INS + GNSS Integration

*

This document presents the derivation of a simple INS solutions obtain by a loosely coupling of IMU and GNSS. In addition, the terrestrial vehicle constraints will be included to improve the results.

I. IMU Inertial Solution

I. Coordinate Frame Conversion

In this work the attitude will be measured with the Euler angles: roll (ϕ), pitch (θ) and yaw (ψ). The order of rotation chosen from the navigation frame, (n), to the body frame, (b), is:

$$(\text{n}) \rightarrow \psi \rightarrow \theta \rightarrow \phi \rightarrow (\text{b})$$

also referred to as {3,2,1}. The rotation matrix that converts from (n) to (b) is obtained concatenating rotations. First, a yaw rotation around the (n) Z-axis to transform to the intermediate axis (j) such that:

$$\mathbf{a}^n = \underbrace{\begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{R}_\psi} \mathbf{a}^j \tag{1}$$

Then a pitch rotation around the (j) Y-axis to transform to intermediate frame (e) as:

$$\mathbf{a}^j = \underbrace{\begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix}}_{\mathbf{R}_\theta} \mathbf{a}^e \tag{2}$$

Finally a roll rotation around the (e) X-axis to transform to the body frame (b) as:

$$\mathbf{a}^e = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix}}_{\mathbf{R}_\phi} \mathbf{a}^b \tag{3}$$

Finally, concatenating all three rotations:

$$\mathbf{a}^n = \underbrace{\mathbf{R}_\psi \mathbf{R}_\theta \mathbf{R}_\phi}_{{}^n\mathbf{R}^b} \mathbf{a}^b \tag{4}$$

where:

$${}^n\mathbf{R}^b(\mathbf{E}) = \begin{bmatrix} c\psi c\theta & c\psi s\phi s\theta - c\phi s\psi & s\phi s\psi + c\phi c\psi s\theta \\ c\theta s\psi & c\phi c\psi + s\phi s\psi s\theta & c\phi s\psi s\theta - c\psi s\phi \\ -s\theta & c\theta s\phi & c\phi c\theta \end{bmatrix} \tag{5}$$

*

and the vector of Euler angles is:

$$\mathbf{E} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \tag{6}$$

II. From Body Frame to Euler Angle Rates, i.e. $\boldsymbol{w}^b \rightarrow \dot{\mathbf{E}}$

This section converts from the angular rates measured in the body frame to the Euler angular rates used to measure the attitude. The robot angular velocity is measured by the gyros in the body frame, (b), as:

$$\boldsymbol{w} = w_x^b \hat{\mathbf{b}}_1 + w_y^b \hat{\mathbf{b}}_2 + w_z^b \hat{\mathbf{b}}_3 \tag{7}$$

where the quantities w_x^b, w_y^b and w_z^b are measured by the gyros. The angular velocity can also be defined in terms of Euler angle rates using the intermediate frames from Section I as:

$$\boldsymbol{w} = \dot{\psi} \hat{\mathbf{n}}_3 + \dot{\theta} \hat{\mathbf{j}}_2 + \dot{\phi} \hat{\mathbf{e}}_1 \tag{8}$$

Transforming to (b) each of the unitary vectors:

$$\boldsymbol{w} = \dot{\psi}^b \mathbf{R}^n \hat{\mathbf{n}}_3 + \dot{\theta}^b \mathbf{R}^j \hat{\mathbf{j}}_2 + \dot{\phi}^b \mathbf{R}^e \hat{\mathbf{e}}_1 \tag{9}$$

where:

$$^b\mathbf{R}^e = \mathbf{R}_\phi^T \quad , \quad ^b\mathbf{R}^j = \mathbf{R}_\phi^T \mathbf{R}_\theta^T \quad \text{and} \quad ^b\mathbf{R}^n = \mathbf{R}_\phi^T \mathbf{R}_\theta^T \mathbf{R}_\psi^T = \left(^n\mathbf{R}^b \right)^T \tag{10}$$

This results in the angular velocity in terms of Euler angles rates:

$$\boldsymbol{w} = \underbrace{\left(\dot{\phi} - \dot{\psi} \sin\theta\right)}_{w_x^b} \hat{\mathbf{b}}_1 + \underbrace{\left(\dot{\theta} \cos\phi + \dot{\psi} \sin\phi \cos\theta\right)}_{w_y^b} \hat{\mathbf{b}}_2 + \underbrace{\left(-\dot{\theta} \sin\phi + \dot{\psi} \cos\phi \cos\theta\right)}_{w_z^b} \hat{\mathbf{b}}_3 \tag{11}$$

By identification with equation (8), the components measured by the IMU gyros can be identified as it is shown underbraced. Using a matrix notation:

$$\begin{bmatrix} w_x^b \\ w_y^b \\ w_z^b \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{12}$$

Inverting the matrix, we obtain the conversion from body angle rates to Euler angle rates:

$$\dot{\mathbf{E}} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & \sin\phi & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \dfrac{\sin\phi}{\cos\theta} & \dfrac{\cos\phi}{\cos\theta} \end{bmatrix}}_{^E\mathbf{Q}^b} \begin{bmatrix} w_x^b \\ w_y^b \\ w_z^b \end{bmatrix} = \boldsymbol{w}^b \tag{13}$$

or

$$\dot{\mathbf{E}} =^E \mathbf{Q}^b \boldsymbol{w}^b \tag{14}$$

where matrix that transforms from body to Euler rates is defined as $^E\mathbf{Q}^b(\boldsymbol{\theta})$.

III. State Evolution Model with IMU Inputs

The states vector is composed of Cartesian position and velocity in the navigation frame and, the Euler angles (roll, pitch, yaw) from the body frame to the navigation frame. The navigation frame is defined with North East Down (NED) axis and some known location, e.g. the initial location of the robot. The system inputs that will be provided by the IMU are the specific forces and angular rates measured in the body frame.

$$\mathbf{x} = \begin{bmatrix} \mathbf{r}^n \\ \mathbf{v}^n \\ \mathbf{E} \\ \mathbf{b}_f \\ \mathbf{b}_w \end{bmatrix} = \begin{bmatrix} x^n \\ y^n \\ z^n \\ \dot{x}^n \\ \dot{y}^n \\ \dot{z}^n \\ \phi \\ \theta \\ \psi \\ b_{f,x} \\ b_{f,y} \\ b_{f,z} \\ b_{w,x} \\ b_{w,y} \\ b_{w,z} \end{bmatrix} \qquad \mathbf{u} = \begin{bmatrix} \mathbf{f}_u^b \\ \boldsymbol{w}_u^b \end{bmatrix} = \begin{bmatrix} f_{u,x}^b \\ f_{u,y}^b \\ f_{u,z}^b \\ w_{u,x}^b \\ w_{u,y}^b \\ w_{u,z}^b \end{bmatrix} \tag{15}$$

If we could perfectly measure the accelerations and angular rates of the system in the body frame ($\mathbf{a}^b$ and $\boldsymbol{w}^b$), the state evolution model would be:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{r}}^n \\ \dot{\mathbf{v}}^n \\ \dot{\mathbf{E}} \end{bmatrix} = \begin{bmatrix} \mathbf{v}^n \\ {}^n\mathbf{R}^b \mathbf{a}^b \\ {}^E\mathbf{Q}^b \boldsymbol{w}^b \end{bmatrix} \tag{16}$$

The accelerometers measure the specific forces $\mathbf{f}$ which do not include the gravity acceleration, i.e. without error:

$$\mathbf{a}^b = \mathbf{f}^b + \mathbf{g}^b \tag{17}$$

Then:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{r}}^n \\ \dot{\mathbf{v}}^n \\ \dot{\mathbf{E}} \end{bmatrix} = \begin{bmatrix} \mathbf{v}^n \\ {}^n\mathbf{R}^b \mathbf{f}^b + \mathbf{g}^n \\ {}^E\mathbf{Q}^b \boldsymbol{w}^b \end{bmatrix} \tag{18}$$

where the gravity vector is $\mathbf{g}^n = \begin{bmatrix} 0 & 0 & 9.81 \end{bmatrix}$ in the NED coordinate frame. The sensors are not perfect, they have biases and random noise. Our sensor model is:

$$\hat{\mathbf{f}}^b = \mathbf{f}^b + \mathbf{b}_f + \boldsymbol{\nu}_f \tag{19}$$

$$\hat{\mathbf{w}}^b = \mathbf{w}^b + \mathbf{b}_w + \boldsymbol{\nu}_w \tag{20}$$

where $\mathbf{b}_f$ and $\mathbf{b}_w$ are sensor biases that vary over time, and $\boldsymbol{\nu}_f$ and $\boldsymbol{\nu}_w$ are white random noise. Note that $\hat{\mathbf{f}}^b$ and $\hat{\mathbf{w}}^b$ are the calibrated measurements where the constant biases and the gravity scale factor are removed (see the IMU calibration in II). Substituting the sensor model (19), (20) into (18):

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{r}}^n \\ \dot{\mathbf{v}}^n \\ \dot{\mathbf{E}} \end{bmatrix} = \begin{bmatrix} \mathbf{v}^n \\ {}^n\mathbf{R}^b \left(\hat{\mathbf{f}}^b - \mathbf{b}_{f,0} - \mathbf{b}_f - \boldsymbol{\nu}_f \right) + \mathbf{g}^n \\ {}^E\mathbf{Q}^b \left(\hat{\mathbf{w}}^b - \mathbf{b}_{w,0} - \mathbf{b}_w - \boldsymbol{\nu}_w \right) \end{bmatrix} \tag{21}$$

The random biases are modeled as first order linear Gaussian processes. For example, for the x component of the specific force bias:

$$\dot{b}_{f,x} = -\tau_{f,x} b_{f,x} + \eta_{f,x} \qquad \text{where} \qquad \eta_{f,x} \sim \mathcal{N}\left(0, s_{f,x}\right) \tag{22}$$

and $s_{f,x}$ is the Power Spectral Density (PSD) of the Gaussian noise specified by the manufacturer. The time constant $\tau_{f,x}$ is also specified by the manufacturer ($\tau_{f,x} \approx 3600$s in the code). The same model applies for the six biases. This random biases are included in the state vector to finally obtain the continuous time nonlinear state model:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{r}}^n \\ \dot{\mathbf{v}}^n \\ \dot{\mathbf{E}} \\ \dot{\mathbf{b}}_f \\ \dot{\mathbf{b}}_w \end{bmatrix} = \begin{bmatrix} \mathbf{v}^n \\ {}^n\mathbf{R}^b \left(\hat{\mathbf{f}}^b - \mathbf{b}_{f,0} - \mathbf{b}_f - \boldsymbol{\nu}_f \right) + \mathbf{g}^n \\ {}^E\mathbf{Q}^b \left(\hat{\mathbf{w}}^b - \mathbf{b}_{w,0} - \mathbf{b}_w - \boldsymbol{\nu}_w \right) \\ -\boldsymbol{\tau}_f \mathbf{b}_f + \boldsymbol{\eta}_f \\ -\boldsymbol{\tau}_w \mathbf{b}_w + \boldsymbol{\eta}_w \end{bmatrix} = f\left(\mathbf{x}, \hat{\mathbf{u}}, \boldsymbol{\nu}, \boldsymbol{\eta}\right) \tag{23}$$

where $\hat{\mathbf{u}}$ is not a random variable, but an observed value and the white random noises are:

$$\boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{\nu}_f \\ \boldsymbol{\nu}_w \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{\eta}_f \\ \boldsymbol{\eta}_w \end{bmatrix} \tag{24}$$

Note that the dependencies of ${}^n\mathbf{R}^b$ and ${}^E\mathbf{Q}^b$ on the current attitude $\mathbf{E}$ are not made explicit in (23)

The discretization of (23) is:

$$\frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{\Delta T} \approx \dot{\mathbf{x}} = f\left(\mathbf{x}_k, \hat{\mathbf{u}}_k, \boldsymbol{\nu}_k, \boldsymbol{\eta}_k\right) \tag{25}$$

such that:

$$\mathbf{x}_{k+1} \approx \mathbf{x}_k + \Delta T f\left(\mathbf{x}_k, \hat{\mathbf{u}}_k, \boldsymbol{\nu}_k, \boldsymbol{\eta}_k\right) \tag{26}$$

Thus, taking the expected value of this expression, the state is propagated as:

$$\mathbb{E}\left[\mathbf{x}_{k+1}\right] = \mathbb{E}\left[\mathbf{x}_k\right] + \mathbb{E}\left[f\left(\mathbf{x}_k, \hat{\mathbf{u}}_k, \boldsymbol{\nu}_k, \boldsymbol{\eta}_k\right)\right] \tag{27}$$

Since the expected values of the noises are zeros and the expected values of the state is the prediction denoted with $\bar{\mathbf{x}}$, then:

$$\bar{\mathbf{x}}_{k+1} \approx \bar{\mathbf{x}}_k + \Delta T f\left(\bar{\mathbf{x}}_k, \hat{\mathbf{u}}_k, \mathbf{0}, \mathbf{0}\right) \tag{28}$$

Equation (28) is used to propagate the state, the following sections, linearize and discretize the model in (23) to propagate the uncertainty in the system.

IV. Linearization

From the state evolution model in (23), we must obtain a discretized model for the state mean propagation and a linear discrete model for the error propagation. We have two options: 1) linearize first and then discretize, or 2) discretize and then linearize.

Linearize continuous model

Using a first order Taylor expansion on the function $f(\mathbf{x}, \hat{\mathbf{u}}, \boldsymbol{\nu}, \boldsymbol{\eta})$:

$$\begin{aligned} f(\mathbf{x}, \hat{\mathbf{u}}, \boldsymbol{\nu}, \boldsymbol{\eta}) &\approx f(\mathbf{x}^*, \hat{\mathbf{u}}^*, \boldsymbol{\nu}^*, \boldsymbol{\eta}^*) + \left.\frac{\partial f}{\partial \mathbf{x}}\right|_* (\mathbf{x} - \mathbf{x}^*) + \left.\frac{\partial f}{\partial \hat{\mathbf{u}}}\right|_* (\hat{\mathbf{u}} - \hat{\mathbf{u}}^*) + \left.\frac{\partial f}{\partial \boldsymbol{\nu}}\right|_* (\boldsymbol{\nu} - \boldsymbol{\nu}^*) + \left.\frac{\partial f}{\partial \boldsymbol{\eta}}\right|_* (\boldsymbol{\eta} - \boldsymbol{\eta}^*) \\ &\approx f(\mathbf{x}^*, \hat{\mathbf{u}}^*, \boldsymbol{\nu}^*, \boldsymbol{\eta}^*) + \left.\frac{\partial f}{\partial \mathbf{x}}\right|_* (\mathbf{x} - \mathbf{x}^*) + \left.\frac{\partial f}{\partial \boldsymbol{\nu}}\right|_* (\boldsymbol{\nu} - \boldsymbol{\nu}^*) + \left.\frac{\partial f}{\partial \boldsymbol{\eta}}\right|_* (\boldsymbol{\eta} - \boldsymbol{\eta}^*) \end{aligned} \tag{29}$$

where the linearization around $\hat{\mathbf{u}}^*$ is nonsense if we choose $\hat{\mathbf{u}}$ as the linearization point, which makes sense since $\hat{\mathbf{u}}$ is known (it is not a random variable). The linearized state evolution in error terms:

$$\begin{gathered} \underbrace{f(\mathbf{x}, \hat{\mathbf{u}}, \boldsymbol{\nu}, \boldsymbol{\eta}) - f(\mathbf{x}^*, \hat{\mathbf{u}}, \boldsymbol{\nu}^*, \boldsymbol{\eta}^*)}_{\Delta\dot{\mathbf{x}}} \approx \left.\frac{\partial f}{\partial \mathbf{x}}\right|_* \underbrace{(\mathbf{x} - \mathbf{x}^*)}_{\Delta\mathbf{x}} + \left.\frac{\partial f}{\partial \boldsymbol{\nu}}\right|_* \underbrace{(\boldsymbol{\nu} - \boldsymbol{\nu}^*)}_{\Delta\boldsymbol{\nu}} + \left.\frac{\partial f}{\partial \boldsymbol{\eta}}\right|_* \underbrace{(\boldsymbol{\eta} - \boldsymbol{\eta}^*)}_{\Delta\boldsymbol{\eta}} \\ \Delta\dot{\mathbf{x}} \approx \mathbf{F}\Delta\mathbf{x} + \mathbf{G}_\nu \Delta\boldsymbol{\nu} + \mathbf{G}_\eta \Delta\boldsymbol{\eta} = \mathbf{F}\Delta\mathbf{x} + \mathbf{G}\Delta\boldsymbol{\delta} \end{gathered} \tag{30}$$

where the partial derivative (or Jacobian) matrices are indicated by $\mathbf{F}$, $\mathbf{G}_\nu$ and $\mathbf{G}_\eta$ and

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_\nu & \mathbf{G}_\eta \end{bmatrix} \qquad \text{and} \qquad \Delta\boldsymbol{\delta} = \begin{bmatrix} \Delta\boldsymbol{\nu} \\ \Delta\boldsymbol{\eta} \end{bmatrix} \tag{31}$$

Discretize linearized model

Once the model is linearized, we can apply standard techniques to discretize it. The discrete model is:

$$\Delta\mathbf{x}_{k+1} = \boldsymbol{\Phi}_k \Delta\mathbf{x}_k + \boldsymbol{\Gamma}_{k,\nu} \Delta\boldsymbol{\nu}_k + \boldsymbol{\Gamma}_{k,\eta} \Delta\boldsymbol{\eta}_k \tag{32}$$

where:

$$\begin{aligned} \Delta\mathbf{x}_k &= \mathbf{x}_k - \mathbf{x}_k^* = \mathbf{x}_k - \hat{\mathbf{x}}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_k) \\ \Delta\boldsymbol{\nu}_k &= \boldsymbol{\nu}_k - \boldsymbol{\nu}_k^* = \boldsymbol{\nu}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{V}_k) \\ \Delta\boldsymbol{\eta}_k &= \boldsymbol{\eta}_k - \boldsymbol{\eta}_k^* = \boldsymbol{\eta}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{N}_k) \end{aligned} \tag{33}$$

This model is only employed to propagate the covariance matrix of the state which is:

$$\mathbf{P}_{k+1} = \boldsymbol{\Phi}_k \mathbf{P}_k \boldsymbol{\Phi}_k^T + \underbrace{\boldsymbol{\Gamma}_{k,\nu} \mathbf{V}_k \boldsymbol{\Gamma}_{k,\nu}^T}_{\bar{\mathbf{V}}_k} + \underbrace{\boldsymbol{\Gamma}_{k,\eta} \mathbf{N}_k \boldsymbol{\Gamma}_{k,\eta}^T}_{\bar{\mathbf{N}}_k} \tag{34}$$

For the simplified model in (31):

$$\Delta\mathbf{x}_{k+1} = \boldsymbol{\Phi}_k \Delta\mathbf{x}_k + \boldsymbol{\Gamma}_k \Delta\boldsymbol{\delta}_k \tag{35}$$

where:

$$\Delta\boldsymbol{\delta}_k = \boldsymbol{\delta}_k - \boldsymbol{\delta}_k^* = \boldsymbol{\delta}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{D}_k) \quad \text{and} \quad \mathbf{D}_k = \begin{bmatrix} \mathbf{V}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{N}_k \end{bmatrix} \tag{36}$$

The uncertainty propagation for this model is:

$$\mathbf{P}_{k+1} = \boldsymbol{\Phi}_k \mathbf{P}_k \boldsymbol{\Phi}_k^T + \underbrace{\boldsymbol{\Gamma}_k \mathbf{V}_k \boldsymbol{\Gamma}_k^T}_{\bar{\mathbf{D}}_k} \tag{37}$$

The state evolution matrix is obtained as:

$$\mathbf{\Phi}_k = \mathbf{e}^{\mathbf{F}T} \tag{38}$$

and the matrix $\bar{\mathbf{D}}_k$ is obtained following the next procedure:

1. Build $C = \begin{bmatrix} -\mathbf{F} & \mathbf{GSG}^T \\ \mathbf{0} & \mathbf{F}^T \end{bmatrix}$ where $\mathbf{S}$ is the Power Spectral Density (PSD) matrix of the white noise as $\mathbf{S} = \begin{bmatrix} \mathbf{S}_\nu & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_\eta \end{bmatrix}$. The PSD of the biases noise is given by the manufacturer and the PSD of the sensor noise can be obtained from the measure variance.
2. Compute $\mathbf{e}^{\mathbf{C}T} = \begin{bmatrix} \mathbf{F}_2 & \mathbf{G}_2 \\ \mathbf{0} & \mathbf{F}_3 \end{bmatrix}$
3. Finally $\bar{\mathbf{D}}_k = \mathbf{F}_3^T \mathbf{G}_2$

The PSD deserve a second look. The two PSD matrices are:

$$\mathbf{S}_\nu = \begin{bmatrix} \mathbf{S}_{\nu_f} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{\nu_w} \end{bmatrix} \qquad \text{and} \qquad \mathbf{S}_\eta = \begin{bmatrix} \mathbf{S}_{\eta_f} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{\eta_w} \end{bmatrix} \tag{39}$$

where:

$$\mathbf{S}_{\nu_f} = \begin{bmatrix} s^x_{\nu_f} & 0 & 0 \\ 0 & s^y_{\nu_f} & 0 \\ 0 & 0 & s^z_{\nu_f} \end{bmatrix} \quad \mathbf{S}_{\nu_w} = \begin{bmatrix} s^x_{\nu_w} & 0 & 0 \\ 0 & s^y_{\nu_w} & 0 \\ 0 & 0 & s^z_{\nu_w} \end{bmatrix}$$

$$\mathbf{S}_{\eta_f} = \begin{bmatrix} s^x_{\eta_f} & 0 & 0 \\ 0 & s^y_{\eta_f} & 0 \\ 0 & 0 & s^z_{\eta_f} \end{bmatrix} \quad \mathbf{S}_{\eta_w} = \begin{bmatrix} s^x_{\eta_w} & 0 & 0 \\ 0 & s^y_{\eta_w} & 0 \\ 0 & 0 & s^z_{\eta_w} \end{bmatrix} \tag{40}$$

The PSD of the ν components are related to the sensor noise. They can be obtained by first measuring the sensor white noise variances empirically to obtain:

$$\mathbf{V} = \text{diagonal}\left(\left[\left(\sigma^x_{\nu_f}\right)^2 \quad \left(\sigma^y_{\nu_f}\right)^2 \quad \left(\sigma^z_{\nu_f}\right)^2 \quad \left(\sigma^x_{\nu_w}\right)^2 \quad \left(\sigma^y_{\nu_w}\right)^2 \quad \left(\sigma^z_{\nu_w}\right)^2\right]\right) \tag{41}$$

Then the PSD can be approximated by:

$$\mathbf{S}_\nu = \mathbf{V} T_{IMU} \tag{42}$$

The PSD of the bias white random noise must be specified by the manufacturer.

II. Calibration

In this section the IMU is calibrated to obtain the static biases and the gravity scale factor to compensate the gravity value used by the manufacturer not coinciding with the value at our location. The goal of this section is obtaining the $\hat{\mathbf{f}}^b$ and $\hat{\mathbf{w}}^b$ values used in the measurement model (19) (20). The uncalibrated IMU readings are denote with a superindex $\check{}$ and the true values without any superindex. Then, the IMU accelerometers model, ignoring the b frame superindex, is:

$$\begin{aligned}
\check{f}_x &= c_x \left(f_x + b_{x_{f,0}} + b_{x_f} + \nu_{x_f} \right) \\
\check{f}_y &= c_y \left(f_y + b_{y_{f,0}} + b_{y_f} + \nu_{y_f} \right) \\
\check{f}_z &= c_z \left(f_z + b_{z_{f,0}} + b_{z_f} + \nu_{z_f} \right)
\end{aligned} \tag{43}$$

Assuming a flat surface, we gather data such that the acceleration components of the corresponding axis are zero, then for the x axis:

$$\check{f}_x = c_x \left(b_{x_{f,0}} + b_{x_f} + \nu_{x_f} \right) \tag{44}$$

Taking the mean of a lot of data points and assuming that the random moving bias $\mathbf{b}_{x_f}$ does not vary during this time (approximately true because it moves very slowly), we can measure:

$$\begin{aligned}
c_x b_{x_{f,0}} &= \text{mean}\left(\check{f}_x\right) \triangleq \mu_x \quad \text{and similarly} \\
c_y b_{y_{f,0}} &= \text{mean}\left(\check{f}_y\right) \triangleq \mu_y \\
c_z b_{z_{f,0}} &= \text{mean}\left(\check{f}_z\right) \triangleq \mu_z
\end{aligned} \tag{45}$$

Then, we can align each axis with the gravity direction such that we should be measuring the gravity acceleration exactly, which we know is $g = 9.80279\text{m}/\text{s}^2$ at our location (at IIT). For the x axis again:

$$\check{f}_x = c_x \left(g + b_{x_{f,0}} + b_{x_f} + \nu_{x_f} \right) \tag{46}$$

Taking the mean of the values and assuming again a zero-mean moving bias for this dataset:

$$\hat{g}_x \triangleq \text{mean}\left(\check{f}_x\right) = c_x g + \underbrace{c_x b_{x_{f,0}}}_{\mu_x} \tag{47}$$

and using the measured value of $c_x b_{x_{f,0}}$ from (45):

$$\begin{aligned}
c_x &= \frac{1}{g} \left(\hat{g}_x - \mu_x \right) \quad \text{and similarly} \\
c_y &= \frac{1}{g} \left(\hat{g}_y - \mu_y \right) \\
c_z &= \frac{1}{g} \left(\hat{g}_z - \mu_z \right)
\end{aligned} \tag{48}$$

The static biases can then be calculated from (45) as:

$$\begin{aligned}
b_{x_{f,0}} &= \frac{\mu_x}{c_x} \\
b_{y_{f,0}} &= \frac{\mu_y}{c_y} \\
b_{z_{f,0}} &= \frac{\mu_z}{c_z}
\end{aligned} \tag{49}$$

Finally, from (43):

$$f_x = \underbrace{\frac{\breve{f}_x}{c_x} - b_{x_{f,0}}}_{\hat{f}_x} - b_{x_f} - \nu_{x_f} \tag{50}$$

where $\hat{f}_x$ is identified from (19). The readings obtained from the IMU will be "corrected" as follows:

$$\begin{bmatrix} \hat{f}_x \\ \hat{f}_y \\ \hat{f}_z \end{bmatrix} = \underbrace{\begin{bmatrix} c_x^{-1} & 0 & 0 \\ 0 & c_y^{-1} & 0 \\ 0 & 0 & c_z^{-1} \end{bmatrix}}_{\mathbf{C}^{-1}} \begin{bmatrix} \breve{f}_x \\ \breve{f}_y \\ \breve{f}_z \end{bmatrix} - \underbrace{\begin{bmatrix} b_{x_{f,0}} \\ b_{y_{f,0}} \\ b_{z_{f,0}} \end{bmatrix}}_{\mathbf{b}_{f,0}} \tag{51}$$

or, in matrix notation:

$$\hat{\mathbf{f}} = \mathbf{C}^{-1}\breve{\mathbf{f}} - \mathbf{b}_{f,0} \tag{52}$$

References