

加速度センサーによる傾きの検出

著者： Christopher J. Fisher

はじめに

システムの傾斜または傾きを判定するためによく用いられる方法の1つは、ジャイロ스코ープの出力を積分することです。この方法は単純明快ですが、積分時間が長くなるとヌル・バイアス安定性による積分誤差により、たとえデバイスが静止状態であっても傾きが生じているような出力が得られることがあります。

アプリケーションによっては、システムに加わる主な加速度が重力である場合に、加速度センサーを用いて傾きまたは傾斜角度を測定できます。このようなアプリケーションとしては、ゲーム機器、カメラの水平検出、産業用/医用アプリケーションでのデバイスの傾きの検出などがあります。

加速度センサーによる傾きの検出では、印加される加速度は重力のみであるということが前提になります。実際には、信号処理によって出力信号の高周波数成分を除去できるため、若干のAC加速度入力がある場合でも傾きを計測することが可能です。

傾きの検出では、重力ベクトルと加速度センサーの軸上のその投影を利用し傾斜角を求めます。重力はDC加速度であるため、ほかのDC加速度を増加させる力が存在すると、誤差の原因となります。たとえばDC加速度の原因としては、車両が一定の割合で加速する、加速度センサー上に向心加速度を発生させる回転装置などがあります。さらに、加速度センサーを回転させると、対象となる軸上の重力の投影が変化するため、AC加速度が生じます。傾きを計算する前に加速度信号をフィルタリングすることが一般的です。

このアプリケーション・ノートでは、加速度センサーの出力を傾き角度に変換するための基本原理を解説します。この解説では、1軸、2軸、3軸計測での理想的な傾き角の計算方法も示します。さらに、オフセットや感度のミスマッチに起因する誤差を減らすためのキャリブレーションについても基本的な事柄を説明します。

目次

はじめに.....	1	オフセットと感度のミスマッチ誤差のキャリブレーション.....	7
傾き／傾斜の計算.....	3	オフセット誤差の影響.....	7
1軸の傾斜の計算.....	3	感度ミスマッチ誤差の影響.....	7
2軸の傾斜の計算.....	4	基本的なキャリブレーション技術.....	8
3軸の傾斜の計算.....	6		

傾き／傾斜の計算

1軸の傾斜の計算

0度を中心として限られた角度の傾きを検出し、分解能がやや粗くても構わないアプリケーションでは、1軸デバイス（または多軸デバイスの1軸）を使用できます。

たとえば、図1では、1軸（この例ではx軸）が重力ベクトルの方向に沿って回転します。この方式では1軸だけを使用し、重力ベクトルを必要とするため、x軸が必ず重力ベクトルの平面内に来るようにデバイスを向けるようにしておかなければ、計算された傾き角度が正確になりません。ほかの軸を中心とする回転があると、x軸上の加速度の大きさが減少し、傾き角度の計算に誤差が生じます。

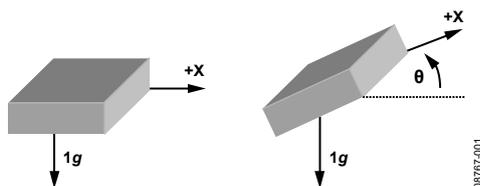


図1. 1軸による傾斜の検出

基本的な三角関数を用いて計算すれば、x軸の出力は重力ベクトルのx軸上の投影によって、加速度センサーのx軸と水平線がなす角度のサイン（正弦）に等しい出力加速度となることが解ります。水平線は、一般的に重力ベクトルと直交する平面と見なされます。重力が理想値である1gであった場合、出力加速度は次式で表すことができます。

$$A_{x,OUT} [g] = 1 g \times \sin(\theta) \quad (1)$$

1軸ソリューションを使用する場合、傾きの計算の感度（つまり、入力の変化に対する出力の変化）は、水平線とx軸がなす角度が大きくなるにつれて減少し、角度が±90°に近づくと感度は0に近づきます。これを図2に示します。図2は、傾き角度に対する出力加速度（単位：g）を表しています。±90°の近辺では、傾き角が大きく変化しても出力加速度の変化は小さくなります。

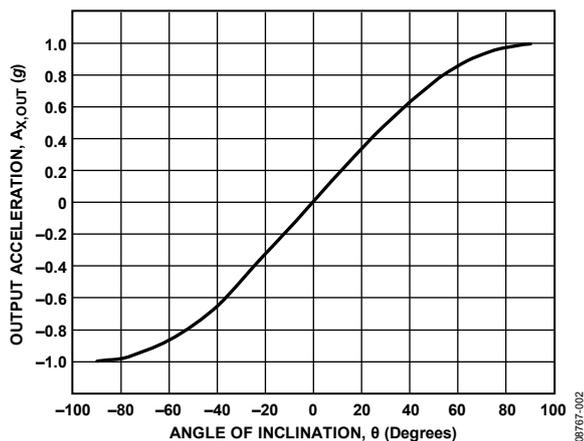


図2. 傾き角度 対 出力加速度
(1軸の傾きの検出の場合)

傾きの計算はデジタルで行われます。A/Dコンバータ（ADC）またはデジタル出力製品から得られる最下位ビットの重みは加速度換算で一定なので、傾き角換算での分解能は加速度センサーの傾きに応じて変化します。0°近くで最高の分解能が得られ、±90°で分解能が最低になります。

図3と図4に、1°と0.25°の傾き角ステップに対する傾き換算での感度を示します。傾き換算での感度は、傾き角ステップ当たりの出力変化（単位：mg）であり、次式のようになります。

$$S [g] = 1 g \times (\sin(N+P) - \sin(N)) \quad (2)$$

ここで、
Nは現在の角度です。
Pはステップ・サイズです。

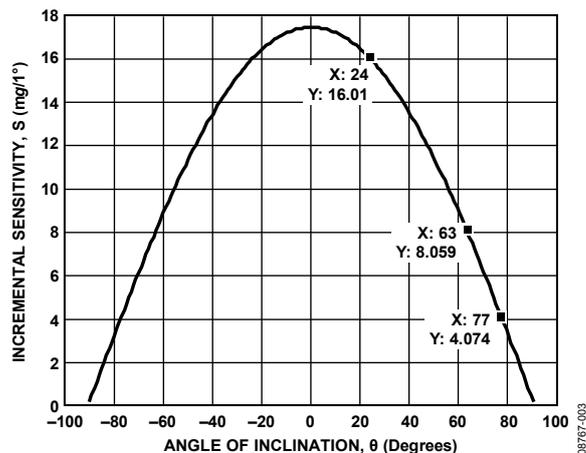


図3. 傾き感度（ステップ：1°）

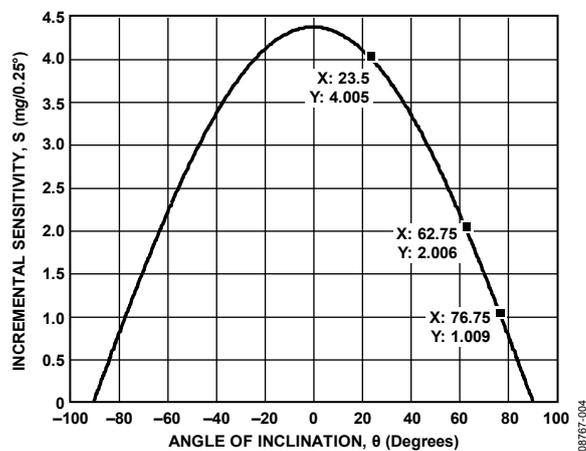


図4. 傾き感度（ステップ：0.25°）

これらの曲線を使用すれば、アプリケーションの全範囲で所望の傾き分解能を満たすために、出力加速度の測定に必要な最小分解能を決定することができます。たとえば、1°の分解能をアプリケーションで必要な場合は、±63°を計測範囲とした場合は少なくとも8 mg/LSBの分解能が必要です。同様に、±63°の範囲に対して0.25°の分解能を確保するには、少なくとも2 mg/LSBの分解能が必要です。なお、感度が固定されているデジタル出力タイプの加速度センサーの場合でも、オーバーサンプリングによってさらに優れた分解能を達成することができる可能性があります。

重力加速度の軸に沿って加速度センサーを回転させた場合の出力は、サイン波の関係に従います。したがって、加速度から角度への変換は、次の逆サイン関数を使用して行われます。

$$\theta = \sin^{-1}\left(\frac{A_{X,OUT} [g]}{1g}\right) \quad (3)$$

ここで、傾き角 (θ) の単位はラジアンです。

水平線を基準として狭い範囲の傾きが要求される場合は、逆サイン関数の代わりに線形近似を使用できます。線形近似は、小さな角度のサインの近似に関係します。

$$\sin(\theta) \cong \theta, \theta \ll 1 \quad (4)$$

ここで、傾き角 (θ) の単位はラジアンです。

傾き角の線形近似には、さらにスケーリング係数 (k) を組み込むことができます。これによって、許容できる誤差が増大した場合に、近似の有効範囲を拡大することができます。

$$\theta \cong k \times \left(\frac{A_{X,OUT} [g]}{1g}\right) \quad (5)$$

ここで、傾き角 (θ) の単位はラジアンです。

角度に変換するには、式 5 の結果に $(180/\pi)$ を掛けます。図 5 に、逆サイン関数を使用した場合と k を 1 とした線形近似の場合の比較を示します。傾き角が大きくなるとともに、線形近似は機能しなくなり、計算した角度と実際の角度とのずれが大きくなります。

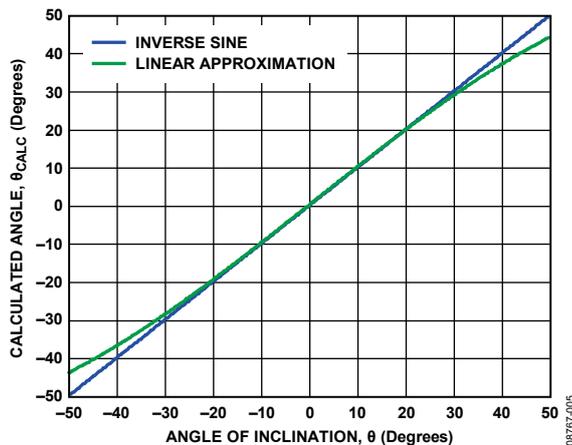


図 5. 逆サイン関数と線形近似の比較：
傾き角の計算

実際の傾き角度を基準に計算した角度が描かれているため、線形近似は両端付近で曲がっているように見えます。その理由は、線形近似は実際の傾き角と出力加速度を線形であると近似しているため、その場合には実際の傾き角と計算された傾き角の関係は線形とならないためです。しかし、逆サイン関数では実際の傾き角と計算される角度が 1 対 1 の結果を出すため、実際の傾き角度を基準にして描くと、計算した角度が直線になります。

線形近似の例として、傾きの検出の所望の分解能が 1° であり、 $\pm 0.5^\circ$ の誤差であれば許容することができるアプリケーションを想定します。図 6 に示すように、 k を 1 として実際の傾き角度と計算した傾き角度との誤差を描いた場合、線形近似の有効範囲はわずか $\pm 20^\circ$ です。所望の誤差要求を達成できる範囲内でスケーリング係数を調整した場合、線形近似の有効範囲は $\pm 30^\circ$ を上回る値まで増加します。

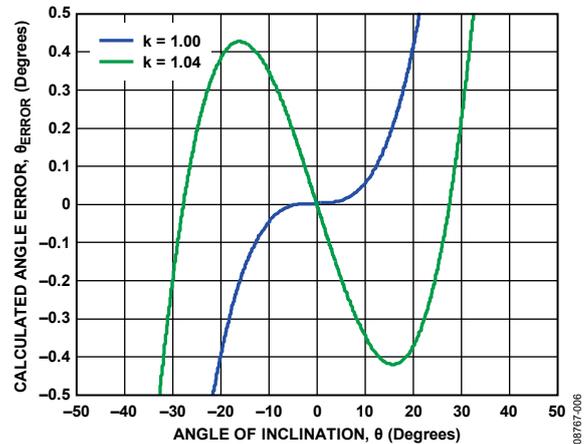


図 6. さまざまなスケーリング係数に対する角度の計算誤差

2 軸の傾斜の計算

1 軸の傾きの検出の 1 つの制約は、図 3 と図 4 に示すように、有効な傾き角の範囲を大きくするために、高分解能の ADC またはデジタル出力が必要になることです。もう 1 つの制約は、1 軸測定では 360° の測定ができないことです。これは、 N° の傾きで生成される加速度が、 $180^\circ - N^\circ$ の傾きで生成される加速度と同一になるためです。アプリケーションによっては、これでも構いません。しかし、高い分解能を必要とするアプリケーションや、 360° の全範囲で傾き角度を識別する必要があるアプリケーションの場合、図 7 に示すように、第 2 の軸か 2 番目のセンサーが必要です。2 番目のセンサーを使用する場合、2 番目のセンサーの検出軸が最初のセンサーの検出軸に直交するような向きになっている必要があります。

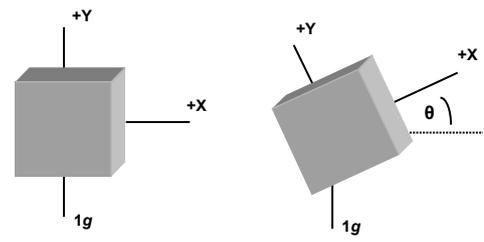


図 7. 2 軸による傾斜の検出

第 2 の軸を組み込んで傾き角度を求める場合、3 つの大きな利点を得られます。次に、これらの利点について説明します。

一定の感度

第2の軸の使用による最初の大きな利点は、軸の直交性によるものです。1軸ソリューションと同様に、x軸によって検出される加速度は、傾きの角度のサインに比例します。y軸の加速度は、直交性のために傾きの角度のコサインに比例します（図8を参照）。1つの軸で加速度が+1gまたは-1gに近づくことなどにより傾き感度が弱くなると、もう1つの軸の傾き感度が増大します。

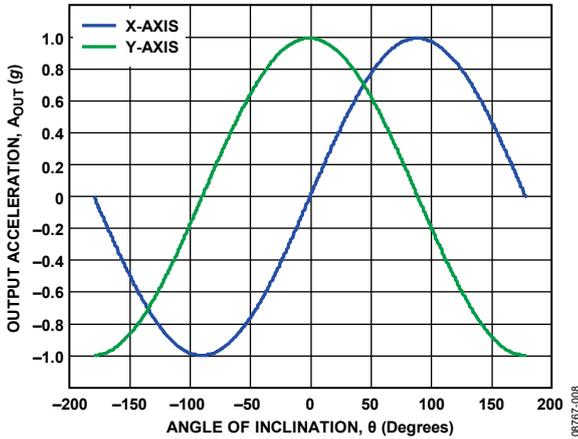


図8. 傾き角度 対 出力加速度
(2軸の傾きの検出)

測定した加速度を傾き角に変換する1つの方法は、1軸ソリューションの場合と同様に、x軸の逆サインとy軸の逆コサインを計算します。しかし、もっと簡単で効率のよい方法は、次に示すように、2つの値の比を使用することです。

$$\frac{A_{X,OUT}}{A_{Y,OUT}} = \frac{1g \times \sin(\theta)}{1g \times \cos(\theta)} = \tan(\theta) \tag{6}$$

$$\theta = \tan^{-1}\left(\frac{A_{X,OUT}}{A_{Y,OUT}}\right) \tag{7}$$

ここで、傾き角 (θ) の単位はラジアンです。

1軸による例とは異なり、2つの軸の比を使用して傾き角度を求める方法では、傾き感度が1つの軸で増加し、もう1つの軸で減少するので、最終的な傾き感度の実効値はほぼ一定になります。つまり、ある角度において所望の傾きステップ・サイズを達成するために十分な分解能をもつ加速度センサーを選択すれば、あらゆる角度に対して十分ということになります。

加速度センサーに必要な最小分解能は、式6から求められます。各軸の出力は傾き角度のサインまたはコサインに依存し、各関数の傾き角度は同じであるため、分解可能な最小角度は分解可能な最小加速度に対応します。

図3と図4に示すように、サイン関数は0°の近辺で最も大きな変化を示し、コサイン関数はこのポイントで変化が最も小さくなります。このため、傾きの変化に起因するx軸の加速度の変化が認識されてから、y軸の加速度の変化が認識されます。したがって、0°近辺のシステムの分解能は、主にx軸の分解能に依存することになります。P°の傾き変化を検出するには、加速度センサーは、およそ次の変化を検出しなければなりません。

$$\Delta A_{OUT} [g] \cong 1g \times \sin(P) \tag{8}$$

図9を使用すれば、所望の傾きステップ・サイズに対して加速度センサーに必要な最小分解能がわかります。なお、加速度センサーの分解能が増大するほど、加速度換算での最下位ビットの重みは小さくなり、小さな加速度の変化を検出できるようになります。したがって、適切な分解能をもつ加速度センサーを選択する場合、意図する傾きステップ・サイズに対して最下位ビットの重みは図9に示す限度より小さい値にします。

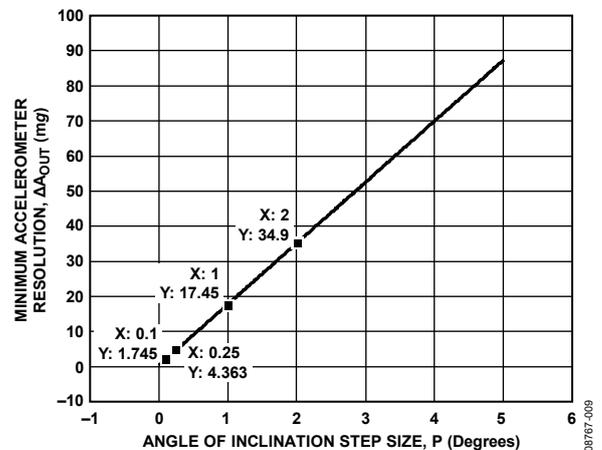


図9. 所望の傾き角度分解能を実現する
加速度センサーの最小分解能

重力平面との位置関係に対する依存性の低減

複数の軸の使用によって得られる2番目の大きな利点は、x軸以外の軸での傾斜が大きな誤差を引き起こすことのある1軸ソリューションとは異なり、第2の軸を使用することで、たとえ第3の軸に傾きが存在しても正確な値を測定できることです。これは、傾き感度の実効値が、対象となる軸における重力の2乗和の平方根 (RSS) 値に比例するためです。

重力がxy平面に完全に含まれる場合、両軸で検出される加速度のRSS値は、理想的には1gになります。xz平面またはyz平面に傾斜が存在する場合は、重力に起因する合計加速度が減少するため、傾き感度の実効値も減少します。これにより、今度は所定の加速度センサーの分解能に対し傾き角度換算での分解能は減少しますが、xz平面もしくはyz平面への傾き角が大きくない場合はある程度正確な測定値が得られます。傾きの計算から得られる角度は、xy平面の回転に対応します。

システムが相当傾斜していてxy平面の重力加速度がごくわずかしかない場合は、傾き角の正確な測定は困難となります。したがって、できる限りxz平面またはyz平面の傾斜を抑えるようなシステム的设计を推奨します。

360°の傾斜の検出

第2の軸の使用による3番目の大きな利点は、各象限を識別できることと、360°の全範囲で角度の測定ができることです。図10に示すように、各象限でx軸とy軸の加速度の符号の組み合わせが異なります。

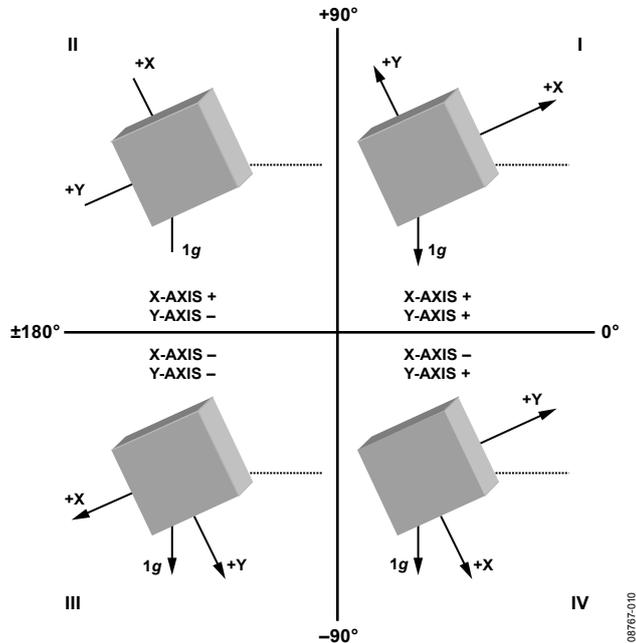


図 10. 傾き角度と加速度の符号による象限の検出

オペランド ($A_{X,OUT}/A_{Y,OUT}$) が正の値の場合、逆タンジェント関数が象限 I の値を示します。オペランドが負の値の場合、逆タンジェント関数は象限 IV の値を示します。象限 II のオペランドは負の値であるため、角度がその象限にある場合は計算結果に 180° を足す必要があります。象限 III のオペランドは正の値であるため、角度がその象限にある場合は計算結果から 180° を引かなければなりません。計算した角度の正しい象限を判定するには、各軸で測定した加速度の符号を調べます。

3軸の傾斜の計算

3番目の軸を導入することで、センサーの向きを完全な球体で判定することができます。以下のような直交座標 (x, y, z) から球面座標 (ρ, θ, ϕ) への古典的な変換方法を使用することで xy 平面での傾斜角度 (θ) と重力ベクトルを基準とした傾き角度 (ϕ) を各軸で測定した加速度に関連付けることができます。

$$\theta = \tan^{-1} \left(\frac{A_{X,OUT}}{A_{Y,OUT}} \right) \quad (9)$$

$$\phi = \cos^{-1} \left(\frac{A_{Z,OUT}}{\sqrt{A_{X,OUT}^2 + A_{Y,OUT}^2 + A_{Z,OUT}^2}} \right) \quad (10)$$

測定した加速度がすべて重力に起因するものであるとすると、すべての軸の出力値のRSS値は定数となるため、式10のオペランドの分母を定数（理想的には1）に置き換えることができます。これらの角度を図11に示します。ここで、図11cは xy 平面での θ 、図11dは z 軸と重力ベクトルの間の角度である ϕ を示します。

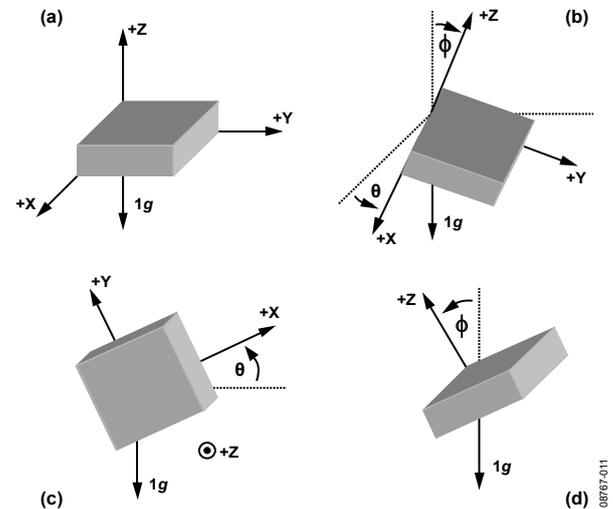


図 11. 球面座標系の角度

3軸方式の式と1軸および2軸方式の式は類似しており、3軸ソリューションの解析は1軸および2軸方式を組み合わせたものと同じです。 θ の測定には2つの直交軸の比を利用することができ、所望の傾き分解能には式8に示す加速度センサーの最小分解能が必要になります。

ϕ の測定は、1軸ソリューションでの傾き角の測定に対応し、必要な加速度センサーの最小分解能は1軸ソリューションと同様の方法で求められます。違いは、逆コサイン関数を使って ϕ を決定するため、 ϕ が 90° のときに最大の傾き感度になり、 0° と 180° において最小の傾き感度になることです。

式2のサインをコサインに置き換えることによって、図3と図4のようなプロットを描くことができます。なお、 θ の範囲は -180° から $+180^\circ$ ですが、 ϕ の範囲は 0° から 180° までになることに注意してください。 ϕ が負の角度の場合、 θ の角度は負の値になりません。

3軸による傾きを検出する別の方法として、基準となる軸と加速度センサーの各軸との傾き角度を個別に決定する方法があります。基準となる軸とは、x軸とy軸は水平線、z軸は重力ベクトルとなります。これを図12に示します。ここで、 θ は水平線と加速度センサーのx軸との角度、 ψ は水平線と加速度センサーのy軸との角度、 ϕ は重力ベクトルとz軸との角度です。x軸とy軸で0g、z軸で1gの最初の位置では、計算した角度はすべて0°になります。

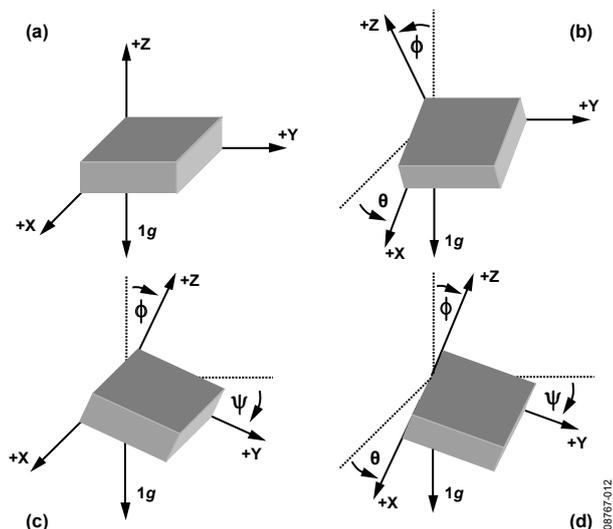


図 12. 各軸の傾きを検出するための角度

基本的な三角法を使用すれば、式 11、式 12、式 13 を使用して傾き角度を計算できることがわかります。

$$\theta = \tan^{-1} \left(\frac{A_{X,OUT}}{\sqrt{A_{Y,OUT}^2 + A_{Z,OUT}^2}} \right) \quad (11)$$

$$\psi = \tan^{-1} \left(\frac{A_{Y,OUT}}{\sqrt{A_{X,OUT}^2 + A_{Z,OUT}^2}} \right) \quad (12)$$

$$\phi = \tan^{-1} \left(\frac{\sqrt{A_{X,OUT}^2 + A_{Y,OUT}^2}}{A_{Z,OUT}} \right) \quad (13)$$

式 13 でのオペランドの見かけの反転は、最初の位置が 1g フィールドであることに起因しています。z 軸の基準を水平にした場合は、オペランドを反転することができます。正の角度は、加速度センサーの対応する検出軸の正極性方向が水平線より上向いていることを意味し、負の角度は、検出軸の正極性方向が水平線より下に向いていることを意味します。

逆タンジェント関数と加速度の比を使用するため、2 軸の例で説明したのと同じ利点があてはまります。つまり、傾き感度の実効値は定数となり、単位球面を中心とするあらゆる位置で角度を正確に測定できます。

オフセットと感度のミスマッチ誤差のキャリブレーション

このアプリケーション・ノートの解析は、理想的な加速度センサーを使用することが前提となっていました。つまり、0 g オフセット誤差がなく、感度（アナログ・センサーでは mV/g、デジタル・センサーでは LSB/g の単位）誤差もないデバイスです。センサーはトリミング済みで納品されますが、そもそも機械部分を持っているデバイスであるため、システムの組立て時に PCB 等から製品に加わる応力によってオフセットと感度が影響を受けることがあります。さらに出荷時のトリミングの精度も限界があるので、アプリケーションで誤差が生じることがあります。

オフセット誤差の影響

どのくらいの大きさの誤差が生じるかを示すために、まず、感度は理想的ですが、x 軸に 50 mg のオフセットがある 2 軸ソリューションを考えてみましょう。0°において、x 軸の測定値は 50 mg、y 軸の測定値は 1 g です。これによって計算した角度は 2.9°となり、2.9°の誤差となります。±180°においては、x 軸は 50 mg、y 軸は -1 g になります。この場合、計算した角度と誤差は -2.9° になります。図 13 に、この例で計算した角度と実際の角度との誤差を示します。複数の軸のオフセットが関係する場合は、さらに複雑になります。

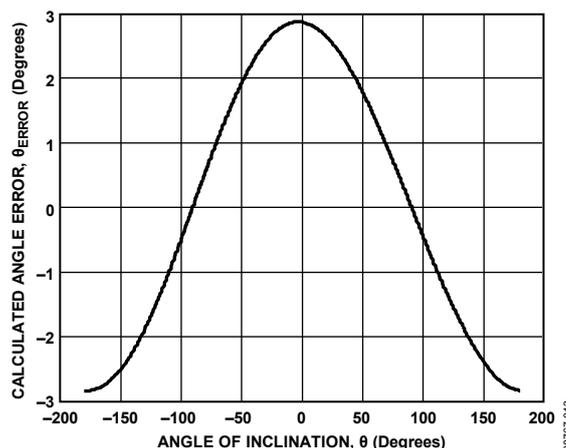


図 13. 加速度センサーのオフセットに起因する角度の計算誤差

感度ミスマッチ誤差の影響

2 軸の傾きの検出アプリケーションにおいて加速度センサーの感度に起因する主な誤差成分が生じるのは、対象となる軸間に感度の差が存在するときです（1 軸ソリューションの場合は、これとは異なり、実際の感度と予想される感度の間にずれがあれば誤差が生じます）。x 軸と y 軸の比を使用するため、感度が同じであれば、大部分の誤差が打ち消されます。

加速度センサーの感度のミスマッチによる影響の一例として、オフセット誤差がなく、y軸で感度誤差がなく、x軸に+5%の感度をもつ2軸ソリューションを使用する場合を考えてみましょう。つまり、1g入力時には、y軸は1gを示しますが、x軸は1.05gを示すことになります。図14は、この感度ミスマッチに起因する計算角度の誤差です。オフセット誤差と同様に、加速度センサーの感度のミスマッチに起因する誤差は回転の全範囲にわたって変動するため、傾き角の計算後に誤差を補償することは困難です。y軸の感度が変動しミスマッチがさらに大きくなると、誤差がさらに大きくなります。

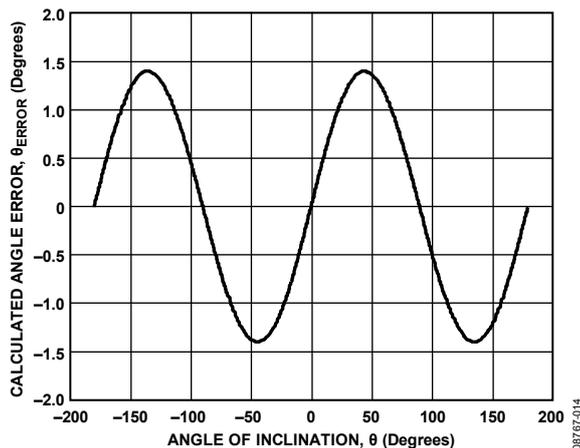


図 14. 加速度センサーの感度のミスマッチに起因する角度の計算誤差

基本的なキャリブレーション技術

オフセットと感度のミスマッチに起因する誤差が組み合わさると、誤差が非常に大きくなり、傾きの検出アプリケーションの許容できる誤差を大幅に上回ることがあります。この誤差を抑制するには、オフセットと感度のキャリブレーションを行い、キャリブレーション済みの出力加速度を使って傾き角度を計算する必要があります。オフセットと感度の影響を組み込むと、加速度センサー出力は次のようになります。

$$A_{OUT} [g] = A_{OFF} + (Gain \times A_{ACTUAL}) \quad (14)$$

ここで、

A_{OFF} はオフセット誤差であり、単位は g です。

$Gain$ は加速度センサーのゲインであり、理想的な値は 1 です。

A_{ACTUAL} は加速度センサーと所望の値に作用する実際の加速度であり、単位は g です。

簡単なキャリブレーションの方法は、ゲインを 1 としてオフセットを測定します。このキャリブレーションによって、システムの精度をキャリブレーションなしの感度の誤差のみになるように制限できます。この簡単なキャリブレーションを実行するには、対象となる軸を 0g フィールドに配置して出力を測定します。この出力がオフセットになります。アプリケーション上では加速度センサーの出力値からその値を引いてから、角度の計算を行います。この方法は、x軸とy軸のみを補正する場合は補正中に加速度センサーの方向を変える必要がないため、一般にノーターンまたはシングルポイント・キャリブレーションと呼ばれます。3軸デバイスを使用する場合は、z軸を 0g フィールドに配置するために少なくとも 1 回は方向転換を行う必要があります。

もっと正確にキャリブレーションを行う場合は、対象となる軸ごとに 2 つの点を使用します (3 軸設計で最大 6 点)。+1g と -1g の入力がある軸に加わっているとすると、測定される出力は次のようになります。

$$A_{+1g} [g] = A_{OFF} + (1g \times Gain) \quad (15)$$

$$A_{-1g} [g] = A_{OFF} - (1g \times Gain) \quad (16)$$

ここで、オフセット A_{OFF} の単位は g です。

この 2 つの点を用いて、オフセットとゲインを次のように求めることができます。

$$A_{OFF} [g] = 0.5 \times (A_{+1g} + A_{-1g}) \quad (17)$$

$$Gain = 0.5 \times \left(\frac{A_{+1g} - A_{-1g}}{1g} \right) \quad (18)$$

ここで、+1g と -1g の測定値である A_{+1g} と A_{-1g} の単位は g です。

このキャリブレーション方法は、対象となる軸の測定時に他の軸が 0g フィールドにあるため、他軸感度の影響を最小限に抑えることができます。この方法でキャリブレーションを行った場合には、まず加速度センサーの測定値からオフセットを引き、その結果をゲインで割ったものを使用して角度を計算します。

$$A_{ACTUAL} [g] = \frac{A_{OUT} - A_{OFF}}{Gain} \quad (19)$$

ここで、 A_{OUT} と A_{OFF} の単位は g です。

式 15～式 19 における A_{OFF} とゲインの計算では、加速度値 A_{+1g} と A_{-1g} の単位を g としています。mg 単位の加速度を使用する場合は、式 17 の A_{OFF} の計算は同じですが、式 18 のゲインの計算では、単位が変わるため 1000 で割る必要があります。