Chapter 1

Aided Inertial Navigation System

In this chapter, the background and notation for a Global Positioning System (GPS) aided inertial navigation system (INS) [1] is reviewed. To make this book self-contained, this chapter first reviews the INS notation and mechanization. To improve the accuracy and reliability of the INS, aiding is performed using an optimal estimator. It is common to perform optimal estimation using an EKF framework, thus an introduction to the EKF is presented in Section 1.7. Additional accuracy and reliability is achieved through the nonlinear MAP optimization based INS, which is presented in Chapter **??**.

1.1 Aided Inertial Navigation

Let $\boldsymbol{x} \in \mathbb{R}^{n_s}$ denote the rover state vector, where

$$\boldsymbol{x}(t) = [\mathbf{p}^\intercal(t), \mathbf{v}^\intercal(t), \mathbf{q}^\intercal(t), \mathbf{b}_a^\intercal(t), \mathbf{b}_g^\intercal(t)]^\intercal \in \mathbb{R}^{n_s},$$

where $\mathbf{p}$, $\mathbf{v}$, $\mathbf{b}_a$, $\mathbf{b}_g$ each in $\mathbb{R}^3$ represent the position, velocity, accelerometer bias and gyro bias vectors, respectively, and $\mathbf{q} \in \mathbb{R}^4$ represents the attitude quaternion ($n_s = 16$), each at time t.

The kinematic equations for the rover state are

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \mathbf{u}(t)), \tag{1.1}$$

where $\boldsymbol{f} : \mathbb{R}^{n_s} \times \mathbb{R}^6 \mapsto \mathbb{R}^{n_s}$ represents the kinematic model, and $\mathbf{u} \in \mathbb{R}^6$ is the vector of specific forces and angular rates. The function $\boldsymbol{f}$ is accurately known (see eqns. 11.31-11.33 in [1], derivations specific to this book are provided in Section 1.3). The user applies forces and torques causing accelerations and angular rates which determine $\mathbf{u}(t)$. The kinematic integration of $\mathbf{u}(t)$ through eqn. (1.1) determines $\boldsymbol{x}(t)$.

Let τ_i denote the time instants of the i^{th} IMU measurements of $\mathbf{u}$. The IMU measurements are modeled as

$$\tilde{\mathbf{u}}(\tau_i) = \mathbf{u}(\tau_i) - \mathbf{b}(\tau_i) - \boldsymbol{\omega}_u(\tau_i), \tag{1.2}$$

with stochastic errors $\boldsymbol{\omega}_u(\tau_i) \sim \mathcal{N}(\mathbf{0}, \mathbf{Qd})$ and $\mathbf{b} = [\mathbf{b}_a^\intercal, \mathbf{b}_g^\intercal]^\intercal$.

Given the initial condition $\boldsymbol{x}(t_0) \sim \mathcal{N}(\boldsymbol{x}_0, \mathbf{P}_0)$ and measurements $\tilde{\mathbf{u}}$, an inertial navigation system propagates an estimate of the vehicle state as the solution of

$$\dot{\hat{\boldsymbol{x}}}(t) = \boldsymbol{f}(\hat{\boldsymbol{x}}(t), \tilde{\mathbf{u}}(t)), \tag{1.3}$$

where $\hat{\boldsymbol{x}}(t)$ denotes the real-time estimate of $\boldsymbol{x}(t)$, and $\hat{\boldsymbol{x}}(0) = x_0$.

Let $\boldsymbol{x}_i$ and $\mathbf{u}_i$ denote $\boldsymbol{x}(\tau_i)$ and $\mathbf{u}(\tau_i)$. The solution of eqn. (1.3) over the interval $t \in [\tau_{i-1}, \tau_i]$ from the initial condition $\hat{\boldsymbol{x}}_{i-1}$ is $\hat{\boldsymbol{x}}_i \doteq \phi(\hat{\boldsymbol{x}}_{i-1}, \hat{\mathbf{u}}_{i-1})$, where $\hat{\mathbf{u}}_{i-1} \doteq \tilde{\mathbf{u}}_{i-1} - \hat{\mathbf{b}}_{i-1}$ and

$$\phi(\boldsymbol{x}_{i-1}, \mathbf{u}_{i-1}) = \boldsymbol{x}_{i-1} + \int_{\tau_{i-1}}^{\tau_i} \boldsymbol{f}(\boldsymbol{x}(\tau), \mathbf{u}(\tau))d\tau. \tag{1.4}$$

In this book, the equality symbol '$=$' will be used in its normal sense. The symbol '$\doteq$' will be used to indicate computations that are implemented in software.

Define $\mathbf{U}_{k-1} = \{\tilde{\mathbf{u}}(\tau_i) \text{ for } \tau_i \in [t_{k-1}, t_k]\}$. The integral operator in (1.4) can be iterated for all IMU measurements in $\mathbf{U}_{k-1}$ to propagate the state from t_{k-1} to t_k. Denote this iterative application of eqn. (1.4) as $\hat{\boldsymbol{x}}_k \doteq \boldsymbol{\Phi}(\hat{\boldsymbol{x}}_{k-1}, \mathbf{U}_{k-1})$.

1.2 GPS/DGPS Aiding

Let $t_k = kT$ denote the time instants at which GPS measurements are valid, and $\boldsymbol{x}_k$ denote the state $\boldsymbol{x}(kT)$ at t_k. It is typically the case that there are numerous IMU measurements available between GPS epochs: $T \gg [\tau_i - \tau_{i-1}]$.

For $(m+1)$ satellites, $\mathbf{y}_k$ represents the double-differenced code (pseudorange) and Doppler measurement vector, as defined in Section 8.8 of [1]. For notational simplicity, it is assumed that the double difference approach removes all common-mode errors (e.g., ionosphere, troposphere, satellite clock and ephemeris errors), as well as the receiver clock biases. The double-differenced measurement vector at t_k is modeled as

$$\mathbf{y}_k = \mathbf{h}_k(\boldsymbol{x}_k) + \boldsymbol{\eta}_{y,k} + \mathbf{s}_k \tag{1.5}$$

where $\boldsymbol{\eta}_{y,k} = [\boldsymbol{\eta}_{\rho,k}, \boldsymbol{\eta}_{d,k}]$, and $\mathbf{y}_k,\ \boldsymbol{\eta}_{y,k} \in \mathbb{R}^{2m}$. The symbol $\boldsymbol{\eta}_{\rho,k} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{\rho,k})$ represents the pseudorange measurement noise, and $\boldsymbol{\eta}_{d,k} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{d,k})$ represents the Doppler measurement noise. Depending on receiver design, environmental factors and the performance of multipath mitigation techniques, the noise level $\mathbf{R}_\rho$ and $\mathbf{R}_d$ can vary for each available satellite. The measurement noise covariance $\mathbf{R} = \text{blkdiag}(\mathbf{R}_\rho, \mathbf{R}_d) \in \mathbb{R}^{2m \times 2m}$.

The symbol $\mathbf{s}_k = [\mathbf{s}_{\rho,k}^\intercal, \mathbf{s}_{d,k}^\intercal]^\intercal \in \mathbb{R}^{2m}$ represents the error due to outliers, where $\mathbf{s}_{\rho,k} = [\mathbf{s}_{\rho,1}, \ldots, \mathbf{s}_{\rho,m}]^\intercal$ and $\mathbf{s}_{d,k} = [\mathbf{s}_{d,1}, \ldots, \mathbf{s}_{d,m}]^\intercal$. Throughout this book, the ability to accommodate outliers will be considered from different perspectives. In Chapter **??**, the estimation approach is presented in the ideal case where $\mathbf{s}_k = \mathbf{0}$. In Chapter **??**, the ability to detect and remove outliers is considered from the hypothesis testing point-of-view. Alternatively, in Chapter **??**, the outliers will be directly accommodated from the Least Soft-thresholded Squares (LSS) perspective.

Using the state estimate, the GPS measurements at t_k are predicted to be $\hat{\mathbf{y}}_k \doteq \mathbf{h}_k(\hat{\boldsymbol{x}}_k)$. The GPS measurement residual vector is computed as $\delta\mathbf{y}_k \doteq \mathbf{y}_k - \hat{\mathbf{y}}_k$.

1.3 INS Temporal Propagation

1.3.1 Problem Formulation

While many reference frames may be used, the Earth-centered Earth-fixed (ECEF) reference frame is a convenient reference frame for GPS-aided INS. One reason for this choice is that satellite navigation solutions are resolved in the ECEF reference frame. Another is that the ECEF frame works globally, not yielding singularities in the polar regions.

Define the e-frame as the ECEF-frame (or Earth-frame), and the i-frame as the Earth-Centered-Inertial (ECI) frame (or inertial-frame). The direction cosine matrix (DCM), or rotation matrix, from body-frame to Earth-frame is $\mathbf{R}_b^e$. The position of the body b-frame with respect to the Earth e-frame resolved in the e-frame is $\mathbf{r}_{eb}^e$. Similarly the velocity of the b-frame with respect to the e-frame resolved in the e-frame is $\mathbf{v}_{eb}^e$. The rotation matrix from b-frame to e-frame is $\mathbf{R}_b^e$.

Expanding eqn. (1.1), the INS kinematic equations defining $\boldsymbol{f}(\boldsymbol{x}(t), \mathbf{u}(t))$ in the ECEF frame (see Section 11.2.2 of [1]) are

$$\dot{\mathbf{r}}_{eb}^e = \mathbf{v}_{eb}^e \tag{1.6}$$

$$\dot{\mathbf{v}}_{eb}^e = \mathbf{R}_b^e \mathbf{f}_{ib}^b + \mathbf{g}_b^e - 2\boldsymbol{\Omega}_{ie}^e \mathbf{v}_{eb}^e \tag{1.7}$$

$$\dot{\mathbf{R}}_b^e = \mathbf{R}_b^e(\boldsymbol{\Omega}_{ib}^e - \boldsymbol{\Omega}_{ie}^e). \tag{1.8}$$

The inputs $\mathbf{u}$, are specific force $\mathbf{f}_{ib}^b$ and angular rate $[\boldsymbol{\omega}_{ib}^b \times] = \boldsymbol{\Omega}_{ib}^b$, with respect to the inertial-frame i. The local gravity vector is $\mathbf{g}^e$, and the

Earth-rotation rate is $[\boldsymbol{\omega}^e_{ie}\times] = \boldsymbol{\Omega}^e_{ie}$. The notation $[\mathbf{a}\times]$ represents the skew-symmetric matrix corresponding to the vector $\mathbf{a}$.

Fig. 1.1 is a block diagram showing how the angular-rate and specific-force measurements, of an Inertial Measurement Unit (IMU), are used to update the Earth-referenced attitude, velocity, and position states at t_i.

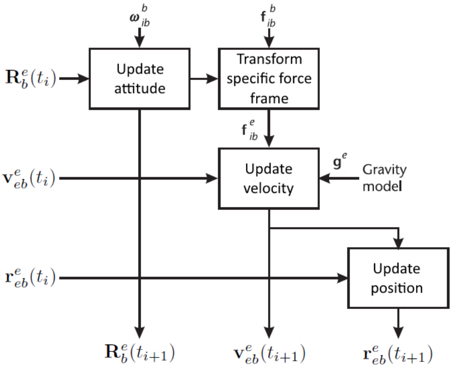

Figure 1.1: Block diagram of the ECEF referenced INS equations.

1.3.2 INS Time Propagation Equations

Attitude Update

Using eqn. (2.54) in [1], the time derivative of the rotation matrix $\mathbf{R}^e_b$ is

$$\dot{\mathbf{R}}^e_b = \mathbf{R}^e_b \boldsymbol{\Omega}^b_{eb}, \tag{1.9}$$

where $\boldsymbol{\Omega}^b_{eb} = [\boldsymbol{\omega}^b_{eb}\times]$.

Let the attitude increment over the IMU measurement interval τ_i be defined as $\boldsymbol{\alpha}^b_{ib} = \boldsymbol{\omega}^b_{ib}\tau_i$. Appendix **??** shows that the discrete-time rotation matrix update that is equivalent to eqn. (1.9) can be computed as

$$\mathbf{R}^e_b(t_{i+1}) = \mathbf{R}^e_b(t_i)\mathbf{R}^{b(t+\tau_i)}_{b(t)} - \boldsymbol{\Omega}^b_{ie}\mathbf{R}^e_b(t_i)\tau_i + \mathbf{R}^e_b(t_i). \tag{1.10}$$

The integrating factor $\mathbf{R}^{b(t+\tau_i)}_{b(t)}$ represents the effect of the angle rotation corresponding to $\boldsymbol{\Omega}^b_{ib}$ over the interval $[t, t+\tau_i]$, and $\mathbf{R}^e_b(t_i)$ and $\mathbf{R}^e_b(t_{i+1})$ are the prior and updated rotation matrix, respectively.

Attitude Increment

The IMU angular rate has two components $\boldsymbol{\omega}^b_{ib}$ and $\boldsymbol{\omega}^b_{ie}$. The body rate portion $\boldsymbol{\omega}^b_{ib}$ can be large and change rapidly, which is the reason its portion of the attitude increment receives special treatment.

By eqn. (2.62) of [1],

$$\mathbf{R}_{b(t)}^{b(t+\tau_i)} = \mathbf{R}_{b-}^{b+} = \exp[\boldsymbol{\alpha}_{ib}^{b}\times]. \tag{1.11}$$

The power-series expansion of the matrix exponential is

$$\exp[\boldsymbol{\alpha}_{ib}^{b}\times] = \sum_{r=0}^{\infty} \frac{[\boldsymbol{\alpha}_{ib}^{b}\times]^r}{r!}. \tag{1.12}$$

As shown in eqns. (2.66) and (2.67) of [1], the odd and even powers of the attitude increment skew-symmetric matrix are,

$$\begin{aligned} [\boldsymbol{\alpha}_{ib}^{b}\times]^{2r+1} &= (-1)^r ||\boldsymbol{\alpha}_{ib}^{b}||^{r} [\boldsymbol{\alpha}_{ib}^{b}\times] & (1.13)\\ [\boldsymbol{\alpha}_{ib}^{b}\times]^{2r} &= (-1)^r ||\boldsymbol{\alpha}_{ib}^{b}||^{2r} [\boldsymbol{\alpha}_{ib}^{b}\times]^2, & (1.14) \end{aligned}$$

where $r = 1, 2, 3, \cdots$. Expanding eqn. (1.11) using eqns. (1.12)-(1.14),

$$\begin{aligned} \mathbf{R}_{b-}^{b+} = \mathbf{I}_3 &+ \left(\sum_{r=0}^{\infty} (-1)^r \frac{||\boldsymbol{\alpha}_{ib}^{b}||^{2r}}{(2r+1)!} \right) [\boldsymbol{\alpha}_{ib}^{b}\times] \\ &+ \left(\sum_{r=0}^{\infty} (-1)^r \frac{||\boldsymbol{\alpha}_{ib}^{b}||^{2r}}{(2r+2)!} \right) [\boldsymbol{\alpha}_{ib}^{b}\times]^2. \end{aligned} \tag{1.15}$$

Eqn. (1.15) is equivalent to (see eqn. (2.69) of [1], and [2, 3]),

$$\mathbf{R}_{b-}^{b+} = \mathbf{I}_3 + \frac{\sin(||\boldsymbol{\alpha}_{ib}^{b}||)}{||\boldsymbol{\alpha}_{ib}^{b}||} [\boldsymbol{\alpha}_{ib}^{b}\times] + \frac{1 - \cos(||\boldsymbol{\alpha}_{ib}^{b}||)}{||\boldsymbol{\alpha}_{ib}^{b}||^2} [\boldsymbol{\alpha}_{ib}^{b}\times]^2. \tag{1.16}$$

When the CPU requires tradeoffs related to implementation, computation of trigonometric functions can be prohibitive. In these cases, the Taylor series expansions may be truncated. For example, the fourth-order approximation of eqn. (1.15) is

$$\mathbf{R}_{b-}^{b+} = \mathbf{I}_3 + \left(1 - \frac{||\boldsymbol{\alpha}_{ib}^{b}||^2}{6}\right) [\boldsymbol{\alpha}_{ib}^{b}\times] + \left(\frac{1}{2} - \frac{||\boldsymbol{\alpha}_{ib}^{b}||^2}{24}\right) [\boldsymbol{\alpha}_{ib}^{b}\times]^2. \tag{1.17}$$

Such tradeoffs should be thoroughly evaluated in simulation.

Velocity Update

Neglecting acceleration in the ECEF frame, assume the ECI frame instantaneously coincides with the ECEF frame, such that $\ddot{\mathbf{r}}_{ib}^{e} = \ddot{\mathbf{r}}_{eb}^{e}$, $\dot{\mathbf{r}}_{ib}^{e} = \dot{\mathbf{r}}_{eb}^{e}$, and $\mathbf{r}_{ib}^{e} = \mathbf{r}_{eb}^{e}$. The velocity update in the ECEF frame is

$$\mathbf{v}_{eb}^{e}(t_{i+1}) = \mathbf{v}_{eb}^{e}(t_i) + \left(\mathbf{f}_{ib}^{e} + \mathbf{g}_{b}^{e}(\mathbf{r}_{eb}^{e}(t_i)) - 2\boldsymbol{\Omega}_{ie}^{e}\mathbf{v}_{eb}^{e}(t_i)\right)\tau_i \tag{1.18}$$

where $\mathbf{r}^e_{eb}(t_i)$ and $\mathbf{v}^e_{eb}(t_i)$ are the prior position and velocity, respectively, and $\mathbf{v}^e_{eb}(t_{i+1})$ is the updated velocity for the IMU interval τ_i.

Specific-Force Frame Transformation

The frame transformation of specific-force takes the form

$$\mathbf{f}^e_{ib} = \mathbf{R}^e_b \mathbf{f}^b_{ib}. \tag{1.19}$$

Incorporating the updated rotation matrix $\mathbf{R}^e_b(t_{i+1})$ from eqn. (1.10), the specific force transformation is

$$\mathbf{f}^e_{ib} = \mathbf{R}^e_b(t_{i+1})\mathbf{f}^b_{ib}. \tag{1.20}$$

Position Update

Using eqn. (1.6), the position update is derived as follows:

$$\mathbf{r}^e_{eb}(t_{i+1}) = \mathbf{r}^e_{eb}(t_i) + (\mathbf{v}^e_{eb}(t_i) + \mathbf{v}^e_{eb}(t_{i+1}))\frac{\tau_i}{2}, \tag{1.21}$$

$$= \mathbf{r}^e_{eb}(t_i) + \mathbf{v}^e_{eb}(t_i)\tau_i + (\mathbf{f}^e_{ib} + \mathbf{g}^e(\mathbf{r}^e_{eb}(t_i)) - 2\mathbf{\Omega}^e_{ie}\mathbf{v}^e_{eb}(t_i))\,\frac{\tau_i^2}{2}. \tag{1.22}$$

where $\mathbf{r}^e_{eb}(t_i)$ and $\mathbf{r}^e_{eb}(t_{i+1})$ are the prior and updated positions, respectively.

Gravity Model

A precise gravity model [4] formulated in the ECEF frame, is defined as

$$\boldsymbol{\gamma}^e_{ib} = -\frac{\mu}{||\mathbf{r}^e_{eb}||^3} \times \left\{ \mathbf{r}^e_{eb} + \frac{3}{2}J_2\frac{R_0^2}{||\mathbf{r}^e_{eb}||^2} \begin{bmatrix} (1-5(\mathbf{r}^e_{eb,z})^2/||\mathbf{r}^e_{eb}||)\mathbf{r}^e_{eb,x} \\ (1-5(\mathbf{r}^e_{eb,z})^2/||\mathbf{r}^e_{eb}||)\mathbf{r}^e_{eb,y} \\ (3-5(\mathbf{r}^e_{eb,z})^2/||\mathbf{r}^e_{eb}||)\mathbf{r}^e_{eb,z} \end{bmatrix} \right\}, \tag{1.23}$$

where the Earth's second gravitational constant $J_2 = 1.082627E^{-3}$ (m^3/s^2), the Equatorial radius is $R_0 = 6.378137E^{+6}$ (m), the gravitational constant $\mu = 3.986004418E^{+14}$ (m^3/s^2), and the Earth-rotation rate is $\omega_{ie} = 7.292115E^{-5}$ (rad/s).

1.4 State Correction

1.4.1 Problem Formulation

Let $\hat{\boldsymbol{x}} \in \mathbb{R}^{n_s}$ denote the estimate of the rover state vector:

$$\hat{\boldsymbol{x}}(t) = [\hat{\mathbf{p}}^\intercal(t), \hat{\mathbf{v}}^\intercal(t), \hat{\mathbf{q}}^\intercal(t), \hat{\mathbf{b}}^\intercal_a(t), \hat{\mathbf{b}}^\intercal_g(t)]^\intercal \in \mathbb{R}^{n_s}.$$

The error between $\boldsymbol{x}(t)$ and $\hat{\boldsymbol{x}}(t)$ is denoted as $\delta\boldsymbol{x}$. The error vector is

$$\delta\boldsymbol{x} = [\delta\mathbf{p}^\intercal, \delta\mathbf{v}^\intercal, \boldsymbol{\rho}^\intercal, \delta\mathbf{b}_a^\intercal, \delta\mathbf{b}_g^\intercal]^\intercal \in \mathbb{R}^{n_e},$$

where $\delta\mathbf{p}$ and $\delta\mathbf{v}$, each in $\mathbb{R}^3$, represent the error between the true and computed position and velocity, respectively. The small-angle error state, denoted as $\boldsymbol{\rho} \in \mathbb{R}^{3\times 1}$, is defined in Section 2.5.5 of [1], and discussed in Section 1.4.3. The errors $\delta\mathbf{b}_a$ and $\delta\mathbf{b}_g$, each in $\mathbb{R}^3$, represent the accelerometer bias, and gyro bias errors, respectively. Therefore $\delta\boldsymbol{x} \in \mathbb{R}^{15}$ (i.e. $n_e = 15$). The fact that $n_s = 16$ and $n_e = 15$ is discussed in Section 1.4.3.

Let $\delta\hat{\boldsymbol{x}}$ denote an estimate of $\delta\boldsymbol{x}$. The state correction to the state vector $\hat{\boldsymbol{x}}$ is denoted as

$$\hat{\boldsymbol{x}}^+ = \hat{\boldsymbol{x}}^- \oplus \delta\hat{\boldsymbol{x}}.$$

The symbol $(-)$ denotes the prior estimate, whereas $(+)$ is the updated estimate. The symbol $\oplus$ is discussed in Sections 1.4.2 and 1.4.3

1.4.2 Position, Velocity, and Bias Updates

Position, velocity, accelerometer bias and gyro bias, each have corrections which are additive. The state correction step is

$$\begin{aligned}
\hat{\mathbf{p}}^+ &= \hat{\mathbf{p}}^- + \delta\mathbf{p} \\
\hat{\mathbf{v}}^+ &= \hat{\mathbf{v}}^- + \delta\mathbf{v} \\
\hat{\mathbf{b}}_a^+ &= \hat{\mathbf{b}}_a^- + \delta\mathbf{b}_a \\
\hat{\mathbf{b}}_g^+ &= \hat{\mathbf{b}}_g^- + \delta\mathbf{b}_g.
\end{aligned}$$

1.4.3 Attitude Update

When the attitude error is sufficiently small (see Section 2.5.5 of [1]), the attitude can be represented as a set of small-angle planar rotations $\{\rho_x, \rho_y, \rho_z\}$ about three orthogonal axes $\{x, y, z\}$, thus the attitude error can be defined in $\mathbb{R}^3$.

Rotation Matrix

Let $\mathbf{R}_b^n \in \mathbb{R}^{3\times 3}$ represent the true rotation from body-frame (b) to navigation-frame (n) that is equivalent to $\mathbf{q}(t)$ (see eqn. D.13 in [1]). Let $\hat{\mathbf{R}}_b^n \in \mathbb{R}^{3\times 3}$ represent the computed rotation that is equivalent to $\hat{\mathbf{q}}(t)$. The error between the true and computed rotation is

$$\mathbf{R}_{\hat{n}}^n = (\mathbf{R}_b^n)(\hat{\mathbf{R}}_n^b),$$

where $\mathbf{R}_{\hat{n}}^n$ represents the rotation matrix from the computed to actual navigation frame. When the error between the true and computed rotation is zero, then $\mathbf{R}_{\hat{n}}^n = \mathbf{I}$. Otherwise, as discussed in Section 2.6.1 of [1],

$$\mathbf{R}_{\hat{n}}^n = [\mathbf{I} - \mathbf{P}]$$

where $\mathbf{P} = [\boldsymbol{\rho}\times]$, and $\boldsymbol{\rho} = [\rho_x, \rho_y, \rho_z]^\intercal \in \mathbb{R}^3$ (see eqn. 10.28 of [1]).

Using this notation, the attitude update (as defined in eqn. 10.29 of [1]) is

$$(\mathbf{R}_b^n)^+ = [\mathbf{I} - \mathbf{P}](\hat{\mathbf{R}}_b^n)^-.$$

Note that the attitude correction is multiplicative.

Quaternion

A similar approach to Section 1.4.3 is valid when the attitude error is represented by a quaternion. Let $\mathbf{q}_b^n$ represent the true quaternion from b-frame to n-frame. Let $\hat{\mathbf{q}}_b^n$ represent the computed quaternion. The error may be represented as

$$\mathbf{q}_{\hat{n}}^n = \mathbf{q}_b^n \otimes \hat{\mathbf{q}}_n^b$$

where $\mathbf{q}_{\hat{n}}^n$ represents the quaternion from the computed to actual navigation frame. The symbol $\otimes$ represents the quaternion multiplication operation defined in Section D of [1]. When the error between the true and computed rotation is zero, then $\mathbf{q}_{\hat{n}}^n = [1, 0, 0, 0]^\intercal$, otherwise $\mathbf{q}_{\hat{n}}^n$ may be represented as

$$\mathbf{q}_{\hat{n}}^n = \begin{bmatrix} \hat{\mathbf{q}}_s \\ \hat{\mathbf{q}}_v \end{bmatrix} = \begin{bmatrix} \sqrt{1 - \|\frac{1}{2}\boldsymbol{\rho}\|_2^2} \\ \frac{1}{2}\boldsymbol{\rho} \end{bmatrix} \approxeq \begin{bmatrix} 1 \\ \frac{1}{2}\boldsymbol{\rho} \end{bmatrix}, \tag{1.24}$$

where the scalar part of the quaternion is $\hat{\mathbf{q}}_s = 1$, and the vector part is $\hat{\mathbf{q}}_v = \boldsymbol{\rho}$. The approximation on the right-hand side of eqn. (1.24) is derived in Appendix **??**.

Using this notation, the multiplicative quaternion update is

$$\hat{\mathbf{q}}_b^{n\ +} = \mathbf{q}_{\hat{n}}^n \otimes \hat{\mathbf{q}}_b^{n\ -}.$$

Quaternion operations are defined in Section D of [1].

1.5 INS Error Model

Due to initial condition errors, system calibration errors, and measurement noise, estimation error develops over time such that $\boldsymbol{x}(t)$ are not equal $\hat{\boldsymbol{x}}(t)$ The error state vector is

$$\delta\boldsymbol{x} = [\delta\mathbf{p}^\intercal,\ \delta\mathbf{v}^\intercal,\ \delta\boldsymbol{\theta}^\intercal,\ \delta\mathbf{b}_a{}^\intercal,\ \delta\mathbf{b}_g{}^\intercal]^\intercal \in \mathbb{R}^{n_e},$$

where $\delta\mathbf{p}$, $\delta\mathbf{v}$, $\delta\boldsymbol{\theta}$, $\delta\mathbf{b}_a$, and $\delta\mathbf{b}_g$ each in $\mathbb{R}^3$ are the position, velocity, attitude, accelerometer bias and gyro bias error vectors, respectively.

The error state $\delta\boldsymbol{x}(t)$ is related to $\boldsymbol{x}(t)$ and $\hat{\boldsymbol{x}}(t)$ by $\delta\boldsymbol{x}(t) = \boldsymbol{x}(t) \ominus \hat{\boldsymbol{x}}(t)$. The symbol '$\ominus$', which is similar to the discussion in Section 1.4, represents the subtraction operation for position, velocity and bias states, and the multiplication operation of the attitude states. The fact that $n_s = 16$ and $n_e = 15$ is discussed in Section 1.4. The dynamics and stochastic properties of this estimation error vector are well understood, and can be found in Section 11.4 of [1]. The derivation and definitions of linear state transition error model is presented in Section 1.6.

1.6 INS Noise Propagation

1.6.1 Problem Formulation

Let $\boldsymbol{x}_v(t) = [\mathbf{p}^\intercal(t), \mathbf{v}^\intercal(t), \mathbf{q}^\intercal(t)]^\intercal \in \mathbb{R}^{10}$ represent the vehicle state comprised of position, velocity and attitude. Let $\boldsymbol{x}_c(t) = [\mathbf{b}_a^\intercal(t), \mathbf{b}_g^\intercal(t)]^\intercal \in \mathbb{R}^6$ represent the IMU calibration terms: accelerometer bias and gyro bias. Then $\boldsymbol{x}(t)$ can be represented as $\boldsymbol{x}(t) = [\boldsymbol{x}_v^\intercal(t), \boldsymbol{x}_c^\intercal(t)]^\intercal$.

Let τ_i denote the time instants of the i^{th} IMU measurements of $\mathbf{u}$, as defined in Section 1.1. Let $\boldsymbol{x}_i = \boldsymbol{x}(\tau_i)$ and $\mathbf{u}_i = \mathbf{u}(\tau_i)$.

Let the state estimate time propagation be represented as

$$\hat{\boldsymbol{x}}_{i+1} \doteq \boldsymbol{\phi}(\hat{\boldsymbol{x}}_i, \tilde{\mathbf{u}}_i),$$

where the vehicle state estimate is $\hat{\boldsymbol{x}}_{v,i+1} \doteq \boldsymbol{\phi}_v(\hat{\boldsymbol{x}}_{v,i}, \tilde{\mathbf{u}}_i)$.

Let the true state time propagation be represented as

$$\boldsymbol{x}_{i+1} = \boldsymbol{\phi}(\boldsymbol{x}_i, \mathbf{u}_i),$$

and the true vehicle state as $\boldsymbol{x}_{v,i+1} = \boldsymbol{\phi}_v(\boldsymbol{x}_{v,i}, \mathbf{u}_i)$.

Define the state error as

$$\delta\boldsymbol{x}_i = \boldsymbol{x}_i \ominus \hat{\boldsymbol{x}}_i \in \mathbb{R}^{n_e},$$

where the symbol '$\ominus$' is discussed in Section 1.5. Let $\delta\boldsymbol{x}_{v,i} \in \mathbb{R}^9$ represent the vehicle state error for position, velocity and attitude. Let $\delta\boldsymbol{x}_{c,i} \in \mathbb{R}^6$ represent the error in the IMU calibration terms: accelerometer bias and gyro bias. Then $\delta\boldsymbol{x}_i$ can be represented as $\delta\boldsymbol{x}_i = [\delta\boldsymbol{x}_{v,i}^\intercal, \delta\boldsymbol{x}_{c,i}^\intercal]^\intercal$.

Let the IMU measurement be defined as

$$\tilde{\mathbf{u}}(\tau_i) \triangleq \mathbf{u}(\tau_i) - \mathbf{b}(\tau_i) - \boldsymbol{\omega}_u(\tau_i) \in \mathbb{R}^6,$$

with additive stochastic errors $\boldsymbol{\omega}_u(\tau_i) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q_d})$ and $\mathbf{b} = [\mathbf{b}_a^\intercal, \mathbf{b}_g^\intercal]^\intercal$. The sensor bias $\mathbf{b}$ represents time correlated measurement errors, and $\boldsymbol{\omega}_u(\tau_i)$

represents the white measurement errors. Let the estimate $\hat{\mathbf{u}}_i \triangleq \tilde{\mathbf{u}}_i + \hat{\mathbf{b}}_i$, where $\hat{\mathbf{b}}_i$ is the estimate of $\mathbf{b}_i$ (i.e. $\hat{\boldsymbol{x}}_{c,i}$). Let the measurements $\tilde{\mathbf{u}}(\tau_i)$ be defined for IMU measurement times i, between aiding measurement times k, such that $\tau_i \in [t_{k-1}, t_k]$.

Define

$$\begin{aligned}
\delta\mathbf{u}_i &\triangleq \mathbf{u}_i - \hat{\mathbf{u}}_i \\
&= \mathbf{u}_i - \tilde{\mathbf{u}}_i - \hat{\mathbf{b}}_i \\
&= \mathbf{u}_i - (\mathbf{u}_i - \mathbf{b}_i - \boldsymbol{\omega}_{u,i}) - \hat{\mathbf{b}}_i \\
&= \delta\mathbf{b}_i + \boldsymbol{\omega}_{u,i},
\end{aligned}$$

where $\delta\mathbf{b}_i \triangleq \mathbf{b}_i - \hat{\mathbf{b}}_i$, and $\delta\mathbf{b}_i$ (i.e. $\delta\mathbf{b}_i = \delta\boldsymbol{x}_{c,i}$) is a state calibration term.

Linearization of the state error, using Taylor series to first order, yields

$$\begin{aligned}
\delta\boldsymbol{x}_{v,i+1} &= \boldsymbol{\phi}_v(\boldsymbol{x}_{v,i}, \mathbf{u}_i) - \boldsymbol{\phi}_v(\hat{\boldsymbol{x}}_{v,i}, \hat{\mathbf{u}}_i) \\
&= \boldsymbol{\phi}_v(\hat{\boldsymbol{x}}_{v,i}, \hat{\mathbf{u}}_i) + \left.\frac{\partial\boldsymbol{\phi}_v}{\partial\boldsymbol{x}_{v,i}}\right|_{\hat{\boldsymbol{x}}_{v,i}} \delta\boldsymbol{x}_{v,i} + \left.\frac{\partial\boldsymbol{\phi}_v}{\partial\mathbf{u}_i}\right|_{\hat{\mathbf{u}}_i} \delta\mathbf{u}_i - \boldsymbol{\phi}_v(\hat{\boldsymbol{x}}_{v,i}, \hat{\mathbf{u}}_i) \\
&= \left.\frac{\partial\boldsymbol{\phi}_v}{\partial\boldsymbol{x}_{v,i}}\right|_{\hat{\boldsymbol{x}}_{v,i}} \delta\boldsymbol{x}_{v,i} + \left.\frac{\partial\boldsymbol{\phi}_v}{\partial\mathbf{u}_i}\right|_{\hat{\mathbf{u}}_i} \delta\mathbf{u}_i \\
&= \mathbf{A}_i\delta\boldsymbol{x}_{v,i} + \mathbf{B}_i\delta\mathbf{u}_i \\
&= \mathbf{A}_i\delta\boldsymbol{x}_{v,i} + \mathbf{B}_i\delta\mathbf{b}_i + \mathbf{B}_i\boldsymbol{\omega}_{u,i}, \qquad (1.25)
\end{aligned}$$

where $\mathbf{A}_i = \frac{\partial\boldsymbol{\phi}_v}{\partial\boldsymbol{x}_{v,i}}\Big|_{\hat{\boldsymbol{x}}_{v,i},\hat{\mathbf{u}}_i} \in \mathbb{R}^{9\times9}$, and $\mathbf{B}_i = \frac{\partial\boldsymbol{\phi}_v}{\partial\mathbf{u}_i}\Big|_{\hat{\boldsymbol{x}}_{v,i},\hat{\mathbf{u}}_i} \in \mathbb{R}^{9\times6}$.

Let the model of the sensor bias be defined as a first-order Gauss-Markov process

$$\delta\mathbf{b}_{i+1} = \mathbf{F}_b\delta\mathbf{b}_i + \boldsymbol{\nu}, \qquad (1.26)$$

where $\mathbf{F}_\mathbf{b} \in \mathbb{R}^{6\times6}$ is selected such that the bias errors are modeled as either random constants or random walk plus constants (see eqns. 11.106 and 11.107 of [1]), and $\boldsymbol{\nu} \sim \mathcal{N}(\mathbf{0}, \sigma_\nu\mathbf{I})$.

Rewriting eqns. (1.25) and (1.26) in matrix form:

$$\begin{bmatrix} \delta\boldsymbol{x}_{v,i+1} \\ \delta\boldsymbol{x}_{c,i+1} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_i & \mathbf{B}_i \\ \mathbf{0} & \mathbf{F}_\mathbf{b} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{x}_{v,i} \\ \delta\mathbf{b}_i \end{bmatrix} + \begin{bmatrix} \mathbf{B}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}_{u,i} \\ \boldsymbol{\nu} \end{bmatrix}. \qquad (1.27)$$

For analysis, in the following section let $\mathbf{F}_\mathbf{b} = \mathbf{I}$.

1.6.2 Propagation of State Error

This section analyzes the error accumulation over the time interval $t \in [k-1, k]$ using superposition.

Propagation of Initial State Error

Consider eqn. (1.27) over the interval $t \in [k-1, k]$, where $\boldsymbol{\omega}_{u,i-1} = \mathbf{0}$ and $\boldsymbol{\nu} = \mathbf{0}$. Without loss of generality let $k = 1$, such that $t \in [0, 1]$. For each time instant, eqn. (1.27) can be represented in terms of $\mathbf{A}_i$, and $\mathbf{B}_i$, with initial condition errors $\delta\boldsymbol{x}_{v,0}$, and $\delta\mathbf{b}_0$:

$$\begin{aligned}
\delta\boldsymbol{x}_1 &= \begin{bmatrix} \mathbf{A}_0 & \mathbf{B}_0 \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{x}_{v,0} \\ \delta\mathbf{b}_0 \end{bmatrix} \\
\delta\boldsymbol{x}_2 &= \begin{bmatrix} \mathbf{A}_1 & \mathbf{B}_1 \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{x}_{v,1} \\ \delta\mathbf{b}_1 \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{A}_1 & \mathbf{B}_1 \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A}_0 & \mathbf{B}_0 \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{x}_{v,0} \\ \delta\mathbf{b}_0 \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{A}_1\mathbf{A}_0 & \mathbf{A}_1\mathbf{B}_0 + \mathbf{B}_1 \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{x}_{v,0} \\ \delta\mathbf{b}_0 \end{bmatrix} \\
\delta\boldsymbol{x}_3 &= \begin{bmatrix} \mathbf{A}_2 & \mathbf{B}_2 \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{x}_{v,2} \\ \delta\mathbf{b}_2 \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{A}_2 & \mathbf{B}_2 \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A}_1\mathbf{A}_0 & \mathbf{A}_1\mathbf{B}_0 + \mathbf{B}_1 \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{x}_{v,0} \\ \delta\mathbf{b}_0 \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{A}_2\mathbf{A}_1\mathbf{A}_0 & \mathbf{A}_2\mathbf{A}_1\mathbf{B}_0 + \mathbf{A}_2\mathbf{B}_1 + \mathbf{B}_2 \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{x}_{v,0} \\ \delta\mathbf{b}_0 \end{bmatrix}. \qquad (1.28)
\end{aligned}$$

Define F_s as the sample frequency of the sensor (e.g. IMU). Let $\mathbf{U}_k = \{\tilde{\mathbf{u}}(\tau_i) \text{ for } \tau_i \in [t_{k-1}, t_k]\}$. Let $\mathbf{X}_k = [\boldsymbol{x}(t_{k-L})^\intercal, \ldots, \boldsymbol{x}(t_k)^\intercal]^\intercal \in \mathbb{R}^{n_s(L+1)}$ denote the vehicle trajectory over a sliding time window that contains L one second GPS measurement epochs: $[\mathbf{y}_{k-L+1}, \ldots, \mathbf{y}_k]$. After F_s IMU time steps (i.e. $F_{s,k=1}$)

$$\begin{aligned}
\delta\boldsymbol{x}_{F_s} &= \left\{ \prod_{i=1}^{F_s} \begin{bmatrix} \mathbf{A}_i & \mathbf{B}_i \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \right\} \begin{bmatrix} \delta\boldsymbol{x}_{v,0} \\ \delta\mathbf{b}_0 \end{bmatrix} \qquad (1.29) \\
&= \boldsymbol{\Upsilon}(\hat{\mathbf{X}}_k, \mathbf{U}_k)[\delta\boldsymbol{x}_{v,0}, \delta\mathbf{b}_0]^\intercal, \qquad (1.30)
\end{aligned}$$

where the operator $\boldsymbol{\Upsilon}(\hat{\mathbf{X}}_k, \mathbf{U}_k)$ in eqn. (1.30) represents the product operation in eqn. (1.29), and $\hat{\mathbf{X}}_k$ is the estimate of $\mathbf{X}_k$. The product operation in eqn. (1.29) must follow the order of multiplications shown in eqn. (1.28).

Noise Propagation

Again consider eqn. (1.27) over the interval $t \in [k-1, k]$. Now analyze the effect of the noise terms $\boldsymbol{\omega}_u$ and $\boldsymbol{\nu}$, with $\delta\boldsymbol{x}_{v,0}$ and $\delta\mathbf{b}_0$ both zero.

To simplify notation, let

$$\mathbf{C}_i \triangleq \begin{bmatrix} \mathbf{A}_i & \mathbf{B}_i \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \mathbf{D}_i \triangleq \begin{bmatrix} \mathbf{B}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix},$$

and

$$\delta \boldsymbol{x}_i \triangleq \begin{bmatrix} \delta \boldsymbol{x}_{v,i} \\ \delta \boldsymbol{x}_{c,i} \end{bmatrix}, \ \mathbf{n}_i \triangleq \begin{bmatrix} \boldsymbol{\omega}_{u,i} \\ \boldsymbol{\nu} \end{bmatrix}.$$

Defining eqn. (1.27) using the terms above,

$$\delta \boldsymbol{x}_{i+1} = \mathbf{C}_i \delta \boldsymbol{x}_i + \mathbf{D}_i \mathbf{n}_i. \tag{1.31}$$

Performing operations on eqn. (1.31) (similar to the operations leading up to eqn. (1.28)),

$$\begin{aligned}
\delta \boldsymbol{x}_1 &= \mathbf{C}_0 \delta \boldsymbol{x}_0 + \mathbf{D}_0 \mathbf{n}_0 \\
\delta \boldsymbol{x}_2 &= \mathbf{C}_1 \delta \boldsymbol{x}_1 + \mathbf{D}_1 \mathbf{n}_1 \\
&= \mathbf{C}_1 (\mathbf{C}_0 \delta \boldsymbol{x}_0 + \mathbf{D}_0 \mathbf{n}_0) + \mathbf{D}_1 \mathbf{n}_1 \\
&= \mathbf{C}_1 \mathbf{C}_0 \delta \boldsymbol{x}_0 + \mathbf{C}_1 \mathbf{D}_0 \mathbf{n}_0 + \mathbf{D}_1 \mathbf{n}_1 \\
\delta \boldsymbol{x}_3 &= \mathbf{C}_2 \delta \boldsymbol{x}_2 + \mathbf{D}_2 \mathbf{n}_2 \\
&= \mathbf{C}_2 (\mathbf{C}_1 \mathbf{C}_0 \delta \boldsymbol{x}_0 + \mathbf{C}_1 \mathbf{D}_0 \mathbf{n}_0 + \mathbf{D}_1 \mathbf{n}_1) + \mathbf{D}_2 \mathbf{n}_2 \\
&= \mathbf{C}_2 \mathbf{C}_1 \mathbf{C}_0 \delta \boldsymbol{x}_0 + \mathbf{C}_2 \mathbf{C}_1 \mathbf{D}_0 \mathbf{n}_0 + \mathbf{C}_2 \mathbf{D}_1 \mathbf{n}_1 + \mathbf{D}_2 \mathbf{n}_2. \qquad (1.32)
\end{aligned}$$

For $i = F_s$, and $\delta \boldsymbol{x}_0 = \mathbf{0}$, the terms in eqn. (1.32) can be defined as

$$\begin{aligned}
\mathbf{w}_{k-1} &= \left\{ \sum_{i=0}^{F_s-2} \left(\prod_{j=i+1}^{F_s-1} \mathbf{C}_j \right) \mathbf{D}_i \mathbf{n}_i \right\} + \mathbf{D}_{F_s-1} \mathbf{n}_{F_s-1} \qquad (1.33) \\
&= \boldsymbol{\Gamma} \boldsymbol{\eta}.
\end{aligned}$$

Let the product of $\mathbf{C}_j$ in eqn. (1.33) be defined as

$$\mathbf{C}_j^p \triangleq \begin{cases} \prod_{j=q}^{p} \mathbf{C}_j = \mathbf{C}_p \cdots \mathbf{C}_{q-1} \mathbf{C}_q & \text{for } q \neq p \\ \mathbf{C}_q & \text{for } q = p \end{cases} \tag{1.34}$$

where product operation in eqn. (1.34) must follow the order of operations shown in eqn. (1.32). Let $\boldsymbol{\Gamma}$ and $\boldsymbol{\eta}$ be defined as

$$\begin{aligned}
\boldsymbol{\Gamma} &\triangleq \left[\mathbf{C}_1^{F_s-1} \mathbf{D}_0, \mathbf{C}_2^{F_s-1} \mathbf{D}_1, \ldots, \mathbf{C}_{F_s-1}^{F_s-1} \mathbf{D}_{F_s-2}, \mathbf{D}_{F_s-1} \right] \\
\boldsymbol{\eta} &\triangleq \left[\mathbf{n}_0, \mathbf{n}_1, \ldots, \mathbf{n}_{F_s-1} \right].
\end{aligned}$$

Summary

Combining the results from Sections 1.6.2 and 1.6.2, the linear state transition error model over $t \in [t_{k-1}, t_k]$ is

$$\delta \boldsymbol{x}_k = \boldsymbol{\Upsilon}_{k-1} \delta \boldsymbol{x}_{k-1} + \mathbf{w}_{k-1}. \tag{1.35}$$

with

$$\begin{aligned}
\mathbf{Q}_{\mathbf{D}} &= \mathrm{Cov}(\mathbf{w}_{k-1}) \in \mathbb{R}^{n_e \times n_e} \\
&= E\left\langle \mathbf{\Gamma} \boldsymbol{\eta} \boldsymbol{\eta}^{\mathsf{T}} \mathbf{\Gamma}^{\mathsf{T}} \right\rangle \\
&= E\left\langle \mathbf{\Gamma} \begin{bmatrix} \boldsymbol{\eta}_0 \\ \boldsymbol{\eta}_1 \\ \vdots \\ \boldsymbol{\eta}_{F_s-1} \end{bmatrix} \left[\boldsymbol{\eta}_0 \; \boldsymbol{\eta}_1 \; \cdots \; \boldsymbol{\eta}_{F_s-1}\right] \mathbf{\Gamma}^{\mathsf{T}} \right\rangle \\
&= \mathbf{\Gamma} \begin{bmatrix} \mathbf{Q}_{\mathbf{d},0} & & \\ & \ddots & \\ & & \mathbf{Q}_{\mathbf{d},F_s-1} \end{bmatrix} \mathbf{\Gamma}^{\mathsf{T}},
\end{aligned} \tag{1.36}$$

where $\mathbf{Q}_{\mathbf{d},i} = \boldsymbol{\eta}_i \boldsymbol{\eta}_i^{\mathsf{T}}$. The stochastic properties of eqn. (1.36) are well understood, and can be found in Sections 4.7 and 7.2.5.2 of [1].

1.7 Extended Kalman Filtering for GPS-aided INS

When GPS aiding measurements of the form (repeating eqn. (1.5) for clarity)

$$\tilde{\mathbf{y}}(t) = \mathbf{h}(\boldsymbol{x}(t)) + \boldsymbol{\eta}_y(t), \qquad \boldsymbol{\eta}_y \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_y), \tag{1.37}$$

are available, various methods use the initial state, inertial measurements, and aiding measurement information to estimate the vehicle state vector [1, 5, 6]. The Extended Kalman Filter (EKF) is a widely used method in aided INS, due to its implementation simplicity and real-time efficiency [7].

The standard EKF can be reformulated in Weighted Least Square (WLS) form [1, 7, 8]. Given the INS prior and the aiding measurements at time step t_k,

$$\hat{\boldsymbol{x}}_k^- = \boldsymbol{x}_k + \delta \boldsymbol{x}_k^-, \qquad \delta \boldsymbol{x}_k^- \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_k^-), \tag{1.38}$$

$$\tilde{\boldsymbol{y}}_k = \mathbf{h}_k(\boldsymbol{x}_k) + \boldsymbol{\eta}_k, \qquad \boldsymbol{\eta}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k), \tag{1.39}$$

where the process noise $\boldsymbol{w}_k$ and the measurement noise $\boldsymbol{n}_k$ [1, 5, 6] are assumed White-Gaussian-Noise (WGN). A *Maximum Likelihood* estimation of the state correction $\delta \boldsymbol{x}_k$ at t_k can be formulated as

$$\delta \boldsymbol{x}_k^+ = \underset{\delta \boldsymbol{x}_k}{arg\,max} \left\{ p_{\delta \boldsymbol{x}_k^-}(\delta \boldsymbol{x}_k) p_{\mathbf{n}_k}(\tilde{\mathbf{y}}_k - \mathbf{H}_k \hat{\boldsymbol{x}}_k^- - \mathbf{H}_k \delta \boldsymbol{x}_k) \right\}, \tag{1.40}$$

where $\mathbf{H}_k = \frac{\partial \mathbf{h}_k}{\partial \boldsymbol{x}}\big|_{\boldsymbol{x} = \hat{\boldsymbol{x}}_k^-}$ is the Jacobian matrix of the measurement model evaluated at the INS prior $\hat{\boldsymbol{x}}_k^-$.

A Least Square problem can be derived by evaluating the negative log-likelihood of the right hand side of eqn. (1.40),

$$\delta\boldsymbol{x}_k^+ = \underset{\delta\boldsymbol{x}_k}{arg\,min}\left\{\|\delta\boldsymbol{x}_k\|^2_{\mathbf{P}_k^-} + \|\delta\mathbf{y}_k - \mathbf{H}_k\delta\boldsymbol{x}_k\|^2_{\mathbf{R}_k}\right\}, \tag{1.41}$$

where $\delta\mathbf{y}_k \triangleq \tilde{\mathbf{y}}_k - \mathbf{H}_k\hat{\boldsymbol{x}}_k^-$ is the measurement residual (i.e. innovation) and the notation $\|\boldsymbol{x}\|^2_{\mathbf{C}} \triangleq \boldsymbol{x}^\intercal\mathbf{C}^{-1}\boldsymbol{x}$ is defined as the squared Mahalanobis distance.

Let

$$\mathbf{A}_k = \begin{bmatrix}\mathbf{I}\\ \mathbf{H}_k\end{bmatrix}, \quad \mathbf{C}_k = \begin{bmatrix}\mathbf{P}_k^- & \mathbf{0}\\ \mathbf{0} & \mathbf{R}_k\end{bmatrix} \quad \text{and} \quad \delta\mathbf{Y}_k = \begin{bmatrix}\mathbf{0}\\ \delta\mathbf{y}_k\end{bmatrix},$$

the Kalman Filter measurement update can be derived by solving a Weighted Least Square problem:

$$\min_{\delta\boldsymbol{x}} \; \|\delta\mathbf{Y}_k - \mathbf{A}_k\delta\boldsymbol{x}\|_{\mathbf{C}_k}. \tag{1.42}$$

The solution of eqn. (1.42) is

$$\delta\boldsymbol{x}_k^+ = \left(\mathbf{A}_k^\intercal\mathbf{C}_k^{-1}\mathbf{A}_k\right)^{-1}\mathbf{A}_k^\intercal\mathbf{C}_k^{-1}\delta\mathbf{Y}_k, \tag{1.43}$$

and the corresponding covariance of $\delta\boldsymbol{x}_k^+$ is $\mathbf{P}_k^+ = \left(\mathbf{A}_k^\intercal\mathbf{C}_k^{-1}\mathbf{A}_k\right)^{-1}$. With $\delta\boldsymbol{x}_k^+$, the INS state is updated as

$$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- \oplus \delta\boldsymbol{x}_k^+,$$

where the $\oplus$ operator is discussed in Section 1.4, and the Kalman gain is

$$\begin{aligned}\mathbf{K}_k &= \left(\mathbf{A}_k^\intercal\mathbf{C}_k^{-1}\mathbf{A}_k\right)^{-1}\mathbf{A}_k^\intercal\mathbf{C}_k^{-1}\\ &= \mathbf{P}_k^-\mathbf{H}_k^\intercal\left(\mathbf{R}_k + \mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^\intercal\right),\end{aligned} \tag{1.44}$$

where eqn. (1.44) is derived by expanding $\left(\mathbf{A}_k^\intercal\mathbf{C}_k^{-1}\mathbf{A}_k\right)^{-1}\mathbf{A}_k^\intercal\mathbf{C}_k^{-1}$.

The EKF works well for many aided INS applications, e.g. GPS-INS [1], Underwater-INS [9], Vision-Inertial-Odometry (VIO) [10–13]. However, the performance of the EKF depends on initial conditions and nonlinearities (see [14]), and prior EKF linearization points cannot be corrected at later times.

Two test cases are provided in Figs. 1.2 and 1.3, wherein the EKF divergence due to incorrect yaw initialization is shown. In both cases the EKF uses double-differenced GPS pseudorange and Doppler measurements and a consumer-grade IMU. Both Figs. 1.2 and 1.3 display norm of the position, velocity and attitude error, where error is defined as ground-truth minus the estimate. In the experiment, the vehicle is level and at $t = 20sec$.

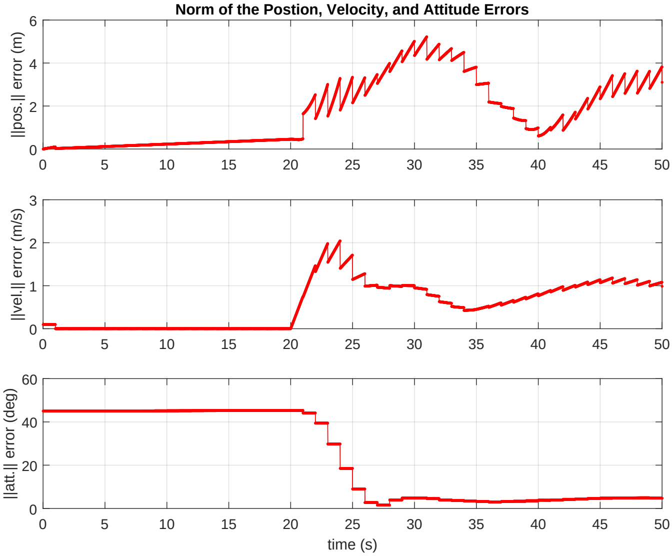

Figure 1.2: Case 1: Divergence of the GPS-INS EKF under poor yaw initialization.

accelerates (from stationary) with a heading of North. In Case 1, Fig. 1.2, the INS is initialized with 45° yaw error. While the EKF is able to remove most of the yaw error within a few GPS epochs, both position and velocity errors are large, and never recover. In Case 2, Fig. 1.3, the INS is initialized with 180° yaw error (worst-case). Here the EKF position, velocity, and attitude estimates diverge (oscillating indefinitely), and never recover.

To overcome the limitations of the EKF, a sliding window smoother is proposed for the GPS-INS problem, based on recent advances in the Simultaneous Localization and Mapping (SLAM) research community [14–16]. In Chapter **??**, the MAP optimization based smoothing method is extended to GPS-INS, to enhance the navigation performance accuracy and reliability. In Chapter **??**, the ability to completely remove the yaw initialization error of Case 2 (Fig. 1.3) is demonstrated using the proposed estimator of Chapter **??**.

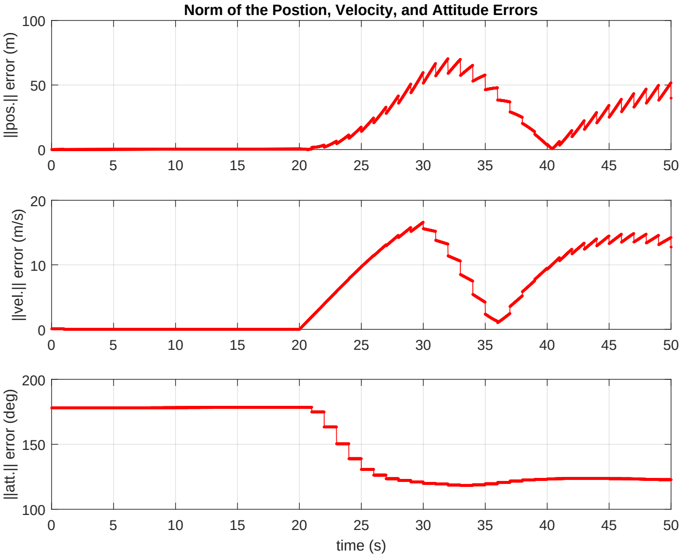

Figure 1.3: Case 2: Divergence of the GPS-INS EKF under worst-case yaw initialization.

Bibliography

[1] J. A. Farrell, *Aided Navigation: GPS with High Rate Sensors.* McGraw Hill, 2008.

[2] M. Shuster, “A survey of attitude representations,” *Journal of the Astronautical Sciences*, vol. 41, no. 4, pp. 439–517, 1993.

[3] M. Shuster, “The nature of the quaternion,” *Journal of the Astronautical Sciences*, vol. 56, no. 3, p. 359, 2010.

[4] M. Wei and K. P. Schwarz, “A strapdown inertial algorithm using earth-fixed cartesian frame,” *Navigation: JION*, vol. 371, no. 2, pp. 153–167, 1990.

[5] S. M. Kay, *Fundamentals of Statistical Signal Processing, Vol. I - Estimation Theory.* Prentice Hall PTR, 2013.

[6] R. G. Brown and P. Y. Hwang, *Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises, 4th Edition.* Wiley, 2012.

[7] J. A. Farrell and P. F. Roysdon, “Advanced Vehicle State Estimation: A Tutorial and Comparative Study,” *22th IFAC World Congress*, 2017.

[8] S. Hewitson and J. Wang, “Extended Receiver Autonomous Integrity Monitoring (eRAIM) for GNSS/INS Integration,” *Journal of Surveying Engineering*, vol. 136, no. 1, pp. 13–22, 2010.

[9] P. Miller, J. Farrell, Y. Zhao, and V. Djapic, “Autonomous Underwater Vehicle Navigation,” *IEEE J. of Oceanic Eng.*, vol. 35, no. 3, pp. 663 – 678, 2010.

[10] M. Li and A. I. Mourikis, “Vision-aided inertial navigation for resource-constrained systems,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Vilamoura, Portugal), pp. 1057–1063, Oct. 2012.

[11] M. Li and A. I. Mourikis, "Optimization-based estimator design for vision-aided inertial navigation," in *Proceedings of Robotics: Science and Systems*, (Sydney, Australia), Jul. 2012.

[12] M. Li, B. Kim, and A. I. Mourikis, "Real-time motion estimation on a cellphone using inertial sensing and a rolling-shutter camera," in *Proceedings of the IEEE International Conference on Robotics and Automation*, (Karlsruhe, Germany), pp. 4697–4704, May 2013.

[13] M. Li, H. Yu, X. Zheng, and A. I. Mourikis, "High-fidelity sensor modeling and self-calibration in vision-aided inertial navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, (Hong Kong), pp. 409–416, June 2014.

[14] F. Dellaert and M. Kaess, "Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing," *Int. J. Rob. Res.*, vol. 25, no. 12, pp. 1181–1203, 2006.

[15] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental Smoothing and Mapping," *IEEE Trans. Robotics,*, vol. 24, no. 6, pp. 1365–1378, 2008.

[16] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *Intl. J. of Robotics Research*, vol. 31, pp. 216–235, Feb. 2012.