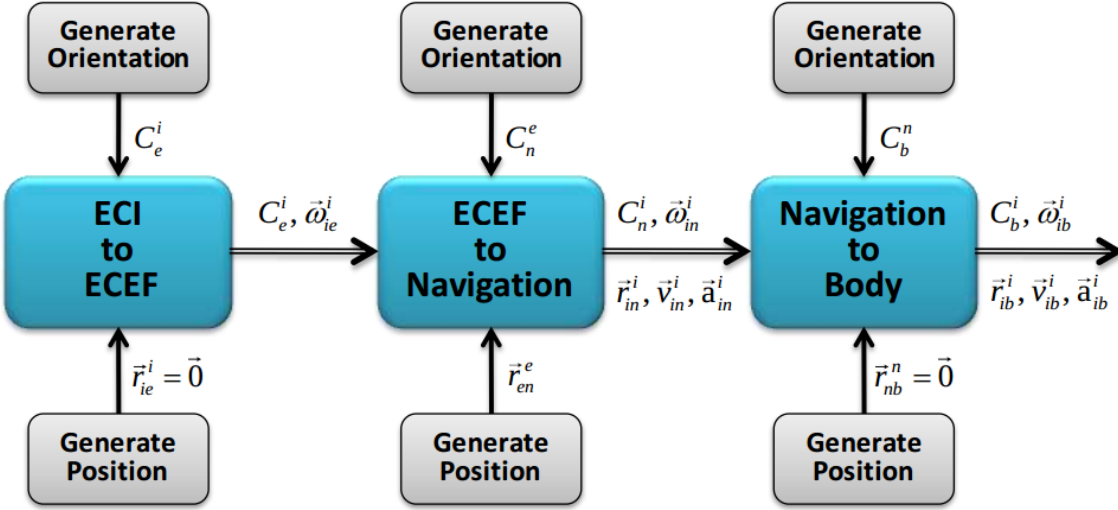


EE 570: Project 1

This project involves development of a MATLAB program to generate consistent, synthetic “ground truth data” using a process outlined in the figure below. This data will then be utilized to test our INS mechanization implementations assuming perfect sensors and later (projects 2 and 3) we will use this “ground truth generator” to stimulate the IMU in generating more realistic (i.e. noisy with errors) scenarios.

The inputs to each stage along the way will be some combination of position, orientation, velocity (both angular and linear), and acceleration (both angular and linear) such that any reasonable kinematic quantity can be computed and resolved in any of the four frames.



Note that analytical functions may or may not be given to specify position and/or orientation, as such, numerical differentiation may be required in some cases. As was the case in the last homework we might use an intermediate ECEF frame (e.g. the tracking station) to describe the position of the origin of the body/nav frame. Notation for MATLAB variable names is $\vec{a}_{jk}^i = \mathbf{a}_{i_j_k}$ and $C_e^i = \mathbf{C}_{i_e}$

1 Ground Truth Generation

An aircraft carrier traveling in a circular pattern in the North Pacific Ocean uses its tracking radar to observe an aircraft spiraling up above the carrier. The location of the origin of the tracking radar coordinates frame (i.e., t -frame) is given in geodetic curvilinear coordinates by:

$$L_t = (30 + 0.01 \cos(0.01t)) \frac{\pi}{180} \text{ rad}$$

$$\lambda_t = (-135 + 0.01 \sin(0.01t)) \frac{\pi}{180} \text{ rad}$$

$$h_t = 15 + 2 \sin(\pi t) \text{ meters}$$

The tracking radar on the aircraft carrier is very precisely inertially stabilized and as such always reports the location of objects (e.g. an aircraft) in a coordinate frame which remains aligned with its initial nav frame $C_t^e = C_n^e(L_t(0), \lambda_t(0))$. Specifically, the position vector describing the location of the origin of the body frame of the aircraft with respect to the origin of the tracking radar frame resolved in the tracking radar frame is given by

$$\vec{r}_{tb}^t = \begin{bmatrix} 10,000 \cos(0.03t) \\ 10,000 \sin(0.03t) \\ -5,000 - 5t \end{bmatrix}$$

Finally, the orientation of the body frame of the aircraft with respect to the aircraft's navigation frame is given by ($a = 10,000$, $b = 5$, and $\omega = 0.03$)

$$C_b^n = \begin{bmatrix} \frac{-a\omega \sin(\omega t)}{\sqrt{a^2\omega^2 + b^2}} & -\cos(\omega t) & \frac{-b \sin(\omega t)}{\sqrt{a^2\omega^2 + b^2}} \\ \frac{a\omega \cos(\omega t)}{\sqrt{a^2\omega^2 + b^2}} & -\sin(\omega t) & \frac{b \cos(\omega t)}{\sqrt{a^2\omega^2 + b^2}} \\ \frac{-b}{\sqrt{a^2\omega^2 + b^2}} & 0 & \frac{a\omega}{\sqrt{a^2\omega^2 + b^2}} \end{bmatrix}$$

The angular velocity of the body frame wrt the nav frame resolved in the nav frame can be obtained from

$$\dot{C}_b^n = \Omega_{nb}^n C_b^n \Rightarrow \Omega_{nb}^n = \dot{C}_b^n (C_b^n)^T = \begin{bmatrix} 0 & -\omega & 0 \\ \omega & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \Rightarrow \vec{\omega}_{nb}^n = \begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix}$$

The MATLAB file “project_1.m” performs the ground truth generation of some of the quantities you will need later. Run the code from $t = 0$ to $t = 180$ seconds with 0.01 second time steps (i.e., $\Delta t = 0.01$ sec) and perform a “sanity” check on the plots which are generated. Position is plotted as $\vec{r}_{?b}^? = [r_x \ r_y \ r_z]^T$, velocity as $\vec{v}_{?b}^? = [v_x \ v_y \ v_z]^T$, and orientation as an angle-axis vector $C = [\vec{k} \times]$.

2 ECI Mechanization

Develop a function to perform the ECI mechanization (code skeleton is provided) and plot the errors between the ground thruth PVA and the PVA derived from the output of your INS mechanization

1. Initialize (i.e., $t = 0$) the PVA with the “truth”
2. Plot the error in the position as truth - INS estimate (i.e., $\delta \vec{r}_{ib}^i = \vec{r}_{ib}^i - \hat{\vec{r}}_{ib}^i$)
3. Plot the error in the velocity as truth - INS estimate (i.e., $\delta \vec{v}_{ib}^i = \vec{v}_{ib}^i - \hat{\vec{v}}_{ib}^i$)
4. Plot the three components of the error in the attitude defined by truth DCM \times INS estimate DCM^T, i.e.,

$$\delta C_b^i = C_b^i (\hat{C}_b^i)^T \approx \mathcal{I}_{3 \times 3} + \Psi_{ib}^i$$

where

$$\Psi_{ib}^i = [\hat{\psi}_{ib}^i \times] = \begin{bmatrix} 0 & -\psi_z & \psi_y \\ \psi_z & 0 & -\psi_x \\ -\psi_y & \psi_x & 0 \end{bmatrix}$$

3 ECEF Mechanization

Develop a function to perform the ECEF mechanization (code skeleton is provided) and plot the errors between the ground truth PVA and the PVA derived from the output of your INS mechanization

1. Initialize (i.e., $t = 0$) the PVA with the “truth”
2. Plot the error in the position as truth - INS estimate (i.e., $\delta\vec{r}_{eb}^e = \vec{r}_{eb}^e - \hat{\vec{r}}_{eb}^e$)
3. Plot the error in the velocity as truth - INS estimate (i.e., $\delta\vec{v}_{eb}^e = \vec{v}_{eb}^e - \hat{\vec{v}}_{eb}^e$)
4. Plot the three components of the error in the attitude defined by truth DCM \times INS estimate DCM^T, i.e.,

$$\delta C_b^e = C_b^e(\hat{C}_b^e)^T \approx \mathcal{I}_{3 \times 3} + \Psi_{eb}^e$$

where

$$\Psi_{eb}^e = [\vec{\psi}_{eb}^e \times] = \begin{bmatrix} 0 & -\psi_z & \psi_y \\ \psi_z & 0 & -\psi_x \\ -\psi_y & \psi_x & 0 \end{bmatrix}$$

4 Nav Mechanization

Develop a function to perform the Nav mechanization (code skeleton is provided) and plot the errors between the ground truth PVA and the PVA derived from the output of your INS mechanization.

1. Initialize (i.e., $t = 0$) the PVA with the “truth”
2. plot the error in the position as the truth-INS estimate (i.e., $\delta L_b = L_b - \hat{L}_b$, $\delta \lambda_b = \lambda_b - \hat{\lambda}_b$, and $\delta h_b = h_b - \hat{h}_b$)
3. Plot the error in the velocity as truth - INS estimate (i.e., $\delta\vec{v}_{eb}^n = \vec{v}_{eb}^n - \hat{\vec{v}}_{eb}^n$)
4. Plot the three components of the error in the attitude defined by truth DCM \times INS estimate DCM^T, i.e.,

$$\delta C_b^n = C_b^n(\hat{C}_b^n)^T \approx \mathcal{I}_{3 \times 3} + \Psi_{nb}^n$$

where

$$\Psi_{nb}^n = [\vec{\psi}_{nb}^n \times] = \begin{bmatrix} 0 & -\psi_z & \psi_y \\ \psi_z & 0 & -\psi_x \\ -\psi_y & \psi_x & 0 \end{bmatrix}$$

5 %10 Bonus

Develop the implementation for the ECEF and the Nav frame using quaternions.