# 1 Szenario

Gegeben sind ein ruhender, oder sich mit gleichmäßiger Geschwindigkeit bewegender, triaxialer Beschleunigungssensor. Wie bestimmt man seine Orientierung relativ zum Gravitationsvektor ?

# 2 Idee

Da außer der Gravitationskraft keine weitere Kraft aktiv unseren Sensor beeinflusst, so lässt sich aus den einzelnen Messergebnissen unsere Orientierung zum Gravitationsvektor bestimmen. Zur Erinnerung, ein Beschleunigungssensor ist ein Masse-Feder System, dessen Masse sich bei einer auftretenden externen Kraft (und damit Beschleunigung) in Bewegung setzt und unsere Feder auslenkt. Ordnet man nun 3 dieser Sensoren orthogonal im Beschleunigungssensor an, so liefern die 3 Sensoren die Kraftkomponenten in $x$,$y$ und $z$-Richtung. Wir kennen also von unserem Gravitationsvektor jeweils seine Komponenten entlang der Achsen. Wir führen nun die Winkel $\alpha$ als Winkel von der $x$-Achse, den Winkel $\beta$ als Winkel von der $y$-Achse und den Winkel $\gamma$ als Winkel von der $z$-Achse ein.Betrachten wir nun die Projektion von $\vec{g}$ in beispielsweise die $XY$-Ebene.Wir entdecken den Winkel $\alpha$ als Winkel in einem Rechtwinkligen Dreieck mit Katheten $g_x$ und $g_y$.Daraus können wir nun sofort unsere Bestimmungsgleichung für $\alpha$ (und analog für $\beta$ und $\gamma$) ermitteln :

$$\begin{aligned}
\alpha &= atan2(g_y, g_x) \\
\beta &= atan2(g_z, g_x) \\
\gamma &= atan2(g_x, g_z)
\end{aligned}$$

# 3 Implementierung

Es kann analog zu obiger Überlegung implementiert werden.

```
Vector3D getOrtsvector(double gx,double gy,double gz){
        double alpha = atan2(gy,gx);
        alpha = alpha*180/M_PI;

        double beta = atan2(gz,gy);
        beta = beta*180/M_PI;

        double gamma = atan2(gx,gz);
        gamma = gamma*180/M_PI;

        Vector3D g(alpha,beta,gamma);
        return g;
}
```

## 3.1 Wichtiger Hinweis

Es ist wichtig, die Werte der Sensoren in Ausgangslage zu kennen, so lassen sich später einfacher die Einflüsse anderer wirkenden Kräfte identifizieren.

# 4 Veränderung der Lage mittels Quaternion

Erhalten wir jetzt 2 (oder mehrere) solcher Messungen, und berücksichtigen wir, dass der Gravitationsvektor stets in die selbe Richtung zeigt, so können wir über :

$$\Delta\alpha = \alpha_2 - \alpha_1$$
$$\Delta\beta = \beta_2 - \beta_1$$
$$\Delta\gamma = \gamma_2 - \gamma_1$$

Die Orientierungsänderung berechnen. Sei nun :

$$\Delta\alpha =: Yaw$$
$$\Delta\beta =: \mathit{Roll}$$
$$\Delta\gamma =: Pitch$$

Über diese 3 Winkel können wir nun unser Quaternion erstellen, über welches wir aus unserer Ausgangslage P in die aktuelle Lage $P_{akt}$ gekommen sind.In der Bibliothek steht hierfür der Konstruktor:

```
Quaternion(const YPR& ypr)
```

zur Verfügung. Kannten wir unsere Lage P, so können wir nun unsere neue Lage errechnen.

```
Vector3D P(1,2,3) // ausgangslage
Vector3D P_akt;
YPR ypr (gamma,beta,alpha);
Quaternion q(ypr);
P_akt = P.qRotate(q);
```