

Lecture

INS Initialization

EE 570: Location and Navigation

Lecture Notes Update on April 19, 2016

Aly El-Osery and Kevin Wedeward, Electrical Engineering Dept., New Mexico Tech
In collaboration with
Stephen Bruder, Electrical & Computer Engineering, Embry-Riddle Aeronautical University

_____ .1

Overview

Position, velocity and attitude drift unless the INS is aided. There are some opportunistic situations that provide information to the INS to initialize itself. Two categories of alignment

- Coarse Alignment
- Fine Alignment

_____ .2

Self-Alignment

1. *Coarse Alignment*: Use knowledge of the gravity vector and earth rate provided by the three accelerometers, and the knowledge of the earth rate vector provided by the gyroscopes.
2. *Fine Alignment*: Needed in quasi-stationary situations. Uses the fact that any position, velocity changes are considered disturbances, and the knowledge of the gravity vector and earth rate to estimate the body's attitude.

Latitude needs to be known.

_____ .3

Coarse Alignment: Approach 1

$$\vec{f}_{ib}^b = -C_n^b \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} = \begin{pmatrix} \sin(\theta) \\ -\cos(\theta) \sin(\phi) \\ -\cos(\theta) \cos(\phi) \end{pmatrix} g$$

Only provides pitch and roll angles
g (+ve)

_____ .4

Coarse Alignment: Approach 2

$$\begin{pmatrix} \tilde{f}_{ib}^b & \tilde{\omega}_{ib}^b & \tilde{f}_{ib}^b \times \tilde{\omega}_{ib}^b \end{pmatrix} = \hat{C}_n^b \begin{pmatrix} \vec{f}_{ib}^n & \vec{\omega}_{ib}^n & \vec{f}_{ib}^n \times \vec{\omega}_{ib}^n \end{pmatrix}$$
$$\hat{C}_n^b = \begin{pmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{pmatrix}$$

where

$$\begin{aligned}
C_{11} &= \frac{\tilde{\omega}_x^b}{\omega_{ie} \cos(L_b)} - \frac{\tilde{f}_x^b \tan(L_b)}{g} \\
C_{21} &= \frac{\tilde{\omega}_y^b}{\omega_{ie} \cos(L_b)} - \frac{\tilde{f}_y^b \tan(L_b)}{g} \\
C_{31} &= \frac{\tilde{\omega}_z^b}{\omega_{ie} \cos(L_b)} - \frac{\tilde{f}_z^b \tan(L_b)}{g} \\
C_{12} &= \frac{\tilde{f}_z^b \tilde{\omega}_y^b - \tilde{f}_y^b \tilde{\omega}_z^b}{g \omega_{ie} \cos(L_b)} \\
C_{22} &= \frac{-\tilde{f}_z^b \tilde{\omega}_x^b + \tilde{f}_x^b \tilde{\omega}_z^b}{g \omega_{ie} \cos(L_b)} \\
C_{32} &= \frac{\tilde{f}_y^b \tilde{\omega}_x^b - \tilde{f}_x^b \tilde{\omega}_y^b}{g \omega_{ie} \cos(L_b)} \\
C_{13} &= \frac{-\tilde{f}_x^b}{g} \\
C_{23} &= \frac{-\tilde{f}_y^b}{g} \\
C_{33} &= \frac{-\tilde{f}_z^b}{g}
\end{aligned}$$

Must ensure that the DCM is properly orthogonalized.

.5

Fine Alignment

- Use full INS mechanization
- Use equivalent to GPS aided error mechanization
- Setup up measurements
 1. Specific force measurement

$$\delta \vec{f}_{ib}^b = \vec{f}_{ib}^b - \hat{\vec{f}}_{ib}^b$$

2. Angular rate measurement

$$\delta \vec{\omega}_{ib}^b = \vec{\omega}_{ib}^b - \hat{\vec{\omega}}_{ib}^b$$

3. Position measurement: deviation from initial position
4. Velocity measurement: deviation from zero

.6

Specific force measurement

$$\begin{aligned}
\delta \vec{f}_{ib}^n &= \vec{f}_{ib}^n - \hat{\vec{f}}_{ib}^n \\
&= \vec{f}_{ib}^n - (I - [\delta \vec{\psi}_{nb}^n \times]) \hat{C}_b^n (\vec{f}_{ib}^b - \delta \vec{f}_{ib}^b) + f_d \\
&= [\delta \vec{\psi}_{nb}^n \times] \hat{C}_b^n \vec{f}_{ib}^b + \hat{C}_b^n \delta \vec{f}_{ib}^b + f_d \\
&= \begin{pmatrix} 0 & -\delta \psi_D & \delta \psi_E \\ \delta \psi_D & 0 & -\delta \psi_N \\ -\delta \psi_E & \delta \psi_N & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ -g \end{pmatrix} + \hat{C}_b^n \delta \vec{f}_{ib}^b + f_d \\
&= \begin{pmatrix} 0 & -g & 0 \\ g & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \delta \psi_N \\ \delta \psi_E \\ \delta \psi_D \end{pmatrix} + \hat{C}_b^n \delta \vec{f}_{ib}^b + f_d \\
&= G \delta \vec{\psi}_{nb}^n + \hat{C}_b^n \delta \vec{f}_{ib}^b + f_d^b
\end{aligned}$$

Recoordinatize in the body frame

$$\delta \vec{f}_{nb}^b = \hat{C}_n^b G \delta \vec{\psi}_{nb}^n + \delta \vec{f}_{ib}^b + \vec{f}_d^b$$

$\delta \vec{f}_{ib}^b$ captures bias-drift (sinking) + Markov, ..., and f_d represents variations in the nav frame

.7

Angular Rate Measurement

$$\begin{aligned}
\delta \vec{\omega}_{in}^n &= \vec{\omega}_{in}^n - \hat{\vec{\omega}}_{in}^n \\
&= (I + [\delta \vec{\psi}_{nb}^n \times]) \hat{C}_b^n (\vec{\omega}_{ib}^b + \vec{\omega}_{bn}^b) - \hat{C}_b^n (\vec{\omega}_{ib}^b - \delta \vec{\omega}_{ib}^b) \\
&= \begin{pmatrix} 0 & \Omega_D & 0 \\ -\Omega_D & 0 & \Omega_N \\ 0 & -\Omega_N & 0 \end{pmatrix} \begin{pmatrix} \delta \psi_N \\ \delta \psi_E \\ \delta \psi_D \end{pmatrix} + \hat{C}_b^n \delta \vec{\omega}_{ib}^b - \vec{\omega}_d^n \\
&= W \delta \vec{\psi}_{nb}^n + \hat{C}_b^n \delta \vec{\omega}_{ib}^b - \vec{\omega}_d^n
\end{aligned}$$

Recoordinatize in the body frame

$$\delta \vec{\omega}_{in}^b = \hat{C}_n^b W \delta \vec{\psi}_{nb}^n + \delta \vec{\omega}_{ib}^b - \vec{\omega}_d^b$$

$\delta \vec{\omega}_{ib}^b$ captures bias-drift (sinking) + Markov, ..., $vcs\omega_d$ represents variations from stationarity.

.8

Error State and Measurement Matrix

$$\begin{aligned}
\dot{\vec{x}}(t) &= F(t)\vec{x}(t) + \vec{w}(t) \\
\vec{y}(t) &= H(t)\vec{x}(t) + \vec{v}(t) \\
\vec{x} &= \left(\delta \vec{\psi}_{nb}^n \quad \delta \vec{v}_{nb}^n \quad \delta \vec{r}_{nb}^n \quad \vec{b}_a \quad \vec{b}_g \right)^T \\
\mathbf{H} &= \begin{pmatrix} 0_{3 \times 3} & 0_{3 \times 3} & \mathbf{I}_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & \mathbf{I}_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ \hat{C}_n^b G & 0_{3 \times 3} & 0_{3 \times 3} & \mathbf{I}_{3 \times 3} & 0_{3 \times 3} \\ \hat{C}_n^b W & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{pmatrix}
\end{aligned}$$

where the measurements are: position error, velocity error, specific force error, and angular velocity errors, respectively.

.9

Challenges

There is no mechanism in the above formulation to estimate $\vec{\omega}_d^n$. If it can be modelled as white noise then the filter will be able to handle it. On the other hand, if it is correlated type of disturbance, additional measures must be taken to account for it.

.10