1 Szenario

Man befindet sich in einem Flugzeug auf einem Flug entlang eines Breitengrades ϕ , zB von Europa nach Amerika. Am Flugzeug sei ein körpereigenes Koordinatensystem angebracht, der Frame B. Als globales Referenzsystem soll hier das ECEF zugrunde liegen. Die aktuelle Lage des Flugzeugs wird in den Eulerwinkeln Yaw α , Pitch β und Roll γ als Referenz zum Bodyframe B angegeben. Zu Beginn sei das Koordinatensystem des Flugzeugs wie in Figur 1 angeordnet.

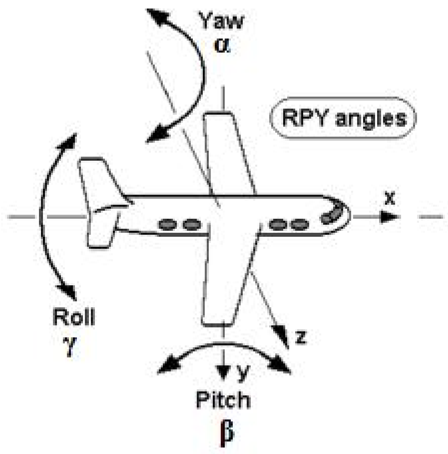

Abbildung 1: Definition des Body-Frames

Ein Flug entlang eines Breitengrades kann nun gedeutet werden, als eine Rotation im ECEF um die z-Achse. Wie ändert sich nun die Lage des Flugzeugs bei diesem Flug, insofern keine Nachkorrekturen während des Fluges stattfinden ?

2 Idee

Betrachtet man den Origin des Bodyframes B als Vektor $\vec{o}$, so kann der Origin $\vec{O}$ nach der Rotation wie folgt errechnet werden:

$$\vec{O} = R_z \cdot \vec{o}$$

Hier stellt R_z die Fundamentalrotationsmatrix um die z-Achse dar, und ist gegeben als:

$$R_z(\alpha) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Dies kann sehr einfach Implementiert werden:

```
Vector3D origin(1000,2000,3000); // origin ECEF
Matrix3D Id;        // Einheitsmatrix
double angle = M_PI/2;
Matrix3D R_z = Id.rotateZ(angle);
Vector3D  rotated_or= origin.mRotate(R_z);
```

Da hier nur die Orientierung des Flugzeugs von Interesse ist, setzen wir den Origin $\vec{o}$ auf 0 Zu jeder Zeit kann die Orientierung im eigenen Frame durch die Winkel α, β und γ gemessen werden. Eine Eigenschaft einer Rotation, ist die Erhaltung des Skalarprodukts, es bleiben also alle Winkel zwischen Vektoren erhalten, und damit bleibt auch die Lage im Bodyframe B konstant. Was sich jedoch ändert ist die Orientierung des Bodyframes gegenüber des ECEF. Eine Rotation eines Frames kann als Rotation aller seiner Achsen verstanden werden. Der Einfachheit halber werden sowohl α, β und γ zu Beginn mit 0 Initialisiert, das Flugzeug ist also längs der x-Achse Orientiert. Die Koordinatenachsen x,y,z werden als $(1,0,0)$, $(0,-1,0)$ und $(0,0,-1)$ gemäß Figur 1 initialisiert.

```
Vector3D x(1,0,0);
Vector3D y(0,-1,0);
Vector3D z(0,0,-1);

CoordinateFrame3D body(x,y,z);
CoordinateFrame3D rotated;
rotated = body.rotate(R_z);
```

Als Ergebnis erhalten wir für die Rotation um $\phi = \frac{\pi}{2}$ $x = (0,1,0)$, $y = (1,0,0)$ und $z = (0,0,-1)$. Im globalen ECEF Frame hat sich unser Flugzeug also ebenfalls um $\frac{\pi}{2}$ gegenüber der Urpsrungslage gedreht.