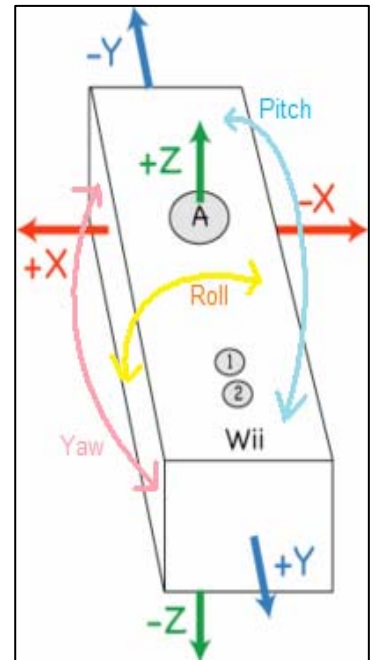


Motion analysis

From WiiLi.org

Wii リモコンとヌンチャクコントローラは3軸加速度計 (ASXL330) を内蔵しています。それはプレーヤによって与えられた瞬間的な力を、任意の単位で返します。静止時、上方向に重力があります。ですから自由落下時にはおよそ0の値を返します。3次元空間では6つの自由な角度を持ちます。3軸上の平行移動と3つの回転方向(前に倒れる→pitch、転がる→roll、振る→yaw)は右回りの値で示す。



目次

1. 較正。
2. 傾斜位置
3. 力
 - 3.1 純粋な回転
 - 3.2 回転以外
 - 3.3 ジャーク
4. ジェスチャー(身振り)
5. 引用。

1. 校正

加速度に反応するセンサーがおおかた線形で、かつチップが基板上の正確な位置にあると仮定するならば、Wii リモコンの構成は3段階でできます。必要とする力に関するデータは、平らに置いたときの位置と、レベルテーブルです。

上にAボタンがある水平状態を：

$$(x_1, y_1, z_1)$$

I Rセンサーを下にテーブルに載せて、そして拡張ポートを表面を上にする(?)：

$$(x_2, y_2, z_2)$$

その面を標準に、そして左側を上にしたときを(?)：

$$(x_3, y_3, z_3)$$

このとき、それぞれの1軸には重力がかかり、その他の2軸がゼロ点を示しています、これによって各軸のゼロ点を推測することができます。

$$\begin{aligned} x_0 &= (x_1 + x_2)/2 \\ y_0 &= (y_1 + y_3)/2 \\ z_0 &= (z_2 + z_3)/2 \end{aligned}$$

センサーから数値を読み始めると、その較正值は

$$\begin{aligned} x &= \frac{x_{raw} - x_0}{x_1 - x_0} \\ y &= \frac{y_{raw} - y_0}{y_2 - y_0} \\ z &= \frac{z_{raw} - z_0}{z_3 - z_0} \end{aligned}$$

この較正值はG = 1の場所での値です。

あるリモコンではこれらの定数は次のようになりました。

x_0	7
y_0	10
z_0	9.5
$x_1 - x_0$	27
$y_2 - y_0$	27
$z_3 - z_0$	27

3つの手順の較正に従うと、それが必要とするテーブルはユーザーにとって面倒かもしれません。ですから、正確でない点順も好ましいのかもしれませんが。上記の例にもとついて、センサーの利得は(基本的にはサイズはG)リモコンのゼロへのオフセットよりももっと近いのでは、と推測します(おそらく誤っていますが)

その場合の必要値は、リモコンを測定するために一時的に静止状態させ、そのときの0オフセットを3軸に一度適用すればよいのです。

実際のところこのような手順は平らな面を持たないヌンチャクセンサーの唯一の較正方法かもしれません。

2. 傾斜位置

リモコンが加速していないのならば、そのリモコンの傾きは重力を使って直接計算できます。初めにセンサーからの加速信号を $-1 \sim 1 \text{ g}$ になるように（上記の較正を見て）変換してください。すると、

$$\begin{aligned} \text{pitch} &= \text{asin}(a_x/1 \text{ g}) \\ \text{roll} &= \text{asin}(a_y/1 \text{ g}) \end{aligned}$$

これらの方程式に注意してください

- 手の動きによってリモコンに加速度が生じている場合、正常化された加速度値は1以上、 -1 未満になります。そしてASINはこれに定まりません。確認する方法のひとつとして、3個の加速度のベクトル和が g の近くにあるとき、リモコンが加速していないのを見ることです。

- それが垂直な方向から重力に向きを変えた時、加速度計が位置に対してより敏感でなくなることを注意してください。これによりリモコンを標準的な持ち方で持ったとき、 z 軸は加速度計の位置検出はあまり役立ちません。しかしリモコンがその位置から傾けられると、他の2軸に加えて z 軸を使うことが必要とされます。

- リモコンが加速しているならば、最新の値と比較して傾きの変化を取得し、加速度の2つの相和が必要となります。

- 揺れ(Yaw)の値を得るためにはIRセンサーを使わなければなりません。揺れはリモコンがセンサーの方から向きを変えたとき、そのときのセンサーのポイント速度、センサーとの距離（センサーの中の赤外線ポイントの間の距離から推定）から推定することが可能です。そして加速度計の全体を記録します。しかしながら、その遠すぎるとエラーが出させるでしょう。

3. 力

すべての既知の Wii リモコンは地球の表面で使われているので、プレイヤーが加速度をかけていないときでも、それにかかる力は一定です。(地球上にいるという予備知識なしでは、この力はプレイヤーによる加速度と見分けられません。一般相対性理論のすばらしさを思わせてくれるものです。)

$$\begin{aligned} a(t) &= R(t)(g\hat{z} + a_r(t)) \\ \dot{a}(t) &= \dot{R}(t)(g\hat{z} + a_r(t)) + R(t)\dot{a}_r(t) \end{aligned}$$

すべての回転の軸が、“モーションセンサー”または“回転が線形ではなく加速すること”に十分に近いと仮定します。(すなわち、レバーアームがありません)

加速度 $a(t)$ はセンサーによって計測されます (リモコンの加速度は真実ではない)。

R は回転行列で、リビングルームの枠をリモコンのフレームまでに変換します。

そして a はリビングルームの枠でのリモコンの実際の加速度です。

回転行列は 3 つのオイラーの角度の法則によりパラメーター化されます。またリモコンの線形加速は 3 つのコンポーネントを持っています。

6 つのダイナミックな変数とともにしますが 3 つしか観測できないので劣決定系します。

しかしながら、見せかけを単純化することで、いくらかの情報を抜粋することができます。

3.1 純粋な回転。

リモコンが加速していないとみなすならば (最適な通常使用状況で静的なとき) モーションセンサーは直接どちらが“上”かを示します。

これがオイラーの角度の法則のうちの 2 つを引き出します。しかしリビングルームフレーム (コントローラーが水平なとき、Yaw と一致する) の z 軸周りの角度ではないです。

この情報の不足は、3 つの加速度計でスクリーン上のポインターをコントロールする難しさの理由の一つです。

ユーザーのためにポインターの水平運動を示すもっとも自然な運動は、コントローラーを z 軸周りに回転させることです。

しかしながら 2 つの角度は、様々な用途のためには十分な規制です。

“上”のベクトルはセンサーの応答が直接使われることができる、または角度に分解します。

(公式をここにおいでください)

3.2 回転しないとき

リモコンが回転しない ($\dot{R}(t) = 0$) のは、運動方程式がそのままの唯一の条件です。

$$\begin{aligned} a(t) &= R(0)(g\hat{z} + a_r(t)) \\ \dot{a}(t) &= R(0)\dot{a}_r(t) \end{aligned}$$

このように $a_r(t)$ のよい推定を得るためには $R(0)$ の初期条件がさらに必要です。前のセクションであがっていた方法のようにリモコンが静止しているとき、この情報を得ることができます。

3.3 ジャーク

加速の変化率に面白く名付けられた語、それがジャークです（ほとんど使われていない）
ジャークは“センサーによって記録された力を時間による導関数で示したものがリモコンを上を示す”ことを称します。回転をもまた角速度を含む意味で同じです。少しだと仮定するならば、回転と線形加速を引き出すことができます。

運動が始まる前に“上”方向は分かっていますから、これは $R(t=0)\hat{z}$ です。

運動を通して、水平なりモコン上のジャークは重力の方向への小さいです。

それから時間による力の構成要素の導関数は、ユーザーがコントローラーを回転することによって起因する上方向を現在と推定すれば垂直です。これは次の時間ステップのために、上方向の推定を

更新することを許します。各時間ステップでは、力センサーの現在の $gR(t)\hat{z}$ の推定から引くことによって、リモコンの線形加速を手に入れることもできます。実質的に常微分方程式のセットの中にあります。（注. 数学をここでチェックしてください。注意します）

この技術に関する主な問題は“上”に関する予想の蓄積によるエラーです。これはユーザーがコントローラーを一定の線形動作を「テレビ」「彼らの対戦相手」「彼ら自身」を負傷させずに保持しそ

もないですし、力が $G=1.0$ から最新の $R(t=0)$ に近いと合計を報告したときに時間を見つけることができます。

4. ジェスチャー

最初に想像するよりも多い状況のために、それは正確な回転やリビングルーム参照フレームで線形加速を引き出すことは完全に不必要です。そのこととして記述されました。Wii のための多くのゲームはあらかじめ定義された運動パターンを通してリモコンの問題命令を使うことの力を示しました。たとえば、剣攻撃をシミュレートするためにリモコンを切り払ったり、投げ縄のような円でリモコンの周りに激しく当たります。これらのジェスチャーのような命令を運動データはいくらかの曖昧なパターンマッチングを必要とします。これはコントローラーの小さな回転にもまた鈍感です。

AiLive は Wii 開発者が動的に利用できるように、彼らが身体的にコントローラーを望ましいパターンを何度か動かすことで異なる動作を認識する訓練をします。

詳細な情報は彼らの白書にあります。

(その他に発表されたこの類の運動認識への研究はここで歓迎されます。)

5. 引用

Introduction to using accelerometers (A presentation from Analog Devices):

http://www.analog.com/Analog_Root/static/library/techArticles/mems/sensor971.pdf

Some of the data sheets for Analog Devices accelerometers have hints for how to analyze the output:

<http://www.analog.com/en/subCat/0,2879,764%255F800%255F0%255F%255F0%255F,00.html>