life.augmented

DT0064
設計ヒント

MEMS センサのノイズの解析と識別
アラン分散、時間分散、アダマール分散、重複分散、修正分散、全分散

Andrea Vitali

主要部品	
LSM6DS3H	iNEMO 慣性モジュール：3 軸加速度センサおよび 3 軸ジャイロセンサ
LSM6DSM/LSM6DSL	iNEMO 慣性モジュール：3 軸加速度センサおよび 3 軸ジャイロセンサ
STEVAL-STLKT01V1	SensorTile 開発キット

目的とベネフィット

この設計ヒントでは、MEMS センサのノイズの解析および識別方法を説明します。アラン分散とアダマール分散について、それぞれのバリエーション（重複分散、修正分散、全分散）と併せて説明します。また、理論的分散 1（Theo1）についても取り上げます。

ベネフィット：

- アラン分散やその他の分散によって MEMS センサの特性を把握する方法を理解します。

信号とノイズ

基本前提は、対象信号が測定中に一定であることです。センサ出力は対象信号とノイズの和です。概して、ノイズは長期的に平均した場合 0 でなければなりません。

測定中に多数のサンプルを取得します。ノイズの解析と識別は、どれだけのサンプルを平均するとセンサ出力の分散を最小化できるのかを決定するのに役立ちます。

標準分散の問題は、データ·ランが長くなると正常に機能しなくなることです。この問題を解決するために開発されたのがアラン分散です。アラン分散 σ^2 は、連続する「サンプル」間の 2 乗差の平均値として計算されます（2 サンプル分散）。「サンプル数」は、時間間隔 τ = m*Ts における m 個のサンプルの平均を求めることよって計算されます。ここで、Ts=1/Fs はサンプリング間隔、Fs はサンプリング周波数です。

アラン分散及びその他の分散

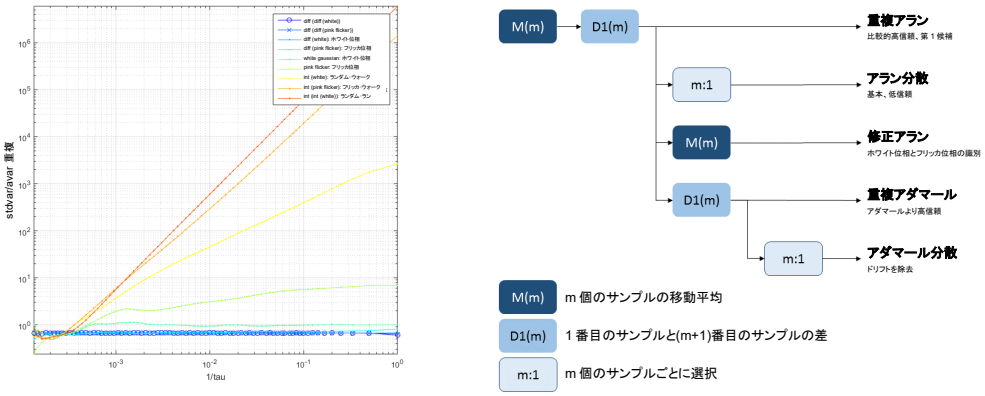
アラン偏差 $\sigma(\tau)$は、アラン分散 $\sigma^2(\tau)$の平方根です。両対数プロットの傾きは、ノイズの種類によって決まります。この傾きこそがノイズの種類の識別を可能にします。下記を参照してください。

アラン偏差（非重複アラン分散 AVAR、重複アラン分散 OAVAR、修正アラン分散 MAVAR）とアダマール分散（非重複アダマール分散 HVAR、重複アダマール分散 OHVAR）は、一連のデジタル・フィルタを使用して計算できます。

- M(m)　m 個のサンプルの移動平均
- D1(m)　n 番目のサンプルと（n+m+1）番目のサンプルの階差
- m:1　m 個のサンプルから 1 個を選択することによるダウンサンプリング

過渡信号が出力されてはなりません。分散は、出力サンプルの 2 乗平均によって計算されます。偏差は分散の平方根です。偏差の信頼性は、偏差自体を平均出力サンプル数の平方根で割ることによって推定できます。

図1. 左：var/AVAR の比、右：処理チェーンのブロック図



重要な注意点：

- 重複アラン分散（OAVAR）と重複アダマール分散（OHVAR）は、信頼性が比較的高く、m = データ・ラン長の 10%まで使用できます。
- 時間分散（TVAR）は、修正アラン分散（MAVAR）をスケーリングしたもので、スケーリング係数は $\tau^2/3$ です。これはホワイト位相ノイズ（つまり、ホワイト周波数ノイズとして知られるホワイト・ガウス・ノイズの導関数）に最適です。
- アラン全分散、修正全分散、アダマール全分散の信頼性は、m = データ・ラン長の 30 ~ 50% まで比較的良好です。全分散は、両側の反射によってデータ・ラン長を延長することで計算されます。反射するサンプル数は、アラン分散が 2m、修正アラン分散およびアダマール分散が 3m です。
- 理論的分散 1（Theo1）の信頼性は、m = データ・ラン長の 75%まで比較的良好です。その他のすべての分散では、ストライド時間は「サンプル数」の計算に使用する平均化時間 var(τ =m*Ts)と同じです。Theo1 の場合、ストライド時間は m*Ts から平均化時間を

差し引いた値です。平均化時間は m/2 ~ 1、ストライド時間は m/2 ~ m-1、平均ストライド時間は 0.75*m*Ts: Theo1(τ =0.75*m*Ts)です。Theo1 は偏りを含む分散で、Theo1BR(偏り除去)は不偏分散です。Theo1H は、m<10%ではアラン分散、m>10%では Theo1BR です。

分散を計算するための MatLab コード

以下は、前述の分散を一連のデジタル･フィルタとして計算するためのリファレンス MatLab コードです。

```
  % デジタル･フィルタとしてのアラン分散
n % 平均係数、n 個のサンプルを平均化
Fs % サンプリング周波数
Ts=1/Ts;
tau=n*Ts;

%---- STDVAR および OSTDVAR
Mn=ones(1,n)/n; % 平均化フィルタ
dataM=filter(Mn,1,data); dataM=dataM(n:end); % 過渡信号をフィルタで除去
ostdvar()=var(dataM);
stdvar()=var(dataM(1:n:end));

%---- AVAR とおよび AVAR
D1n=zeros(1,n+1); D1n(1)=+1; D1n(end)=-1; % 微分フィルタ
dataMD=filter(D1n,1,dataM); dataMD=dataMD(n+1:end); % 過渡信号をフィルタで除去
L=length(dataMD);           oavar()=0.5*sum(dataMD.^2)/(L);
L=length(dataMD(1:n:end));  avar()=0.5*sum(dataMD(1:n:end).^2)/(L);

%---- MAVAR
dataMDM=filter(Mn,1,dataMD);
dataMDM=dataMDM(n:end);
L=length(dataMDM);          mavar()=0.5*sum(dataMDM.^2)/(L);

%---- HVAR および OHVAR
dataMDD=filter(D1n,1,dataMD); dataMDD=dataMDD(n+1:end); % 過渡信号をフィルタで除去
L=length(dataMDD);          ohvar()=0.5*sum(dataMDD.^2)/(L);
L=length(dataMDD(1:n:end)); hvar()=0.5*sum(dataMDD(1:n:end).^2)/(L);
```

上記コードで最も遅い行は移動平均フィルタです。新しい項が追加され、古い項が破棄される場合、累積和を維持すると、実行が大幅に高速化されます。

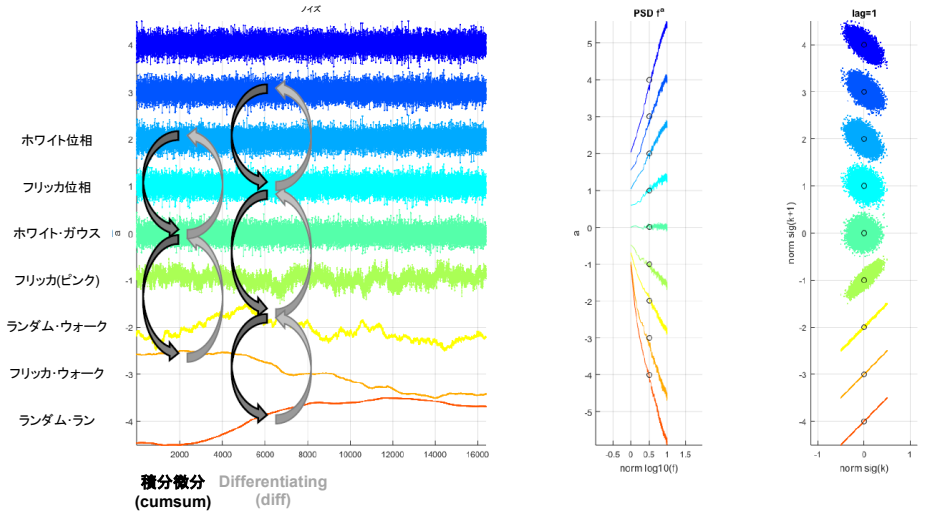
```
% 最適化された OAVAR
n2=n*n;
acc(1)=sum(data(1:n)); % 初期累積和
for i=1:N-n, acc(i+1)=acc(i)-data(i)+data(i+n); end; % 累積和
oavar()=0.5*sum( (acc(1:N-2*n+1) - acc(1+n:N-n+1)).^2 )/(N-2*n+1)/n2;

% AVAR
diffL=fix((N-2*n+1 -1)/n)+1;
avar()=0.5*sum( (acc(1:n:N-2*n+1)-acc(1+n:n:N-n+1)).^2 )/(diffL)/n2;

% STDVAR および OSTDVAR
stdvar()=var(acc(1:n:N-n+1))/n2;
ostdvar()=var(acc(1:N-n+1))/n2;

% 最適化された MAVAR
n4=n2*n2;
acc2(1)=sum(acc(1:n)); % init running sum
for i=1:N-2*n+1, acc2(i+1)=acc2(i)-acc(i)+acc(i+n); end; % 累積和
mavar()=0.5*sum( (acc2(1:N-3*n+2) - acc2(1+n:N-2*n+2)).^2 )/(N-3*n+2)/n4;

% OHVAR
ohvar()=0.5*sum( ( (acc(1  :N-3*n+1) - acc(1+  n:N-2*n+1)) - ...
                   (acc(1+n:N-2*n+1) - acc(1+2*n:N-  n+1)) ).^2 )/(N-3*n+1)/n2;

% HVAR
diffL=fix((N-3*n+1 -1)/n)+1;
hvar()=0.5*sum( ( (acc(1  :n:N-3*n+1) - acc(1+  n:n:N-2*n+1)) - ...
                  (acc(1+n:n:N-2*n+1) - acc(1+2*n:n:N-  n+1)) ).^2 )/(diffL)/n2;
```

さらにコードを最適化できます。一例として、最適化された C コードでは、ワンパスで分散を計算できる可能性があります(通常は第 1 パスで平均、第 2 パスで実際の分散の形で、2 パスが必要です)。

ノイズモデル

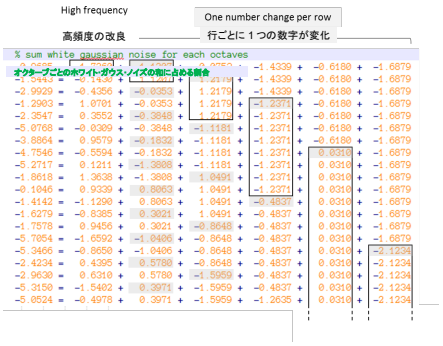
ホワイト･ガウス（WH）ノイズは、最も基本的なタイプのノイズです。積分（cumsum）すると**ランダム･ウォーク**（RW）が得られます。さらに積分すると**ランダム･ラン**（RR）が得られます。ホワイト周波数ノイズ（WHFM）の導関数（diff）を取ると、ホワイト位相ノイズ（WHPM）になります。

図2. ノイズ･プロット、パワー･スペクトル密度（PSD）、ラグ･プロット（x_k 対 x_{k+1}）



フリッカ 1/f ノイズ（**ピンク**･ノイズ）も基本的なタイプのノイズです。これは、dx(t)/dt = n1*x(t) + n2*w(t)を解くことで発生するノイズに相当します（ここで、w(t)はホワイトノイズ）。複数のホワイト･ノイズ発生源を異なるオクターブで加算することで発生させることができます。積分（cumsum）すると**フリッカ･ウォーク**（FW）が得られます。フリッカ周波数（FLFM）の導関数（diff）を取ると、フリッカ位相（FLPM）になります。

```
% 1/f ノイズ（フリッカ、ピンク）発生器
  N % ノイズ･ベクトルの長さ
  sd; % ノイズ発生器の標準偏差
  n=zeros(1,N); % 初期ノイズ･ベクトル
  imax=floor(log2(N)); ngen=randn(1,imax+1)*sd; % 初期ノイズ発生
  ngensum=sum(ngen); % 初期累積和
  for i=1:N,
    % カウンタで末尾のゼロの数を検出
    tlz=0; t=i; while mod(t,2)==0, tlz=tlz+1; t=t/2; end;

    % 累積和を更新
    ngensum=ngensum-ngen(tlz+1); % 古い値を除去
    ngen(tlz+1)=randn(1)*sd; % ノイズ発生器を更新
    ngensum=ngensum+ngen(tlz+1); % 新しい値を追加
    n(i)=ngensum+(rand(1)*sd); % 高い周波数の和 + ノイズ
  end;
```

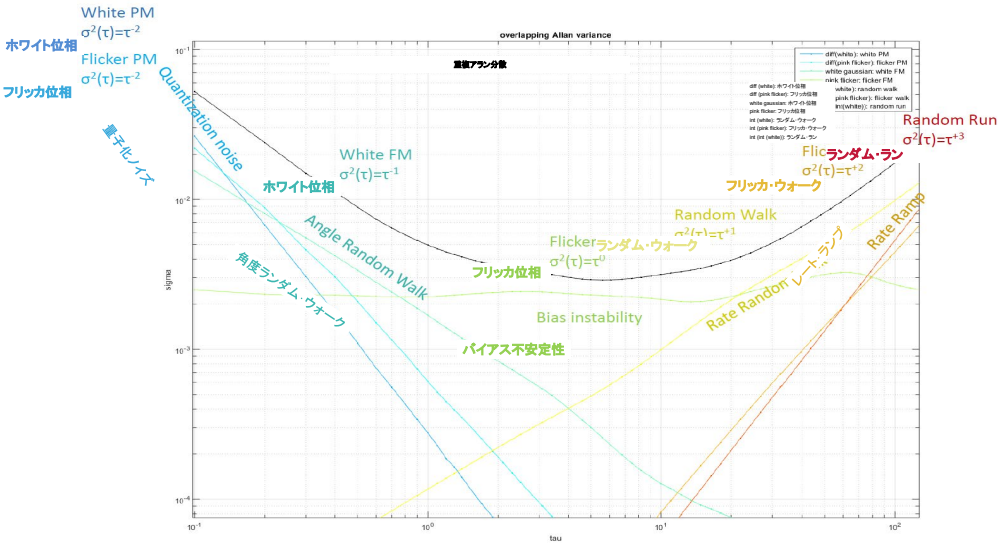


ノイズの識別

ノイズを識別するには、$\sigma\ \tau$ プロット（偏差、分散の平方根）と$\sigma^2\ \tau$ プロット（分散）の傾きに注目する必要があります。ホワイト位相ノイズとフリッカ位相ノイズを識別するには、修正アランまたは時間偏差/分散が必要です。

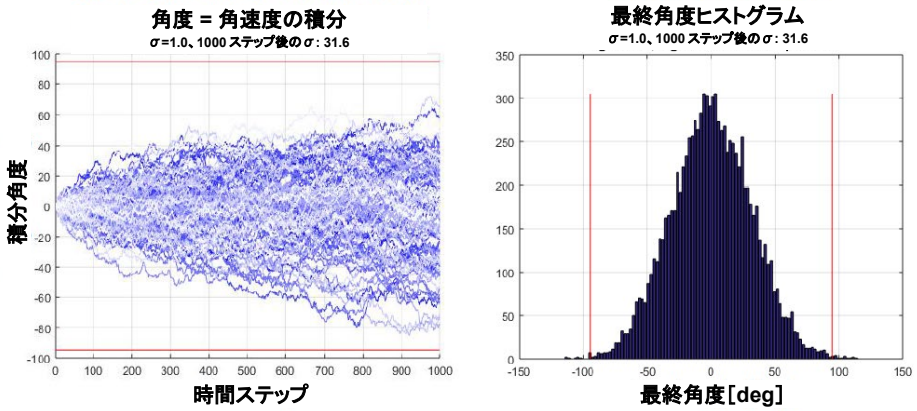
ノイズの種類	Matlab コード	スペクトル f^x	ADEV $\sigma(\tau)$	AVAR $\sigma^2(\tau)$	MADEV mod.$\sigma(\tau)$	MAVAR mod.$\sigma^2(\tau)$	TDEV $\sigma_T(\tau)$	TVAR $\sigma^2_T(\tau)$
ホワイト位相	Diff(WhiteFM)	f^{+2}	τ^{-1}	τ^{-2}	$\tau^{-3/2}$	τ^{-3}	$\tau^{-1/2}$	τ^{-1}
フリッカ位相	Diff(FlickerFM)	f^{+1}	τ^{-1}	τ^{-2}	τ^{-1}	τ^{-2}	τ^{0}	τ^{0}
ホワイト周波数	Randn(1)	f^{0}	$\tau^{-1/2}$	τ^{-1}	$\tau^{-1/2}$	τ^{-1}	$\tau^{+1/2}$	τ^{+1}
フリッカ周波数	上記参照	f^{-1}	τ^{0}	τ^{0}	τ^{0}	τ^{0}	τ^{+1}	τ^{+2}
ランダム·ウォーク	Cumsum(WhiteFM)	f^{-2}	$\tau^{+1/2}$	τ^{+1}	$\tau^{+1/2}$	τ^{+1}	$\tau^{+3/2}$	τ^{+3}
フリッカ·ウォーク	Cumsum(FlickerFM)	f^{-3}	τ^{+1}	τ^{+2}	τ^{+1}	τ^{+2}		
ランダム·ラン	Cumsum(RandomWalk)	f^{-4}	$\tau^{+3/2}$	τ^{+3}	$\tau^{+3/2}$	τ^{+3}		

図3. 固有の傾きを持つ各種ノイズアラン分散プロット、$\sigma^2(\tau)$対 τ



ジャイロスコープのノイズ識別

ジャイロスコープ出力は角速度で、ホワイト·ノイズの影響を受けます。角度位置は積分によって得られます。ジャイロスコープが回転していないときの出力は本来ゼロですが、そうはならず、ある標準偏差を持ち平均ゼロのホワイトノイズとなります。積分すると、非ゼロの最終角度が得られます。これが角度ランダム·ウォーク(ARW)です。最終角度誤差 RMS = ARW*sqrt(time)。例：ARW 1deg/sqrt(s)*sqrt(1000s) = 31.6deg RMS



角度ランダム·ウォークはデータシートに記載されており、**ARW (deg/sqrt(s)) = ノイズ密度 deg/s/sqrt(Hz)**です。ARW はアランプロットで調べることもでき、τ =1s での傾きが $\tau^{-1/2}$ のアラン偏差区間、または τ =1s での傾きが τ^{-1} でのアラン分散区間の交点です。

サポート資料

関連設計サポート資料
STEVAL-STLKT01V1、SensorTile 開発キット
技術資料
LSM6DS3H,iNEMO inertial module: always-on 3D accelerometer and 3D gyroscope
LSM6DSM, iNEMO inertial module: always-on 3D accelerometer and 3D gyroscope
LSM6DSL, iNEMO inertial module: always-on 3D accelerometer and 3D gyroscope
AN4844, Application Note, LSM6DS3H: always on 3D accelerometer and 3D gyroscope

改版履歴

日付	版	変更内容
2016 年 7 月 13 日	1	初版発行

重要なお知らせ（よくお読み下さい）

STMicroelectronics NV およびその子会社（以下、ST）は、ST 製品及び本書の内容をいつでも予告なく変更、修正、改善、改定及び改良する権利を留保します。購入される方は、発注前に ST 製品に関する最新の関連情報を必ず入手してください。ST 製品は、注文請書発行時点で有効な ST の販売条件に従って販売されます。

ST 製品の選択並びに使用については購入される方が全ての責任を負うものとします。購入される方の製品上の操作や設計に関して ST は一切の責任を負いません。

明示又は黙示を問わず、ST は本書においていかなる知的財産権の実施権も許諾致しません。

本書で説明されている情報とは異なる条件で ST 製品が再販された場合、その製品について ST が与えたいかなる保証も無効となります。

ST および ST ロゴは STMicroelectronics の商標です。ST の登録商標については ST ウェブサイトをご覧ください。
www.st.com/trademarks

その他の製品またはサービスの名称は、それぞれの所有者に帰属します。

本書の情報は本書の以前のバージョンで提供された全ての情報に優先し、これに代わるものです。