センサー工学

2013年12月18日（水）,
2014年1月8日（水）,1月15日（水）
第11回,12回,13回,14回

知能情報工学科　横田孝義

センサー工学

回	授業内容	予習・復習内容	備考
1	授業のガイダンス	計測工学で学んだ単位系、誤差、精度などの復習を行う	
2	ジャイロセンサの仕組み	ジャイロセンサの仕組み	
3	最近の動向MEMSセンサII	MEMSセンサシステムの考え方を理解する	レポート#1 MEMSセンサの動向
4	「長さ」のセンシング	リミットスイッチや光電センサなどが身近に多用されていることを知る変位センサに応用されている物理学を学ぶ	
5	「振動、加速度」のセンシングI	基本的な力学的センサーの原理を理解する	
6	「振動、加速度」のセンシングII	基本的な力学的センサーの原理、サイズモの原理を理解する	
7	中間テスト		レポート#2 進捗に合わせて課題を設定
8	「力・圧力」のセンシング I	力、圧力の単位を確認し、各種の力・圧力センサの動作原理を理解する	
9	「力・圧力」のセンシング II	力、圧力の単位を確認し、各種の力・圧力センサの動作原理を理解する	
10	「力・圧力」のセンシング II	力、圧力の単位を確認し、各種の力・圧力センサの動作原理を理解する	
11	釣り合いのセンシング	回転体の釣り合いの検出について学ぶ	
12	センサフュージョン	センサーフュージョンとは何かを学ぶ	
13	センサーフュージョンとカルマンフィルタI ：状態方程式について学ぶ	異種センサーの組み合わせ技術を理解する。	
14	センサーフュージョンとカルマンフィルタII ：観測方程式について学ぶ	異種センサーの組み合わせ技術を理解する。	
15	センサーフュージョンとカルマンフィルタIII ：カルマンゲインについて学ぶ	異種センサーの組み合わせ技術を理解する。	レポート#3 センサフュージョンについて

センサーフュージョンとは何か？

Sensor Fusion

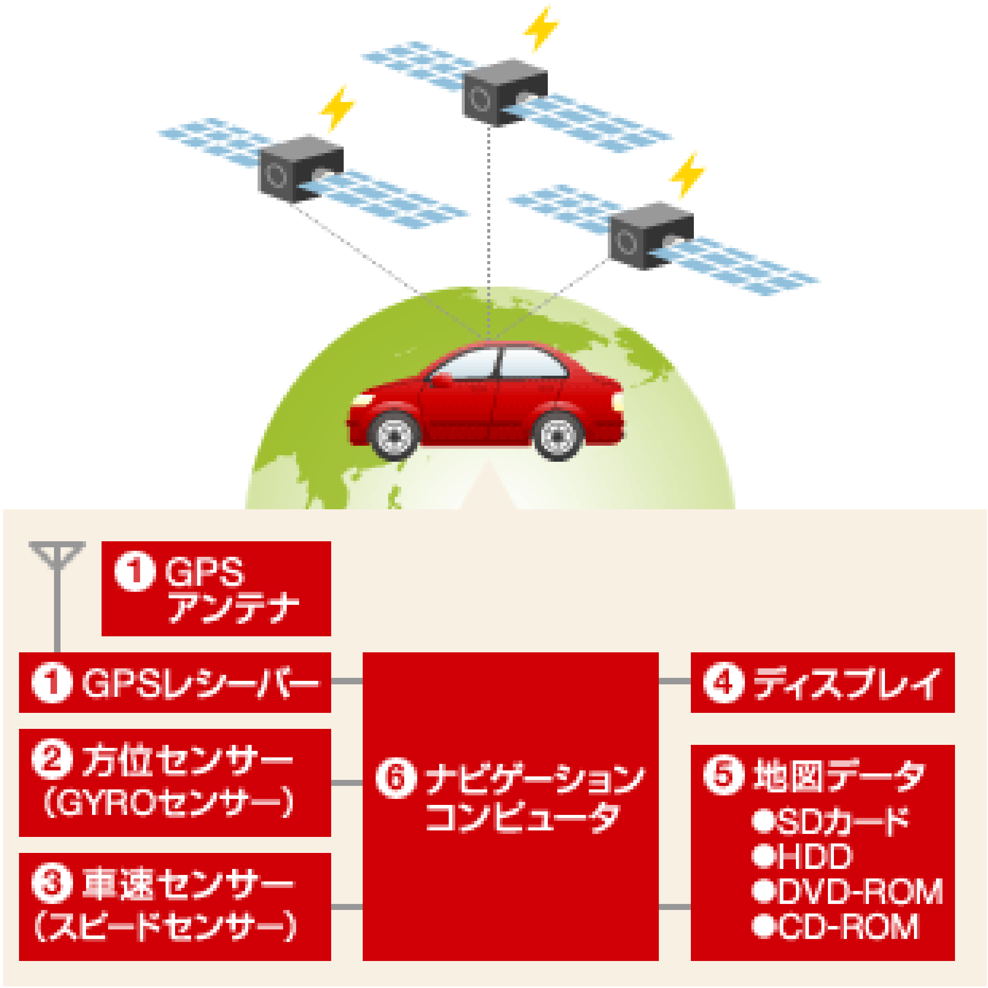
↓

融合

異なるセンサ(情報）あるいは複数のセンサ(情報）を
融合して単独のセンサでは得られない情報を引き出したり
精度や品質を高める技術。

センサーフュージョンとは何か？

例　カーナビゲーションシステム



位置を正確に推定するために

(1)GPS
(2)Gyroセンサ
(3)車速センサ
(4)加速度センサ
(5)ディジタル道路地図
(6)路車間通信

を駆使している。

センサーフュージョンとは何か？

1.GPSアンテナ・GPSレシーバー

GPS（Global Positioning System）とは、アメリカで軍事目的に打ち上げられた衛星を民間用に利用し、位置情報を知る方法です。現在、ナビのほとんどは、GPSの位置情報を基本としています。GPSの位置情報を知るには、少なくとも3個以上の衛星電波を受信しなければなりません。ナビでは、この衛星からの電波を受信する役目をしているのがGPSアンテナとGPSレシーバーになります。

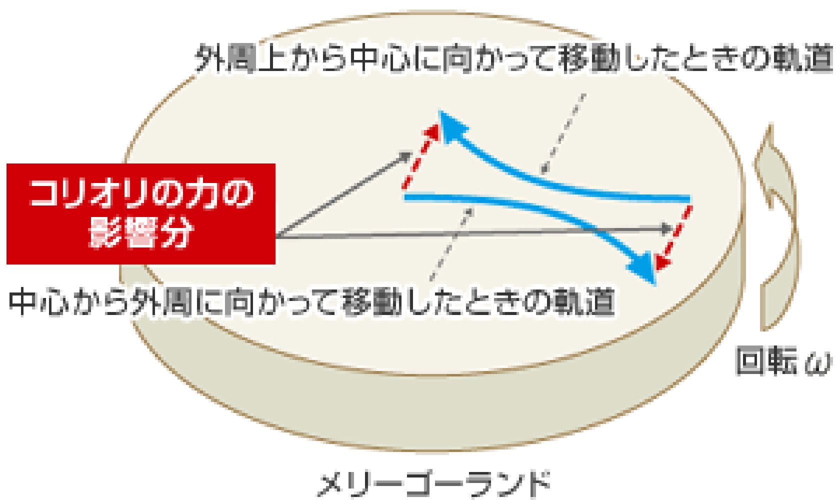
センサーフュージョンとは何か？

2.方位センサー（GYROセンサー）

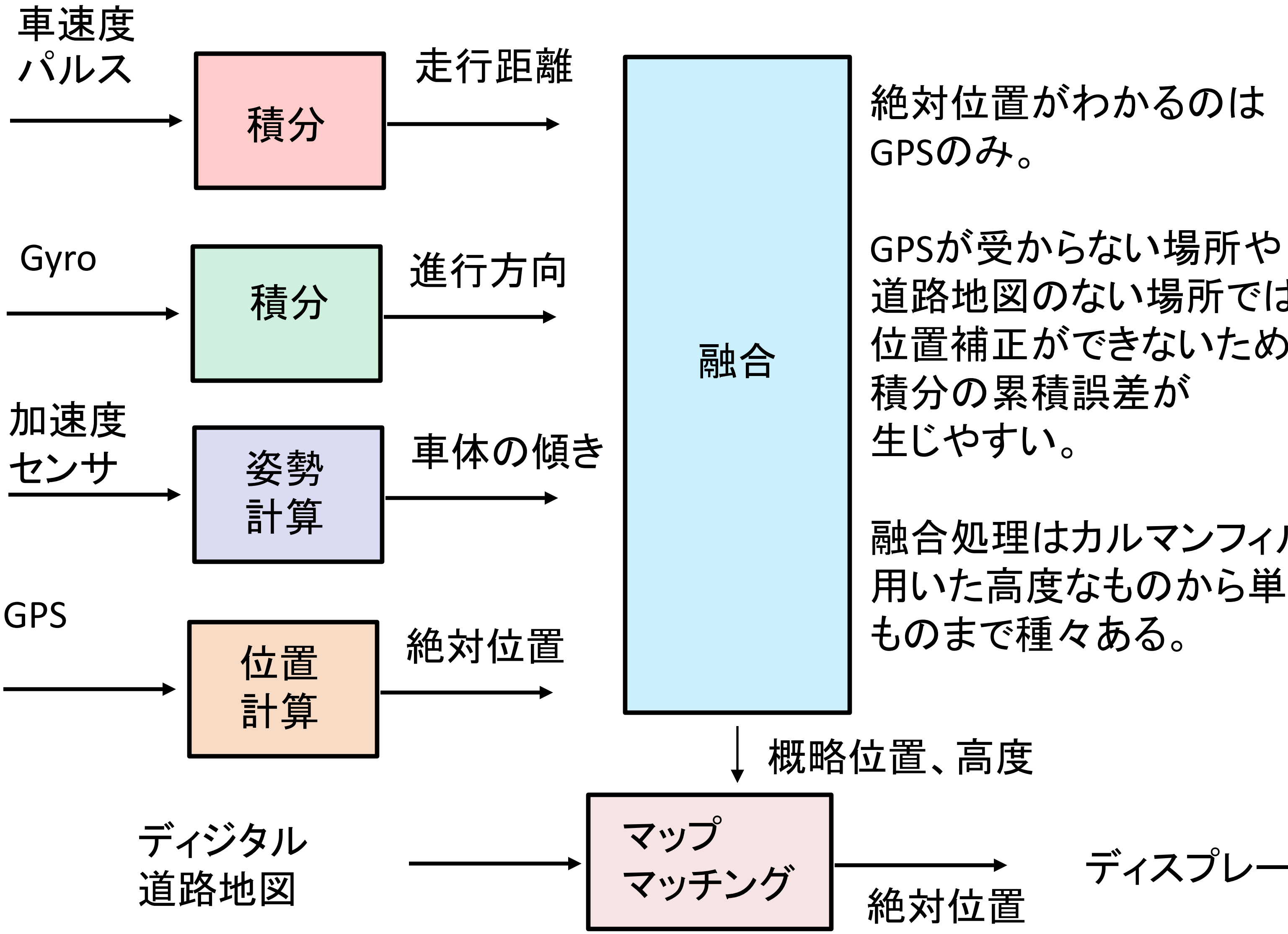
GPSアンテナでは、地図上の位置を把握することはできても方向を把握することはできません。ナビは、自動車の向きや回転方向を把握するために方位センサー（ジャイロセンサー）を使用しています。方位センサー（ジャイロセンサー）は、センサー内の振動素子を振動させ、自動車の向きが変化したときの振動の変化（コリオリの力）を検出して自動車の回転方向や向きを計算しています。ナビは、この方位センサー（ジャイロセンサー）の信号をもとに自動車の向きを画面に表示しています。

コリオリの力

たとえば、メリーゴーランドの中心（外周）から外周（中心）に目をつぶってまっすぐ歩いたときに曲がってしまう分の力をコリオリの力といいます。



センサーフュージョンとは何か？



絶対位置がわかるのはGPSのみ。

GPSが受からない場所や道路地図のない場所では位置補正ができないため積分の累積誤差が生じやすい。

融合処理はカルマンフィルタを用いた高度なものから単純なものまで種々ある。

センサーフュージョンとは何か？

3.車速センサー（スピードセンサー）

GPSで自車位置の確認はできますが、3個以上の衛星信号が受信できる場合に限るため、長いトンネルや高層ビルの間を走行中は位置の更新ができなくなります。このような場合にも自車位置を更新するための移動距離を把握するために、車速センサーの信号を使っています。

車輪1回転で

日産は2パルスや8パルス
トヨタはほとんど4パルス　など

センサーフュージョンとは何か？

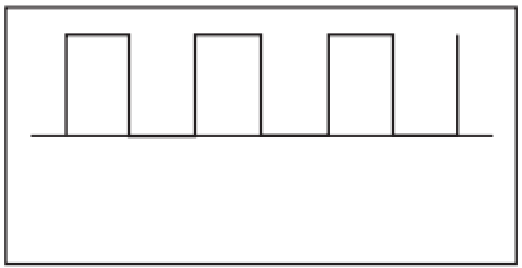
車速パルスについて

パルスを発生する方法として、リードスイッチによるものとホール素子によるものに大別する事ができます。通常リードスイッチ方式は、リードスイッチがメーターの裏にあり、ホール素子方式はミッションに取付けられております。
どちらもエンジンコントロールユニットやATコントロールユニット等の電子制御ユニット用に車速信号パルスとして使用されています。

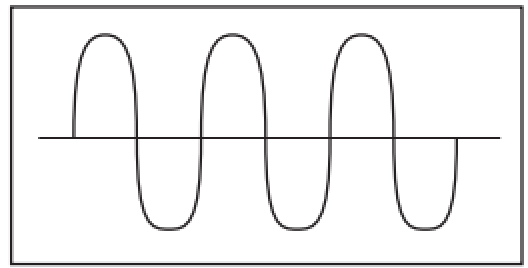
車速パルス信号のタイプとは

車速パルス信号タイプには、次の2タイプがあります。

A：デジタル波形（矩形波）



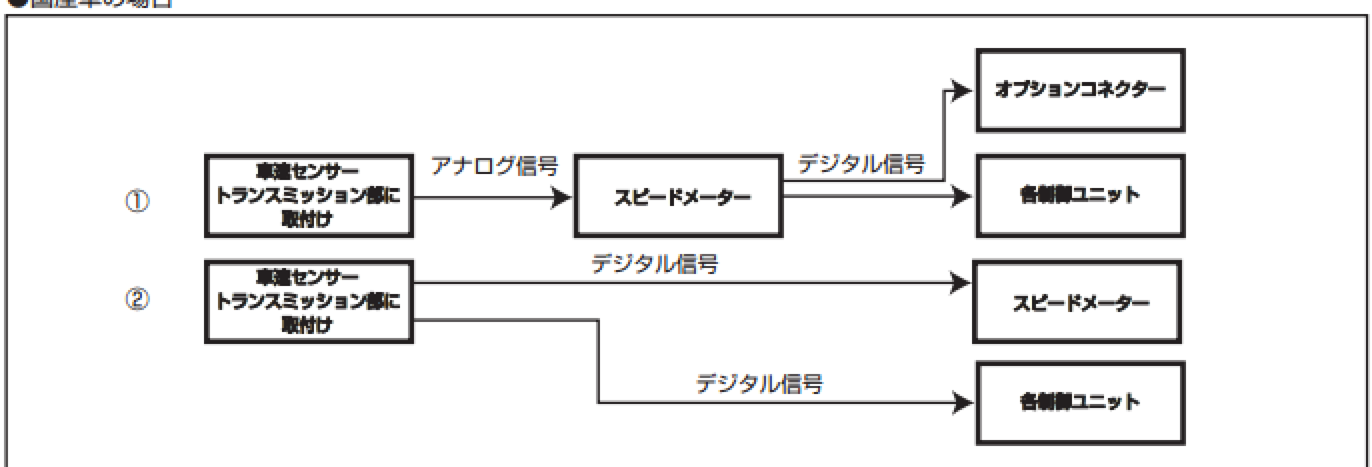
B：アナログ波形（交流波形）



Clarionナビゲーションシステムでは、A：デジタル波形のみ接続可能となります。

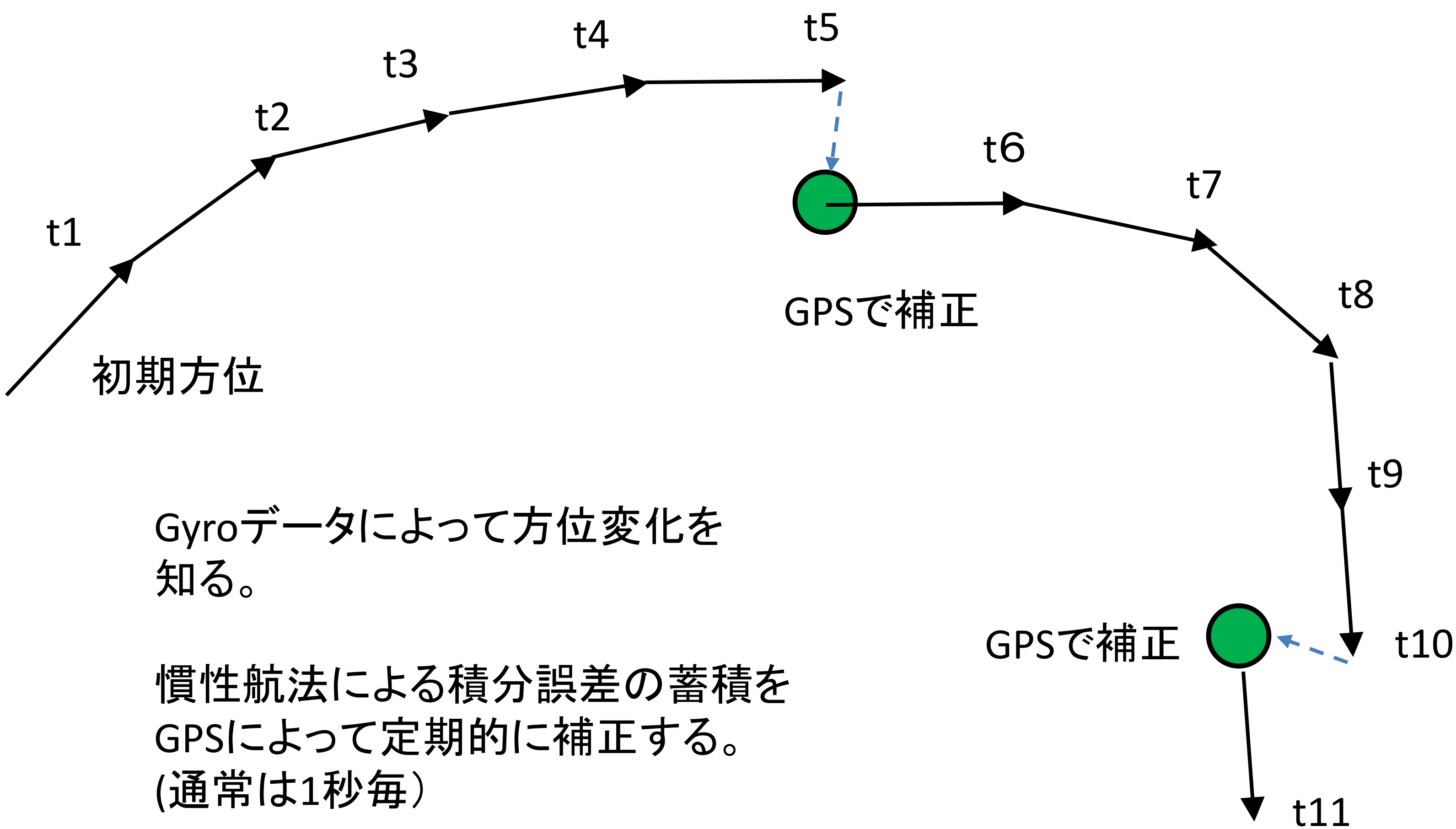
車速信号の経路

●国産車の場合



センサーフュージョンとは何か？

慣性航法(INS, inertial navigation system)＋GPS　のフュージョン

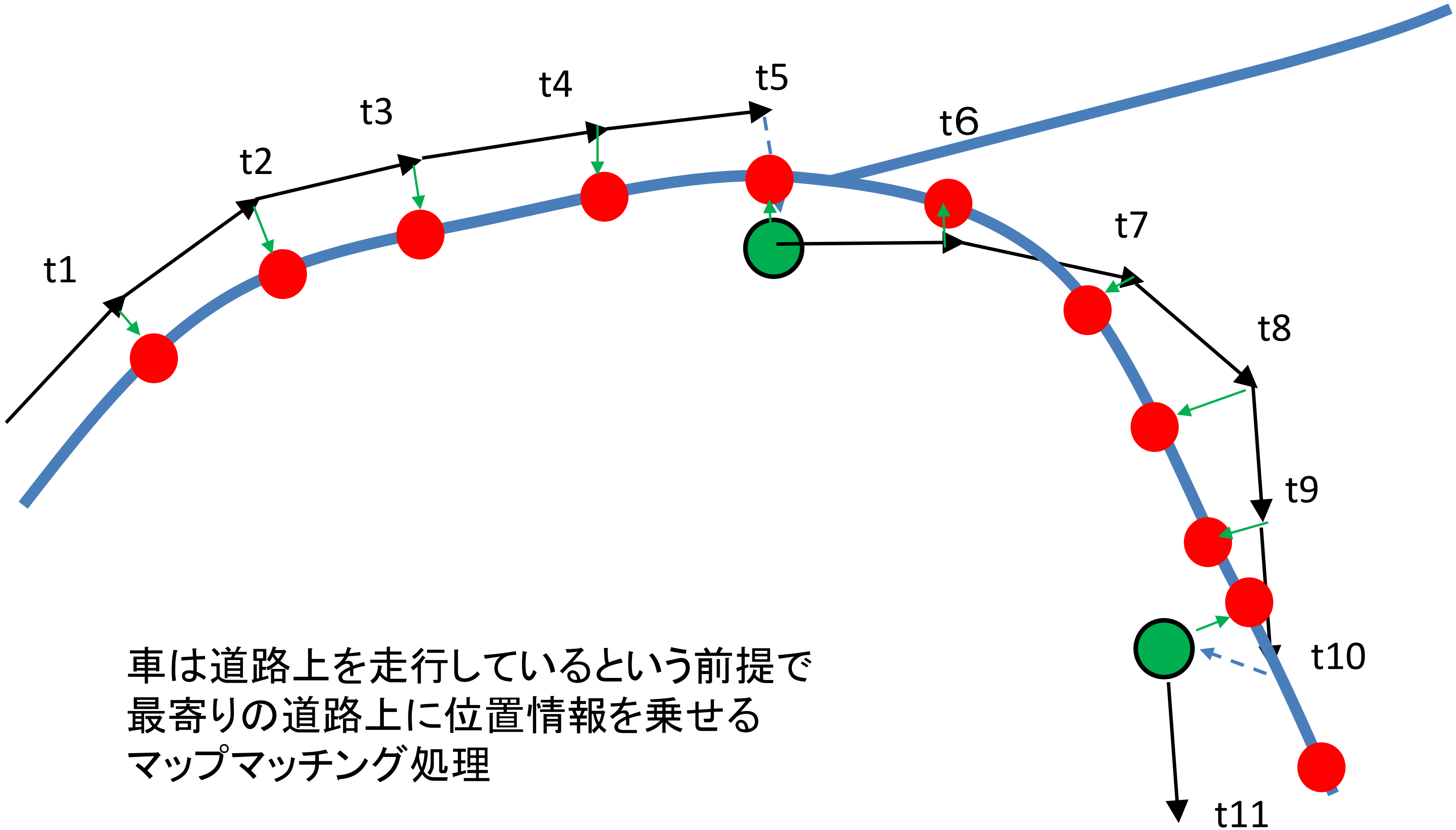


Gyroデータによって方位変化を知る。

慣性航法による積分誤差の蓄積をGPSによって定期的に補正する。
(通常は1秒毎）

センサーフュージョンとは何か？

慣性航法(INS, inertial navigation system)＋GPS＋ディジタル道路地図のフュージョン



車は道路上を走行しているという前提で
最寄りの道路上に位置情報を乗せる
マップマッチング処理

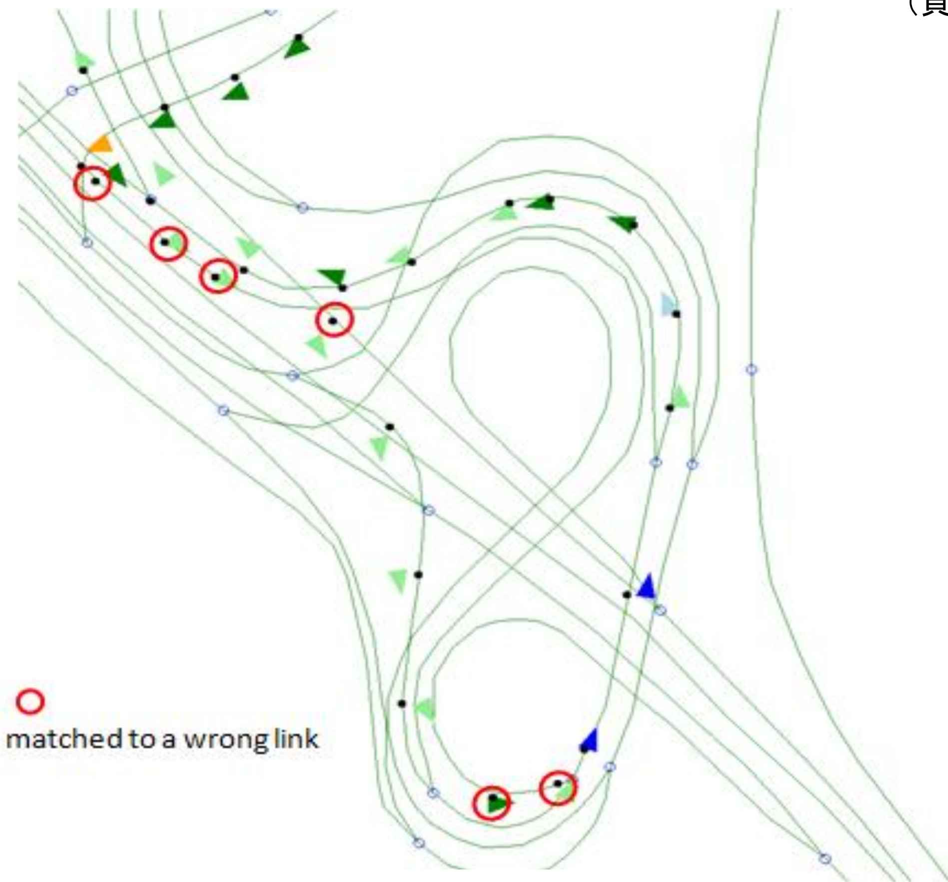
センサーフュージョンとは何か？

京都　大山崎ジャンクション付近

並走道路、立体交差が多く、
マップマッチングが難しい場所。

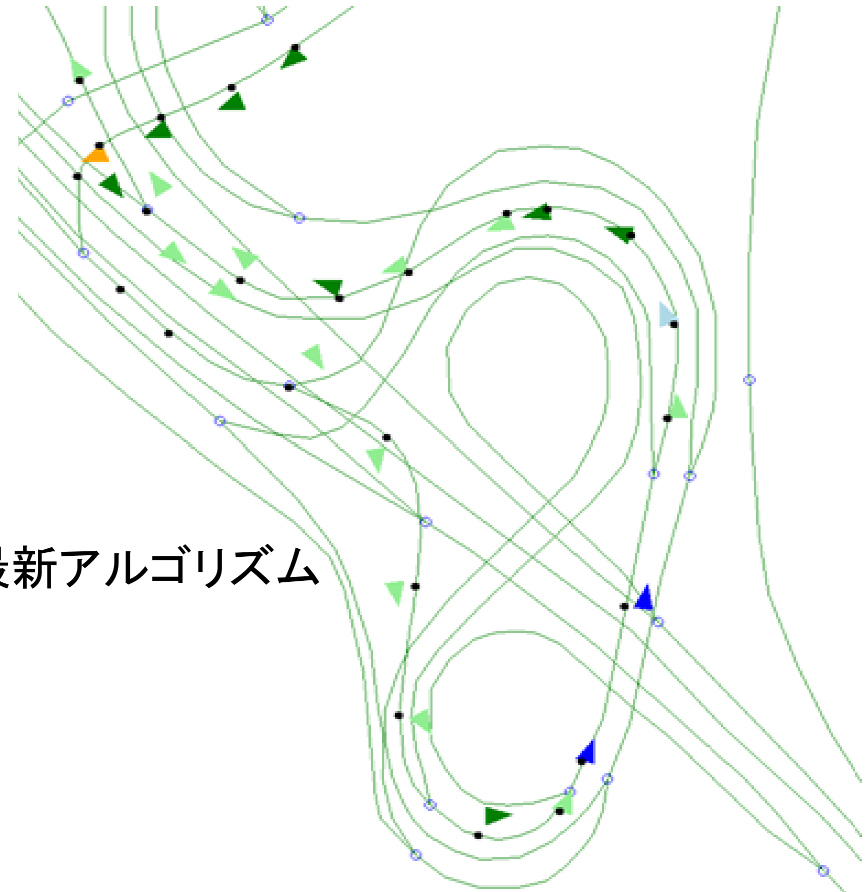
センサーフュージョンとは何か？

Route Identification of Freight Vehicles Tour Using GPS Probe
Takayoshi Yokota 、Procedia - Social and Behavioral Sciences 、39巻
（頁 255 ～ 266）、2012年07月



誤った道路に
マッチングしたプローブ情報

正答率　19/25=76%

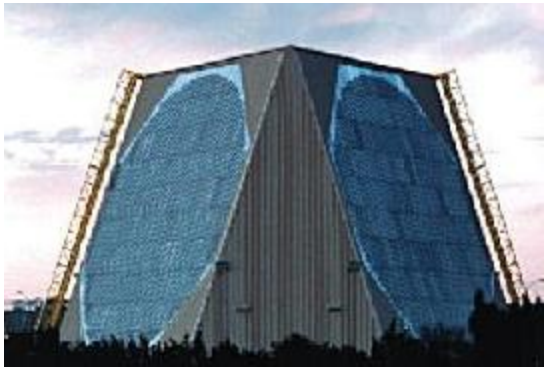


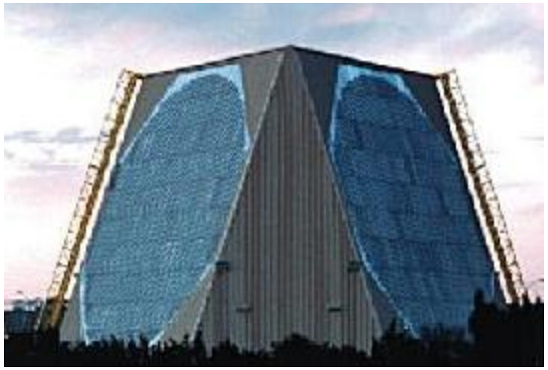
最新アルゴリズム

正答率　100%

センサーフュージョンとは何か？

センサーの種類は同一であるが、複数のセンサ情報を融合する場合も
センサーフュージョンと呼んで良いであろう。 ミサイルの追尾レーダーなどが良い例。

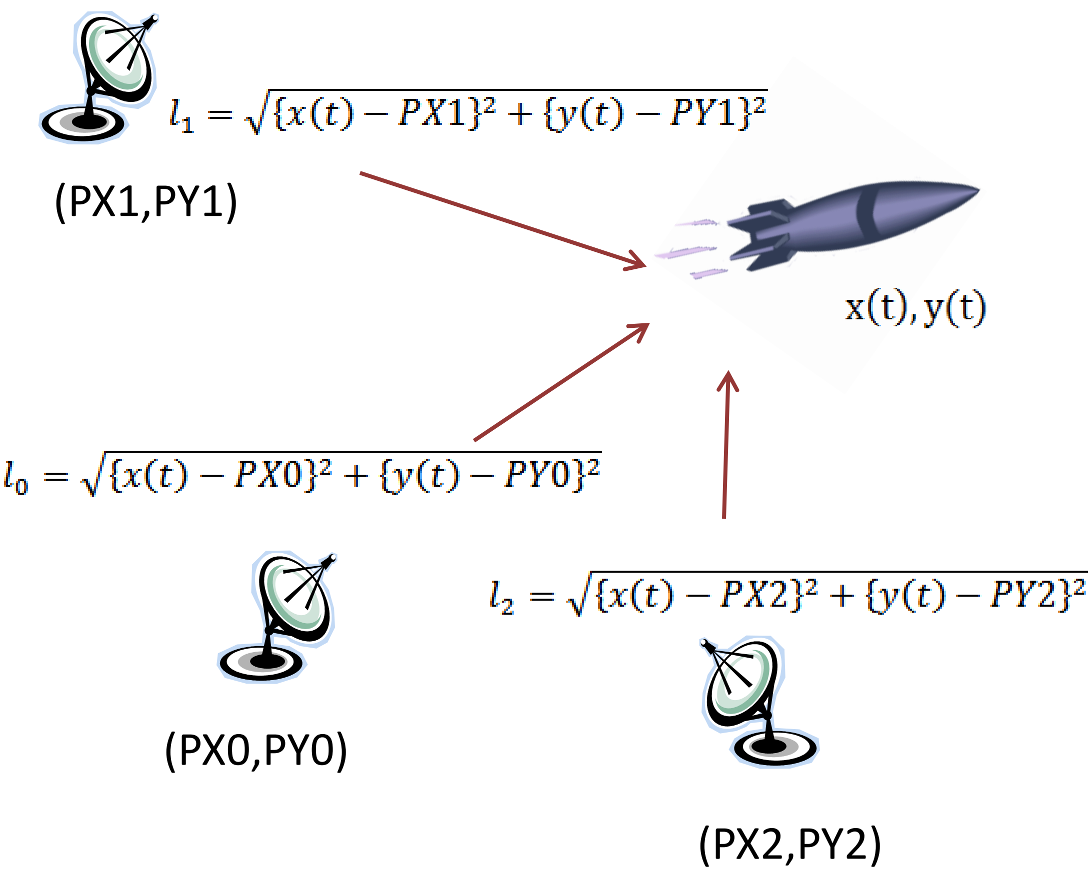




Upgraded Early Warning Radar (UEWR).

センサーフュージョンとは何か？

ミサイルの追跡（簡単のためにx,yの２次元で扱う）



複数の観測情報を融合

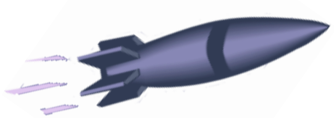
$$l_0 = \sqrt{\{x(t) - PX0\}^2 + \{y(t) - PY0\}^2}$$

$$l_1 = \sqrt{\{x(t) - PX1\}^2 + \{y(t) - PY1\}^2}$$

$$l_2 = \sqrt{\{x(t) - PX2\}^2 + \{y(t) - PY2\}^2}$$

センサーフュージョンとは何か？

ミサイルの追跡（簡単のためにx,yの２次元で扱う）



$x(t), y(t)$

対象を状態で捉える…………状態　$x(t), y(t)$

対象の時間変化をモデル化

状態方程式あるいは状態遷移モデルで表現する。
例えば角度45度の等速直線運動としてミサイルの軌跡をモデル化すると

$$\begin{bmatrix} x \\ y \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}_{k} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} v\Delta t + \begin{bmatrix} n_0 \\ n_1 \end{bmatrix}_{k+1}$$

Δt: サンプリング時間

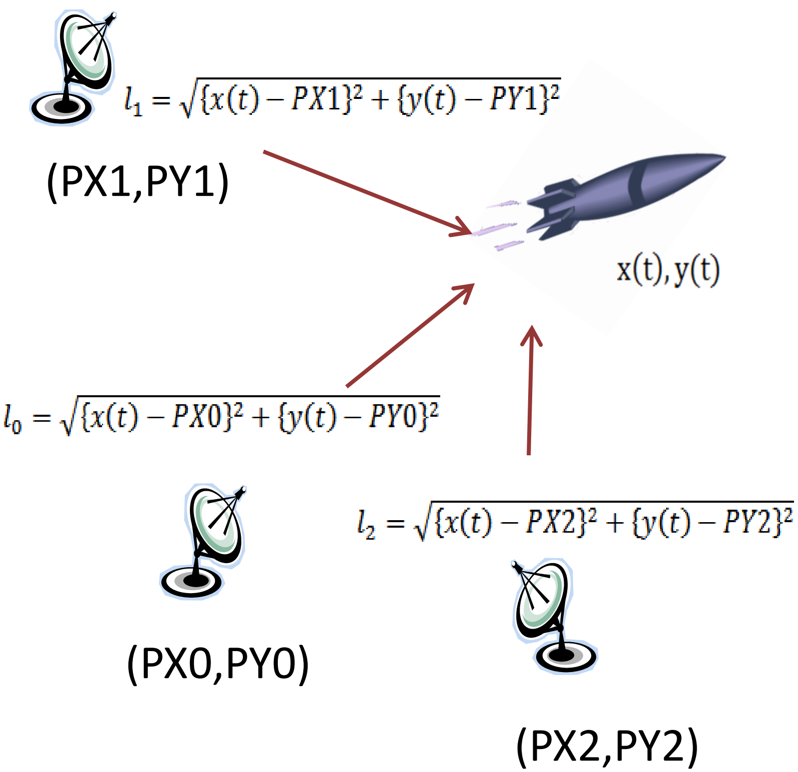
v: 速度

外乱（風の影響など）

センサーフュージョンとは何か？

ミサイルの追跡（簡単のためにx,yの２次元で扱う）

観測情報と状態変数の関係をモデル化



$$\begin{bmatrix} l_0 \\ l_1 \\ l_2 \end{bmatrix}_k = \begin{bmatrix} \sqrt{\{x(t)-PX0\}^2+\{y(t)-PY0\}^2} \\ \sqrt{\{x(t)-PX1\}^2+\{y(t)-PY1\}^2} \\ \sqrt{\{x(t)-PX2\}^2+\{y(t)-PY2\}^2} \end{bmatrix} + \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}_k$$

観測誤差

観測情報(l_0,l_1,l_2)と状態x_k,y_kは非線形な関係

これを線形近似する。

センサーフュージョンとは何か？

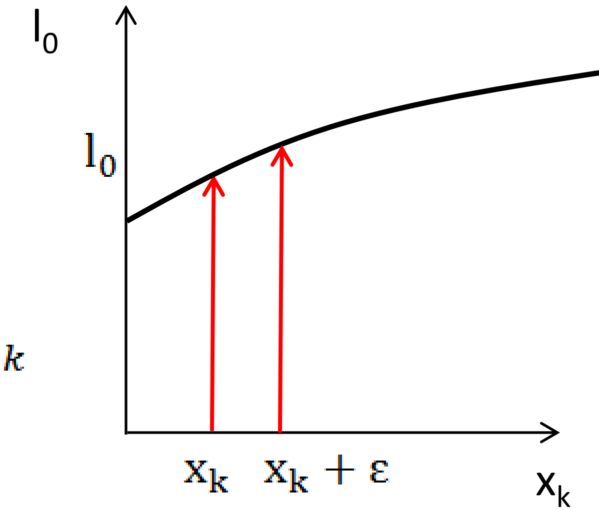
いわゆるJacob行列　すなわち　変数変換を行う行列

$$\begin{bmatrix} \frac{\partial l_0}{\partial x_k} & \frac{\partial l_0}{\partial y_k} \\ \frac{\partial l_1}{\partial x_k} & \frac{\partial l_1}{\partial y_k} \\ \frac{\partial l_2}{\partial x_k} & \frac{\partial l_2}{\partial y_k} \end{bmatrix} = \begin{bmatrix} \frac{x(t) - PX0}{\sqrt{\{x(t) - PX0\}^2 + \{y(t) - PY0\}^2}} & \frac{y(t) - PY0}{\sqrt{\{x(t) - PX0\}^2 + \{y(t) - PY0\}^2}} \\ \frac{x(t) - PX1}{\sqrt{\{x(t) - PX1\}^2 + \{y(t) - PY1\}^2}} & \frac{y(t) - PY1}{\sqrt{\{x(t) - PX1\}^2 + \{y(t) - PY1\}^2}} \\ \frac{x(t) - PX2}{\sqrt{\{x(t) - PX2\}^2 + \{y(t) - PY2\}^2}} & \frac{y(t) - PY2}{\sqrt{\{x(t) - PX2\}^2 + \{y(t) - PY2\}^2}} \end{bmatrix}$$

センサーフュージョンとは何か？

観測方程式の線形化

$$\begin{bmatrix} l_0 \\ l_1 \\ l_2 \end{bmatrix}_k = \begin{bmatrix} \sqrt{\{x(t) - PX0\}^2 + \{y(t) - PY0\}^2} \\ \sqrt{\{x(t) - PX1\}^2 + \{y(t) - PY1\}^2} \\ \sqrt{\{x(t) - PX2\}^2 + \{y(t) - PY2\}^2} \end{bmatrix} + \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}_k$$



かなり線形性は高いが、下記の近似式(Taylor展開）
を用いて線形化する。

$$\mathrm{f}(\mathbf{x}) \cong \mathrm{f}(\mathbf{x}_\mathrm{k}) + \frac{\partial \mathrm{f}(\boldsymbol{\zeta})}{\partial \boldsymbol{\xi}} \Big|_{\xi=\mathbf{x}_\mathrm{k}} (\mathbf{x} - \mathbf{x}_\mathrm{k})$$

センサーフュージョンとは何か？

観測方程式の線形化

事前推定値

観測値の推定

$$\begin{bmatrix}\hat{l}_0\\ \hat{l}_1\\ \hat{l}_2\end{bmatrix}_k = \begin{bmatrix}h_{00} & h_{01}\\ h_{10} & h_{11}\\ h_{11} & h_{21}\end{bmatrix}_k \begin{bmatrix}\hat{x}^-\\ \hat{y}^-\end{bmatrix}_k = H_k \begin{bmatrix}\hat{x}^-\\ \hat{y}^-\end{bmatrix}_k$$

線形近似で得た観測行列

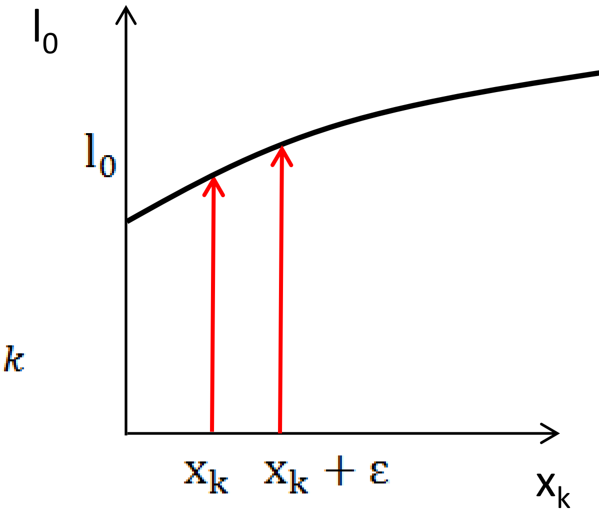
$$\boldsymbol{H}_k = \begin{bmatrix}\frac{\partial l_0}{\partial x} & \frac{\partial l_0}{\partial y}\\ \frac{\partial l_1}{\partial x} & \frac{\partial l_1}{\partial y}\\ \frac{\partial l_2}{\partial x} & \frac{\partial l_2}{\partial y}\end{bmatrix}_{|x=\hat{x}_k^-, y=\hat{y}_k^-} = \begin{bmatrix}\frac{x(t)-PX0}{\sqrt{\{x(t)-PX0\}^2+\{y(t)-PY0\}^2}} & \frac{y(t)-PY0}{\sqrt{\{x(t)-PX0\}^2+\{y(t)-PY0\}^2}}\\ \frac{x(t)-PX1}{\sqrt{\{x(t)-PX1\}^2+\{y(t)-PY1\}^2}} & \frac{y(t)-PY1}{\sqrt{\{x(t)-PX1\}^2+\{y(t)-PY1\}^2}}\\ \frac{x(t)-PX2}{\sqrt{\{x(t)-PX2\}^2+\{y(t)-PY2\}^2}} & \frac{y(t)-PY2}{\sqrt{\{x(t)-PX2\}^2+\{y(t)-PY2\}^2}}\end{bmatrix}_{|x=\hat{x}_k^-, y=\hat{y}_k^-}$$

x, y に事前推定値 $\hat{x}_k^-, \hat{y}_k^-$ を代入。

センサーフュージョンとは何か？

観測方程式の線形化

$$\begin{bmatrix} l_0 \\ l_1 \\ l_2 \end{bmatrix}_k = \begin{bmatrix} \sqrt{\{x(t) - PX0\}^2 + \{y(t) - PY0\}^2} \\ \sqrt{\{x(t) - PX1\}^2 + \{y(t) - PY1\}^2} \\ \sqrt{\{x(t) - PX2\}^2 + \{y(t) - PY2\}^2} \end{bmatrix} + \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}_k$$



かなり線形性は高いが、下記の近似式(Taylor展開）
を用いて線形化する。

$$f(\mathbf{x}) \cong f(\mathbf{x}_k) + \frac{\partial f(\boldsymbol{\zeta})}{\partial \boldsymbol{\xi}}\Big|_{\xi=\mathbf{x}_k}(\mathbf{x} - \mathbf{x}_k)$$

センサーフュージョンとは何か？

観測方程式の線形化

事前推定値

観測値の推定
$$\begin{bmatrix}\hat{l}_0\\ \hat{l}_1\\ \hat{l}_2\end{bmatrix}_k = \begin{bmatrix}h_{00} & h_{01}\\ h_{10} & h_{11}\\ h_{11} & h_{21}\end{bmatrix}_k \begin{bmatrix}\hat{x}^-\\ \hat{y}^-\end{bmatrix}_k = H_k \begin{bmatrix}\hat{x}^-\\ \hat{y}^-\end{bmatrix}_k$$

線形近似で得た観測行列

$$\boldsymbol{H}_k = \begin{bmatrix}\frac{\partial l_0}{\partial x} & \frac{\partial l_0}{\partial y}\\ \frac{\partial l_1}{\partial x} & \frac{\partial l_1}{\partial y}\\ \frac{\partial l_2}{\partial x} & \frac{\partial l_2}{\partial y}\end{bmatrix}_{|x=\hat{x}_k^-, y=\hat{y}_k^-} = \begin{bmatrix}\frac{x(t)-PX0}{\sqrt{\{x(t)-PX0\}^2+\{y(t)-PY0\}^2}} & \frac{y(t)-PY0}{\sqrt{\{x(t)-PX0\}^2+\{y(t)-PY0\}^2}}\\ \frac{x(t)-PX1}{\sqrt{\{x(t)-PX1\}^2+\{y(t)-PY1\}^2}} & \frac{y(t)-PY1}{\sqrt{\{x(t)-PX1\}^2+\{y(t)-PY1\}^2}}\\ \frac{x(t)-PX2}{\sqrt{\{x(t)-PX2\}^2+\{y(t)-PY2\}^2}} & \frac{y(t)-PY2}{\sqrt{\{x(t)-PX2\}^2+\{y(t)-PY2\}^2}}\end{bmatrix} |x=\hat{x}_k^-, y=\hat{y}_k^-$$

x, y に事前推定値 $\hat{x}_k^-, \hat{y}_k^-$ を代入。

センサーフュージョンとは何か？

観測値と観測値の推定値との誤差***err***

$$\begin{bmatrix} \mathrm{err}_0 \\ \mathrm{err}_1 \\ \mathrm{err}_2 \end{bmatrix}_k = \begin{bmatrix} l_0 \\ l_1 \\ l_2 \end{bmatrix}_k - \begin{bmatrix} \hat{l}_0 \\ \hat{l}_1 \\ \hat{l}_2 \end{bmatrix}_k = \begin{bmatrix} l_0 \\ l_1 \\ l_2 \end{bmatrix}_k - H_k \begin{bmatrix} \hat{x}^- \\ \hat{y}^- \end{bmatrix}_k$$

errが減るように状態の事前推定値をフィードバック修正し、事後推定値とする。

$$\begin{bmatrix} \hat{x}^- \\ \hat{y}^- \end{bmatrix}_k = \begin{bmatrix} \hat{x}^- \\ \hat{y}^- \end{bmatrix}_k + \boxed{\begin{bmatrix} k_{00} & k_{01} & k_{02} \\ k_{10} & k_{11} & k_{12} \end{bmatrix}_k} \begin{bmatrix} \mathrm{err}_0 \\ \mathrm{err}_1 \\ \mathrm{err}_2 \end{bmatrix}_k$$

誤差をfeedback
して事前推定値
を修正

事後推定値　事前推定値　カルマンゲイン　観測値と
観測値の推定値との誤差

$$\boldsymbol{K}_k = \boxed{\begin{bmatrix} k_{00} & k_{01} & k_{02} \\ k_{10} & k_{11} & k_{12} \end{bmatrix}_k}$$

センサーフュージョンとは何か？

カルマンゲインすなわちフィードバックの程度はどのように求めるか？

$$\boldsymbol{K}_{\mathrm{k}} = \begin{bmatrix} \mathrm{k}_{00} & \mathrm{k}_{01} & \mathrm{k}_{02} \\ \mathrm{k}_{10} & \mathrm{k}_{11} & \mathrm{k}_{12} \end{bmatrix}_{\mathrm{k}}$$

$$\boldsymbol{K}_{\boldsymbol{k}} = \boldsymbol{P}_{\boldsymbol{k}}^{-} \boldsymbol{H}_{\boldsymbol{k}}^{t} \left(\boldsymbol{H}_{\boldsymbol{k}} \boldsymbol{P}_{\boldsymbol{k}}^{-} \boldsymbol{H}_{\boldsymbol{k}}^{t} + \boldsymbol{R}_{\boldsymbol{k}} \right)^{-1}$$

$\boldsymbol{P}_{\boldsymbol{k}}^{-}$:事前推定値の推定誤差の共分散行列

$\boldsymbol{R}_{k}$:観測誤差の共分散行列

$$\boldsymbol{P}_{\boldsymbol{k}}^{+} = (\boldsymbol{I} - \boldsymbol{K}_{\boldsymbol{k}} \boldsymbol{H}_{\boldsymbol{k}}) \boldsymbol{P}_{\boldsymbol{k}}^{-}$$

:事後推定値の推定誤差の共分散行列

簡単のために、この例では、

$$\boldsymbol{R}_{k} = \begin{bmatrix} 1.0 & 0 & 0 \\ 0 & 1.0 & 0 \\ 0 & 0 & 1.0 \end{bmatrix} \quad (\mathrm{m}^2)$$

$$\boldsymbol{Q}_{k}^{-} = \begin{bmatrix} 1.0 & 0 \\ 0 & 1.0 \end{bmatrix} \quad (\mathrm{m}^2)$$

とする。

センサーフュージョンとは何か？

カルマンゲイン

$$\boldsymbol{K}_k = \boldsymbol{P}_k^- \boldsymbol{H}_k^t \left(\boldsymbol{H}_k \boldsymbol{P}_k^- \boldsymbol{H}_k^t + \boldsymbol{R}_k\right)^{-1}$$

この導出は省略

$\boldsymbol{P}_k^-$:事前推定値の推定誤差の共分散行列

$\boldsymbol{R}_k$:観測誤差の共分散行列

$$\boldsymbol{P}_k^+ = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k)\boldsymbol{P}_k^-$$

:事後推定値の推定誤差の共分散行列

簡単のために、この例では、

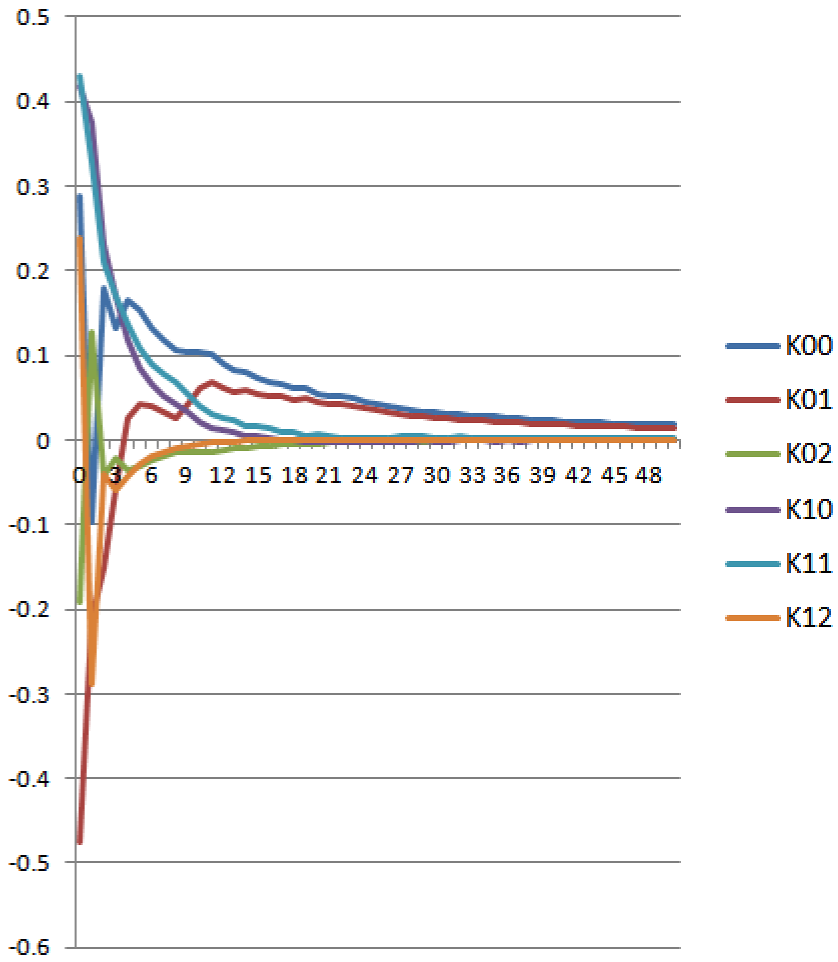
$$\boldsymbol{Q}_k^- = \sigma_Q^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (\mathrm{m}^2)$$

$$\boldsymbol{R}_k = \sigma_R^2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\mathrm{m}^2)$$

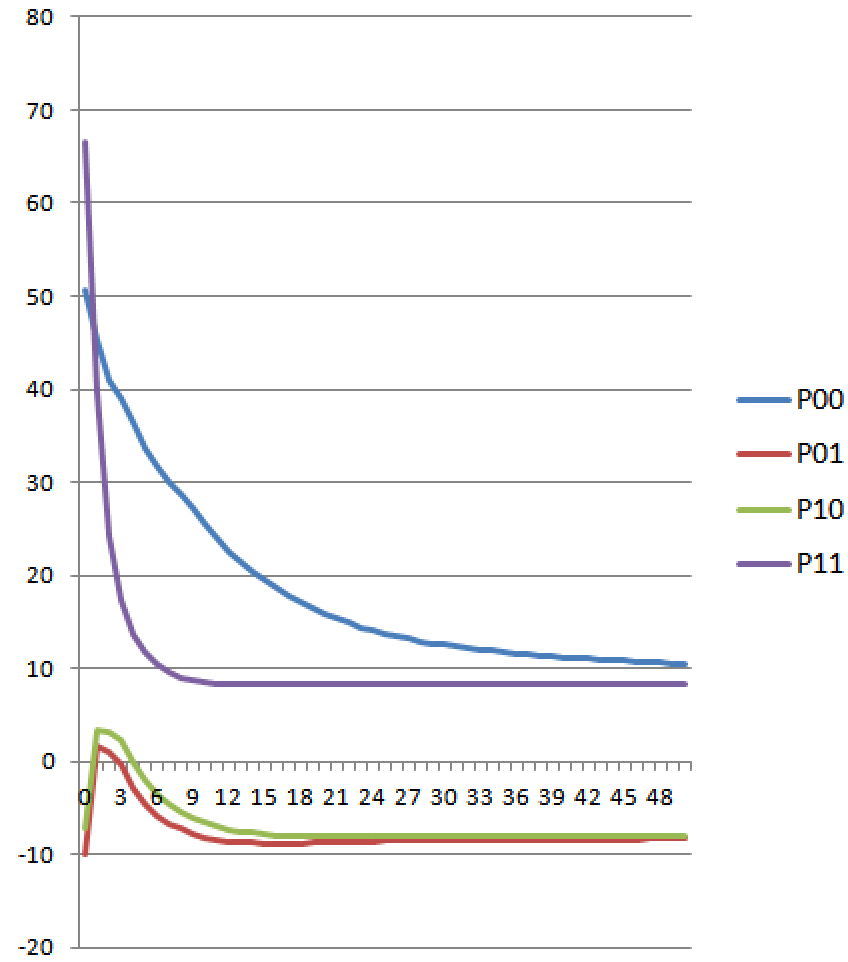
とする。

実行結果

カルマンゲイン
50クロック程度で収束

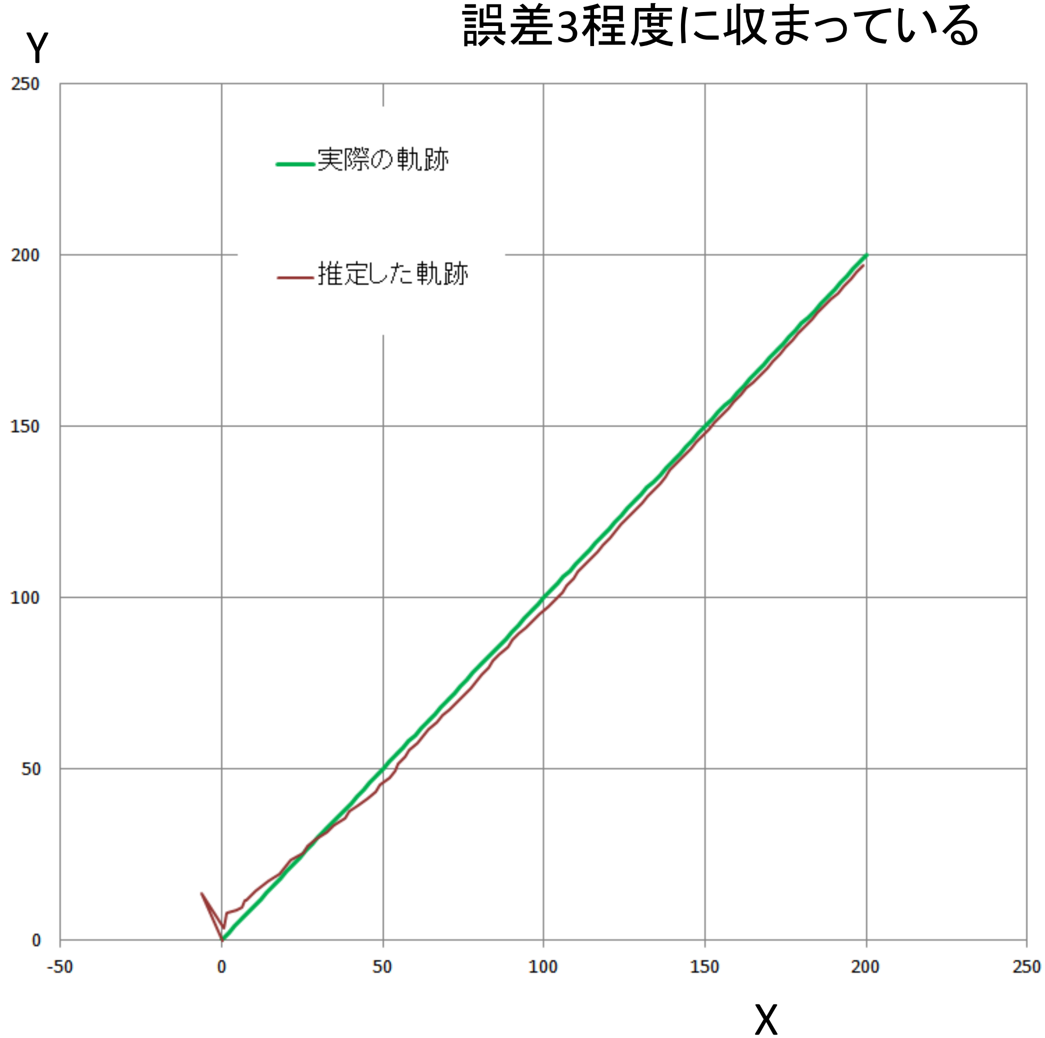


推定誤差の共分散行列
50クロック程度で収束

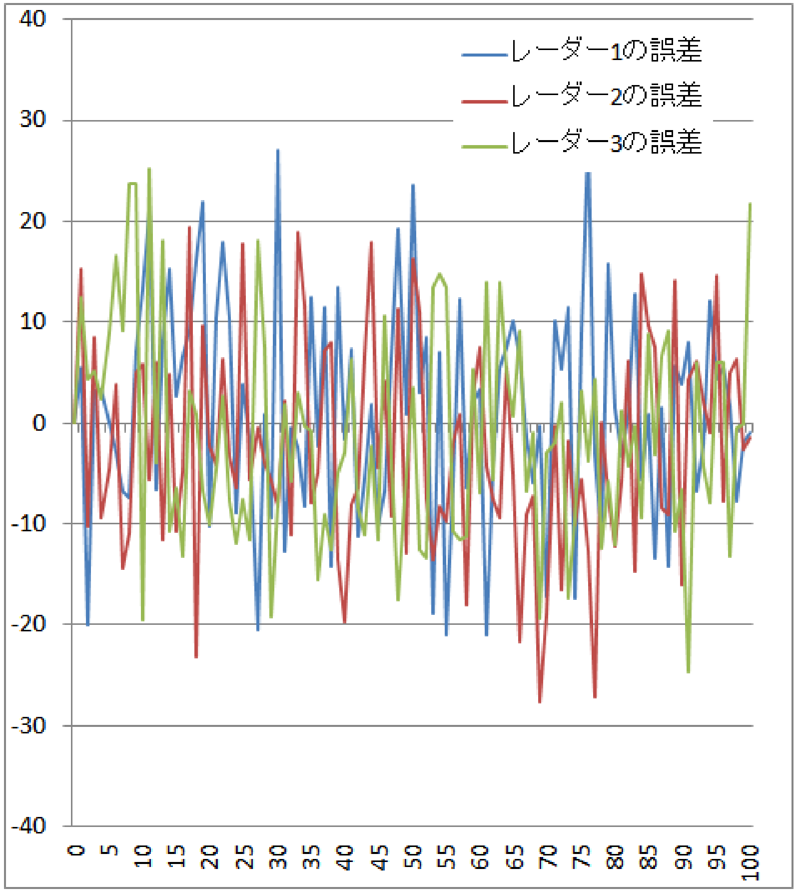


実行結果

誤差3程度に収まっている

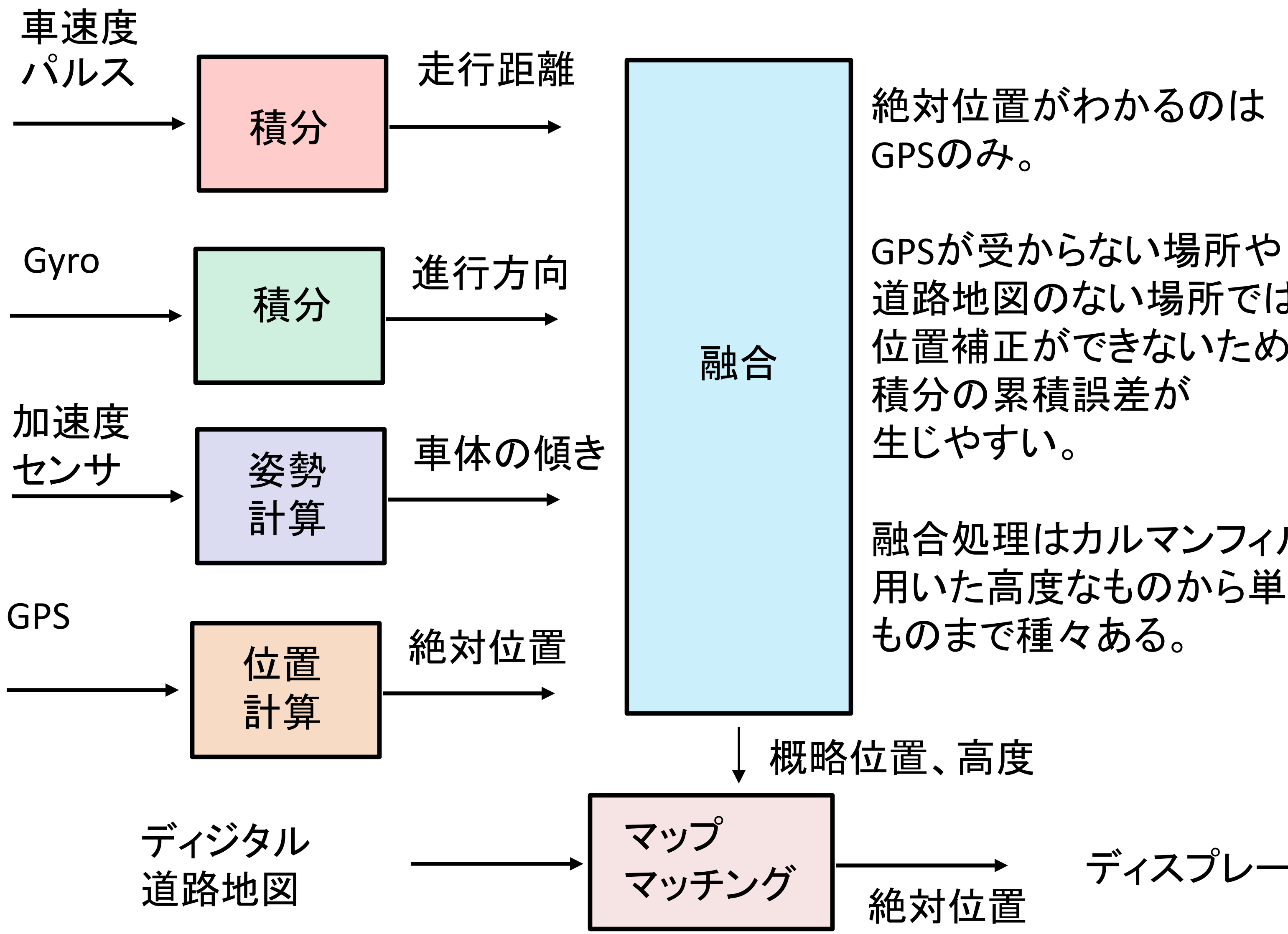


標準偏差　10の誤差



先週に引き続きセンサーフュージョンの実例を
紹介する。

センサーフュージョンとは何か？



絶対位置がわかるのは
GPSのみ。

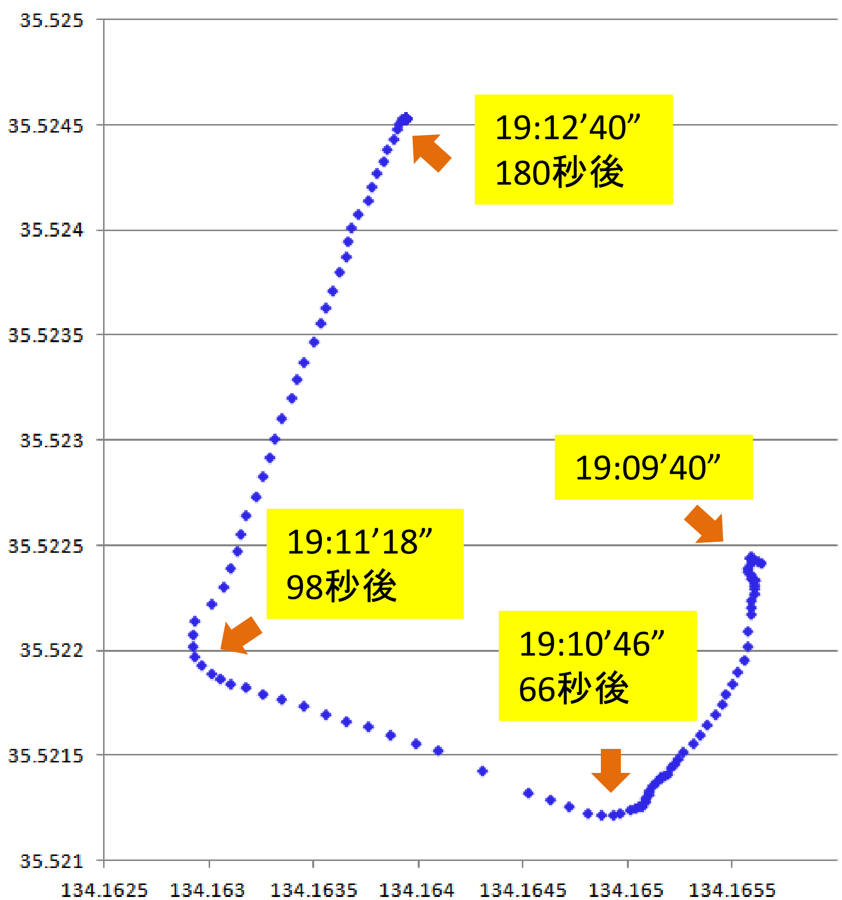
GPSが受からない場所や
道路地図のない場所では
位置補正ができないため
積分の累積誤差が
生じやすい。

融合処理はカルマンフィルタを
用いた高度なものから単純な
ものまで種々ある。

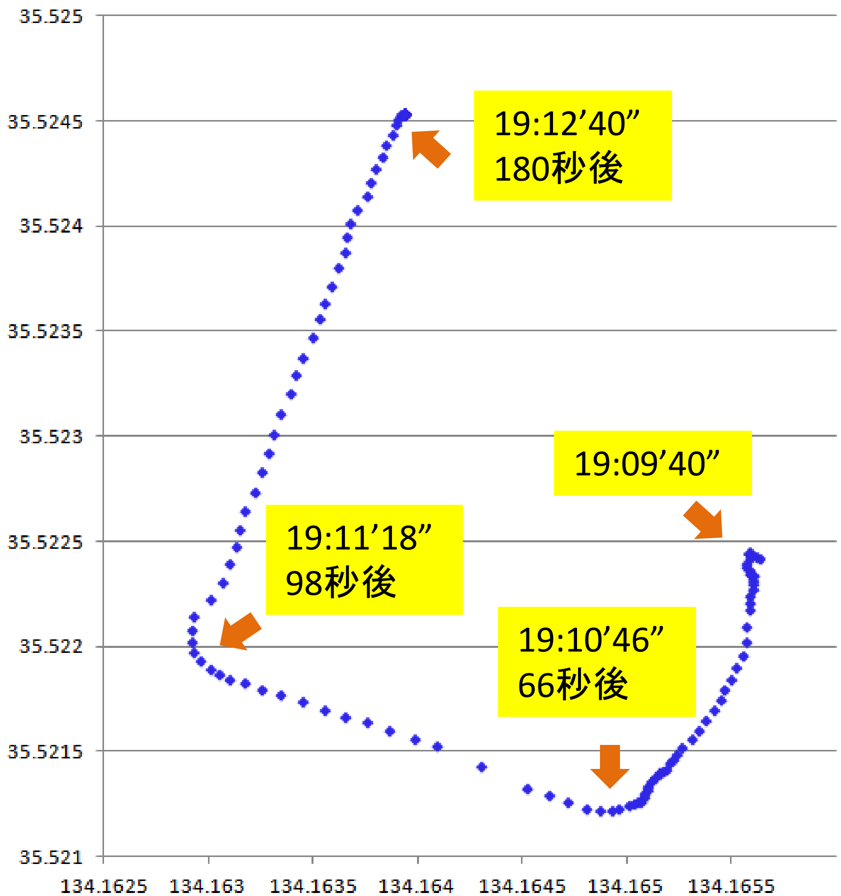
車速パルスのないスマートフォンでINSを行う。

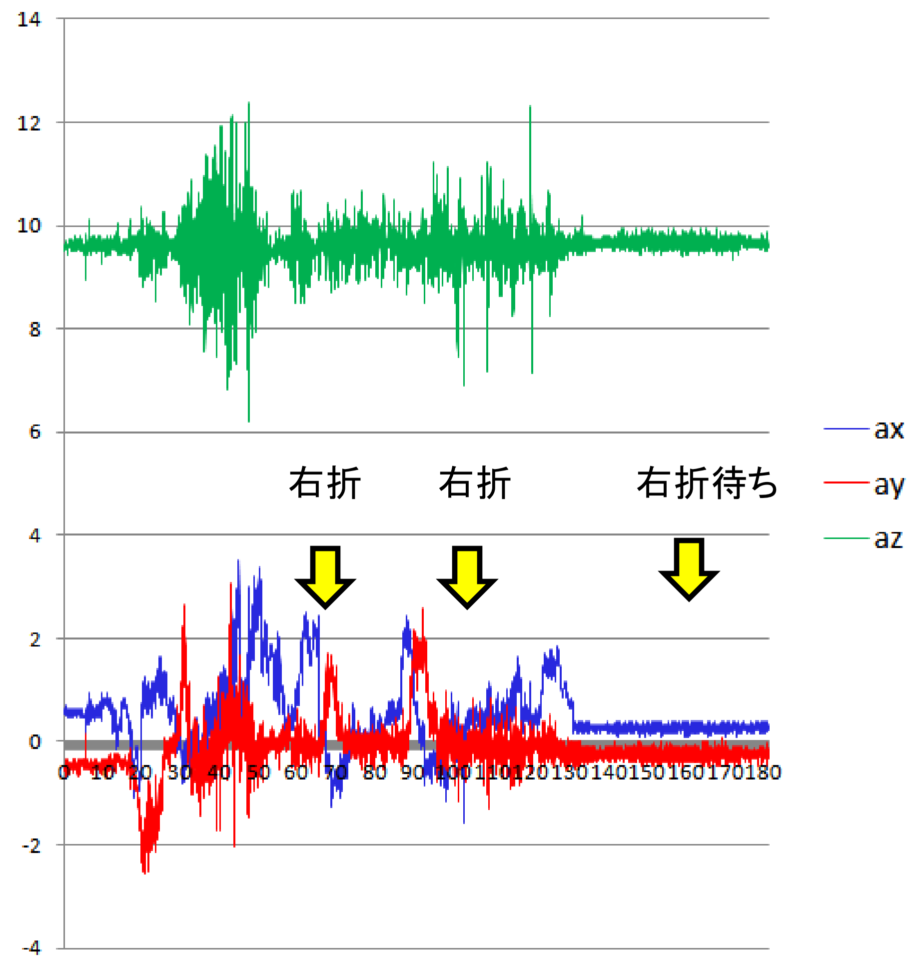
実験データ

白浜宿舎駐車場から出発して3分後に停止するまでの軌跡
2012.7.26 19:09’40’’～19:12’:40’’



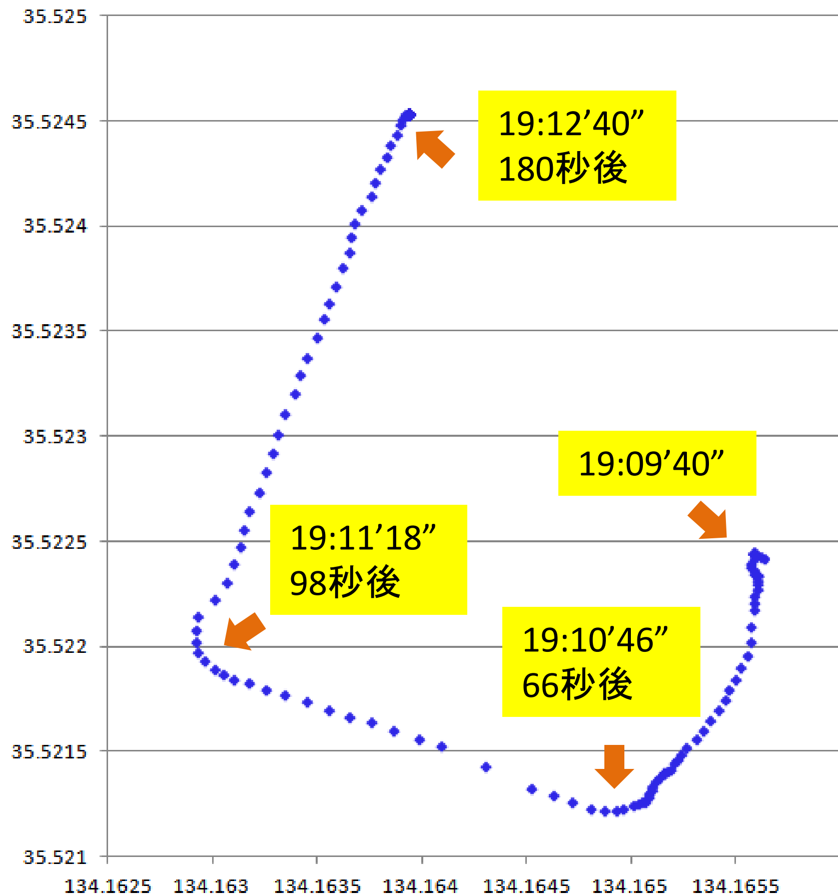
車速パルスのないスマートフォンでINSを行う。

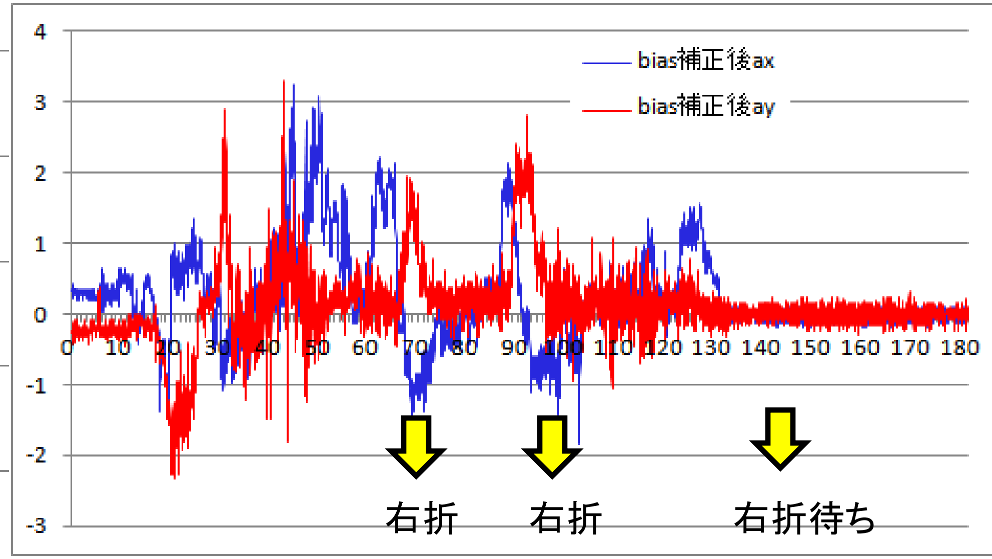


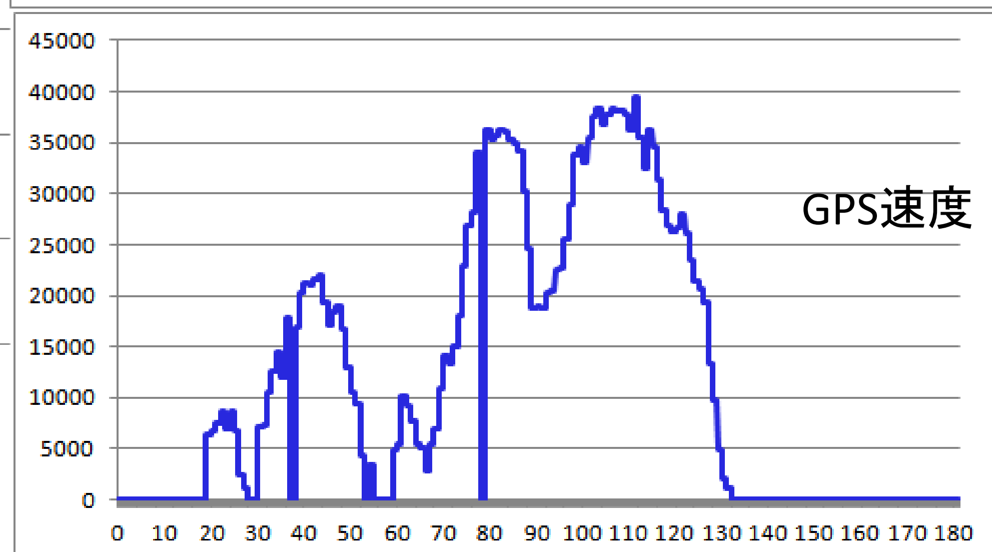


車速パルスのないスマートフォンでINSを行う。

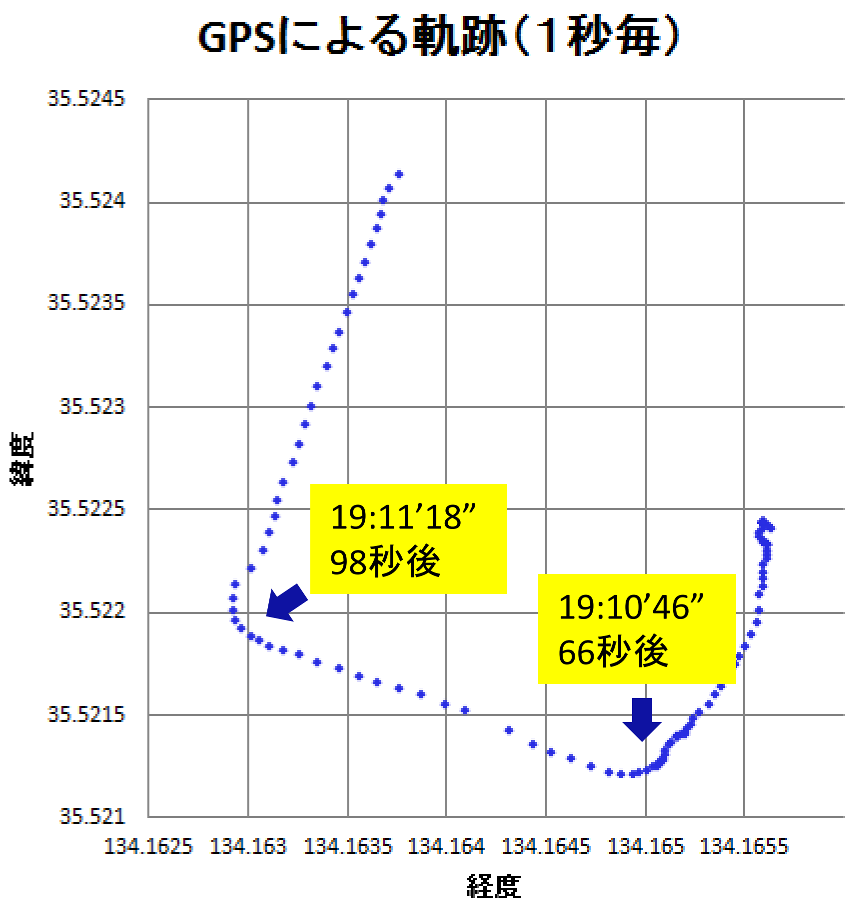
バイアス補正



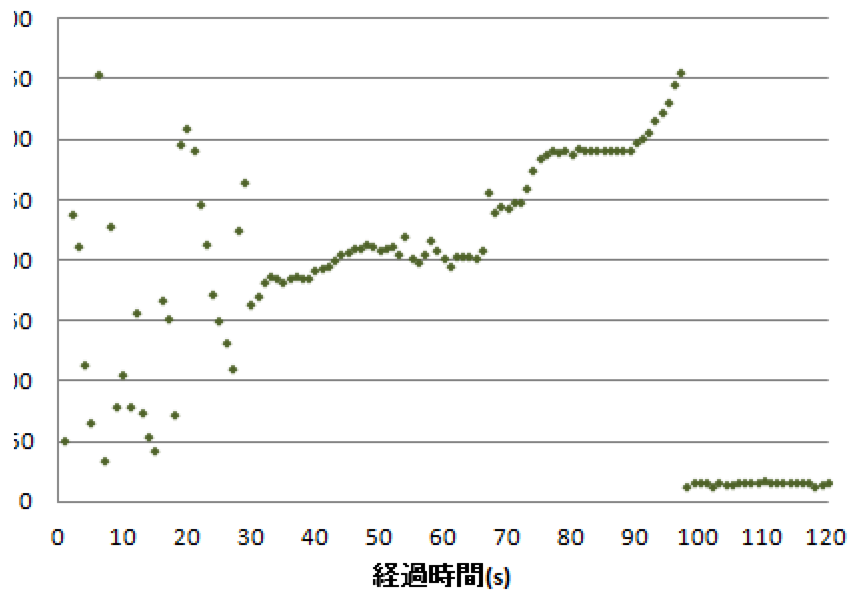


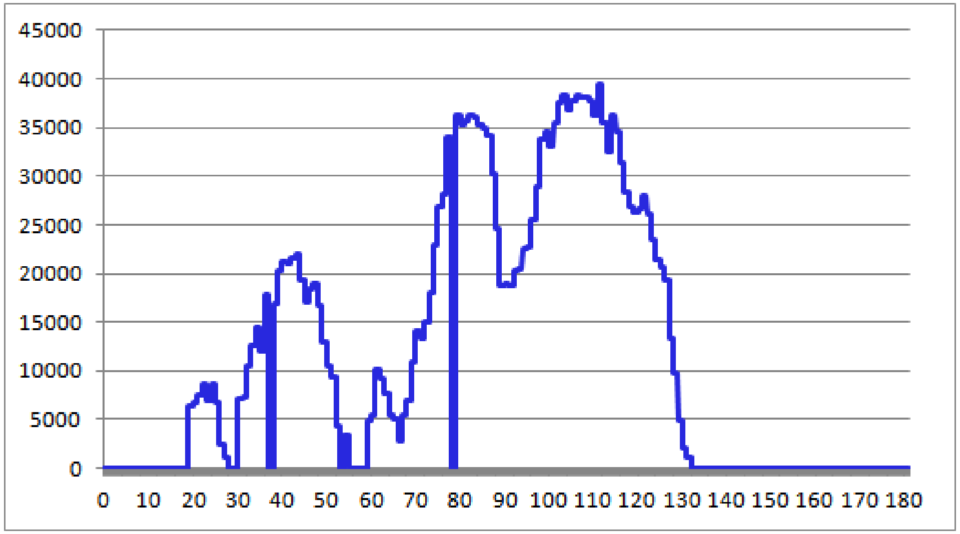


車速パルスのないスマートフォンでINSを行う。

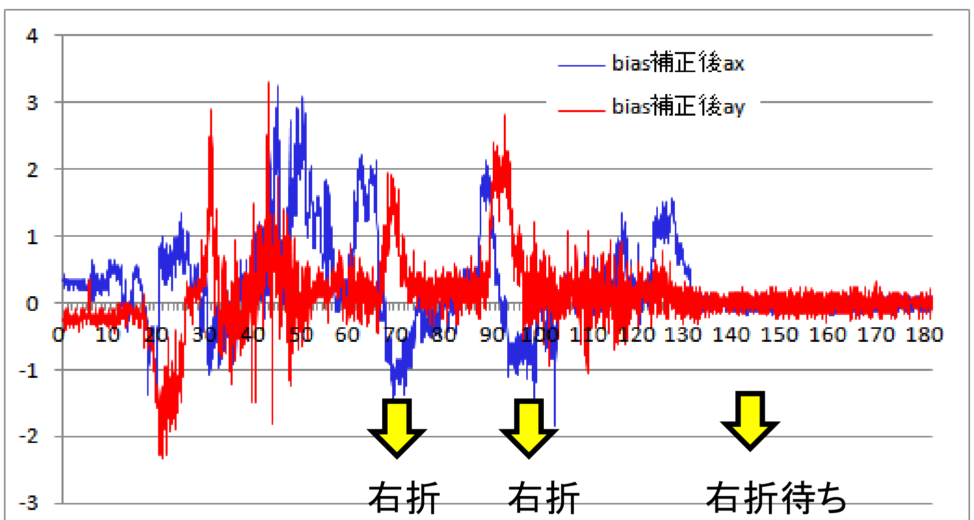


方位変化 （GPS)

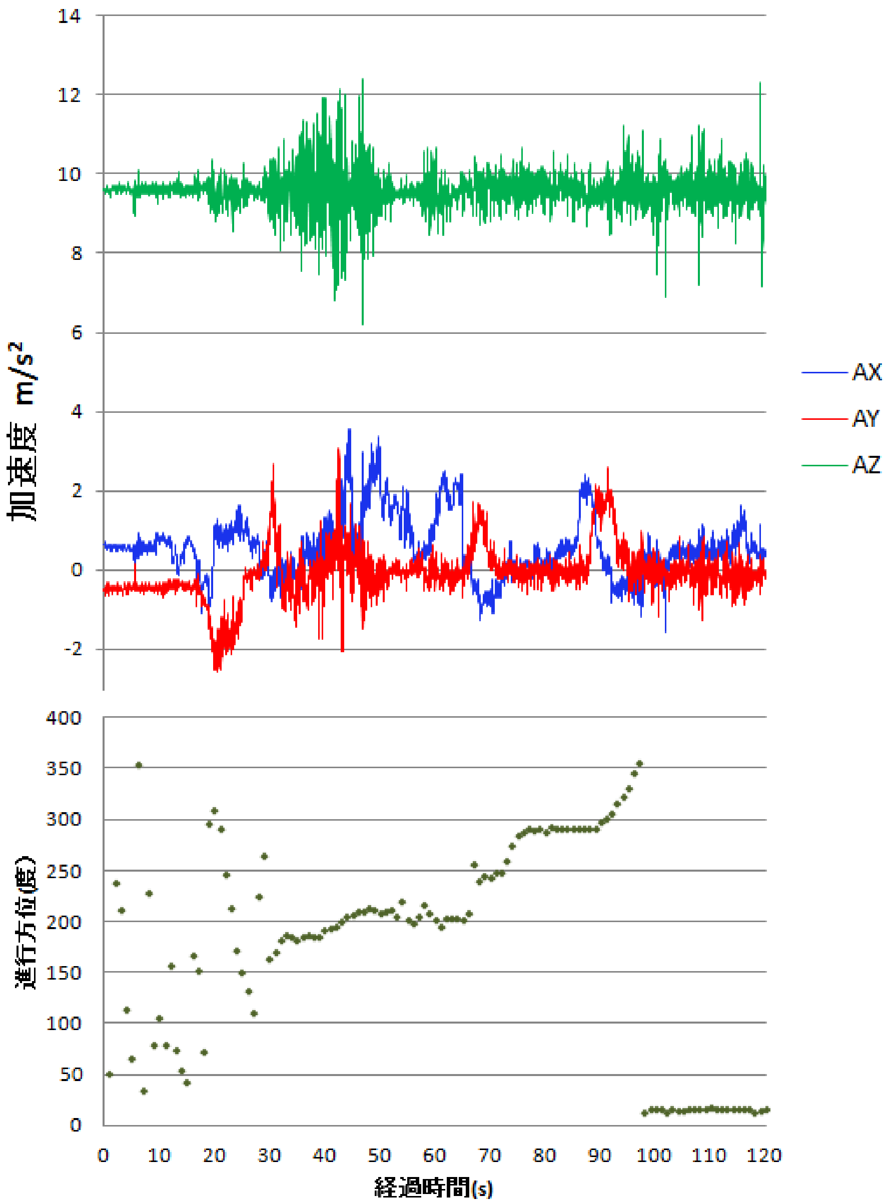


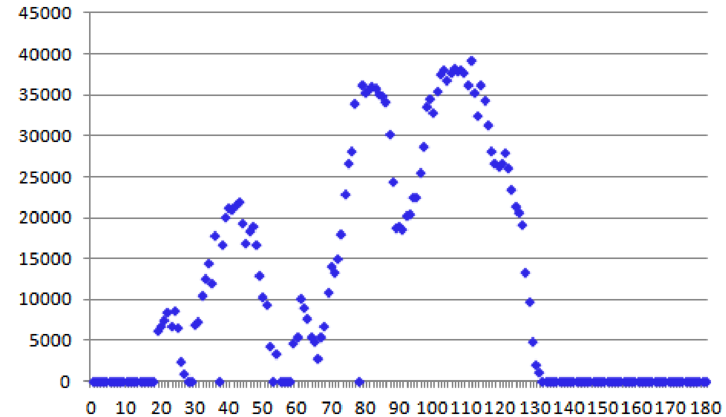


GPSの停止判定毎に
静的なバイアス（傾き）
補正での限界



車速パルスのないスマートフォンでINSを行う。 観測データ

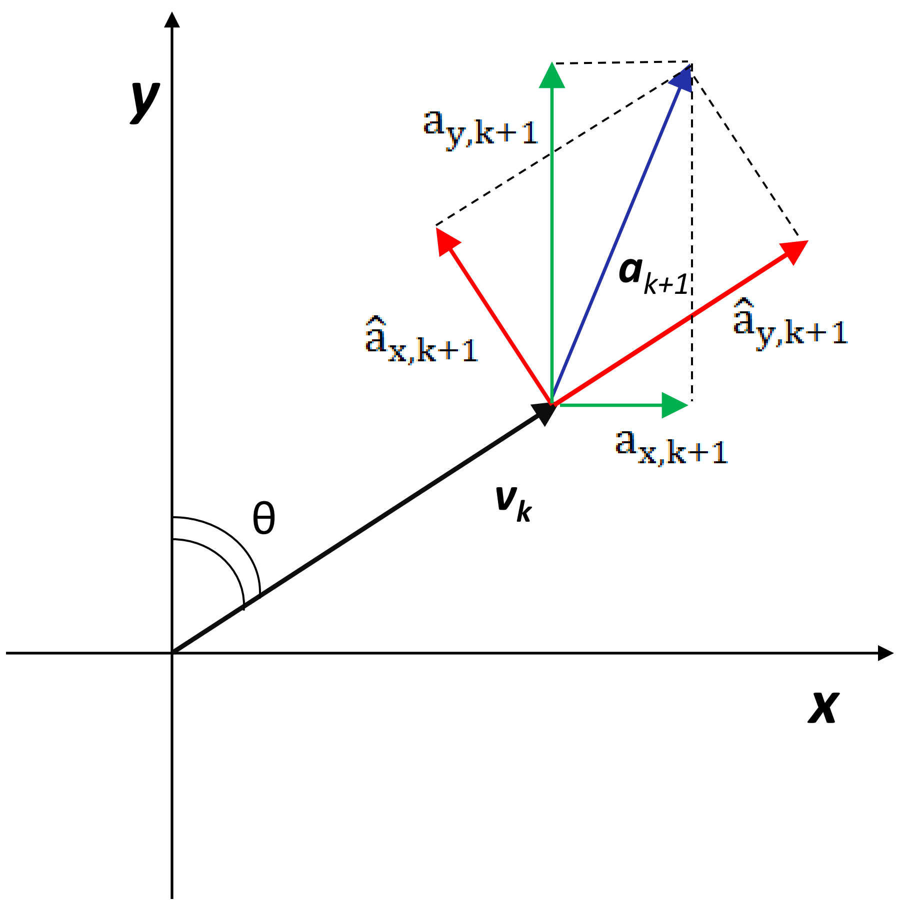




GPS速度

車速パルスのないスマートフォンでINSを行う。

スマートフォン座標系から地球座標系への変換



$$\boldsymbol{a}_{\mathrm{k}} = \hat{\mathrm{a}}_{\mathrm{x}}\boldsymbol{i}_{\boldsymbol{s}} + \hat{\mathrm{a}}_{\mathrm{y}}\boldsymbol{j}_{\boldsymbol{s}} + (\hat{\mathrm{a}}_{\mathrm{z}} - \mathrm{G})\boldsymbol{k}_{\boldsymbol{s}} \cong \hat{\mathrm{a}}_{\mathrm{x}}\boldsymbol{i}_{\boldsymbol{s}} + \hat{\mathrm{a}}_{\mathrm{y}}\boldsymbol{j}_{\boldsymbol{s}}$$

$$\boldsymbol{a}_{\mathrm{k}} = \mathrm{a}_{\mathrm{x}}\boldsymbol{i} + \mathrm{a}_{\mathrm{y}}\boldsymbol{j} + (\mathrm{a}_{\mathrm{z}} - \mathrm{G})\boldsymbol{k} \cong \mathrm{a}_{\mathrm{x}}\boldsymbol{i} + \mathrm{a}_{\mathrm{y}}\boldsymbol{j}$$

$$\begin{bmatrix} a_{x,k+1} \\ a_{y,k+1} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \hat{a}_{x,k+1} \\ \hat{a}_{y,k+1} \end{bmatrix}$$

Θ=π/4だと、$\begin{bmatrix} a_{x,k+1} \\ a_{y,k+1} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \hat{a}_{x,k+1} \\ \hat{a}_{y,k+1} \end{bmatrix}$

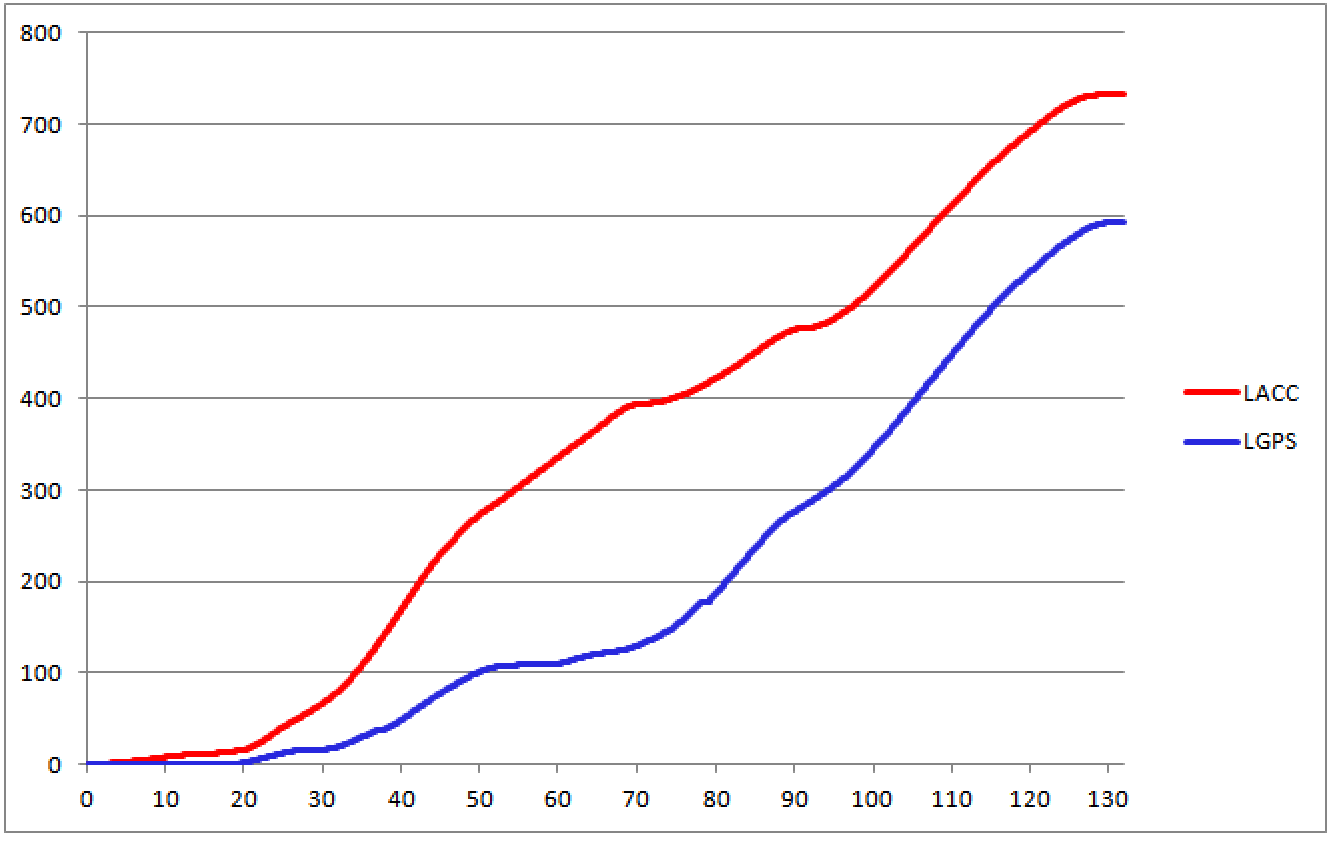
Θ=0だと、$\begin{bmatrix} a_{x,k+1} \\ a_{y,k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{a}_{x,k+1} \\ \hat{a}_{y,k+1} \end{bmatrix}$

Θ=π/2だと、$\begin{bmatrix} a_{x,k+1} \\ a_{y,k+1} \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \hat{a}_{x,k+1} \\ \hat{a}_{y,k+1} \end{bmatrix}$

車速パルスのないスマートフォンでINSを行う。

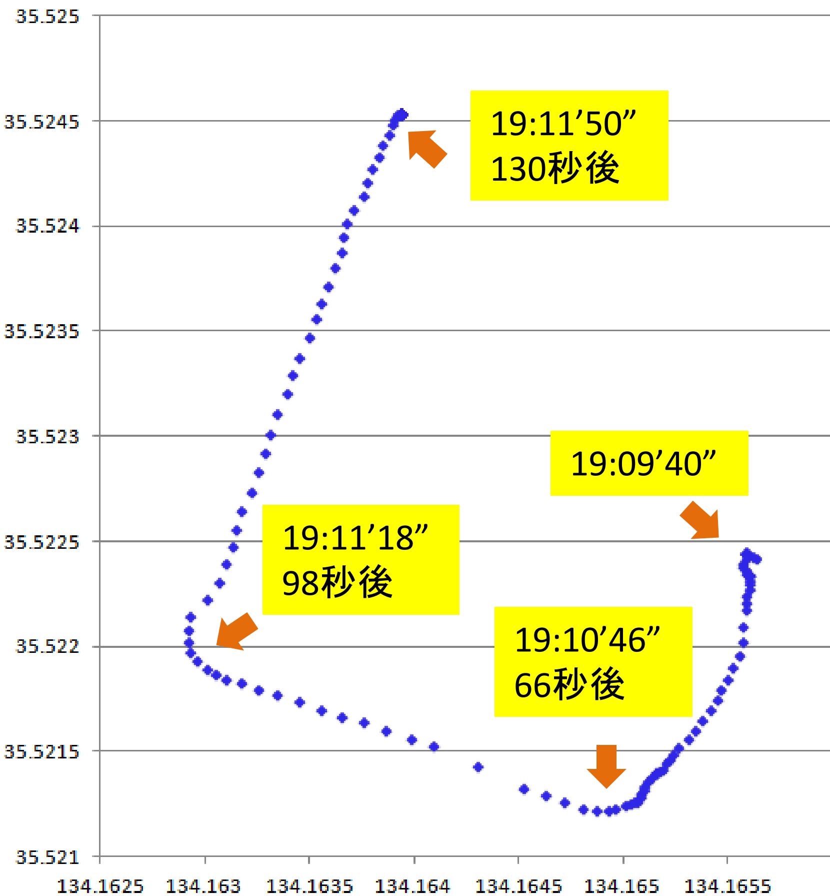
GPSによる積算距離と加速度センサーによる積算距離では
このような誤差が発生する。

積算距離
(m)

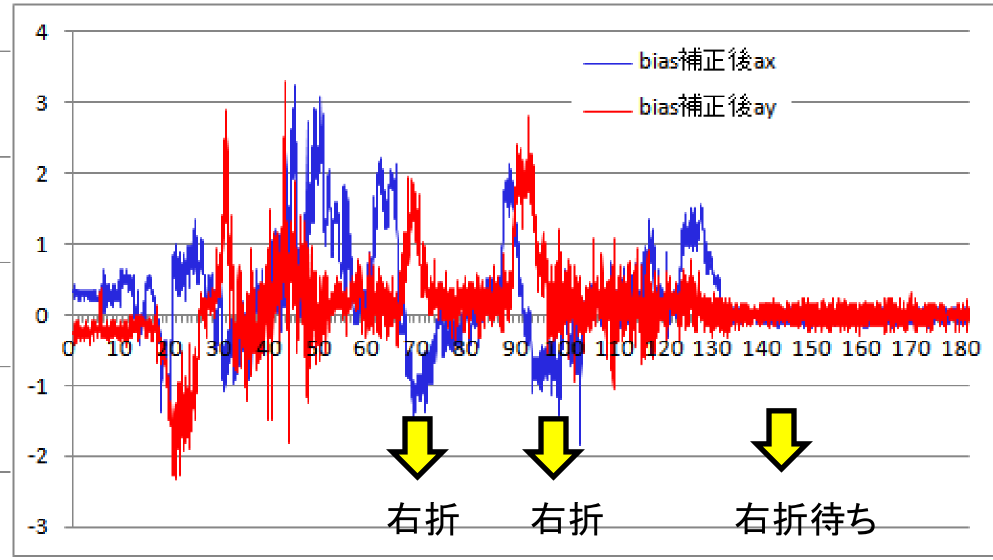


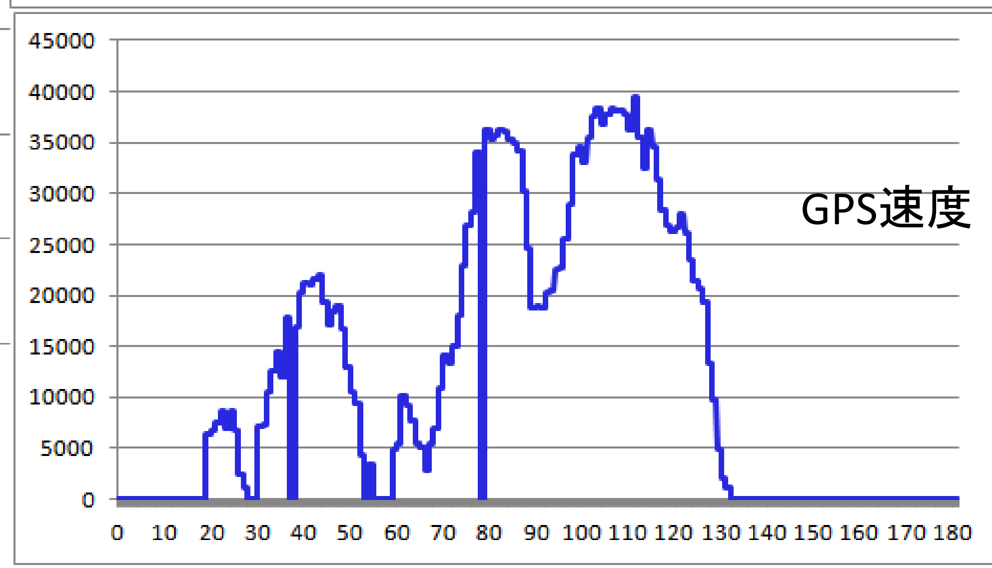
経過時間(s)

車速パルスのないスマートフォンでINSを行う。

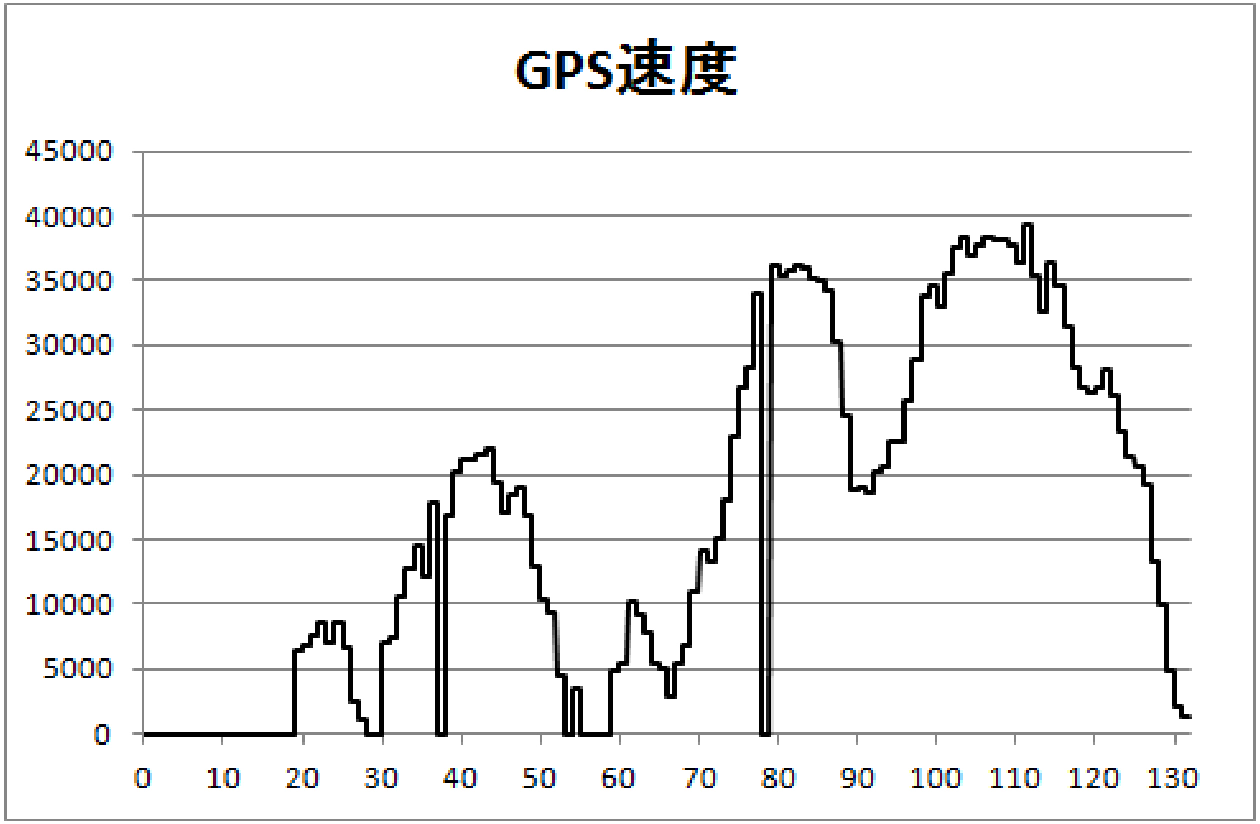


バイアス補正(スマホ座標系)





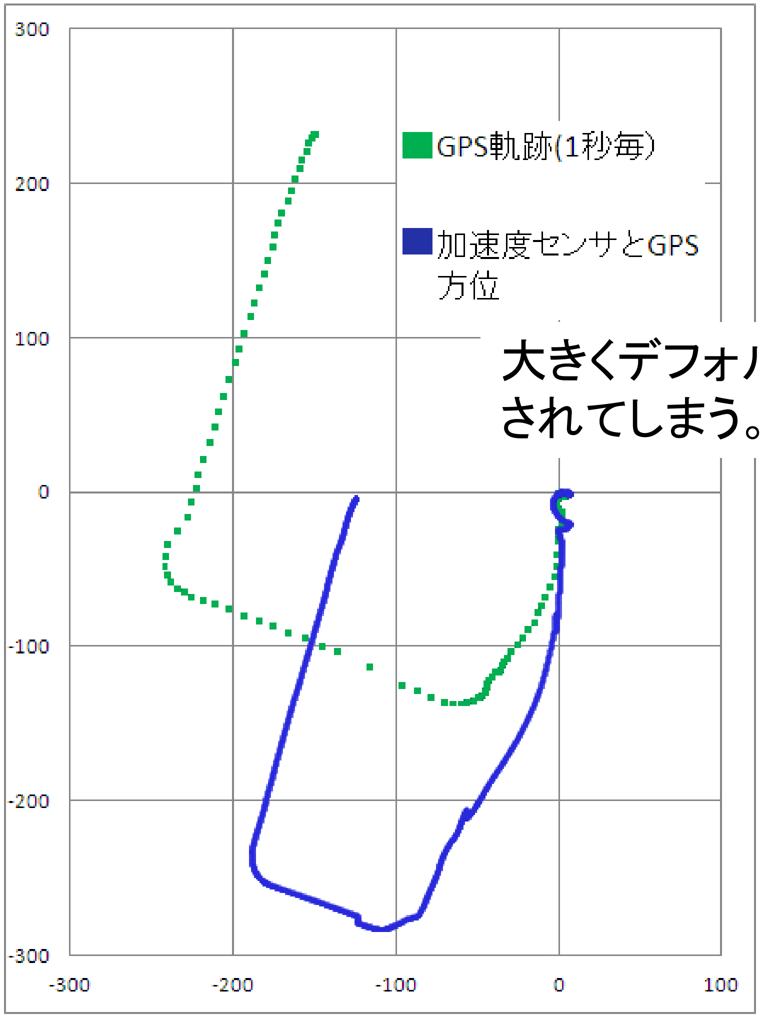
車速パルスのないスマートフォンでINSを行う。



GPSで得られる速度は測位に失敗すると算出できないような瞬断が生じる場合がある。

車速パルスの取得できないスマートフォンでINSを行う。

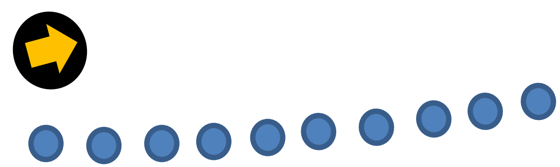
加速度センサの傾きによって重力加速度の
他軸への混入などが影響して累積距離の誤差が大きくなる。



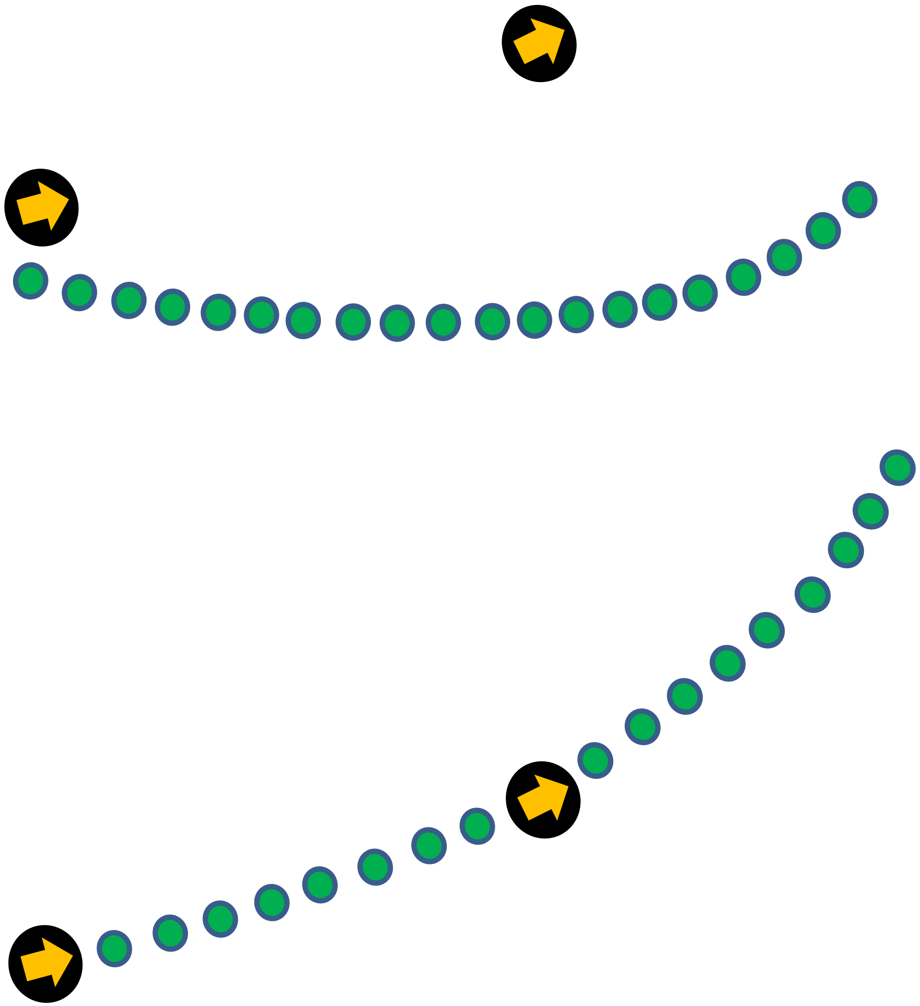
大きくデフォルメ
されてしまう。

GPSによる
絶対位置補正
速度補正
方位補正

を実施。（1秒ピッチ）



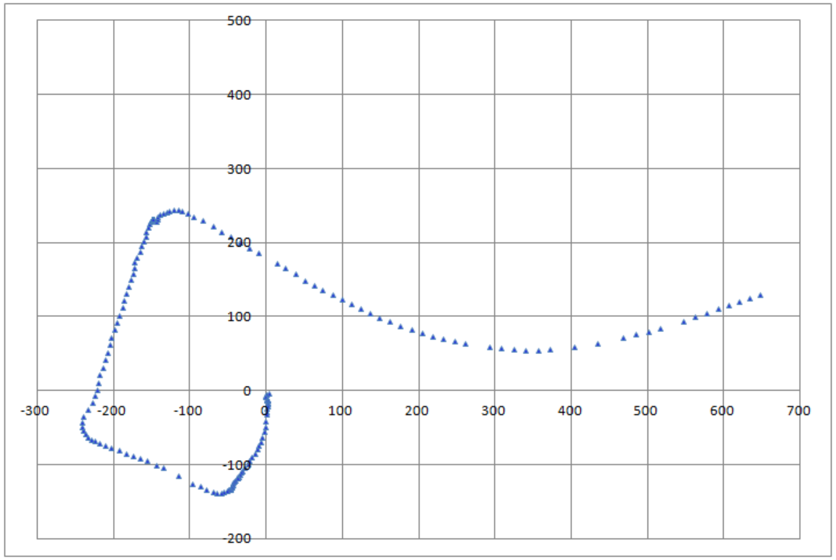
車速パルスの取得できないスマートフォンでINSを行う。



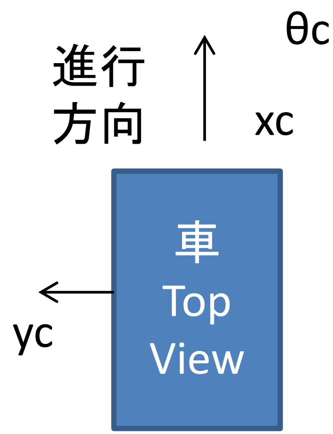
GPSによる
絶対位置補正
速度補正
方位補正

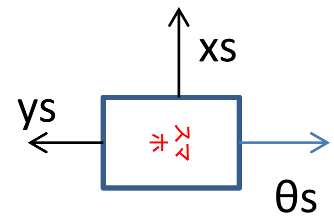
を実施。（１秒ピッチ）

実験データはAndroidのスマートフォンで収集,GPSは別途



2012年
19:10-19:14





座標軸の対応付けが混乱しやすいので要注意。

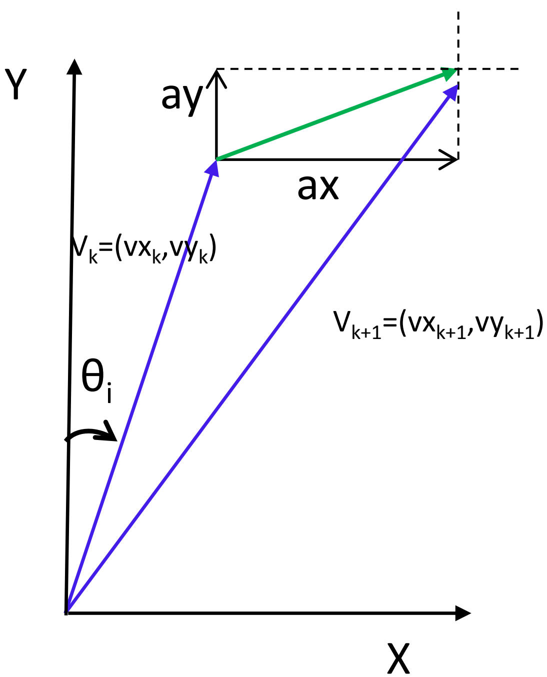
加速度からの速度ベクトルの算出

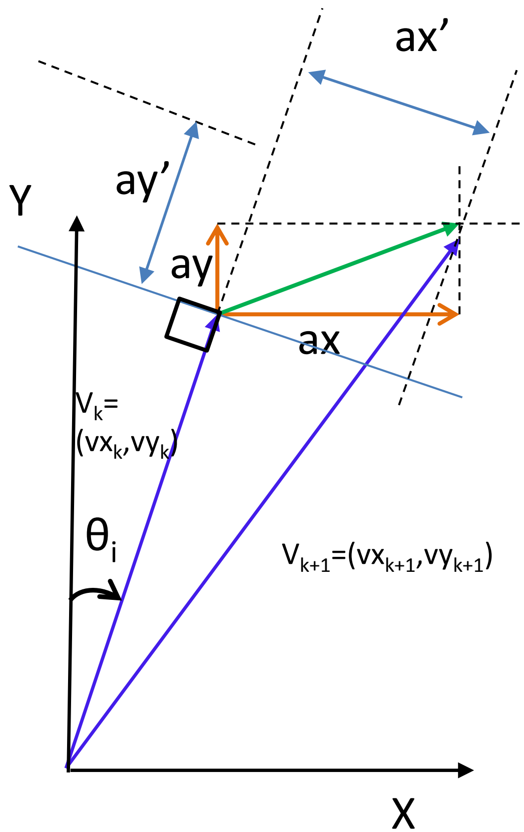
慣習的に進行方向がx軸。
車速パルス（Wheel Tick)のように、移動距離に比例した出力を得るセンサがない場合、加速度から速度を算出する必要がある。

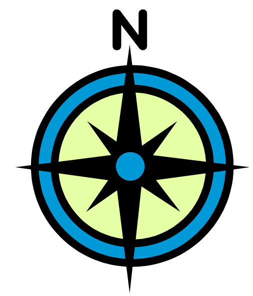
$$vx_{k+1}=vx_k+\varDelta t\cdot ax_k$$

$$vy_{k+1}=vy_k+\varDelta t\cdot ay_k$$

$$\begin{bmatrix} a_x \\ a_y \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} a'_x \\ a'_y \end{bmatrix}$$





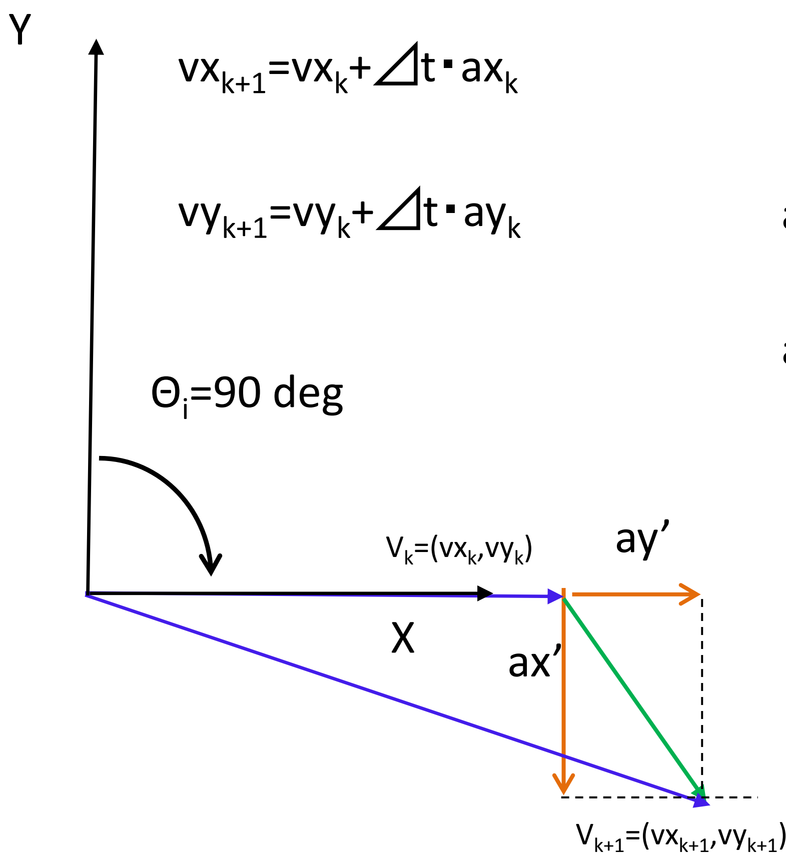


加速度からの速度ベクトルの算出

車速パルス(Wheel Tick)のように、移動距離に比例した出力を得るセンサがない場合、加速度から速度を算出する必要がある。

$vx_{k+1}=vx_k+\Delta t\cdot ax_k$

$vy_{k+1}=vy_k+\Delta t\cdot ay_k$



$$\begin{bmatrix} a_x \\ a_y \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} a'_x \\ a'_y \end{bmatrix}$$

ax=a'y

ay=-a'x

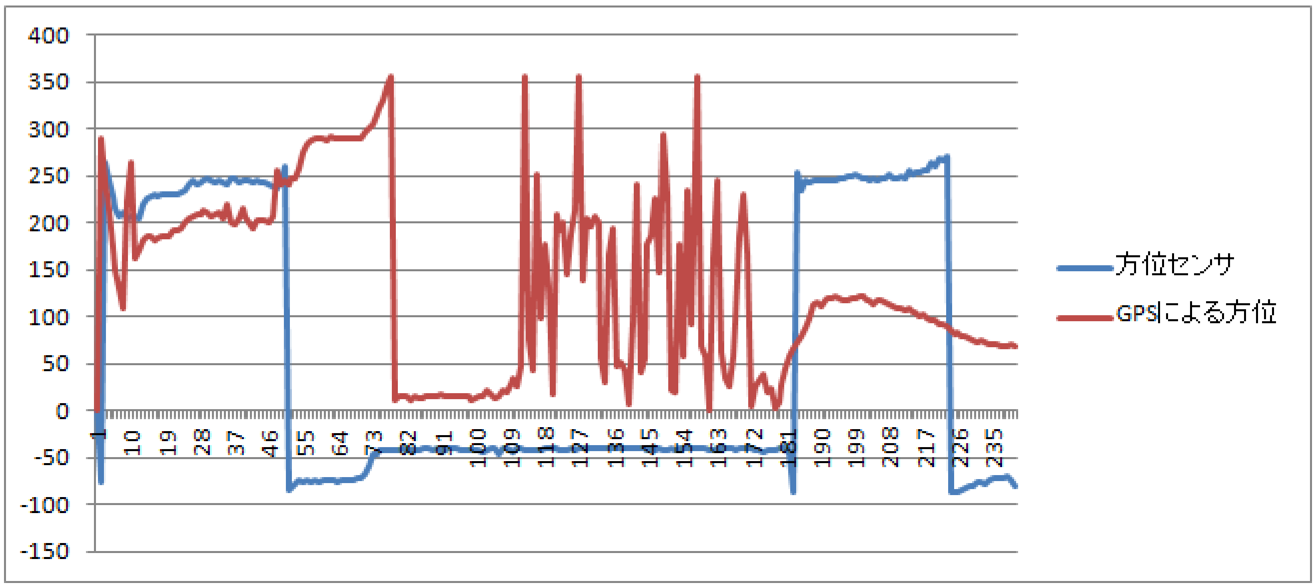
スマホで感じる加速度
ax',ay'
地球座標系での加速度
ax,ay

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix}_{k+1} = \begin{bmatrix} v_x \\ v_y \end{bmatrix}_k + \Delta t \begin{bmatrix} a_x \\ a_y \end{bmatrix}_k$$

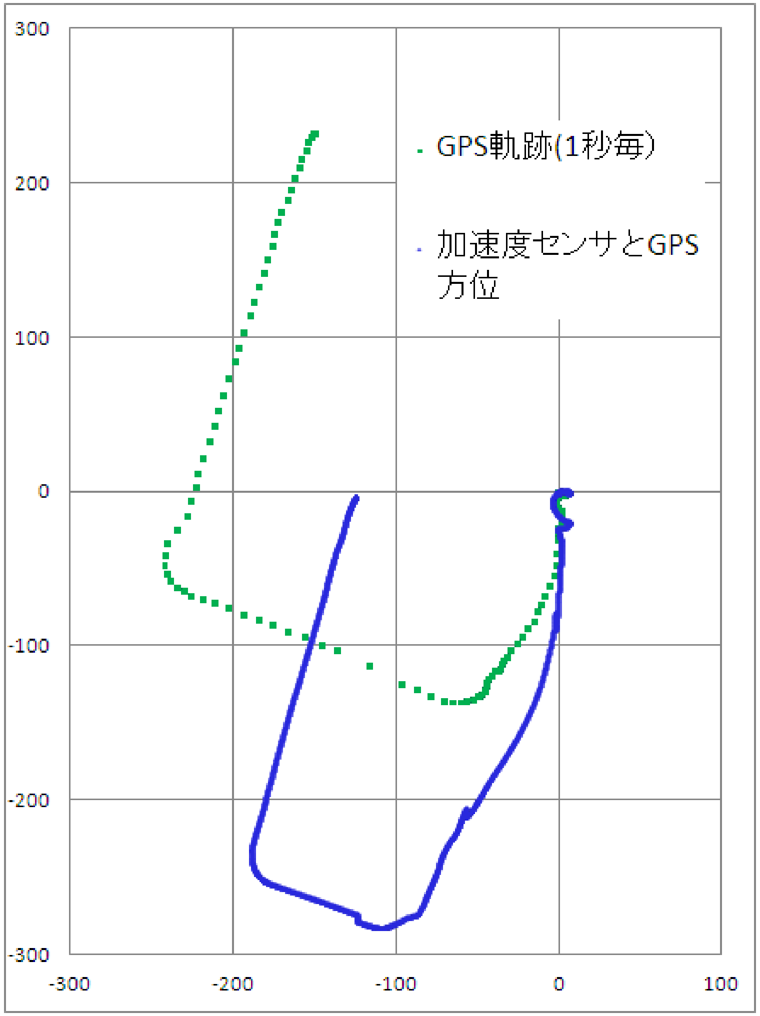
$$= \begin{bmatrix} v_x \\ v_y \end{bmatrix}_k + \Delta t \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}_k \begin{bmatrix} a'_x \\ a'_y \end{bmatrix}_k$$

方位のリファレンスデータ

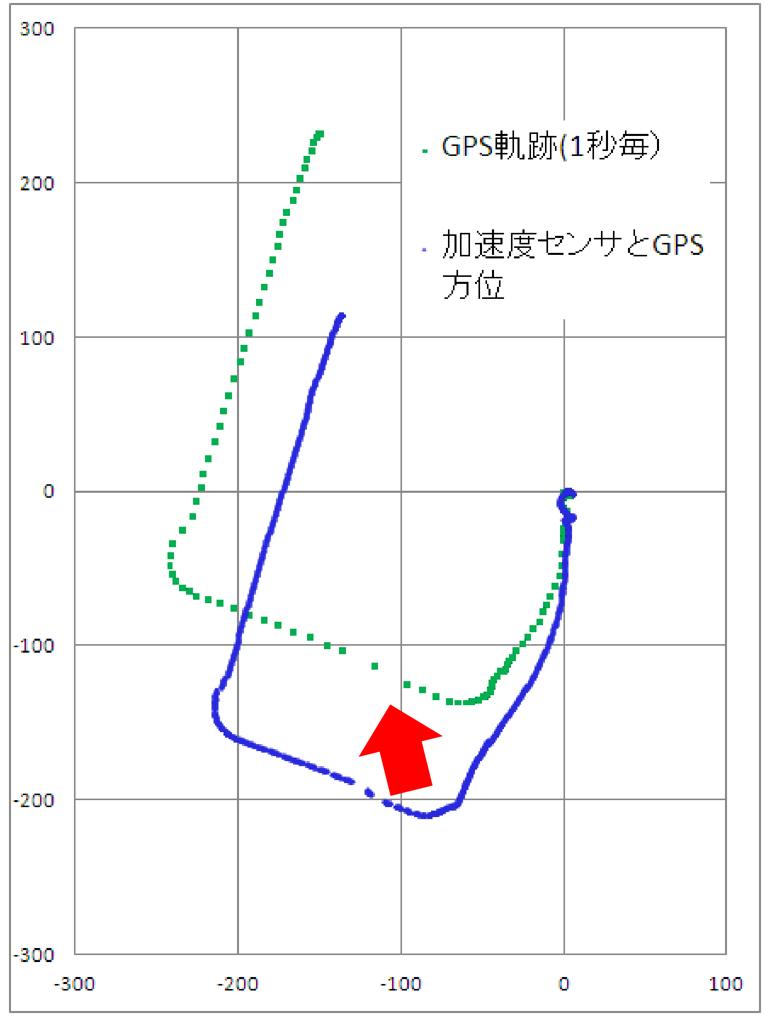
地磁気を用いた方位センサのとGPSの出力する方位を
比較した。走行中であればGPSは位置の差分から
良好な絶対方位を算出できる。
一方、地磁気センサ出力は一見安定であるが、
車内では金属の影響で大きな誤差が生じてしまい
利用不可であった。
従って、ジャイロとGPSの組み合わせが好ましい。



加速度センサとGPSの融合

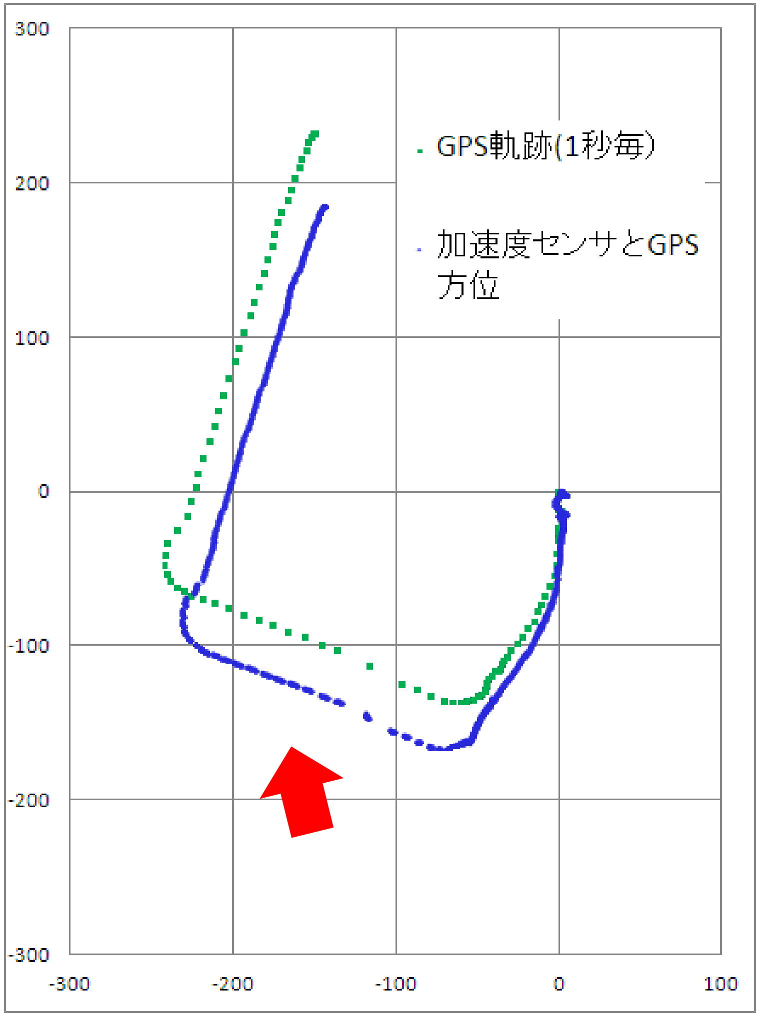


融合　0%

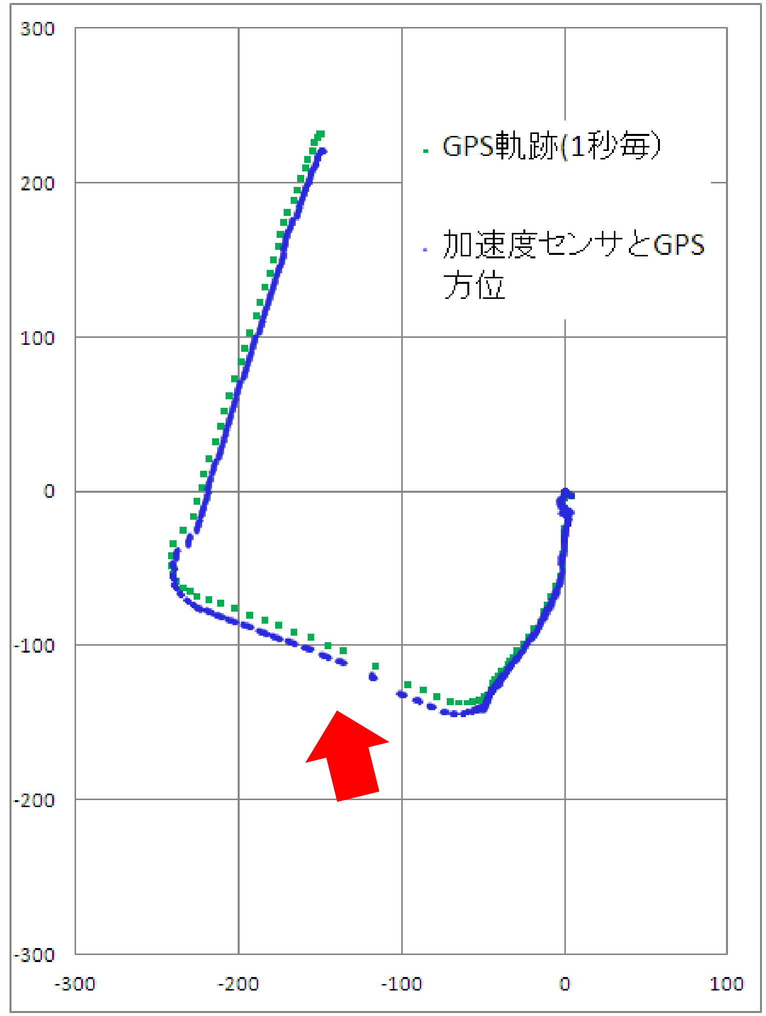


融合　50%

加速度センサとGPSの融合

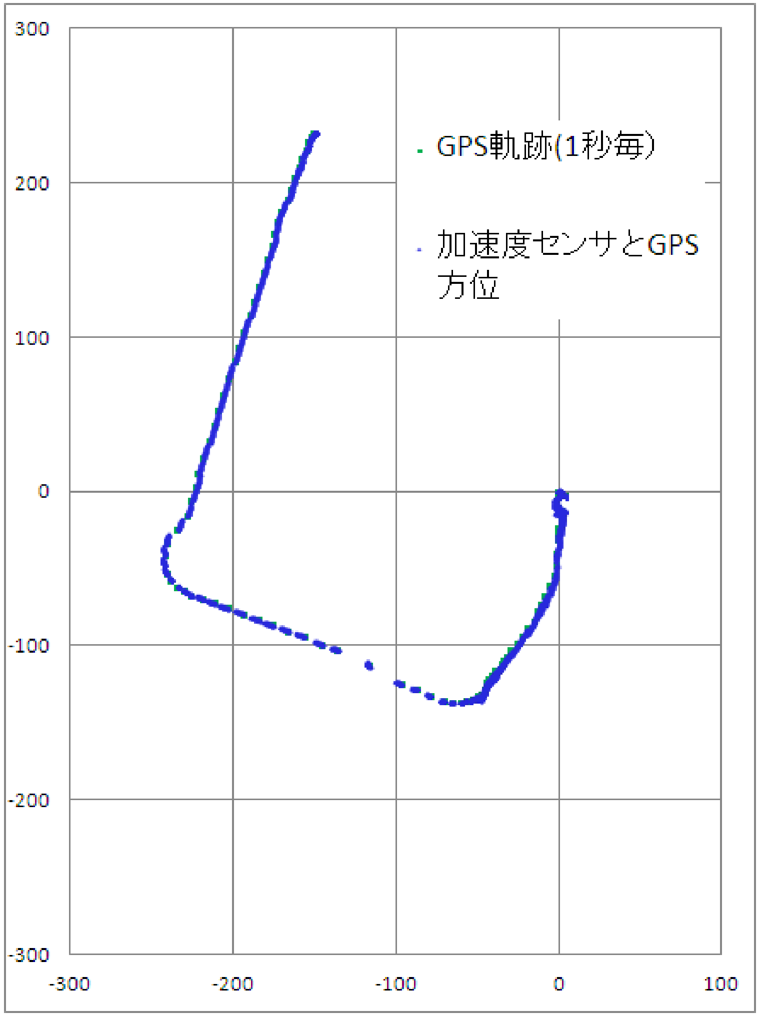


融合　80



融合　95%

加速度センサとGPSの融合



融合　100%

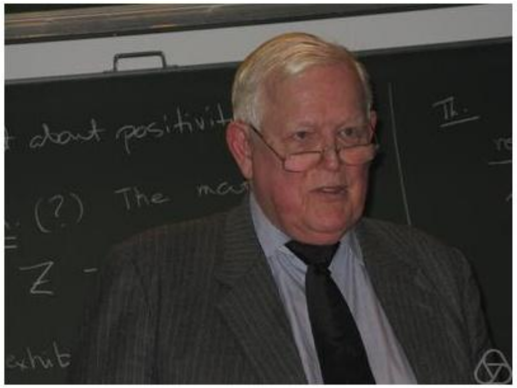
絶対位置に関しては加速度センサより計算した値よりもGPSの方が信頼できる。

従って、100% 絶対位置はGPSによるものを採用する。

ただし、1秒毎にしか得られない。

残りの980msecは加速度センサの値から積分した走行距離を採用する。

カルマンフィルタ



ルドルフ・カルマン（Rudolf Emil Kalman、1930年5月16日 - ）

ハンガリー系アメリカ人

米国国家科学賞(2008年)

カルマンフィルタについて分かりやすく説明された文献は非常に少ない。
考え方：
複数の情報源からの情報をブレンドして最終的な情報を求める。
信頼性の高い情報源の情報ほど高い重みでブレンドする。
それによって信頼性の高い情報を得ることができる。

カルマンフィルターの特性は扱うデータに依存して決定されるために、
説明が難しいのも原因。

カルマンフィルタの基礎

カルマンフィルタは一群の方程式
二乗誤差を最小にするようにプロセスの状態を逐次推定する。

状態方程式	$\boldsymbol{x}_k = \boldsymbol{A}\boldsymbol{x}_{k-1} + \boldsymbol{B}\boldsymbol{u}_{k-1} + \boldsymbol{w}_{k-1}$
状態ベクトル	$\boldsymbol{x} \in \boldsymbol{R}^n$
状態遷移行列	$\boldsymbol{A}$
制御入力ベクトル	$\boldsymbol{u} \in \boldsymbol{R}^l$
制御入力を状態に反映させる行列	$\boldsymbol{B}$
プロセス雑音	$\boldsymbol{w}_k$
観測方程式	$\boldsymbol{z}_k = \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{v}_k$
観測値ベクトル	$\boldsymbol{z} \in \boldsymbol{R}^m$
観測行列	$\boldsymbol{H}$
観測雑音	$\boldsymbol{v}_k$

プロセス雑音 $\boldsymbol{w}_k$ と観測雑音 $\boldsymbol{v}_k$ は独立

プロセス雑音 $\boldsymbol{w}_k$	観測雑音 $\boldsymbol{v}_k$
分布は $N(0, \boldsymbol{Q})$	分布は $N(0, \boldsymbol{R})$
$\boldsymbol{Q}$ 共分散行列	$\boldsymbol{R}$ 共分散行列

カルマンフィルタの基礎

事前推定値 $\hat{\boldsymbol{x}}_k^- \in \boldsymbol{R}^n$

事後推定値 $\hat{\boldsymbol{x}}_k \in \boldsymbol{R}^n$ 観測値 z_k を反映して補正

事前推定値の誤差 $\boldsymbol{e}_k^- = \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^-$

事後推定値の誤差 $\boldsymbol{e}_k = \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k$

事前推定値の誤差の共分散行列 $\boldsymbol{P}_k^- = E\left[\boldsymbol{e}_k^- \boldsymbol{e}_k^{-T}\right]$

事後推定値の誤差の共分散行列 $\boldsymbol{P}_k = E\left[\boldsymbol{e}_k \boldsymbol{e}_k^T\right]$

カルマンゲイン***K***はP$_k$のエネルギー(主対角要素の和)を最小化するように逐次決定する。

観測による事前推定値の修正処理 $\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k \underbrace{(\boldsymbol{z}_k - \boldsymbol{H}\hat{\boldsymbol{x}}_k^-)}_{\text{観測イノベーション}}$

カルマンゲイン

カルマンフィルタの基礎

事後推定値の誤差の共分散行列 $P_k = E\left[e_k e_k^T\right]$

カルマンゲイン***K***は
P$_k$のエネルギー
(主対角要素の和)を
最小化するように
逐次決定する。

観測による事前推定値の修正処理 $\hat{x}_k = \hat{x}_k^- + K_k\underbrace{(z_k - H\hat{x}_k^-)}_{\text{観測イノベーション}}$

カルマンゲイン

最適なカルマンゲインは次の式で与えられる。

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}$$

カルマンフィルタの基礎

観測による事前推定値の修正処理 $\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k \underbrace{(\boldsymbol{z}_k - \boldsymbol{H}\hat{\boldsymbol{x}}_k^-)}_{\text{観測イノベーション}}$

カルマンゲイン

最適なカルマンゲイン

$$\boxed{\boldsymbol{K}_k = \boldsymbol{P}_k^- \boldsymbol{H}^T (\boldsymbol{H}\boldsymbol{P}_k^- \boldsymbol{H}^T + \boldsymbol{R})^{-1}}$$

$\boldsymbol{R} \to \boldsymbol{0}$ すなわち、もし、観測雑音が0になると、

$$\boldsymbol{K}_k = \boldsymbol{P}_k^- \boldsymbol{H}^T (\boldsymbol{H}\boldsymbol{P}_k^- \boldsymbol{H}^T)^{-1} = \boldsymbol{P}_k^- \boldsymbol{H}^T (\boldsymbol{P}_k^- \boldsymbol{H}^T)^{-1} \boldsymbol{H}^{-1} = \boldsymbol{H}^{-1}$$

となり、観測値を最大限信頼し、事前推定値を軽視する。

$\boldsymbol{P}_k^- \to \boldsymbol{0}$ すなわち、もし、事前推定値の誤差が0になると、

$$\boldsymbol{K}_k = \boldsymbol{P}_k^- \boldsymbol{H}^T (\boldsymbol{H}\boldsymbol{P}_k^- \boldsymbol{H}^T + \boldsymbol{R})^{-1} = \boldsymbol{0}$$

となり、観測イノベーションに影響されなくなる。

カルマンフィルタの基礎

離散的カルマンフィルタのアルゴリズム

カルマンフィルタはフィードバック制御系の一種。

初期値 $\hat{x}_0, P_0$

時間更新

状態を進める $\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$

事後推定誤差の
共分散行列の算出 $P_k^- = AP_{k-1}A^T + Q$ ← この導出は次ページ

観測更新

カルマンゲインの算出 $K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$

観測により事後推定 $\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$

事後推定誤差の
共分散行列の算出 $P_k = (I - K_k H)P_k^-$ ← この導出は煩雑

カルマンフィルタの基礎

$\boldsymbol{P}_k^- = \boldsymbol{A}\boldsymbol{P}_{k-1}\boldsymbol{A}^T + \boldsymbol{Q}$ の導出

$$
\begin{aligned}
\boldsymbol{P}_k^- = \boldsymbol{E}\{(\hat{\boldsymbol{x}}_k^- - \boldsymbol{x}_k)(\hat{\boldsymbol{x}}_k^- - \boldsymbol{x}_k)^T\} \\
&= \boldsymbol{E}\{(\boldsymbol{A}\hat{\boldsymbol{x}}_{k-1} - \boldsymbol{A}\boldsymbol{x}_{k-1} - \boldsymbol{w}_{k-1})(\boldsymbol{A}\hat{\boldsymbol{x}}_{k-1} - \boldsymbol{A}\boldsymbol{x}_{k-1} - \boldsymbol{w}_{k-1})^T\} \\
&= \boldsymbol{E}\left\{\{\boldsymbol{A}(\hat{\boldsymbol{x}}_{k-1} - \boldsymbol{x}_{k-1}) - \boldsymbol{w}_{k-1}\}\{\boldsymbol{A}(\hat{\boldsymbol{x}}_{k-1} - \boldsymbol{x}_{k-1}) - \boldsymbol{w}_{k-1}\}^T\right\} \\
&= \boldsymbol{A}\boldsymbol{E}\{(\hat{\boldsymbol{x}}_{k-1} - \boldsymbol{x}_{k-1})(\hat{\boldsymbol{x}}_{k-1} - \boldsymbol{x}_{k-1})^T\}\boldsymbol{A}^T + \boldsymbol{E}(\boldsymbol{w}_{k-1}{\boldsymbol{w}_{k-1}}^T) \\
&= \boldsymbol{A}\boldsymbol{P}_{k-1}\boldsymbol{A}^T + \boldsymbol{Q}
\end{aligned}
$$

$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H})\boldsymbol{P}_k^-$ の導出は省略

カルマンフィルタの基礎

簡単な例(スカラー）　雑音に埋もれた一定電圧を計測する例。

状態方程式

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \qquad A = 1, B = 0$$

$$= x_{k-1} + w_{k-1}$$

観測方程式

$$z_k = Hx_k + v_k \qquad H = 1$$

$$= x_k + v_k$$

カルマンフィルタの基礎

簡単な例(スカラーの例)

初期値 $x_0 = 0, P_0 = 0$

時間更新

状態を進める $\hat{x}_k^- = \hat{x}_{k-1}$

事後推定誤差の
共分散行列(分散)の算出 $P_k^- = P_{k-1} + Q$

観測更新

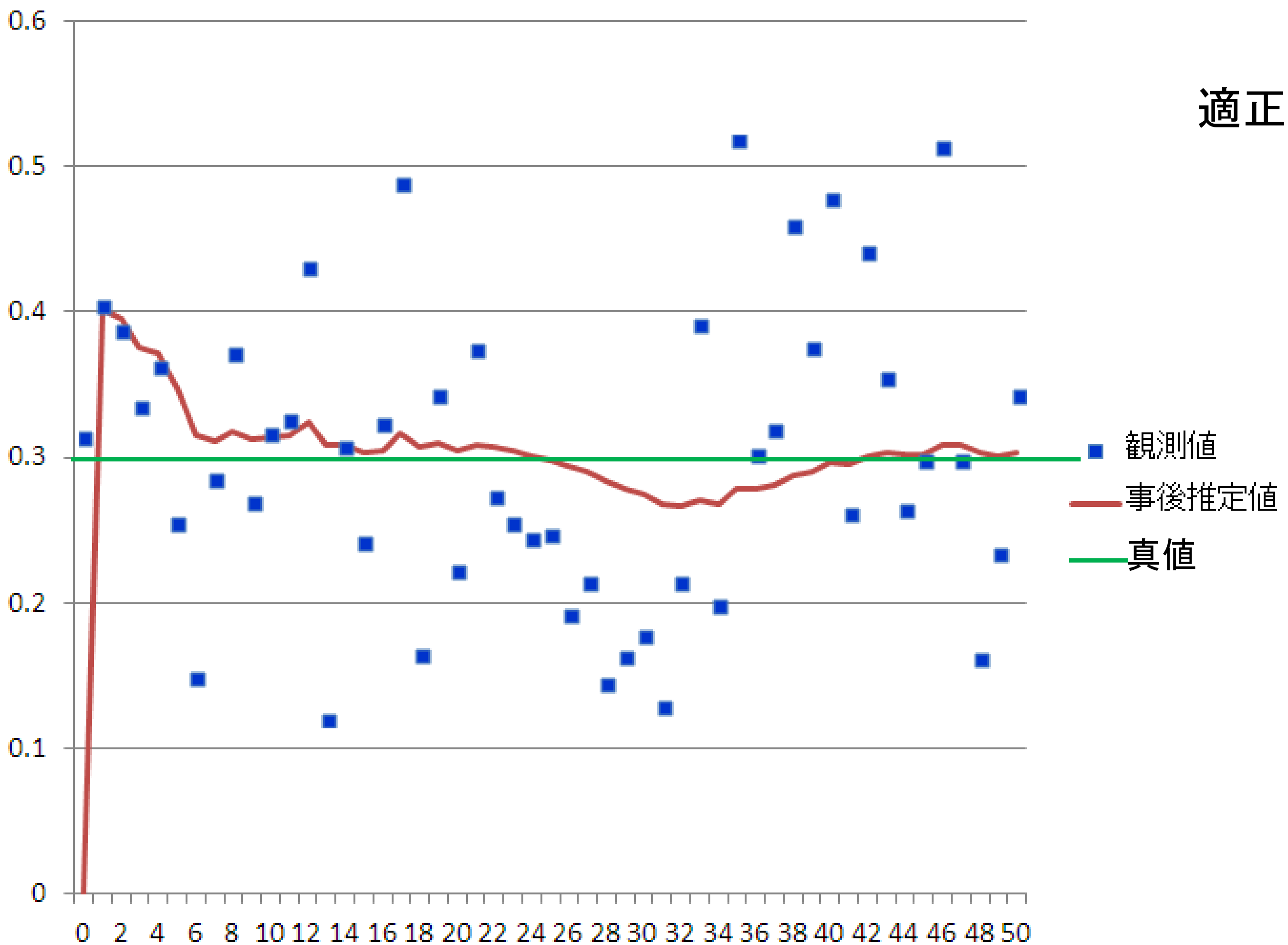
カルマンゲインの算出 $K_k = P_k^-(P_k^- + R)^{-1} = \dfrac{P_k^-}{P_k^- + R}$

観測により事後推定 $\hat{x}_k = \hat{x}_k^- + K_k(z_k - \hat{x}_k^-)$

事後推定誤差の
共分散行列(分散)の算出 $P_k = (1 - K_k)P_k^-$

カルマンフィルタの基礎

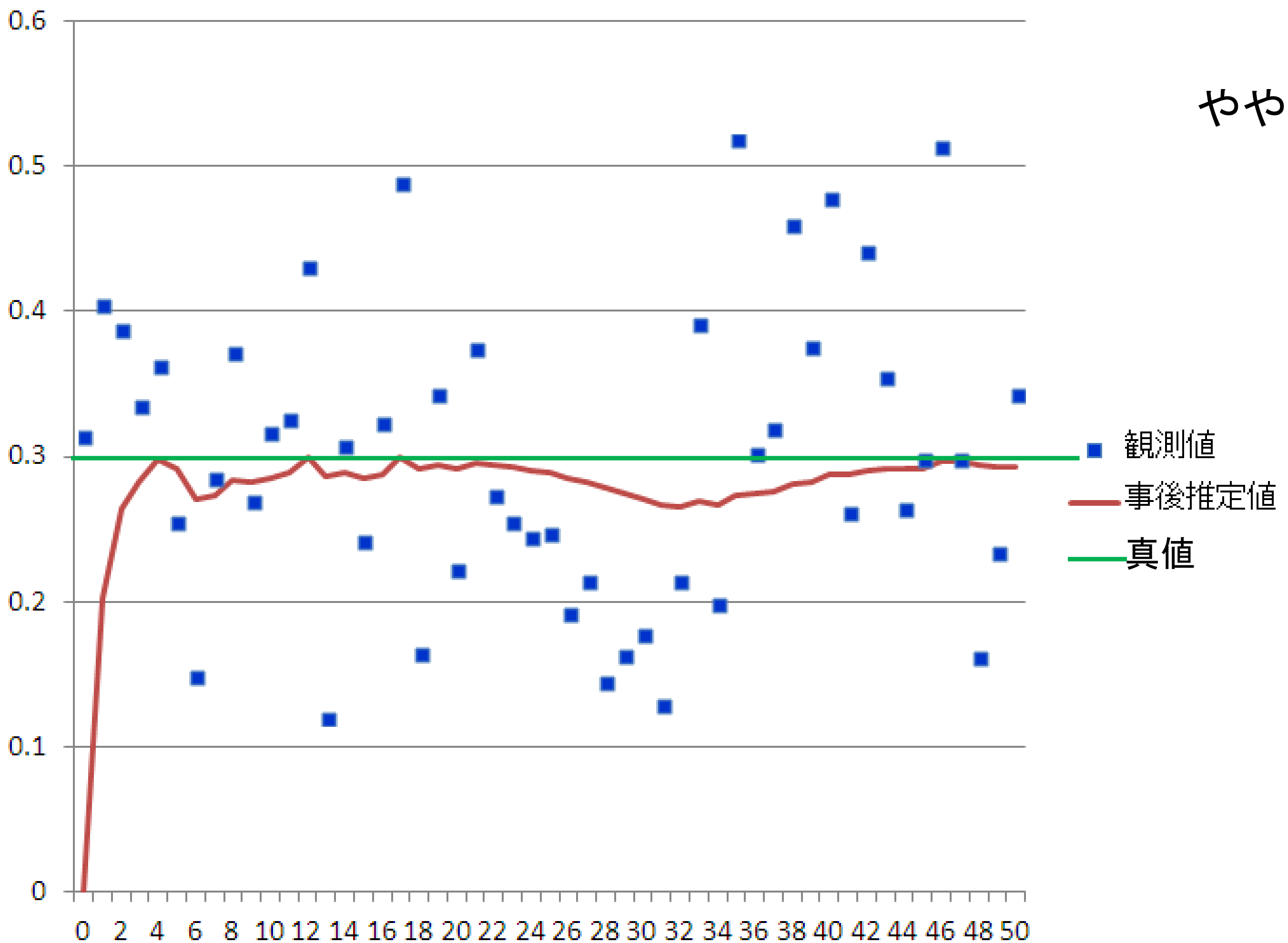
R:観測雑音の分散　0.01とした場合　　実際も0.01



適正な反応

カルマンフィルタの基礎

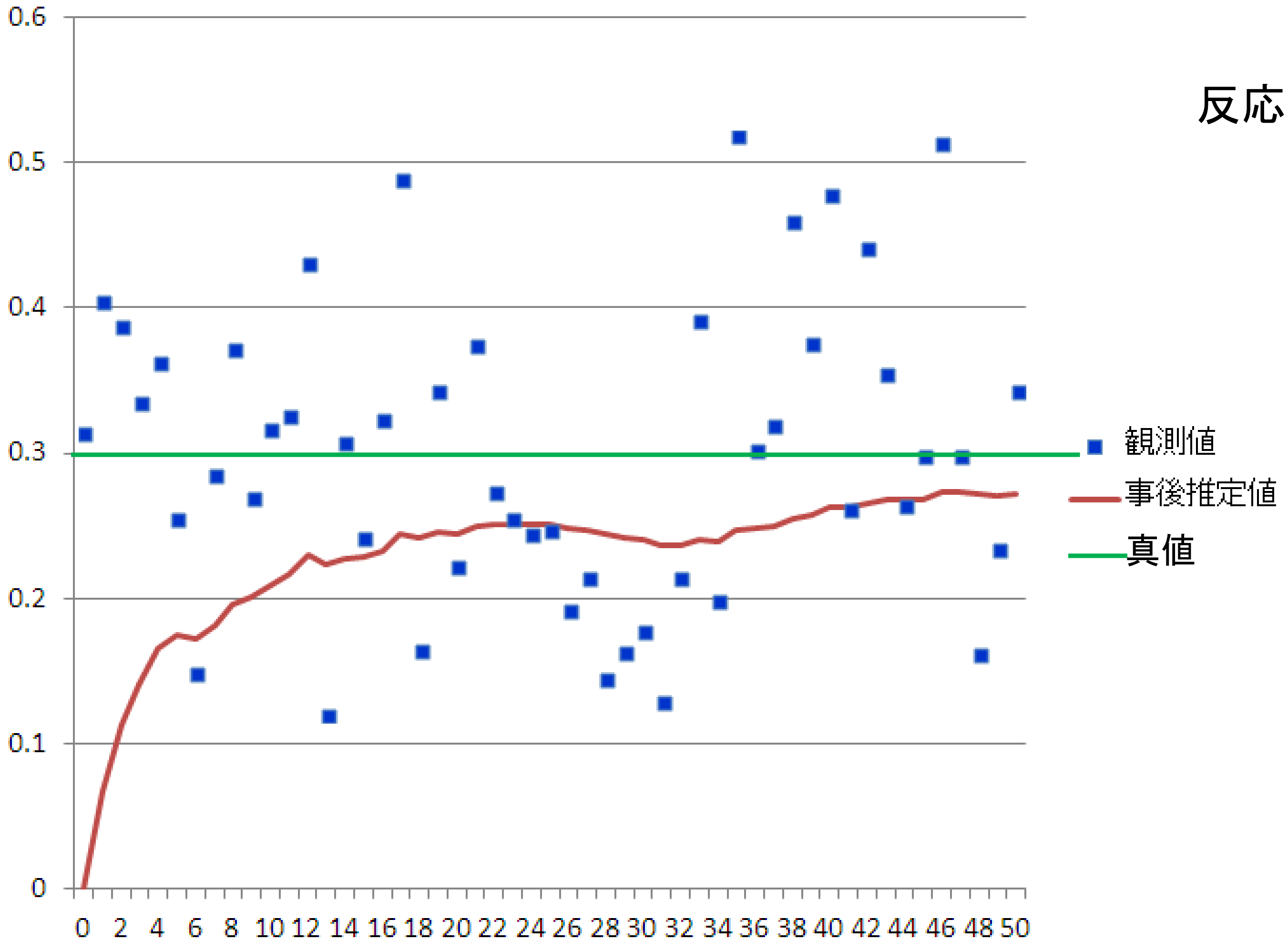
R:観測雑音の分散　1とした場合　　実際は0.01



やや反応が遅い

カルマンフィルタの基礎

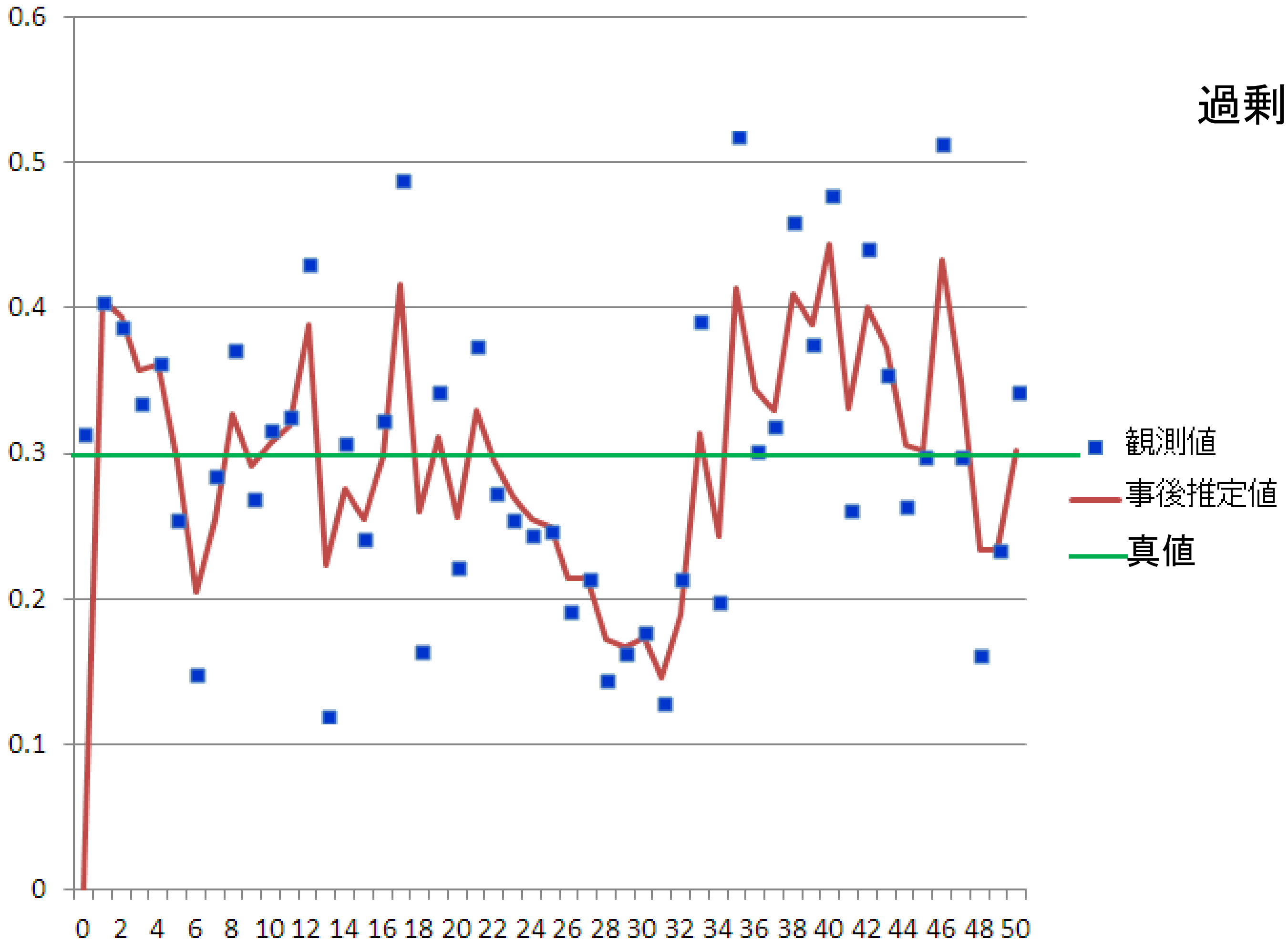
R:観測雑音の分散　5とした場合　　実際は0.01



反応が遅い

カルマンフィルタの基礎

R:観測雑音の分散　0.00001とした場合　　実際は0.01



過剰に反応

カルマンフィルタの基礎

事後推定値の誤差の分散 P_k の推移の例

R:観測雑音の分散 0.01とした場合 実際も0.01