

KF-GINS Algorithm Documentation

Liqiang Wang, wlq@whu.edu.cn

October 31, 2025

1 Parameter Definitions in KF-GINS

1.1 Coordinate System Definitions

The coordinate system definitions used in KF-GINS are shown in Table 1.

Table 1: Coordinate System Definitions

Symbol	Definition	Origin	Axis (X-Y-Z)
n-frame	Navigation Frame	IMU Geometric Center	North-East-Down
b-frame	IMU/Body Frame	IMU Geometric Center	Forward-Right-Down
i-frame	Inertial Frame (Earth-Centered)	Earth's Center	X points to the vernal equinox, Z points to the North Pole, XYZ follows the right-hand rule
e-frame	Earth-Centered Earth-Fixed Frame	Earth's Center	X points to the Prime Meridian, Y points to 90°E, Z points to the North Pole

1.2 Symbol Specification

(1) Symbol Font Definition

The symbol font definition specifications used in the document are shown in Table 2.

Table 2: Symbol Font Definitions

Symbol	Font	Example
Scalar	Italic, Non-Bold	g
Constant Scalar	Upright, Non-Bold	VRW
Coordinate System	Upright, Non-Bold	n-frame
Vector	Italic, Bold	\mathbf{v}_b^n
General Matrix	Italic, Bold	\mathbf{P}
Rotation Matrix	Upright, Bold	\mathbf{C}_b^n
Quaternion	Upright, Bold	\mathbf{q}_b^n

(2) Subscript and Superscript Definition

Subscripts and superscripts for vectors and matrices are placed on the right side of the symbol. The conventions for subscripts and superscripts are as follows:

For a vector \mathbf{v}_{rb}^p , the first subscript r represents the reference position (reference frame), the second subscript b represents the body position (body frame), and the superscript p represents the projection frame. Thus, \mathbf{v}_{rb}^p represents the vector from the reference position (origin of the reference frame) to the body position (origin of the body frame), expressed in the projection frame.

If the vector symbol \mathbf{v}_b^p has only one subscript, it is assumed that the projection frame are the reference frame. This represents the position of the b frame's origin relative to the p frame's origin, projected into the p frame. Additional parameters in the subscript, such as those after a comma or within parentheses, typically indicate time, e.g., k in $\mathbf{q}_{n(k)}^{e(k)}$ and $\Delta \mathbf{v}_{f,k}^b$ represent the time step. For IMU measurement errors, the subscript indicates the property, e.g., g and a in \mathbf{b}_g and \mathbf{b}_a represent gyroscope and accelerometer, respectively.

Specifically, the IMU's velocity and position with respected to the earth projected in the navigation frame, \mathbf{v}_{eb}^n and \mathbf{r}_{eb}^n , are simplified as \mathbf{v}^n and \mathbf{r}^n for convenience.

When the rotation matrix \mathbf{C}_b^n is used for coordinate transformations, its subscript b indicates the source coordinate system and the superscript n indicates the target coordinate system. Left-multiplying a vector by the rotation matrix transforms it between the source and target coordinate systems, e.g., $\mathbf{v}_{\text{rb}}^n = \mathbf{C}_b^n \mathbf{v}_{\text{rb}}^b$.

When representing attitude, the rotation matrix \mathbf{C}_b^n and the quaternion \mathbf{q}_b^n indicate the attitude of the body frame b-frame relative to the reference frame n-frame.

1.3 Attitude Representation

Four methods, including euler angles, rotation matrices, quaternions, and rotation vectors, are employed in KF-GINS for attitude representation.

The rotation sequence for Euler angles is defined as 'ZYX', corresponding to yaw angle ψ , pitch angle θ , and roll angle ϕ . Euler angles are used only for parameter input and result output.

The rotation matrix \mathbf{C}_b^n and the quaternion \mathbf{q}_b^n are commonly used in the algorithm. The rotation matrix is primarily used for coordinate transformations, while the quaternion is used for attitude and position updates.

The equivalent rotation vector $\boldsymbol{\phi}$ is a non-singular minimal representation of attitude parameters, widely used in inertial navigation algorithms and attitude error representation.

1.4 Geographical Parameters

The WGS84 ellipsoid model is employed in KF-GINS. Table 3 lists the relevant geographical parameters.

Table 3: WGS84 Ellipsoid Model Parameters		
Parameter	Symbol	Value
Semi-major axis	a	6378137 m
Semi-minor axis	b	6356752.3142 m
Flattening	f	1/298.257223563
First eccentricity	e	0.081819190842621
Second eccentricity	e'	0.082094437949696
Earth's gravitational constant	GM	$3.986004418 \times 10^{14}$ m
Earth's angular velocity	ω_e	7.292115×10^{-5} rad/s

The IMU's position and velocity in the n-frame are needed to compute the geographical parameters. The components of the IMU's position and velocity are expressed as:

$$\begin{aligned} \mathbf{p} &= [\varphi \quad \lambda \quad h]^T \\ \mathbf{v}^n &= [v_N \quad v_E \quad v_D]^T \end{aligned} \quad (1)$$

where, φ , λ , and h represent latitude, longitude, and altitude, respectively, and v_N , v_E , and v_D represent the northward, eastward, and downward components of ground velocity in the n-frame.

Using the above geographical parameters, the local gravity acceleration \mathbf{g}_p^n , the meridian radius R_M , and the prime vertical radius R_N can be calculated. The formula for local gravity acceleration is:

$$\begin{aligned} g(h, \varphi) &= g(\varphi) - (3.0877 \times 10^{-6} - 4.3 \times 10^{-9} \sin^2 \varphi) h + 0.72 \times 10^{-12} h^2 \\ g(\varphi) &= \gamma_a \left(1 + 0.0052790414 \sin^2 \varphi + 0.0000232718 \sin^4 \varphi \right. \\ &\quad \left. + 0.0000001262 \sin^6 \varphi + 0.0000000007 \sin^8 \varphi \right) \end{aligned} \quad (2)$$

where $\gamma_a = 9.7803267715 \text{ m/s}^2$ is the normal gravity at the equator, $g(\varphi)$ is the normal gravity at the ellipsoid surface ($h = 0$), and $g(h, \varphi)$ is the normal gravity at the corresponding latitude and altitude. The unit of $g(h, \varphi)$ is m/s^2 , φ is in radians, and h is in meters.

The gravity acceleration vector on the ellipsoid surface \mathbf{g}_p^n is expressed as:

$$\mathbf{g}_p^n = [0 \quad 0 \quad g(h, \varphi)]^T. \quad (3)$$

The formulas for the meridian radius R_M and the prime vertical radius R_N are:

$$\begin{aligned} R_M(\varphi) &= \frac{a(1-e^2)}{(1-e^2\sin^2\varphi)^{3/2}} \\ R_N(\varphi) &= \frac{a}{\sqrt{1-e^2\sin^2\varphi}} \end{aligned} \quad (4)$$

The Earth's rotation angular velocity in the e frame $\boldsymbol{\omega}_{ie}^e$ and in the n frame $\boldsymbol{\omega}_{ie}^n$, the angular velocity of the n-frame to the e-frame $\boldsymbol{\omega}_{en}^n$, and the angular velocity of the n-frame to the i-frame $\boldsymbol{\omega}_{in}^n$ are expressed as:

$$\begin{aligned} \boldsymbol{\omega}_{ie}^e &= [0 \quad 0 \quad \omega_e]^T \\ \boldsymbol{\omega}_{ie}^n &= [\omega_e \cos \varphi \quad 0 \quad -\omega_e \sin \varphi]^T \\ \boldsymbol{\omega}_{en}^n &= \begin{bmatrix} v_E/(R_N + h) \\ -v_N/(R_M + h) \\ -v_E \tan \varphi/(R_N + h) \end{bmatrix} \\ \boldsymbol{\omega}_{in}^n &= \boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n \end{aligned} \quad (5)$$

2 Inertial Navigation Algorithm

Incremental IMU measurements are serves as input of KF-GINS. Between t_{k-1} and t_k , the angular increment and velocity increment measured by the IMU are denoted as $\Delta\tilde{\boldsymbol{\theta}}_k$ and $\Delta\tilde{\boldsymbol{v}}_k$. These measurements are corrected for biases and scale factors to obtain compensated angular and velocity increments, $\Delta\boldsymbol{\theta}_k$ and $\Delta\boldsymbol{v}_k$, which are then used for navigation computation.

2.1 IMU Error Compensation

The mathematical model for compensating IMU biases and scale factors is:

$$\begin{aligned} \Delta\boldsymbol{\theta}_k &= [\boldsymbol{I}_3 + \text{diag}(\boldsymbol{s}_g)]^{-1} [\Delta\tilde{\boldsymbol{\theta}}_k - \boldsymbol{b}_g \Delta t_k] \\ \Delta\boldsymbol{v}_k &= [\boldsymbol{I}_3 + \text{diag}(\boldsymbol{s}_a)]^{-1} [\Delta\tilde{\boldsymbol{v}}_k - \boldsymbol{b}_a \Delta t_k]. \end{aligned} \quad (6)$$

Here, \boldsymbol{b}_g and \boldsymbol{b}_a are the gyroscope and accelerometer biases, while \boldsymbol{s}_g and \boldsymbol{s}_a are the gyroscope and accelerometer scale factors. $\Delta t_k = t_k - t_{k-1}$ is the sampling interval.

2.2 Velocity Update

The velocity update equation in the n frame is:

$$\boldsymbol{v}_k^n = \boldsymbol{v}_{k-1}^n + \Delta\boldsymbol{v}_{f,k}^n + \Delta\boldsymbol{v}_{g/cor,k}^n \quad (7)$$

where, \boldsymbol{v}_{k-1}^n and \boldsymbol{v}_k^n are the ground velocities in the n frame at t_{k-1} and t_k , respectively. $\Delta\boldsymbol{v}_{f,k}^n$ is the velocity increment due to specific force, and $\Delta\boldsymbol{v}_{g/cor,k}^n$ is the velocity increment due to gravity and Coriolis effects.

(1) Specific Force Integration

For short integration intervals, the rotation of the n frame can be approximated as linear. Thus, the specific force integration term can be simplified as:

$$\Delta\boldsymbol{v}_{f,k}^n = \frac{1}{2} \left(\boldsymbol{C}_{n(k-1)}^{n(k)} + \boldsymbol{I}_3 \right) \underbrace{\int_{t_{k-1}}^{t_k} \boldsymbol{C}_{b(t)}^{b(k-1)} \boldsymbol{f}^b(t) dt}_{\Delta\boldsymbol{v}_{f,k}^b} \quad (8)$$

Here, $\Delta\boldsymbol{v}_{f,k}^b$ is the specific force integration term in the b frame. $\boldsymbol{C}_{n(k-1)}^{n(k)}$ represents the rotation of the n frame from t_{k-1} to t_k , which can be approximated using the equivalent rotation vector $\boldsymbol{\zeta}_{n(k-1)n(k)}$:

$$\boldsymbol{C}_{n(k-1)}^{n(k)} \approx \boldsymbol{I}_3 - \left(\boldsymbol{\zeta}_{n(k-1)n(k)} \times \right) \quad (9)$$

where

$$\zeta_{n(k-1)n(k)} = \int_{t_{k-1}}^{t_k} \omega_{in}^n(t) dt \approx [\omega_{ie,k-1/2}^n + \omega_{en,k-1/2}^n] \Delta t_k \quad (10)$$

The specific force integration term in the b frame, $\Delta \mathbf{v}_{f,k}^b$, can be computed using IMU measurements. Employing two-sample assumption (linear variations of acceleration and angular velocity), it is expressed as:

$$\Delta \mathbf{v}_{f,k}^b = \Delta \mathbf{v}_k + \frac{1}{2} (\Delta \boldsymbol{\theta}_k \times \Delta \mathbf{v}_k) + \frac{1}{12} (\Delta \boldsymbol{\theta}_{k-1} + \Delta \mathbf{v}_k + \Delta \mathbf{v}_{k-1} \times \Delta \boldsymbol{\theta}_k) \quad (11)$$

The second and third terms on the right-hand side are the rotational effect and sculling effect compensations, respectively.

(2) Gravity and Coriolis Integration

The gravity and Coriolis acceleration is expressed as:

$$\mathbf{a}_{gc}(t) = \mathbf{g}_p^n(t) - [2\omega_{ie}^n(t) + \omega_{en}^n(t)] \times \mathbf{v}^n(t) \quad (12)$$

For short intervals, the integration term can be approximated using trapezoidal integration:

$$\Delta \mathbf{v}_{g/cor,k}^n \approx [\mathbf{g}_p^n(t_{k-1/2}) - (2\omega_{ie}^n(t_{k-1/2}) + \omega_{en}^n(t_{k-1/2})) \times \mathbf{v}^n(t_{k-1/2})] \Delta t_k \quad (13)$$

(3) Algorithm Implementation

The specific force and gravity-Coriolis integration terms can be computed separately. First, the geographical parameters at $t_{k-1/2}$ are calculated using the IMU position and velocity at t_{k-1} as an initial approximation. Then, $\Delta \mathbf{v}_{f,k}^n$ is computed using equations (11), (10), and (8). The gravity-Coriolis term $\Delta \mathbf{v}_{g/cor,k}^n$ is computed using equation (13). Finally, the velocity update equation (7) is used to compute the updated velocity in the n frame.

To refine the approximation, the velocity at $t_{k-1/2}$ is interpolated using the updated velocity at t_k , and the position at $t_{k-1/2}$ is extrapolated. The velocity and position at $t_{k-1/2}$ are calculated as:

$$\begin{aligned} \mathbf{v}_{k-1/2}^n &= \frac{1}{2} (\mathbf{v}_k^n + \mathbf{v}_{k-1}^n) \\ \mathbf{p}_{k-1/2} &= \mathbf{p}_{k-1} + \frac{1}{2} \mathbf{D}_R^{-1} \mathbf{v}_{k-1/2}^n \Delta t_k \end{aligned} \quad (14)$$

Here, \mathbf{D}_R is the transformation matrix of relative quantities between the n frame and geodetic coordinates:

$$\mathbf{D}_R = \begin{bmatrix} R_M + h & 0 & 0 \\ 0 & (R_N + h) \cos \varphi & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (15)$$

The calculated position and velocity at $t_{k-1/2}$ are then used to recompute the geographical parameters and refine the velocity update at t_k .

2.3 Position Update

(1) Position Update Equation

The IMU position is represented using latitude φ , longitude λ , and altitude h . Latitude and longitude are two angles that can be represented by a quaternion, referred to as the position quaternion \mathbf{q}_n^e . The position update for latitude and longitude can be expressed as:

$$\mathbf{q}_{n(k)}^{e(k)} = \mathbf{q}_{e(k-1)}^{e(k)} \otimes \mathbf{q}_{n(k-1)}^{e(k-1)} \otimes \mathbf{q}_{n(k)}^{n(k-1)} \quad (16)$$

Here, $\mathbf{q}_{n(k)}^{e(k)}$ and $\mathbf{q}_{n(k-1)}^{e(k-1)}$ represent the position quaternions at time t_k and t_{k-1} . $\mathbf{q}_{e(k-1)}^{e(k)}$ and $\mathbf{q}_{n(k)}^{n(k-1)}$ are the rotation quaternions for the e frame and n frame, respectively.

The rotation vector corresponding to the n-frame rotation quaternion $\mathbf{q}_{n(k)}^{n(k-1)}$ is denoted as $\zeta_{n(k)n(k-1)}$, and the rotation vector corresponding to the e-frame rotation quaternion $\mathbf{q}_{e(k-1)}^{e(k)}$ is denoted as $\xi_{e(k-1)e(k)}$. Their expressions are:

$$\begin{aligned} \zeta_{n(k)n(k-1)} &= -[\omega_{ie}^n(t_{k-1/2}) + \omega_{en}^n(t_{k-1/2})] \Delta t_k \\ \xi_{e(k-1)e(k)} &= [\omega_{ie,e,k-1/2}^e] \Delta t_k \end{aligned} \quad (17)$$

The altitude is updated using velocity integration, which can be expressed as:

$$h_k = h_{k-1} + v_{D,k-1/2}^n \Delta t_k \quad (18)$$

Here, $v_{D,k-1/2}^n$ represents the downward component of the ground velocity $\mathbf{v}_{k-1/2}^n$. Since the positive direction of altitude is opposite to the positive direction of downward velocity, a negative sign is added when updating altitude using velocity.

(2) Algorithm Implementation

In the algorithm implementation, the velocity at time t_k after the velocity update is substituted into equation (14) to recompute the IMU position and velocity at time $t_{k-1/2}$. The corresponding geographical parameters are then calculated. Using equation (17), the rotation vectors for the n-frame and e-frame are computed. The position quaternion $\mathbf{q}_{n(k)}^{e(k)}$ is updated using equation (16). The altitude h_k is updated using equation (18). Finally, the updated quaternion and altitude are converted into latitude and longitude to obtain the IMU position at time t_k .

2.4 Attitude Update

(1) Attitude Update Equation

The attitude update is performed using the attitude quaternion \mathbf{q}_b^n , which expressed as follows:

$$\mathbf{q}_{b(k)}^{n(k)} = \mathbf{q}_{n(k-1)}^{n(k)} \otimes \mathbf{q}_{b(k-1)}^{n(k-1)} \otimes \mathbf{q}_{b(k)}^{b(k-1)} \quad (19)$$

Here, $\mathbf{q}_{b(k)}^{n(k)}$ and $\mathbf{q}_{b(k-1)}^{n(k-1)}$ represent the attitude quaternions at time t_k and t_{k-1} . $\mathbf{q}_{n(k)}^{n(k-1)}$ and $\mathbf{q}_{b(k)}^{b(k-1)}$ are the rotation quaternions for the n frame and b frame, respectively.

The rotation vector corresponding to the n-frame rotation quaternion $\mathbf{q}_{n(k)}^{n(k-1)}$ is denoted as $\boldsymbol{\zeta}_{n(k)n(k-1)}$, as shown in equation (17). The rotation vector corresponding to the b-frame rotation quaternion $\mathbf{q}_{b(k)}^{b(k-1)}$ is denoted as $\boldsymbol{\phi}_k$. Based on the two-sample assumption (linear variation of angular velocity within the integration interval), the equivalent rotation vector for the b-frame is expressed as:

$$\boldsymbol{\phi}_k = \Delta \boldsymbol{\theta}_k + \frac{1}{12} (\Delta \boldsymbol{\theta}_{k-1} \times \Delta \boldsymbol{\theta}_k) \quad (20)$$

where, the second term on the right-hand side is the coning effect compensation.

(2) Position Interpolation Equation

To compute the equivalent rotation vector for the n frame, the position and velocity at time $t_{k-1/2}$ are required. The velocity at $t_{k-1/2}$ can be obtained using equation (18), and the altitude at $t_{k-1/2}$ is interpolated as $h_{k-1/2} = (h_{k-1} + h_k)/2$. The latitude and longitude at $t_{k-1/2}$ are interpolated using the position quaternion. First, the position quaternion at $t_{k-1/2}$ is computed as:

$$\begin{aligned} \mathbf{q}_{\delta\theta} &= \left(\mathbf{q}_{n(k-1)}^{n(k-1)} \right)^{-1} \otimes \mathbf{q}_{n(k)}^{n(k)} \\ \mathbf{q}_{n(k-1/2)}^{n(k-1/2)} &= \mathbf{q}_{n(k-1)}^{n(k-1)} \otimes \mathbf{q}_{0.5\delta\theta} \end{aligned} \quad (21)$$

Here, $\delta\theta$ is the equivalent rotation vector corresponding to the position change from t_{k-1} to t_k , and $0.5\delta\theta$ represents the interpolated rotation vector. Then $\mathbf{q}_{n(k-1/2)}^{n(k-1/2)}$ is converted into latitude and longitude at $t_{k-1/2}$.

(3) Algorithm Implementation

In the algorithm implementation, the IMU position and velocity at time $t_{k-1/2}$ are first computed using equation (14) and equation (21). Then the position and velocity at $t_{k-1/2}$ are employed to compute the corresponding geographical parameters. The rotation vector for the b frame is then calculated using equation (20), and the attitude quaternion $\mathbf{q}_{b(k)}^{n(k)}$ is updated using equation (19).

3 Integrated Navigation Algorithm

3.1 Kalman Filter Equations

Since the system equations of integrated navigation are highly nonlinear, to meet the linearization requirements of the Kalman filter for the state and observation equations, we adopt the Extended Kalman Filter (EKF) or

Error-State Kalman Filter (ESKF) to implement the integrated navigation algorithm. The discrete state and observation equations are expressed as:

$$\begin{aligned}\delta\mathbf{x}_k &= \mathbf{\Phi}_{k/k-1}\delta\mathbf{x}_{k-1} + \mathbf{w}_{k-1} \\ \delta\mathbf{z}_k &= \mathbf{H}_k\delta\mathbf{x}_k + \mathbf{v}_k\end{aligned}\quad (22)$$

Here, $\delta\mathbf{x}_{k-1}$ and $\delta\mathbf{x}_k$ denote the error state vectors ($n \times 1$) at the time steps t_{k-1} and t_k , respectively; $\mathbf{\Phi}_{k/k-1}$ is the state transition matrix ($n \times n$) from t_{k-1} to t_k ; $\delta\mathbf{z}_k$ is the measurement vector ($m \times 1$) at t_k ; \mathbf{H}_k is the measurement matrix ($m \times n$); \mathbf{w}_{k-1} is the discrete-time system noise ($n \times 1$), and \mathbf{v}_k is the measurement noise ($m \times 1$). Here, n and m represent the dimensions of the system state and measurement vector, respectively.

Denote \mathbf{P}_k as the covariance of the system state vector $\delta\mathbf{x}_k$ at time t_k , and \mathbf{Q}_{k-1} as the covariance of the system noise between t_{k-1} and t_k . The prediction of the system state and covariance in the Kalman filter can be expressed as:

$$\begin{aligned}\delta\mathbf{x}_{k/k-1} &= \mathbf{\Phi}_{k/k-1}\delta\mathbf{x}_{k-1} \\ \mathbf{P}_{k/k-1} &= \mathbf{\Phi}_{k/k-1}\mathbf{P}_{k-1}\mathbf{\Phi}_{k/k-1}^T + \mathbf{Q}_{k-1}\end{aligned}\quad (23)$$

According to the minimum variance criterion, the observation update equations of the Kalman filter at time t_k can be derived as:

$$\begin{aligned}\mathbf{K}_k &= \mathbf{P}_{k/k-1}\mathbf{H}_k^T \left(\mathbf{H}_k\mathbf{P}_{k/k-1}\mathbf{H}_k^T + \mathbf{R}_k \right)^{-1} \\ \delta\mathbf{x}_k &= \delta\mathbf{x}_{k/k-1} + \mathbf{K}_k\delta\mathbf{z} \\ \mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k/k-1}(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T\end{aligned}\quad (24)$$

Here, \mathbf{K}_k is the Kalman gain ($n \times m$), \mathbf{R}_k is the covariance matrix of the observation noise ($m \times m$), and $\delta\mathbf{z}_{k/k-1}$ is the measurement vector at time t_k ($m \times 1$).

3.2 Error Definition and System State Vector

(1) Error Perturbation Definition

The system state is perturbed with the error state, which is added to the system state vector. Position and velocity errors are perturbed in the n frame, while attitude errors adopt the ϕ angle error model. The error perturbation is defined as:

$$\begin{aligned}\hat{\mathbf{r}}^n &= \mathbf{r}^n + \delta\mathbf{r}^n \\ \hat{\mathbf{v}}^n &= \mathbf{v}^n + \delta\mathbf{v}^n \\ \hat{\mathbf{C}}_b^n &= [\mathbf{I}_3 - (\phi \times)] \mathbf{C}_b^n\end{aligned}\quad (25)$$

The error definitions for IMU biases and scale factors are:

$$\begin{aligned}\hat{\mathbf{b}}_g &= \mathbf{b}_g - \delta\mathbf{b}_g \\ \hat{\mathbf{b}}_a &= \mathbf{b}_a - \delta\mathbf{b}_a \\ \hat{\mathbf{s}}_g &= \mathbf{s}_g - \delta\mathbf{s}_g \\ \hat{\mathbf{s}}_a &= \mathbf{s}_a - \delta\mathbf{s}_a\end{aligned}\quad (26)$$

(2) IMU Measurements and Error Representation

Substituting the bias and scale factor errors into equation (6), the compensated IMU measurements can be expressed as:

$$\begin{aligned}\Delta\hat{\boldsymbol{\theta}}_k &= [\mathbf{I}_3 + \text{diag}(\hat{\mathbf{s}}_g)]^{-1} \left[\Delta\tilde{\boldsymbol{\theta}}_k - \hat{\mathbf{b}}_g\Delta t_k \right] \\ \Delta\hat{\mathbf{v}}_k &= [\mathbf{I}_3 + \text{diag}(\hat{\mathbf{s}}_a)]^{-1} \left[\Delta\tilde{\mathbf{v}}_k - \hat{\mathbf{b}}_a\Delta t_k \right].\end{aligned}\quad (27)$$

Dividing both sides of the above equations by Δt_k , the compensated IMU measurements in rate form can be obtained. Considering that the biases and scale factors are small and ignoring second-order error terms, the following approximation can be made: $[\mathbf{I}_3 + \text{diag}(\hat{\mathbf{s}}_g)]^{-1} \approx [\mathbf{I}_3 - \text{diag}(\hat{\mathbf{s}}_g)]$. Considering the random noise in IMU measurements, the rate-form IMU measurements can be expressed as:

$$\begin{aligned}\dot{\mathbf{f}}_{ib}^b &= [\mathbf{I}_3 - \text{diag}(\mathbf{s}_g)] \left(\dot{\boldsymbol{\omega}}_{ib}^b - \mathbf{b}_g \right) + \delta\mathbf{b}_g + \text{diag}(\delta\mathbf{s}_g)\dot{\boldsymbol{\omega}}_{ib}^b \approx \boldsymbol{\omega}_{ib}^b + \delta\mathbf{b}_g + \text{diag}(\boldsymbol{\omega}_{ib}^b)\delta\mathbf{s}_g + \mathbf{w}_g \\ \dot{\mathbf{f}}_{ib}^b &= [\mathbf{I}_3 - \text{diag}(\mathbf{s}_a)] \left(\dot{\tilde{\mathbf{f}}}_{ib}^b - \mathbf{b}_a \right) + \delta\mathbf{b}_a + \text{diag}(\delta\mathbf{s}_a)\dot{\tilde{\mathbf{f}}}_{ib}^b \approx \mathbf{f}_{ib}^b + \delta\mathbf{b}_a + \text{diag}(\mathbf{f}_{ib}^b)\delta\mathbf{s}_a + \mathbf{w}_a\end{aligned}\quad (28)$$

Here, ω_{ib}^b and f_{ib}^b represent the true angular velocity and acceleration, respectively, while w_g and w_a represent the white noise in angular velocity and acceleration measurements. The errors in angular velocity and acceleration can be expressed as:

$$\begin{aligned}\delta\omega_{ib}^b &= \hat{\omega}_{ib}^b - \omega_{ib}^b = \delta b_g + \text{diag}(\omega_{ib}^b)\delta s_g + w_g \\ \delta f_{ib}^b &= \hat{f}_{ib}^b - f_{ib}^b = \delta b_a + \text{diag}(f_{ib}^b)\delta s_a + w_a\end{aligned}\quad (29)$$

Considering the characteristics of IMU biases and scale factors, the bias and scale factor errors are modeled as first-order Gauss-Markov processes, with the continuous-time differential equations expressed as:

$$\begin{aligned}\delta\dot{b}_g(t) &= -\frac{1}{T_{gb}}\delta b_g(t) + w_{gb}(t) \\ \delta\dot{b}_a(t) &= -\frac{1}{T_{ab}}\delta b_a(t) + w_{ab}(t) \\ \delta\dot{s}_g(t) &= -\frac{1}{T_{gs}}\delta s_g(t) + w_{gs}(t) \\ \delta\dot{s}_a(t) &= -\frac{1}{T_{as}}\delta s_a(t) + w_{as}(t)\end{aligned}\quad (30)$$

Here, T_{gb} , T_{ab} , T_{gs} , and T_{as} represent the correlation times of gyroscope bias, accelerometer bias, gyroscope scale factor, and accelerometer scale factor, respectively. w_{gb} , w_{ab} , w_{gs} , and w_{as} represent the driving Gaussian white noise for gyroscope bias, accelerometer bias, gyroscope scale factor, and accelerometer scale factor, respectively.

(3) System State Vector and Propagation Noise

The system state vector for Kalman filtering is defined as:

$$\delta x = \left[(\delta r^n)^T \quad (\delta v^n)^T \quad (\phi)^T \quad (\delta b_g)^T \quad (\delta b_a)^T \quad (\delta s_g)^T \quad (\delta s_a)^T \right]^T \quad (31)$$

The propagation noise includes IMU measurement noise and the driving noise of IMU bias and scale factor errors, defined as:

$$w = \left[(w_g)^T \quad (w_a)^T \quad (w_{gb})^T \quad (w_{ab})^T \quad (w_{gs})^T \quad (w_{as})^T \right]^T \quad (32)$$

3.3 Error Differential Equations

To compute the discrete system equations for Kalman filtering, we first derive the continuous error differential equations. The differential equations for IMU position, velocity, and attitude are:

$$\begin{aligned}\dot{\varphi} &= \frac{v_N}{R_M + h}, \quad \dot{\lambda} = \frac{v_E}{(R_N + h)\cos\varphi}, \quad \dot{h} = -v_D \\ \dot{v}^n &= C_b^n f^b - (2\omega_{ie}^n + \omega_{en}^n) \times v^n + g_p^n \\ \dot{C}_b^n &= C_b^n (\omega_{ib}^b \times) - (\omega_{in}^n \times) C_b^n\end{aligned}\quad (33)$$

Since position perturbance are defined in the n-frame, while absolute positions are expressed in geodetic coordinates (latitude, longitude, and altitude), the relationship between the n frame position error $\delta r^n = [\delta r_N \quad \delta r_E \quad \delta r_D]^T$ and the geodetic coordinate error $\delta p = [\delta\varphi \quad \delta\lambda \quad \delta h]^T$ is:

$$\delta r_N = \delta\varphi(R_M + h), \quad \delta r_E = \delta\lambda(R_N + h)\cos\varphi, \quad \delta r_D = -\delta h \quad (34)$$

Combining the above differential equations with the error perturbation definitions in equation (25), the error differential equations for IMU position, velocity, and attitude are:

$$\begin{aligned}\delta\dot{r}^n &= -\omega_{en}^n \times \delta r^n - v^n \times \delta\theta + \delta v^n \\ \delta\dot{v}^n &= C_b^n \delta f_{ib}^b + f^n \times \phi - (2\omega_{ie}^n + \omega_{en}^n) \times \delta v^n + v^n \times (2\delta\omega_{ie}^n + \delta\omega_{en}^n) + \delta g_p^n \\ \dot{\phi} &= -\omega_{in}^n \times \phi + \delta\omega_{in}^n - C_b^n \delta\omega_{ib}^b\end{aligned}\quad (35)$$

Where:

$$\begin{aligned}
\delta\boldsymbol{\theta} &= \begin{bmatrix} 0 & 1/(R_N+h) & 0 \\ -1/(R_M+h) & 0 & 0 \\ 0 & -\tan\varphi/(R_N+h) & 0 \end{bmatrix} \delta\mathbf{r}^n \\
\delta\boldsymbol{\omega}_{ie}^n &= \begin{bmatrix} -\frac{\omega_e \sin\varphi}{R_M+h} \delta r_N & 0 & -\frac{\omega_e \cos\varphi}{R_M+h} \delta r_N \end{bmatrix}^T, \quad \delta\mathbf{g}_p^n = \begin{bmatrix} 0 & 0 & \frac{2g(\varphi, h)}{\sqrt{R_M R_N + h}} \delta r_D \end{bmatrix}^T \\
\delta\boldsymbol{\omega}_{en}^n &= \begin{bmatrix} \frac{v_E}{(R_N+h)^2} \delta r_D + \frac{1}{R_N+h} \delta v_E \\ -\frac{v_N}{(R_M+h)^2} \delta r_D - \frac{1}{R_M+h} \delta v_N \\ -\frac{v_E}{(R_M+h)(R_N+h)\cos^2\varphi} \delta r_N - \frac{v_E \tan\varphi}{(R_N+h)^2} \delta r_D - \frac{\tan\varphi}{(R_N+h)} \delta v_E \end{bmatrix}
\end{aligned} \tag{36}$$

3.4 State Transition Matrix and Noise Covariance Matrix

The continuous-time system state differential equation for Kalman filtering is expressed as:

$$\dot{\delta\mathbf{x}}(t) = \mathbf{F}(t)\delta\mathbf{x}(t) + \mathbf{G}(t)\mathbf{w}(t) \tag{37}$$

Substituting the IMU acceleration and angular velocity measurement errors from equation (29) into the position, velocity, and attitude differential equations (35), the continuous-time state transition matrix is:

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_{rr} & \mathbf{I}_3 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{F}_{vr} & \mathbf{F}_{vv} & [(\mathbf{C}_b^n \mathbf{f}^b) \times] & 0 & \mathbf{C}_b^n & 0 & \mathbf{C}_b^n \text{diag}(\mathbf{f}^b) \\ \mathbf{F}_{\phi r} & \mathbf{F}_{\phi v} & -(\boldsymbol{\omega}_{in}^n \times) & -\mathbf{C}_b^n & 0 & -\mathbf{C}_b^n \text{diag}(\boldsymbol{\omega}_{ib}^b) & 0 \\ 0 & 0 & 0 & \frac{-1}{T_{gb}} \mathbf{I}_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{-1}{T_{ab}} \mathbf{I}_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{-1}{T_{gs}} \mathbf{I}_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{T_{as}} \mathbf{I}_3 \end{bmatrix} \tag{38}$$

The continuous-time noise driving matrix is:

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{C}_b^n & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{C}_b^n & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{I}_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{I}_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I}_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{I}_3 \end{bmatrix} \tag{39}$$

Based on the propagation noise definition in equation (32), the system noise covariance matrix is:

$$\mathbf{q}(t) = \begin{bmatrix} \text{VRW}^2 \mathbf{I}_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & \text{ARW}^2 \mathbf{I}_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{2\sigma_{gb}^2}{T_{gb}} \mathbf{I}_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{2\sigma_{ab}^2}{T_{ab}} \mathbf{I}_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{2\sigma_{gs}^2}{T_{gs}} \mathbf{I}_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{2\sigma_{as}^2}{T_{as}} \mathbf{I}_3 \end{bmatrix} \tag{40}$$

Here, VRW and ARW are the velocity random walk and angle random walk parameters of the IMU, while σ_{gb} , σ_{ab} , σ_{gs} , and σ_{as} are the standard deviations of the gyroscope bias, accelerometer bias, gyroscope scale factor, and accelerometer scale factor, respectively.

By applying a first-order approximation to the continuous-time system differential equations, the discrete-time state transition matrix $\Phi_{k/k-1}$ and the noise covariance matrix \mathbf{Q}_k can be derived as:

$$\begin{aligned}\Phi_{k/k-1} &= \exp(\mathbf{F}(t_{k-1})\Delta t_k) \approx \mathbf{I} + \mathbf{F}(t_{k-1})\Delta t_k \\ \mathbf{Q}_k &\approx \frac{1}{2} \left[\Phi_{k/k-1} \mathbf{G}(t_{k-1}) \mathbf{q}(t_{k-1}) \mathbf{G}^T(t_{k-1}) \Phi_{k/k-1}^T + \mathbf{G}(t_k) \mathbf{q}(t_k) \mathbf{G}^T(t_k) \right] \Delta t_k\end{aligned}\quad (41)$$

Then, Kalman prediction equations (23) are employed to predict the state and update the covariance matrix.

3.5 GNSS Position Observation Equation

GNSS measures the geodetic coordinates of the GNSS antenna phase center, while INS calculates the position of the IMU measurement center. Therefore, when integrating GNSS and INS, it is necessary to compensate for the lever arm ℓ_{ant}^b between the IMU and the GNSS antenna. When constructing the GNSS position observation equation, the IMU position is first converted to the GNSS antenna phase center position:

$$\mathbf{r}_G^n = \mathbf{r}_I^n + \mathbf{C}_b^n \ell_{ant}^b \quad (42)$$

The GNSS-measured position is expressed as $\tilde{\mathbf{r}}_G^n = \mathbf{r}_G^n - \mathbf{n}_r$, where \mathbf{n}_r represents the GNSS position measurement error in the n frame, and its noise covariance is denoted as \mathbf{R}_r .

The GNSS position measurement vector is expressed as:

$$\begin{aligned}\delta \mathbf{z}_r^n &= \hat{\mathbf{r}}_G^n - \tilde{\mathbf{r}}_G^n \\ \delta \mathbf{z}_r^n &\approx \delta \mathbf{r}^n + (\mathbf{C}_b^n \ell_{ant}^b) \times \phi + \mathbf{n}_r \\ \delta \mathbf{z}_r^n &= \mathbf{H}_r \delta \mathbf{x} + \mathbf{n}_r\end{aligned}\quad (43)$$

where \mathbf{H}_r is the measurement matrix, expressed as:

$$\mathbf{H}_r = [\mathbf{I}_3 \quad \mathbf{0}_3 \quad [(\mathbf{C}_b^n \ell_{ant}^b) \times] \quad \mathbf{0}_3 \quad \mathbf{0}_3 \quad \mathbf{0}_3 \quad \mathbf{0}_3] \quad (44)$$

Generally, the GNSS receiver and INS output positions are expressed in geodetic coordinates, i.e., $\hat{\mathbf{p}}_I$ and $\tilde{\mathbf{p}}_G$. The predicted GNSS antenna phase center position $\hat{\mathbf{p}}_G$ and the GNSS measurement vector can be expressed as:

$$\begin{aligned}\hat{\mathbf{p}}_G &= \hat{\mathbf{p}}_I + \mathbf{D}_R^{-1} \hat{\mathbf{C}}_b^n \ell_{ant}^b \\ \delta \mathbf{z}_r^n &= \mathbf{D}_R (\hat{\mathbf{p}}_G - \tilde{\mathbf{p}}_G)\end{aligned}\quad (45)$$

where

$$\mathbf{D}_R = \begin{bmatrix} R_M + h & 0 & 0 \\ 0 & (R_N + h) \cos \varphi & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (46)$$

Using the INS-predicted position and GNSS-measured position at time t_k , the measurement vector $\delta \mathbf{z}_{r,k}^n$, measurement matrix $\mathbf{H}_{r,k}$, and observation noise covariance $\mathbf{R}_{r,k}$ can be calculated. The Kalman update equation (24) is then used to update the system state vector and covariance matrix.

3.6 GNSS Velocity Observation Equation

Differentiating Eq.(42) with respect to time, we obtain the GNSS antenna phase center velocity expressed in terms of the IMU velocity:

$$\mathbf{v}_G^n = \mathbf{v}_I^n + \mathbf{C}_b^n (\omega_{nb}^b \times) \ell_{ant}^b \quad (47)$$

where $\omega_{nb}^b = \omega_{ib}^b - \mathbf{C}_n^b \omega_{in}^n$.

The GNSS-measured velocity is expressed as $\tilde{\mathbf{v}}_G^n = \mathbf{v}_G^n - \mathbf{n}_v$, and the covariance of the velocity measurement noise \mathbf{n}_v is denoted as \mathbf{R}_v .

Substituting the error definition from Sec.3.2 into Eq.(47), we obtain the GNSS velocity measurement vector:

$$\begin{aligned}\delta \mathbf{z}_v^n &= \hat{\mathbf{r}}_G^n - \tilde{\mathbf{r}}_G^n \\ \delta \mathbf{z}_v^n &\approx \delta \mathbf{v}^n - \omega_{in}^n \times (\mathbf{C}_b^n \ell_{ant}^b) \times \phi - [\mathbf{C}_b^n (\ell_{ant}^b \times \omega_{ib}^b)] \times \phi - \mathbf{C}_b^n (\ell_{ant}^b \times) \delta \omega_{ib}^b + \mathbf{n}_v \\ \delta \mathbf{z}_v^n &= \mathbf{H}_v \delta \mathbf{x} + \mathbf{n}_v\end{aligned}\quad (48)$$

where the error-state of $\boldsymbol{\omega}_{\text{in}}^{\text{n}}$ is neglected due to its minimal magnitude during typical motion, and \mathbf{H}_v is the GNSS velocity measurement matrix:

$$\mathbf{H}_v = \begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{H}_{v,3} & -\mathbf{C}_{\text{b}}^{\text{n}}(\boldsymbol{\ell}_{\text{ant}}^{\text{b}} \times) & \mathbf{0}_3 & -\mathbf{C}_{\text{b}}^{\text{n}}(\boldsymbol{\ell}_{\text{ant}}^{\text{b}} \times) \text{diag}(\boldsymbol{\omega}_{\text{ib}}^{\text{b}}) & \mathbf{0}_3 \end{bmatrix} \quad (49)$$

where $\mathbf{H}_{v,3} = -\boldsymbol{\omega}_{\text{in}}^{\text{n}} \times (\mathbf{C}_{\text{b}}^{\text{n}} \boldsymbol{\ell}_{\text{ant}}^{\text{b}}) \times - [\mathbf{C}_{\text{b}}^{\text{n}}(\boldsymbol{\ell}_{\text{ant}}^{\text{b}} \times \boldsymbol{\omega}_{\text{ib}}^{\text{b}})] \times$.

The GNSS velocity measurement model is then employed to perform the system update by Eq.(24).

3.7 ODO/NHC Observation Equation

ODO/NHC measurements are designed for wheeled platforms, where the odometer (ODO) provides the forward speed v_{odo} and the non-holonomic constraint (NHC) enforces zero lateral and vertical velocities. To formulate the ODO/NHC observation equation, we define the vehicle frame, denoted as the v-frame, with forward-right-down axes and its origin at the ground contact point of the odometer wheel. The ODO/NHC measurements in the v-frame are given by $\tilde{\mathbf{v}}_{\text{odo,nhc}}^{\text{v}} = [v_{\text{odo}} \ 0 \ 0]^T - \mathbf{n}_{\text{odo,nhc}}$, where $\mathbf{n}_{\text{odo,nhc}}$ is the measurement noise and $\mathbf{R}_{\text{odo,nhc}}$ is the corresponding covariance matrix.

The INS-predicted velocity are the odometer wheel are expressed as:

$$\begin{aligned} \mathbf{v}_{\text{odo,nhc}}^{\text{v}} &= \mathbf{C}_{\text{b}}^{\text{v}} \mathbf{C}_{\text{n}}^{\text{b}} \left(\mathbf{v}_I^{\text{n}} + \mathbf{C}_{\text{b}}^{\text{n}}(\boldsymbol{\omega}_{\text{nb}}^{\text{b}} \times) \boldsymbol{\ell}_{\text{odo}}^{\text{b}} \right) \\ \mathbf{v}_{\text{odo,nhc}}^{\text{v}} &= \mathbf{C}_{\text{b}}^{\text{v}} \mathbf{C}_{\text{n}}^{\text{b}} \mathbf{v}_I^{\text{n}} + \mathbf{C}_{\text{b}}^{\text{v}}(\boldsymbol{\omega}_{\text{nb}}^{\text{b}} \times) \boldsymbol{\ell}_{\text{odo}}^{\text{b}} \end{aligned} \quad (50)$$

where $\mathbf{C}_{\text{b}}^{\text{v}}$ is the rotation matrix between the body and vehicle frames, and $\boldsymbol{\ell}_{\text{odo}}^{\text{b}}$ is the odometer lever arm in the body frame.

Applying error perturbation to Eq.(50), the ODO/NHC measurement vector is:

$$\begin{aligned} \delta \mathbf{z}_{\text{odo,nhc}}^{\text{v}} &= \tilde{\mathbf{v}}_{\text{odo,nhc}}^{\text{v}} - \mathbf{v}_{\text{odo,nhc}}^{\text{v}} \\ \delta \mathbf{z}_{\text{odo,nhc}}^{\text{v}} &\approx \mathbf{C}_{\text{b}}^{\text{v}} \mathbf{C}_{\text{n}}^{\text{b}} \delta \mathbf{v}^{\text{n}} - \mathbf{C}_{\text{b}}^{\text{v}} \mathbf{C}_{\text{n}}^{\text{b}}(\mathbf{v}^{\text{n}} \times) \boldsymbol{\phi} - \mathbf{C}_{\text{b}}^{\text{v}}(\boldsymbol{\ell}_{\text{odo}}^{\text{b}} \times) \mathbf{C}_{\text{n}}^{\text{b}}(\boldsymbol{\omega}_{\text{in}}^{\text{n}} \times) \boldsymbol{\phi} - \mathbf{C}_{\text{b}}^{\text{v}}(\boldsymbol{\ell}_{\text{odo}}^{\text{b}} \times) \delta \boldsymbol{\omega}_{\text{ib}}^{\text{b}} + \mathbf{n}_{\text{odo,nhc}} \\ \delta \mathbf{z}_{\text{odo,nhc}}^{\text{v}} &= \mathbf{H}_{\text{odo,nhc}} \delta \mathbf{x} + \mathbf{n}_{\text{odo,nhc}} \end{aligned} \quad (51)$$

where the error-state of $\boldsymbol{\omega}_{\text{in}}^{\text{n}}$ is also neglected, and the ODO/NHC measurement matrix $\mathbf{H}_{\text{odo,nhc}}$ is given by:

$$\mathbf{H}_{\text{odo,nhc}} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{C}_{\text{b}}^{\text{v}} \mathbf{C}_{\text{n}}^{\text{b}} & \mathbf{H}_{\text{odo,nhc},3} & -\mathbf{C}_{\text{b}}^{\text{v}}(\boldsymbol{\ell}_{\text{odo}}^{\text{b}} \times) & \mathbf{0}_3 & -\mathbf{C}_{\text{b}}^{\text{v}}(\boldsymbol{\ell}_{\text{odo}}^{\text{b}} \times) \text{diag}(\boldsymbol{\omega}_{\text{ib}}^{\text{b}}) & \mathbf{0}_3 \end{bmatrix} \quad (52)$$

where $\mathbf{H}_{\text{odo,nhc},3} = -\mathbf{C}_{\text{b}}^{\text{v}} \mathbf{C}_{\text{n}}^{\text{b}}(\mathbf{v}^{\text{n}} \times) - \mathbf{C}_{\text{b}}^{\text{v}}(\boldsymbol{\ell}_{\text{odo}}^{\text{b}} \times) \mathbf{C}_{\text{n}}^{\text{b}}(\boldsymbol{\omega}_{\text{in}}^{\text{n}} \times)$.

Finally, the ODO/NHC velocity measurement model is employed in Eq.(24) to update the system state vector and covariance matrix.

3.8 Error State Feedback

From the updated system error state vector $\delta \mathbf{x}_k$, the position error $\delta \mathbf{r}_k^{\text{n}}$, velocity error $\delta \mathbf{v}_k^{\text{n}}$, attitude error $\boldsymbol{\phi}_k$, IMU gyroscope bias error $\delta \mathbf{b}_{g,k}$, IMU accelerometer bias error $\delta \mathbf{b}_{a,k}$, IMU gyroscope scale factor error $\delta \mathbf{s}_{g,k}$, and IMU accelerometer scale factor error $\delta \mathbf{s}_{a,k}$ are extracted. The estimated error state is then fed back to the predicted navigation state $\hat{\mathbf{x}}$ as follows:

$$\begin{aligned} \mathbf{r}_k^{\text{n}} &= \hat{\mathbf{r}}_k^{\text{n}} - \delta \mathbf{r}_k^{\text{n}}, \quad \mathbf{p}_I = \hat{\mathbf{p}}_I - \mathbf{D}_R^{-1} \delta \mathbf{r}_k^{\text{n}} \\ \mathbf{v}_k^{\text{n}} &= \hat{\mathbf{v}}_k^{\text{n}} - \delta \mathbf{v}_k^{\text{n}} \\ \mathbf{C}_{\text{b},k}^{\text{n}} &= [\mathbf{I}_3 + \boldsymbol{\phi}_k \times] \hat{\mathbf{C}}_{\text{b},k}^{\text{n}} \\ \mathbf{b}_{g,k} &= \hat{\mathbf{b}}_{g,k} + \delta \mathbf{b}_{g,k} \\ \mathbf{b}_{a,k} &= \hat{\mathbf{b}}_{a,k} + \delta \mathbf{b}_{a,k} \\ \mathbf{s}_{g,k} &= \hat{\mathbf{s}}_{g,k} + \delta \mathbf{s}_{g,k} \\ \mathbf{s}_{a,k} &= \hat{\mathbf{s}}_{a,k} + \delta \mathbf{s}_{a,k} \end{aligned} \quad (53)$$

After error state feedback, it is assumed that the estimated navigation state no longer contains errors, and the system error state is reset to zero: $\delta \mathbf{x}_k = \mathbf{0}$.