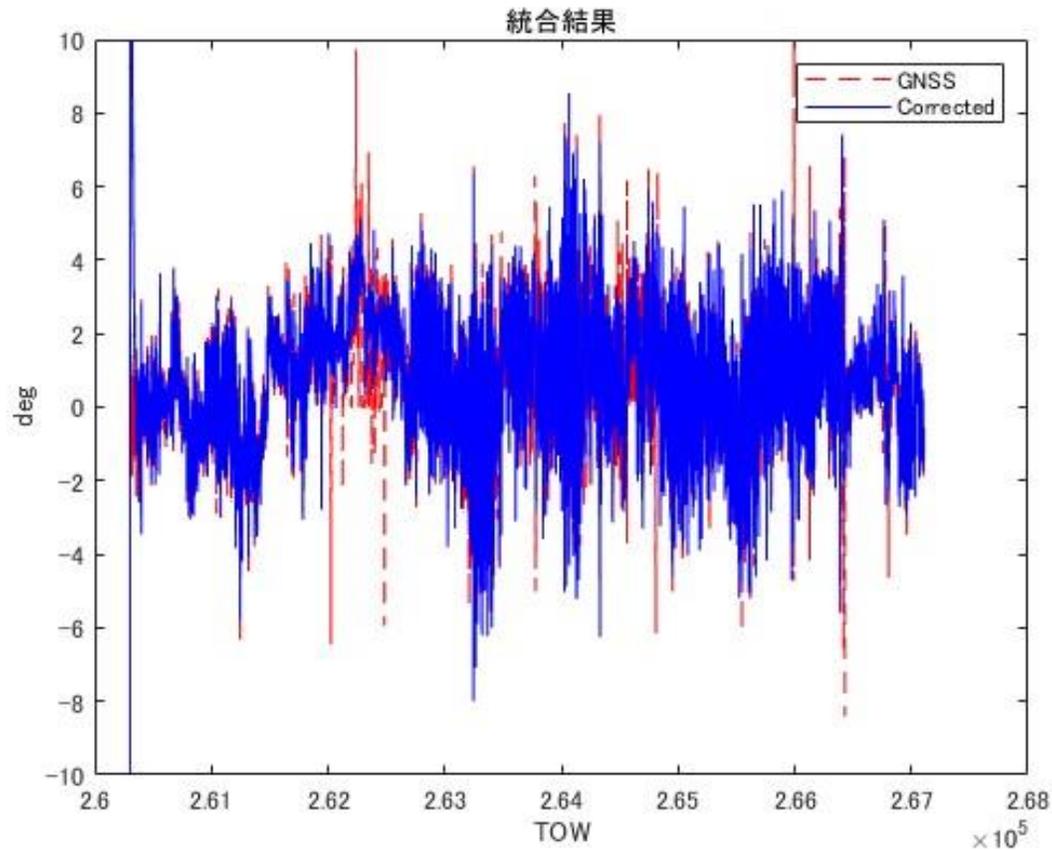
- 共分散更新式

$$P = (I - KH)P_{n-1}(I - KH)^T + KRK^T$$

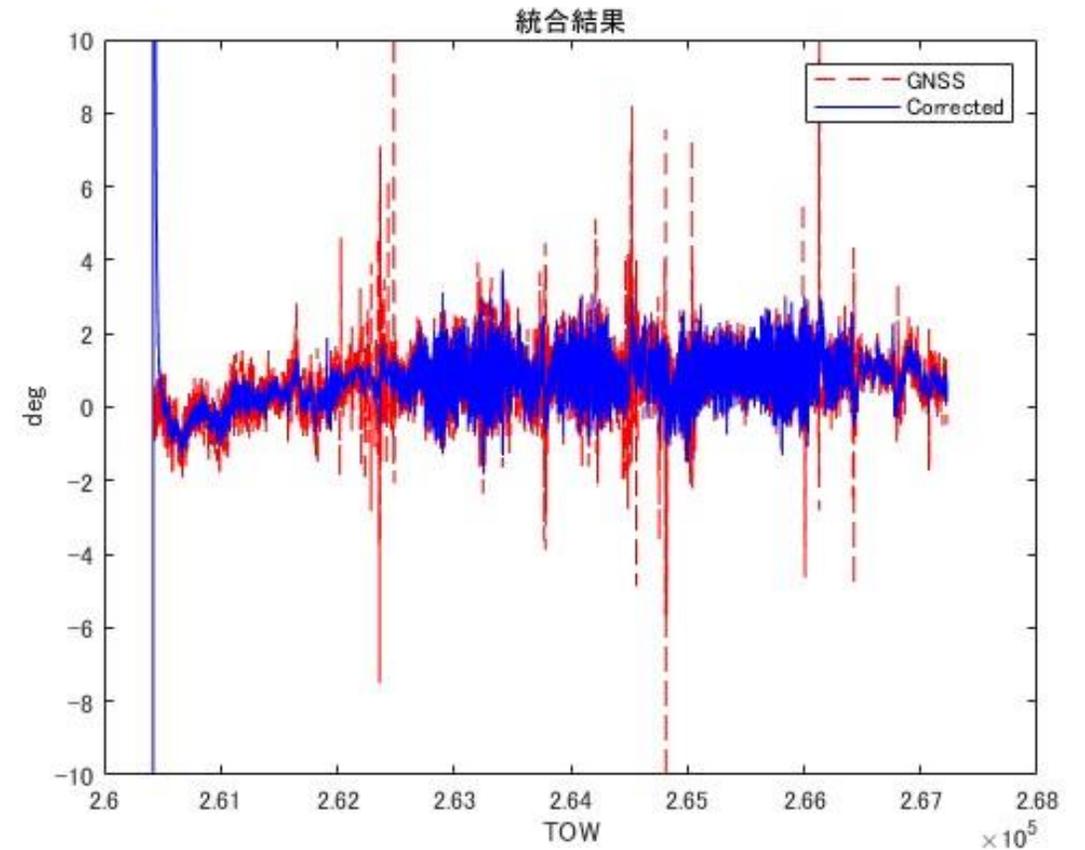
上記を用いて統合する。

# 角速度、加速度統合結果

ロール



ピッチ



# GNSS,IMU 相補フィルタ統合方法 (1)

- 相補フィルタを用いてGNSSとIMUの信号を統合することによってリアルタイムで雑音を除去することができる。
- 使用するGNSSのデータは船舶の前後、左右に設置したアンテナから左舷後方のアンテナを基準としGNSSコンパスによって得たものを用いる。
- IMUは先述KFによって統合したデータを用いる。
- 相補フィルタを使用することで非線形状態方程式を誤差システムに対して設計することにより線形カルマンフィルタで対応することが可能となる。

# GNSS,IMU 相補フィルタ統合方法 (2)

- システムモデルの行列A,Bを以下とする。

$$A = \begin{bmatrix} 0.9 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- 状態方程式

$$x(k+1) = Ax(k) + Bv(k) \quad (2.1)$$

$$x(0) = \begin{bmatrix} 0 \\ 20 \\ 0 \end{bmatrix} \quad v(k) = \begin{bmatrix} v_1(k) \\ v_2(k) \end{bmatrix}$$

- $v_1(k), v_2(k)$ はそれぞれ独立な平均値0、分散60の正規性白色雑音である。
- 相補フィルタでは二つのセンサの差分を観測値とするため観測方程式
$$y(k) = y_1(k) - y_2(k) \quad (2.2)$$
- $y_1(k), y_2(k)$ はそれぞれGNSS,IMUの観測値である。

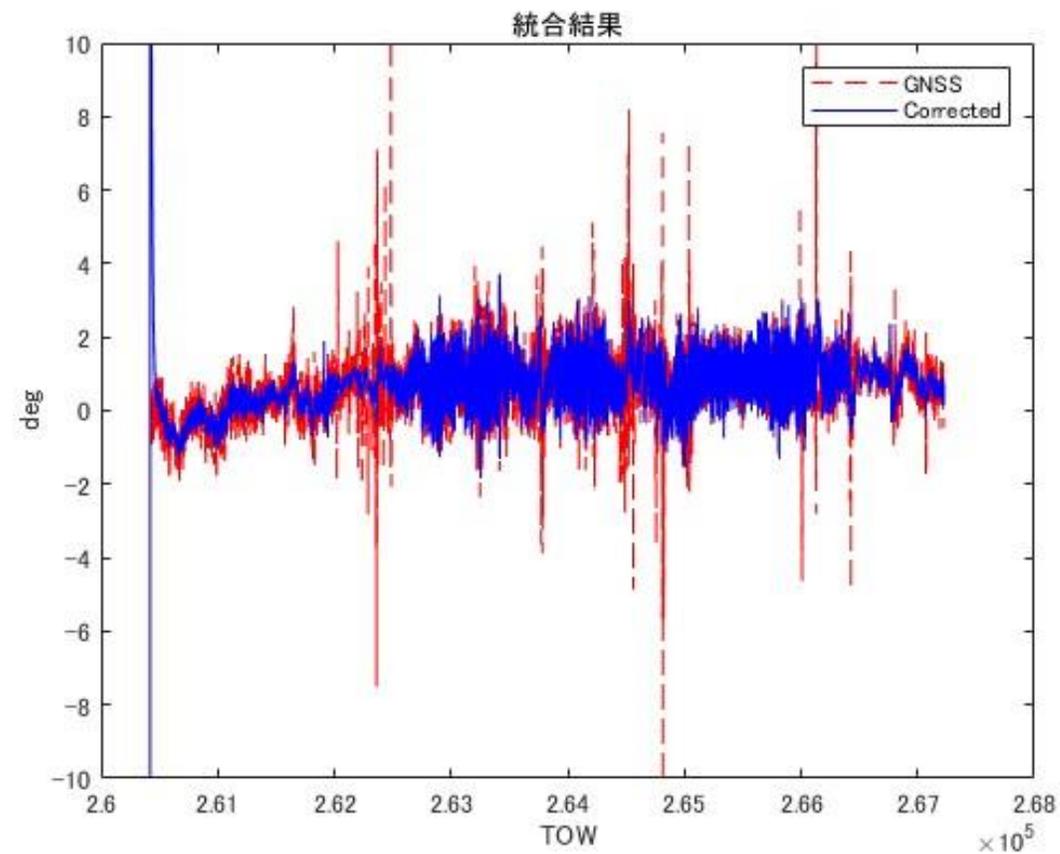
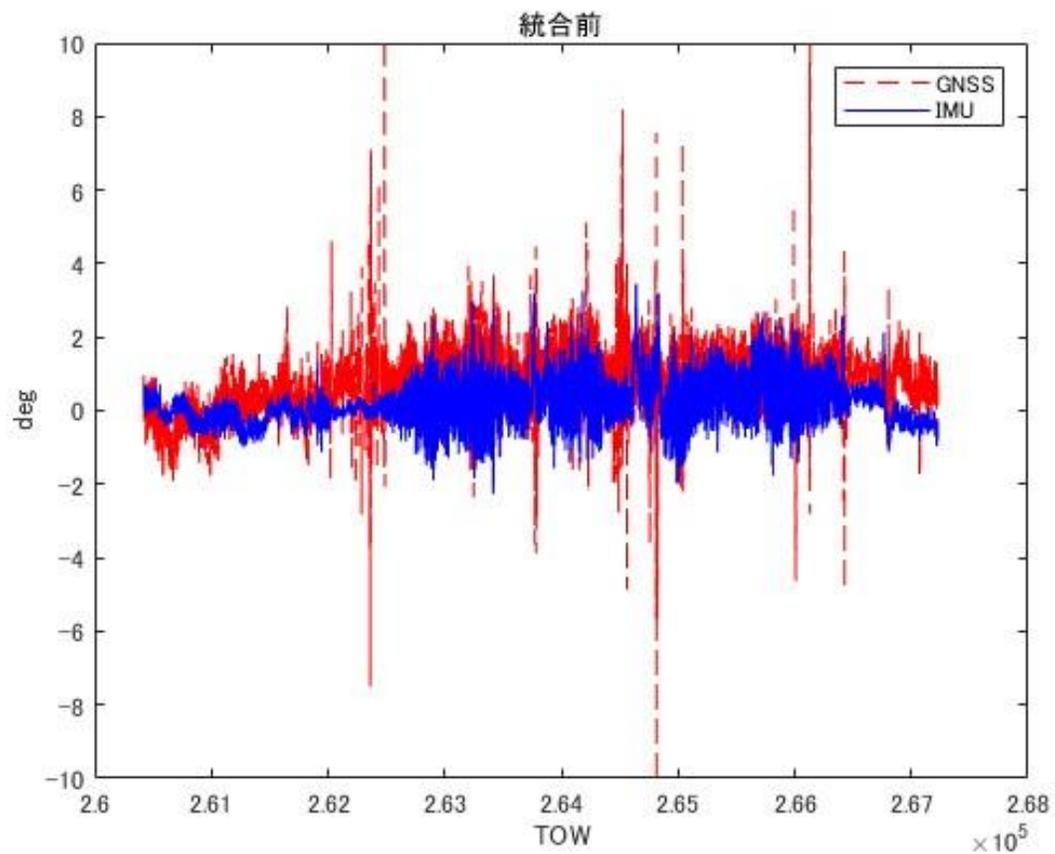
# GNSS,IMU 相補フィルタ統合方法 (3)

- 式(2.1),(2.2)から線形カルマンフィルタを用いて状態推定を行う。
- 状態推定値と共分散行列の初期値,システム雑音 $v(k)$ と観測雑音の共分散行列として

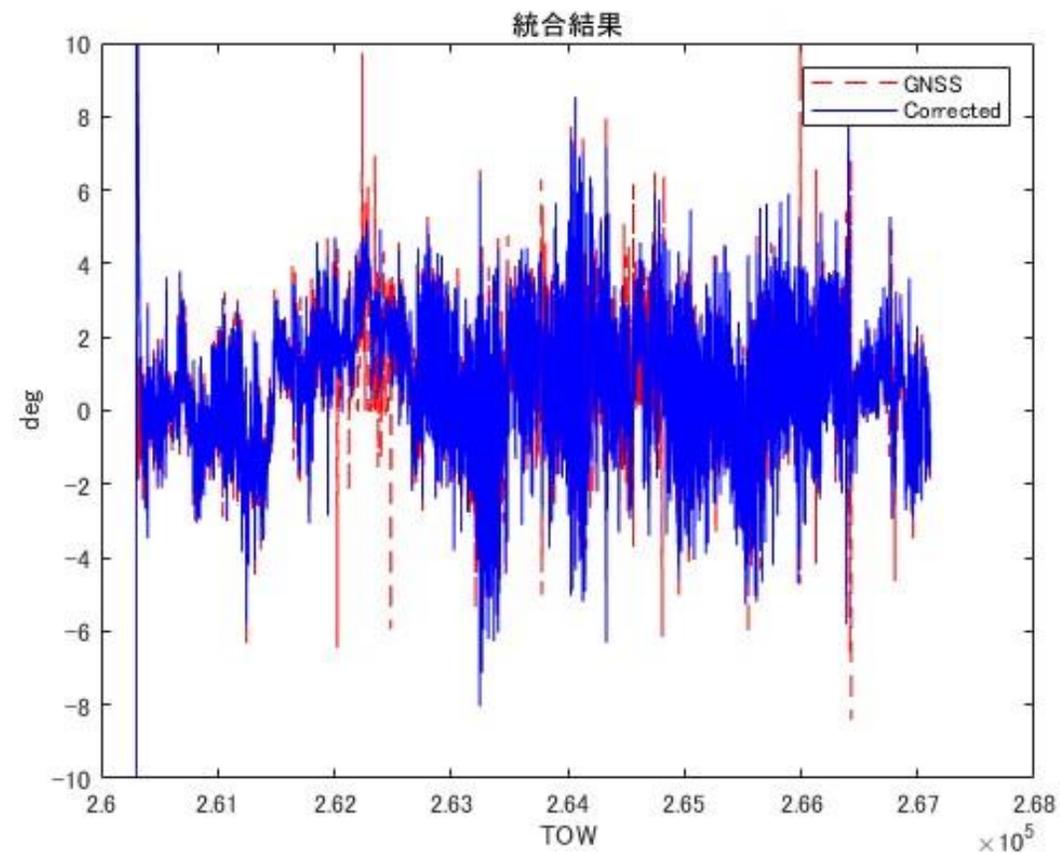
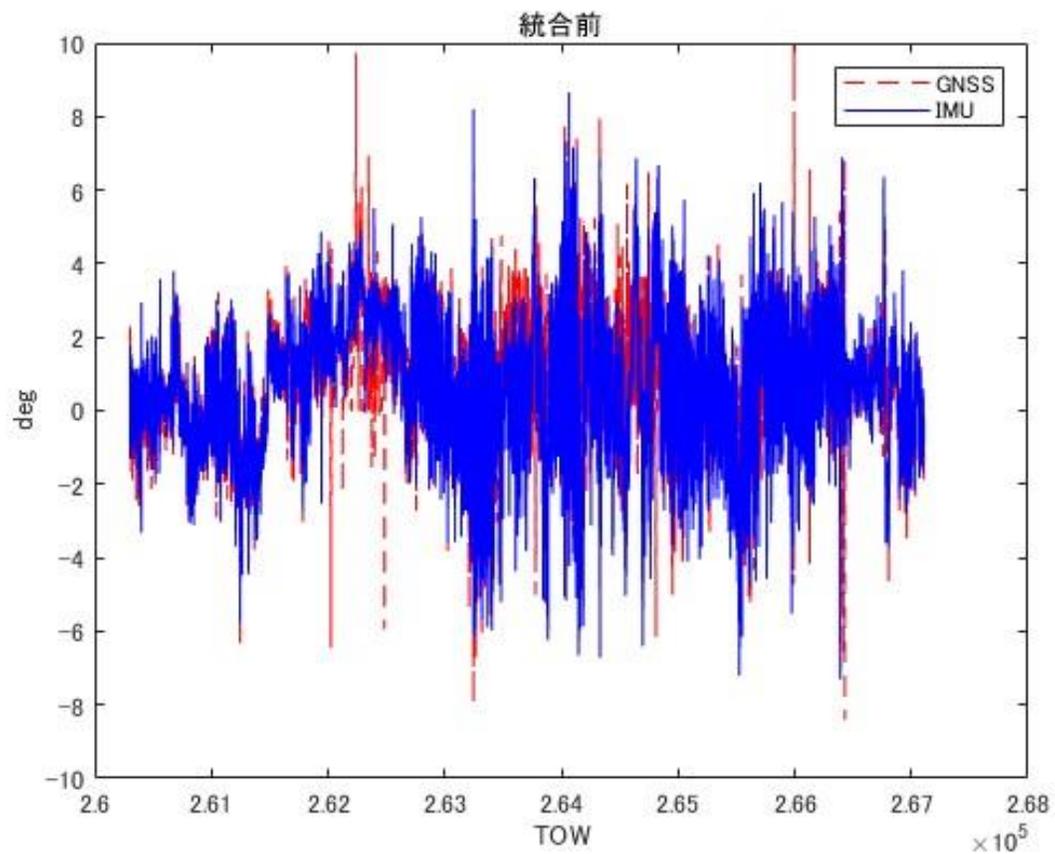
$$\hat{x}(0) = [10 \quad 10 \quad 20]^T, P(0) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, Q = \begin{bmatrix} 1.25 & 0 \\ 0 & 1 \end{bmatrix}, R = 10^{-4}$$

を用い統合した。

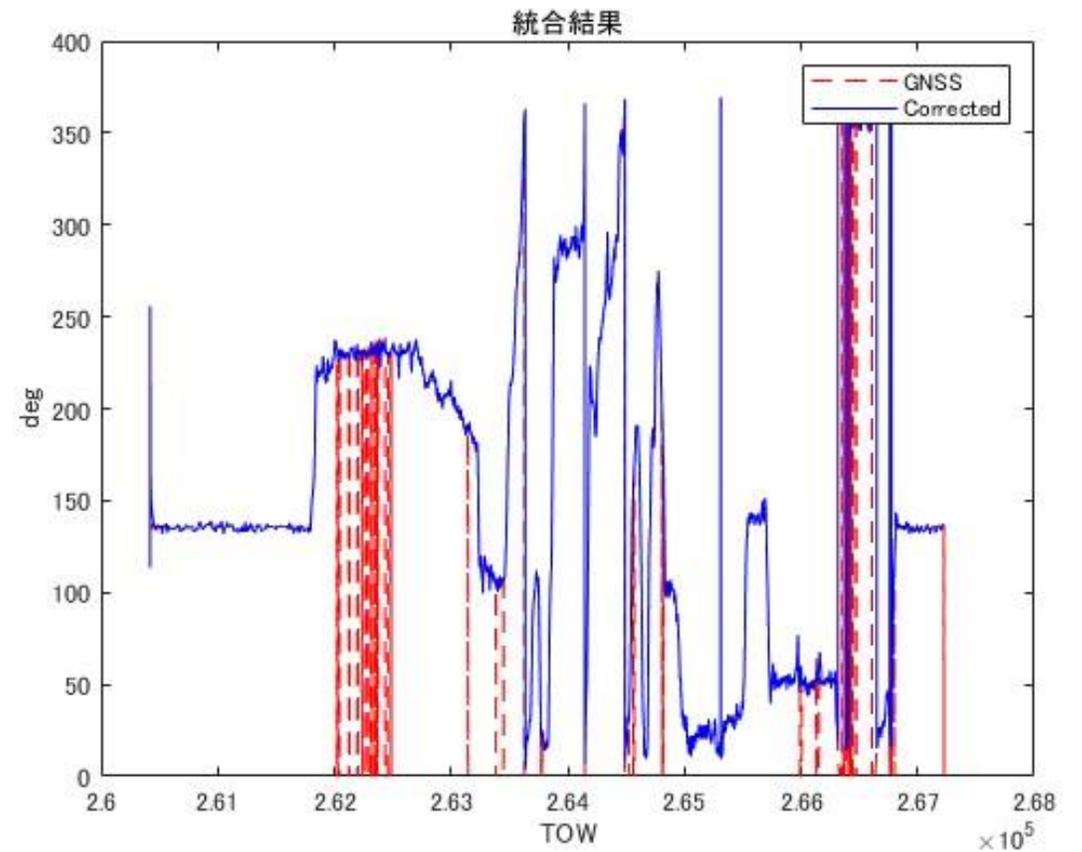
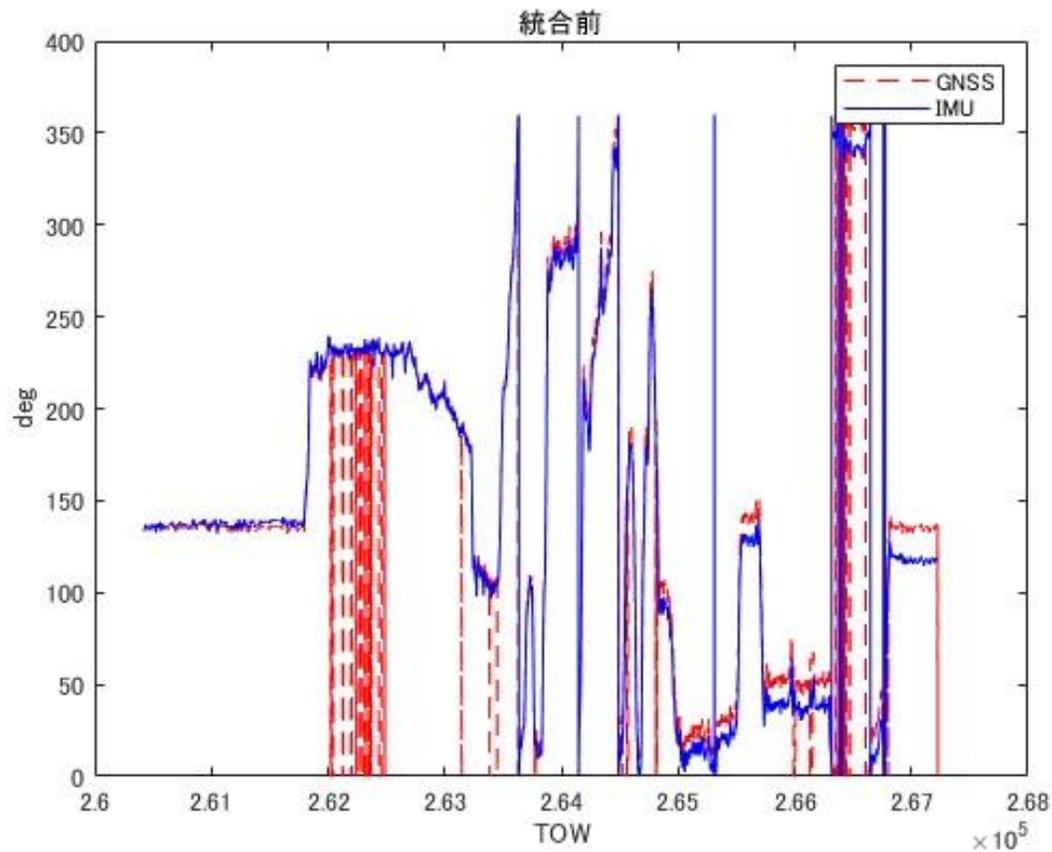
# ピッチ GNSS,IMU統合結果



# ローカル GNSS, IMU統合結果



# 方位 GNSS,IMU統合結果



# まとめ

- AMOアンテナ比ベノバテルアンテナの方が測位精度が高いことが別の実験でわかったため最近の実験では船体上部のアンテナはノバテルアンテナに統一している。
- カルマンフィルタ、相補フィルタを用いることで姿勢、方位を正確に推定することができることが分かった。
- 姿勢の完全な真値はわからないため、今後どのようにしていくのか。

# 参考文献

- /1/Phil Kin、Kalman Filter for Beginners: With MATLAB
- /2/足立修一、丸田一郎、カルマンフィルタの基礎