2

Fundamentals of Indoor Positioning and Navigation Systems

This chapter introduces the fundamental knowledge required to address the estimation problems encountered in IPIN applications. Section 2.1 covers the coordinate systems commonly adopted and the transformations among these systems, which are essential for accurate position representation and system integration. Section 2.2 focuses on the definition and representation of attitude, describing how an object's orientation is characterized using different mathematical forms, including Euler angles, rotation matrices, and quaternions. These foundational topics set the stage for the estimation methods and algorithms discussed in later chapters.

2.1 Coordinate Systems and Transformations

The fundamental question to be solved for positioning is: "Where am I?" To answer this question, one needs a system to indicate our position with respect to a specific referencing position by coordinate values (i.e., a coordinate system). Besides, different applications or sensors have their own preferring coordinate system, which are usually not the same. Therefore, it is necessary to establish transformations between different coordinate systems. This section covers the essential knowledge of coordinate systems for IPIN, whereas a comprehensive introduction can be found in [1]. An overview of the coordinate systems for IPIN is shown in Figure 2.1. The coordinate systems are categorized into local coordinate system and global coordinate system, depending on whether the system is applicable in a certain area around a reference or universally applicable with respect to the Earth.

2.1.1 Local Body Coordinate System

A popular coordinate system for IPIN applications is the local body coordinate system [2], which describes the movement of an object from its own point of view or describes the position of an object from another object's point of view. The local body coordinate system is a Cartesian coordinate system fixed with respect to a reference body, either being static or under movements. An example is shown as Figure 2.2. The origin B can be defined as the center of mass of the body. For IPIN applications, users are often concerned with objects located above themselves or above a reference point, such as ceiling-mounted beacons. Consequently, the Z-axis typically points upward from the origin. Following the right-hand rule and the commonly used convention, the X-axis is defined as pointing to the right of the body, while the Y-axis points forward. The position of the target object A can be represented by its coordinate values along these three axes with respect to the origin, as follows:

$$\mathbf{x}_{\text{BODY}} = \begin{bmatrix} x_{\text{BODY}} & y_{\text{BODY}} & z_{\text{BODY}} \end{bmatrix}^{\text{T}} \tag{2.1}$$

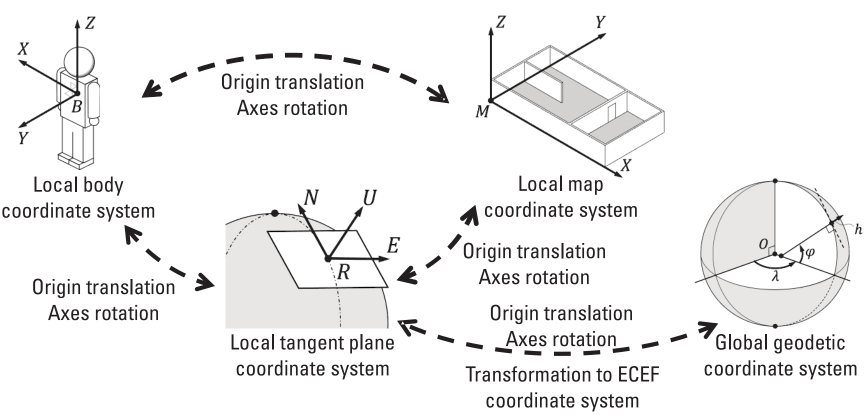

Figure 2.1 Overview of the coordinate systems and transformations.

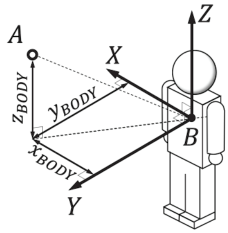

Figure 2.2 Local body coordinate system.

Similarly, the movement of an object can be represented by its projection component on each axis. Noted that definition of local body coordinate system here is different from that in aviation applications, where *Z*-axis is pointing down to better describe objects below an aircraft.

2.1.2 Local Map Coordinate System

IPIN applications usually prefer a local coordinate system that can represent the position information with respect to the local environment feature. Such a coordinate system is usually constructed based on a "map" of the related area (e.g., the building plan), as Figure 2.3 shows. Similar to the local body coordinate system, a reference point *M* is selected as the origin to build a Cartesian coordinate system that the axes are along with the map features, such as the walls. Then the position of an object can be represented on this map by the coordinate values, as follows:

$$\mathbf{x}_{\mathrm{MAP}} = \begin{bmatrix} x_{\mathrm{MAP}} & y_{\mathrm{MAP}} & z_{\mathrm{MAP}} \end{bmatrix}^{\mathrm{T}} \tag{2.2}$$

For many cases, the *XY*-planes of the local map coordinate system and the local body coordinate system are parallel, with a yaw rotation about *Z*-axis.

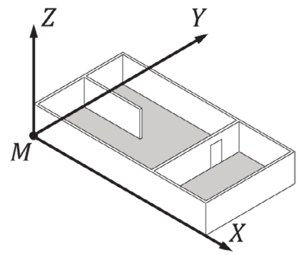

Figure 2.3 Local map coordinate system.

Then, for an object A, the coordinate transformation from the local map coordinate system to the local body coordinate system is achieved as follows:

$$\mathbf{x}_{\mathrm{BODY},A} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(\mathbf{x}_{\mathrm{MAP},A} - \mathbf{x}_{\mathrm{MAP},B}\right) \tag{2.3}$$

where ψ is the yaw angle from the X and Y axes in the local map coordinate system to align with that in the local body coordinate system. $\mathbf{x}_{\mathrm{MAP},A}$ and $\mathbf{x}_{\mathrm{MAP},B}$ are the position of object A and the local body coordinate origin B in the local map coordinate system, respectively. When the XY-planes of the local map coordinate system and the local body coordinate system are not parallel, an additional rotation is required to complete the coordinate transformation. The detailed representation and computation of these rotations (attitude angles) will be introduced in Section 2.2.

2.1.3 Local Tangent Plane Coordinate System: East, North, Up and North, East, Down Coordinate System

Some applications prefer to describe the position of an object with respect to the geographical features on the Earth's ground like the directions of East and North. For convenience, if such an application is within a certain area (e.g., within hundreds of meters), the ground is approximated as a tangent plane on a selected reference location on the Earth, neglecting the ground curvature. Then a local Cartesian coordinate system is constructed on this reference location, in which the position of the target object is expressed by coordinate values with respect to this reference point. A widely used local coordinate system is the East, North, Up (ENU) coordinate system, as Figure 2.4 shows. On the selected reference point R, a plane tangent to the Earth's ellipsoidal surface can be obtained. Then two orthogonal axes of a Cartesian coordinate system are resolved on this tangent plane, including the N-axis pointing to the North Pole and the E-axis pointing to the East direction with respect to the reference point. Following the right-hand rule, the third axis is the U-axis perpendicular to the tangent plane, pointing outwards from the ellipsoid. Then the position of object A can be represented by the coordinate values x_{ENU}, y_{ENU}, and z_{ENU} on the E, N, and U axes respect to the reference point, using

$$\mathbf{x}_{\mathrm{ENU}} = \begin{bmatrix} x_{\mathrm{ENU}} & y_{\mathrm{ENU}} & z_{\mathrm{ENU}} \end{bmatrix}^{\mathrm{T}} \tag{2.4}$$

By approximating Earth's surface as the flat plane parallel to the tangent plane, the E-axis and N-axis are parallel to the ground. Then the coordinate

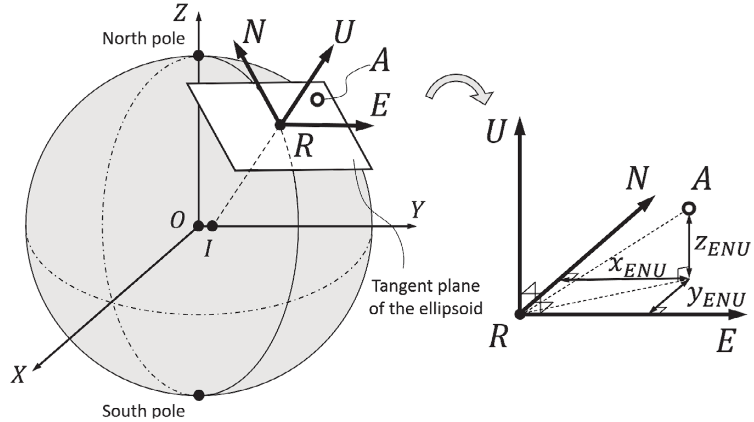

Figure 2.4 ENU coordinate system.

values can intuitively indicate how far an object on the ground is located with respect to the reference position in terms of East and North directions.

An alternative local tangent plane coordinate systems for aviation applications is the North, East, Down (NED) coordinate system [1], where its axes are point to the North, East, and down directions. It can be transformed from the ENU coordinate system by rotations about Y-axis and Z-axis as follows:

$$\begin{aligned}\mathbf{x}_{\text{NED}} = \mathbf{C}_{\text{ENU}}^{\text{NED}}\mathbf{x}_{\text{ENU}} &= \begin{bmatrix} \cos\frac{\pi}{2} & \sin\frac{\pi}{2} & 0 \\ -\sin\frac{\pi}{2} & \cos\frac{\pi}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \cos\pi & 0 & -\sin\pi \\ 0 & 1 & 0 \\ \sin\pi & 0 & \cos\pi \end{bmatrix}\mathbf{x}_{\text{ENU}} \\ &= \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}\mathbf{x}_{\text{ENU}}\end{aligned} \tag{2.5}$$

In coordinate transformations, especially when aligning the local body coordinate system with the local tangent plane coordinate system (ENU), it is often necessary to describe how an object is oriented relative to the chosen reference frame. This orientation, known as attitude, can be parameterized in various ways. In the ENU coordinate system, the attitude of an object can be represented by the deviations of its local body coordinate system axes from the ENU axes. This is typically expressed in terms of Euler angles, describing 3 degrees of freedom (3-DOF): pitch, roll, and yaw. For example, as shown in Figure 2.5 (left), the yaw angle ψ (positive in the counterclockwise direction following the right-hand rule) describes the rotation of the local X- and Y-axes about the Z-axis away from the E- and N-axes of the ENU system. Similarly, pitch and roll angles describe the rotation of the local body coordinate system about its X- and Y-axes relative to the ENU frame.

It is important to note that attitude (orientation) is different from bearing, which describes the direction toward which an object is located. Bearing in the ENU system is often represented by the elevation angle, the angle from the ground plane to the object's line of sight (positive when above the ground), and the azimuth angle, the angle from the North direction to the object's projection on the ground plane (positive counter-clockwise). Figure 2.5 (right) shows an example of bearing represented by azimuth angle. For applications requiring different reference planes or axes, bearing can also be defined relative to other coordinate system elements. It is worth noting that, in the NED coordinate system, the azimuth angle is defined as the down axis and has the opposite convention compared to the ENU system. A more detailed discussion of attitude representation, including alternative parameterizations such as rotation matrices and quaternions, is provided in Section 2.2.

For the position of object A, the transformation between the ENU coordinate system and the local body coordinate system can be achieved by

$$\mathbf{x}_{\mathrm{BODY},A} = \mathbf{C}_{\mathrm{ENU}}^{\mathrm{BODY}}\left(\mathbf{x}_{\mathrm{ENU},A} - \mathbf{x}_{\mathrm{ENU},B}\right) \tag{2.6}$$

$$\mathbf{x}_{\mathrm{ENU},A} = \mathbf{C}_{\mathrm{BODY}}^{\mathrm{ENU}}\left(\mathbf{x}_{\mathrm{BODY},A} - \mathbf{x}_{\mathrm{BODY},R}\right) \tag{2.7}$$

where $\mathbf{C}_{\mathrm{ENU}}^{\mathrm{BODY}}$ is the rotation matrix based on the attitude pitch, roll, and yaw angle of the body coordinate system from the ENU coordinate system. $\mathbf{x}_{\mathrm{ENU},B}$ is the position of the origin of the local body coordinate system in the ENU coordinate system. $\mathbf{x}_{\mathrm{BODY},R}$ is the position of the origin of the ENU coordinate system in the local body coordinate system.

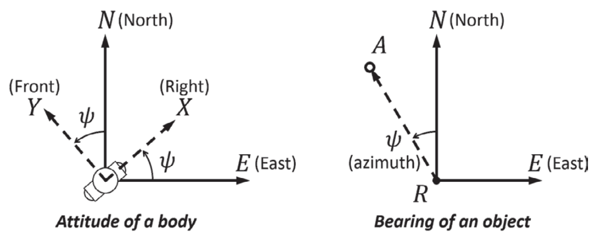

Figure 2.5 (Left) Attitude representation of a body by yaw angle. (Right) Bearing representation of an object by azimuth angle.

2.1.4 Geodetic Coordinate System: Latitude, Longitude, Height Coordinate System

Another popular coordinate system for positioning in a large-scale area is the geodetic coordinate system [1], in which the ground curvature cannot be neglected. Then the Earth is modeled by an ellipsoid, as Figure 2.6 shows. One of the popular standards is the World Geodetic System 1984 (WGS-84), which defines necessary parameters for the coordinate system, such as the equatorial radius, the flattening, and the geoid [3]. Based on such a standard, the position of the concerning object on the Earth can be described by the geodetic coordinates: latitude φ, longitude λ, and ellipsoid height h. For an object (or a user) A, the unique line $\overline{AI}$ passing through the object normal to the ellipsoid can be obtained, which intersects the ellipsoid by point P. The distance from point P to point A denotes the shortest distance between the object and the Earth, namely the ellipsoid height h. Then the angle between the normal line $\overline{AI}$ and the equatorial plane is the geodetic latitude φ, which is positive for the Northern Hemisphere and is negative for the Southern Hemisphere. Note that the intersection of this normal line and the equatorial plane, I, may not coincide with the ellipsoid center O. Finally, the longitude λ is the angle from the plane of Prime Meridian to the projection of the normal line $\overline{AI}$ on the equatorial plane. The longitude λ is positive for the East of the Prime Meridian and is negative for the West of the Prime Meridian. Then the position of object A can be represented by

$$\mathbf{x}_{LLH} = \begin{bmatrix} \varphi & \lambda & h \end{bmatrix}^{\mathrm{T}} \tag{2.8}$$

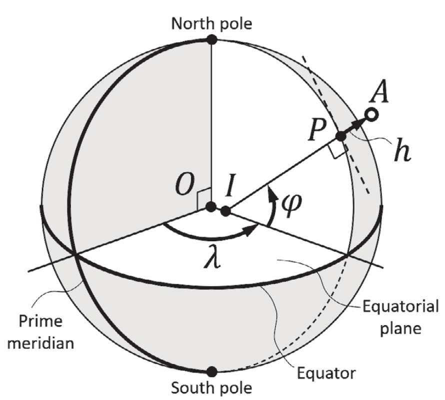

Figure 2.6 Geodetic coordinate system.

Meanwhile, the actual surface of the Earth is fluctuated compared to the ellipsoid surface. An alternative measure is the height above the geoid (or the mean sea level), which considers the natural curvature of the Earth's surface. This is a global coordinate system that describes the position of an object with respect to the Earth.

The position expression can be transformed between the global geodetic coordinate system and the local ENU coordinate system. Note that the global geodetic coordinate system is not a Cartesian coordinate system. Thus, a global Cartesian coordinate system, the Earth-centered Earth-fixed (ECEF) coordinate system, is needed during the transformation in between. A detailed introduction of the ECEF coordinate system can be found in [2]. The position transformation from the geodetic coordinate system in (2.1) to the ECEF coordinate system is achieved by

$$\mathbf{x}_{\mathrm{ECEF}} = \begin{bmatrix} \dfrac{a\cos\lambda}{\sqrt{1+\left(1-e^2\right)\tan^2\varphi}} + h\cos\lambda\cos\varphi \\ \dfrac{a\sin\lambda}{\sqrt{1+\left(1-e^2\right)\tan^2\varphi}} + h\sin\lambda\cos\varphi \\ \dfrac{a\left(1-e^2\sin\varphi\right)}{\sqrt{1-e^2\sin^2\varphi}} + h\sin\varphi \end{bmatrix} \tag{2.9}$$

where a and e are the Earth's ellipsoid semimajor axis and eccentricity, respectively. Then the position of an object A in the ENU coordinate system with a selected reference point R as the origin is obtained by

$$\begin{aligned} \mathbf{x}_{\mathrm{ENU},A} &= \mathbf{C}_{\mathrm{ECEF}}^{\mathrm{ENU}}\left(\mathbf{x}_{\mathrm{ECEF},A} - \mathbf{x}_{\mathrm{ECEF},R}\right) \\ &= \begin{bmatrix} -\sin\lambda & \cos\lambda & 0 \\ -\sin\varphi\cos\lambda & -\sin\varphi\sin\lambda & \cos\varphi \\ \cos\varphi\cos\lambda & \cos\varphi\sin\lambda & \sin\varphi \end{bmatrix}\left(\mathbf{x}_{\mathrm{ECEF},A} - \mathbf{x}_{\mathrm{ECEF},R}\right) \end{aligned} \tag{2.10}$$

where $\mathbf{x}_{\mathrm{ECEF},A}$ and $\mathbf{x}_{\mathrm{ECEF},R}$ are the positions of object A and reference R in the ECEF coordinate system, respectively, based on (2.9). $(\mathbf{x}_{\mathrm{ECEF},A} - \mathbf{x}_{\mathrm{ECEF},R})$ can be regarded as the translation of the origin from the Earth ellipsoid center to the reference point R. $\mathbf{C}_{\mathrm{ECEF}}^{\mathrm{ENU}}$ is the rotation matrix, which rotates the axes in the ECEF coordinate system to align with the East, North, and up directions. The position transformation from the local ENU coordinate system to the global geodetic coordinate system can be achieved via the ECEF coordinate system with an iteration method, which can be found in [2]. Note that the global geodetic coordinate system is less popular for IPIN applications, as it is not intuitive to relate the solution with the distance information we care.

{AU: The last part of this sentence is unclear; please reword for clarity}

2.2 Attitude: Definition and Representation

2.2.1 Definition

In the context of IPIN applications, attitude refers to the orientation of an object (or its local body coordinate frame) relative to a chosen reference coordinate system (e.g., a local map or tangent plane coordinate system). Attitude is a purely rotational concept, distinct from position, and it describes how the object is oriented in 3-D space. For a rigid body in 3-D space, attitude is characterized by 3-DOF, representing rotations about three orthogonal axes of the local body coordinate system.

These three rotations are commonly described by roll, pitch, and yaw angles. The precise assignment of axes depends on the coordinate system convention adopted. For example, in a right-forward-up (*X*-right, *Y*-forward, *Z*-up) coordinate system, roll is typically defined as rotation about the *Y*-axis, pitch about the *X*-axis, and yaw about the *Z*-axis. The exact rotation sequence and axis assignment may vary by convention (e.g., aerospace versus robotics). Nonetheless, these three angles collectively describe the orientation of the object relative to the reference frame. Figure 2.7 illustrates these three principal rotations defining attitude (roll, pitch, yaw), showing the object's rotations about its *Y*, *X*, and *Z*-axes. Figure 2.7 also shows an example of how a yaw rotation transforms the reference coordinate system.

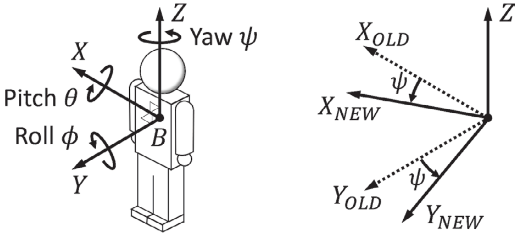

Figure 2.7 (Left) Attitude movement. (Right) Example of transformation from old to new coordinate systems by the yaw angle.

2.2.2 Representation

There are multiple ways to represent a 3-D orientation (attitude) in mathematical terms. The three major representation methods discussed here are Euler angles, rotation matrices, and quaternions [4]. Each representation has a different form, respectively: a set of three angles, a 3×3 matrix, and a 4-parameter unit quaternion, and each comes with its own advantages, limitations, and theoretical underpinnings. All these methods are equivalent in the sense that they can describe any possible orientation in 3-D space, and one can convert from one representation to another. However, they differ in interpretability, computational convenience, presence or absence of singularities, and compactness. We will describe each in turn, providing definitions and the underlying theory, followed by a comparative summary.

2.2.2.1 Euler Angles

The attitude movement of an object can also be described in terms of a local body coordinate system, as Figure 2.7 (left) shows. Following the right-hand rule, rotations on X, Y, and Z-axes are denoted by the pitch angle θ, roll angle ϕ, and yaw angle ψ, respectively. Thus, the 3-DOF angular movement of an object can be represented correspondingly. Besides, the attitude movement of an object can be regarded as a transformation between the old and the new coordinate systems, as Figure 2.7 (right) shows.

However, Euler angles have important limitations, notably the issue of singularities. Since the space of orientations is not topologically equivalent to a 3-D Euclidean space, no single continuous set of three parameters can cover all possible orientations without discontinuities. In practice, this results in a gimbal lock, a phenomenon where, at certain critical orientations, two of the rotation axes align and the system loses 1 DOF. In the standard yaw-pitch-roll sequence, this singularity occurs at $\theta = \pm 90°$; at that attitude, the yaw and roll axes coincide, and an increment of yaw produces the same effect as an increment of roll. Consequently, the system effectively “locks up” with only two independent rotation directions instead of three. As an example, at $\theta = 90°$, changing yaw or roll results in the same motion, making those angles indistinguishable. As a result, Euler angle representations become unreliable near these singular configurations, small changes in orientation can lead to large, abrupt changes in the angle values or even make them undefined. Additionally, Euler angles complicate certain calculations: rotation composition requires applying the appropriate rotation matrices rather than simply adding angles, and the noncommutative nature of rotations necessitates strict adherence to the specified rotation order. Nonetheless, Euler angles remain popular due to their intuitive interpretability and minimal parameter count. In summary, while Euler angles are highly interpretable, their main drawbacks include the presence of singularities (gimbal lock) and the complexity of composing rotations.

2.2.2.2 Rotation Matrix

Many applications may prefer a coordinate system with a customized orientation for better position expression, which requires rotation adjustments on both three axes from a standard coordinate system. This adjustment can be regarded as changing the basis of a vector space in linear algebra while preserving the distance of a point, as multiplying a transformation matrix:

$$\mathbf{x}_{\mathrm{NEW},A} = \mathbf{C}_{\mathrm{OLD}}^{\mathrm{NEW}} \mathbf{x}_{\mathrm{OLD},A} \tag{2.11}$$

where $\mathbf{x}_{\mathrm{OLD},A}$ and $\mathbf{x}_{\mathrm{NEW},A}$ denote the position of object A before and after the transformation. $\mathbf{C}_{\mathrm{OLD}}^{\mathrm{NEW}}$ is the transformation matrix where the subscript and superscript indicate the coordinate systems before and after the transformation, namely the rotation matrix. As the rotation matrix is orthogonal, it has the following property:

$$\mathbf{C}_{\mathrm{OLD}}^{\mathrm{NEW}} = \left(\mathbf{C}_{\mathrm{NEW}}^{\mathrm{OLD}}\right)^{-1} = \left(\mathbf{C}_{\mathrm{NEW}}^{\mathrm{OLD}}\right)^{\mathrm{T}} \tag{2.12}$$

where the superscript T denotes the transpose of a matrix. The transformation can be applied backwards by multiplying the transpose of the rotation matrix. For a Cartesian coordinate system, such as the local body coordinate system, its rotation about each axis with the corresponding Euler angle can be achieved by multiplying a transformation matrix. As shown in the example in Figure 2.8, the position of object A in the new coordinate system XYZ^* after a yaw rotation ψ can be represented by the coordinate values from the old coordinate system XYZ as follows:

$$\mathbf{x}^* = \begin{bmatrix} x^* \\ y^* \\ z^* \end{bmatrix} = \begin{bmatrix} x\cos\psi + y\sin\psi \\ -x\sin\psi + y\cos\psi \\ z \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.13}$$

where the superscript $*$ denotes the variable in the new coordinate system. Note that the coordinate value on the Z-axis remained unchanged after the rotation. Thus, the rotation matrix in (2.11) representing the rotation of XY-plane to new XY^*-plane about the Z-axis with a yaw angle ψ is

$$\mathbf{C}_{XY}^{XY^*} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.14}$$

Similarly, the rotation matrix of the XZ-plane to a new XZ^*-plane about the Y-axis by a roll angle ϕ is:

$$\mathbf{C}_{XZ}^{XZ^*} = \begin{bmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{bmatrix} \tag{2.15}$$

The rotation matrix of the YZ-plane to a new YZ^*-plane about the X-axis by a pitch angle θ is:

$$\mathbf{C}_{YZ}^{YZ^*} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \tag{2.16}$$

Then the rotation matrix with all three axes from the old XYZ coordinate to the new XYZ^* is represented by Euler angles as below, following the sequence of yaw rotation, row rotation, and pitch rotation (i.e., about the Z-, Y-, and X-axes):

$$\begin{aligned} \mathbf{C}_{\mathrm{OLD}}^{\mathrm{NEW}} &= \mathbf{C}_{YZ}^{YZ^*}\mathbf{C}_{XZ}^{XZ^*}\mathbf{C}_{XY}^{XY^*} \\ &= \begin{bmatrix} \cos\phi\cos\psi & \cos\phi\sin\psi & -\sin\phi \\ -\cos\theta\sin\psi + \sin\theta\sin\phi\cos\psi & \cos\theta\cos\psi + \sin\theta\sin\phi\sin\psi & \sin\theta\cos\phi \\ \sin\theta\sin\psi + \cos\theta\sin\phi\cos\psi & -\sin\theta\cos\psi + \cos\theta\sin\phi\sin\psi & \cos\theta\cos\phi \end{bmatrix} \end{aligned} \tag{2.17}$$

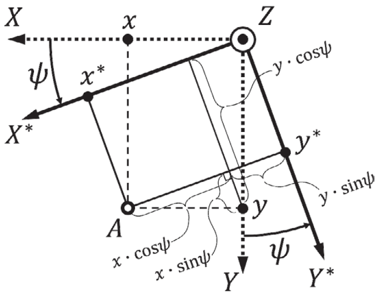

Figure 2.8 Example of position representation after yaw rotation. The superscript * denotes the variable in the new coordinate system.

Note that the rotations based on Euler angles (as matrix multiplication) are not commutative. For a general case, two coordinate systems are not necessary to have the same origin, which needs a translation of the origin to perform coordinate transformation. Thus, the position representation of object A in a new coordinate system can be converted from its position $\mathbf{x}_{\mathrm{OLD},A}$ in the old coordinate system by

$$\mathbf{x}_{\mathrm{NEW},A} = \mathbf{C}_{\mathrm{OLD}}^{\mathrm{NEW}}\left(\mathbf{x}_{\mathrm{OLD},A} - \mathbf{x}_{\mathrm{OLD},O_{\mathrm{NEW}}}\right) \tag{2.18}$$

where $\mathbf{x}_{\mathrm{OLD},O_{\mathrm{NEW}}}$ is the origin position of the new coordinate system in the old coordinate system.

Rotation matrices are also straightforward to use in transformations. The downside is that rotation matrices are less compact than minimal representations (since 9 numbers are stored, or 3 × 3 matrix, although only 3 are independent). They also may require maintaining orthonormality in a computational system; for example, if one integrates many small rotations (from gyroscope readings, say), numerical errors might cause the matrix to drift away from perfect orthogonality and require re-orthonormalization. In terms of interpretability, rotation matrices are not as immediately intuitive as Euler angles; one can interpret the columns or rows as basis vectors directions, but it is not as simple as "an angle about an axis" for a layperson. However, they are very convenient mathematically: combining rotations is just matrix multiplication, and many algorithms in navigation and robotics use rotation matrices internally due to their lack of singularity and direct applicability in coordinate transformation calculations. They are also the standard representation in many scientific and engineering libraries. To summarize, a rotation matrix provides a complete and nonsingular description of attitude and is well-suited for computations, at the expense of using more parameters (with constraints) and being less human-readable than Euler angles.

2.2.2.3 Quaternion

There are conditions that the rotations cannot be represented by means of Euler angles. For example, when the roll angle ϕ equals $\pi/2$, the rotation matrix in (2.17) is represented as

$$\mathbf{C}_{\mathrm{OLD}}^{\mathrm{NEW}} = \begin{bmatrix} 0 & 0 & -1 \\ \sin(\theta-\psi) & \cos(\theta-\psi) & 0 \\ \cos(\theta-\psi) & -\sin(\theta-\psi) & 0 \end{bmatrix} \tag{2.19}$$

The Euler angles of pitch and yaw have the same effect on the final rotation representation, in which 1 DOF is lost. Thus, the attitude related to the lost DOF cannot be represented. Note that this gimbal lock is more likely to occur for IPIN with a portable device, which is expected to have all possible orientations during a practical application. To avoid it, the quaternion is usually employed. Quaternion is a 4-dimensional (4-D) complex number $\mathbf{q}_{\mathrm{OLD}}^{\mathrm{NEW}} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$, which is usually denoted by only the coefficients using

$$\mathbf{q}_{\mathrm{OLD}}^{\mathrm{NEW}} = \left[q_0, q_1, q_2, q_3\right] \tag{2.20}$$

It is popular for describing 3-D rotations. A comprehensive introduction can be found in [5]. The rotation matrix corresponding to a set of quaternions and its inverse representation are given by

$$\mathbf{C}_{\mathrm{OLD}}^{\mathrm{NEW}} = \begin{bmatrix} 1-2q_2^2-2q_3^2 & 2q_1q_2+2q_0q_3 & 2q_1q_3-2q_0q_2 \\ 2q_1q_2-2q_0q_3 & 1-2q_1{}^2-2q_3{}^2 & 2q_2q_3+2q_0q_1 \\ 2q_1q_3+2q_0q_2 & 2q_2q_3-2q_0q_1 & 1-2q_1^2-2q_2^2 \end{bmatrix} \tag{2.21}$$

The Euler angles corresponding to a set of quaternions and its inverse representation are given by

$$\begin{aligned} \theta &= \tan^{-1}\frac{2\left(q_0q_1+q_2q_3\right)}{1-2\left(q_1^2+q_2^2\right)} \\ \phi &= \sin^{-1}2\left(q_0q_2-q_3q_1\right) \\ \psi &= \tan^{-1}\frac{2\left(q_0q_3+q_1q_2\right)}{1-2\left(q_2^2+q_3^2\right)} \end{aligned} \tag{2.22}$$

$$\begin{aligned} q_0 &= \cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\phi}{2}\right)\cos\left(\frac{\psi}{2}\right)+\sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\phi}{2}\right)\sin\left(\frac{\psi}{2}\right) \\ q_1 &= \sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\phi}{2}\right)\cos\left(\frac{\psi}{2}\right)-\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\phi}{2}\right)\sin\left(\frac{\psi}{2}\right) \\ q_2 &= \cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\phi}{2}\right)\cos\left(\frac{\psi}{2}\right)+\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\phi}{2}\right)\sin\left(\frac{\psi}{2}\right) \\ q_3 &= \cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\phi}{2}\right)\sin\left(\frac{\psi}{2}\right)-\sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\phi}{2}\right)\cos\left(\frac{\psi}{2}\right) \end{aligned} \tag{2.23}$$

A key advantage of the quaternion representation is that it avoids gimbal lock singularities. By using four parameters instead of three, quaternions provide smooth and continuous coverage of all possible orientations in 3-D space. The only caveat is the double-cover property represent the same physical orientation, so one must account for sign ambiguity when comparing quaternions. Additionally, quaternions must remain unit-length, so algorithms often renormalize them periodically to mitigate drift from numerical integration. Compared to rotation matrices, quaternions are more compact (4 numbers versus 9) and computationally efficient, especially for combining rotations and performing interpolation. However, they lack intuitive interpretability: their components do not directly correspond to roll, pitch, and yaw angles, requiring conversion for human-understandable descriptions. Despite this, quaternions are widely used in navigation systems due to their stability, efficiency, and freedom from singularities, even if outputs are often ultimately presented in Euler angles for readability.

Table 2.1 compares the three attitude representation methods. For IPIN applications, quaternions representation is the most popular option due to its computational efficiency and its nature on nonsingularity issues. In the rest of the book, we use rotation matrix representation to show the derivation for clarity.

Table 2.1
Comparison of Attitude Representation Methods

Representation	**Interpretability**	**Computational Efficiency**	**Numerical Stability**	**Singularity Issues**
Euler angles	Highly interpretable; it directly corresponds to yaw, pitch, roll rotations that are easy to visualize and understand	Moderate for computation; it has only 3 parameters, but combining rotations requires careful use of rotation formulas (not a simple vector addition); efficient for small rotations but composition can be cumbersome	Moderate for stability; numerical drift can occur in long computations, especially near singular configurations, requiring careful numerical treatment	Yes; it is susceptible to gimbal lock (loss of a DOF) at specific orientations (e.g., pitch = ±90° in a typical sequence)

Table 2.1
(Cont.)

Representation	Interpretability	Computational Efficiency	Numerical Stability	Singularity Issues
Rotation matrix	Moderate interpretability; it contains direction cosines of axes; harder to read at a glance, but rows/columns indicate orientation of basis vectors	Good for many operations; applying rotation to vectors is a simple matrix multiply; composition of rotations is matrix multiplication (costlier than quaternion multiplication, but still straightforward)	Moderate for stability; numerical drift can occur over long computations, requiring periodic orthonormalization to maintain valid rotations	No singularities; it can represent all orientations with no gimbal lock
Quaternions	Low interpretability; the components are not directly meaningful to humans (must be converted to axis-angle or Euler angles to interpret physically)	High efficiency; the rotation composition via quaternion multiplication is faster and more stable than using matrices; suitable for real-time systems; easy to normalize	High for stability; it requires special attention to maintain unit norm (which is essential to avoid numerical drift)	No singularities; global coverage without gimbal lock

2.3 Conclusions

This chapter introduced the foundational concepts of coordinate systems and attitude representation essential for IPIN applications. These fundamental topics provide the basis for the estimation methods and algorithms discussed in the following chapters.

References

[1] Groves, P., *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, Second Edition, Norwood, MA: Artech House, 2013.

[2] Kaplan, E. D., and C. Hegarty, (eds.), *Understanding GPS/GNSS: Principles and Applications*, Norwood, MA: Artech House, 2017.

[3] Mularie, W. M., *Department of Defense World Geodetic System 1984, Its Definition and Relationships with Local Geodetic Systems*, National Geospatial-Intelligence Agency, Tech. Rep 152, 2000.

[4] Diebel, J., "Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors." *Matrix*, Vol. 58, No. 15-16, 2006, pp. 1–35.

[5] Barfoot, T. D., *State Estimation for Robotics*, Cambridge, UK: Cambridge University Press, 2024.