6

Positioning by Proprioceptive Sensors and Environmental Sensors

This chapter focuses on positioning techniques based on proprioceptive sensors, such as inertial sensors and wheel speed sensors, as well as environmental sensors, including magnetometers and barometers. These sensors enable self-contained navigation using dead reckoning, an approach that estimates position based on recorded motion data from a known initial state (as illustrated in Figure 6.1, which provides an example of dead reckoning at sea). Proprioceptive sensors measure internal motion characteristics of a platform [1]. For instance, an inertial measurement unit (IMU), which consists of accelerometers and gyroscopes, can be used to compute a moving object's position by integrating acceleration and angular rate data over time. Similarly, wheel speed sensors (or encoders) provide velocity information that allows dead reckoning to track a mobile robot's displacement. Dead reckoning is particularly useful in indoor environments, where external localization signals (e.g., Wi-Fi, or UWB) may be unavailable or unreliable.

While inertial navigation and wheel odometry are commonly used, they have cumulative errors due to sensor drift, noise, and bias. Over time, small errors in acceleration, angular velocity, or wheel speed measurements accumulate, leading to significant position drift. To address this issue, sensor fusion strategies integrating data from multiple sources or motion constraints

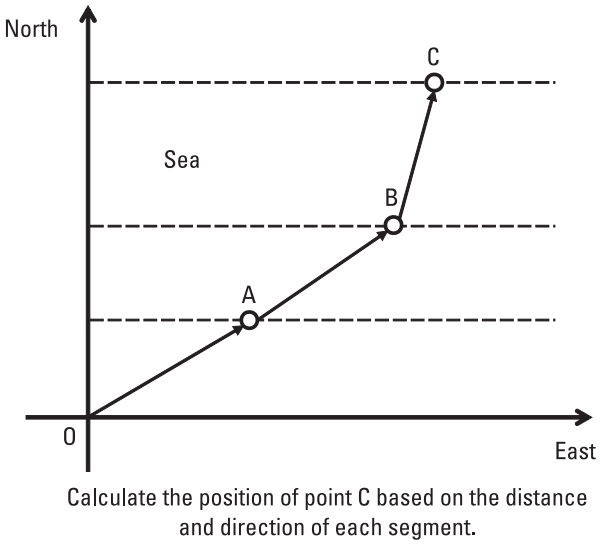

Figure 6.1 Dead reckoning at sea.

are integrated to improve positioning accuracy. For example, IMU and wheel speed sensor fusion can correct odometry errors caused by wheel slip, while zero-velocity updates (ZUPT) can help to mitigate drift in pedestrian navigation. Additionally, environmental sensors, such as magnetometers for heading estimation and barometers for altitude sensing, provide complementary information that enhances the accuracy of dead reckoning.

This chapter is organized as follows. Section 6.1 introduces inertial navigation, including the principles of inertial sensors and navigation equations that govern position estimation. Then it is followed by discussing wheel odometry and pedestrian inertial navigation. Section 6.4 introduces environmental sensors, including magnetometers for heading estimation and barometers for altitude sensing. Section 6.5 introduces error modeling and calibration. Section 6.6 explores drift mitigation techniques, covering initialization methods, and motion constraints. Finally, real-world examples and conclusions are provided.

6.1 Inertial Navigation

Inertial navigation is a self-contained navigation technique that employs measurements from inertial sensors to track the position, velocity, and attitude of an object relative to a known starting point [2]. Unlike other navigation

methods, it does not rely on external references and does not emit signals to the external environment. This technology finds extensive use in indoor scenarios where continuous positioning is essential, especially when other information sources are unavailable.

6.1.1 Inertial Sensors

Inertial sensors, comprising gyroscopes and accelerometers, provide essential motion data by measuring angular velocity and acceleration, respectively. These sensors form the core components of an IMU, which enables motion tracking in 3-D space. In most navigation applications, an IMU consists of one gyroscope and one accelerometer per axis, allowing it to capture 6 degrees of freedom (6-DOF) motion.

IMUs are classified based on their performance characteristics, which determine their suitability for different applications. Table 6.1 presents a classification of IMUs by grade, detailing their typical position error, bias stability, and cost.

Strategic-grade and navigation-grade IMUs are used in weapons systems, military applications, and high-precision mapping, while commercial-grade IMUs are widely adopted for indoor positioning and consumer electronics due to their low cost and compact size [4]. These commercial IMUs rely heavily on microelectromechanical systems (MEMS) technology, which enables significant miniaturization and integration into mobile devices, robots, and even wearable systems.

Table 6.1
Classification of IMUs Based on Performance [3]

Grade	Strategic	Navigation	Tactical	Commercial
Position error	30–100 m/h	1 nmi*/h or 0.5 m/s	10–20 nmi/h	Large variation
Gyroscope bias instability	0.0001–0.001°/h	<0.01°/h	1–10°/h	0.1°/s
Accelerometer bias instability	0.1–1 μg	<100 μg	1–5 mg	100–1,000 μg
Cost	≈$1 million	≈$100,000	$2,000–$50,000	<$1 (accelerometers) <$10 (gyroscopes)

*1 nautical mile (nmi) ≈ 1,851m.

MEMS-based inertial sensors are commonly used in indoor positioning due to their small size, low power consumption, and affordability. They operate based on physical principles that enable the measurement of motion parameters:

MEMS gyroscopes use the Coriolis effect to measure angular velocity. When a mass m moves with velocity $\mathbf{v}$ within a rotating system, a Coriolis force $\mathbf{F}_c$ is exerted perpendicular to the motion direction. This force is given by:

$$\mathbf{F}_c = -2 \cdot m \cdot (\omega \times \mathbf{v}) \tag{6.1}$$

where ω is the angular velocity vector of the rotating object. MEMS gyroscopes use vibrating elements to detect this force, which is then converted into an electrical signal proportional to angular velocity. Figure 6.2 illustrates the working principle of a MEMS gyroscope.

MEMS accelerometers measure specific force by detecting the displacement of a proof mass, which is suspended by a spring mechanism. When subjected to acceleration, the proof mass shifts, and the resulting deflection is proportional to the applied acceleration. The displacement is measured using capacitive, piezoelectric, or optical sensing techniques, as shown in Figure 6.3. MEMS accelerometers provide motion tracking, but their measurements include gravitational acceleration, requiring compensation in positioning applications.

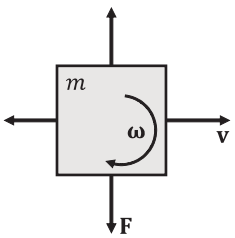

Figure 6.2 A vibrating mass gyroscope [5].

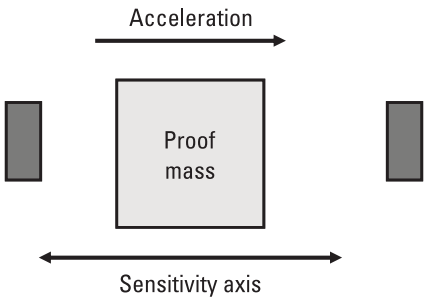

Figure 6.3 A mechanical accelerometer [6].

6.1.2 Inertial Navigation Equations

Inertial navigation is a dead reckoning technique that estimates an object's position, velocity, and attitude based on measurements from gyroscopes and accelerometers [7]. A system that integrates inertial sensors for navigation is referred to as an Inertial Navigation System (INS). INS can be categorized into two types:

1. *Stable-platform INS:* The inertial sensors are mounted on a gimballed platform that remains mechanically stabilized with respect to an inertial reference frame. While this setup reduces computational complexity, it comes with high cost, large volume, and complex maintenance requirements.
2. *Strap-down INS (SINS):* The IMU is directly attached to the moving object, and its orientation is updated numerically using onboard computing. This removes the need for mechanical gimbals, reducing cost and increasing reliability. Due to these advantages, strap-down inertial navigation is more widely adopted in modern applications [3].

MEMS-based IMUs are commonly used for indoor navigation due to their small size and affordability. However, MEMS-IMUs have higher noise levels and cannot detect the Earth's rotation rate (≈15°/h). Additionally, in low-dynamic environments such as indoor settings, traditional INS algorithms have been simplified to accommodate the limited motion characteristics of the user or robot [6]. The generalized framework of strap-down inertial navigation using MEMS-IMUs is shown in Figure 6.4. This framework consists of three core modules: attitude update (determining orientation), velocity update (determining speed and direction), and position update (determining displacement).

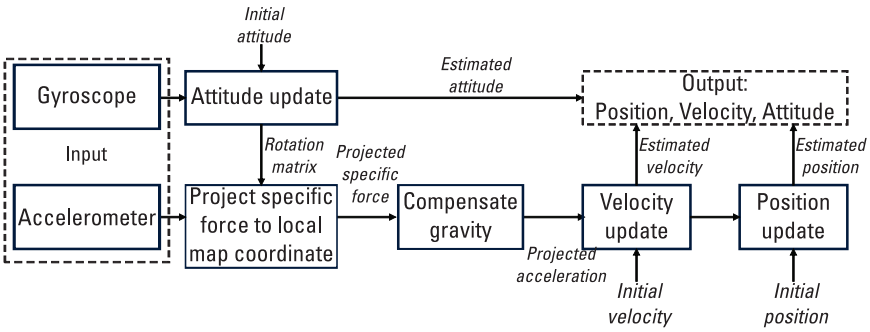

Figure 6.4 The generalized framework of strap-down inertial navigation based on MEMS-IMU.

6.1.2.1 Attitude Update

Attitude estimation is the first step in strap-down inertial navigation. The attitude of an object is commonly represented using Euler angles, rotation matrices (direction cosine matrices (DCM)), and quaternions. Among these, quaternions are preferred due to their numerical stability, computational efficiency, and lack of singularities (which occur in Euler angles) [8]. The kinematic equation for a quaternion-based attitude update is given by:

$$\dot{\mathbf{q}}_{k-1} = \frac{1}{2} \cdot \mathbf{\Omega}_{k-1} \cdot \mathbf{q}_{k-1} \tag{6.2}$$

where $\mathbf{q}_k$ is the quaternion at the instant t_k. $\mathbf{\Omega}_{k-1}$ is a skew-symmetric matrix constructed from the angular velocity vector $\boldsymbol{\omega}^B_{k-1}$, which is measured by the gyroscope. At the initial instant, the quaternion $\mathbf{q}_0$ is determined through the initial alignment. In general, for navigation-grade IMUs and above, all attitudes (roll, pitch, and heading) can be estimated using the gyroscope and accelerometer. For lower-grade IMUs, however, the heading must be estimated with the aid of additional sensors, such as a magnetometer [6].

The angular velocity matrix $\mathbf{\Omega}_{k-1}$ is given by:

$$\mathbf{\Omega}_{k-1} = \begin{bmatrix} 0 & -\omega^B_{x,k-1} & -\omega^B_{y,k-1} & -\omega^B_{z,k-1} \\ \omega^B_{x,k-1} & 0 & \omega^B_{z,k-1} & -\omega^B_{y,k-1} \\ \omega^B_{y,k-1} & -\omega^B_{z,k-1} & 0 & \omega^B_{x,k-1} \\ \omega^B_{z,k-1} & \omega^B_{y,k-1} & -\omega^B_{x,k-1} & 0 \end{bmatrix} \tag{6.3}$$

where $\boldsymbol{\omega}^B_{k-1} = [\omega^B_{x,k-1}, \omega^B_{y,k-1}, \omega^B_{z,k-1}]^{\mathrm{T}}$ is the angular velocity expressed in the local body coordinate.

The quaternion update shown in (6.2) can be rewritten in discrete form:

$$\mathbf{q}_k = \mathbf{q}_{k-1} + \frac{1}{2} \cdot \mathbf{\Omega}_{k-1} \cdot \mathbf{q}_{k-1} \cdot \Delta t \tag{6.4}$$

where $\mathbf{q}_k$ is the current attitude estimate. In practical systems, the angular velocity ω is measured by gyroscopes includes bias and noise:

$$\tilde{\boldsymbol{\omega}}^B_{k-1} = \boldsymbol{\omega}^B_{k-1} + \mathbf{b}^B_G + \mathbf{n}_G \tag{6.5}$$

where $\boldsymbol{\omega}^B_{k-1}$ is the true value of angular velocity. $\mathbf{b}^B_G$ is the gyroscope bias (assumed constant over short durations, meaning $\dot{\mathbf{b}}^B_G = 0$). $\mathbf{n}_G$ is the Gaussian white noise; each component in $\mathbf{n}_G$ satisfies $N(0, \sigma^2_G)$.

To improve accuracy, the gyroscope bias $\mathbf{b}_G^B$ is estimated during static periods by averaging multiple gyroscope readings. The corrected angular velocity is:

$$\hat{\boldsymbol{\omega}}_{k-1}^B = \tilde{\boldsymbol{\omega}}_{k-1}^B - \hat{\mathbf{b}}_G^B \tag{6.6}$$

which is then used in the quaternion update equations (6.2). Finally, the current estimate of the attitude $\hat{\mathbf{q}}_k$ can be obtained.

6.1.2.2 Velocity Update

Once the attitude is determined, the next step is to compute velocity. The velocity update equation is given by:

$$\mathbf{v}_k = \mathbf{v}_{k-1} + \left(\left(\mathbf{C}_M^B\right)^{\mathrm{T}} \mathbf{a}_{k-1}^B - \mathbf{g}^M\right)\Delta t \tag{6.7}$$

where $\mathbf{v}_k$ is the velocity at the instant t_k. $\mathbf{a}_{k-1}^B$ is the specific force (including motion acceleration and gravity) measured by the accelerometer, and $\mathbf{C}_M^B$ is the rotation matrix from the local map coordinate to local body coordinate. $\mathbf{g}^M$ is the gravity vector, expressed as $\mathbf{g}^M = [0, 0, g]^{\mathrm{T}}$ expressed in local map coordinate. The gravity varies with the location on Earth, and its calculation can be approximated as:

$$g = 9.7803 \cdot \left[1 + 0.0053024 \cdot \sin^2\psi - 0.000005 \cdot \sin^2\left(2 \cdot \psi\right)\right] \tag{6.8}$$

where ψ is the latitude. In practical systems, the specific force is given by accelerometers, which is modeled as:

$$\tilde{\mathbf{a}}_{k-1}^B = \mathbf{a}_{k-1}^B + \mathbf{b}_A^B + \mathbf{n}_A \tag{6.9}$$

where $\mathbf{a}_{k-1}^B$ is the true value of the specific force. $\mathbf{b}_A^B$ is the accelerometer bias (assumed constant over short durations, meaning $\dot{\mathbf{b}}_A^B = 0$).$\mathbf{n}_A$ represents the white noise in accelerometers and assumed as Gaussian; each component in $\mathbf{n}_A$ satisfies $N(0, \sigma_A^2)$.

Unlike gyroscope bias, the accelerometer bias is unavailable even in static situations, as the gravity in the local body frame can affect the estimation. Therefore, the measured values of accelerometers $\tilde{\mathbf{a}}_{k-1}^B$ will be directly substituted into (6.7) when there is no external reference for bias estimation, and then the current estimate of the velocity $\hat{\mathbf{v}}_k$ can be obtained.

6.1.2.3 Position Update

The position update is achieved with the estimated velocity, which is shown as:

$$\hat{\mathbf{p}}_k = \hat{\mathbf{p}}_{k-1} + \hat{\mathbf{v}}_{k-1} \cdot \Delta t \tag{6.10}$$

where $\hat{\mathbf{p}}_k$ and $\hat{\mathbf{p}}_{k-1}$ are the current and previous estimates of the position.

6.2 Wheel Odometry for Mobile Robots

In inertial navigation, data from accelerometers is used to calculate speed, which is then used to estimate position. This method is versatile as it can work in any environment and with different objects. However, it has a significant drawback: the position error increases quickly over time because of the double integration. Researchers have used wheel speed sensors to reduce this error for position estimation [9, 10].

Wheel speed sensors measure the rotation of the wheels of a mobile robot to determine its speed and distance traveled. There are two types of wheel speed sensors: passive and active. Passive sensors are inductive analog devices that have a low signal-to-noise ratio (SNR) and are susceptible to vibration and interference. Active sensors are digital Hall devices that are widely used in navigation but are more expensive [11]. Wheel speed sensors can be used in two ways for positioning:

1. *Dead reckoning:* The speed or distance measured in the local body coordinate system is converted to the local map coordinate system, and then the converted values are summed up to calculate the position. If the robot has only one wheel speed sensor, it needs other sensors, such as a gyroscope or IMU, to provide the heading for the conversion.
2. *Information fusion:* The speed measured by the wheel speed sensors is combined with other sensors, such as IMU, as an observation. Kalman filter-based methods [12, 13] are often used for the integration.

Next we will introduce the two methods in detail.

6.2.1 Dead Reckoning Based on the Wheel Speed Sensor

Dead reckoning is a fundamental technique in robot navigation, estimating the robot's position over time using motion sensor data. In a system equipped with only a single wheel speed sensor, the direct measurement available is the radial speed of the wheel. However, this alone is insufficient for dead reckoning, as additional information is required to determine the robot's full velocity vector in the local map coordinate frame.

To achieve accurate dead reckoning, the system requires attitude measurements (e.g., from an IMU or other orientation sensors) to convert the wheel's radial speed into velocity components in the global reference frame. The generalized framework for dead reckoning based on speed measurements is illustrated in Figure 6.5.

In practical implementations, the attitude sensor (e.g., an IMU) is often physically mounted separately from the wheel speed sensor. As a result, the spatial transformation between these sensors must be accounted for. Figure 6.6 states the spatial relationship between the attitude measurement sensor (A-frame) and the wheel speed sensor (S-frame).

From Figure 6.6, we observe that there exists both a rotation between the speed sensor frame (S-frame) and the attitude sensor frame (A-frame) and a translation offset between these two sensors. Thus, to correctly transform the wheel speed measurement into the attitude coordinate frame, we must consider these extrinsic parameters (sensor rotation and translation). If the wheel speed sensor measures $\tilde{v}_{k-1}^{S}$, the measurement in the speed coordinate

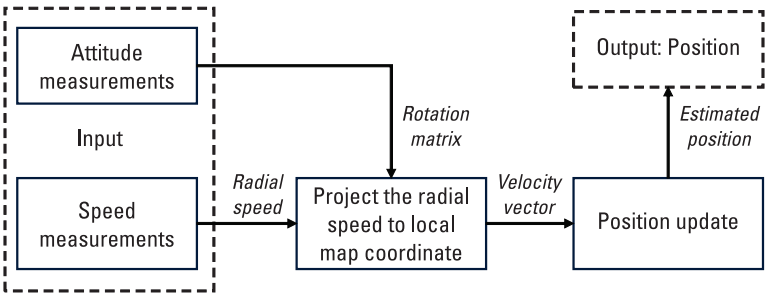

Figure 6.5 The generalized framework of dead reckoning based on speed measurements.

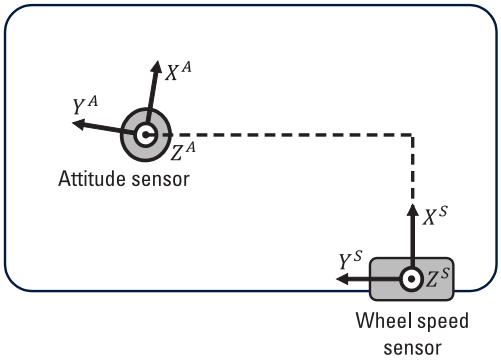

Figure 6.6 Spatial relationship between the attitude sensor and wheel speed sensor.

is $\tilde{\mathbf{v}}_{k-1}^{S} = [0, \tilde{v}_{k-1}^{S}, 0]^{\mathrm{T}}$ based on the nonholonomic constraint (NHC), which assumes that the lateral and vertical speeds of the robot are zero unless it slides or jumps. Let $\mathbf{l}^{A} = [l_x^A, l_y^A, l_z^A]^{\mathrm{T}}$ be the translation between the attitude and wheel speed sensors, which represents the origin of the speed coordinate in the attitude coordinate. Also, let $\mathbf{C}_S^A$ be the rotation from the speed to the attitude coordinate. Then $\tilde{\mathbf{v}}_{k-1}^{S}$ can be transformed to the attitude coordinate as follows:

$$\tilde{\mathbf{v}}_{k-1}^{A} = \hat{\mathbf{C}}_S^A \cdot \tilde{\mathbf{v}}_{k-1}^{S} - \left[\tilde{\boldsymbol{\omega}}^{A}\right]_{\times} \cdot \hat{\mathbf{l}}^{A} \tag{6.11}$$

where $\hat{\mathbf{C}}_S^A$ and $\hat{\mathbf{l}}^{A}$ are estimated extrinsic parameters. $\tilde{\boldsymbol{\omega}}^{A}$ is the angular velocity in the attitude coordinate system, which the attitude sensor can measure. $[\tilde{\boldsymbol{\omega}}^{A}]_{\times}$ can be expressed as:

$$\left[\tilde{\boldsymbol{\omega}}^{A}\right]_{\times} = \begin{bmatrix} 0 & -\tilde{\omega}_z^A & \tilde{\omega}_y^A \\ \tilde{\omega}_z^A & 0 & -\tilde{\omega}_x^A \\ -\tilde{\omega}_y^A & \tilde{\omega}_x^A & 0 \end{bmatrix} \tag{6.12}$$

Based on the attitude measurement, the rotation matrix $\tilde{\mathbf{C}}_A^M$ can be calculated.

$$\tilde{\mathbf{C}}_A^M = \begin{bmatrix} \left(1-2\cdot\tilde{q}_2^2-2\cdot\tilde{q}_3^2\right) & 2\cdot\left(\tilde{q}_1\cdot\tilde{q}_2-\tilde{q}_0\cdot\tilde{q}_3\right) & 2\cdot\left(\tilde{q}_1\cdot\tilde{q}_3+\tilde{q}_0\cdot\tilde{q}_2\right) \\ 2\cdot\left(\tilde{q}_1\cdot\tilde{q}_2+\tilde{q}_0\cdot\tilde{q}_3\right) & \left(1-2\cdot\tilde{q}_1^2-2\cdot\tilde{q}_3^2\right) & 2\cdot\left(\tilde{q}_2\cdot\tilde{q}_3-\tilde{q}_0\cdot\tilde{q}_1\right) \\ \left(\tilde{q}_1\cdot\tilde{q}_3-\tilde{q}_0\cdot\tilde{q}_2\right) & 2\cdot\left(\tilde{q}_2\cdot\tilde{q}_3+\tilde{q}_0\cdot\tilde{q}_1\right) & \left(1-2\cdot\tilde{q}_1^2-2\cdot\tilde{q}_2^2\right) \end{bmatrix} \tag{6.13}$$

where $\tilde{\mathbf{q}}_k = [\tilde{q}_0, \tilde{q}_1, \tilde{q}_2, \tilde{q}_3]^{\mathrm{T}}$ represents the quaternion obtained from the attitude measurement. Then $\tilde{\mathbf{v}}_{k-1}^{A}$ can be converted to the local map coordinate system as follows:

$$\hat{\mathbf{v}}_{k-1} = \tilde{\mathbf{C}}_A^M \cdot \tilde{\mathbf{v}}_{k-1}^{A} \tag{6.14}$$

Finally, the position estimate can be obtained, which is shown as:

$$\hat{\mathbf{p}}_k = \hat{\mathbf{p}}_{k-1} + \hat{\mathbf{v}}_{k-1} \cdot \Delta t \tag{6.15}$$

6.2.2 IMU and Wheel Speed Sensor Integrated Navigation

The wheel speed sensor measurement can also be used to correct the inertial navigation in a Kalman filter for state estimation, the details of which can be

found in Chapter 3. Figure 6.7 shows the general framework of the integrated navigation with IMU and wheel speed sensor. The IMU data is used for inertial navigation and state prediction. The radial speed and NHC are used as the observation. The output consists of the position, velocity, and attitude. Next we will explain the Kalman filter formulation in detail.

The complete system state at the instant t_k consists of 16 states, which is shown as:

$$\mathbf{x}_k = \left[\mathbf{p}_k, \mathbf{v}_k, \mathbf{q}_k, \mathbf{b}^B_{G,\,k}, \mathbf{b}^B_{A,k}\right]^{\mathrm{T}} \tag{6.16}$$

where $\mathbf{x}_k$ includes position, velocity, attitude, gyroscope bias, and accelerometer bias. First, IMU data is utilized for inertial navigation to predict the state. The state transition equation can be expressed as:

$$\mathbf{x}_k = \mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}\right) + \mathbf{w}_{k-1} \tag{6.17}$$

where $\mathbf{u}_{k-1}$ is the input vector from IMU measurements and $\mathbf{w}_{k-1}$ is the process noise. According to (6.4), the state transition of attitude can be written as:

$$\mathbf{q}_k = \mathbf{q}_{k-1} + \frac{1}{2} \cdot \left(\tilde{\boldsymbol{\Omega}}^B_{k-1} - \boldsymbol{\Omega}^B_{G,k-1} - \boldsymbol{\Omega}_G\right) \cdot \mathbf{q}_{k-1} \cdot \Delta t \tag{6.18}$$

where $\tilde{\boldsymbol{\Omega}}^B_{k-1}$, $\boldsymbol{\Omega}^B_{G,k-1}$, and $\boldsymbol{\Omega}_G$ are formed from the components of the angular rate $\tilde{\boldsymbol{\omega}}^B_{k-1}$, angular rate bias $\mathbf{b}^B_{G,k-1}$, and noise $\mathbf{n}_G$. Then (6.18) can be rewritten as:

$$\begin{aligned} \mathbf{q}_k &= \mathbf{q}_{k-1} + \frac{1}{2} \cdot \left(\tilde{\boldsymbol{\Omega}}^B_{k-1} - \boldsymbol{\Omega}^B_{G,k-1}\right) \cdot \mathbf{q}_{k-1} \cdot \Delta t - \frac{1}{2} \cdot \boldsymbol{\Omega}_G \cdot \mathbf{q}_{k-1} \cdot \Delta t \\ \mathbf{q}_k &= \left[\mathbf{I}_4 + \frac{1}{2} \cdot \left(\tilde{\boldsymbol{\Omega}}^B_{k-1} - \boldsymbol{\Omega}^B_{G,\,k-1}\right) \cdot \Delta t\right] \cdot \mathbf{q}_{k-1} - \frac{\Delta t}{2} \cdot \boldsymbol{\Xi} \cdot \mathbf{n}_G \end{aligned} \tag{6.19}$$

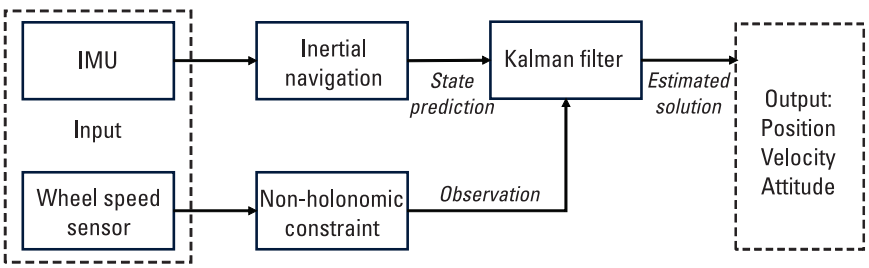

Figure 6.7 The generalized framework of IMU and wheel speed sensor integrated navigation.

where

$$\Xi = \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix} \tag{6.20}$$

The corresponding covariance matrix can be expressed as:

$$\begin{aligned} \boldsymbol{\Sigma}^q &= \frac{\Delta t}{2} \cdot \Xi \cdot \mathbf{n}_G \cdot \left(\frac{\Delta t}{2} \cdot \Xi \cdot \mathbf{n}_G \right)^{\mathrm{T}} \\ \boldsymbol{\Sigma}^q &= \left(\frac{\sigma_G \cdot \Delta t}{2} \right)^2 \cdot \begin{bmatrix} 1-q_0^2 & -q_0 \cdot q_1 & -q_0 \cdot q_2 & -q_0 \cdot q_3 \\ -q_0 \cdot q_1 & 1-q_1^2 & -q_1 \cdot q_2 & -q_1 \cdot q_3 \\ -q_0 \cdot q_2 & -q_1 \cdot q_2 & 1-q_2^2 & -q_2 \cdot q_3 \\ -q_0 \cdot q_3 & -q_1 \cdot q_3 & -q_2 \cdot q_3 & 1-q_3^2 \end{bmatrix} \end{aligned} \tag{6.21}$$

According to (6.7), the state transition of velocity can be written as:

$$\begin{aligned} \mathbf{v}_k &= \mathbf{v}_{k-1} + \left[\mathbf{C}_M^{B\ \mathrm{T}} \cdot \left(\tilde{\mathbf{a}}_{k-1}^B - \mathbf{b}_{A,\,k-1}^B - \mathbf{n}_A \right) - \mathbf{g}^M \right] \cdot \Delta t \\ \mathbf{v}_k &= \mathbf{v}_{k-1} + \left[\mathbf{C}_M^{B\ \mathrm{T}} \cdot \left(\tilde{\mathbf{a}}_{k-1}^B - \mathbf{b}_{A,\,k-1}^B \right) - \mathbf{g}^M \right] \cdot \Delta t - \mathbf{C}_M^{B\ \mathrm{T}} \cdot \mathbf{n}_A \cdot \Delta t \end{aligned} \tag{6.22}$$

The corresponding covariance matrix can be expressed as:

$$\begin{aligned} \boldsymbol{\Sigma}^v &= \mathbf{C}_M^{B\ \mathrm{T}} \cdot \mathbf{n}_A \cdot \Delta t \cdot \left(\mathbf{C}_M^{B\ \mathrm{T}} \cdot \mathbf{n}_A \cdot \Delta t \right)^{\mathrm{T}} \\ \boldsymbol{\Sigma}^v &= \left(\sigma_A \cdot \Delta t \right)^2 \cdot \mathbf{I}_3 \end{aligned} \tag{6.23}$$

According to (6.15), the state transition of position can be written as:

$$\mathbf{p}_k = \mathbf{p}_{k-1} + \mathbf{v}_{k-1} \cdot \Delta t + \mathbf{C}_M^{B\ \mathrm{T}} \cdot \mathbf{n}_A \cdot \Delta t^2 \tag{6.24}$$

The corresponding covariance matrix can be expressed as:

$$\begin{aligned} \boldsymbol{\Sigma}^p &= \mathbf{C}_M^{B\ \mathrm{T}} \cdot \mathbf{n}_A \cdot \Delta t^2 \cdot \left(\mathbf{C}_M^{B\ \mathrm{T}} \cdot \mathbf{n}_A \cdot \Delta t^2 \right)^{\mathrm{T}} \\ \boldsymbol{\Sigma}^p &= \left(\sigma_A \cdot \Delta t^2 \right)^2 \cdot \mathbf{I}_3 \end{aligned} \tag{6.25}$$

Therefore, $\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1})$ and $\mathbf{w}_{k-1}$ can be expressed as:

$$\mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}\right) = \begin{bmatrix} \mathbf{p}_{k-1} + \mathbf{v}_{k-1} \cdot \Delta t \\ \mathbf{v}_{k-1} + \left[\mathbf{C}_M^{B\,\mathrm{T}} \cdot \left(\tilde{\mathbf{a}}_{k-1}^B - \mathbf{b}_{A,\,k-1}^B\right) - \mathbf{g}^M\right] \cdot \Delta t \\ \left[\mathbf{I}_4 + \frac{1}{2} \cdot \left(\tilde{\mathbf{\Omega}}_{k-1}^B - \mathbf{\Omega}_{G,\,k-1}^B\right) \cdot \Delta t\right] \cdot \mathbf{q}_{k-1} \\ \mathbf{I}_3 \\ \mathbf{I}_3 \end{bmatrix} \tag{6.26}$$

$$\mathbf{w}_{k-1} = \begin{bmatrix} \mathbf{C}_M^{B\,\mathrm{T}} \cdot \mathbf{n}_A \cdot \Delta t^2 \\ -\mathbf{C}_M^{B\,\mathrm{T}} \cdot \mathbf{n}_A \cdot \Delta t \\ -\frac{\Delta t}{2} \cdot \Xi \cdot \mathbf{n}_G \\ 0_{3\times1} \\ 0_{3\times1} \end{bmatrix} \tag{6.27}$$

According to (6.26), the process Jacobian can be expressed as:

$$\mathbf{F}_{k-1} = \frac{\partial \mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}\right)}{\partial \mathbf{x}_{k-1}}$$

$$\mathbf{F}_{k-1} = \mathbf{I}_{16} + \begin{bmatrix} 0_3 & \mathbf{I}_3 & 0_{3\times4} & 0_3 & 0_3 \\ 0_3 & 0_3 & \partial\mathbf{v}\,\partial\mathbf{q} & 0_3 & -\mathbf{C}_M^{B\,\mathrm{T}} \\ 0_{4\times3} & 0_{4\times3} & \frac{1}{2} \cdot \left(\tilde{\mathbf{\Omega}}_{k-1}^B - \mathbf{\Omega}_{G,k-1}^B\right) & -\frac{1}{2} \cdot \Xi & 0_{4\times3} \\ 0_3 & 0_3 & 0_{3\times4} & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_{3\times4} & 0_3 & 0_3 \end{bmatrix} \cdot \Delta t \tag{6.28}$$

where

$$\partial\mathbf{v}\,\partial\mathbf{q} = 2 \cdot \mathbf{Q}_F \cdot \begin{bmatrix} 0 & \left(\mathbf{a}_{k-1}^B\right)^{\mathrm{T}} \\ \mathbf{a}_{k-1}^B & -\left[\mathbf{a}_{k-1}^B\right] \end{bmatrix} \tag{6.29}$$

where

$$\mathbf{Q}_F = \begin{bmatrix} q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \tag{6.30}$$

$$\mathbf{a}_{k-1}^B = \tilde{\mathbf{a}}_{k-1}^B - \mathbf{b}_{A,\,k-1}^B$$

The process noise covariance can be expressed as:

$$\mathbf{Q}_{k-1} = \begin{bmatrix} \boldsymbol{\Sigma}^p & 0_3 & 0_{3\times4} & 0_3 & 0_3 \\ 0_3 & \boldsymbol{\Sigma}^v & 0_{3\times4} & 0_3 & 0_3 \\ 0_{4\times3} & 0_{4\times3} & \boldsymbol{\Sigma}^q & 0_{4\times3} & 0_{4\times3} \\ 0_3 & 0_3 & 0_{3\times4} & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_{3\times4} & 0_3 & 0_3 \end{bmatrix} \tag{6.31}$$

Therefore, the covariance matrix prediction can be expressed as:

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1} \cdot \mathbf{P}_{k-1|k-1} \cdot \mathbf{F}_{k-1}^{\mathrm{T}} + \mathbf{Q}_{k-1} \tag{6.32}$$

The measurement model can be expressed as:

$$\begin{aligned} \mathbf{z}_k &= \mathbf{h}\left(\mathbf{x}_k\right) + \mathbf{r}_k \\ \tilde{\mathbf{v}}_k^A &= \mathbf{C}_M^A \cdot \mathbf{v}_k + \mathbf{r}_k \end{aligned} \tag{6.33}$$

$\mathbf{C}_M^A$ can be written as:

$$\mathbf{C}_M^A = \begin{bmatrix} \left(1-2\cdot q_2^2-2\cdot q_3^2\right) & 2\cdot\left(q_1\cdot q_2+q_0\cdot q_3\right) & 2\cdot\left(q_1\cdot q_3-q_0\cdot q_2\right) \\ 2\cdot\left(q_1\cdot q_2-q_0\cdot q_3\right) & \left(1-2\cdot q_1^2-2\cdot q_3^2\right) & 2\cdot\left(q_2\cdot q_3+q_0\cdot q_1\right) \\ 2\cdot\left(q_1\cdot q_3+q_0\cdot q_2\right) & 2\cdot\left(q_2\cdot q_3-q_0\cdot q_1\right) & \left(1-2\cdot q_1^2-2\cdot q_2^2\right) \end{bmatrix} \tag{6.34}$$

where $\mathbf{q}_k = [q_0, q_1, q_2, q_3]^{\mathrm{T}}$. $\mathbf{C}_M^A \cdot \mathbf{v}_k$ can be expressed as:

$$\begin{aligned} \left(\mathbf{C}_M^A \cdot \mathbf{v}_k\right)_1 &= \left(1-2\cdot q_2^2-2\cdot q_3^2\right)\cdot v_{x,k} + 2\cdot\left(q_1\cdot q_2+q_0\cdot q_3\right)\cdot v_{y,k} \\ &\quad + 2\cdot\left(q_1\cdot q_3-q_0\cdot q_2\right)\cdot v_{z,k} \\ \left(\mathbf{C}_M^A \cdot \mathbf{v}_k\right)_2 &= 2\cdot\left(q_1\cdot q_2-q_0\cdot q_3\right)\cdot v_{x,k} + \left(1-2\cdot q_1^2-2\cdot q_3^2\right)\cdot v_{y,k} \\ &\quad + 2\cdot\left(q_2\cdot q_3+q_0\cdot q_1\right)\cdot v_{z,k} \end{aligned}$$

$$\begin{aligned}\left(\mathbf{C}_M^A \cdot \mathbf{v}_k\right)_3 &= 2\cdot\left(q_1\cdot q_3 + q_0\cdot q_2\right)\cdot v_{x,k} + 2\cdot\left(q_2\cdot q_3 - q_0\cdot q_1\right)\cdot v_{y,k} \\ &+ \left(1-2\cdot q_1^2 - 2\cdot q_2^2\right)\cdot v_{z,k}\end{aligned} \tag{6.35}$$

where $(\mathbf{C}_M^A \cdot \mathbf{v}_k)_i$ is the ith element of $\mathbf{C}_M^A \cdot \mathbf{v}_k$, $\mathbf{v}_k = [v_{x,k}, v_{y,k}, v_{z,k}]^{\mathrm{T}}$. Taking the derivative of $\mathbf{v}_k$ and $\mathbf{q}_k$ in (6.35) yields the following form:

$$\frac{\partial\left(\mathbf{C}_M^A \cdot \mathbf{v}_k\right)}{\partial \mathbf{v}_k} = \begin{bmatrix} \left(1-2\cdot q_2^2 - 2\cdot q_3^2\right) & 2\cdot\left(q_1\cdot q_2 + q_0\cdot q_3\right) & 2\cdot\left(q_1\cdot q_3 - q_0\cdot q_2\right) \\ 2\cdot\left(q_1\cdot q_2 - q_0\cdot q_3\right) & \left(1-2\cdot q_2^2 - 2\cdot q_3^2\right) & 2\cdot\left(q_2\cdot q_3 + q_0\cdot q_1\right) \\ 2\cdot\left(q_1\cdot q_3 + q_0\cdot q_2\right) & 2\cdot\left(q_2\cdot q_3 - q_0\cdot q_1\right) & \left(1-2\cdot q_1^2 - 2\cdot q_2^2\right) \end{bmatrix} \tag{6.36}$$

$$\frac{\partial\left(\mathbf{C}_M^A \cdot \mathbf{v}_k\right)}{\partial \mathbf{q}_k} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} & \mathbf{B}_{13} & \mathbf{B}_{14} \\ \mathbf{B}_{21} & \mathbf{B}_{22} & \mathbf{B}_{23} & \mathbf{B}_{24} \\ \mathbf{B}_{31} & \mathbf{B}_{32} & \mathbf{B}_{33} & \mathbf{B}_{34} \end{bmatrix}$$

$$\begin{aligned}
\mathbf{B}_{11} &= 2\cdot q_3\cdot v_{y,k} - 2\cdot q_2\cdot v_{z,k} \\
\mathbf{B}_{12} &= 2\cdot q_2\cdot v_{y,k} + 2\cdot q_3\cdot v_{z,k} \\
\mathbf{B}_{13} &= -4\cdot q_2\cdot v_{x,k} + 2\cdot q_1\cdot v_{y,k} - 2\cdot q_0\cdot v_{z,k} \\
\mathbf{B}_{14} &= -4\cdot q_3\cdot v_{x,k} + 2\cdot q_0\cdot v_{y,k} + 2\cdot q_1\cdot v_{z,k} \\
\mathbf{B}_{21} &= -2\cdot q_3\cdot v_{x,k} + 2\cdot q_1\cdot v_{z,k} \\
\mathbf{B}_{22} &= 2\cdot q_2\cdot v_{x,k} - 4\cdot q_1\cdot v_{y,k} + 2\cdot q_0\cdot v_{z,k} \\
\mathbf{B}_{23} &= 2\cdot q_1\cdot v_{x,k} + 2\cdot q_3\cdot v_{z,k} \\
\mathbf{B}_{24} &= -2\cdot q_0\cdot v_{x,k} - 4\cdot q_3\cdot v_{y,k} + 2\cdot q_2\cdot v_{z,k} \\
\mathbf{B}_{31} &= 2\cdot q_2\cdot v_{x,k} - 2\cdot q_1\cdot v_{y,k} \\
\mathbf{B}_{32} &= 2\cdot q_3\cdot v_{x,k} - 2\cdot q_0\cdot v_{y,k} - 4\cdot q_1\cdot v_{z,k} \\
\mathbf{B}_{33} &= 2\cdot q_0\cdot v_{x,k} + 2\cdot q_3\cdot v_{y,k} - 4\cdot q_2\cdot v_{z,k} \\
\mathbf{B}_{34} &= 2\cdot q_1\cdot v_{x,k} + 2\cdot q_2\cdot v_{y,k}
\end{aligned} \tag{6.37}$$

Then the measurement matrix can be expressed as:

$$\mathbf{H}_k = \begin{bmatrix} 0_{3\times3} & \dfrac{\partial\left(\mathbf{C}_M^A \cdot \mathbf{v}_k\right)}{\partial \mathbf{v}_k} & \dfrac{\partial\left(\mathbf{C}_M^A \cdot \mathbf{v}_k\right)}{\partial \mathbf{q}_k} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{6.38}$$

Let us suppose that three components of $\mathbf{z}_k$ have the same noise characteristics σ_z^2; the covariance matrix of the measurement can be expressed as:

$$\mathbf{R}_k = \sigma_z^2 \cdot \mathbf{I}_3 \tag{6.39}$$

Based on the EKF equation, the innovation covariance can be expressed as:

$$\mathbf{S}_k = \mathbf{H}_k \cdot \mathbf{P}_{k|k-1} \cdot \mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k \tag{6.40}$$

The Kalman filter gain can be obtained as:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \cdot \mathbf{H}_k^{\mathrm{T}} \cdot \mathbf{S}_k^{-1} \tag{6.41}$$

Then the state can be updated by:

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_k \cdot \left[\mathbf{z}_k - \mathbf{h}\left(\mathbf{x}_k\right)\right] \tag{6.42}$$

Finally, the covariance matrix can be updated by:

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \cdot \mathbf{H}_k \cdot \mathbf{P}_{k|k-1} \tag{6.43}$$

6.2.3 Summary

Dead reckoning based on wheel speed sensors has a relatively simple structure, but it is affected by sensor noises and errors as sensor information is directly used for position estimation. Information fusion can filter the noise and estimate the errors online, but it has a more complex structure.

6.3 Inertial Sensor-Based Pedestrian Navigation

Unlike mobile robots, pedestrians do not have odometry-like sensors to measure their speed. However, the human body's movements usually show some regularity, which can be detected by inertial sensors attached to pedestrians. The human motion can be used to reduce the error drift of inertial sensor-based navigation, without requiring any additional sensors. Inertial sensor-based pedestrian navigation can be divided into two types:

1. *Zero velocity updates:* When pedestrians are stationary, for example, while waiting for an elevator or at a red light, their velocity can be

considered zero. This can be used as an observation to correct inertial navigation errors. However, these scenes usually occupy only a small portion of pedestrian movement, which has a limited effect. When pedestrians walk, their feet become stationary periodically, which can be detected by a foot-mounted IMU to perform the ZUPT. This method has a higher update frequency and can effectively suppress the rapid accumulation of errors.

2. *Pedestrian dead reckoning (PDR):* PDR calculates the position by detecting pedestrian steps based on accelerometer data, which are then transformed to the local map coordinate using the heading from gyroscopes or magnetometers. After that, the pedestrians' position can be obtained by accumulating the transformed steps. In this method, the step length is derived based on the step detection rather than the integration from accelerometer data, it avoids the rapid drift.

Next we will introduce the two methods in detail.

6.3.1 Zero Velocity Updates

Figure 6.8 shows the gait cycle phase of a pedestrian during the movement [16]. The gray point represents the stationary point. The stationary situation is considered as a zero-velocity observation to correct the inertial navigation velocity. Figure 6.9 shows a general application mode of the foot-mounted IMU for ZUPT.

{AU: Call out references 14 and 15 sequentially in the text}

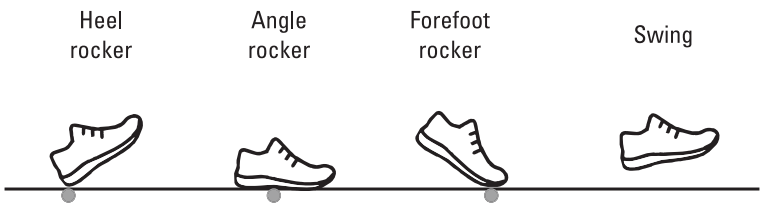

Figure 6.8 The gait cycle phase.

Figure 6.9 Foot-mounted IMU.

In ZUPT-aided inertial navigation methods, Kalman filter [17, 18] is commonly used to perform the fusion. The system state includes position, velocity, attitude, gyroscope bias, and accelerometer bias, as defined in (6.16), and its generalized framework is shown in Figure 6.10.

In stance phase detection, the system uses IMU data to recognize the stationary state by hypothesis testing [19], and the system can be considered stationary at the instant t_k based on the following equation.

$$\frac{1}{N}\cdot\sum_{l\in W_k}\left(\frac{1}{\sigma_A}\cdot\left\|\tilde{a}_l^B-g\cdot\frac{\bar{\mathbf{a}}_k^B}{\left\|\bar{\mathbf{a}}_k^B\right\|}\right\|+\frac{1}{\sigma_G}\cdot\left\|\tilde{\boldsymbol{\omega}}_l^B\right\|^2\right)<\gamma \tag{6.44}$$

where $\bar{\mathbf{a}}_k^B$ is the average of accelerometer measurements over the time window W_k of length N samples centered around instant t_k. σ_A and σ_G are the standard deviation of accelerometer and gyroscope data noise. $\tilde{\mathbf{a}}_l^B$ and $\tilde{\boldsymbol{\omega}}_l^B$ denote the accelerometer and gyroscope data at the instant t_l. γ is a zero-velocity detection threshold.

Once the system is stationary, the filter is built for state estimation. The system state and state prediction process are the same as those in IMU and wheel speed sensor integrated navigation, but the measurement model differs, which can be expressed as:

$$\begin{gathered}\mathbf{z}_k=\mathbf{h}\left(\mathbf{x}_k\right)+\mathbf{r}_k\\ \begin{bmatrix}0\\0\\0\end{bmatrix}=\begin{bmatrix}0_3 & \mathbf{I}_3 & 0_{3\times4} & 0_3 & 0_3\end{bmatrix}\cdot\mathbf{x}_k+\mathbf{r}_k\\ \mathbf{H}_k=\begin{bmatrix}0_3 & \mathbf{I}_3 & 0_{3\times4} & 0_3 & 0_3\end{bmatrix}\end{gathered} \tag{6.45}$$

Equations (6.40) to (6.43) can update the pedestrian state and its covariance matrix.

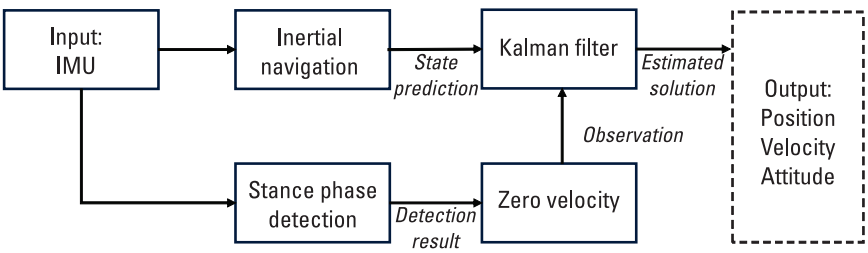

Figure 6.10 The generalized framework of ZUPT-aided inertial navigation.

6.3.2 PDR

PDR is a method that Levi and Judd first proposed in 1996 to provide autonomous navigation for foot travellers when a global positioning system (GPS) is unreliable or unavailable [20]. It resembles the dead reckoning scheme based on speed measurement, but it does not need an additional speed sensor. PDR has only one integral operation, so it can achieve higher positioning accuracy than classical inertial navigation.

PDR mainly comprises step detection, step length, and heading estimation [21], which is shown in Figure 6.11.

Step detection is the critical process of PDR, generally including the zero crossing, peak detection, autocorrelation, stance phase detection, and so on [22]. In this chapter, peak detection will be introduced to implement the detection. The total acceleration at each instance is first calculated using the following equation:

$$\tilde{a}_{k-1}^{To} = \left(\tilde{a}_{x,k-1}^{B\ 2} + \tilde{a}_{y,k-1}^{B\ 2} + \tilde{a}_{z,k-1}^{B\ 2} \right)^{\frac{1}{2}} \tag{6.46}$$

$$a_{k-1}^{To} = a_{k-1}^{To} - g \tag{6.47}$$

where $\tilde{\mathbf{a}}_{k-1}^{B} = [\tilde{a}_{x,k-1}^{B}, \tilde{a}_{y,k-1}^{B}, \tilde{a}_{z,k-1}^{B}]^{\mathrm{T}}$ is the output of accelerometers at the instance t_{k-1}. g denotes the gravity value, calculated using (6.8). The total acceleration during the walking is shown in Figure 6.12. It can be observed that the total acceleration oscillates, allowing its peaks to be identified and marked with red points.

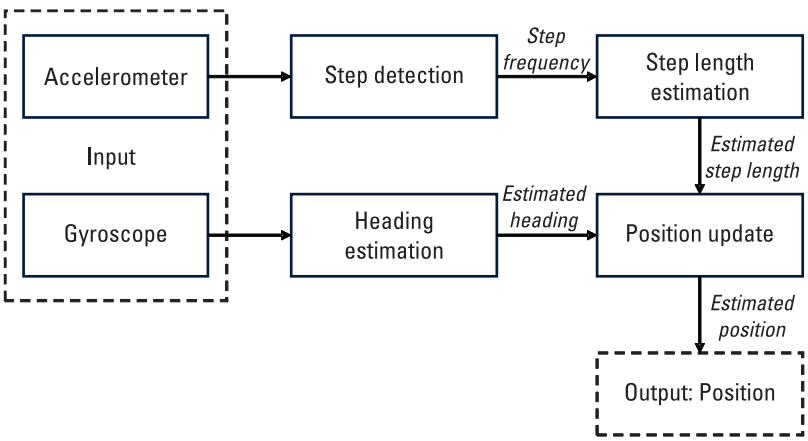

Figure 6.11 The generalized framework of pedestrian dead reckoning.

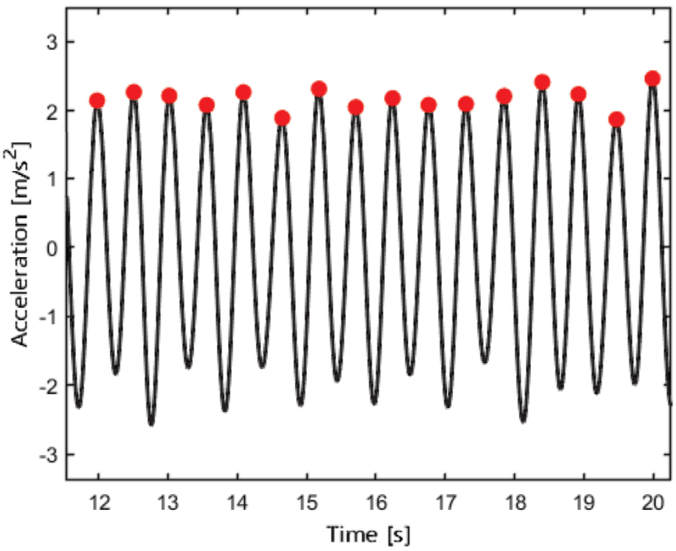

Figure 6.12 The total accelerations during the walking.

Step length estimation refers to calculating the displacement based on step detection results, mainly classified into model-based [23–25] and machine learning-based approaches [26–28]. Model-based approaches can be divided into constant, linear, and empirical models. The step length is obtained in the constant model by dividing the pedestrian walking distance by step numbers [29]. However, the estimation accuracy of this method seriously drops when pedestrian motion states greatly vary. The linear model is formed by analyzing the relationship between step length and step frequency, which will be used in this chapter to derive the step length.

The peaks are determined based on the total acceleration, and the time interval between two peaks is available. Then the step frequency can be calculated, which is shown as:

$$SF = \frac{1}{\Delta t_{pp}} \tag{6.48}$$

where SF and Δt_{pp} are step frequency and time interval between two peaks. The step length can be estimated [22], which is shown as:

$$SL = \left[0.7 + a \cdot (h - 1.75) + b \cdot \frac{(SF - 1.79) \cdot h}{1.75} \right] \cdot c \tag{6.49}$$

where SL is the step length. h is the height of the user. The coefficients a and b are two known parameters of the model, which can be set as 0.371 and 0.227. c is the personal factor that can be approximated to 1.

The step length is along the pedestrian moving direction; thus, it needs to be projected to the coordinate for navigating, which makes the heading estimation another essential issue in PDR. Integration using gyroscope data based on (6.4) is commonly utilized to obtain the heading. Then the position update model of PDR in two dimensions can be expressed as:

$$\begin{aligned} p_{x,k} &= p_{x,k-1} + SL_{k-1}\cos\psi_{k-1} \\ p_{y,k} &= p_{y,k-1} + SL_{k-1}\sin\psi_{k-1} \end{aligned} \tag{6.50}$$

where $\mathbf{P}_i = [p_{x,i}, p_{y,i}]^{\mathrm{T}}$ represents the pedestrian position at the instant t_i. SL_{k-1} and ψ_{k-1} are the step length and heading at the instance t_{k-1}.

{AU: Edit correct, here and elsewhere in the chapter?}

6.3.3 Summary

ZUPT is mainly used for foot-mounted IMU at present. The advantage of this method is that it does not need to consider human kinematics, which leads to a wider range of applicability. However, the foot-mounted IMU is only suitable for special applications, as one cannot expect pedestrians in daily life to wear an IMU on their shoes.

{AU: Edits correct?}

Although PDR is an effective way to improve the accuracy with only IMU, it has two main issues: heading inconsistency and application limitation. The former means that a heading difference generally exists between the pedestrian and the device. The device measures its heading angle, but the displacement vector is along the direction of the pedestrian's movement. In most cases, the mobile device, such as the smartphone, is not fixed on the body, as it can be put in a pocket or bag or held in any posture. The performance of the PDR will degrade if the user's heading cannot be accurately estimated. The second one means that PDR can only be used in pedestrians or legged robots [30], such as robot dogs and biped robots [31], as it relies on step detection. Moreover, PDR is a person-sensitive positioning in which many factors, including heights, weights, ages, genders, leg lengths, dressings, and personal habits, affect step detection and length estimation [32].

Recently, data-driven artificial intelligence (AI) methods, such as deep learning and machine learning algorithms, have emerged as promising solutions to these challenges [33, 34]. AI-driven approaches leverage large-scale, diverse datasets to learn intricate patterns from sensor data, effectively mitigating the problems caused by heading inconsistencies and individual variations.

Furthermore, advances in sensor fusion, adaptive filtering, and personalized models powered by AI techniques offer significant potential to enhance PDR accuracy, reliability, and robustness across various usage scenarios. Looking forward, integrating AI methods with traditional PDR techniques is expected to become increasingly prevalent, paving the way for more accurate, universally adaptable, and user-friendly indoor positioning solutions.

6.4 Environmental Sensors for Enhanced Dead Reckoning

Even small errors in dead reckoning can cause significant drift, as these inaccuracies compound over time during position estimation. To mitigate this drift, dead reckoning systems incorporate environmental sensors, such as magnetometers and barometric altimeters [11]. Magnetometers measure Earth's magnetic field and serve as electronic compasses for heading (yaw) estimation, while barometric pressure sensors (barometers) measure air pressure to infer altitude. These sensors provide absolute orientation and elevation references that complement relative measurements from accelerometers and gyroscopes, thereby improving long-term accuracy.

Magnetometers and barometers are especially important in environments where GPS or external references are unavailable. Table 6.2 summarizes the roles of these environmental sensors in dead reckoning. Both sensors face unique challenges: magnetometers are prone to interference and require careful calibration, and barometers are affected by weather-induced pressure changes. In this section, we focus on the mathematical models and theoretical aspects of using magnetometers for heading estimation and barometers for altitude sensing in dead reckoning.

6.4.1 Magnetometers for Heading Estimation

The Earth's magnetic field provides a global reference for orientation. At any given location, the Earth's field can be modeled as a vector with known

Table 6.2
Key Environmental Sensors in Dead Reckoning and Their Roles

Sensor	Measured Quantity	Role in Dead Reckoning
Magnetometer	Earth's magnetic field (3-axis)	Heading (yaw) estimation
Barometer	Atmospheric pressure	Altitude estimation

inclination and declination angles. In a local North-East-Down (NED) reference frame, the magnetic field vector can be represented as $\mathbf{B}_M = [B_N, B_E, B_D]^{\mathrm{T}}$, where B_N and B_E are the horizontal components (pointing toward magnetic North and East, respectively) and B_D is the vertical (downward) component. A three-axis magnetometer measures the Earth's field in the sensor's body frame. The ideal magnetometer measurement model can be written as:

$$\tilde{\mathbf{m}}^B = \mathbf{C}_M^B \mathbf{B}_M + \mathbf{b}_{\text{bias}} + \boldsymbol{\epsilon} \tag{6.51}$$

where $\tilde{\mathbf{m}}^B$ is the magnetometer measurement vector in the body frame, $\mathbf{C}_M^B$ is the rotation matrix from the navigation frame to the body frame (dependent on the sensor's roll, pitch, yaw), $\mathbf{b}_{\text{bias}}$ represents biases (hard-iron offsets), and $\boldsymbol{\epsilon}$ represents measurement noise (and small distortions). Equation (6.51) indicates that, ignoring noise and bias, the measured field is just the Earth's field rotated into the body frame.

To estimate heading from $\tilde{\mathbf{m}}^B$, the influence of roll and pitch must be removed so that only the horizontal component of $\mathbf{B}_M$ is considered. If the roll ϕ and pitch θ of the sensor (with respect to the horizontal plane) are known (e.g., from an accelerometer), the magnetometer reading can be tilt-compensated. Let $[\tilde{m}_x^B, \tilde{m}_y^B, \tilde{m}_z^B]^{\mathrm{T}}$ denote the magnetometer readings in the body frame axes. The horizontal components of the Earth's field in the body frame, (M_x, M_y), can be obtained by rotating the magnetometer measurement back to a level orientation:

$$\begin{aligned} M_x &= \tilde{m}_x^B \cos(\theta) + \tilde{m}_z^B \sin(\theta) \\ M_y &= \tilde{m}_y^B \cos(\phi) + \tilde{m}_x^B \sin(\theta)\sin(\phi) - \tilde{m}_z^B \cos(\theta)\sin(\phi) \end{aligned} \tag{6.52}$$

The heading relative to magnetic North is then given by:

$$\psi = \operatorname{atan2}\left(M_y, M_x\right) \tag{6.53}$$

which yields the magnetometer-based heading. In practice, a known local magnetic declination can be added to ψ to convert a magnetic heading to a true heading (relative to geographic North).

Heading estimation via magnetometers faces several challenges. First, magnetometer readings require calibration to correct for sensor biases and scale factors. Hard-iron effects (constant additive biases caused by permanent magnets or direct current (DC) in the vicinity) and soft-iron effects (scale distortions caused by nearby ferrous materials) can skew $\tilde{\mathbf{m}}^B$. Calibration is typically performed by rotating the sensor in a clean magnetic environment

{AU: Edits correct?}

and fitting the measurements to a sphere or ellipsoid to solve for biases and misalignments [35]. Second, external magnetic disturbances can cause temporary heading errors. Nearby ferromagnetic objects or electromagnetic fields can distort the local field, leading to heading errors if interpreted as changes in $\mathbf{B}_M$. Theoretically, one can improve accuracy by modeling and filtering out such disturbances. For example, algorithms can estimate time-varying bias terms $\mathbf{b}_{\text{bias}}$ as part of the state in a filter or reject outlier measurements when sudden magnetic distortions are detected. Another approach is to use multiple magnetometers or reference sensors to detect and cancel interference, although this adds complexity.

Various theoretical methods have been proposed to enhance magnetometer heading accuracy. One approach is to integrate magnetometer data with gyroscope data in a Kalman filter or complementary filter so that gyro-derived heading (which is smooth but drifts) is continuously corrected by magnetometer measurements (which are absolute but noisy) [36, 37]. Another approach is to apply optimal estimation or smoothing techniques that account for sensor noise characteristics. In summary, magnetometers provide a crucial absolute heading reference for dead reckoning, but careful calibration and filtering are required to mitigate errors from sensor imperfections and environmental disturbances.

6.4.2 Barometer for Altitude Sensing and Integration

Barometric pressure sensors measure ambient air pressure, which decreases with increasing altitude. This physical principle allows barometers to function as altimeters. The relationship between pressure and altitude can be derived from the barometric formula under standard atmospheric conditions. Assuming a standard tropospheric lapse rate (temperature decreases linearly with altitude), the pressure altitude h as a function of the current atmospheric pressure P is given by:

$$h = \frac{T_0}{L} \times \left(1 - \left(\frac{P}{P_0} \right)^{\frac{R \times L}{g \times M}} \right) \tag{6.54}$$

where T_0 is the temperature at sea level (standard atmosphere: 288.15K), and L is the temperature lapse rate (standard: 0.0065 K/m). P_0 represents the pressure at standard ground level (101,325 Pa).

Depending on the desired altitude reference, the atmospheric pressure P used in the above equation can be defined in two ways:

- Local pressure is used if the height above the ground level is desired.
- Mean sea level (MSL) pressure is used if geodetic height above sea level is desired.

R is the gas constant (8.314 J/(mol·K)), g is the gravitational acceleration, and M is the molar mass of air (0.029 kg/mol). Figure 6.13 illustrates the barometer solution, which generally applies when using MSL pressure with QNH correction (QNH refers to adjusting the local pressure to the MSL pressure and using it to set the altimeter so that it displays the altitude relative to sea level (i.e., geodetic height)).

Barometric altimeters are often integrated into dead reckoning systems to provide the vertical position component. A barometer on its own can measure changes in altitude with good relative accuracy (often resolving altitude differences on the order of meters). However, for absolute altitude, the sensor must be calibrated to a reference pressure. In a dead reckoning context, barometric altitude can be combined with other sensor data (such as vertical acceleration from an IMU) to improve overall altitude estimation [38]. For example, one can use the barometer to correct low-frequency drift in altitude that comes from double-integrating accelerometer data, while relying on accelerometers to capture rapid vertical motions.

Several challenges arise in using barometers for altitude sensing. One major issue is that atmospheric pressure is influenced not only by altitude but also by weather systems (e.g., high-pressure and low-pressure regions). This means that, over time, P_0 can effectively change, causing the inferred altitude to drift even if the actual altitude is constant. Another challenge is sensor noise

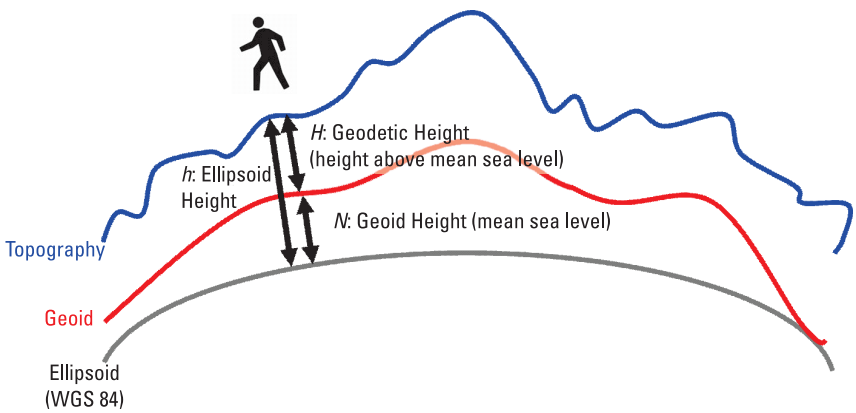

Figure 6.13 Barometer solution (in general if MSL pressure is used (QNH correction)). Google Earth used this one.

{AU: Please specify "one"; solution?}

and resolution; barometric readings can fluctuate due to turbulence or sensor quantization, introducing noise in altitude estimates. Additionally, in indoor environments or sealed areas, pressure readings might be affected by heating, ventilation, and air conditioning (HVAC) systems or other factors unrelated to true altitude changes.

Theoretical solutions have been developed to address these challenges. To handle slow pressure drift due to weather, one method is to periodically recalibrate P_0 using known altitude references when available. In a filter framework, one can also treat the reference pressure or bias as part of the state to be estimated, thus correcting for it in real time. To reduce noise, optimal filtering (e.g., a Kalman filter) can be applied to smooth the altitude estimates, using a process model for altitude. For instance, a simple process model might assume that altitude changes only when there is a corresponding acceleration, so that sudden pressure changes without corroborating acceleration can be discounted as noise. Another theoretical approach is to fuse the barometric data with other environmental data such as temperature and humidity to adjust the model in (6.54) (since T_0 and L can vary with weather), thereby improving the accuracy of the altitude calculation. In summary, barometers are effective for sensing altitude in dead reckoning, but they must be carefully integrated and calibrated to mitigate errors from atmospheric variations and sensor noise.

6.4.3 Sensor Fusion Strategies for Environmental Sensors

A key theoretical aspect of using environmental sensors in dead reckoning is how to fuse them with other sensors (such as IMU sensors) to obtain a more accurate and robust estimate of the navigation state. Sensor fusion algorithms combine multiple sensor inputs by weighting them according to their uncertainty and dynamics. Common frameworks include the Kalman filter (and its variants such as the EKF) and particle filters. These frameworks differ in complexity and optimality, as summarized in Table 6.3.

Table 6.3
Comparison of Sensor Fusion Strategies

{AU: Edits correct?}

Method	Advantages	Limitations
EKF	Optimal (minimum variance) estimate if models are accurate	Requires linearization and precise noise models
Particle filter	Handles nonlinear/non-Gaussian cases	High computational cost, complexity in implementation

In a Kalman filter framework, one defines a state vector (e.g., including position, velocity, orientation, sensor biases) and uses a dynamic model to predict state evolution and measurement models for sensor readings. For example, a simplified state might include heading ψ and altitude h as components. The process (state transition) and measurement equations can be expressed as:

$$\begin{aligned} \mathbf{x}_{k+1} &= f\left(\mathbf{x}_k, \mathbf{u}_k\right) + \mathbf{w}_k \\ \mathbf{z}_k &= h\left(\mathbf{x}_k\right) + \mathbf{v}_k \end{aligned} \tag{6.55}$$

where $\mathbf{x}_k$ is the state at time t_k, $\mathbf{u}_k$ represents control inputs or inertial measurements (e.g., gyro turn rate or acceleration), $\mathbf{z}_k$ is the measurement vector (e.g., magnetometer heading and barometric altitude readings), and $\mathbf{w}_k$, $\mathbf{v}_k$ are process and measurement noise, respectively. The functions $f(\ )$ and $h(\ )$ represent the system's dynamic model and measurement model. For instance, $f(\ )$ might propagate the heading by adding the integrated gyro turn rate to ψ, and $h(\ )$ might extract the expected magnetometer reading. The Kalman filter (or EKF for nonlinear $f(\ )$, $h(\ )$) uses these equations to predict the state and then update it when new measurements arrive, weighting the uncertainty of the prediction versus the observation.

The theoretical advantage of combining magnetometer and barometer data with inertial sensors is error mitigation. Gyroscope integration alone would let heading drift unbounded, but a magnetometer measurement provides an absolute reference to clamp that drift. Likewise, integrating accelerometer data for vertical position is prone to drift due to biases, but barometric altitude can serve as a reference to correct that drift. By weighting these appropriately (for instance, a Kalman filter automatically assigns higher weight to the barometer reading when accelerometer-derived altitude is uncertain, and vice versa), the fused estimate can be much more accurate than what each sensor would achieve on its own. Sensor fusion also enables estimation of sensor biases (e.g., magnetometer bias or accelerometer bias) as additional state variables, thus improving long-term accuracy by correcting those biases online.

6.5 Error Modeling and Calibrations

As described previously, each sensor type exhibits unique error characteristics that must be modeled and calibrated to achieve accurate navigation solutions. In this section, we discuss the error sources and calibration techniques for four key sensors: the IMU, wheel odometry sensor, magnetometer, and barometric altimeter.

6.5.1 Error Sources

In this section, we review the error sources associated with various sensors. The main error sources are summarized in Table 6.4.

6.5.1.1 IMUs

IMUs (accelerometers and gyroscopes) have errors such as bias, noise, scale factor, and misalignment. A constant gyroscope bias causes orientation drift over time, leading to position errors that grow cubically if uncorrected. Similarly, accelerometer biases result in errors in velocity and position estimates. Additionally, inertial sensor outputs contain noise, causing drift when raw angular velocity and acceleration data are integrated to obtain angles and velocities. For gyroscopes and accelerometers, this drift is characterized as angle random walk (ARW) and velocity random walk (VRW), respectively. Stochastic noise in IMU readings is commonly analyzed using the Allan variance method [39], which helps to identify specific noise types, including random walk and bias instability. Here we provide an example demonstrating how Allan variance can be applied to estimate these parameters. We utilize gyroscope data and compute the Allan deviation, as shown in Figures 6.14 and 6.15. The vertical

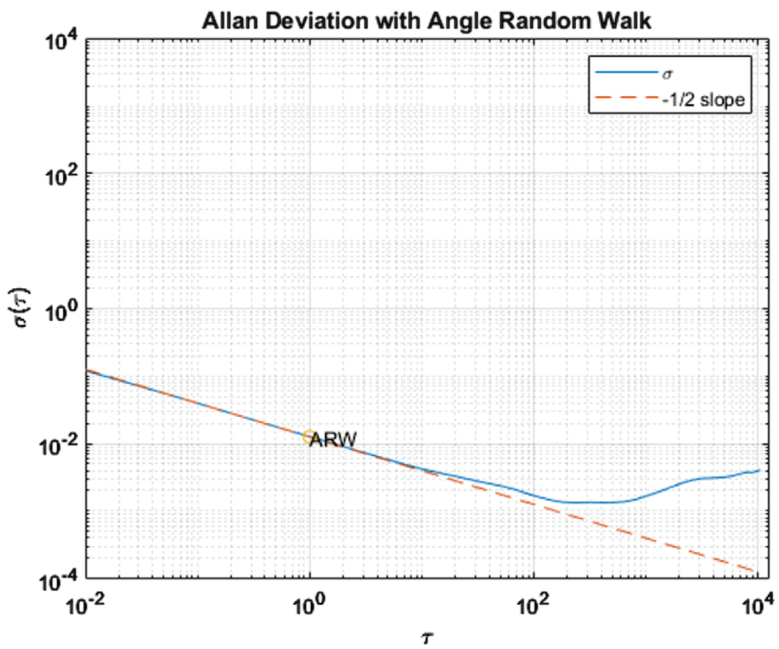

Figure 6.14 Allan deviation analysis with ARW for gyroscope data.

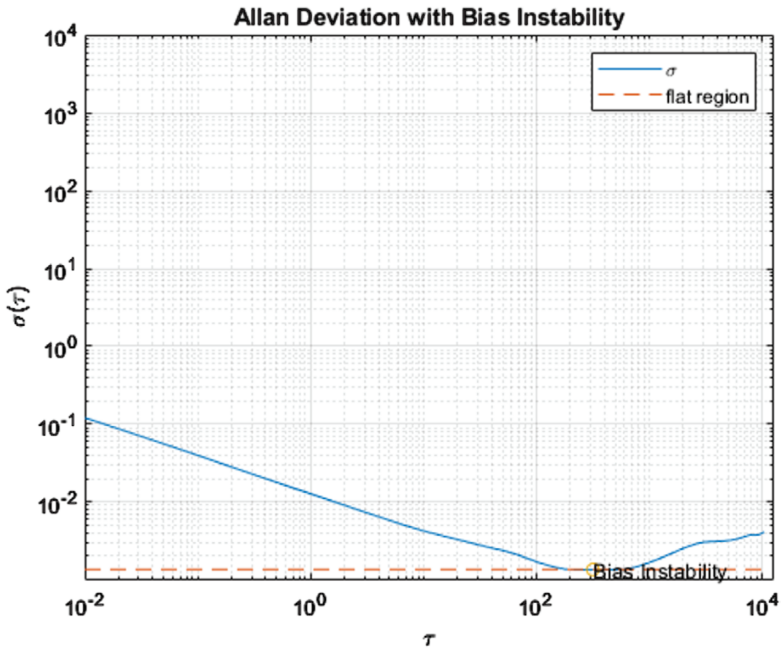

Figure 6.15 Allan deviation analysis with bias instability for gyroscope data.

axis represents the Allan deviation, $\sigma(\tau)$, while the horizontal axis denotes the averaging time. The ARW can be expressed as follows:

$$\sigma(\tau) = \frac{\text{ARW}}{\sqrt{\tau}} \tag{6.56}$$

Taking the logarithm of the equation, we obtain:

$$\log\left(\sigma(\tau)\right) = \log(\text{ARW}) - \frac{1}{2}\log(\tau) \tag{6.57}$$

This means that, by performing a log-linear fit on the Allan deviation, where the slope is approximately −1/2, the exponential of the fitted intercept gives the ARW value. Once the ARW is determined, the standard deviation of the white noise can be calculated as follows:

$$\sigma_\omega = \text{ARW} \times \sqrt{\Delta t} \tag{6.58}$$

where Δt represents the sampling interval of the gyroscope data.

For bias instability, locate the lowest point (flat region) of the Allan deviation curve and use it as the estimated value of bias instability. The code for computing the Allan variance is available in the Sensor Models section of the MATLAB Navigation Toolbox.

In addition, scale factor errors (gain errors) and axis misalignment lead to each axis measuring a slightly incorrect or coupled value of the true motion. Thereby, IMU error models often take the form:

$$\begin{aligned}\tilde{\mathbf{a}} &= \left(\mathbf{I} + \mathbf{S}_a + \Delta_a\right)\mathbf{a}_{\text{true}} + \mathbf{b}_a + \mathbf{n}_a \\ \tilde{\boldsymbol{\omega}} &= \left(\mathbf{I} + \mathbf{S}_\omega + \Delta_\omega\right)\boldsymbol{\omega}_{\text{true}} + \mathbf{b}_\omega + \mathbf{n}_\omega\end{aligned} \tag{6.59}$$

For an accelerometer measuring specific force $\mathbf{a}$ and a gyroscope measuring rotation rate $\boldsymbol{\omega}$, errors include scale factor errors $\mathbf{S}$, misalignments Δ, biases $\mathbf{b}$, and noises $\mathbf{n}$. Environmental factors, such as temperature changes, can cause these biases to drift over time.

6.5.1.2 Wheel Odometry

Wheel odometry errors are typically classified as either systematic errors (such as scale factors or misalignments) or nonsystematic errors (random or terrain-induced). A common systematic error arises from the misestimation of wheel circumference or wheel base, resulting in consistent scale-factor errors in distance and turning calculations, which lead to cumulative position drift. Nonsystematic errors include wheel slip (e.g., wheels spinning without corresponding forward motion on low-friction surfaces) and uneven contact with the ground, both of which cause unpredictable jumps in odometry. Additionally, wheel encoder measurements introduce quantization noise due to finite encoder resolution, although this is typically minor compared to other error sources. In a simplified odometry error model, deterministic errors can be effectively captured using parameters representing left/right wheel scale factors and a fixed angular misalignment, whereas random slip events are modeled as process noise within a navigation filter.

6.5.1.3 Magnetometer

Magnetometers have bias, scale factor, misalignment and environment-induced errors. Noise in magnetometers is typically low (sensor resolution on the order of microteslas), but environmental interference inside buildings can be severe. Nearby ferrous metals or electrical currents produce local magnetic fields effectively introducing a time-varying bias. These environment-induced errors can lead to large heading inaccuracies if not accounted for.

6.5.1.4 Barometer

Barometric altimeters measure air pressure to infer altitude. A pressure sensor's primary error is a bias in the pressure reading, which translates to an altitude offset. Noise in pressure readings causes short-term altitude jitter (usually a few centimeters after filtering). A scale (sensitivity) error is usually small due to factory calibration. However, barometer readings are strongly affected by environment changes: weather-induced pressure variation or indoor HVAC systems can cause the measured pressure to drift independent of actual altitude changes (e.g., a mere 0.1% change in pressure can induce several meters of apparent altitude change). We can model the measured pressure as $\tilde{h} = h_{\text{true}} + b_h + n_h$, where b_h may slowly vary with the environment. Without compensation, such drift b_h would appear as a false change in altitude.

6.5.2 Error Calibrations

To mitigate these errors, calibrations are performed both in controlled settings and during operation. Broadly, we have: (1) offline calibration in laboratories or at initialization to identify biases, scale factors, and misalignments; (2) online mitigation using algorithms (filtering or real-time estimation) to compensate for errors during use; and (3) hardware approaches that reduce errors via physical design or redundancy. Table 6.5 summarizes examples of each.

6.5.2.1 IMU Calibration

For IMUs, offline calibration involves measuring sensor output under known conditions, for example, static orientations (to solve for accelerometer biases) and constant rotation on a precision turntable (to determine gyro scale factor error) [47]. Temperature chamber (Figure 6.16) tests can map bias variation versus temperature, and Allan variance analysis of static data yields noise parameters. The identified calibration parameters are then applied to correct raw measurements. Online methods refine this calibration: for instance, foot-mounted IMUs use zero-velocity updates (when a foot is on the ground, true velocity is zero) to reset accumulated error and estimate bias in real time [19]. Additionally, an EKF in an INS can include bias states that are continuously estimated by integrating other sensors or known position updates. As a hardware approach, using multiple IMUs and averaging their outputs can reduce random noise (since uncorrelated noise tends to cancel out) [48]. Higher-grade sensors or mechanically isolating the IMU from vibrations are other hardware strategies to improve performance.

Table 6.4
Error Sources in Different Sensors

Error Source	IMU	Wheel Odometry	Magnetometer	Barometer
Bias (offset) [2, 40, 41]	Yes: Small constant biases in accelerometers and gyroscopes (e.g., nonzero outputs at rest) cause errors that accumulate over time. In IPIN, bias drift significantly impacts accuracy, requiring regular calibration or zero-velocity updates to reduce drift.	No (negligible): An encoder measures incremental rotation; it generally produces no output when the wheel is still, so there is not an additive bias term.	Yes: It typically exhibit constant bias due to hard-iron effects (permanent magnetization of the device or nearby ferrous materials). This appears as an additive offset in the magnetic field reading that persists until calibrated out. Indoor devices often require magnetometer calibration to remove hard-iron bias. If uncorrected, the bias skews heading estimates.	Yes: It often exhibits baseline offsets varying between devices and over time, causing altitude biases of several meters. Barometers are usually calibrated to known reference pressures or altitudes. Periodic recalibration is essential for accurate elevation estimates indoors.
Noise (random error) [2, 40, 42, 43]	Yes: Their readings contain significant random noise (thermal and electrical). In low-cost MEMS ones, this noise causes orientation and velocity errors to grow stochastically (often modeled as angular random walk and velocity random walk).	Yes: It has random errors due to encoder resolution limits and wheel slip variations. Encoder quantization (discrete wheel tick counts) introduces small rounding errors in distance, and uneven floor surfaces or slight slippage cause nonrepeatable deviations.	Yes: The measurements include sensor noise and short-term field fluctuations. The MEMS ones exhibit inherent noise in the measured field (e.g., on the order of tens of microteslas resolution), and indoor magnetic readings can fluctuate due to transient electromagnetic	Yes: Their readings are prone to small random fluctuations. Sensor thermal noise and air turbulence cause the pressure to vary by a few tenths of a hectopascal even in a steady environment, translating to noise of decimeters in altitude.

Table 6.4
(Cont.)

Error Source	IMU	Wheel Odometry	Magnetometer	Barometer
	In IPIN, this high-frequency sensor noise is a key contributor to inertial drift. Advanced filtering (e.g., Kalman filters) is used to partially mitigate IMU noise.	These noise-like errors are zero-mean over time but cause the robot's computed position to wander as they accumulate. Probabilistic models often treat such odometry uncertainty as random noise in localization algorithms.	disturbances (appliances, electrical currents). This noise is usually modeled as zero-mean Gaussian noise in the magnetometer error model. Averaging or lowpass filtering is often applied to reduce random magnetic noise.	IPIN systems often apply averaging or Kalman filtering to smooth out barometric noise while tracking floor levels.
Scale factor error [40, 44, 45]	Yes: They have scale factor errors, meaning the sensor's output gain deviates from the ideal. Without proper factory calibration, each axis can have a sensitivity error (ppm or % of full-scale). In indoor use, these errors lead to distortion in the computed trajectory unless the IMU is calibrated. Modern IMUs often undergo multi-axis calibration to correct scale factors before use.	Yes: A wheel sensor's distance per tick is based on an assumed wheel circumference and wheel base. Any miscalibration here yields a scale factor error. For example, if the actual wheel diameter is larger than assumed, every rotation covers more distance than the encoder calculation, introducing a proportional error. Such scale errors are systematic and common if the odometry is not calibrated for the specific robot and floor conditions.	Yes: Their scale factor errors occur when the sensor axes have unequal gains or nonlinearity, often due to manufacturing tolerances (soft-iron effects can also appear as scaling on axes). This causes the true geomagnetic field, which should plot as a sphere when the device is rotated, to appear as an ellipsoid. Indoor magnetometers frequently exhibit this error, requiring calibration by fitting an ellipsoid to measured data and scaling axes accordingly.	No (minimal): The sensors are factory-calibrated and generally have negligible scale factor errors. Height-pressure relationships follow the barometric formula, with deviations mainly due to environmental assumptions (e.g., temperature, humidity), not sensor gain. As long as outputs are in proper units, errors are dominated by bias and corrected through offset calibration, not scaling.

Table 6.4
(Cont.)

Error Source	IMU	Wheel Odometry	Magnetometer	Barometer
Misalignment (nonorthogonality) [40, 45, 46]	Yes: Axis misalignment is common in MEMS IMUs, where sensor axes may not be perfectly orthogonal or aligned with the device frame. This causes cross-axis coupling and inaccurate readings. Misalignment must be corrected using a calibration matrix. High-end IMUs are pre-calibrated, but low-cost units typically require alignment during initialization.	Yes: In differential drive odometry, misalignment refers to physical alignment errors like wheels not perfectly parallel or an unknown effective wheel base length. A slight skew in wheel mounting can cause one wheel to travel a different path, leading to a gradual heading bias (systematic turning error). Even a tiny angular misalignment between the wheel encoder axis and the robot frame can accumulate into significant position error over distance. This is one of the two dominant systematic odometry errors that are routinely calibrated out.	Yes: The axis misalignment arises from assembly tolerances and misalignment with the device's heading frame, leading to heading errors. Calibration typically estimates a misalignment matrix along with scale factors to align the sensor to the navigation frame. Indoors, this is handled during commissioning or in real-time via sensor fusion.	No: It measures ambient air pressure, a scalar quantity, so orientation or alignment has no effect on its reading. Unlike directional sensors, there are no axes to misalign in a pressure sensor; it outputs the same pressure regardless of how the device is rotated. Indoor barometric readings depend on location (and airflow conditions) but not on the sensor orientation, so misalignment is not applicable for barometers.

Table 6.4
(Cont.)

Error Source	IMU	Wheel Odometry	Magnetometer	Barometer
Specific environment-induced error [2, 40–42]	Yes (temperature and vibration): Indoor environments indirectly affect IMU accuracy. MEMS IMU biases are temperature-sensitive. Vibrations from machinery or walking impacts can also introduce errors. While IMUs are not affected by electromagnetic or pressure factors, the lack of external references indoors allows bias and noise to accumulate.	Yes: Indoor surfaces impact wheel sensor accuracy. Slippery floors cause wheel slip—rotation without actual movement—leading to odometry errors. Soft carpets can alter effective wheel radius, and floor bumps or cables disrupt normal motion. These nonsystematic errors accumulate unpredictably. To mitigate them, robots often use slip detection, recalibration, or fuse data from IMUs or vision.	Yes: Indoor environments often distort magnetometer readings due to steel structures, wiring, and appliances. Elevators, power lines, or reinforced walls can shift the magnetic heading by tens of degrees. Magnetic anomalies vary across rooms, making headings unreliable without frequent calibration. Many systems either avoid magnetometers indoors or fuse them with gyros to filter out bad data.	Yes: Indoor barometer readings can be affected by HVAC systems, ventilation, and door or elevator movements, causing pressure shifts unrelated to altitude. Weather changes also cause slow drift. As a result, barometric height estimates indoors can be unreliable without compensation. Systems often rely on relative changes, floor-based references, or filtering to separate true elevation from environmental effects.

Table 6.5
Examples of Calibration and Compensation Methods for Each Sensor

Sensor	Offline Calibration	Online Calibration	Hardware Improvements
IMU	Multiposition static tests; turntable rotation and Allan variance	Filter bias estimation; ZUPT	Redundant IMUs (IMU array); thermal isolation
Wheel odometry	Drive known patterns (e.g., squares)	Adaptive slip correction	High-resolution encoders; extra wheel sensor
Magnetometer	Rotate in all axes (fit ellipsoid)	Auto bias update; outlier rejection	Isolate from motors; dual units
Barometer	Set reference offset	Periodic reference update; filter	Multiple sensors (averaging)

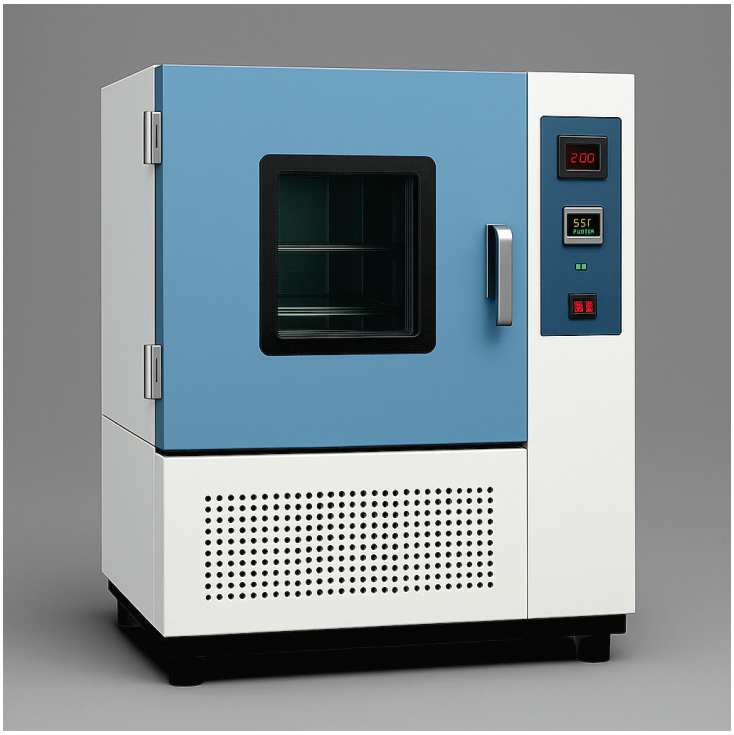

Figure 6.16 The temperature chamber.

6.5.2.2 Wheel Odometry Calibration

Wheel odometry offline calibration is often done by driving the robot in known patterns and adjusting parameters. A classic example is the UMBmark test (driving in a square and returning to the start) to quantify and correct systematic odometry errors such as unequal wheel diameters [49]. By analyzing the final pose error, one can estimate wheel radius and wheel base corrections. Online odometry calibration can be achieved by sensor fusion: for instance, if the robot has an IMU or vision system, it can detect when odometry diverges from other sensors and adjust wheel slip or scale factors on-the-fly [50]. Some robots use adaptive algorithms to reduce reliance on wheel data when slip is detected (e.g., if wheel encoders report motion but an IMU indicates the robot has stopped, the system recognizes slippage). Hardware improvements include using high-resolution encoders to reduce the quantization error and add auxiliary sensing. For example, an extra free-rolling wheel with its own encoder can provide ground-truth displacement to detect slipping of the drive wheels. Ensuring good wheel traction (tire material, weight distribution) is also important to minimize slip-induced errors.

6.5.2.3 Magnetometer Calibration

Magnetometer offline calibration is performed by collecting data in many orientations and fitting an ellipsoid (in high-end setups, a three-axis Helmholtz coil can apply known fields for calibration) to find the hard-iron bias and soft-iron scaling. This yields a bias vector and calibration matrix so that the output can be corrected to true field values. Many devices ask users to perform a "figure-8" motion that effectively gathers such calibration data. Online, systems can monitor the magnetic field magnitude and, if a deviation from the expected Earth field is detected over time, slowly adjust the bias estimate. Also, if a sudden magnetic disturbance is sensed (e.g., the magnetometer reading changes rapidly while the IMU indicates no rotation), the system can temporarily ignore or down-weight magnetometer data. In terms of hardware, the magnetometer should be placed away from current-carrying wires or ferrous components in the device. Some designs use dual magnetometers at different locations; if one is disturbed, the other might provide a cleaner signal. Using magnetically "quiet" materials and adding shielding can further reduce biases.

6.5.2.4 Barometer Calibration

Barometer calibration is straightforward. Offline, one can set an initial offset so that the barometer reads the correct altitude at a known reference (for example, zero at a known ground level or a specific floor altitude). High-end

systems might calibrate the pressure sensor at multiple known pressures and temperatures to refine its accuracy curve. Online, since pressure can drift with weather and indoor conditions, the system may regularly recalibrate using known altitude references. For instance, if an indoor map provides floor heights, the barometric altitude reading can be reset when the user is detected to reach a new floor (e.g., upon taking an elevator). Filters such as Kalman filters are often used to blend barometric altitude with other vertical measurements (accelerometer integration or other altimetric sensors) to correct drift. As a hardware solution, some devices incorporate multiple barometric sensors and average their readings to reduce noise and detect anomalies (if one sensor diverges, it can be flagged). Ensuring the sensor is properly vented to ambient air is also important for accuracy. Overall, thorough calibration of these sensors, combined with robust error modeling, significantly improves the accuracy and robustness of indoor navigation systems. In practice, these methods are often used in combination (e.g., applying a factory calibration and then running an online bias estimator) to ensure long-term stability.

6.6 Drift Mitigation and Corrections

This section introduces theoretical approaches to mitigate drift, built on three pillars: (1) robust initialization, including initial alignment and proper covariance setup for the Kalman filter; (2) exploitation of motion constraints, such as zero-velocity updates (ZUPT), zero-angular-rate updates (ZARU), and nonholonomic constraints (NHC); and (3) external aiding via sensor fusion with additional sensors. We provide a detailed overview of the algorithms and the underlying theory for each method, focusing on how they constrain error growth and enhance indoor positioning performance.

6.6.1 Initialization

Proper initialization of an INS is critical to prevent large initial errors from seeding future drift. Initialization involves two main components: (1) initial alignment, which determines the orientation of the INS with respect to a reference frame, and (2) initialization of the state error covariance matrix $\mathbf{P}$ (e.g., for an EKF), which reflects confidence in the initial state estimates.

6.6.1.1 Initial Alignment

Initial alignment is the process of finding the orientation (attitude) of the sensor platform relative to the navigation frame before navigation begins. For

land-based or indoor INS, this often means determining heading (yaw) with respect to north and leveling the pitch and roll with respect to gravity [11]. A common method for static initial alignment is to use accelerometer readings to define the gravity vector (thus setting the roll and pitch angles by assuming the device is stationary and gravity points downwards), and to use either a magnetometer or known reference direction to establish yaw. In the absence of a magnetometer, if the device remains static for a period, one can also utilize the gyroscope to detect Earth's rotation (for example, in high-precision systems, the gyroscope can sense the Earth's rotation rate projected in the sensor axes to aid in finding North). Once coarse alignment is done (perhaps yielding attitude within a few degrees), a fine alignment can be performed by an attitude Kalman filter that further refines the orientation using the INS equations themselves, treating the initial position as known and velocity as zero. The fine alignment stage often runs the INS in a feedback loop for a short duration while stationary, estimating residual orientation errors and gyroscope biases.

Mathematically, if we denote the true orientation (as a rotation matrix or quaternion) as $\mathbf{C}_B^M$ (body to nav frame rotation) and our initial guess as $\mathbf{C}_{B,0}^M$, the alignment error can be represented by a small rotation $\tilde{\boldsymbol{\theta}}$ such that $\mathbf{C}_B^M \approx \exp([\tilde{\boldsymbol{\theta}}]_\times)\mathbf{C}_{B,0}^M$, where $[\ ]_\times$ is the skew-symmetric matrix of a vector. Fine alignment algorithms attempt to drive $\tilde{\boldsymbol{\theta}}$ to zero by observing inertial sensor outputs under the assumption of known true motion (typically that the system is at rest, so true specific force is just $-\mathbf{g}$ in the vertical and true angular rate is Earth's rotation in nav-frame coordinates). Any discrepancy between measured acceleration/rotation and the expected values given the current state estimate is attributed to misalignment or sensor bias, and a Kalman filter can estimate those errors.

6.6.1.2 Initial Covariance Setting

In an EKF-based INS, we maintain a state error covariance matrix $\mathbf{P}$ that represents the uncertainty in the navigation states (position, velocity, attitude) and any included sensor error states (e.g., gyroscope bias, accelerometer bias). The initialization of $\mathbf{P}$ is crucial for filter performance; if the values are too small (overconfident), the filter will underestimate initial uncertainty and might not properly correct initial errors, while if too large, the filter may be overly sensitive and respond erratically to measurements.

Typically, one sets the initial variances for position and velocity states based on knowledge of the starting condition. For example, if the starting position is known precisely (e.g., at the origin of a local coordinate system), position variance can be set very low (almost zero). Velocity can be assumed

zero initially for a stationary start, but one might still assign a small variance to account for any slight movement or sensor noise. Attitude uncertainty depends on the alignment process; after coarse leveling and heading initialization, one might assume a few degrees of uncertainty in yaw and smaller in pitch/roll. Gyroscope and accelerometer biases often have very large initial uncertainty if uncalibrated (since they could be anywhere within the sensor's specified bias error bounds).

During the initial alignment procedure, the filter may also be employed to reduce these uncertainties. For example, as the fine alignment runs while the device is static, the filter will update and typically drive down the attitude and bias uncertainty rapidly as it converges. After successful initialization, the INS is ready to navigate with a consistent reference frame and a well-characterized error covariance, which is the foundation for applying further drift mitigation techniques.

6.6.2 Motion Constraints for Drift Mitigation

Table 6.6 summarizes these motion constraints. Each provides a form of "ground truth" for certain state components (zero velocity or zero turn-rate) under known conditions, thereby stopping the unchecked growth of those errors. By integrating such updates into the navigation filter, the dead reckoning can operate for longer periods without external position fixes while keeping drift to manageable levels. Next we will briefly introduce the motion constraints.

6.6.2.1 Zero Velocity Update (ZUPT)

As introduced in the previous section, ZUPT [18] is a technique primarily used in pedestrian (foot-mounted) INS or other applications where the sensor

Table 6.6
Summary of Motion Constraints for Drift Mitigation

Constraint	Quantity Forced to Zero	Applicable Condition	Primary Effect
ZUPT	Linear velocity	Come to a complete stop or let your foot touch the ground	Corrects velocity and position drift
ZARU	Angular turn-rate	No rotation (often at stop)	Calibrates gyro bias, corrects orientation
NHC	Lateral and vertical velocity	Normal wheeled motion	Prevents sideways/ vertical drift, improves heading

periodically comes to rest (i.e., has zero velocity) for short durations. The basic idea is to detect moments when the IMU is stationary relative to the navigation frame and, at those moments, inject a pseudo-measurement indicating that the velocity is zero. This approach can significantly correct accumulated velocity errors and, indirectly, improve position accuracy.

For example, in a foot-mounted INS (often used for indoor pedestrian tracking), each time that the foot is on the ground flat during the stance phase of a step, its velocity in the global frame is effectively zero for a brief period. At the detected zero-velocity instants, a measurement update is performed in the EKF. Essentially, the filter is told that the true velocity is zero with some measurement noise $\mathbf{r}_k$. The Kalman gain will then adjust the state estimates: velocity errors are set to near zero, and accordingly, position error (which is correlated with velocity error in the state covariance) is also pulled back. Additionally, the filter can estimate and correct accelerometer biases that would otherwise cause a nonzero velocity drift if no external frame measurement is given.

ZUPT has been shown to effectively bound the positional drift in foot-mounted INS to a few percent of distance traveled, even without external sensors. However, it relies on the presence of frequent zero-velocity periods. In cases of wheeled robots or other platforms that may not come to a full stop regularly, ZUPT is less directly applicable (although one can sometimes use ZUPTs at deliberate stops or if the platform is known to stop occasionally).

6.6.2.2 Zero Angular Rate Update (ZARU)

The zero angular rate update (ZARU) is conceptually similar to ZUPT but applies to the rotational motion: it assumes that, at certain times, the device has (approximately) zero angular velocity. This is often the case during the stance phase of a foot in pedestrian dead reckoning, when the foot is flat on the ground and not twisting significantly. If a zero-angular-rate detector identifies that the rotational motion is below a threshold (i.e., any measured gyro output is predominantly bias, with true rotation negligibly small), one can perform a pseudo-measurement that the angular rate is zero [51].

In an EKF context, a ZARU can be implemented by observing the gyroscope measurements:

$$\mathbf{z}_{\text{zaru}} = 0,\ h(\mathbf{x}) = \boldsymbol{\omega}_b \tag{6.60}$$

for each axis of the gyroscope (or all three simultaneously as a vector measurement), where $\boldsymbol{\omega}_b$ is the angular rate of the body relative to the inertial frame (as measured by the gyro, in body axes). By updating the filter with

the information that true rotation is zero (within some noise tolerance), the filter primarily corrects the gyroscope bias states. Essentially, any observed gyro output is interpreted as bias. Over time, ZARU can significantly reduce gyro bias drift, which, in turn, stabilizes the heading (yaw) estimate that would otherwise drift. This is especially useful as yaw in an INS is otherwise unobservable without external references (no component of gravity provides yaw information, unlike roll and pitch).

ZARU is complementary to ZUPT in foot-mounted systems [51]. However, its effectiveness depends on the sensor and motion characteristics. If the threshold for "zero rotation" is too high, we might treat actual motion as bias, causing errors; if too low, we rarely apply ZARU. In normal walking, feet often have slight rotations even during stance (people can swivel their feet), limiting opportunities for true zero-rate updates. Therefore, some implementations combine ZARU with heuristic heading drift reduction methods that use partial observations of heading change when foot motion is low, rather than a strict zero update [52]. In summary, ZARU is a useful constraint to curtail gyro bias accumulation and heading drift in systems where the device experiences intervals of almost no rotation.

6.6.2.3 Nonholonomic Constraints (NHC)

Nonholonomic constraints (NHC) refer to constraints on motion that are nonintegrable as position constraints but restrict the velocities of a system. In the context of land vehicle navigation (and sometimes applied to pedestrian or other systems in a heuristic way), NHC usually implies that certain velocity components are zero in the vehicle's body frame. For example, a typical wheeled vehicle cannot slip sideways or jump, so its velocity in the lateral (sideways) and vertical directions relative to the vehicle frame should be zero (assuming no wheel slip and a level ground). This provides two constraints:

$$\tilde{\mathbf{v}}_k^S = \left[0,\ \tilde{v}_k^S, 0\right]^{\mathrm{T}} \tag{6.61}$$

Here, $\tilde{\mathbf{v}}_k^S$ represents the measurement in the speed coordinate system, assuming the x, y, and z axes point to the right, forward, and upward, respectively. This equation can be used as a pseudo-measurement in the EKF, similar to ZUPT, but specifically influencing certain components of the velocity.

By applying NHC updates, the filter can constrain lateral and vertical velocity errors to zero whenever the vehicle model is assumed valid. The primary effect is to prevent spurious growth of velocity in those unpowered directions and to correct small velocity errors before they lead to large position errors.

It also aids in estimating accelerometer biases: for instance, a slight bias that would cause a gradual buildup of a nonzero lateral velocity will be corrected when the NHC measurement forces the lateral velocity estimate back to zero.

The NHC are very powerful for vehicle INS during external information is lost (indoor or in tunnels), as they essentially keep the vehicle on track by not allowing the inertial solution to wander sideways. However, one must be cautious: NHC assumes no sideslip or vertical movement. In reality, vehicles can have minor wheel slip or suspension movements. If those are significant, treating them as zero can introduce errors. Still, with appropriate measurement noise tuning, the EKF can accommodate slight violations of the constraint without instability.

It is worth noting that similar constraints can be applied in other contexts: for example, a person walking might have a constraint that average velocity in the vertical direction is zero over a gait cycle (they come back to ground level), or robotic systems with specific kinematic constraints can incorporate them into the filter. The general idea is to incorporate any known nonholonomic behavior of the system as additional information to arrest drift.

6.6.3 External Aiding by Sensor Fusion

While motion constraints significantly mitigate drift, most navigation systems also incorporate external aiding sensors to provide absolute references. Sensor fusion combines the relative motion sensing of an IMU with other sensors that either do not drift or drift much less, ensuring bounded-error performance.

Ultimately, the most robust approach to drift mitigation is a fully integrated multisensor system. This can include an IMU, wheel odometer, magnetometer, barometer, RF receiver, and even cameras or LiDAR. Each sensor has different error characteristics and failure modes. The filter combines all available measurements to estimate the navigation state. For example, a Wi-Fi or BLE position will reset any residual drift in all INS degrees of freedom. Visual or LiDAR odometry can provide relative motion that bounds drift in between absolute positions. The INS in turn bridges gaps when those external signals are unavailable, providing continuous estimates. In such fusion, the INS essentially acts as the core integrator of motion, and all other sensors either constrain its errors or correct its output.

External aiding sensors significantly enhance the long-term accuracy of navigation systems by preventing unbounded error growth. Designing the filter involves accounting for the noise and bias characteristics of each sensor (process and measurement noise tuning) and possibly time synchronization

and alignment between sensors. The considerations of sensor fusion are detailed in Chapter 8.

6.7 Conclusions

Positioning based on proprioceptive and environmental sensors remains a critical area of research with substantial potential for improvement in accuracy and reliability. Future research directions are oriented around advancements in both hardware and software:

- *Advancements in sensor technology:* Efforts continue towards developing next-generation inertial sensors with significantly enhanced performance characteristics. Quantum-based inertial sensors, such as cold-atom interferometry gyroscopes and accelerometers, promise to deliver drastically reduced drift and higher precision, potentially achieving strategic-grade navigation performance. Although currently limited by size, complexity, and cost, ongoing advancements are expected to gradually make these technologies more accessible for practical applications.
- *Machine learning and data-driven approaches:* Recent developments in machine learning, particularly deep learning techniques, have shown promise in leveraging patterns within inertial data. Methods involving recurrent neural networks (RNNs) or transformer-based architectures can implicitly model complex human or vehicular motion dynamics, reducing positional drift significantly. These approaches, particularly suitable for pedestrian tracking and smartphone-based inertial odometry, still face challenges in terms of generalization and interpretability. Nevertheless, hybrid models integrating traditional Kalman filtering with learning-based anomaly detection and corrections represent an active and rapidly expanding research frontier.

Overall, combining cutting-edge sensor technology with sophisticated data-driven represents the most promising pathway forward in proprioceptive and environmental sensor-based positioning systems.

{AU: Word missing here?}

References

[1] Tao, Z., et al., "Mapping and Localization Using GPS, Lane Markings and Proprioceptive Sensors," *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013.

[2] Woodman, O. J. *An Introduction to Inertial Navigation*, No. UCAM-CL-TR-696, University of Cambridge, Computer Laboratory, 2007.

[3] El-Sheimy, N., and A. Youssef, "Inertial Sensors Technologies for Navigation Applications: State of the Art and Future Trends," *Satellite Navigation*, Vol. 1.1, 2020, pp. 1–21.

[4] El-Sheimy, N., and Y. Li, "Indoor Navigation: State of the Art and Future Trends," *Satellite Navigation*, Vol. 2.1, 2021, pp. 1–23.

[5] Roetenberg, D., "Inertial and Magnetic Sensing of Human Motion," These de doctorat, , 2006.

{AU: Provide university name}

[6] Titterton, D., and J. L. Weston, "Strapdown Inertial Navigation Technology," *IET*, Vol. 17, 2004.

[7] Brandt, A., and J. F. Gardner, "Constrained Navigation Algorithms for Strapdown Inertial Navigation Systems with Reduced Set of Sensors," *Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No. 98CH36207)*, Vol. 3, 1998.

[8] Friedland, B., "Analysis Strapdown Navigation Using Quaternions," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 5, 1978, pp. 764–768.

[9] Teslić, L., I. Škrjanc, and G. Klančar, "EKF-Based Localization of a Wheeled Mobile Robot on Structured Environments," *Journal of Intelligent & Robotic Systems*, Vol. 62, 2011, pp. 187–203.

[10] Cho, B. -S., et al., "A Dead Reckoning Localization System for Mobile Robots Using Inertial Sensors and Wheel Revolution Encoding." *Journal of Mechanical Science and Technology*, Vol. 25, 2011, pp. 2907–2917.

[11] Groves, P. D., "Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems [Book Review]," *IEEE Aerospace and Electronic Systems Magazine*, Vol. 30, No. 2, 2015, pp. 26–27.

[12] Wu, Y., C. Goodall, and N. El-Sheimy, "Self-Calibration for IMU/Odometer Land Navigation: Simulation and Test Results," *Proceedings of the 2010 International Technical Meeting of The Institute of Navigation*, 2010.

[13] Mikov, A., et al., "Sensor Fusion for Land Vehicle Localization Using Inertial MEMS and Odometry," *2019 IEEE International Symposium on Inertial Sensors and Systems (INERTIAL)*, 2019.

[14] Niu, X., S. Nassar, and N. El-Sheimy, "An Accurate Land-Vehicle MEMS IMU/GPS Navigation System Using 3D Auxiliary Velocity Updates," *Navigation*, Vol. 54, No. 3, 2007, pp. 177–188.

[15] Li, Y., et al., "Observability Analysis of Non-Holonomic Constraints for Land-Vehicle Navigation Systems," *Proceedings of the 25th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2012)*, 2012.

[16] Wang, Y., A. Chernyshoff, and A. M. Shkel, "Error Analysis of ZUPT-Aided Pedestrian Inertial Navigation," *2018 IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2018.

[17] Xiaofang, L., et al., "Applications of Zero-Velocity Detector and Kalman Filter in Zero Velocity Update for Inertial Navigation System," *Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference*, 2014.

[18] Wang, Y., and A. M. Shkel, "A Review on ZUPT-Aided Pedestrian Inertial Navigation," *2020 27th IEEE Saint Petersburg International Conference on Integrated Navigation Systems (ICINS)*, 2020.

[19] Skog, I., et al., "Zero-Velocity Detection—An Algorithm Evaluation," *IEEE Transactions on Biomedical Engineering*, Vol. 57, No. 11, 2010, pp. 2657–2666.

[20] Levi, R. W., and T. Judd, "Dead Reckoning Navigational System Using Accelerometer to Measure Foot Impacts," U.S. Patent No. 5,583,776, December 10, 1996.

[21] Yan, D., C. Shi, and T. Li, "An Improved PDR System with Accurate Heading and Step Length Estimation Using Handheld Smartphone," *The Journal of Navigation*, Vol. 75, No. 1, 2022, pp. 141–159.

[22] Chen, R., L. Pei, and Y. Chen, "A Smart Phone Based PDR Solution for Indoor Navigation," *Proceedings of the 24th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2011)*, 2011.

[23] Díez, L. E., et al., "Step Length Estimation Methods Based on Inertial Sensors: A Review," *IEEE Sensors Journal*, Vol. 18, No. 17, 2018, pp. 6908–6926.

[24] Weinberg, H., "Using the ADXL202 in Pedometer and Personal Navigation Applications," *Analog Devices AN-602 Application Note*, Vol. 2, No. 2, 2002, pp. 1–6.

[25] Kim, J. W., et al., "A Step, Stride and Heading Determination for the Pedestrian Navigation System," *Journal of Global Positioning Systems*, Vol. 3, No. 1-2, 2004, pp. 273–279.

[26] Wang, Q., et al., "Personalized Stride-Length Estimation Based on Active Online Learning," *IEEE Internet of Things Journal*, Vol. 7, No. 6, 2020, pp. 4885–4897.

[27] Vandermeeren, S., and H. Steendam, "Deep-Learning-Based Step Detection and Step Length Estimation with a Handheld IMU," *IEEE Sensors Journal*, Vol. 22, No. 24, 2022, pp. 24205–24221.

[28] Ping, Z., et al., "Pedestrian Stride-Length Estimation Based on Bidirectional LSTM Network," *2020 Chinese Automation Congress (CAC)*, 2020.

[29] Cho, D. -K., et al., "Autogait: A Mobile Platform That Accurately Estimates the Distance Walked," *2010 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2010.

[30] Wisth, D., M. Camurri, and M. Fallon, "Robust Legged Robot State Estimation Using Factor Graph Optimization," *IEEE Robotics and Automation Letters*, Vol. 4, No. 4, 2019, pp. 4507–4514.

[31] Rubio, F., F. Valero, and C. Llopis-Albert, "A Review of Mobile Robots: Concepts, Methods, Theoretical Framework, and Applications," *International Journal of Advanced Robotic Systems*, Vol. 16, No. 2, 2019, 1729881419839596.

[32] Wang, Q., et al., "Recent Advances in Pedestrian Inertial Navigation Based on Smartphone: A Review," *IEEE Sensors Journal*, 2022.

[33] Herath, S., H. Yan, and Y. Furukawa, "Ronin: Robust Neural Inertial Navigation in the Wild: Benchmark, Evaluations, & New Methods," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3146–3152.

[34] Bai, S., W. Wen, and C. Shi, "3DIO: Low-Drift 3D Deep-Inertial Odometry for Indoor Localization Using an IMU," *IEEE Internet of Things Journal*, 2024.

[35] Foster, C. C., and G. H. Elkaim, "Extension of a Two-Step Calibration Methodology to Include Nonorthogonal Sensor Axes," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 44, No. 3, 2008, pp. 1070–1078.

[36] Suh, Y. S., "Orientation Estimation Using a Quaternion-Based Indirect Kalman Filter with Adaptive Estimation of External Acceleration," *IEEE Transactions on Instrumentation and Measurement*, Vol. 59, No. 12, 2010, pp. 3296–3305.

[37] Madgwick, S. O. H., A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG Orientation Using a Gradient Descent Algorithm," *2011 IEEE International Conference on Rehabilitation Robotics*, 2011, pp. 1–7.

[38] Sabatini, A. M., and V. Genovese, "A Sensor Fusion Method for Tracking Vertical Velocity and Height Based on Inertial and Barometric Altimeter Measurements," *Sensors*, Vol. 14, No. 8, 2014, pp. 13324–13347.

[39] El-Sheimy, N., H. Hou, and X. Niu, "Analysis and Modeling of Inertial Sensors Using Allan Variance," *IEEE Transactions on Instrumentation and Measurement*, Vol. 57, No. 1, 2007, pp. 140–149.

[40] Ouyang, G., and K. Abed-Meraim, "Analysis of Magnetic Field Measurements for Indoor Positioning," *Sensors*, Vol. 22, No. 11, 2022, p. 4014.

[41] Fetzer, T., et al., "Using Barometer for Floor Assignation Within Statistical Indoor Localization," *Sensors*, Vol. 23, No. 1, 2022, p. 80.

[42] Chong, K. S., and L. Kleeman, "Accurate Odometry and Error Modelling for a Mobile Robot," *Proceedings of IEEE International Conference on Robotics and Automation*, Vol. 4, 1997, pp. 2783–2788.

[43] Li, B., B. Harvey, and T. Gallagher, "Using Barometers to Determine the Height for Indoor Positioning," *International IEEE Conference on Indoor Positioning and Indoor Navigation*, 2013, pp. 1–7.

[44] Pan, J., C. Zhang, and Q. Cai, "An Accurate Calibration Method for Accelerometer Nonlinear Scale Factor on a Low-Cost Three-Axis Turntable," *Measurement Science and Technology*, Vol. 25, No. 2, 2014, 025102.

[45] Bai, S., et al., "Improved Preintegration Method for GNSS/IMU/In-Vehicle Sensors Navigation Using Graph Optimization," *IEEE Transactions on Vehicular Technology*, Vol. 70, No. 11, 2021, pp. 11446–11457.

[46] Bai, S., et al., "A System-Level Self-Calibration Method for Installation Errors in a Dual-Axis Rotational Inertial Navigation System," *Sensors*, Vol. 19, No. 18, 2019, p. 4005.

[47] Syed, Z. F., et al., "A New Multi-Position Calibration Method for MEMS Inertial Navigation Systems," *Measurement Science and Technology*, Vol. 18, No. 7, 2007, p. 1897.

[48] Wang, L., et al., "Improving the Navigation Performance of the MEMS IMU Array by Precise Calibration," *IEEE Sensors Journal*, Vol. 21, No. 22, 2021, pp. 26050–26058.

[49] Borenstein, J., and L. Feng, "UMBmark: A Method for Measuring, Comparing, and Correcting Dead-Reckoning Errors in Mobile Robots, " , 1994.

{AU: Provide complete publication information}

[50] Martinelli, A., N. Tomatis, and R. Siegwart, "Simultaneous Localization and Odometry Self Calibration for Mobile Robot," *Autonomous Robots*, Vol. 22, 2007, pp. 75–85.

[51] Rajagopal, S., "Personal Dead Reckoning System with Shoe Mounted Inertial Sensors," Master's Degree Project, , Stockholm, Sweden, 2008.

{AU: Provide university name}

[52] Zampella, F., et al., "Unscented Kalman Filter and Magnetic Angular Rate Update (MARU) for an Improved Pedestrian Dead-Reckoning," *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*, 2012, pp. 129–139.