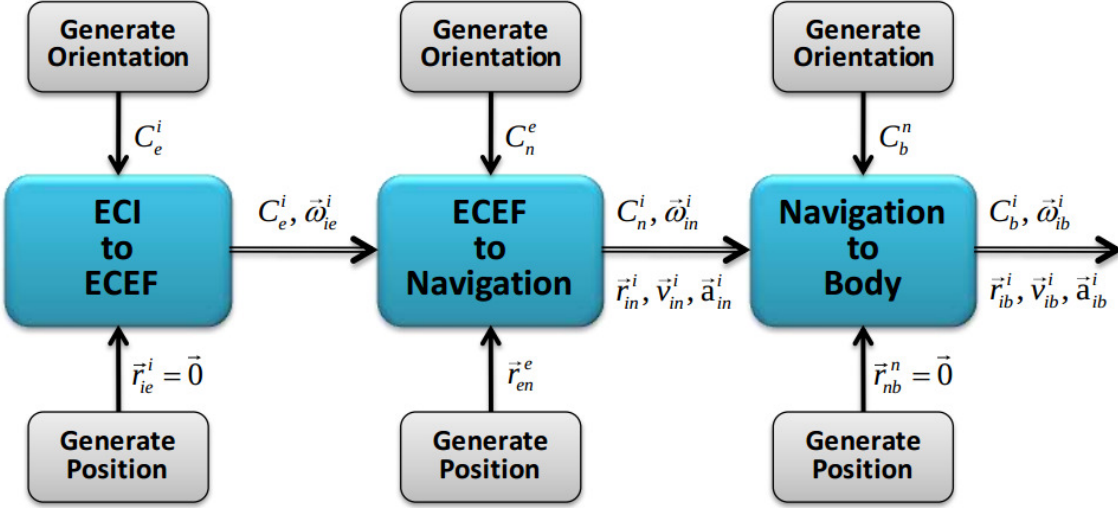


EE 570: Project 3

Project 1 involved the development of a MATLAB program to generate consistent, synthetic “ground truth” data using a process outlined in the figure below. In project 2 you have used realistic sensor performance to generate the INS measurements. In this project you will design a GPS aided INS using a loosely coupled configuration (with feedback).



1 Ground Truth Generation (from Project #1)

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 ECEF Mechanization (from Project #1)

Develop a function to perform the “**high-fidelity**” 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

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

3 IMU Implementation (from Project #2)

Develop a function to perform the IMU implementation (code skeleton is provided as imu.m and an updated load_constants.m file). The IMU takes the error free version of the specific force \vec{f}_{ib}^b and angular velocity $\vec{\omega}_{ib}^b$ as inputs and outputs measured versions of the same, namely,

$$\tilde{\vec{f}}_{ib}^b = \vec{f}_{ib}^b + \delta \vec{f}_{ib}^b = \vec{b}_a + (\mathcal{I} + M_a) \vec{f}_{ib}^b + \vec{w}_a$$

and

$$\tilde{\vec{\omega}}_{ib}^b = \vec{\omega}_{ib}^b + \delta \vec{\omega}_{ib}^b = \vec{b}_g + (\mathcal{I} + M_g) \vec{\omega}_{ib}^b + G_g \vec{f}_{ib}^b + \vec{w}_g$$

For this project we will consider a commercially available kvh 1750 IMU. For the questions below you will need to plot the errors between the ground truth PVA and the PVA derived from the

output of your ECEF-frame INS mechanization, now using measurements provided by a simulated (imperfect) IMU.

1. Use the datasheet to obtain all of the relevant IMU error characteristics:

Gyroscope Bias stability ($b_{g,BS}$), bias instability ($b_{g,BI}$), assume that fixed bias has been calibrated and is zero ($b_{g,FB} = 0$), ARW (\vec{w}_g), scale factor stability (s_g). Also, assume that the gyro g-sensitivity (G_g) is $0.5^\circ/\text{hr}/\text{g}$ and the misalignment terms (e.g., $m_{g,xy}$) are as given in the code skeleton (see `imu.m`).

Accelerometers Bias stability ($b_{a,BS}$), bias instability ($b_{a,BI}$), assume that fixed bias has been calibrated and is zero ($b_{a,FB} = 0$), VRW (\vec{w}_a), scale factor stability (s_a). Also, assume that the accel offset error is zero and that misalignment terms (e.g., $m_{a,xy}$) are as given in the code skeleton (see `imu.m`).

4 Design an Aided INS

The following are the parameters used for the aided INS

- GPS (position: $\sigma_p = 3\text{m}$; velocity: $\sigma_v = 0.01\text{m/s}$)
- GPS rate: 1Hz
- INS rate: 1kHz
- IMU specs (same as in project 2, see above)
- Simulation run time: at least 180sec

Develop an aided INS system using the GPS signal (use the GPS sensor function provided). Your system should be able to handle corrupt GPS measurements.

Your report should include the following:

1. Description of the aiding system you are designing
2. What are your aiding measurement
3. Explicitly define your system equations and define the matrices with units: $F(t)$, $G(t)$, $Q(t)$, H , P , and R .
4. Describe how you create your discretize system, i.e., what is $\phi(k)$ and $Q(k)$
5. Show a diagram of your closed-loop system
6. Figures showin the performance of the system including
 - (a) The INS system alone without noise
 - (b) The INS with IMU and no aiding
 - (c) The INS with aiding
7. Simulate GPS outage or corruption, from $t = 30\text{s}$ to $t = 60\text{s}$, by adding bad values to its position and velocity values
8. What is the effect of the IMU sampling rate on your results (use simulation to support your claims).
9. Provide your code with comments in the apendix