

shipmotionSTFT_win マニュアル

作成日:2024 年 5 月 28 日

最終更新日: 2024 年 8 月 4 日

バージョン:0.0.1(仮)

作成者:東京海洋大学大学院 海運ロジスティクス専攻 小森健史

目次

1. 概要.....	3
1.1. 本システムの目的.....	3
1.2. 本システムの動作概要	3
1.3 本システムの特徴.....	3
1.4. 本システムに必要な物品.....	3
2. 使用方法.....	4
2.1. アプリ使用前の事前準備	4
2.2. アプリ使用前の事前準備	7
2.3. shipmotionSTFT_win.exe の動かし方	9
2.4. 使用上の注意.....	13
利用ライブラリ	14

1. 概要

1.1. 本システムの目的

本システムは動揺周期と GM 推定値を操船者が確認できるようにすることで、復原力不足による転覆や激しい動揺による貨物損害事故を未然に防ぐことを目的とする。

1.2. 本システムの動作概要

このアプリは船体座標各軸方向/周りの加速度/角速度を短時間フーリエ変換することによって、船体の動揺周期を計算し、結果を csv・txt ファイルに出力する。船のモーションデータは主にスマートフォンアプリで取得することを想定しており、実行ファイルは通常の Windows 環境で動作する。解析したいデータは csv ファイルに書き込んでおく。

1.3 本システムの特徴

本システムの特徴を以下に列挙する。

- ・スマートフォン(Android™・iOS®両方可)と Windows® PC があれば、利用可能(動揺周期を測定できる)。
- ・実行ファイルに組み込んだ Python プログラムは オープンソース であり、それぞれのニーズに合うようにカスタマイズすることも可能。
- ・動作の仕組み上、リアルタイムで動揺周期を測定することはできない。

1.4. 本システムに必要な物品

(i)スマートフォンなどの 3 軸の角速度・加速度を計測できる機器

-Roll,Pitch(姿勢角)も計測できるとなお良い。

(ii)Windows PC

-Python の開発環境があれば、MacOS、Linux の PC でも可能。

2. 使用方法

2.1. アプリ使用前の事前準備

① プログラムのダウンロード

[Komori-GitHub/motiondataSTFT](https://github.com/Komori-GitHub/motiondataSTFT) から実行ファイル/プログラムと設定ファイルを PC にダウンロードして、任意の場所に保存。プログラムの内訳は表 1 に示すとおりである。

表 1 配布したファイルの内訳

ファイル名	説明
stft_csvdata.exe	Csv ファイルに保存したデータを読み込んで、短時間フーリエ変換を行い、結果をファイルに保存する Windows アプリ(実行ファイル)。Pyinstaller を用いて stft_csvdata.py を組み込んだ。
dist/setting.txt	stft_csvdata.exe(短時間フーリエ変換)の設定ファイル。
dist/motiondata.csv	モーションデータを保存しておく csv ファイル。1 列目が時間で、2 列目に解析データを入力
stft_csvdata.py	Csv ファイルに保存した角速度データを読み込んで、短時間フーリエ変換を行い、結果をファイルに保存する Python プログラム。 Python の開発環境が PC に備わっている場合は exe ファイルよりも、こちらをコンパイルして動作させた方が、速く処理できる。

stft_csvdata.exe は Releases からダウンロードした後、dist フォルダ内に移動させる。図 1 のようにダウンロードがセキュリティー上ブロックされた場合(Edge だとよくブロックされる)は、…→保存をクリックすればダウンロードできる。

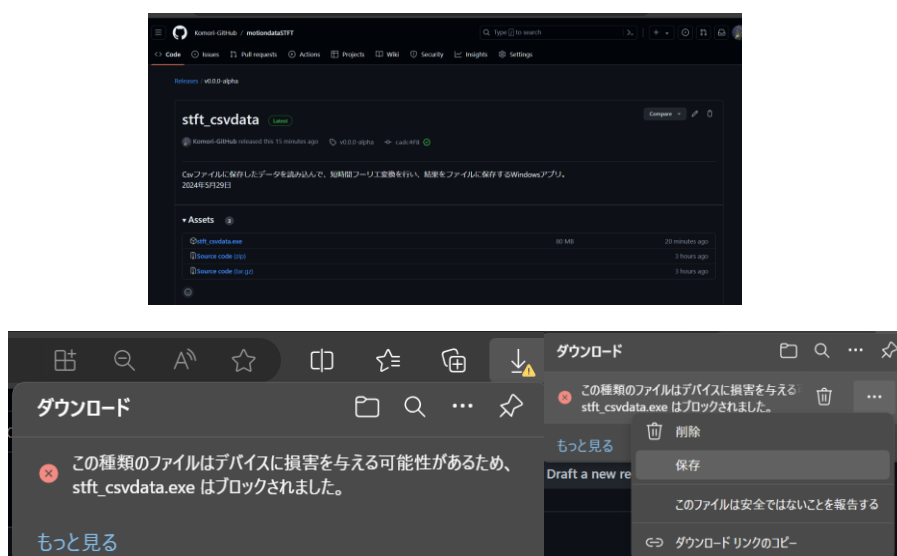


図 1 GitHub Releases と exe ダウンロード時の警告

② モーションデータを記録するスマートフォンアプリのダウンロードと設定

- (i) スマートフォン(Android™・iPhone®両方 OK)に”Sensor Logger”というアプリケーションをダウンロードする。他にも、無料で使えてスマートフォン内蔵のモーションセンサーのデータをファイルに記録する機能を持ったアプリは存在するのだが、著者は個人的にこのアプリが一番使いやすいと思う。



図 2 スマートフォンアプリ “Sensor Logger”

- (ii) スマートフォンを適当に回したり/傾けたりして、3 軸の角速度・加速度から X,Y,Z 軸がスマホのどちらを向いているかの確認を行う。角速度は軸回りの値、加速度はその軸方向(軸に沿った)の値が出力される。一般的なスマートフォンは図 3 右の通りの座標系(右手系)のはずである。

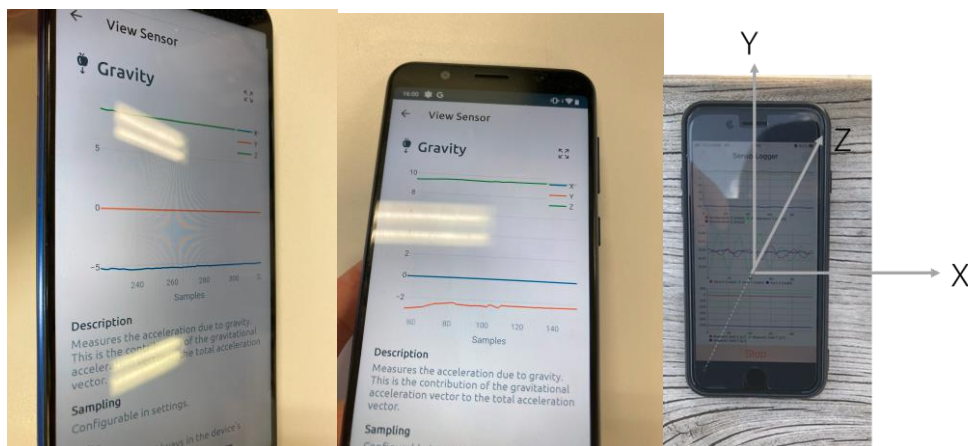


図 3 加速度の表示とスマートフォンの座標系

※Logger Tab の各 Sensor 右横の目(👁)のボタンをクリックすると、該当 Sensor の計測データのグラフが表示される。グラフの横軸は時間(サンプル数)で、データがリアルタイムでストリーミングされていく。

- (iii) “Sensor Logger”アプリを開き、“Gravity”, “Gyroscope”, “Orientation”を選択。Settings(⚙)→Sampling Frequencies 画面で、サンプリング周波数を変更できる。サンプリング周波数は 2Hz 以上(実際のサンプリング周波数が 2Hz 以上になるよう)に設定することを勧める。Custom で Change ボタンをクリックすれば任意の値に変更できる。
- (iv) “Recording&Workflow”の Keep Awake のところで、“On”もしくは“On&Dim”を選択。ここが“Off”になっているとスマートフォンがスリープして、機種によってはアプリのバックグラウンド処理に支障が出て、データ落ちが頻発してしまう。

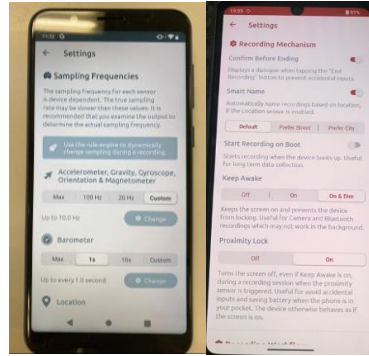


図 4 サンプリング周波数の設定

- (v) (Start Recording を押したときにも促されるが、)Android の場合は位置情報へのアクセスを「常に許可」に設定する。

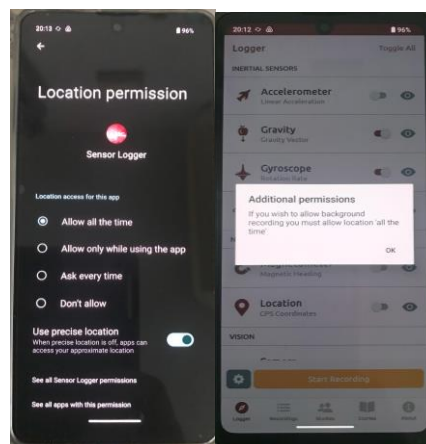


図 5 位置情報の許可設定(Android)

2.2. アプリ使用前の事前準備

スマートフォンで船体動揺を計測することを前提に説明するが、船体動揺の計測については計測データをファイルに出力さえできれば、別のモーションセンサーユニット(例えば [SensorTile.Box: 今すぐ使える IoT 端末開発キット - ST マイクロエレクトロニクス](#))を使ってもよい。

- (i) スマートフォン本体をアプリで表示されている Roll・Pitch が船のそれらと一致する向きにして、振動の少ないところに設置(防振マットを敷くのもあり)&揺れで動かないように固定する。Roll・Pitch に誤差を生じさせないためにも、スマートフォンの辺が船首尾線と平行になるように&画面が水平になるように設置する。船の上下揺れを測定したい場合は船体中央/浮面心付近にスマートフォンを設置。

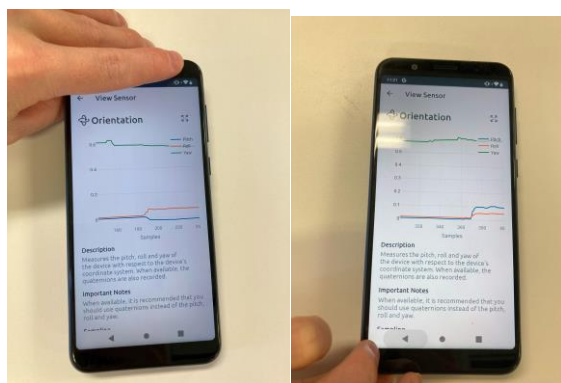


図 6 スマートフォンの Roll・Pitch

- (ii) Logger 画面で Start Recording を押して、記録開始。船の動揺は非定常なので観測は少なくとも 20~30 分以上(大きい船ほど長く)は行った方が良く、計測中はスマートフォンを動かしたり(画面に触れる必要がある場合はそっとタップすること)、振動を与えないように注意する。記録を停止する際は End Recording を押す。

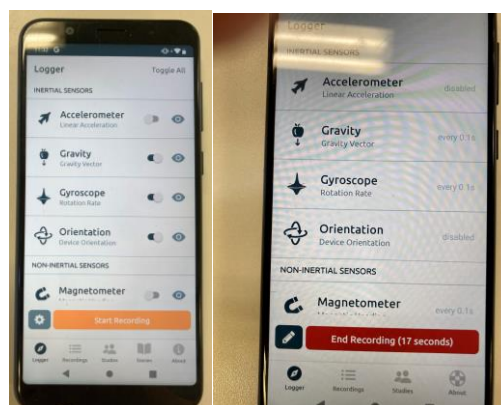


図 7 スマートフォンでの Recording

- (iii) Recordings(:三)タブに切り替えて、実験データを選択し、Export(📄) ボタンをクリック。CSV in Zip File を選択して、Export Recording をクリック。Share の選択を促されるので、自分が希望する場所を選択して保存。スマートフォン上のフォルダに保存したい場合は Android の場合 Drive を、iOS の場合は「ファイルの保存」を選択すればよい。

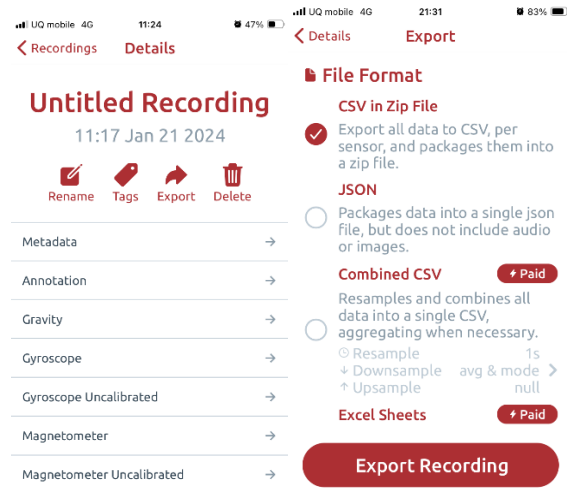


図 8 計測データの export

- (iv) 保存した zip ファイルを解凍すればデータが保存された csv ファイルが手に入る。このファイルでスマートフォン内蔵のモーションセンサーで計測した角速度・加速度や姿勢(Roll・Pitch・Heading)が確認できる。

nents > Sensor data > 2024-05-29_03-15-39.zip						
Name	Type	Compressed size	Password pr...	Size	Ratio	
Annotation.csv	Microsoft Excel CSV ファイル	1 KB	No	0 KB	0%	
Gravity.csv	Microsoft Excel CSV ファイル	39 KB	No	106 KB	64%	
Gyroscope.csv	Microsoft Excel CSV ファイル	47 KB	No	115 KB	60%	
GyroscopeUncalibrated.csv	Microsoft Excel CSV ファイル	46 KB	No	114 KB	61%	
Metadata.csv	Microsoft Excel CSV ファイル	1 KB	No	1 KB	32%	
Orientation.csv	Microsoft Excel CSV ファイル	79 KB	No	198 KB	60%	

図 9 スマートフォンアプリで出力された csv ファイル

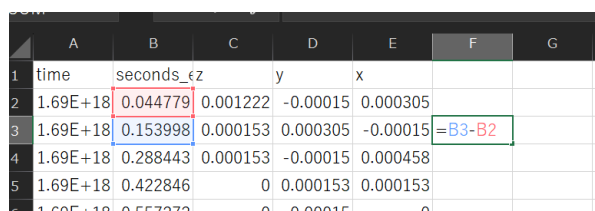
2.3. shipmotionSTFT_win.exe の動かし方

① 短時間フーリエ変換の設定

表 2 setting.txt の設定項目

行目	項目	内容
1	Sampling frequency	データのサンプリング周波数
2	Number of sample per segment	短時間フーリエ変換(STFT)1 セグメント毎のサンプル数

スマートフォンアプリの場合サンプリング周波数の設定値と、実際の周波数が異なっていることがある(特に Android で)。実際のサンプリング周波数を出力された csv ファイルで seconds_elapsed の差分をとることによって調べて、この値を設定して欲しい。サンプリング周波数の誤差(設定値との差)はそのまま結果(動揺周期)の誤差になる。



	A	B	C	D	E	F	G
1	time	seconds_elapsed	z	y	x		
2	1.69E+18	0.044779	0.001222	-0.00015	0.000305		
3	1.69E+18	0.153998	0.000153	0.000305	-0.00015	=B3-B2	
4	1.69E+18	0.288443	0.000153	-0.00015	0.000458		
5	1.69E+18	0.422846	0	0.000153	0.000153		
6	1.69E+18	0.557272	0	-0.00015	0		

図 10 実際のサンプリング周期(データファイル)

フーリエ変換の周波数分解能はサンプリング周波数を F_s 、1 セグメント毎のサンプル数を N とすると、 F_s/N で表される。動揺周期が 10s の場合は周波数分解能が 0.01Hz、動揺周期が 30s の場合は周波数分解能が 0.002Hz 以上確保できるとよいと思う。また、シャノンのサンプリング定理よりサンプリング周波数は観測したい周波数の 2 倍以上にしておかなければならない。例えば周期 1s/周波数 1Hz の動揺を観測したい場合はサンプリング周波数が 2Hz 以上ないといけない。これらのことを考慮して設定する必要がある。

② 解析したいデータの用意

“motiondata.csv”に解析したいデータを書き込んで、保存しておく。1 列目は時間[s]、2 列目は解析したいデータ(例えば Roll,Pitch)で、1 行目から記入。時間の単位は必ず秒にすること。そうでないと、正しい計算結果が求まらない。

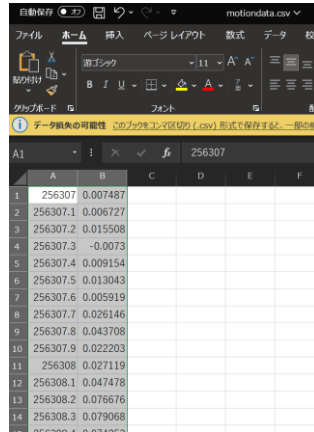


図 11 解析データを書き込む csv ファイル

③ exe ファイルの実行させる

“stft_csvdata.exe”をダブルクリックすれば、ターミナルみたいな Window が立ち上がって、プログラムが実行される。Windows Defender にブロックされた場合は詳細情報→実行をクリックすれば、実行できる。



図 12 Windows Defender によるブロック

④ 処理結果の確認

- (i) プログラム内で短時間フーリエ変換が終了すると、その結果がグラフで出力される。グラフ表示に使用したライブラリ“matplotlib”の使用上、処理を次へ進めるためにはグラフのウィンドウを閉じる必要がある。

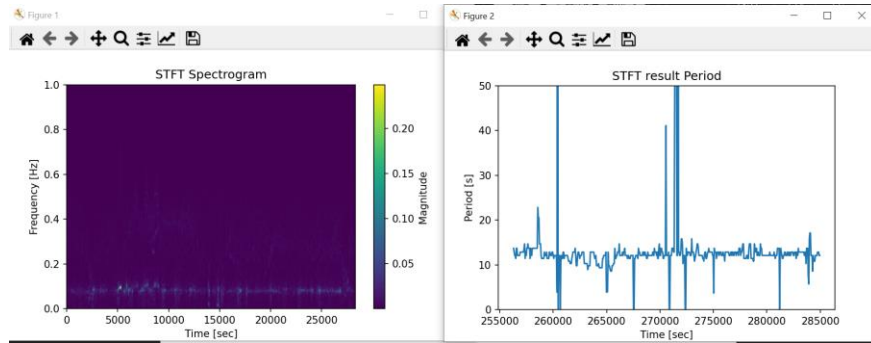


図 13 出力されるグラフ

- (ii) グラフのウィンドウを閉じて、処理が完了した後にターミナルに周期の最頻値・中央値・平均値が出力される。この 3つの値は外れ値を含め全データを使って計算しているので、信頼性が欠けている点に注意して欲しい。Dist フォルダ内の”output.csv”には STFT の結果として各時刻の周波数・周期が出力されている。”period.txt”には最頻値・中央値・平均値が出力されている。出力ファイルを保存したい場合は、場所を移すか名前を変えるかしないと、次の実行時に上書きされてしまうので注意されたい。

```

C:\Users\t1810\OneDrive\デスクトップ\Pythonexe\dist\stft_csvdata.exe
stft_csvdata.py:36: RuntimeWarning: divide by zero encountered in divide
最頻値[s]: 12.094117647058827
中央値[s]: 12.094117647058827
平均値[s]: 13.203658132032798

```

図 14 ターミナルに出力された最頻値・中央値・平均値

Name	Status	Date modified	Type	Size
motiondata.csv	✓	2024/05/27 20:39	Microsoft Excel CSV ...	6,192 KB
output.csv	✓	2024/05/29 14:02	Microsoft Excel CSV ...	27 KB
period.txt	✓	2024/05/29 14:02	Text Document	1 KB
setting.txt	✓	2024/05/28 15:31	Text Document	1 KB
stft_csvdata.exe	✓	2024/05/28 20:23	Application	441,423 KB

period.txt - Notepad

File Edit Format View Help

mode[s]:12.094117647058827
 median[s]:12.094117647058827
 average[s]:13.203658132032798

図 15 出力された結果ファイル

2.4. 使用上の注意

① 配布したファイルについて

設定ファイルの設定値以外は書き換えないでください。設定ファイル(setting.txt)・データファイル(motiondata.csv)の名前は変更しないでください。プログラムが正常に動作しなくなります。

② ライブラリのバージョンについて

このアプリケーション(プログラム)は Python ライブラリを使用しており、このバージョンが異なると正常に動作しない可能性があります。ライブラリ関連で問題が生じたときは本マニュアルの末尾に示したバージョンでインストールし直してみてください。

③ スマートフォンアプリ"Sensor Logger"による計測について

スマートフォンの機種によっては

④ IMU の計測値について

スマートフォンに搭載されている低コスト (Consumer level) MEMS-IMU は温度によって誤差が蓄積し、著者が恒温槽で行った実験では 40°C 程度の温度変化で、Roll, Pitch に 1° 強の誤差が生じました。スマートフォンは一般商船に搭載されている電子傾斜計の性能基準を満たしません。Roll, Pitch の角度はあくまでも参考程度にお使いください。

⑤ GM 推定値の正確性について

著者が汐路丸で行った実験の結果として、横揺れ周期から誤差 10% 以内で G_0M (自由水影響を考慮した横メタセンタ高さ) を推定できましたが、他の船でどれくらい正確に求められるかは自身でご確認下さい。

⑥ 今後のアップデート方針について

アップデートは GitHub 上に新しいプログラム群(バージョン)をアップロードする形で行いたいと思います。フィードバック等が得られましたらそれに基づいた修正を行うつもりですが、著者の時間・能力の制約上、ご要望に応えられないことがございます。あらかじめご了承ください。

⑦ ライセンスについて

GitHub 上で公開されている shipmotionSTFT のプログラムは オープンソース です。引用元を明記していただければ、商用利用・変更したうえでの再配布も可能です。

利用ライブラリ

NumPy, ver1.26.4, <https://numpy.org/ja/>, 2024.2.5.

pandas, ver2.1.4, <https://pandas.pydata.org/>, 2023.12.8.

SciPy, ver1.11.4, <https://scipy.org/>, 2023.11.18.

Matplotlib, ver3.8.0, <https://matplotlib.org/>, 2023.9.14.