

TLIFES における加速度センサを用いた乗車判定方式の提案

Proposal of Entraining Recognition Using Acceleration Sensor in TLIFES

丸山敦志 渡邊晃

(名城大学)

1 はじめに

少子高齢化と核家族化により高齢者の徘徊行動や孤独死などが問題視されている。そこで、我々は、スマートフォンの通信機能とセンサ機能を活用し、見守る側(家族や地域の人など)と見守られる側(高齢者や子どもなど)で位置情報やユーザの行動状態などの情報を共有することにより、住民が安心して生活できる社会を作るシステムとして統合生活支援システム TLIFES(Total LIFE Support system) [1] を提案している。

しかし、初期の TLIFES では行動判定に消費電力の大きい GPS や Wi-Fi の情報を利用しており、スマートフォンの稼働時間の短さが大きな課題となっていた。また、GPS や Wi-Fi の情報は取得できない場所が多く存在し、それに伴う誤判定の多さも課題となっていた。そこで、現在は比較的消費電力が少なく、場所に依存せず行動判定が可能な加速度センサを用いた行動判定が検討されている。しかし、乗車判定が未検討であり、見守りシステムとして大きな課題となっている。

そこで、本稿ではこの課題を解決するため、加速度センサのみを利用した乗車判定方式を提案する。

2 現状の TLIFES の行動判定方式と問題点

TLIFES では、関係する人全員がスマートフォンを所持することを前提とする。スマートフォンに搭載されている様々なセンサから情報を取得し、ユーザの行動判定などを行い定期的にサーバへ報告する。センサ情報の取得には、GPS や Wi-Fi、加速度センサなどを用いる。

初期の TLIFES では、GPS や Wi-Fi の情報をもとにユーザの移動を検出し、行動判定を行っていた。しかし、GPS や Wi-Fi に対する消費電力が大きく、スマートフォンの稼働時間に大きな影響を与えていた。また、GPS や Wi-Fi を受信できない電車内などにおいては行動判定が出来ず、誤判定の原因になっていた。

そこで、加速度センサのみを利用してユーザの移動を検出し、移動したと判断したとき、GPS で位置情報を取得する方法に変更した。これにより、GPS の起動を抑え、Wi-Fi の起動にかかる消費電力を無くすことに成功した。行動判定においては実用性を考慮し、放置中、歩行中、乗車中、静止中の4つを判別することとし、ユーザの行動を把握しやすいようにした。

しかし、現状の TLIFES では乗車判定が実装されていないため、放置中、歩行中以外はすべて静止中と判定されている。そこで、乗車中と静止中を判別するための乗

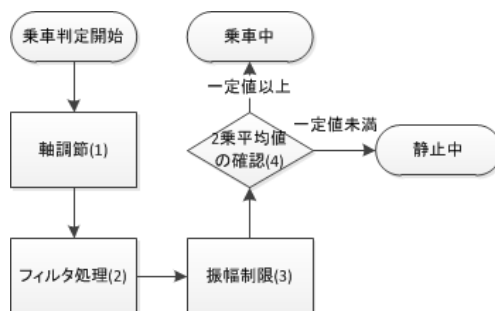


図1 提案する乗車判定方式

車判定を実装する必要がある。

3 乗車判定方式の提案

車や電車などに乗車しているときに加速度センサで高周波の振動を連続的に観測することが出来る。これを利用して、ユーザが何らかの乗り物に乗車しているかどうかの判定を行う。

図1に乗車判定の処理手順を示す。以下に示す番号は図1内の番号に対応している。

(1) 軸調節の処理

加速度センサから得られる情報には、端末の向きや個体差による軸のずれが生じる。そこで、判定に利用する加速度値の平均を算出し、それぞれの加速度値から減算することにより振動の中心を0軸に調整する。

(2) フィルタ処理

車や電車などに乗車しているときの高周波の振動を観測しやすくするため、ユーザの身体の揺れによって生じる低周波の振動をフィルタ処理によって除去する。ここでは、加速度値を下記の式(1)、式(2)で表されたフィルタに通し、低周波の波を除去する。式(1)はLPF(Low Pass Filter)で、 $g(i)$ はLPF通過後の加速度値、 $g(i-1)$ は一つ前のLPF通過後の加速度値、 $x(i)$ は軸調節後の加速度値である。式(2)の $acceleration(i)$ はフィルタ処理後の加速度値を表しており、カットオフ周波数が0.5HzのHPF(High Pass Filter)である。

$$g(i) = 0.9 * g(i-1) + 0.1 * x(i) \quad (1)$$

$$acceleration(i) = x(i) - g(i) \quad (2)$$

(3) 振幅制限の処理

ユーザが立つ、座る、蹴くなどした際に突発的な振動が生じる。この振動は、乗車中と静止中を判定

する際に誤判定の原因となるため、振幅制限の処理によりこれを除去する。しかし、加速度センサから得られる情報には個体差があるうえ、乗車している乗り物によって加速度値が大きく異なる。そこで本提案では閾値を動的に調整する。乗車判定に用いる加速度値のばらつきを見て閾値を動的に決定することにより、端末による個体差や乗り物による制限を無くす。

(4) 2乗平均値による判定

加速度値の2乗平均値を算出し、一定値以上の場合、ユーザが何らかの乗り物に乗車していると判断し、乗車中と判定する。2乗平均値が一定値未満の場合、ユーザはスマートフォンを所持しているが静止していると判断し、静止中と判定する。

4 実装

既存の TLIFES を拡張し、乗車判定を行うモジュールをプロトタイプとして実装した。実装内容としては、乗車判定に利用する加速度合成値の取得、取得した加速度値に対して軸調節、フィルタ処理、振幅制限を行う処理、処理後の加速度値の2乗平均値を算出し、乗車中か静止中の判定を行う処理を Java 言語を用い、Android アプリケーションとして実装した。

5 評価

5.1 評価方法

提案方式の有用性を確認するために乗車時(地下鉄線、近鉄線、JR線、車)と静止時において乗車判定を行い、その認識率で評価を行った。使用する端末は Galaxy Nexus を使用し、端末を所持する場所はズボンのポケットと限定して判定を行った。また、静止時においては被験者3名に研究室内でいつも通りの行動をしてもらい判定を行った。乗車中の判定は電車や車に乗車し、動き出してから目的地に到着するまでを乗車中と判定する。そのため、電車に乗車し、途中の駅に停車した場合でも下車する駅でない場合は、「乗車中」を正判定とする。

5.2 判定結果

表1に乗車時と静止時において乗車判定を行った際のサンプル数と認識率を示す。

JR 乗車時、近鉄乗車時、車乗車時においては認識率が90%近く出ており、高い認識率で判定できていることがわかる。しかし、地下鉄乗車時においては、40%ほどで、他の状態の乗車時よりも低い値となった。また、静止時においては、81.45%の認識率という結果になった。

5.3 考察

表1の結果から地下鉄乗車時を除き認識率が80%を超え、提案方式の有用性を確認した。

表1 それぞれの状態において乗車判定を行った結果

	サンプル数	正判定数	誤判定数	認識率 [%]
地下鉄	51	22	29	43.14
JR	167	149	18	89.22
近鉄	55	53	2	96.36
車	72	72	0	100
合計(乗車)	345	296	49	85.80
静止	124	101	23	81.45

地下鉄乗車時の認識率が低くなった理由は、他の乗車時に比べ、振動が少なかったためと考えられる。また、地下鉄は、他の電車に比べ、駅と駅が近く、走行速度が遅いのに加え、頻繁に駅に停車することが原因であると考えられる。そのため、今回は2分間隔で判定を行っていたが、更に細かく区切って判定を行うことで、認識率が上昇するのではないかと考える。

静止時における誤判定の原因として判定に利用する加速度値に歩行時の加速度値が含まれていることがあげられる。歩行時の加速度値は乗車時の加速度値よりも大きく、ある程度は振幅制限の処理により除去することが出来るが、歩行している時間が長いと振幅制限時の閾値が上昇してしまい除去しきれないことが確認できた。そこで、TLIFESで行っている歩行判定を利用し、歩行時の加速度値を乗車判定から除くことで上記の課題を解決できるのではないかと考える。

さらに、全体として判定結果が信頼できるかを確認し、補正をかけることで更に認識率が上昇するのではないかと考える。例えば、6分の間に「静止→乗車→静止」と判定された場合、ごく短い時間で乗り物に乗車し、下車することは考えにくいので、「静止→静止→静止」と補正し、最終的な判定結果として決定する。

6 まとめ

本稿では、加速度センサによるユーザの乗車判定の手法を提案した。また、提案方式の認識率を算出することにより有用性を確認した。今後は誤判定の原因を調査し、更なる改善を図る。

参考文献

- [1] 大野 雄基, 他 : TLIFES を利用した徘徊行動検出方式の提案と実装, 情報処理学会論文誌コンシューマ・デバイス&システム (CDS), Vol.3, No.3, pp.1-10, July.2013.
- [2] 加藤 大智, 他 : TLIFES における省電力化を目的とした位置測位手法の提案と実装, 研究報告コンシューマ・デバイス&システム (CDS), Vol.2013-CDS-6, No.13, pp.1-6, Jan.2013.

TLIFESにおける 加速度センサを用いた 乗車判定方式の提案

丸山敦志 渡邊晃
名城大学理工学部

研究背景

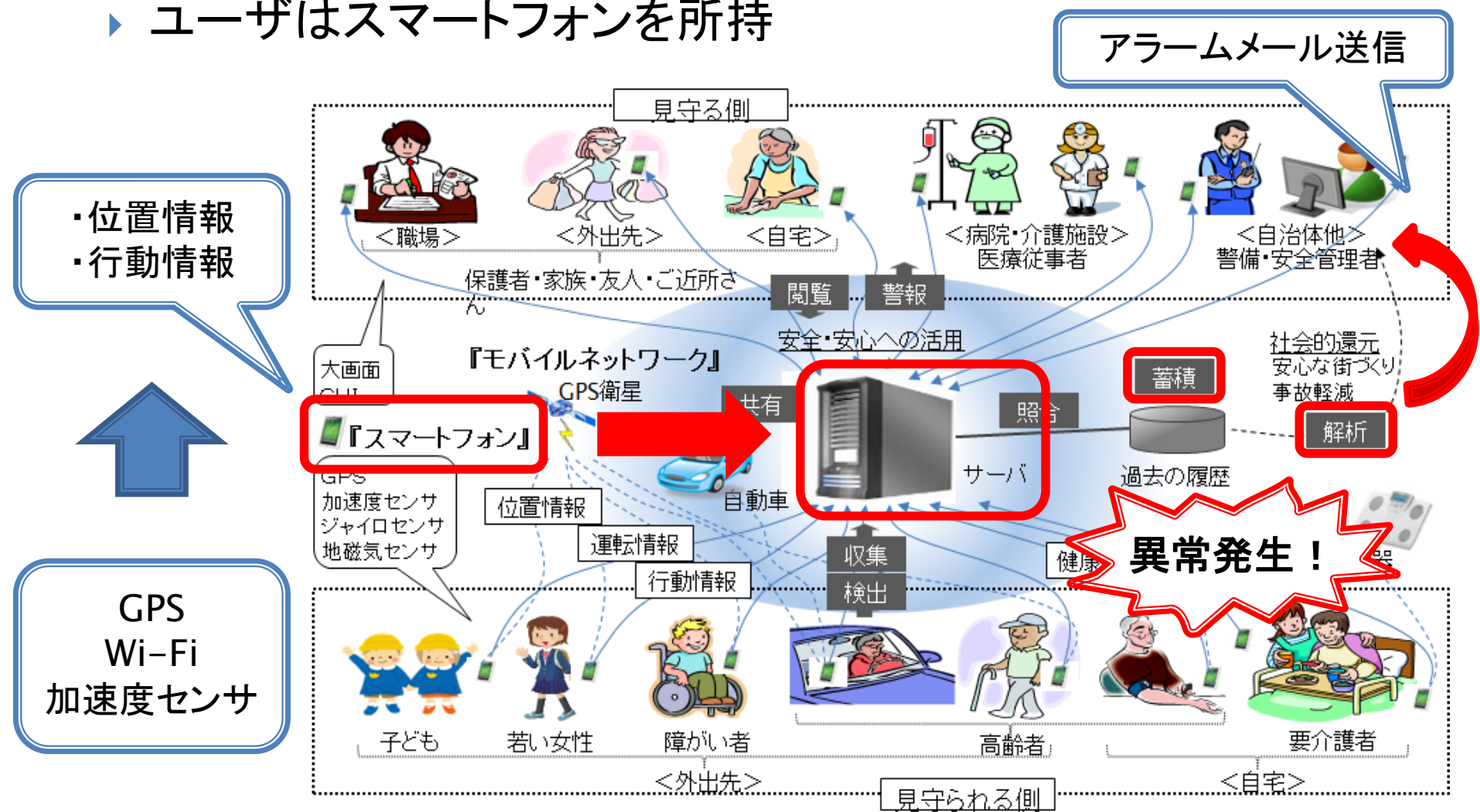
- ▶ 少子高齢化, 核家族化
 - 一人暮らしの高齢者増加
 - 高齢者の徘徊行動が社会問題に
- ▶ スマートフォンの普及
 - GPSやWi-Fi, 加速度センサなど多くの機能を搭載



スマートフォンを利用した見守りシステム
TLIFESを提案

TLIFESの概要

- ▶ ユーザはスマートフォンを所持



初期の行動判定方式の課題

- ▶ 消費電力が大きいGPS, Wi-Fiを利用
 - 稼働時間の低下
- ▶ 誤判定が多い
 - 電車内など, Wi-FiやGPSの情報が取得できない場所が多い
⇒ 正確な行動判定が行えないため, 誤判定を引き起こす



GPSやWi-Fiから得られる位置情報は行動判定に適さない

現状の行動判定方式

▶ 特徴

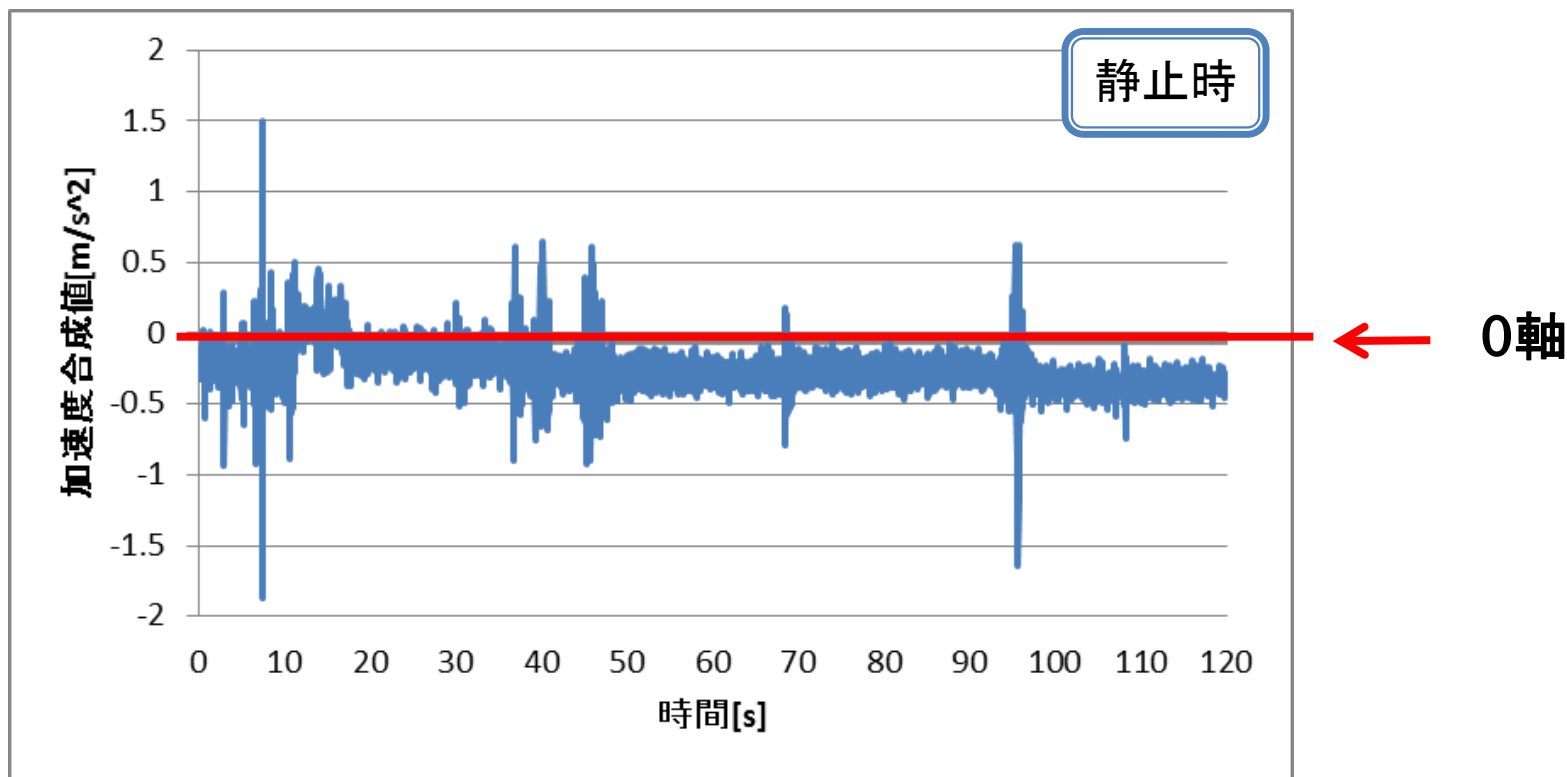
- 行動判定において加速度センサのみを使用
 - ・ Wi-Fi・GPSによる位置情報は使用しない
 - ⇒消費電力の低減
 - ⇒場所に依存しない行動判定が可能
- 判定する行動は実用性を考慮し,「放置中」「歩行中」「乗車中」「静止中」の4つ

▶ 課題

- 乗車中と静止中を判別するための乗車判定が未検討
 - ⇒加速度センサを用いた乗車判定方式を提案する

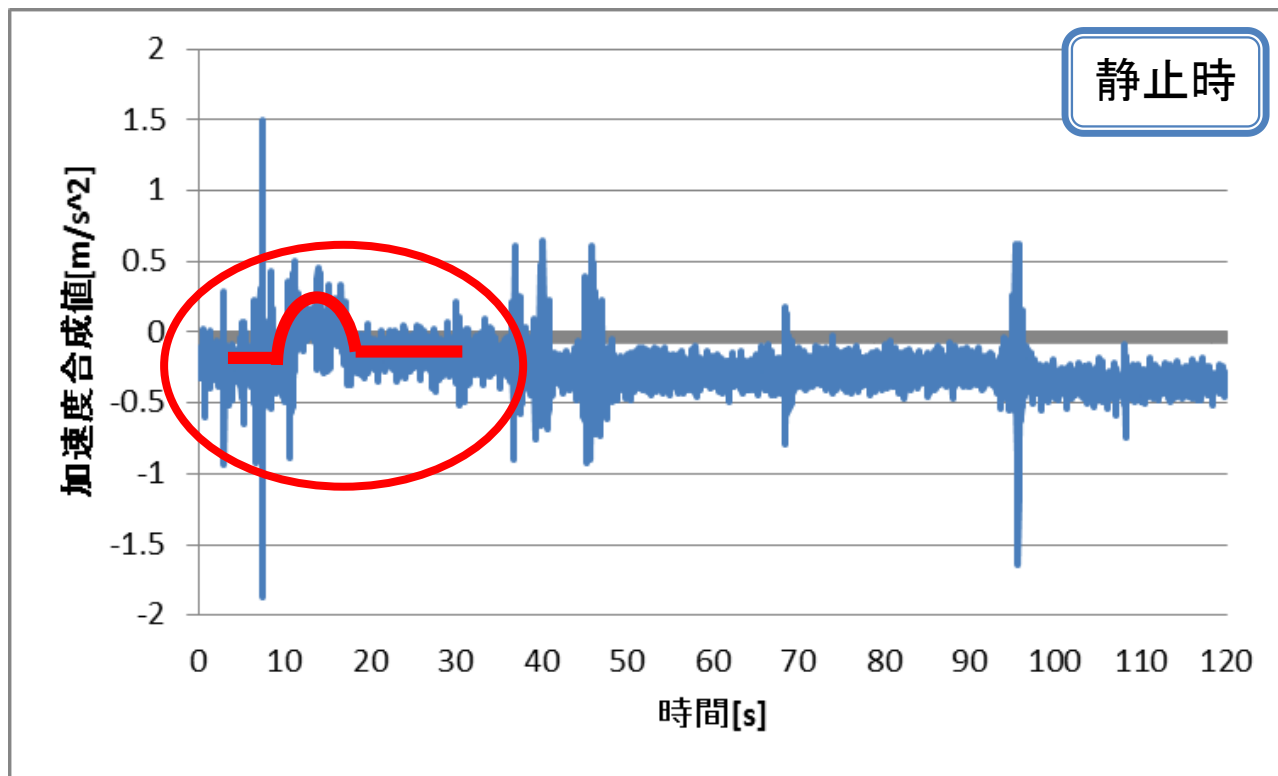
静止時の加速度値の特徴

- ▶ 静止時(椅子に座って作業をしている状態)
 - スマートフォンの向きによる軸のずれ
⇒ スマートフォンの向きや機種により、ずれの向きや大きさが異なる



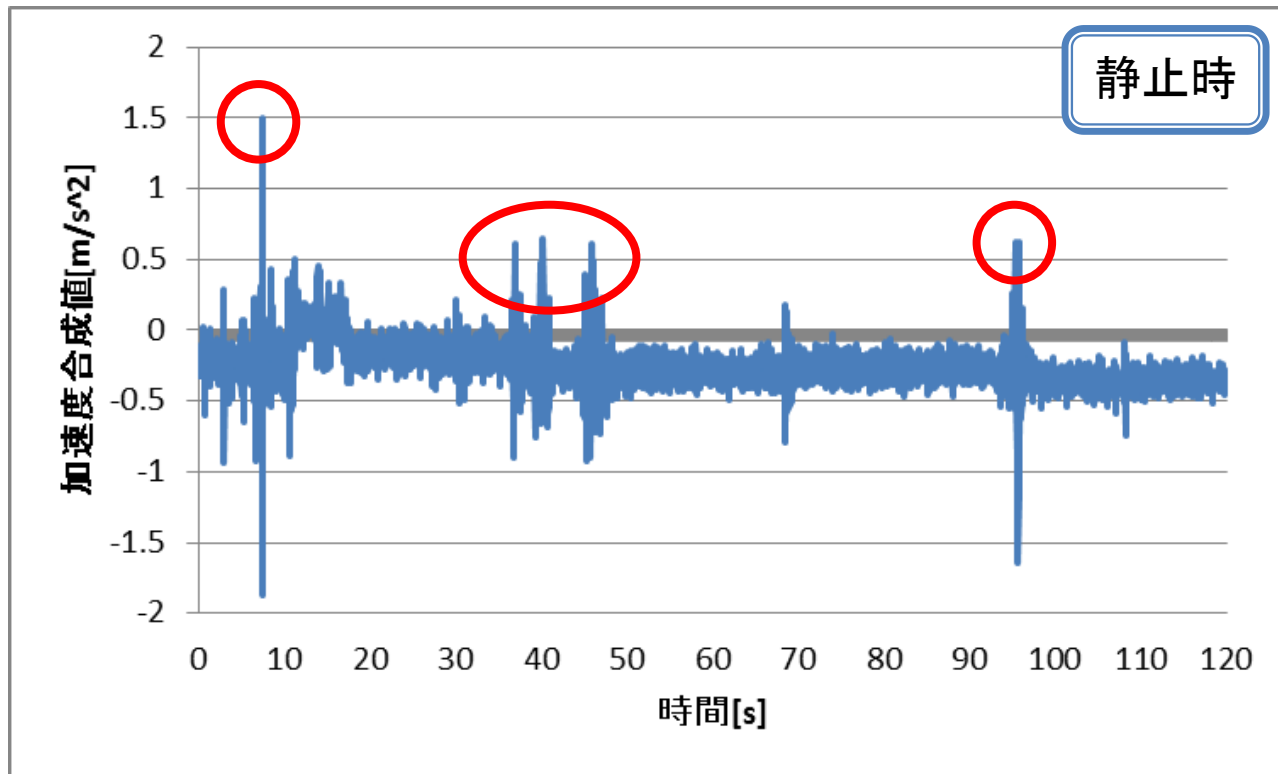
静止時の加速度値の特徴

- ▶ 静止時(椅子に座って作業をしている状態)
 - 身体の揺れによる振動
 - ⇒ユーザの身体の揺れなどにより低周波の波が加わる



静止時の加速度値の特徴

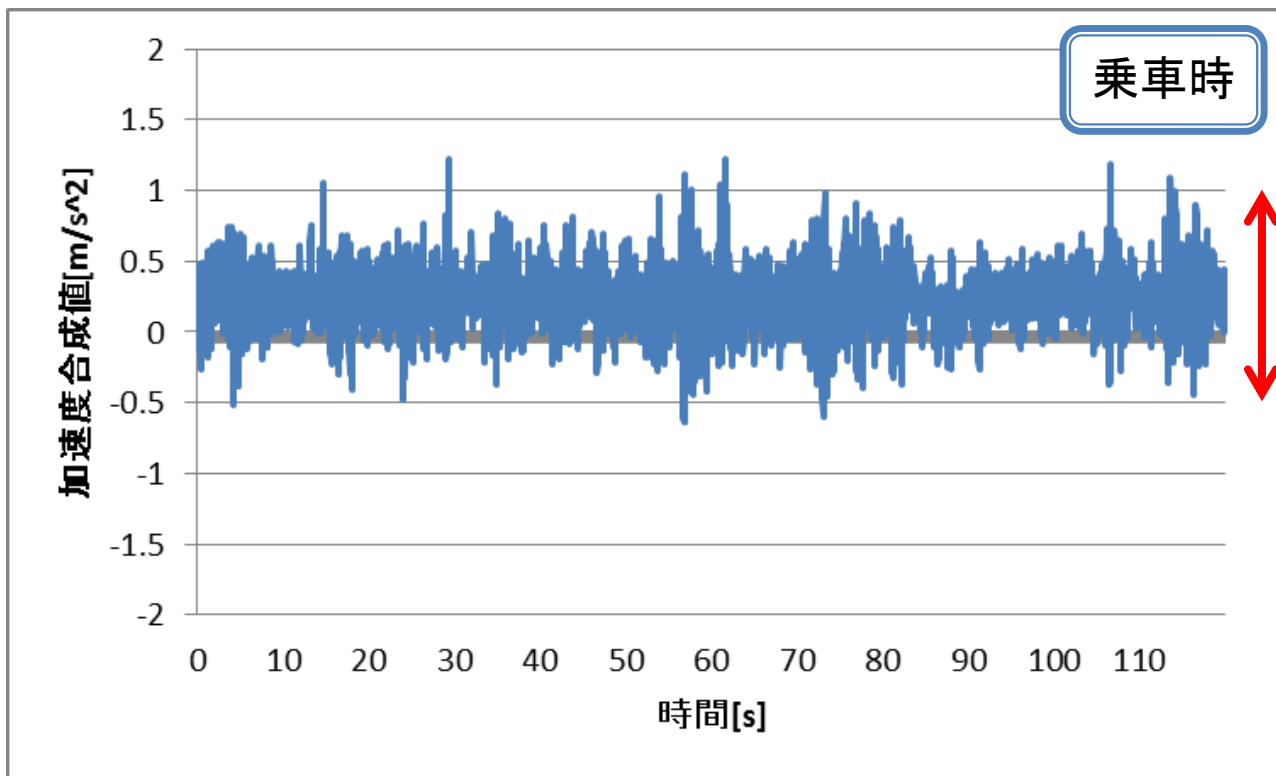
- ▶ 静止時(椅子に座って作業をしている状態)
 - 突発的な振動
 - ⇒ユーザが立つ, 座る, ぶつかるなどした際に発生



乗車時の加速度値の特徴

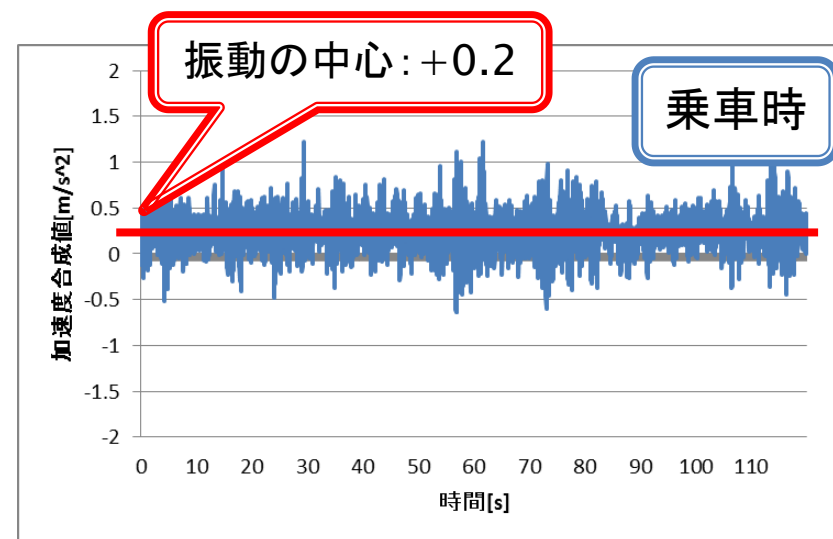
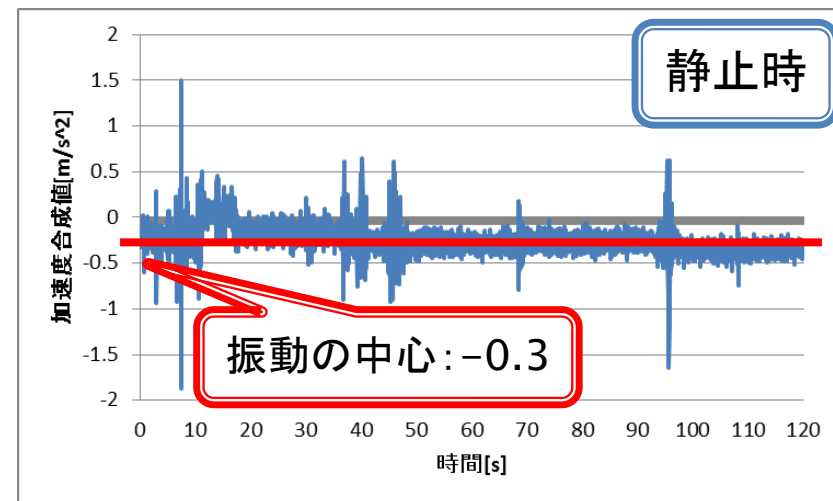
▶ 乗車時(JR線乗車時)

- スマートフォンの向きによる軸のずれ＋身体の揺れによる振動
＋突発的な振動＋乗車時特有の振動



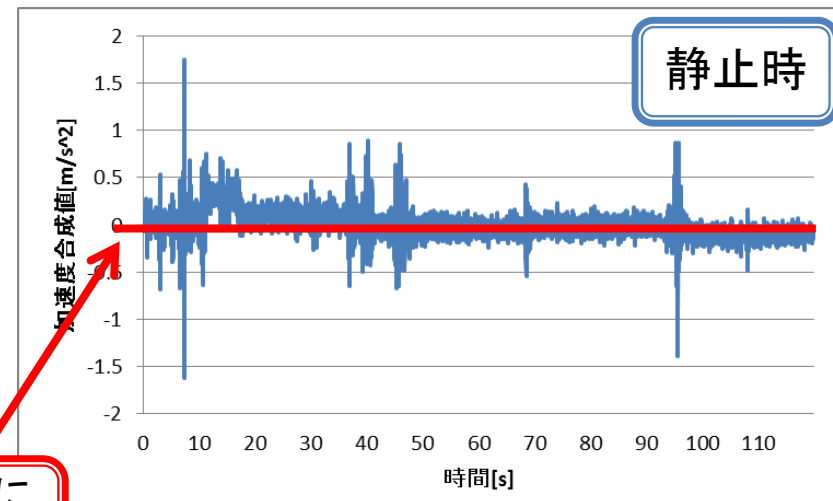
軸調節の処理

- ▶ 2分間の加速度値の平均値を
加速度値から引くことで振動の
中心を0軸に調整
⇒スマートフォンの姿勢による
軸のずれを除去

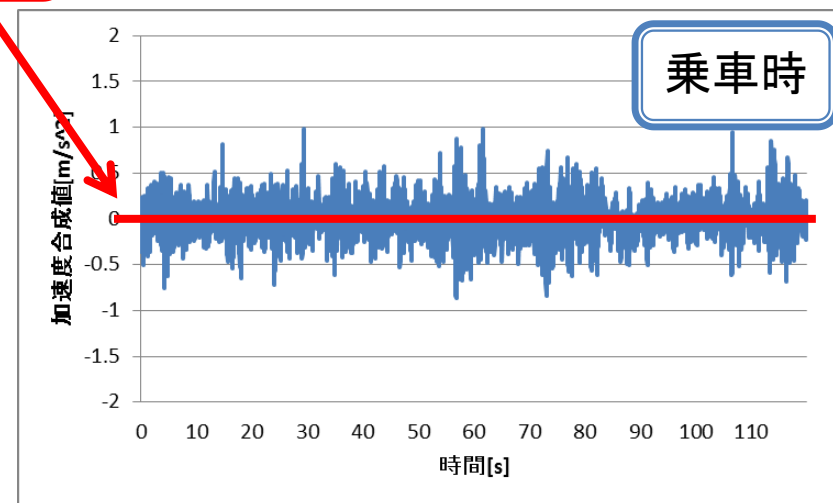


軸調節の処理

- ▶ 2分間の加速度値の平均値を
加速度値から引くことで振動の
中心を0軸に調整
⇒スマートフォンの姿勢による
軸のずれを除去



0軸中心に
振動



フィルタ処理

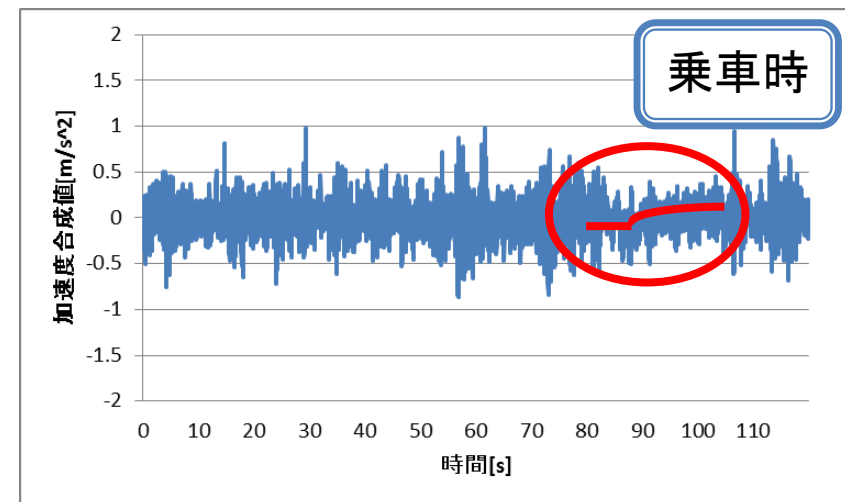
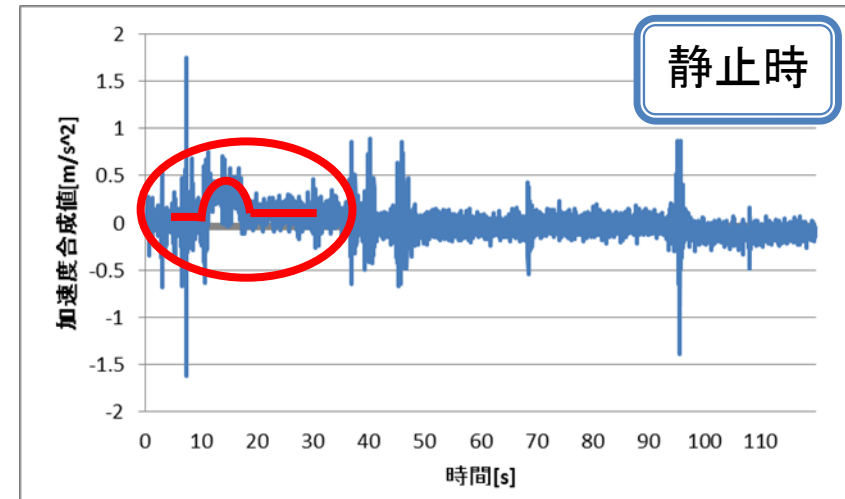
- ▶ HPFをかけることで低周波の振動を除去
⇒ 身体の揺れなどによる
低周波の振動を除去

HPF※式

$$g(i) = 0.9 * g(i - 1) + 0.1 * x(i)$$
$$acceleration(i) = x(i) - g(i)$$

$g(i)$: i 番目のLPF*処理後の加速度値
 $x(i)$: i 番目の実加速度値
 $acceleration(i)$: i 番目のフィルタ処理後の加速度値

※High Pass Filter
* Low Pass Filter



フィルタ処理

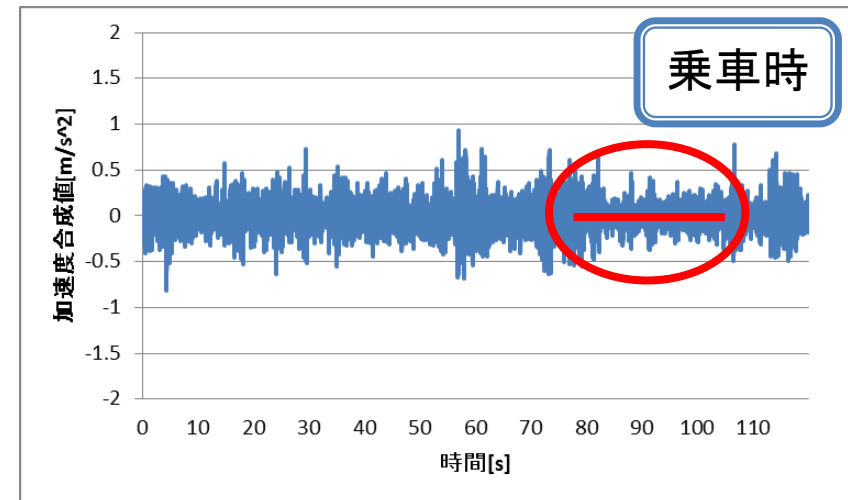
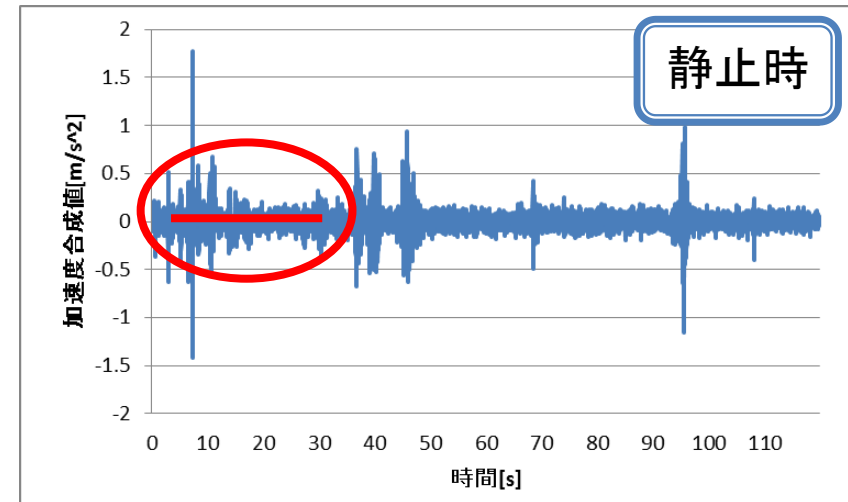
- ▶ HPFをかけることで低周波の振動を除去
⇒ 身体の揺れなどによる
低周波の振動を除去

HPF※式

$$g(i) = 0.9 * g(i - 1) + 0.1 * x(i)$$
$$acceleration(i) = x(i) - g(i)$$

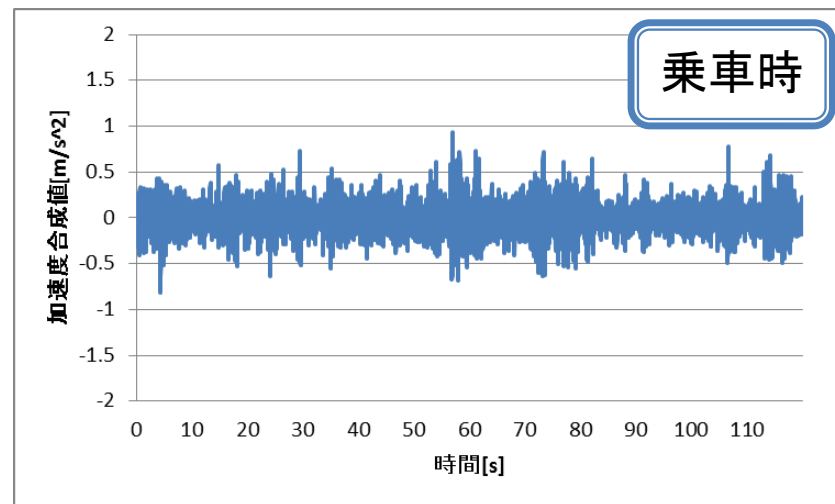
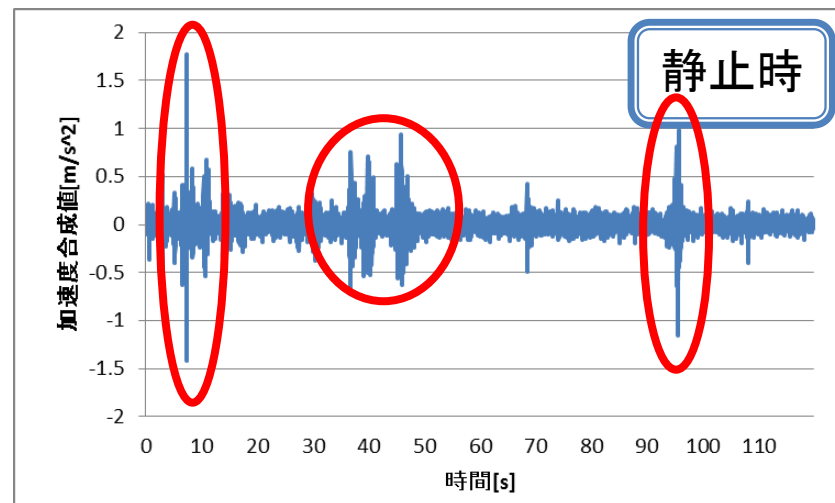
$g(i)$: i 番目のLPF*処理後の加速度値
 $x(i)$: i 番目の実加速度値
 $acceleration(i)$: i 番目のフィルタ処理後の加速度値

※High Pass Filter
* Low Pass Filter



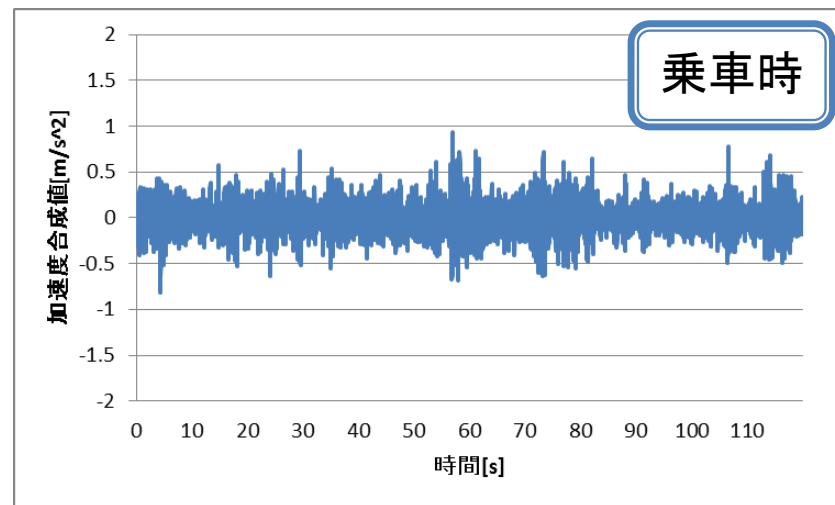
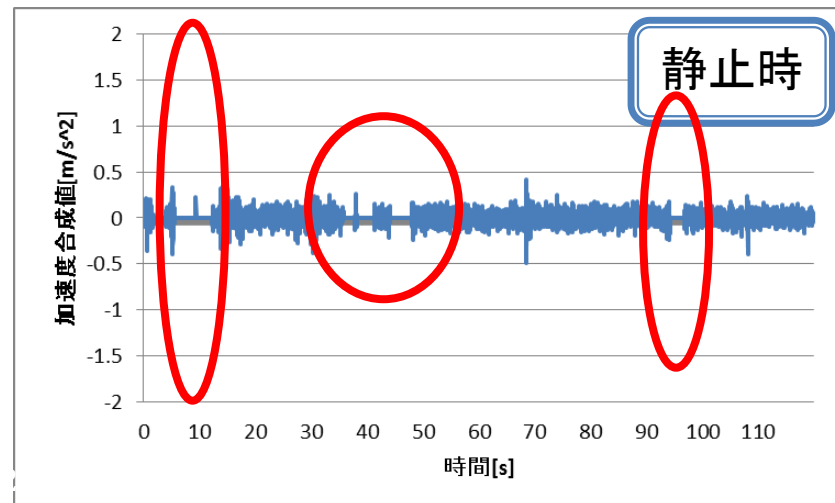
振幅制限の処理

- ▶ 閾値以上の値を検出した場合前後50個のデータを0に書き換える
⇒ 立ったりした際に発生する突発的な振動を除去
- ▶ 振幅制限時の閾値は乗車判定毎にダイナミックに決定
⇒ 乗車判定に用いる加速度値の分散により閾値を決定する



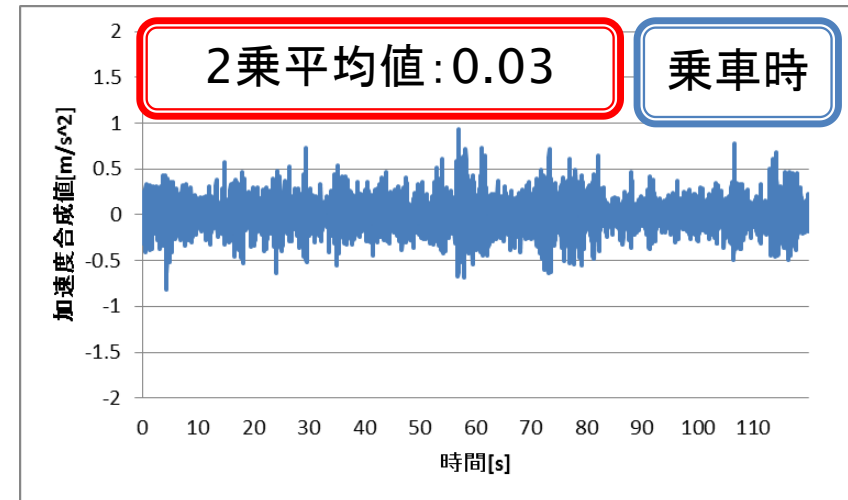
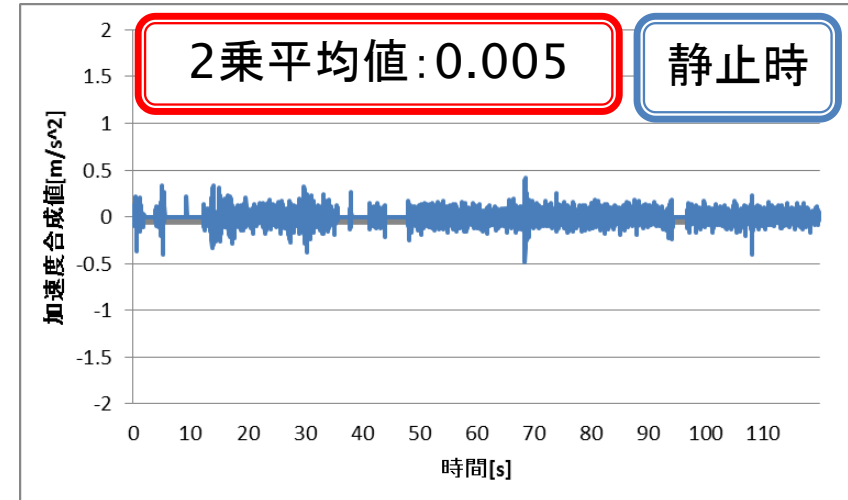
振幅制限の処理

- ▶ 閾値以上の値を検出した場合前後50個のデータを0に書き換える
⇒ 立ったりした際に発生する突発的な振動を除去
- ▶ 振幅制限時の閾値は乗車判定毎にダイナミックに決定
⇒ 乗車判定に用いる加速度値の分散により閾値を決定する



2乗平均値の確認

- ▶ 軸調節, フィルタ処理, 振幅制限後の加速度値の2乗平均値を算出
 - 2乗平均値が0.01以上
⇒「乗車中」
 - 2乗平均値が0.01未満
⇒「静止中」

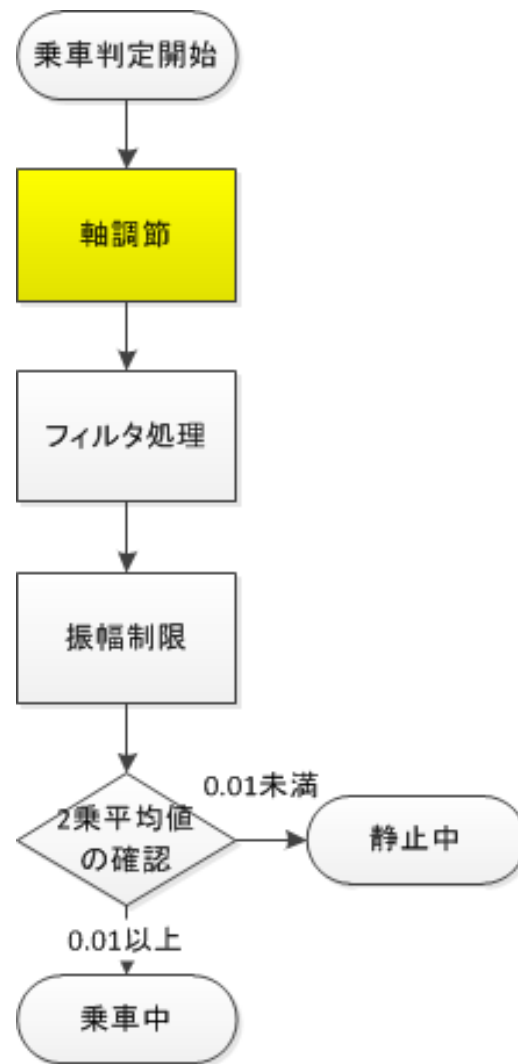


まとめ

- ▶ 加速度センサで観測することが出来る乗車時特有の振動を検出することにより、乗車時と静止時を判別する方式を提案した
- ▶ 今後の予定
 - 乗車判定を行い、認識率の確認を行う
 - 高い認識率を実現するために検討を行う

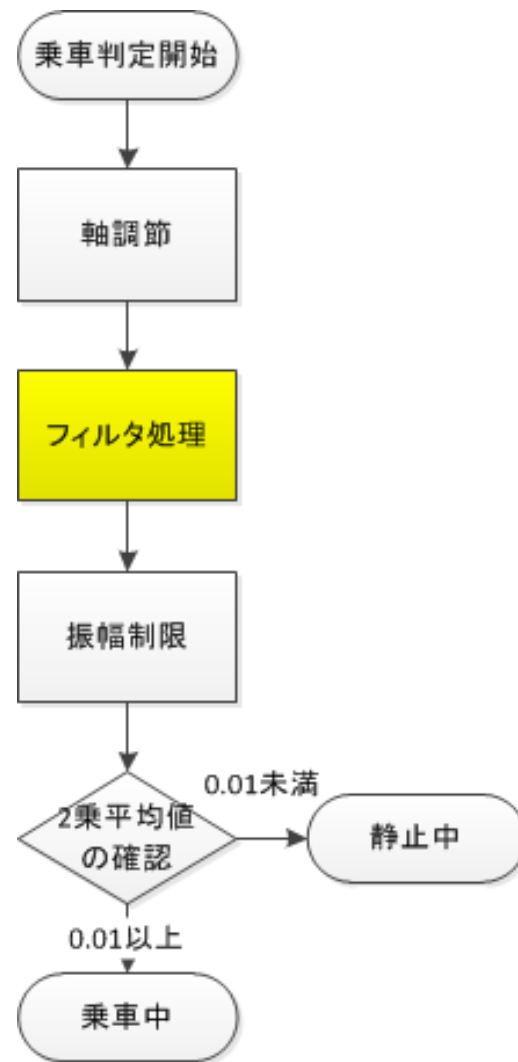
乗車判定フロー

- ▶ 軸調節の処理
 - センサ自体の精度により発生する軸のずれを除去
- ▶ フィルタ処理
 - 身体の揺れなどによって生じる低周波の振動を除去
- ▶ 振幅制限の処理
 - 立つ・座る・つまづくなどによって生じる突発的な振動を除去
- ▶ 2乗平均値により乗車中と静止中を判別



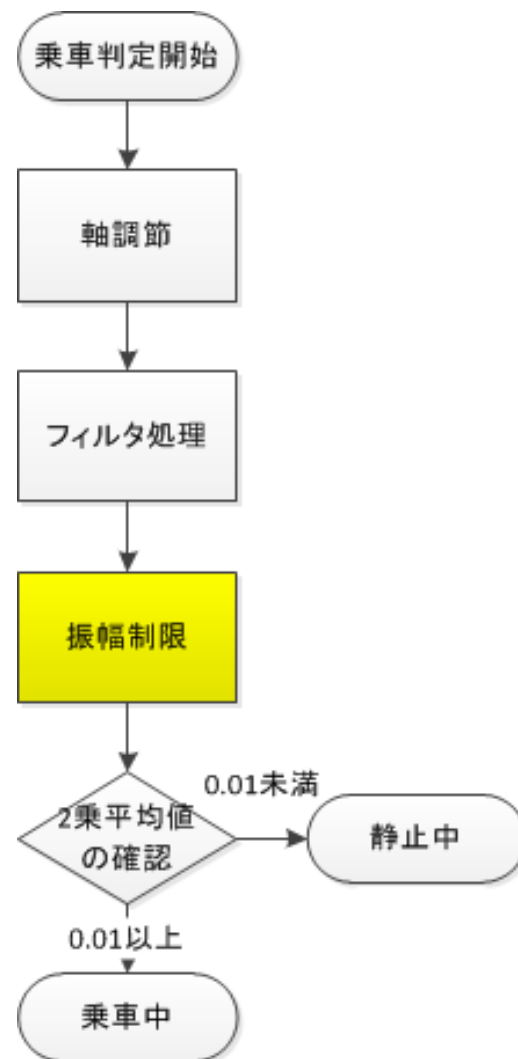
乗車判定フロー

- ▶ 軸調節の処理
 - センサ自体の精度により発生する軸のずれを除去
- ▶ フィルタ処理
 - 身体の揺れなどによって生じる低周波の振動を除去
- ▶ 振幅制限の処理
 - 立つ・座る・つまづくなどによって生じる突発的な振動を除去
- ▶ 2乗平均値により乗車中と静止中を判別



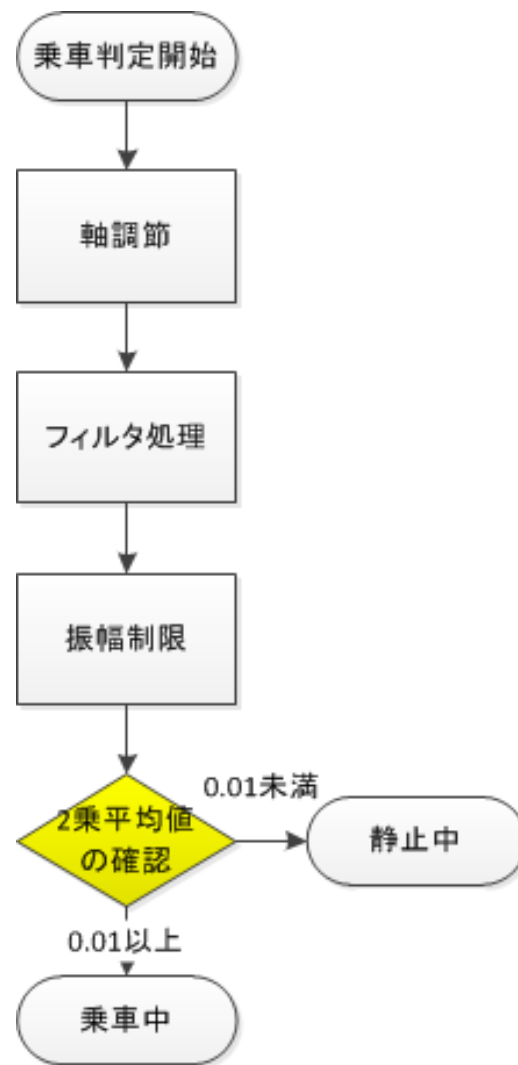
乗車判定フロー

- ▶ 軸調節の処理
 - センサ自体の精度により発生する軸のずれを除去
- ▶ フィルタ処理
 - 身体の揺れなどによって生じる低周波の振動を除去
- ▶ 振幅制限の処理
 - 立つ・座る・つまづくなどによって生じる突発的な振動を除去
- ▶ 2乗平均値により乗車中と静止中を判別



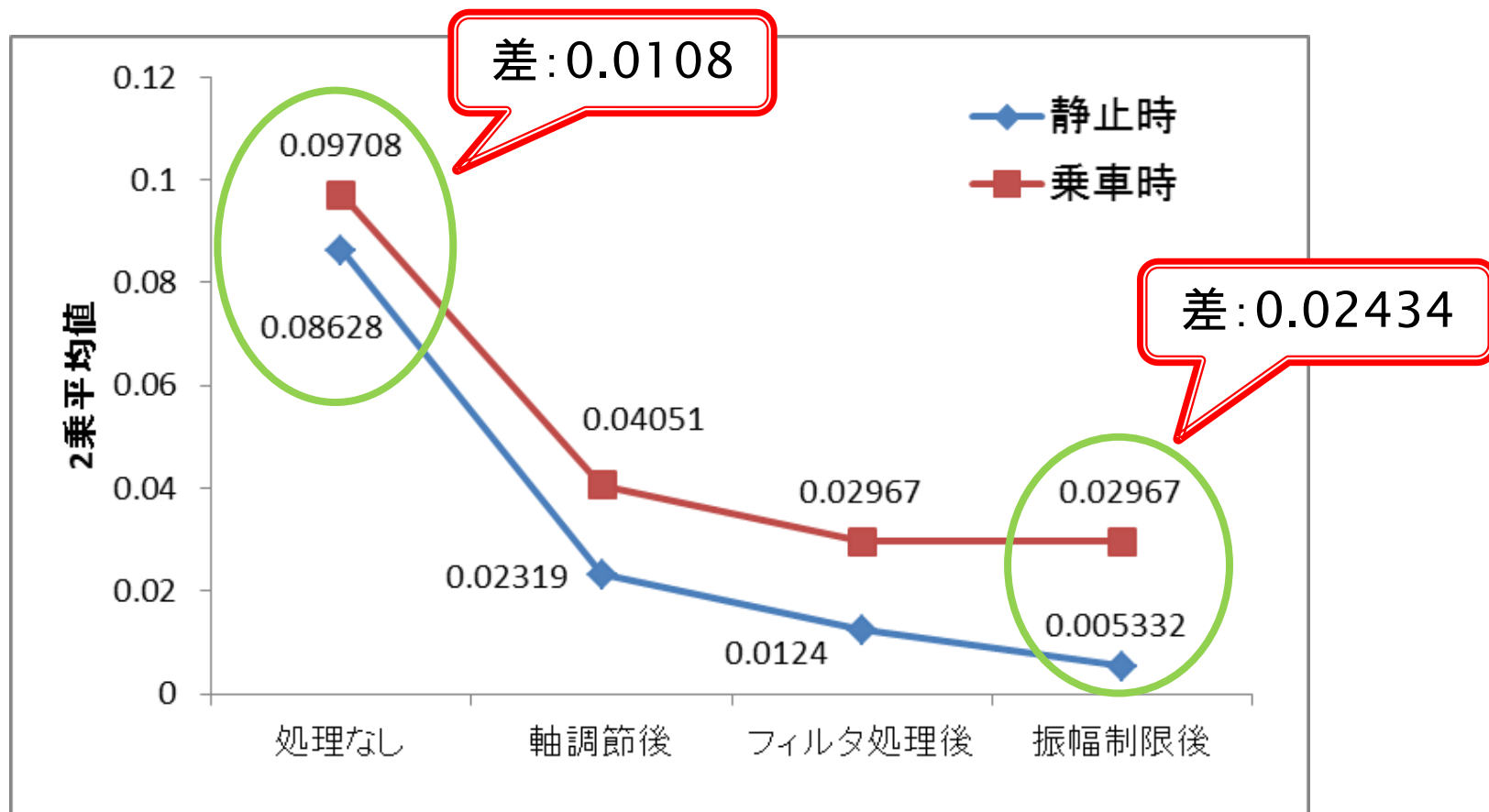
乗車判定フロー

- ▶ 軸調節の処理
 - センサ自体の精度により発生する軸のずれを除去
- ▶ フィルタ処理
 - 身体の揺れなどによって生じる低周波の振動を除去
- ▶ 振幅制限の処理
 - 立つ・座る・つまづくなどによって生じる突発的な振動を除去
- ▶ 2乗平均値により乗車中と静止中を判別



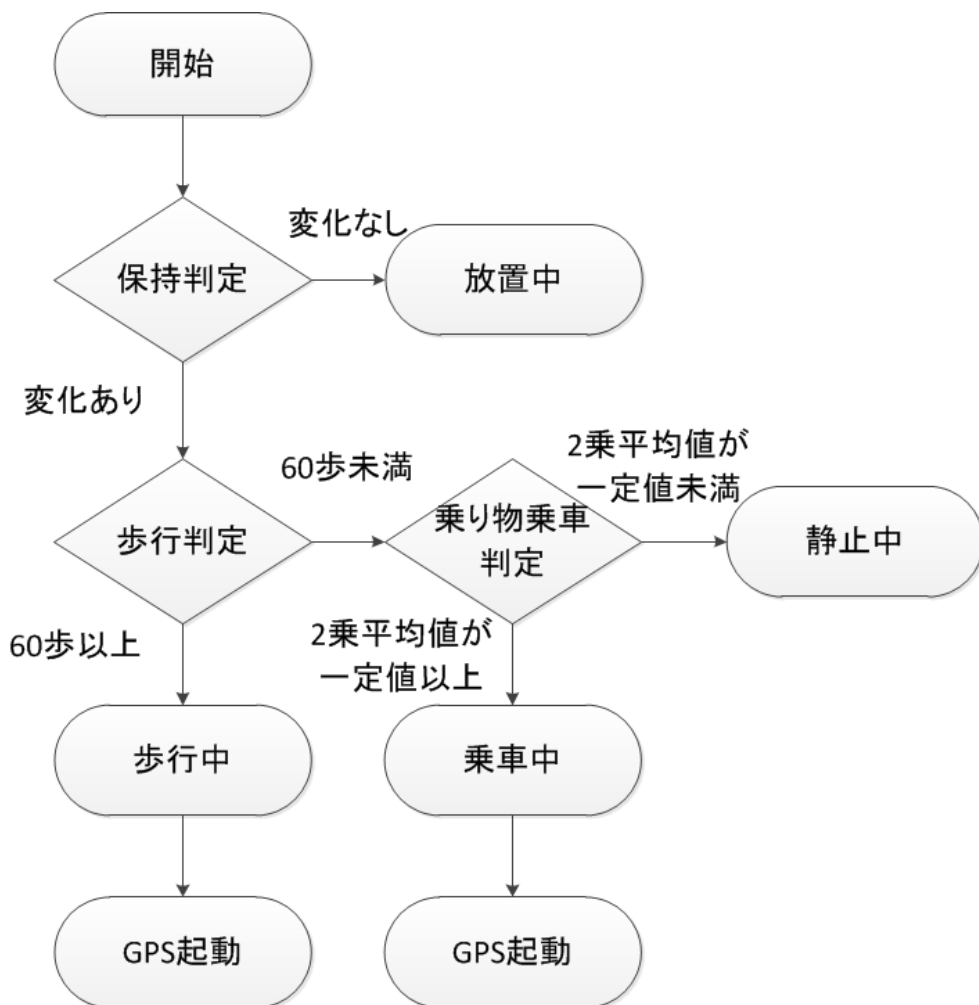
処理毎の2乗平均値の推移

- ▶ 処理毎に静止時と乗車時の2乗平均値の差が大きくなっている



行動判定フロー

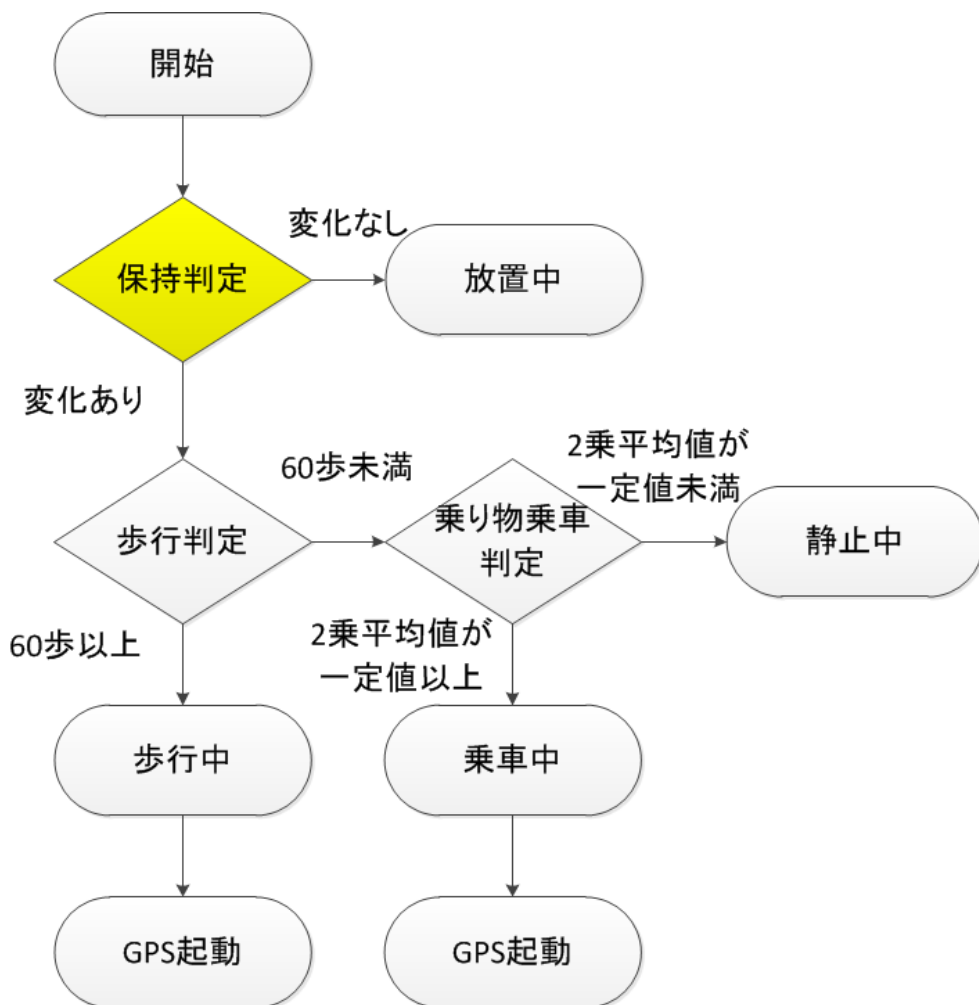
- ▶ 2分毎に判定開始
 - スマートフォンの保持判定
 - 歩数による歩行判定
 - 加速度センサを用いた乗り物乗車判定



行動判定フロー(保持判定)

▶ スマートフォンの保持判定

- 2分間スマートフォンの加速度に変化なし
⇒「放置中」
- 加速度に変化あり
⇒歩行判定を行う



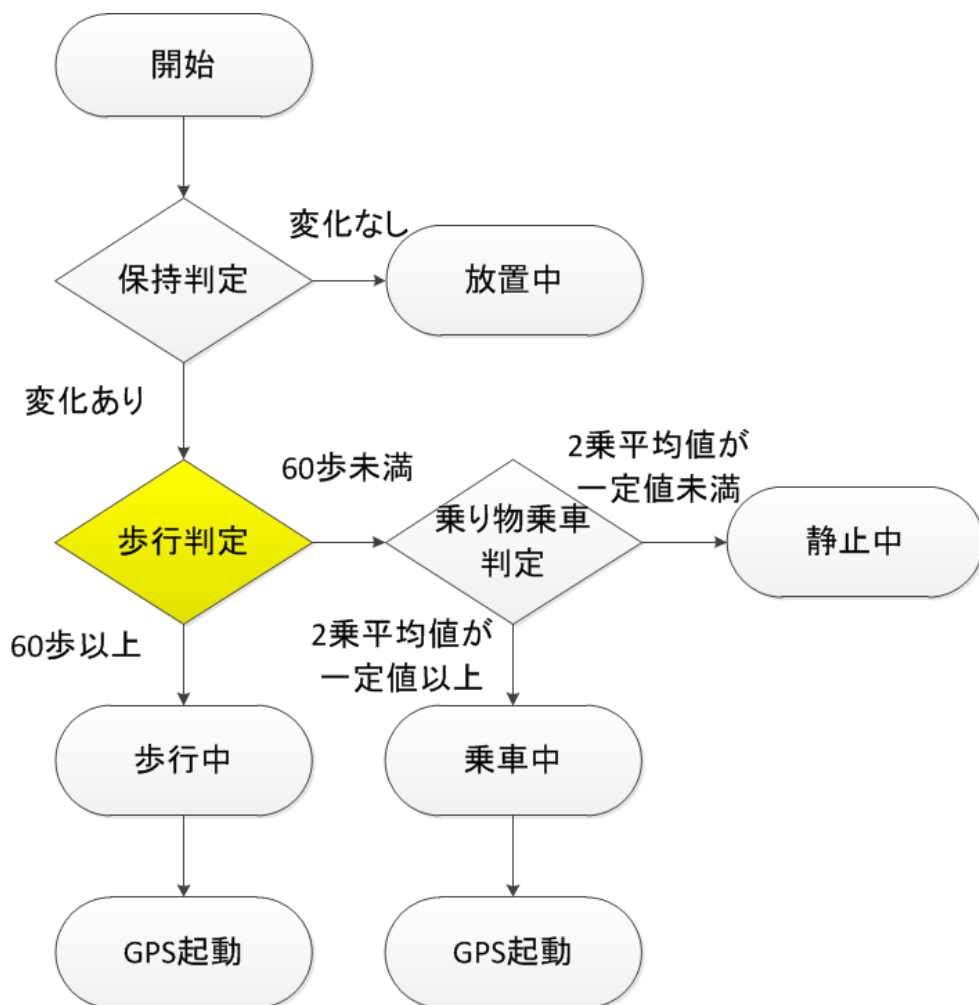
行動判定フロー(歩行判定)

▶ 歩数による歩行判定

- 毎分60歩以上
⇒「歩行中」
⇒GPS起動し、位置情報を更新
- 毎分60歩未満
⇒乗り物乗車判定を行う

※閾値毎分60歩について

人の平均歩数が毎分120歩であり、
判定する時間の半分以上歩いていたら
歩行中と判定するため



行動判定フロー(乗車判定)

▶ 乗り物乗車判定

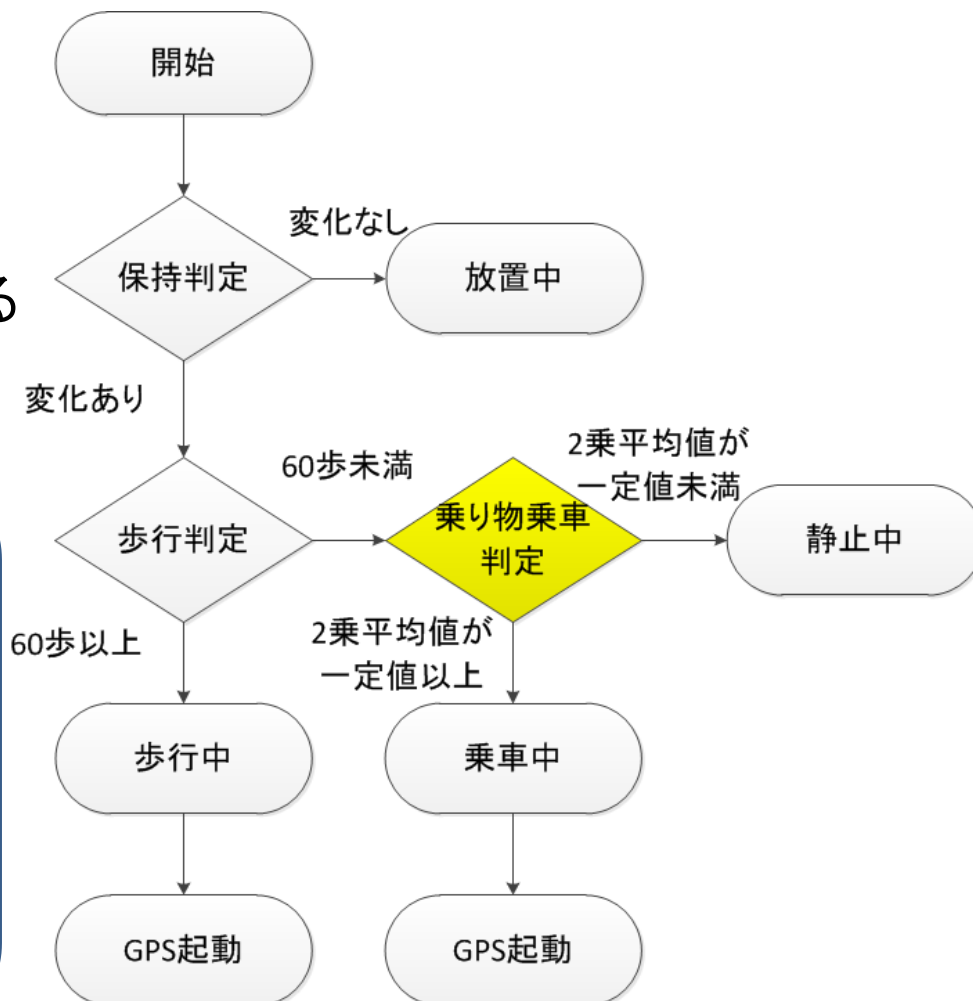
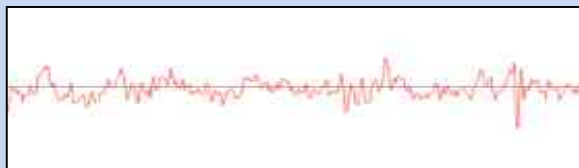
- 加速度データを利用
- 乗り物乗車時に連続で検出できる高周波ノイズを利用

加速度合成値の波形

静止中



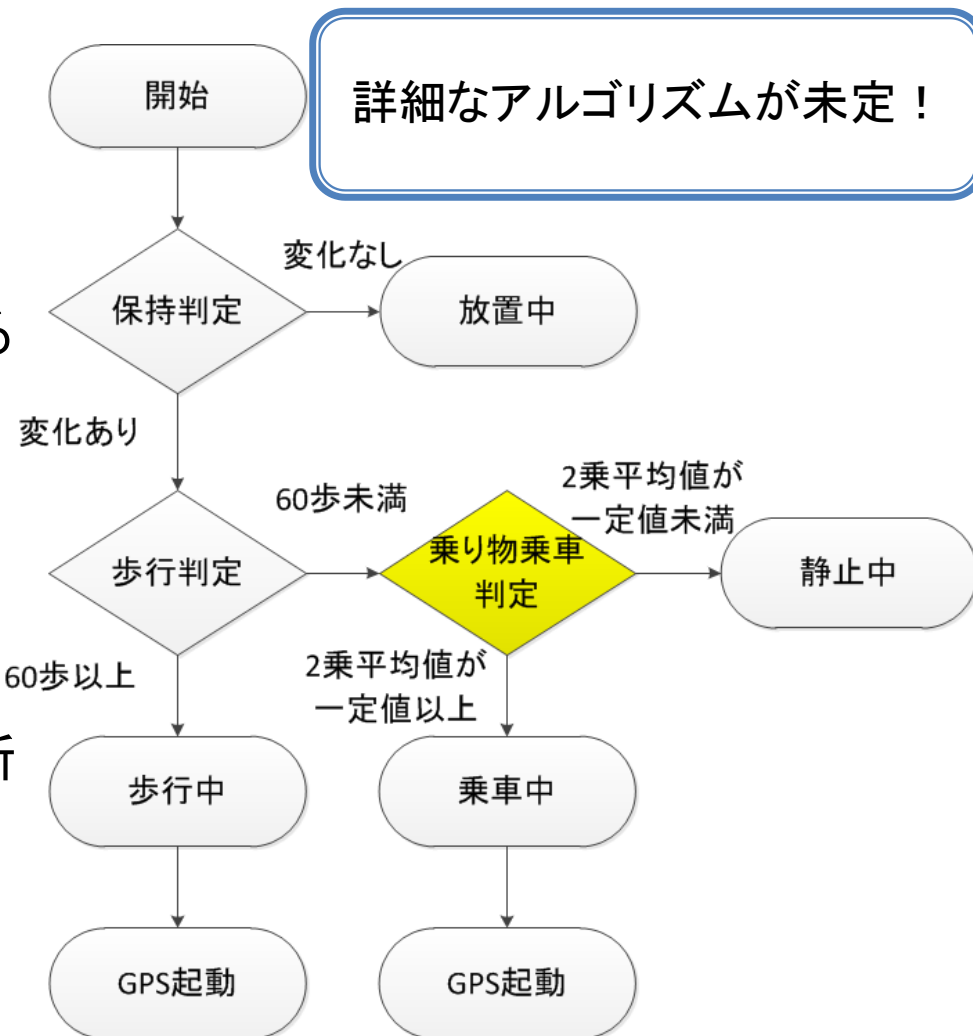
乗り物
乗車中



行動判定フロー(乗車判定)

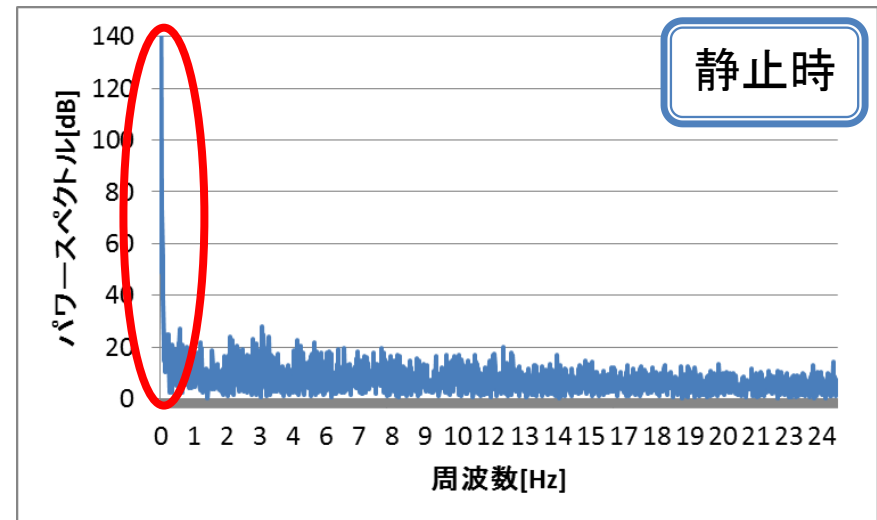
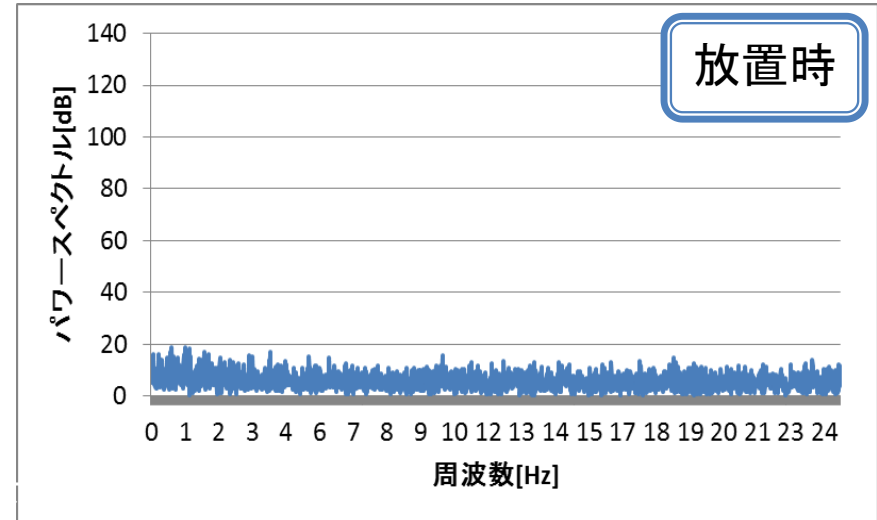
▶ 乗り物乗車判定

- 加速度データを利用
- 乗り物乗車時に連続で検出できる高周波ノイズを利用
- 加速度合成値の2乗平均値が一定値以上
⇒「乗車中」
⇒GPSを起動し、位置情報の更新
- 一定値未満⇒「静止中」



フィルタ処理について

- ▶ 放置時の周波数成分
 - 目立った箇所は無く、
全体的に振動している
- ▶ 静止時の周波数成分
 - 放置時にはない赤枠内の周波数の波が存在する
⇒ 身体の揺れによる低周波の波、または、
突発的な振動である可能性が高い



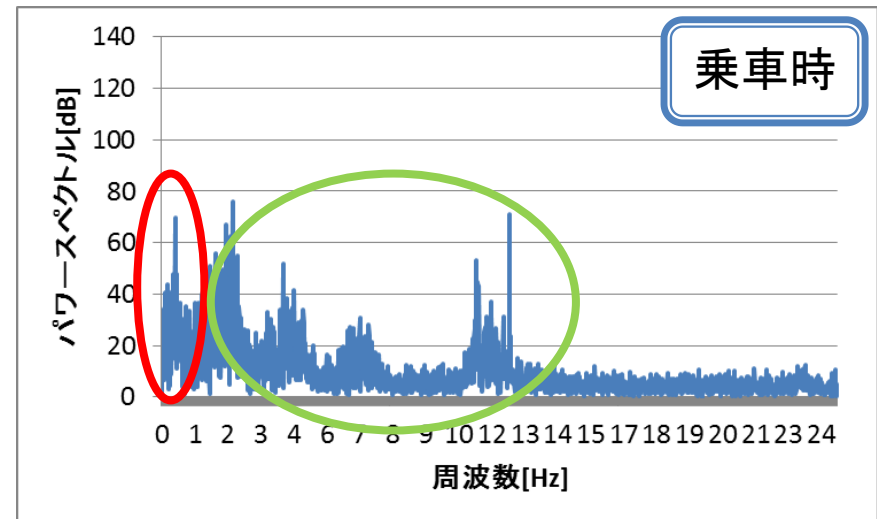
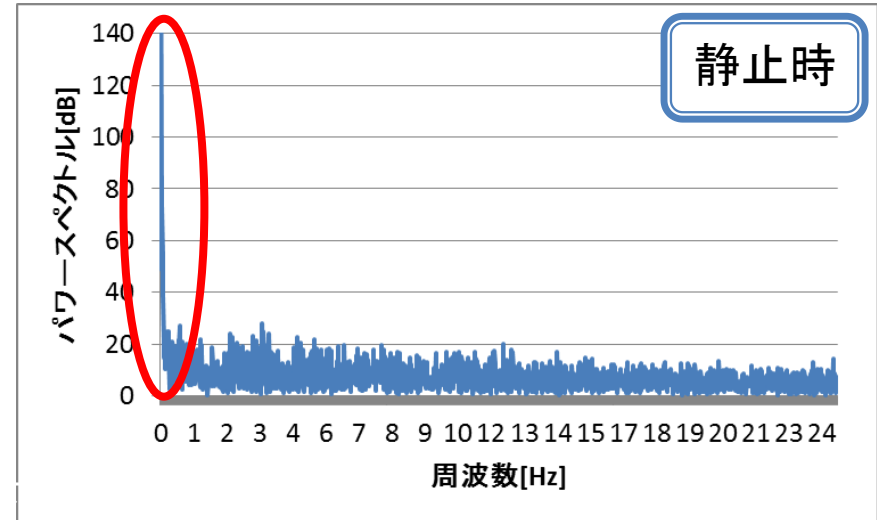
フィルタ処理について

▶ 乗車時の周波数成分

- 静止時同様, 赤枠内の周波数の波が存在する
- 放置時, 静止時にはなかった緑枠内の周波数の波が存在する



赤枠内: 突発的な振動, 低周波の振動
緑枠内: 乗車時特有の振動



フィルタ処理について

▶ HPF

- カットオフ周波数0.5Hz
- LPFをかけ低周波を抽出し、その結果を元のデータから引くことで低周波を除去

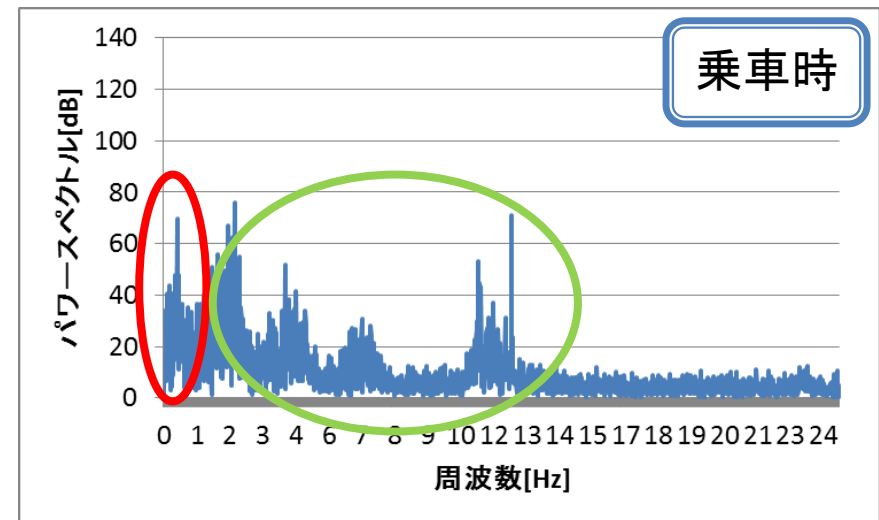
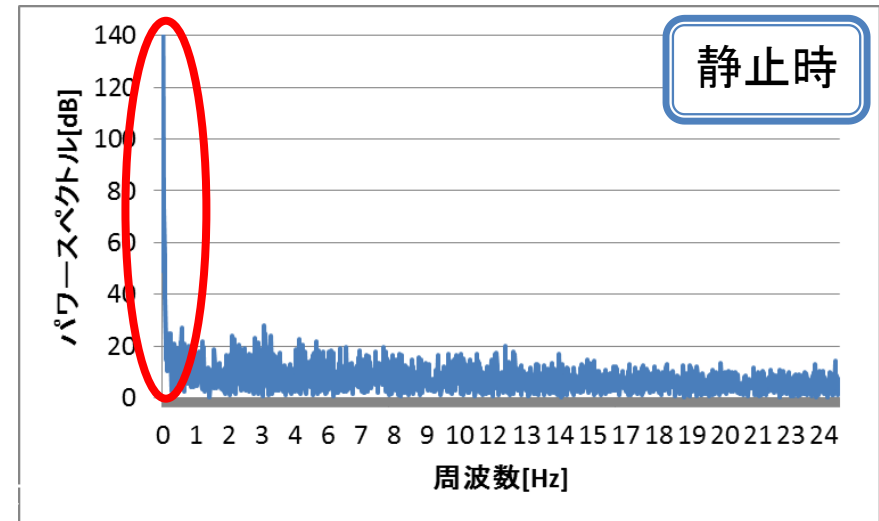
HPF式

$$g(i) = 0.9 * g(i - 1) + 0.1 * x(i)$$
$$acceleration(i) = x(i) - g(i)$$

$g(i)$: i 番目のLPF処理後の加速度値

$x(i)$: i 番目の実加速度値

$acceleration(i)$: i 番目のフィルタ処理後の加速度値



フィルタ処理について

▶ HPF

- カットオフ周波数0.5Hz
- LPFをかけ低周波を抽出し、その結果を元のデータから引くことで低周波を除去

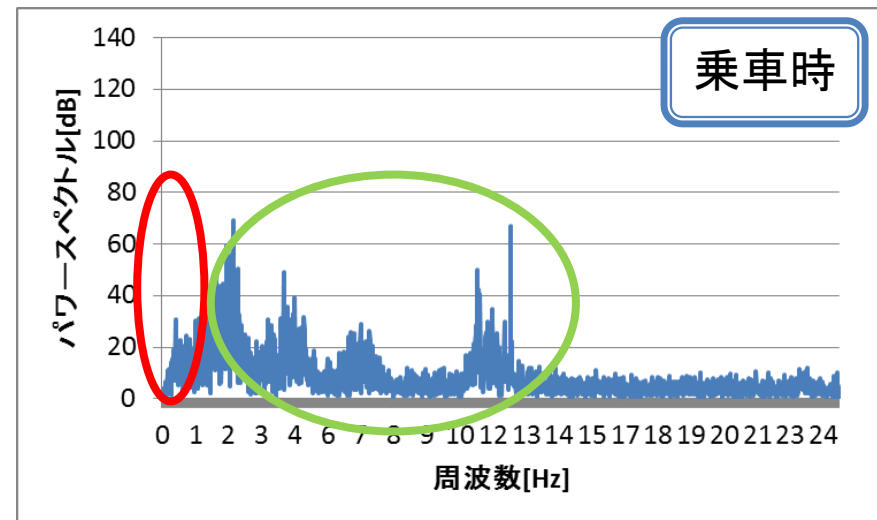
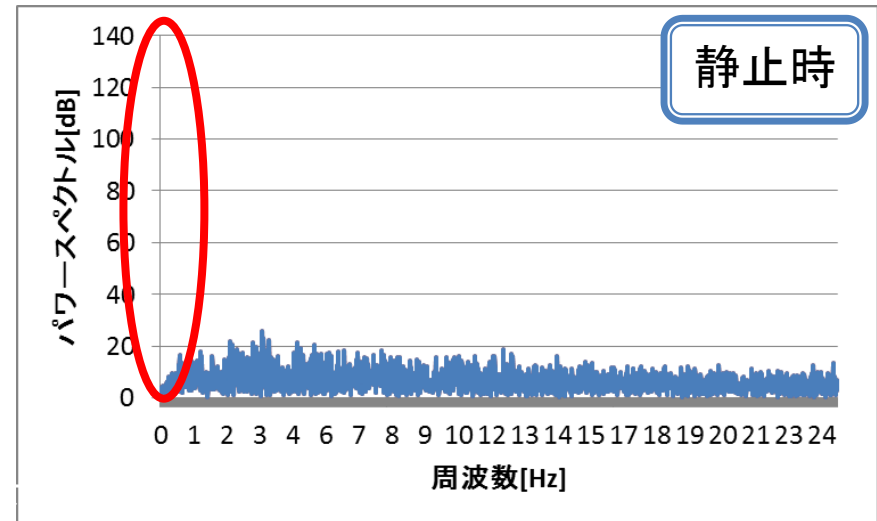
HPF式

$$g(i) = 0.9 * g(i - 1) + 0.1 * x(i)$$
$$acceleration(i) = x(i) - g(i)$$

$g(i)$: i 番目のLPF処理後の加速度値

$x(i)$: i 番目の実加速度値

$acceleration(i)$: i 番目のフィルタ処理後の加速度値



フィルタ処理

- ▶ HPFをかけることで低周波の振動を除去
⇒ 身体の揺れなどによる低周波の振動を除去

HPF※式

$$g(i) = 0.9 * g(i - 1) + 0.1 * x(i)$$
$$acceleration(i) = x(i) - g(i)$$

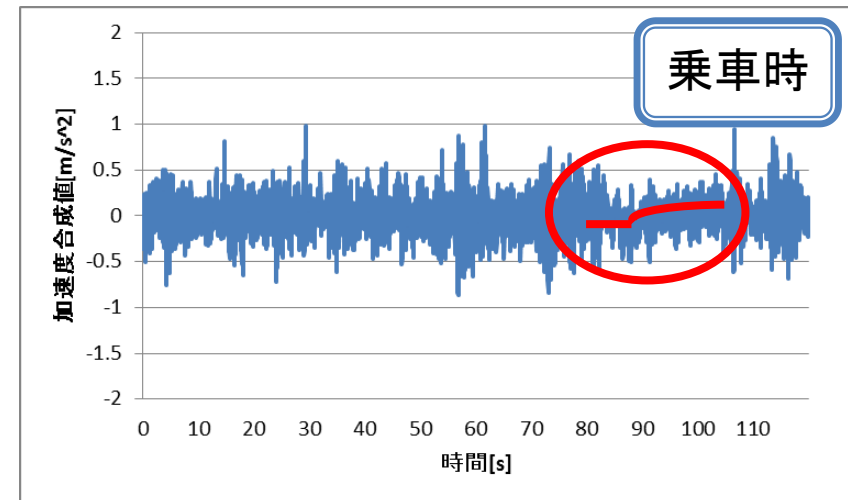
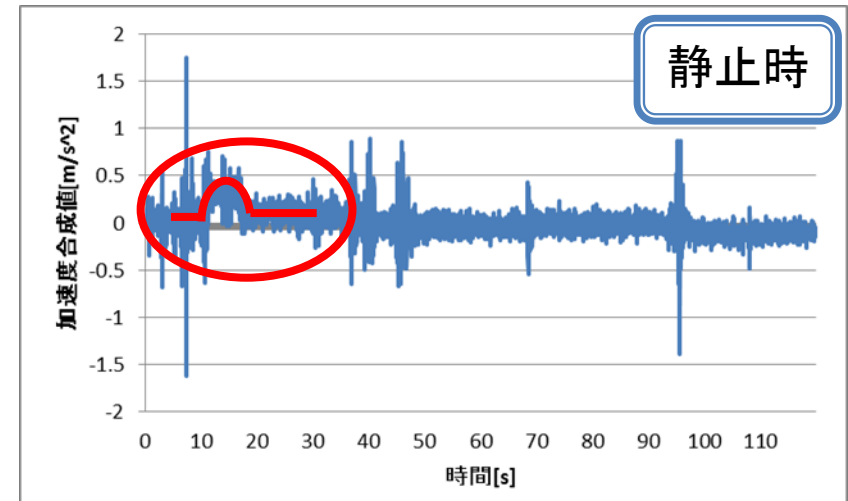
$g(i)$: i 番目のLPF*処理後の加速度値

$x(i)$: i 番目の実加速度値

$acceleration(i)$: i 番目のフィルタ処理後の加速度値

※High Pass Filter

* Low Pass Filter



フィルタ処理

- ▶ HPFをかけることで低周波の振動を除去
⇒ 身体の揺れなどによる
低周波の振動を除去

HPF※式

$$g(i) = 0.9 * g(i - 1) + 0.1 * x(i)$$
$$acceleration(i) = x(i) - g(i)$$

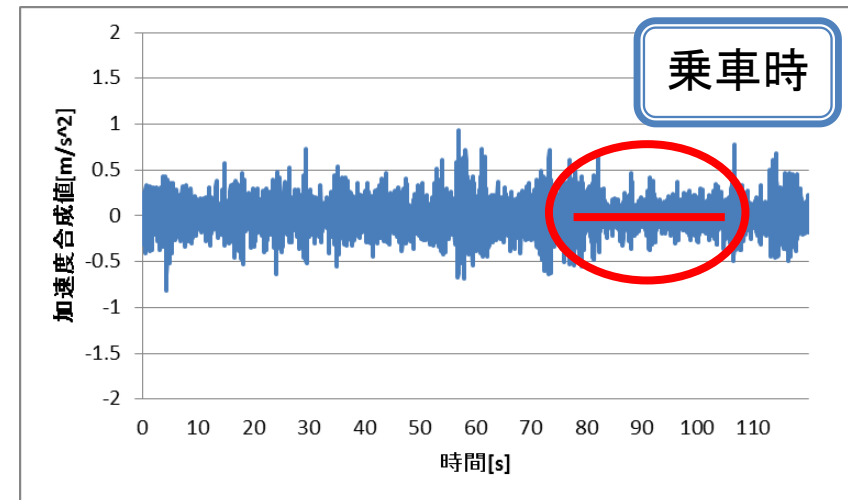
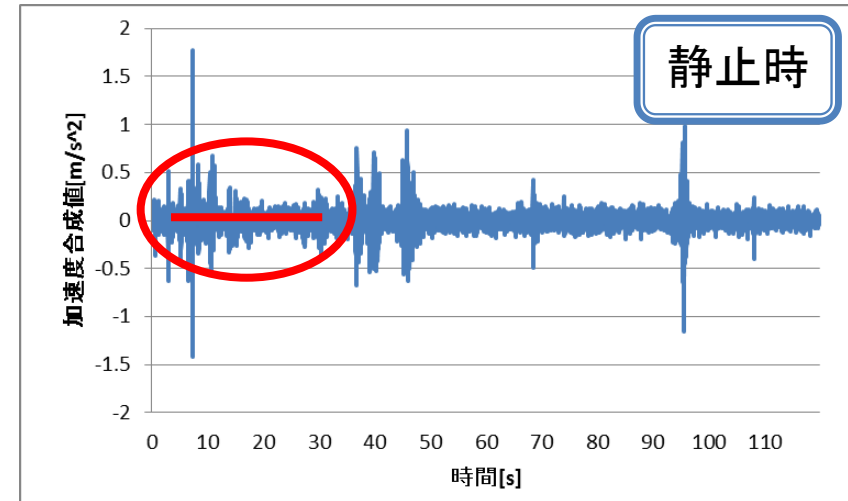
$g(i)$: i 番目のLPF*処理後の加速度値

$x(i)$: i 番目の実加速度値

$acceleration(i)$: i 番目のフィルタ処理後の加速度値

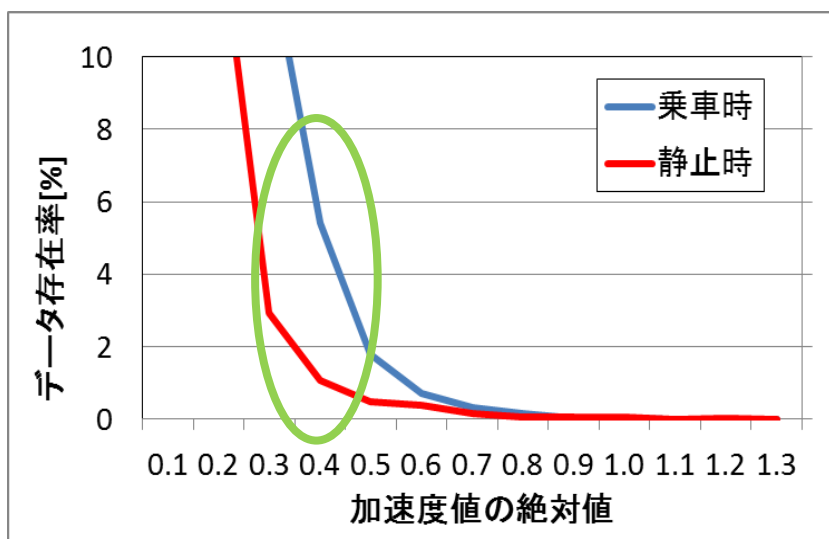
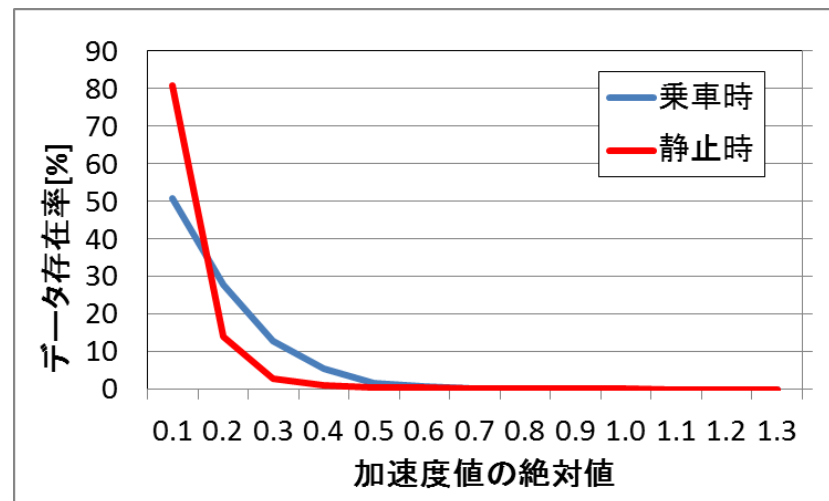
※High Pass Filter

* Low Pass Filter



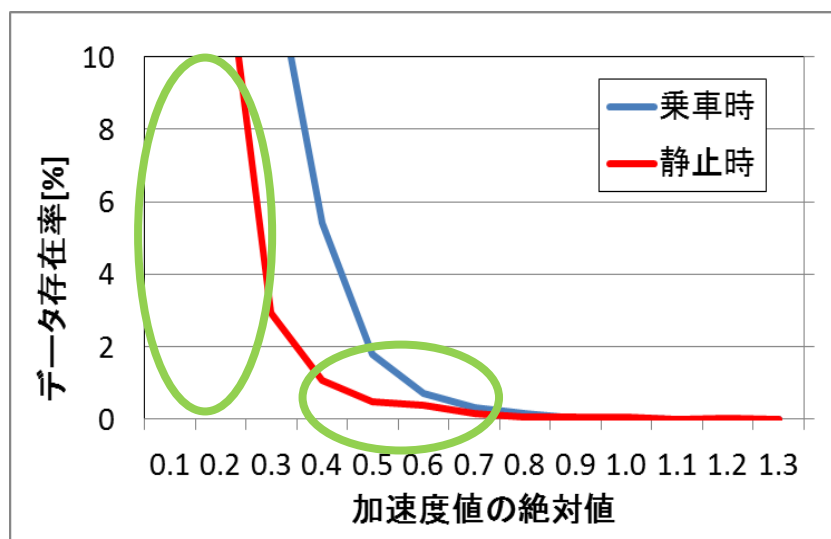
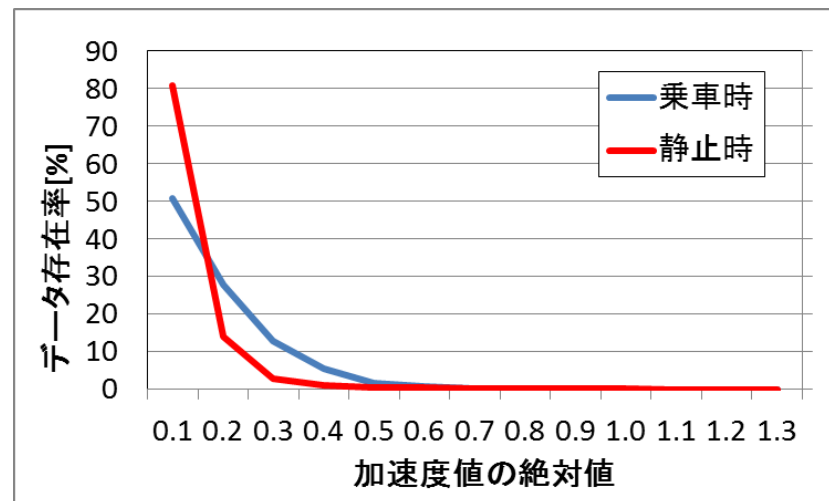
振幅制限時の閾値の決定方法

- ▶ 突発的な振動
 - 他の振動に比べ、大きい振動
- ▶ 乗車時
 - データ存在率の変化が一定に近い
⇒ 突出している振動がない
- ▶ 静止時
 - 加速度データのほとんどが0.2以下に存在している
 - 乗車時に比べ、0.3付近のデータが少ない
⇒ 突出した振動が存在している



振幅制限時の閾値の決定方法

- ▶ 突発的な振動
 - 他の振動に比べ、大きい振動
- ▶ 乗車時
 - データ存在率の変化が一定に近い
⇒ 突出している振動がない
- ▶ 静止時
 - 加速度データのほとんどが0.2以下に存在している
 - 乗車時に比べ、0.3付近のデータが少ない
⇒ 突出した振動が存在している

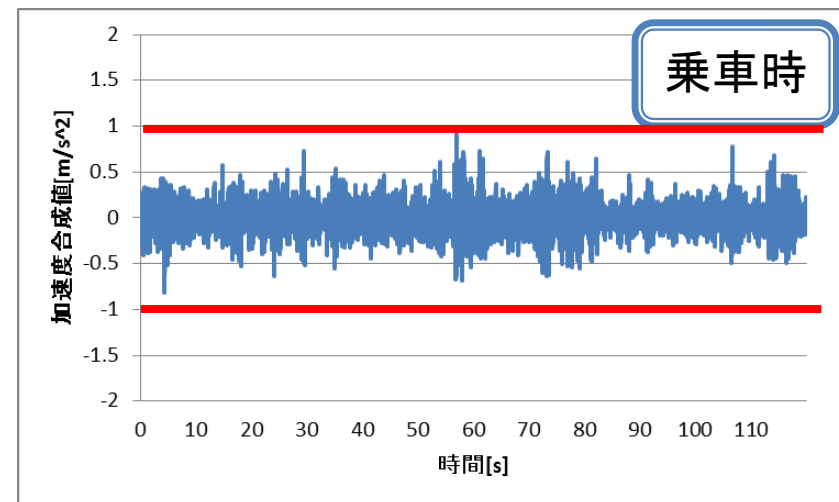
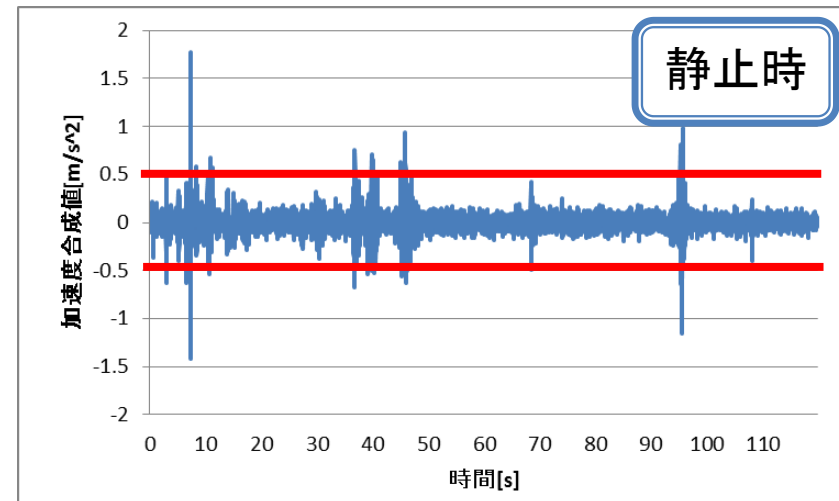


振幅制限時の閾値の決定方法

▶ 閾値の決定

- 低い値から見て加速度データの95%が存在している加速度値を基準に閾値を決定

静止時の閾値: 0.5, -0.5
乗車時の閾値: 1.0, -1.0

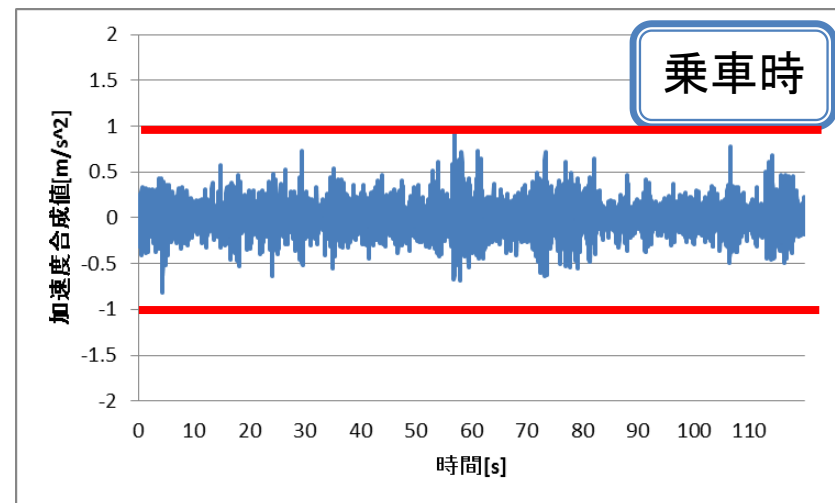
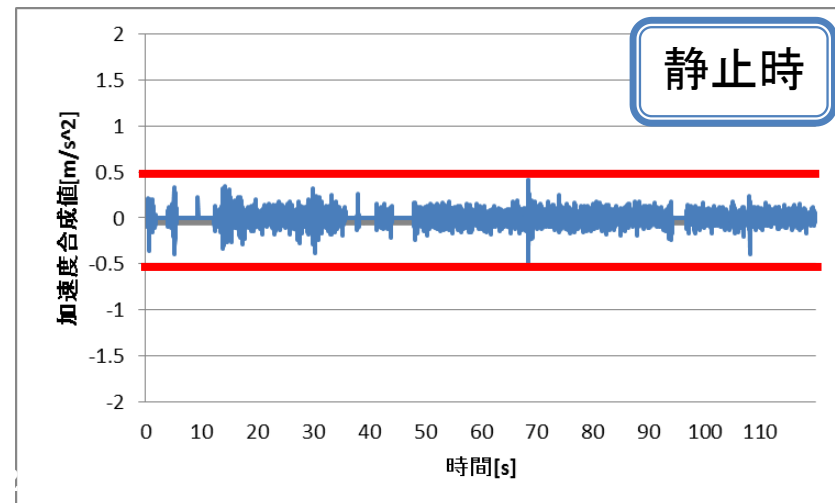


振幅制限時の閾値の決定方法

▶ 閾値の決定

- 低い値から見て加速度データの95%が存在している加速度値を基準に閾値を決定

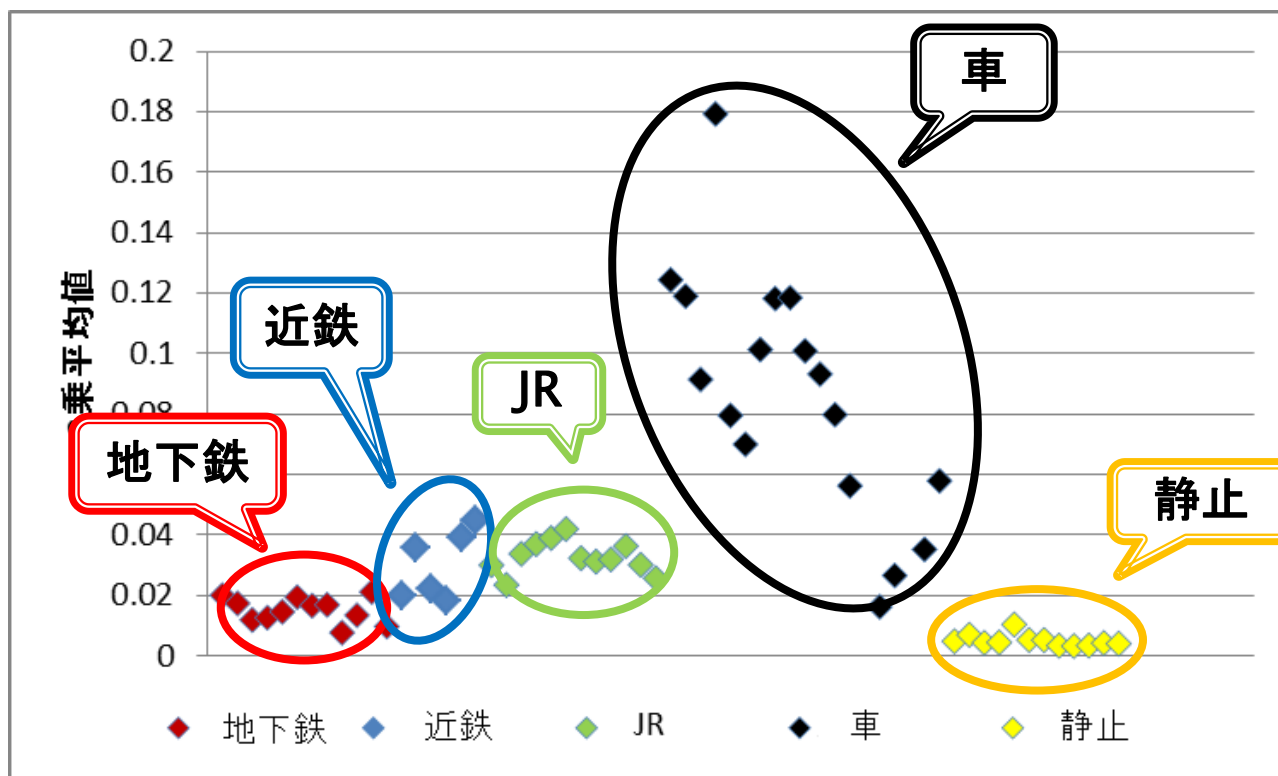
静止時の閾値: 0.5, -0.5
乗車時の閾値: 1.0, -1.0



2乗平均値の閾値の決定方法

▶ 測定条件

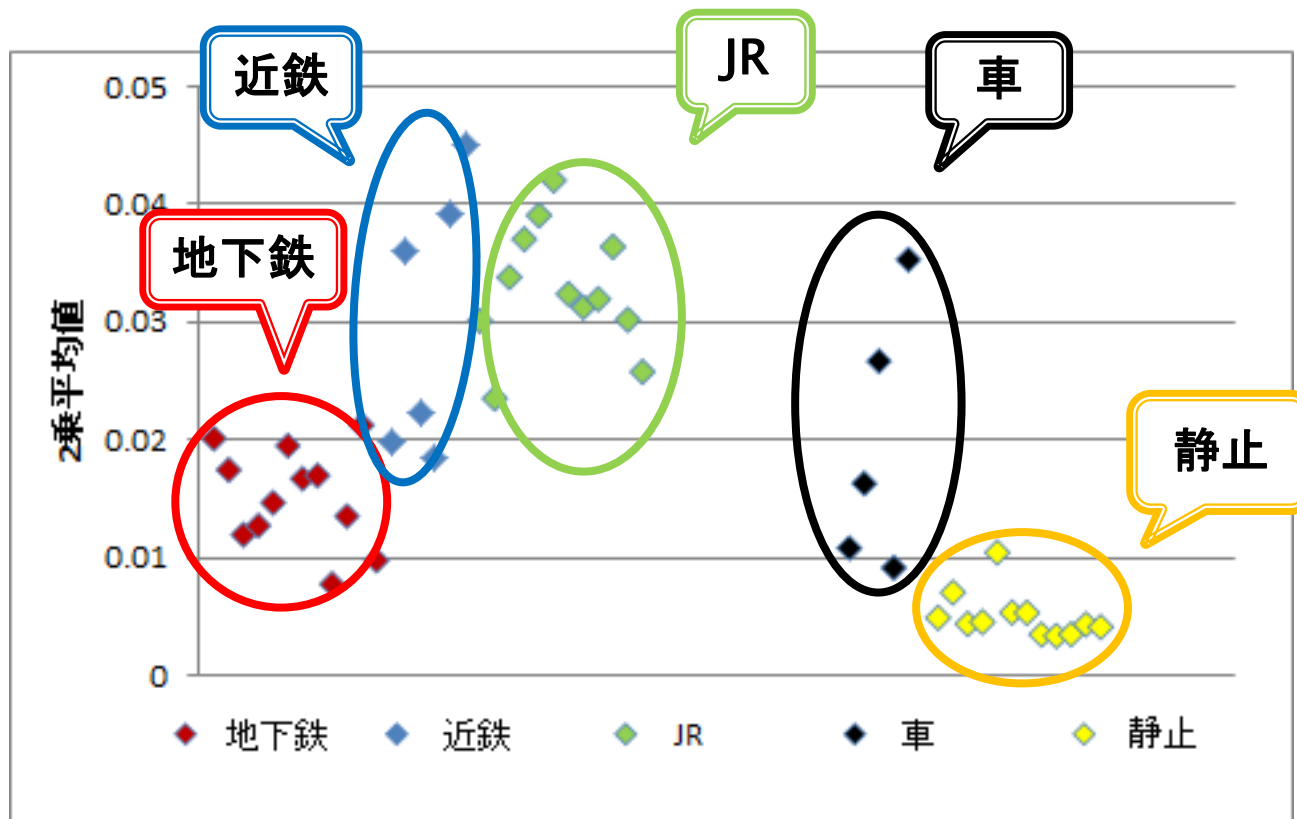
- 使用する端末: Galaxy Nexus(Android4.1)
- 所持する場所:ズボンの前ポケット



2乗平均値の閾値の決定方法

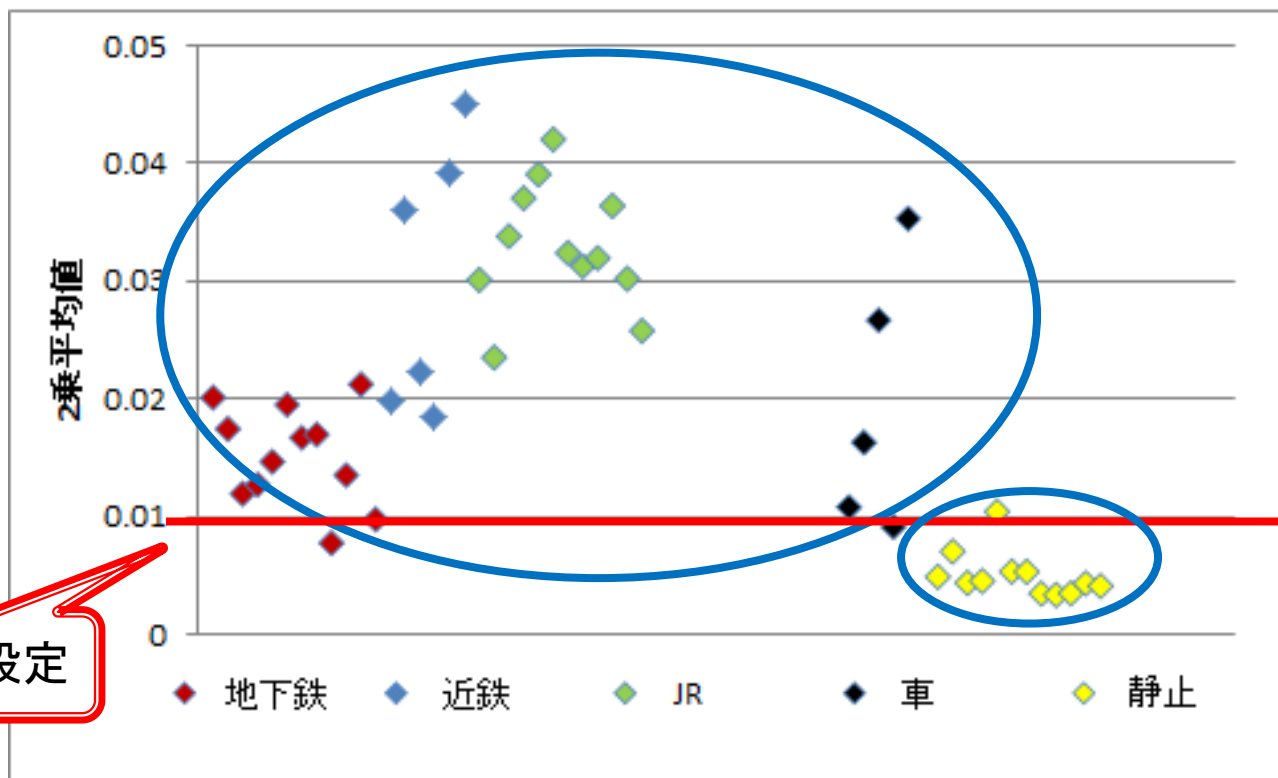
▶ 測定条件

- 使用する端末: Galaxy Nexus(Android4.1)
- 所持する場所:ズボンの前ポケット



2乗平均値の閾値の決定方法

- ▶ 乗車時に比べ、静止時の2乗平均値は値が小さい
⇒ 中間である0.01を閾値として設定



評価

▶ 評価方法

- 提案方式を実装したAndroid端末をユーザが所持し、乗車時と静止時において乗車判定を行う

▶ 使用する端末

- Galaxy Nexus(Android4.1)

▶ 端末を所持する場所

- ズボンの前ポケット

評価

▶ 測定環境

● 乗車時

- 地下鉄鶴舞線(名古屋～塩釜口間), 名城線(八事～上前津間)
- 近鉄線(名古屋～弥富間)
- JR線(舞阪～豊橋間)
- 車(舗装された道)

● 静止時

- 被験者3名に研究室内でいつも通りの生活をしてもらい, 加速度値を測定

評価

▶ 測定結果

- 全体として認識率が84.65%となり, 高い認識率で乗車判定を行うことができた

	サンプル数	正判定数	誤判定数	認識率[%]
地下鉄線	51	22	29	43.14
JR線	167	149	18	89.22
近鉄線	55	53	2	96.36
車	72	72	0	100
静止	124	101	23	81.45
合計(乗車)	345	296	49	85.80
合計(乗車+静止)	469	397	72	84.65

評価

▶ 測定結果

- 地下鉄乗車時の認識率が43%と低い値であった
 - ・ 駅と駅の間隔が短く、電車が停車している間の加速度値が含まれていたため誤判定した
- ⇒判定する間隔を短くすることで認識率上昇

	サンプル数	正判定数	誤判定数	認識率[%]
地下鉄線	51	22	29	43.14
JR線	167	149	18	89.22
近鉄線	55	53	2	96.36
車	72	72	0	100
静止	124	101	23	81.45
合計(乗車)	345	296	49	85.80
合計(乗車＋静止)	469	397	72	84.65