学位論文公開発表会用資料

船舶に対するGNSSスプーフィングの脅威の検討及び時刻に着目したGNSSスプーフィングの検知について

令和7年2月10日
東京海洋大学大学院(社会人学生※)修士2年
横田　健太朗

※個人の見解に基づくものであり、所属する組織の見解に基づくものではない。

次　第

本題に入る前に・・・

GPSスプーフィングっていうのがリスクとしてあるんだなあ…

うちの船でも対策しないといけないんだろうな～

でも、対策にはお金がかかるし、本当に対策しないといけないんだろうか…？

（想定）船舶関係者

自分

船舶に対するGNSSスプーフィングの脅威（リスク）について検討しました。

その結果、脅威度は低いが発生頻度は高い脅威に備える対策が必要と考えました。

その対策について、時刻に着目したスプーフィング検知手法を検討・導出しました。

1　概　要（研究の背景）

船舶においてGNSSは重要
省人化・無人操船に向けた取り組み⇒GNSSへの依存はさらに高まっている。

GNSSの脆弱性を利用した攻撃は多岐に亘る。
・近年のウクライナ紛争やガザ紛争においてはジャミングやスプーフィングといった攻撃が観測
・スプーフィングは相手に攻撃されていることを認知させずに攻撃できる可能性
・SDR(software defined radio) の登場により安価に攻撃機材を購入・整備することが可能
⇒脅威は高まっている。

GNSSの脆弱性を利用した攻撃の検知やその対策の研究が行われている。
どのように相手を攻撃するのかといったユースケースを含んだ研究は少ない。
特に、船舶への影響について言及されたものはない。

既存の攻撃手法のうちGNSSスプーフィングに着目
JPP(Joint Planning Process) を準用して、ユースケースに焦点
船舶におけるGNSSスプーフィングの脅威の分析を行った。

3　MLCOAへの対策の導出

MLCOAに求められる対策とは何か？

MDCOA：脅威度が高いが、発生頻度は低い⇒先行研究等で示される費用が高額な対策でもよい
MLCOA：脅威度は低いが、発生頻度は高い⇒MDCOAより効果は悪くとも安価な対策が求められる

4　船舶の特徴

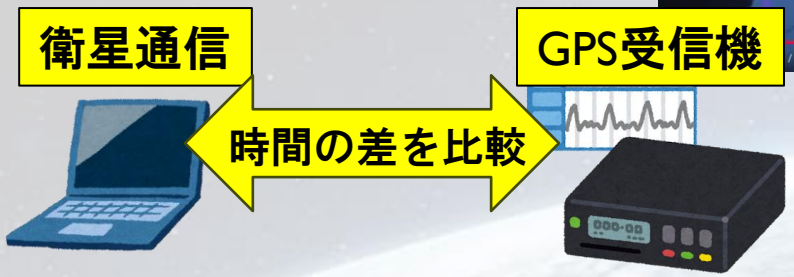
衛星通信を介したインターネット環境があり、（GNSSで得られた）自船位置を陸上まで送信している。

- ***正確な時刻の入手***

1　インターネットを介し、NTPサーバにアクセスして取得
2　GNSSで取得

- ***スプーフィング信号生成上の問題***

⇒完全な時刻一致は安価にはできない。
・送信までの処理遅延
・生成には高額な機材・ソフトが必要



https://www.youtube.com/watch?v=NZMG8-QVNFE

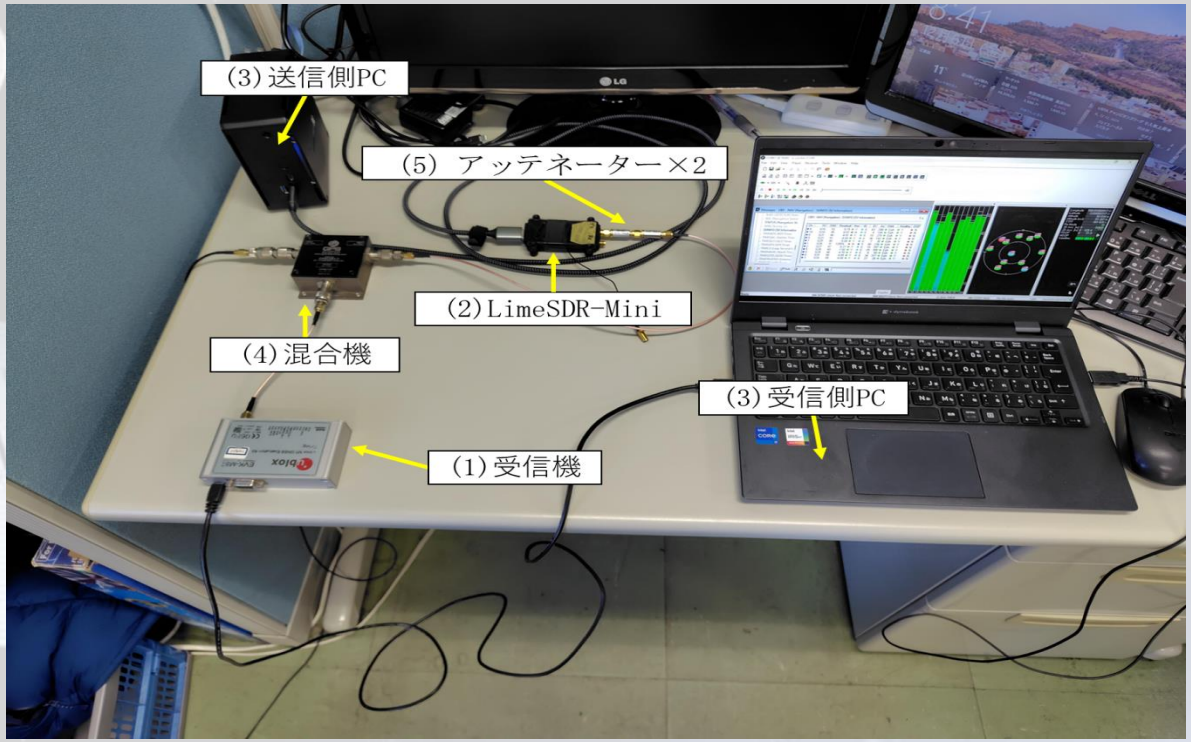
⇒正常時は1と2は同じ時刻で、スプーフィング時には異なる時刻となるのではないか？
これを判別するだけのPythonコードであれば自作することで対策ができるのでは？

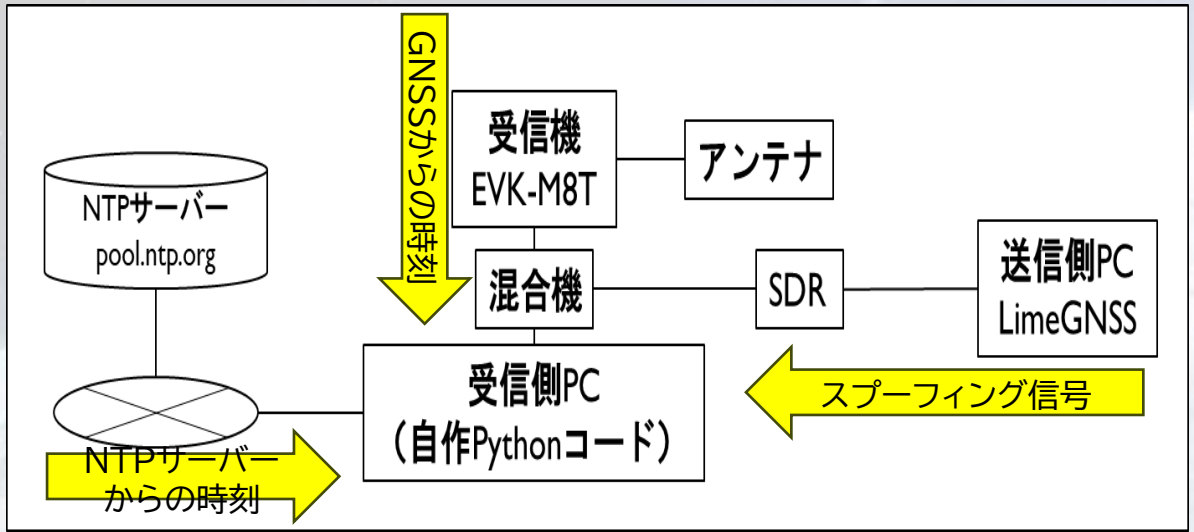
4　実験機材等

1　実験機材
(1)　受信機:Ublox EVK-M8T
(2)　SDR:LimeSDR-Mini
(3)　PC×2
　ア　受信側:Dynabook GZ/HWL
　イ　送信側:研究室ビルトPC
(4)　混合機:CPDC2X1-T
(5)　アッテネーター:10dB*2
(6)　アンテナ:研究室屋上設置アンテナ

2　使用ソフトウエア
(1)　LimeGNSS_v2
(2)　u-center_v23.08
(3)　visual studio2022(自作Pythonコード)

3　機材構成
　右図のとおり。





5　実験結果

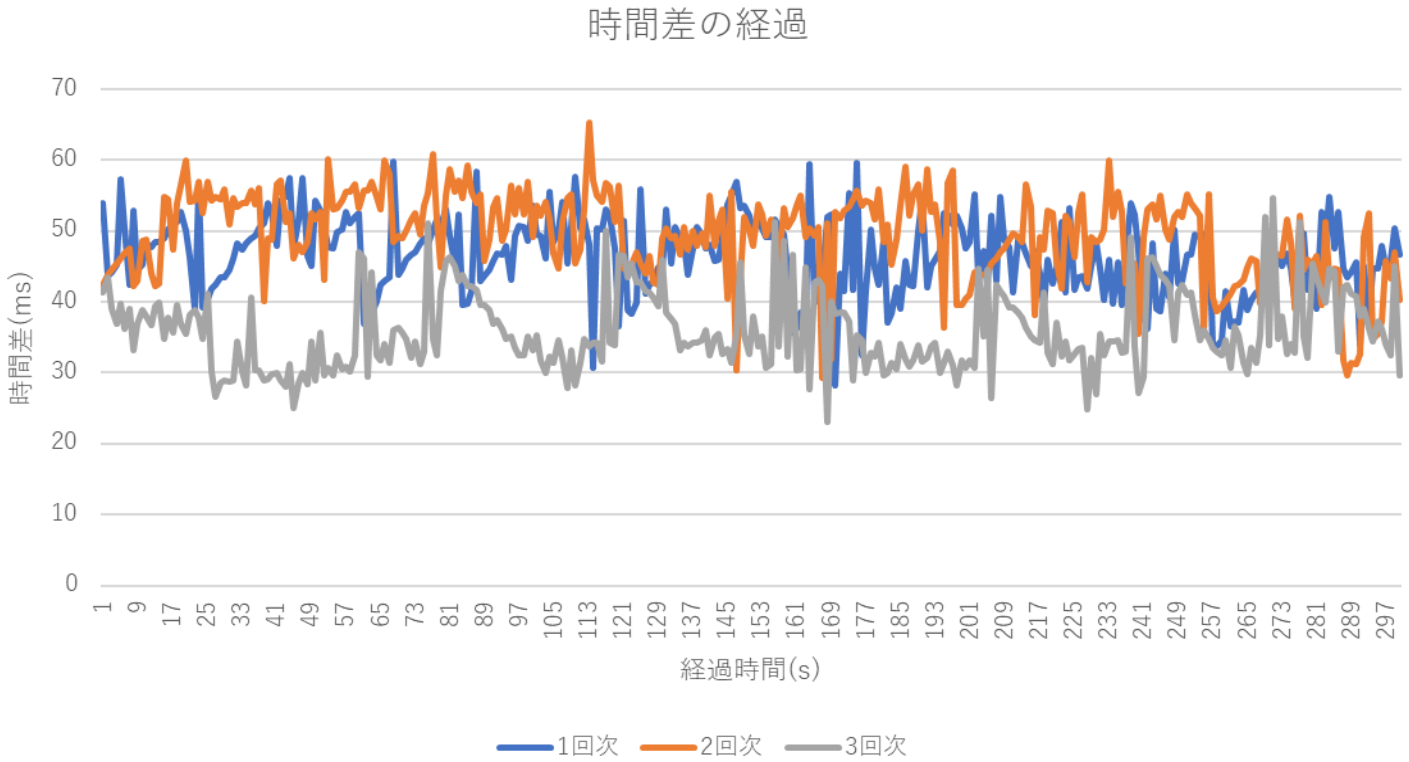
1　実験要領
　自作Pythonコード(試作)をもって実験を行っていく。
(1) インターネットを介し、NTPサーバから取得した時刻とGNSSで取得した時刻は同じか検証
(2) スプーフィング時の時間差は正常時と比べて検知することができるのか検証
(3) (1)及び(2)が成功した場合、検知するためのコードを検討
(4) 導出したコードを検証

自作Pythonコード内容（試作時）

(1)　コード実行時にNTPサーバ（https://www.ntppool.org/）へアクセスし、使用するPCの時刻（システムクロック）をNTPサーバと同期させる。
(2)　GNSSアンテナで取得したGNSS信号は受信機（Ublox M8T）でNMEAメッセージに復号され、このNMEAメッセージのうち「GPRMC」及び「GPZDA」からGNSS時刻を得る。
(3)　NMEAメッセージを使用する制約から、毎秒単位でしか時刻を取得できないため、取得したタイミングでPCのシステムクロックと比較し、その差分を導出する。

5　実験結果等

1 インターネットを介し、NTPサーバから取得した時刻とGNSSで取得した時刻は同じ？



	1回次	2回次	3回次
平均値(ms)	46.57672333	49.45481333	35.85375667
標準偏差	5.515605777	6.165728944	5.558507322
最小値(ms)	41.06111756	43.28908439	30.29524934
最大値(ms)	52.09232911	55.62054228	41.41226399
標準偏差±1以内のデータ数	204	219	202
それ以外のデータ数	96	81	98
1以内のデータが占める割合	68%	73%	67%

実験日：2024/12/25

3回に分けて300秒間計測を実施

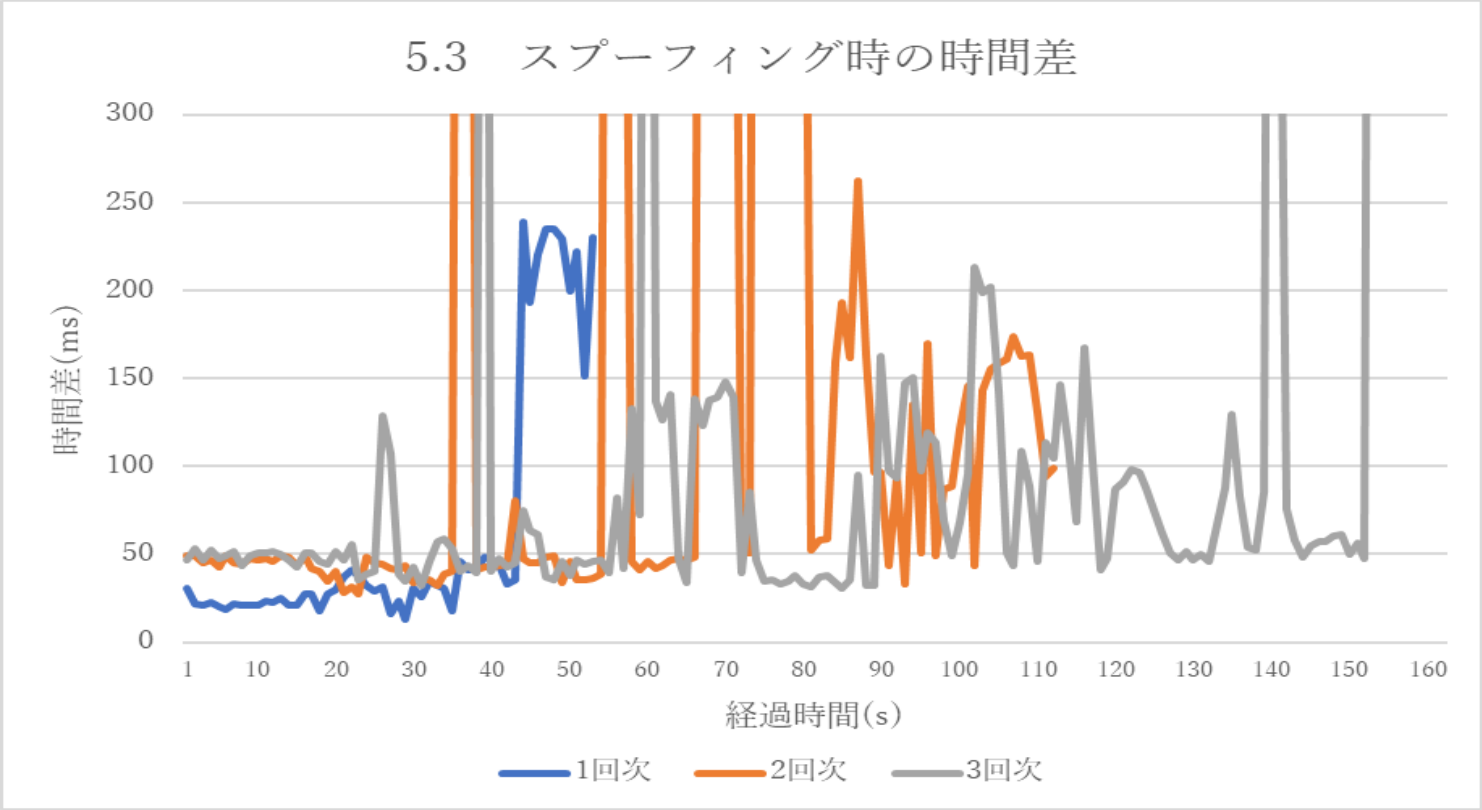
結果：誤差が約50ms生じているが、安定している。

理由は、受信機での処理遅延が考えられるが、将来的に実装する環境（船舶ごと）の受信機の性能にもよるところなので、誤差をなくすことは今回の研究では対象外とする。

各回次の標準偏差±1以内のデータ数は67%以上であり、正規分布の場合は68.27%なので、**正規分布に近いデータが取得できている**ことがわかる。

5　実験結果等

2 スプーフィング時の時間差は正常時と比べて検知することができるのか？



	1回次	2回次	3回次
平均値(ms)	20.8592	50.10535	48.79365
標準偏差	2.807808053	1.4253862	2.974930345
最小値(ms)	18.05139195	48.6799638	45.81871966
最大値(ms)	23.66700805	51.5307362	51.76858034

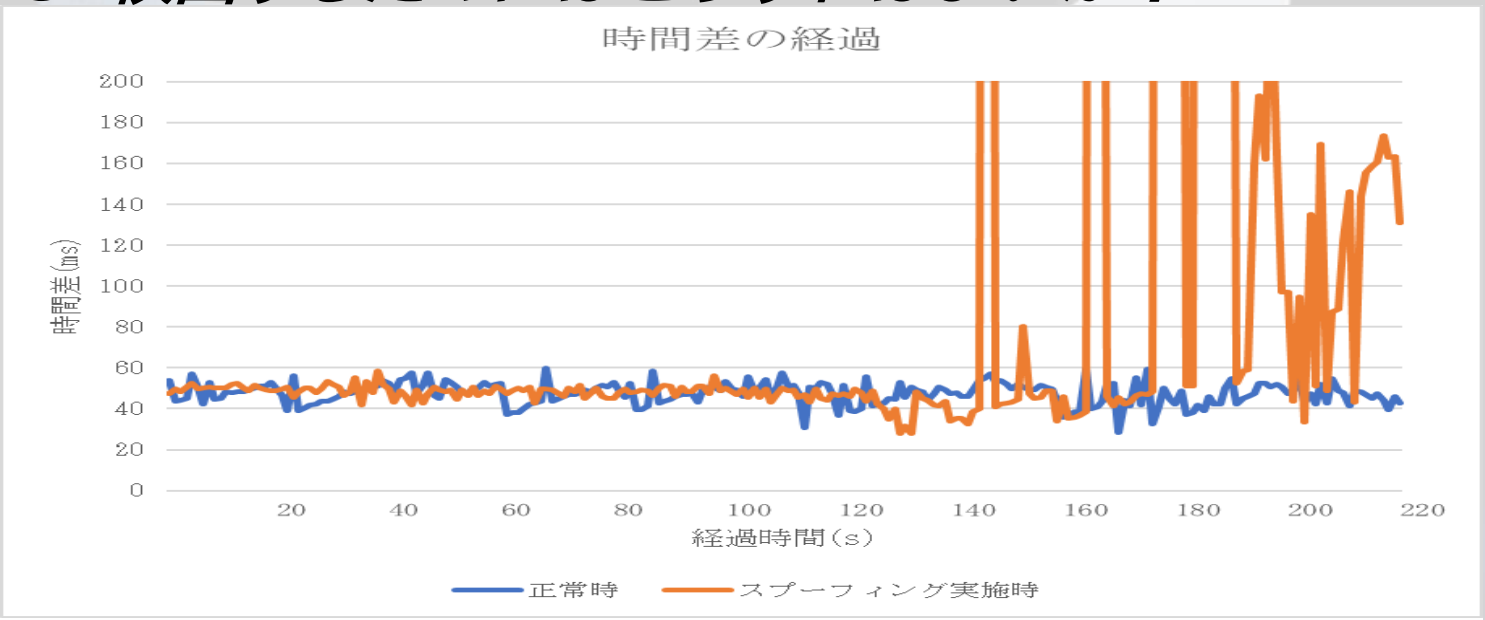
実験日:2024/12/25

正常時よりばらつきが激しくなった。
時間差も約10msの範囲内から100ms以上となった。

結果:検知することは有効である。

5　実験結果等

3 検出するためにはどうすればよいか？



GNSS 時刻	システム時刻	時刻差	範囲内/外
2024-12-26 05:09:10.000	2024-12-26 05:09:10.026	26.63	範囲内
2024-12-26 05:09:11.000	2024-12-26 05:09:11.029	29.87	範囲内
2024-12-26 05:09:12.000	2024-12-26 05:09:12.029	29.29	範囲内
2024-12-26 05:09:13.000	2024-12-26 05:09:13.032	32.27	範囲外
2024-12-26 05:09:14.000	2024-12-26 05:09:14.013	13.66	範囲外
2024-12-26 05:09:15.000	2024-12-26 05:09:15.016	16.4	範囲外
2024-12-26 05:09:16.000	2024-12-26 05:09:16.014	14.93	範囲外
2024-12-26 05:09:17.000	2024-12-26 05:09:17.015	15.72	範囲外
2024-12-26 05:09:18.000	2024-12-26 05:09:18.015	15.98	範囲外
2024-12-26 05:09:19.000	2024-12-26 05:09:19.019	19.2	範囲外
2024-12-26 05:09:20.000	2024-12-26 05:09:20.020	20.47	範囲外
2024-12-26 05:09:21.000	2024-12-26 05:09:21.019	19.01	範囲外
2024-12-26 05:09:22.000	2024-12-26 05:09:22.019	19.13	範囲外

左図：正常時とスプーフィング時の対比

正規分布に近いデータなので、標準偏差±1以外となる割合は32%であり、誤検知となる可能性が高い。

標準偏差±1以外となるデータが10回連続したことをもって検知することを検討

⇒正常時でも10回連続し、誤検知となることが生じた。

標準偏差の±2以内（95.5%）へと変更

＋

標準偏差の値を、計算処理を抑えるため「5」と仮置きしてコード内に反映

5　実験結果等

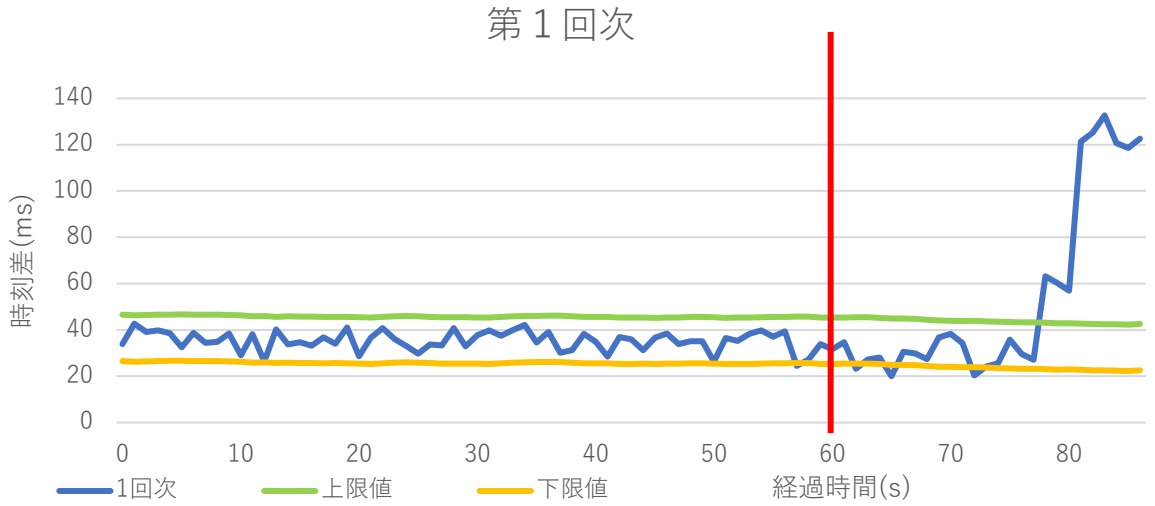
3 検出するためにはどうすればよいか？

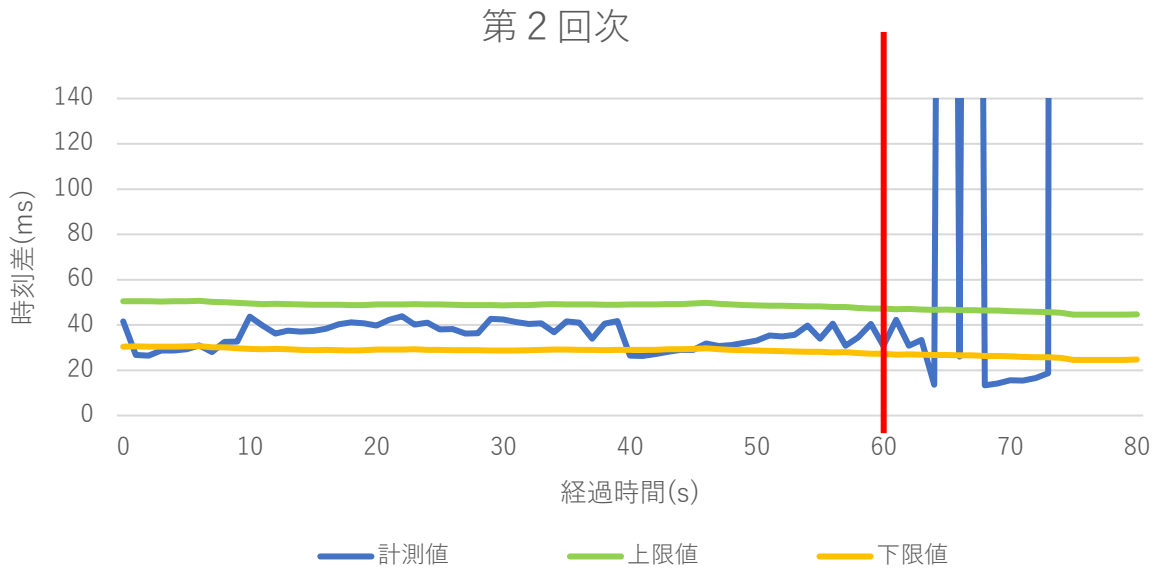
（1） コード実行時にNTPサーバ（https://www.ntppool.org/）へアクセスし、使用するPCの時刻（システムクロック）をNTPサーバと同期させる。
（2） GNSSアンテナで取得したGNSS信号は受信機（Ublox M8T）でNMEAメッセージに復号され、このNMEAメッセージのうち「GPRMC」及び「GPZDA」からGNSS時刻を得る。
（3） NMEAメッセージを使用する制約から、毎秒単位でしか時刻を取得できないため、取得したタイミングでPCのシステムクロックと比較し、その差分を導出する。

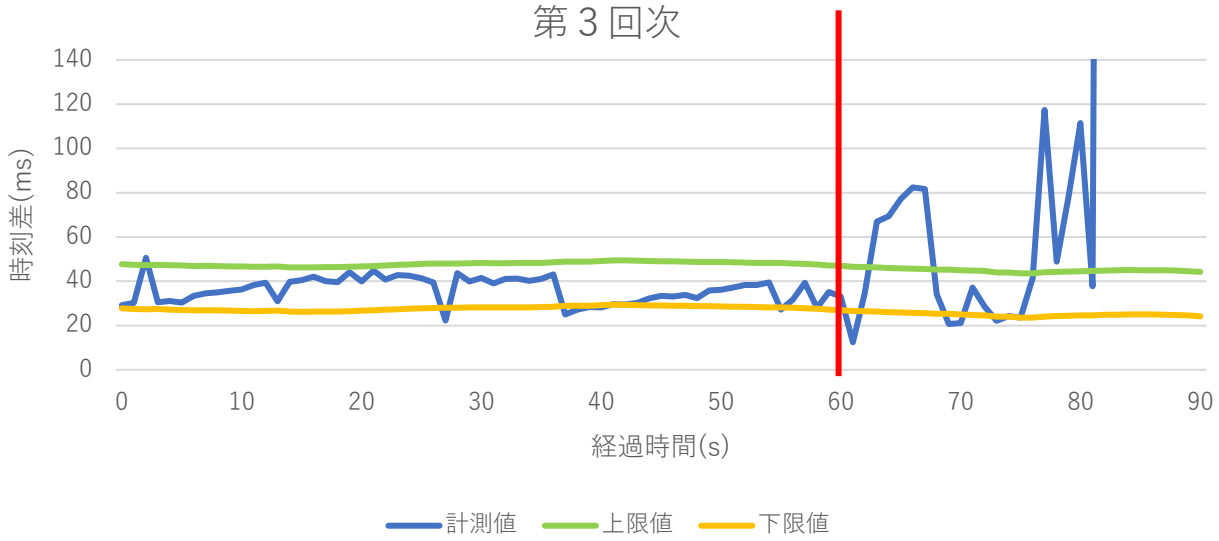
（1） 35回分のデータを保持し、n回次にはn-1回次までで（2）の範囲内と判定された時間差の平均値を求める。
（2） n回次の時間差が求められた平均値の±20ms（標準偏差を5と仮定し、標準偏差の2倍以内）の範囲内かどうか判定する。
（3） 1回次は範囲内として計算する。
（4） 10回連続して範囲外となった場合、スプーフィングの可能性が高いとしてアラートを鳴らす。
（5） 開始30秒以内は検知しない

5　実験結果等

4 検出実験







60秒時点でスプーフィング攻撃を実施

結果：各回次探知までの時間の開きがあるものの、探知することに成功した。

6　まとめ

船舶におけるGNSSスプーフィングについて、JPPを活用した脅威の検討を行い、MDCOA及びMLCOAを導出した。

MDCOA（最も危険な敵の行動方針）：***「沿岸付近での船舶とドローンの併用による攻撃」***

MLCOA（最も蓋然性の高い敵の行動方針）：***「沿岸におけるドローンによるスプーフィング攻撃」***

このうち、MLCOAへの対策についての検討を行い、船舶の機材の特徴からGNSSから取得した時刻と、インターネットを介してNTPサーバから取得した時刻の差に着目したGNSSスプーフィング防御手法について検討し、実験を行った。

その結果、防御手法として自作したPythonコードにおいて、1秒ごとの時刻差を活用することで、スプーフィング攻撃の可能性を検知することに成功した。

ただし、以下の2点については考慮して使用する必要がある。

(1)***標準偏差の2倍以内として平均値から±20msの範囲外のものを検出しているため、スプーフィング信号がこの範囲内であれば検知することは不可能***

(2)コードを実行した際、何度か初期の段階でGNSS時刻との時刻差が安定せず、誤検知してしまう事象が生起した。この場合は、コードの再実行をもって対処とする必要がある。

6　今後の課題・展望

今回の実験における課題は以下の点である。

(1) 使用するGNSSをGPSに限定
⇒他のL1周波数を使用するマルチGNSS対応とする
(2) スプーフィング信号を実放射していない
⇒実環境においてスプーフィング信号を検知できるか
(3) LimeGNSSを使用したスプーフィング信号に限定
⇒より精度の良い信号やミーコニング攻撃を検知できるか
*(4) 船舶環境への適用を目指し、ECDIS上または連接しているPC上でこのコードを実行し、**船舶乗員にとって有効かどうか***

・将来的な展望
(1) 船舶以外のインターネットが使用できる環境への活用が期待できる。
(2) 無人運航状態でもスプーフィングの検知を陸上に信号として送ることが可能ではないか？