

# 2

## Kinematics

The study of *dynamics* can be divided into two parts: *kinematics*, which treats only geometrical aspects of motion, and *kinetics*, which is the analysis of the forces causing the motion. In this chapter kinematics with application to local and terrestrial navigation is discussed. Kinetics is dealt with in Chapters 3–8.

The interested reader is advised to consult Britting (1971), Maybeck (1979), Savage (1990), Forssell (1991), Lin (1992), Hofmann-Wellenhof *et al.* (1994), Parkinson and Spilker (1995), Titterton and Weston (1997), and Farrell and Barth (1998) for a discussion of navigation kinematics and kinematics in general. The development of the kinematic equations of motion are also found in Kane *et al.* (1983) and Hughes (1986). Both of these references use spacecraft systems for illustration. An alternative derivation of the Euler angle representation in the context of ship steering is given by Abkowitz (1964). A more recent discussion of quaternions is found in Chou (1992). An analogy to robot manipulators is given by Craig (1989) or Sciavicco and Siciliano (1996), while a more detailed discussion of kinematics is found in Goldstein (1980) and Egeland and Gravdahl (2002).

### 6 DOF Marine Craft Equations of Motion

The overall goal of Chapters 2–8 is to show that the marine craft equations of motion can be written in a *vectorial setting* according to Fossen (1991):

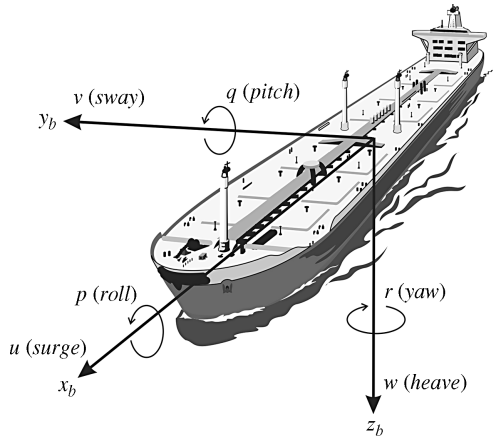
$$\dot{\eta} = J_{\Theta}(\eta)v \quad (2.1)$$

$$M\dot{v} + C(v)v + D(v)v + g(\eta) + g_0 = \tau + \tau_{\text{wind}} + \tau_{\text{wave}} \quad (2.2)$$

where the different matrices and vectors and their properties will be defined in the forthcoming sections. This model representation is used as a foundation for model-based control design and stability analysis in Part II.

### Motion Variables

For marine craft moving in six *degrees of freedom* (DOFs), six independent coordinates are necessary to determine the position and orientation. The first three coordinates, and their time derivatives, correspond to the position and translational motion along the  $x$ ,  $y$  and  $z$  axes, while the last three coordinates and their



**Figure 2.1** The 6 DOF velocities  $u$ ,  $v$ ,  $w$ ,  $p$ ,  $q$  and  $r$  in the body-fixed reference frame  $\{b\} = (x_b, y_b, z_b)$ .

time derivatives are used to describe orientation and rotational motion. For marine craft, the six different motion components are conveniently defined as *surge*, *sway*, *heave*, *roll*, *pitch* and *yaw* (see Figure 2.1 and Table 2.1).

2.1 Reference Frames

When analyzing the motion of marine craft in 6 DOF, it is convenient to define two Earth-centered coordinate frames as indicated in Figure 2.2. In addition several geographic reference frames are needed.

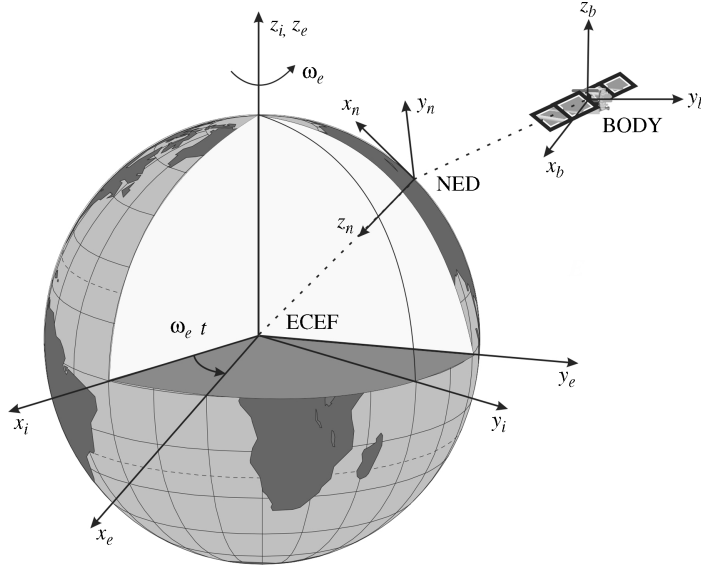
Earth-Centered Reference Frames

**ECI:** The Earth-centered inertial (ECI) frame  $\{i\} = (x_i, y_i, z_i)$  is an inertial frame for terrestrial navigation, that is a nonaccelerating reference frame in which Newton’s laws of motion apply. This includes inertial navigation systems. The origin of  $\{i\}$  is located at the center  $o_i$  of the Earth with axes as shown in Figure 2.2.

**ECEF:** The Earth-centered Earth-fixed (ECEF) reference frame  $\{e\} = (x_e, y_e, z_e)$  has its origin  $o_e$  fixed to the center of the Earth but the axes rotate relative to the inertial frame ECI, which is fixed in space. The angular rate of rotation is  $\omega_e = 7.2921 \times 10^{-5}$  rad/s. For marine craft moving at relatively low speed, the Earth rotation can be neglected and hence  $\{e\}$  can be considered to be inertial. Drifting ships, however, should not neglect the Earth rotation. The coordinate system  $\{e\}$  is usually used for global

**Table 2.1** The notation of SNAME (1950) for marine vessels

DOF		Forces and moments	Linear and angular velocities	Positions and Euler angles
1	motions in the $x$ direction (surge)	$X$	$u$	$x$
2	motions in the $y$ direction (sway)	$Y$	$v$	$y$
3	motions in the $z$ direction (heave)	$Z$	$w$	$z$
4	rotation about the $x$ axis (roll, heel)	$K$	$p$	$\phi$
5	rotation about the $y$ axis (pitch, trim)	$M$	$q$	$\theta$
6	rotation about the $z$ axis (yaw)	$N$	$r$	$\psi$



**Figure 2.2** The Earth-centered Earth-fixed (ECEF) frame  $x_e y_e z_e$  is rotating with angular rate  $\omega_e$  with respect to an Earth-centered inertial (ECI) frame  $x_i y_i z_i$  fixed in space.

guidance, navigation and control, for instance to describe the motion and location of ships in transit between different continents.

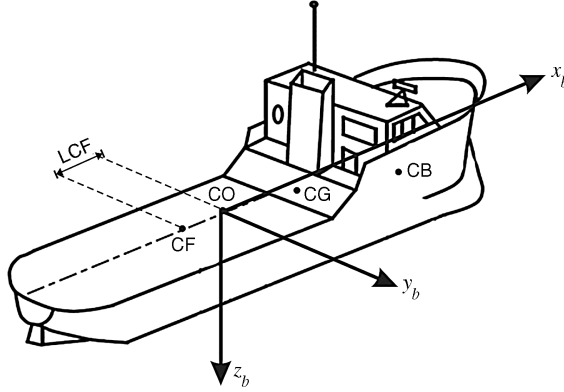
### Geographic Reference Frames

**NED:** The *North-East-Down* (NED) coordinate system  $\{n\} = (x_n, y_n, z_n)$  with origin  $o_n$  is defined relative to the Earth's reference ellipsoid (World Geodetic System, 1984). This is the coordinate system we refer to in our everyday life. It is usually defined as the tangent plane on the surface of the Earth moving with the craft, but with axes pointing in different directions than the body-fixed axes of the craft. For this system the  $x$  axis points towards true *North*, the  $y$  axis points towards *East* while the  $z$  axis points *downwards* normal to the Earth's surface. The location of  $\{n\}$  relative to  $\{e\}$  is determined by using two angles  $l$  and  $\mu$  denoting the *longitude* and *latitude*, respectively.

For marine craft operating in a local area, approximately constant longitude and latitude, an Earth-fixed tangent plane on the surface is used for navigation. This is usually referred to as flat Earth navigation and it will for simplicity be denoted by  $\{n\}$ . For flat Earth navigation one can assume that  $\{n\}$  is inertial such that Newton's laws still apply.

**BODY:** The body-fixed reference frame  $\{b\} = (x_b, y_b, z_b)$  with origin  $o_b$  is a moving coordinate frame that is fixed to the craft. The position and orientation of the craft are described relative to the inertial reference frame (approximated by  $\{e\}$  or  $\{n\}$  for marine craft) while the linear and angular velocities of the craft should be expressed in the body-fixed coordinate system. The origin  $o_b$  is usually chosen to coincide with a point midships in the water line. This point will be referred to as *CO* (see Figure 2.3). For marine craft, the body axes  $x_b$ ,  $y_b$  and  $z_b$  are chosen to coincide with the *principal axes of inertia*, and they are usually defined as (see Figure 2.3):

- $x_b$  - longitudinal axis (directed from aft to fore)
- $y_b$  - transversal axis (directed to starboard)
- $z_b$  - normal axis (directed from top to bottom)



**Figure 2.3** Body-fixed reference points.

In addition to the body-fixed coordinate system  $\{b\}$ , it is convenient to define other body-fixed coordinate systems when performing hydrodynamic computations. This includes a system using *flow axes* (see Section 2.4) and the *seakeeping reference frame*  $\{s\}$  (see Section 5.2).

### Body-Fixed Reference Points

The following reference points are defined with respect to CO:

**CG** - center of gravity

**CB** - center of buoyancy

**CF** - center of flotation (located a distance LCF from CO in the  $x$ -direction)

The center of flotation is the centroid of the water plane area  $A_{wp}$  in calm water. The craft will roll and pitch about this point. Consequently, this point can be used to compute the pitch and roll periods. The eigenvalues of the 6 DOF linear equations of motion are independent of the reference point but the decoupled equations will produce incorrect results if they are formulated in a point different from CF (see Section 4.3).

### 6 DOF Vectorial Notation

We will use the notation  $\vec{u}$  to refer to a *coordinate free vector*, that is a *directed line segment*. When a vector is described relative to a coordinate system  $\{n\}$ , the following notation will be used:

$$\vec{u} = u_1^n \vec{n}_1 + u_2^n \vec{n}_2 + u_3^n \vec{n}_3 \quad (2.3)$$

where  $\vec{n}_i$  ( $i = 1, 2, 3$ ) are the unit vectors that define  $\{n\}$ ,  $u_i^n$  are the measures of  $\vec{u}$  along  $\vec{n}_i$  and  $u_i^n \vec{n}_i$  are the components of  $\vec{u}$  in  $\{n\}$ . We will also use the *coordinate form*  $\mathbf{u}^n$  of  $\vec{u}$  in  $\{n\}$  which is represented by a *column vector* in  $\mathbb{R}^3$ :

$$\mathbf{u}^n = [u_1^n, u_2^n, u_3^n]^T \quad (2.4)$$

For marine craft the following notation will be adopted for vectors in the coordinate systems  $\{b\}$ ,  $\{e\}$  and  $\{n\}$ :

$\mathbf{v}_{b/n}^e$  = linear velocity of the point  $o_b$  with respect to  $\{n\}$  expressed in  $\{e\}$

$\boldsymbol{\omega}_{n/e}^b$  = angular velocity of  $\{n\}$  with respect to  $\{e\}$  expressed in  $\{b\}$

$\mathbf{f}_b^n$  = force with line of action through the point  $o_b$  expressed in  $\{n\}$

$\mathbf{m}_b^n$  = moment about the point  $o_b$  expressed in  $\{n\}$

$\boldsymbol{\Theta}_{nb}$  = Euler angles between  $\{n\}$  and  $\{b\}$

The different quantities in Table 2.1, as defined by SNAME (1950), can now be conveniently expressed in a vectorial setting according to:

ECEF position	$\mathbf{p}_{b/e}^e = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \in \mathbb{R}^3$	Longitude and latitude	$\boldsymbol{\Theta}_{en} = \begin{bmatrix} l \\ \mu \end{bmatrix} \in \mathcal{S}^2$
NED position	$\mathbf{p}_{b/n}^n = \begin{bmatrix} N \\ E \\ D \end{bmatrix} \in \mathbb{R}^3$	Attitude (Euler angles)	$\boldsymbol{\Theta}_{nb} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \in \mathcal{S}^3$
Body-fixed linear velocity	$\mathbf{v}_{b/n}^b = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \in \mathbb{R}^3$	Body-fixed angular velocity	$\boldsymbol{\omega}_{b/n}^b = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \in \mathbb{R}^3$
Body-fixed force	$\mathbf{f}_b^b = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \in \mathbb{R}^3$	Body-fixed moment	$\mathbf{m}_b^b = \begin{bmatrix} K \\ M \\ N \end{bmatrix} \in \mathbb{R}^3$

where  $\mathbb{R}^3$  is the *Euclidean space* of dimension three and  $\mathcal{S}^2$  denotes a *torus* of dimension two (shape of a donut), implying that there are two angles defined on the interval  $[0, 2\pi]$ . In the three-dimensional (3-D) case the set  $\mathcal{S}^3$  is a sphere. Hence, the general motion of a marine craft in 6 DOF with  $o_b$  as coordinate origin is described by the following vectors:

$$\boldsymbol{\eta} = \begin{bmatrix} \mathbf{p}_{b/n}^n \text{ (or } \mathbf{p}_{b/e}^e) \\ \boldsymbol{\Theta}_{nb} \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} \mathbf{v}_{b/n}^b \\ \boldsymbol{\omega}_{b/n}^b \end{bmatrix}, \quad \boldsymbol{\tau} = \begin{bmatrix} \mathbf{f}_b^b \\ \mathbf{m}_b^b \end{bmatrix} \quad (2.5)$$

where  $\boldsymbol{\eta} \in \mathbb{R}^3 \times \mathcal{S}^3$  denotes the position and orientation vector where the position vector  $\mathbf{p}_{b/n}^n \in \mathbb{R}^3$  is the distance from NED to BODY expressed in NED coordinates,  $\boldsymbol{\Theta}_{nb} \in \mathcal{S}^3$  is a vector of Euler angles,  $\mathbf{v} \in \mathbb{R}^6$  denotes the linear and angular velocity vectors that are decomposed in the body-fixed reference frame and  $\boldsymbol{\tau} \in \mathbb{R}^6$  is used to describe the forces and moments acting on the craft in the body-fixed frame.

In many applications, such as flat Earth navigation, the position vector  $\mathbf{p}_{b/n}^n \in \mathbb{R}^3$  from NED to BODY is expressed in NED coordinates. For global navigation it is convenient to express the position of the BODY origin  $o_b$  with respect to ECEF, that is  $\mathbf{p}_{b/e}^e \in \mathbb{R}^3$ . The orientation of the marine craft with respect to NED will be represented by means of the Euler angles  $\boldsymbol{\Theta}_{nb}$  or the quaternions  $\mathbf{q} \in \mathbb{R}^4$ . In the next sections, the kinematic equations relating the BODY, NED and ECEF reference frames will be presented.

## 2.2 Transformations between BODY and NED

The rotation matrix  $\mathbf{R}$  between two frames  $a$  and  $b$  is denoted as  $\mathbf{R}_b^a$ , and it is an element in  $SO(3)$ , that is the *special orthogonal group of order 3*:

$$SO(3) = \{ \mathbf{R} | \mathbf{R} \in \mathbb{R}^{3 \times 3}, \quad \mathbf{R} \text{ is orthogonal and } \det \mathbf{R} = 1 \} \quad (2.6)$$

The group  $SO(3)$  is a subset of all *orthogonal matrices of order 3*, that is  $SO(3) \subset O(3)$  where  $O(3)$  is defined as

$$O(3) := \{ \mathbf{R} | \mathbf{R} \in \mathbb{R}^{3 \times 3}, \quad \mathbf{R}\mathbf{R}^\top = \mathbf{R}^\top\mathbf{R} = \mathbf{I} \} \quad (2.7)$$

Rotation matrices are useful when deriving the kinematic equations of motion for a marine craft. As a consequence of (2.6) and (2.7), the following properties can be stated:

### Property 2.1 (Rotation Matrix)

A rotation matrix  $\mathbf{R} \in SO(3)$  satisfies

$$\mathbf{R}\mathbf{R}^\top = \mathbf{R}^\top\mathbf{R} = \mathbf{I}, \quad \det \mathbf{R} = 1$$

which implies that  $\mathbf{R}$  is orthogonal. Consequently, the inverse rotation matrix is given by  $\mathbf{R}^{-1} = \mathbf{R}^\top$ .

In this book, the following notation is adopted when transforming a vector from one coordinate frame to another:

$$\mathbf{v}^{\text{to}} = \mathbf{R}_{\text{from}}^{\text{to}} \mathbf{v}^{\text{from}} \quad (2.8)$$

Here  $\mathbf{v}^{\text{from}} \in \mathbb{R}^3$  denotes a velocity vector that can be transformed to a new reference frame by applying the rotation matrix  $\mathbf{R}_{\text{from}}^{\text{to}}$ . The result is the vector  $\mathbf{v}^{\text{to}} \in \mathbb{R}^3$ .

A frequently used rotation matrix in guidance, navigation and control is the rotation matrix  $\mathbf{R}_b^n$  between  $\{n\}$  and  $\{b\}$ . When deriving the expression for  $\mathbf{R}_b^n$  we will make use of the following matrix properties:

### Definition 2.1 (Skew-Symmetry of a Matrix)

A matrix  $\mathbf{S} \in SS(n)$ , that is the set of skew-symmetric matrices of order  $n$ , is said to be skew-symmetrical if

$$\mathbf{S} = -\mathbf{S}^\top$$

This implies that the off-diagonal elements of  $\mathbf{S}$  satisfy  $s_{ij} = -s_{ji}$  for  $i \neq j$  while the diagonal elements are zero.

### Definition 2.2 (Cross-Product Operator)

The vector cross-product  $\times$  is defined by

$$\lambda \times \mathbf{a} := \mathbf{S}(\lambda)\mathbf{a} \quad (2.9)$$

where  $\mathbf{S} \in SS(3)$  is defined as

$$\mathbf{S}(\lambda) = -\mathbf{S}^\top(\lambda) = \begin{bmatrix} 0 & -\lambda_3 & \lambda_2 \\ \lambda_3 & 0 & -\lambda_1 \\ -\lambda_2 & \lambda_1 & 0 \end{bmatrix}, \quad \lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} \quad (2.10)$$

**Matlab**

The cross-product operator is included in the MSS toolbox as `Smtx.m`. Hence, the cross-product  $\mathbf{b} = S(\boldsymbol{\lambda})\mathbf{a}$  can be computed as

```
S = Smtx(lambda)
b = S*a
```

**Definition 2.3 (Simple Rotation)**

The motion of a rigid body or reference frame  $\mathcal{B}$  relative to a rigid body or reference frame  $\mathcal{A}$  is called a simple rotation of  $\mathcal{B}$  in  $\mathcal{A}$  if there exists a line  $L$ , called an axis of rotation, whose orientation relative to both  $\mathcal{A}$  and  $\mathcal{B}$  remains unaltered throughout the motion.

Based on this definition, Euler stated the following theorem for rotation of two rigid bodies or reference frames (Euler, 1776).

**Theorem 2.1 (Euler's Theorem on Rotation)**

Every change in the relative orientation of two rigid bodies or reference frames  $\{\mathcal{A}\}$  and  $\{\mathcal{B}\}$  can be produced by means of a simple rotation of  $\{\mathcal{B}\}$  in  $\{\mathcal{A}\}$ .

Let  $\mathbf{v}_{b/n}^b$  be a vector fixed in BODY and  $\mathbf{v}_{b/n}^n$  be a vector fixed in NED. Hence, the vector  $\mathbf{v}_{b/n}^n$  can be expressed in terms of the vector  $\mathbf{v}_{b/n}^b$ , the unit vector  $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \lambda_3]^\top$ ,  $\|\boldsymbol{\lambda}\| = 1$ , parallel to the axis of rotation and  $\beta$  the angle NED is rotated. This rotation is described by (see Hughes, 1986, Kane *et al.*, 1983)

$$\mathbf{v}_{b/n}^n = \mathbf{R}_b^n \mathbf{v}_{b/n}^b, \quad \mathbf{R}_b^n := \mathbf{R}_{\boldsymbol{\lambda}, \beta} \quad (2.11)$$

Here,  $\mathbf{R}_{\boldsymbol{\lambda}, \beta}$  is the rotation matrix corresponding to a rotation  $\beta$  about the  $\boldsymbol{\lambda}$  axis:

$$\mathbf{R}_{\boldsymbol{\lambda}, \beta} = \mathbf{I}_{3 \times 3} + \sin(\beta)S(\boldsymbol{\lambda}) + [1 - \cos(\beta)]S^2(\boldsymbol{\lambda}) \quad (2.12)$$

where  $\mathbf{I}_{3 \times 3}$  is the identity matrix and  $S(\boldsymbol{\lambda})$  is the skew-symmetric matrix according to Definition 2.2. Consequently,  $S^2(\boldsymbol{\lambda}) = \boldsymbol{\lambda}\boldsymbol{\lambda}^\top - \mathbf{I}_{3 \times 3}$  since  $\boldsymbol{\lambda}$  is a unit vector.

Expanding (2.12) yields the following expressions for the matrix elements:

$$\begin{aligned} R_{11} &= [1 - \cos(\beta)]\lambda_1^2 + \cos(\beta) \\ R_{22} &= [1 - \cos(\beta)]\lambda_2^2 + \cos(\beta) \\ R_{33} &= [1 - \cos(\beta)]\lambda_3^2 + \cos(\beta) \\ R_{12} &= [1 - \cos(\beta)]\lambda_1\lambda_2 - \lambda_3\sin(\beta) \\ R_{21} &= [1 - \cos(\beta)]\lambda_2\lambda_1 + \lambda_3\sin(\beta) \\ R_{23} &= [1 - \cos(\beta)]\lambda_2\lambda_3 - \lambda_1\sin(\beta) \\ R_{32} &= [1 - \cos(\beta)]\lambda_3\lambda_2 + \lambda_1\sin(\beta) \\ R_{31} &= [1 - \cos(\beta)]\lambda_3\lambda_1 - \lambda_2\sin(\beta) \\ R_{13} &= [1 - \cos(\beta)]\lambda_1\lambda_3 + \lambda_2\sin(\beta) \end{aligned} \quad (2.13)$$

### 2.2.1 Euler Angle Transformation

The Euler angles, roll ( $\phi$ ), pitch ( $\theta$ ) and yaw ( $\psi$ ), can now be used to decompose the body-fixed velocity vector  $\mathbf{v}_{b/n}^b$  in the NED reference frame. Let  $\mathbf{R}_b^n(\boldsymbol{\Theta}_{nb}) : \mathcal{S}^3 \rightarrow SO(3)$  denote the Euler angle rotation matrix with argument  $\boldsymbol{\Theta}_{nb} = [\phi, \theta, \psi]^\top$ . Hence,

$$\mathbf{v}_{b/n}^n = \mathbf{R}_b^n(\boldsymbol{\Theta}_{nb}) \mathbf{v}_{b/n}^b \quad (2.14)$$

#### Principal Rotations

The principal rotation matrices (one axis rotations) can be obtained by setting  $\boldsymbol{\lambda} = [1, 0, 0]^\top$ ,  $\boldsymbol{\lambda} = [0, 1, 0]^\top$  and  $\boldsymbol{\lambda} = [0, 0, 1]^\top$  corresponding to the  $x$ ,  $y$  and  $z$  axes, and  $\beta = \phi$ ,  $\beta = \theta$  and  $\beta = \psi$ , respectively, in the formula for  $\mathbf{R}_{\lambda, \beta}$  given by (2.12). This yields

$$\mathbf{R}_{x, \phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix}, \quad \mathbf{R}_{y, \theta} = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix}, \quad \mathbf{R}_{z, \psi} = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.15)$$

where  $s \cdot = \sin(\cdot)$  and  $c \cdot = \cos(\cdot)$ .

#### Linear Velocity Transformation

It is customary to describe  $\mathbf{R}_b^n(\boldsymbol{\Theta}_{nb})$  by three *principal* rotations about the  $z$ ,  $y$  and  $x$  axes ( $zyx$  convention). Note that the order in which these rotations is carried out is not arbitrary. In guidance, navigation and control applications it is common to use the  $zyx$  convention *from*  $\{n\}$  *to*  $\{b\}$  specified in terms of the Euler angles  $\phi$ ,  $\theta$  and  $\psi$  for the rotations. This matrix is denoted  $\mathbf{R}_n^b(\boldsymbol{\Theta}_{nb}) = \mathbf{R}_b^n(\boldsymbol{\Theta}_{nb})^\top$ . The matrix transpose implies that the same result is obtained by transforming a vector *from*  $\{b\}$  *to*  $\{n\}$ , that is by reversing the order of the transformation. This rotation sequence is mathematically equivalent to

$$\mathbf{R}_b^n(\boldsymbol{\Theta}_{nb}) := \mathbf{R}_{z, \psi} \mathbf{R}_{y, \theta} \mathbf{R}_{x, \phi} \quad (2.16)$$

and the inverse transformation is then written ( $zyx$  convention)

$$\mathbf{R}_b^n(\boldsymbol{\Theta}_{nb})^{-1} = \mathbf{R}_n^b(\boldsymbol{\Theta}_{nb}) = \mathbf{R}_{x, \phi}^\top \mathbf{R}_{y, \theta}^\top \mathbf{R}_{z, \psi}^\top \quad (2.17)$$

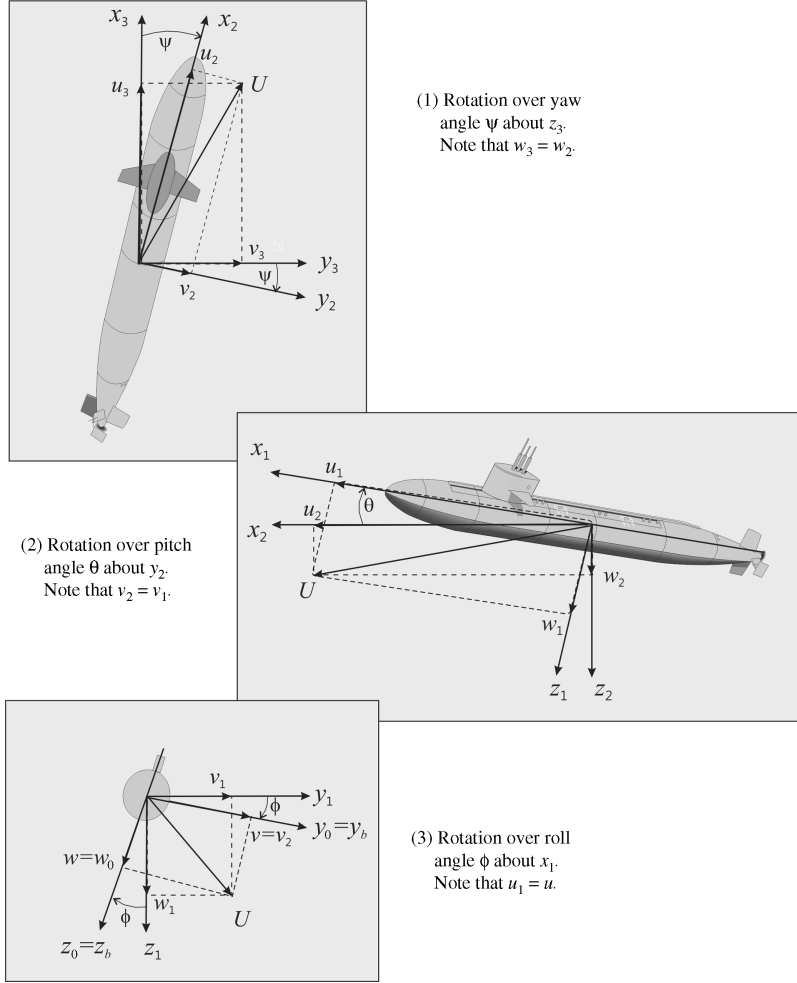
where we have used the result of Property 2.1. This can also be seen by studying Figure 2.4.

Let  $x_3 y_3 z_3$  be the coordinate system obtained by translating the NED coordinate system  $x_n y_n z_n$  parallel to itself until its origin coincides with the origin of the body-fixed coordinate system. The coordinate system  $x_3 y_3 z_3$  is rotated a *yaw* angle  $\psi$  about the  $z_3$  axis. This yields the coordinate system  $x_2 y_2 z_2$ . The coordinate system  $x_2 y_2 z_2$  is rotated a *pitch* angle  $\theta$  about the  $y_2$  axis. This yields the coordinate system  $x_1 y_1 z_1$ . Finally, the coordinate system  $x_1 y_1 z_1$  is rotated a *roll* angle  $\phi$  about the  $x_1$  axis. This yields the body-fixed coordinate system  $x_b y_b z_b$ .

Expanding (2.16) yields

$$\mathbf{R}_b^n(\boldsymbol{\Theta}_{nb}) = \begin{bmatrix} c\psi c\theta & -s\psi c\theta + c\psi s\theta s\phi & s\psi c\theta + c\psi c\theta s\phi \\ s\psi c\theta & c\psi c\theta + s\psi s\theta s\phi & -c\psi c\theta + s\psi c\theta s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \quad (2.18)$$





**Figure 2.4** Euler angle rotation sequence (zyx convention). The submarine is rotated from  $\{n\}$  to  $\{b\}$  by using three principal rotations.

#### Matlab

The rotation matrix  $R_b^n(\Theta_{nb})$  is implemented in the MSS toolbox as

$$R = Rzyx(\phi, \theta, \psi)$$

For small angles  $\delta\phi$ ,  $\delta\theta$  and  $\delta\psi$  the expression (2.18) simplifies to

$$R_b^n(\delta\Theta_{nb}) \approx I_{3 \times 3} + S(\delta\Theta_{nb}) = \begin{bmatrix} 1 & -\delta\psi & \delta\theta \\ \delta\psi & 1 & -\delta\phi \\ -\delta\theta & \delta\phi & 1 \end{bmatrix} \quad (2.19)$$

which is quite useful when applying linear theory.

The body-fixed velocity vector  $\mathbf{v}_{b/n}^b$  can be expressed in  $\{n\}$  as

$$\dot{\mathbf{p}}_{b/n}^n = \mathbf{R}_b^n(\boldsymbol{\Theta}_{nb}) \mathbf{v}_{b/n}^b \quad (2.20)$$

where  $\dot{\mathbf{p}}_{b/n}^n$  is the NED velocity vector. Expanding (2.20) yields

$$\begin{aligned} \dot{N} &= u \cos(\psi) \cos(\theta) + v [\cos(\psi) \sin(\theta) \sin(\phi) - \sin(\psi) \cos(\phi)] \\ &\quad + w [\sin(\psi) \sin(\phi) + \cos(\psi) \cos(\phi) \sin(\theta)] \end{aligned} \quad (2.21)$$

$$\begin{aligned} \dot{E} &= u \sin(\psi) \cos(\theta) + v [\cos(\psi) \cos(\phi) + \sin(\phi) \sin(\theta) \sin(\psi)] \\ &\quad + w [\sin(\theta) \sin(\psi) \cos(\phi) - \cos(\psi) \sin(\phi)] \end{aligned} \quad (2.22)$$

$$\dot{D} = -u \sin(\theta) + v \cos(\theta) \sin(\phi) + w \cos(\theta) \cos(\phi) \quad (2.23)$$

The inverse velocity transformation is obtained by Definition 2.1 as

$$\mathbf{v}_{b/n}^b = \mathbf{R}_b^n(\boldsymbol{\Theta}_{nb})^{-1} \dot{\mathbf{p}}_{b/n}^n = \mathbf{R}_b^n(\boldsymbol{\Theta}_{nb})^\top \dot{\mathbf{p}}_{b/n}^n \quad (2.24)$$

### Example 2.1 (Numerical Computation of Position Trajectory)

The flight path or position trajectory  $\mathbf{p}_{b/n}^n$  of the craft relative to the NED coordinate system is found by numerical integration of (2.20), for instance by using Euler integration:

$$\mathbf{p}_{b/n}^n(k+1) = \mathbf{p}_{b/n}^n(k) + h \mathbf{R}_b^n(\boldsymbol{\Theta}_{nb}(k)) \mathbf{v}_{b/n}^b(k) \quad (2.25)$$

where  $h > 0$  is the sampling time and  $k$  is the sample index.

### Angular Velocity Transformation

The body-fixed angular velocity vector  $\boldsymbol{\omega}_{b/n}^b = [p, q, r]^\top$  and the Euler rate vector  $\dot{\boldsymbol{\Theta}}_{nb} = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^\top$  are related through a transformation matrix  $\mathbf{T}_{\Theta}(\boldsymbol{\Theta}_{nb})$  according to

$$\dot{\boldsymbol{\Theta}}_{nb} = \mathbf{T}_{\Theta}(\boldsymbol{\Theta}_{nb}) \boldsymbol{\omega}_{b/n}^b \quad (2.26)$$

It should be noted that the angular body velocity vector  $\boldsymbol{\omega}_{b/n}^b = [p, q, r]^\top$  cannot be integrated directly to obtain actual angular coordinates. This is due to the fact that  $\int_0^t \boldsymbol{\omega}_{b/n}^b(\tau) d\tau$  does not have any immediate physical interpretation; however, the vector  $\boldsymbol{\Theta}_{nb} = [\phi, \theta, \psi]^\top$  does represent proper generalized coordinates. The transformation matrix  $\mathbf{T}_{\Theta}(\boldsymbol{\Theta}_{nb})$  can be derived in several ways, for instance:

$$\boldsymbol{\omega}_{b/n}^b = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_{x,\phi}^\top \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \mathbf{R}_{x,\phi}^\top \mathbf{R}_{y,\theta}^\top \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} := \mathbf{T}_{\Theta}^{-1}(\boldsymbol{\Theta}_{nb}) \dot{\boldsymbol{\Theta}}_{nb} \quad (2.27)$$

This relationship is verified by inspection of Figure 2.4. Expanding (2.27) yields

$$\mathbf{T}_{\Theta}^{-1}(\Theta_{nb}) = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & c\theta s\phi \\ 0 & -s\phi & c\theta c\phi \end{bmatrix} \implies \mathbf{T}_{\Theta}(\Theta_{nb}) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix} \quad (2.28)$$

where  $s \cdot = \sin(\cdot)$ ,  $c \cdot = \cos(\cdot)$  and  $t \cdot = \tan(\cdot)$ . Expanding (2.26) yields the Euler angle attitude equations in component form:

$$\dot{\phi} = p + q \sin(\phi) \tan(\theta) + r \cos(\phi) \tan(\theta) \quad (2.29)$$

$$\dot{\theta} = q \cos(\phi) - r \sin(\phi) \quad (2.30)$$

$$\dot{\psi} = q \frac{\sin(\phi)}{\cos(\theta)} + r \frac{\cos(\phi)}{\cos(\theta)}, \quad \theta \neq \pm 90^\circ \quad (2.31)$$

Notice that  $\mathbf{T}_{\Theta}(\Theta_{nb})$  is undefined for a pitch angle of  $\theta = \pm 90^\circ$  and that  $\mathbf{T}_{\Theta}(\Theta_{nb})$  does not satisfy Property 2.1. Consequently,  $\mathbf{T}_{\Theta}^{-1}(\Theta_{nb}) \neq \mathbf{T}_{\Theta}^{\top}(\Theta_{nb})$ . For surface vessels this is not a problem whereas both underwater vehicles and aircraft may operate close to this singularity. In this case, the kinematic equations can be described by two Euler angle representations with different singularities and the singular point can be avoided by switching between them. Another possibility is to use a quaternion representation; see Section 2.2.2.

For small angles  $\delta\phi$ ,  $\delta\theta$  and  $\delta\psi$  the transformation matrix  $\mathbf{T}_{\Theta}(\Theta_{nb})$  simplifies to

$$\mathbf{T}_{\Theta}(\delta\Theta_{nb}) \approx \begin{bmatrix} 1 & 0 & \delta\theta \\ 0 & 1 & -\delta\phi \\ 0 & \delta\phi & 1 \end{bmatrix} \quad (2.32)$$

The differential equation for the rotation matrix is given by Theorem 2.2.

**Theorem 2.2 (Rotation Matrix Differential Equation)**

*The differential equation for the rotation matrix between the BODY and NED reference frames is*

$$\dot{\mathbf{R}}_b^n = \mathbf{R}_b^n \mathbf{S}(\omega_{b/n}^b) \quad (2.33)$$

where

$$\mathbf{S}(\omega_{b/n}^b) = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \quad (2.34)$$

This can be written in component form as nine differential equations:

$$\begin{bmatrix} \dot{R}_{11} & \dot{R}_{12} & \dot{R}_{13} \\ \dot{R}_{21} & \dot{R}_{22} & \dot{R}_{23} \\ \dot{R}_{31} & \dot{R}_{32} & \dot{R}_{33} \end{bmatrix} = \begin{bmatrix} R_{12}r - R_{13}q & -R_{11}r + R_{13}p & R_{11}q - R_{12}p \\ R_{22}r - R_{23}q & -R_{21}r + R_{23}p & R_{21}q - R_{22}p \\ R_{32}r - R_{33}q & -R_{31}r + R_{33}p & R_{31}q - R_{32}p \end{bmatrix} \quad (2.35)$$

**Proof.** For a small time increment  $\Delta t$  the rotation matrix  $\mathbf{R}_b^n$  satisfies

$$\mathbf{R}_b^n(t + \Delta t) \approx \mathbf{R}_b^n(t) \mathbf{R}_b^n(\Delta t) \quad (2.36)$$

since  $\sin(\Delta t) \approx \Delta t$  and  $\cos(\Delta t) \approx 1$ . Assume that after time  $t + \Delta t$  there has been an infinitesimal increment  $\Delta\beta$  in the rotation angle. From (2.12) we have

$$\begin{aligned} \mathbf{R}_b^n(\Delta t) &= \mathbf{I}_{3 \times 3} + \sin(\Delta\beta) \mathbf{S}(\lambda) + [1 - \cos(\Delta\beta)] \mathbf{S}^2(\lambda) \\ &\approx \mathbf{I}_{3 \times 3} + \Delta\beta \mathbf{S}(\lambda) \end{aligned} \quad (2.37)$$

From (2.36), it follows that

$$\mathbf{R}_b^n(t + \Delta t) = \mathbf{R}_b^n(t) [\mathbf{I}_{3 \times 3} + \Delta\beta \mathbf{S}(\lambda)] \quad (2.38)$$

Defining the vector  $\Delta\beta^b := \Delta\beta\lambda$ , the time derivative of  $\mathbf{R}_b^n$  is found as

$$\begin{aligned} \dot{\mathbf{R}}_b^n(t) &= \lim_{\Delta t \rightarrow 0} \frac{\mathbf{R}_b^n(t + \Delta t) - \mathbf{R}_b^n(t)}{\Delta t} \\ &= \lim_{\Delta t \rightarrow 0} \frac{\mathbf{R}_b^n(t) \Delta\beta \mathbf{S}(\lambda)}{\Delta t} \\ &= \lim_{\Delta t \rightarrow 0} \frac{\mathbf{R}_b^n(t) \mathbf{S}(\Delta\beta^b)}{\Delta t} \\ &= \mathbf{R}_b^n(t) \mathbf{S}(\omega_{b/n}^b) \end{aligned} \quad (2.39)$$

where  $\omega_{b/n}^b = \lim_{\Delta t \rightarrow 0} (\Delta\beta^b / \Delta t)$ .

## 6 DOF Kinematic Equations

Summarizing the results from this section, the 6 DOF kinematic equations can be expressed in vector form as

$$\begin{aligned} \dot{\eta} &= \mathbf{J}_\Theta(\eta) \mathbf{v} \\ \Updownarrow \\ \begin{bmatrix} \dot{\mathbf{p}}_{b/n}^n \\ \dot{\Theta}_{nb} \end{bmatrix} &= \begin{bmatrix} \mathbf{R}_b^n(\Theta_{nb}) & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{T}_\Theta(\Theta_{nb}) \end{bmatrix} \begin{bmatrix} \mathbf{v}_{b/n}^b \\ \omega_{b/n}^b \end{bmatrix} \end{aligned} \quad (2.40)$$

where  $\eta \in \mathbb{R}^3 \times \mathcal{S}^3$  and  $\mathbf{v} \in \mathbb{R}^6$ .

### Matlab

The transformation matrix  $\mathbf{J}_\Theta(\eta)$  and its diagonal elements  $J_{11}(\eta) = \mathbf{R}_b^n(\Theta_{nb})$  and  $J_{22}(\eta) = \mathbf{T}_\Theta(\Theta_{nb})$  can be computed by using the MSS toolbox command:

```
[J, J11, J22] = eulerang(phi, theta, psi)
```

The differential equations are then found by

$$\begin{aligned} \mathbf{p\_dot} &= \mathbf{J1} * \mathbf{v} \\ \mathbf{theta\_dot} &= \mathbf{J2} * \mathbf{w\_nb} \end{aligned}$$

Alternatively, (2.40) can be written in component form as

$$\begin{aligned} \dot{N} &= u \cos(\psi) \cos(\theta) + v[\cos(\psi) \sin(\theta) \sin(\phi) - \sin(\psi) \cos(\phi)] \\ &\quad + w[\sin(\psi) \sin(\phi) + \cos(\psi) \cos(\phi) \sin(\theta)] \end{aligned} \quad (2.41)$$

$$\begin{aligned} \dot{E} &= u \sin(\psi) \cos(\theta) + v[\cos(\psi) \cos(\phi) + \sin(\phi) \sin(\theta) \sin(\psi)] \\ &\quad + w[\sin(\theta) \sin(\psi) \cos(\phi) - \cos(\psi) \sin(\phi)] \end{aligned} \quad (2.42)$$

$$\dot{D} = -u \sin(\theta) + v \cos(\theta) \sin(\phi) + w \cos(\theta) \cos(\phi) \quad (2.43)$$

$$\dot{\phi} = p + q \sin(\phi) \tan(\theta) + r \cos(\phi) \tan(\theta) \quad (2.44)$$

$$\dot{\theta} = q \cos(\phi) - r \sin(\phi) \quad (2.45)$$

$$\dot{\psi} = q \frac{\sin(\phi)}{\cos(\theta)} + r \frac{\cos(\phi)}{\cos(\theta)}, \quad \theta \neq \pm 90^\circ \quad (2.46)$$

### 3 DOF Model for Surface Vessels

A frequently used simplification of (2.40) is the 3 DOF (surge, sway and yaw) representation for marine craft. This is based on the assumption that  $\phi$  and  $\theta$  are small, which is a good approximation for most conventional ships, underwater vehicles and rigs. Hence,  $\mathbf{R}_b^a(\boldsymbol{\Theta}_{nb}) = \mathbf{R}_{z,\psi} \mathbf{R}_{y,\theta} \mathbf{R}_{x,\phi} \approx \mathbf{R}_{z,\psi}$  and  $\mathbf{T}_\Theta(\boldsymbol{\Theta}_{nb}) \approx \mathbf{I}_{3 \times 3}$ . Neglecting the elements corresponding to heave, roll and pitch finally yields:

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi) \mathbf{v} \quad (2.47)$$

where  $\mathbf{R}(\psi) := \mathbf{R}_{z,\psi}$  with  $\mathbf{v} = [u, v, r]^\top$  and  $\boldsymbol{\eta} = [N, E, \psi]^\top$ .

#### 2.2.2 Unit Quaternions

An alternative to the Euler angle representation is a four-parameter method based on *unit quaternions* or *Euler parameters*. The main motivation for using four parameters is to avoid the representation singularity of the Euler angles.

A quaternion  $\mathbf{q}$  is defined as a complex number (Chou, 1992) with one real part  $\eta$  and three imaginary parts given by the vector

$$\boldsymbol{\varepsilon} = [\varepsilon_1, \varepsilon_2, \varepsilon_3]^\top \quad (2.48)$$

A unit quaternion satisfies  $\mathbf{q}^\top \mathbf{q} = 1$ . The set  $\mathcal{Q}$  of unit quaternions is therefore defined as

$$\mathcal{Q} := \{\mathbf{q} | \mathbf{q}^\top \mathbf{q} = 1, \mathbf{q} = [\eta, \boldsymbol{\varepsilon}^\top]^\top, \quad \boldsymbol{\varepsilon} \in \mathbb{R}^3 \text{ and } \eta \in \mathbb{R}\} \quad (2.49)$$

The motion of the body-fixed reference frame relative to the inertial frame will now be expressed in terms of unit quaternions.

### Unit Quaternions

From (2.12) it is seen that

$$\mathbf{R}_{\beta,\lambda} = \mathbf{I}_{3 \times 3} + \sin(\beta)\mathbf{S}(\lambda) + [1 - \cos(\beta)]\mathbf{S}^2(\lambda) \quad (2.50)$$

The real and imaginary parts of the unit quaternions are defined as (Chou, 1992)

$$\eta := \cos\left(\frac{\beta}{2}\right) \quad (2.51)$$

$$\boldsymbol{\varepsilon} = [\varepsilon_1, \varepsilon_2, \varepsilon_3]^\top := \lambda \sin\left(\frac{\beta}{2}\right) \quad (2.52)$$

where  $\lambda = [\lambda_1, \lambda_2, \lambda_3]^\top$  is a unit vector satisfying

$$\lambda = \pm \frac{\boldsymbol{\varepsilon}}{\sqrt{\boldsymbol{\varepsilon}^\top \boldsymbol{\varepsilon}}} \quad \text{if } \sqrt{\boldsymbol{\varepsilon}^\top \boldsymbol{\varepsilon}} \neq 0 \quad (2.53)$$

Consequently, the unit quaternions can be expressed in the form

$$\mathbf{q} = \begin{bmatrix} \eta \\ \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\beta}{2}\right) \\ \lambda \sin\left(\frac{\beta}{2}\right) \end{bmatrix} \in \mathcal{Q}, \quad 0 \leq \beta \leq 2\pi \quad (2.54)$$

This parametrization implies that the unit quaternions satisfy the constraint  $\mathbf{q}^\top \mathbf{q} = 1$ , that is

$$\eta^2 + \varepsilon_1^2 + \varepsilon_2^2 + \varepsilon_3^2 = 1 \quad (2.55)$$

From (2.50) with (2.51) and (2.52), the following coordinate transformation matrix for the unit quaternions is obtained:

$$\mathbf{R}_b^n(\mathbf{q}) := \mathbf{R}_{\eta,\boldsymbol{\varepsilon}} = \mathbf{I}_{3 \times 3} + 2\eta\mathbf{S}(\boldsymbol{\varepsilon}) + 2\mathbf{S}^2(\boldsymbol{\varepsilon}) \quad (2.56)$$

### Linear Velocity Transformation

The transformation relating the linear velocity vector in an inertial reference frame to a velocity in the body-fixed reference frame can now be expressed as

$$\dot{\mathbf{p}}_{b/n}^n = \mathbf{R}_b^n(\mathbf{q}) \mathbf{v}_{b/n}^b \quad (2.57)$$

where

$$\mathbf{R}_b^n(\mathbf{q}) = \begin{bmatrix} 1 - 2(\varepsilon_2^2 + \varepsilon_3^2) & 2(\varepsilon_1\varepsilon_2 - \varepsilon_3\eta) & 2(\varepsilon_1\varepsilon_3 + \varepsilon_2\eta) \\ 2(\varepsilon_1\varepsilon_2 + \varepsilon_3\eta) & 1 - 2(\varepsilon_1^2 + \varepsilon_3^2) & 2(\varepsilon_2\varepsilon_3 - \varepsilon_1\eta) \\ 2(\varepsilon_1\varepsilon_3 - \varepsilon_2\eta) & 2(\varepsilon_2\varepsilon_3 + \varepsilon_1\eta) & 1 - 2(\varepsilon_1^2 + \varepsilon_2^2) \end{bmatrix} \quad (2.58)$$

Expanding (2.57) yields

$$\dot{N} = u(1 - 2\varepsilon_2^2 - 2\varepsilon_3^2) + 2v(\varepsilon_1\varepsilon_2 - \varepsilon_3\eta) + 2w(\varepsilon_1\varepsilon_3 + \varepsilon_2\eta) \quad (2.59)$$

$$\dot{E} = 2u(\varepsilon_1\varepsilon_2 + \varepsilon_3\eta) + v(1 - 2\varepsilon_1^2 - 2\varepsilon_3^2) + 2w(\varepsilon_2\varepsilon_3 - \varepsilon_1\eta) \quad (2.60)$$

$$\dot{D} = 2u(\varepsilon_1\varepsilon_3 - \varepsilon_2\eta) + 2v(\varepsilon_2\varepsilon_3 + \varepsilon_1\eta) + w(1 - 2\varepsilon_1^2 - 2\varepsilon_2^2) \quad (2.61)$$

As for the Euler angle representation, Property 2.1 implies that the inverse transformation matrix satisfies  $\mathbf{R}_b^n(\mathbf{q})^{-1} = \mathbf{R}_b^n(\mathbf{q})^\top$ .

#### Matlab

The quaternion rotation matrix is easily computed by using the MSS toolbox commands

```
q = [eta, eps1, eps2, eps3]
R = Rquat(q)
```

Notice that  $\mathbf{q}^\top \mathbf{q} = 1$  must be true for `Rquat.m` to return a solution. One way to ensure this is to use the transformation

```
q = euler2q(phi, theta, psi)
```

transforming the three Euler angles  $\phi$ ,  $\theta$  and  $\psi$  to the unit quaternion vector  $\mathbf{q}$ ; see Section 2.2.3 for details.

#### Angular Velocity Transformation

The angular velocity transformation can be derived by substituting the expressions for  $R_{ij}$  from (2.58) into the differential equation  $\dot{\mathbf{R}}_b^n = \mathbf{R}_b^n \mathbf{S}(\boldsymbol{\omega}_{b/n}^b)$ ; see Theorem 2.2. Some calculations yield

$$\dot{\mathbf{q}} = \mathbf{T}_q(\mathbf{q}) \boldsymbol{\omega}_{b/n}^b \quad (2.62)$$

where

$$\mathbf{T}_q(\mathbf{q}) = \frac{1}{2} \begin{bmatrix} -\varepsilon_1 & -\varepsilon_2 & -\varepsilon_3 \\ \eta & -\varepsilon_3 & \varepsilon_2 \\ \varepsilon_3 & \eta & -\varepsilon_1 \\ -\varepsilon_2 & \varepsilon_1 & \eta \end{bmatrix}, \quad \mathbf{T}_q^\top(\mathbf{q}) \mathbf{T}_q(\mathbf{q}) = \frac{1}{4} \mathbf{I}_{3 \times 3} \quad (2.63)$$

Hence,

$$\dot{\eta} = -\frac{1}{2}(\varepsilon_1 p + \varepsilon_2 q + \varepsilon_3 r) \quad (2.64)$$

$$\dot{\varepsilon}_1 = \frac{1}{2}(\eta p - \varepsilon_3 q + \varepsilon_2 r) \quad (2.65)$$

$$\dot{\varepsilon}_2 = \frac{1}{2}(\varepsilon_3 p + \eta q - \varepsilon_1 r) \quad (2.66)$$

$$\dot{\varepsilon}_3 = \frac{1}{2}(-\varepsilon_2 p + \varepsilon_1 q + \eta r) \quad (2.67)$$

An alternative formulation is the vector representation (Kane *et al.*, 1983)

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{\eta} \\ \dot{\boldsymbol{\varepsilon}} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -\boldsymbol{\varepsilon}^\top \\ \eta \mathbf{I}_{3 \times 3} + \mathbf{S}(\boldsymbol{\varepsilon}) \end{bmatrix} \boldsymbol{\omega}_{b/n}^b \quad (2.68)$$

## 6 DOF Kinematic Equations

Consequently, the 6 DOF kinematic equations of motion can be expressed by seven differential equations for  $\boldsymbol{\eta} = [N, E, D, \eta, \varepsilon_1, \varepsilon_2, \varepsilon_3]^\top$  (recall that only six differential equations are needed when using the Euler angle representation):

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= \mathbf{J}_q(\boldsymbol{\eta}) \mathbf{v} \\ &\Downarrow \\ \begin{bmatrix} \dot{\mathbf{p}}_{b/n}^n \\ \dot{\mathbf{q}} \end{bmatrix} &= \begin{bmatrix} \mathbf{R}_b^n(\mathbf{q}) & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{4 \times 3} & \mathbf{T}_q(\mathbf{q}) \end{bmatrix} \begin{bmatrix} \mathbf{v}_{b/n}^b \\ \boldsymbol{\omega}_{b/n}^b \end{bmatrix} \end{aligned} \quad (2.69)$$

where  $\boldsymbol{\eta} \in \mathbb{R}^7$  and  $\mathbf{v} \in \mathbb{R}^6$ , and  $\mathbf{J}_q(\boldsymbol{\eta}) \in \mathbb{R}^{7 \times 6}$  is a nonquadratic transformation matrix. Equation (2.69) in component form is given by (2.59)–(2.61) and (2.64)–(2.67).

### Matlab

The transformation matrix  $\mathbf{J}_q(\boldsymbol{\eta})$  and its elements  $\mathbf{J}_{11} = \mathbf{R}_b^n(\mathbf{q})$  and  $\mathbf{J}_{22} = \mathbf{T}_q(\mathbf{q})$  can be computed directly in the MSS toolbox by using the following commands:

```
q = [eta, eps1, eps2, eps3]';
[J, J11, J22] = quatern(q)
```

The corresponding differential equations are

```
p_dot = J11*v
q_dot = J22*w_bn
```



### Implementation Considerations: Unit Quaternion Normalization

When integrating (2.62), a normalization procedure is necessary to ensure that the constraint

$$\mathbf{q}^\top \mathbf{q} = \varepsilon_1^2 + \varepsilon_2^2 + \varepsilon_3^2 + \eta^2 = 1 \quad (2.70)$$

is satisfied in the presence of measurement noise and numerical round-off errors. For this purpose, the following discrete-time algorithm can be applied.

#### Algorithm 2.1 (Discrete-Time Normalization of the Unit Quaternions)

1.  $k = 0$ . Compute initial values of  $\mathbf{q}(k = 0)$ .
2. For simplicity, Euler integration implies that

$$\mathbf{q}(k + 1) = \mathbf{q}(k) + h \mathbf{T}_q(\mathbf{q}(k)) \boldsymbol{\omega}_{b/n}^b(k) \quad (2.71)$$

where  $h$  is the sampling time.

3. Normalization:

$$\mathbf{q}(k + 1) = \frac{\mathbf{q}(k + 1)}{\|\mathbf{q}(k + 1)\|} = \frac{\mathbf{q}(k + 1)}{\sqrt{\mathbf{q}^\top(k + 1) \mathbf{q}(k + 1)}}$$

4. Let  $k = k + 1$  and return to Step 2.

A continuous time algorithm for unit quaternion normalization can be implemented by noting that

$$\frac{d}{dt} (\mathbf{q}^\top \mathbf{q}) = 2 \mathbf{q}^\top \mathbf{T}_q(\mathbf{q}) \boldsymbol{\omega}_{b/n}^b = 0 \quad (2.72)$$

This shows that if  $\mathbf{q}$  is initialized as a unit vector, then it will remain a unit vector. Since integration of the quaternion vector  $\mathbf{q}$  from the differential equation (2.62) will introduce numerical errors that will cause the length of  $\mathbf{q}$  to deviate from unity, a nonlinear feedback or normalization term is suggested. In Simulink this is done by replacing the kinematic differential equation (2.62) with

$$\dot{\mathbf{q}} = \mathbf{T}_q(\mathbf{q}) \boldsymbol{\omega}_{b/n}^b + \frac{\gamma}{2} (1 - \mathbf{q}^\top \mathbf{q}) \mathbf{q} \quad (2.73)$$

where  $\gamma \geq 0$  (typically 100) is a design parameter reflecting the convergence rate of the normalization. This results in

$$\frac{d}{dt} (\mathbf{q}^\top \mathbf{q}) = \underbrace{2 \mathbf{q}^\top \mathbf{T}_q(\mathbf{q}) \boldsymbol{\omega}_{b/n}^b}_{0 \text{ since } \mathbf{q}(0) \text{ is a unit vector}} + \gamma (1 - \mathbf{q}^\top \mathbf{q}) \mathbf{q}^\top \mathbf{q} = \gamma (1 - \mathbf{q}^\top \mathbf{q}) \mathbf{q}^\top \mathbf{q} \quad (2.74)$$

Observe that  $\mathbf{q}^\top \mathbf{q}$  will decrease if  $\mathbf{q}^\top \mathbf{q} > 1$  while it increases for  $\mathbf{q}^\top \mathbf{q} < 1$ . When  $\mathbf{q}^\top \mathbf{q} = 1$  the usual kinematic differential equations are recovered. A change of coordinates  $x = 1 - \mathbf{q}^\top \mathbf{q}$ ,  $\dot{x} = -d/dt(\mathbf{q}^\top \mathbf{q})$  yields

$$\dot{x} = -\gamma x(1 - x) \quad (2.75)$$

Linearization about  $x = 0$  gives  $\dot{x} = -\gamma x$ . Consequently, the normalization algorithm converges with a time constant  $T = \gamma^{-1}$ .

### 2.2.3 Quaternions from Euler Angles

If the Euler angles  $\Theta_{nb} = [\phi, \theta, \psi]^\top$  are known and therefore the expression for the rotation matrix  $\mathbf{R}_b^n(\Theta_{nb}) = \{R_{ij}\}$ , a singularity free extraction procedure can be used to compute the corresponding unit quaternions (Shepperd, 1978).

#### Algorithm 2.2 (Quaternions From Euler Angles)

1. Given the Euler angles  $\phi, \theta$  and  $\psi$ . Let the transformation matrix  $\mathbf{R}_b^n$  according to (2.18) be written

$$\mathbf{R}_b^n(\Theta_{nb}) := \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}$$

2. The trace of  $\mathbf{R}_b^n(\Theta_{nb})$  is computed as

$$R_{44} = \text{tr}(\mathbf{R}_b^n) = R_{11} + R_{22} + R_{33}$$

3. Let  $1 \leq i \leq 4$  be the index corresponding to

$$R_{ii} = \max(R_{11}, R_{22}, R_{33}, R_{44})$$

4. Compute  $p_i$  corresponding to Index  $i$  of Step 3 according to

$$p_i = \left| \sqrt{1 + 2R_{ii} - R_{44}} \right|$$

where the sign ascribed to  $p_i$  can be chosen to be either positive or negative.

5. Compute the other three  $p_i$ -values from

$$\begin{aligned} p_4 p_1 &= R_{32} - R_{23} & p_2 p_3 &= R_{32} + R_{23} \\ p_4 p_2 &= R_{13} - R_{31} & p_3 p_1 &= R_{13} + R_{31} \\ p_4 p_3 &= R_{21} - R_{12} & p_1 p_2 &= R_{21} + R_{12} \end{aligned}$$

by simply dividing the three equations containing the component  $p_i$  with the known value of  $p_i$  (from Step 4) on both sides.

6. Compute the Euler parameters  $\mathbf{q} = [\eta, \varepsilon_1, \varepsilon_2, \varepsilon_3]^\top$  according to

$$\begin{aligned} \varepsilon_j &= p_j/2 & (j = 1, 2, 3) \\ \eta &= p_4/2 \end{aligned}$$

#### Matlab

Algorithm 2.2 is implemented in the MSS toolbox as `euler2q.m`. This algorithm can also be used to compute the initial values of the Euler parameters corresponding to Step 1 of Algorithm 2.1.

**Example 2.2 (Euler Angles to Unit Quaternions)**

Consider a marine craft with orientation  $\phi = 10.0^\circ$ ,  $\theta = -20.0^\circ$  and  $\psi = 30.0^\circ$ . The unit quaternions are computed in Matlab by using the commands

```
phi=10*(pi/180), theta=-20*(pi/180), psi=30*(pi/180)
q = euler2q(phi,theta,psi)
q = [0.9437, 0.1277, -0.1449, 0.2685]
% normalization test
norm(q) =
1.0000
```

**2.2.4 Euler Angles from Quaternions**

The relationship between the Euler angles  $\phi, \theta$  and  $\psi$  (zyx convention) and the unit quaternions  $q_i$  ( $i = 1, \dots, 4$ ) can be established by requiring that the rotation matrices of the two kinematic representations are equal:

$$\mathbf{R}_b^n(\Theta_{nb}) := \mathbf{R}_b^n(q) \quad (2.76)$$

Let the elements of  $\mathbf{R}_b^n(q)$  be denoted by  $R_{ij}$  where the superscripts  $i$  and  $j$  denote the  $i$ th row and  $j$ th column. Writing expression (2.76) in component form yields a system of nine equations with three unknowns ( $\phi, \theta$  and  $\psi$ ) given by

$$\begin{bmatrix} c\psi c\theta & -s\psi c\theta + c\psi s\theta s\phi & s\psi s\theta + c\psi c\theta s\phi \\ s\psi c\theta & c\psi c\theta + s\psi s\theta s\phi & -c\psi s\theta + s\psi c\theta s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \quad (2.77)$$

**Algorithm 2.3 (Euler Angles from Quaternions)**

One solution to (2.77) is

$$\phi = \text{atan2}(R_{32}, R_{33}) \quad (2.78)$$

$$\theta = -\sin^{-1}(R_{31}) = -\tan^{-1}\left(\frac{R_{31}}{\sqrt{1 - R_{31}^2}}\right); \quad \theta \neq \pm 90^\circ \quad (2.79)$$

$$\psi = \text{atan2}(R_{21}, R_{11}) \quad (2.80)$$

In Algorithm 2.3  $\text{atan2}(y, x)$  is the four-quadrant *arctangent* of the real parts of the elements of  $x$  and  $y$  satisfying

$$-\pi \leq \text{atan2}(y, x) \leq \pi \quad (2.81)$$

Precautions must be taken against computational errors in the vicinity of  $\theta = \pm 90^\circ$ .

**Matlab**

The MSS toolbox script

```
[phi,theta,psi] = q2euler(q)
```

is based on Algorithm 2.3. A singularity test is included in order to avoid  $\theta = \pm 90^\circ$ .

**Example 2.3 (Unit Quaternions to Euler Angles)**

Consider the marine vessel in Example 2.2 where the Euler angles were converted into unit quaternions. The inverse transformation `q2euler.m` results in

```
q = [0.9437, 0.1277, -0.1449, 0.2685]';
[phi,theta,psi] = q2euler(q/norm(q))
phi    = 0.1746
theta  = -0.3491
psi    = 0.5235
```

corresponding to  $\phi = 10.0^\circ$ ,  $\theta = -20.0^\circ$  and  $\psi = 30.0^\circ$ .

## 2.3 Transformations between ECEF and NED

Wide area or terrestrial guidance and navigation implies that the position should be related to the Earth center instead of a local frame on the Earth's surface. This can be done by using the results from the previous sections.

### 2.3.1 Longitude and Latitude Transformations

The transformation between the ECEF and NED velocity vectors is

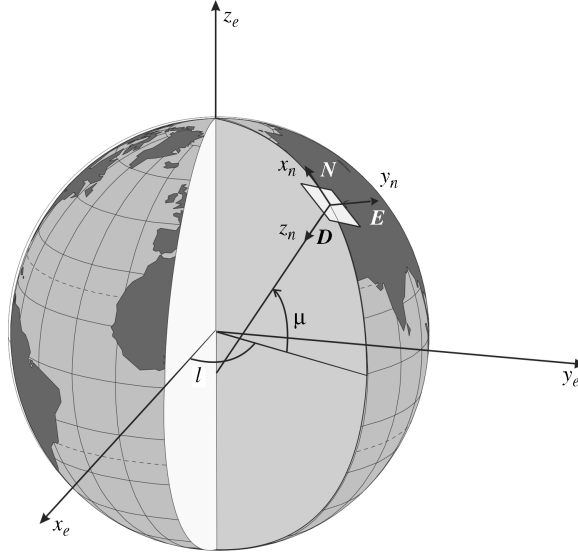
$$\dot{\mathbf{p}}_{b/e}^e = \mathbf{R}_n^e(\boldsymbol{\Theta}_{en}) \dot{\mathbf{p}}_{b/e}^n = \mathbf{R}_n^e(\boldsymbol{\Theta}_{en}) \mathbf{R}_b^n(\boldsymbol{\Theta}_{nb}) \dot{\mathbf{v}}_{b/e}^b \quad (2.82)$$

where  $\boldsymbol{\Theta}_{en} = [l, \mu]^\top \in \mathcal{S}^2$  is a vector formed by longitude  $l$  and latitude  $\mu$  (see Figure 2.5) and  $\mathbf{R}_n^e(\boldsymbol{\Theta}_{en}) : \mathcal{S}^2 \rightarrow SO(3)$  is a rotation matrix between ECEF and NED. This is found by performing two principal rotations: (1) a rotation  $l$  about the  $z$  axis and (2) a rotation  $(-\mu - \pi/2)$  about the  $y$  axis. This gives

$$\begin{aligned} \mathbf{R}_n^e(\boldsymbol{\Theta}_{en}) &= \mathbf{R}_{z,l} \mathbf{R}_{y,-\mu-\frac{\pi}{2}} \\ &= \begin{bmatrix} \cos(l) & -\sin(l) & 0 \\ \sin(l) & \cos(l) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(-\mu-\frac{\pi}{2}) & 0 & \sin(-\mu-\frac{\pi}{2}) \\ 0 & 1 & 0 \\ -\sin(-\mu-\frac{\pi}{2}) & 0 & \cos(-\mu-\frac{\pi}{2}) \end{bmatrix} \end{aligned} \quad (2.83)$$

Using the trigonometric formulae  $\cos(-\mu - \frac{\pi}{2}) = -\sin(\mu)$  and  $\sin(-\mu - \frac{\pi}{2}) = -\cos(\mu)$  yields

$$\mathbf{R}_n^e(\boldsymbol{\Theta}_{en}) = \begin{bmatrix} -\cos(l) \sin(\mu) & -\sin(l) & -\cos(l) \cos(\mu) \\ -\sin(l) \sin(\mu) & \cos(l) & -\sin(l) \cos(\mu) \\ \cos(\mu) & 0 & -\sin(\mu) \end{bmatrix} \quad (2.84)$$



**Figure 2.5** Definitions of longitude  $l$  and latitude  $\mu$  and the NED reference frame on the surface of the Earth. The  $D$  axis points in the normal direction to the Earth's surface.

Hence, the ECEF positions  $\mathbf{p}_{b/e}^e = [x, y, z]^\top$  can be found by integration of (2.82). This equation can also be used when designing a global waypoint tracking control system for ships.

#### Matlab

The rotation matrix  $\mathbf{R}_n^e(\Theta_{en})$  is computed using the MSS toolbox command

$$\mathbf{R} = \text{R11}(l, \mu)$$

#### Flat Earth Navigation

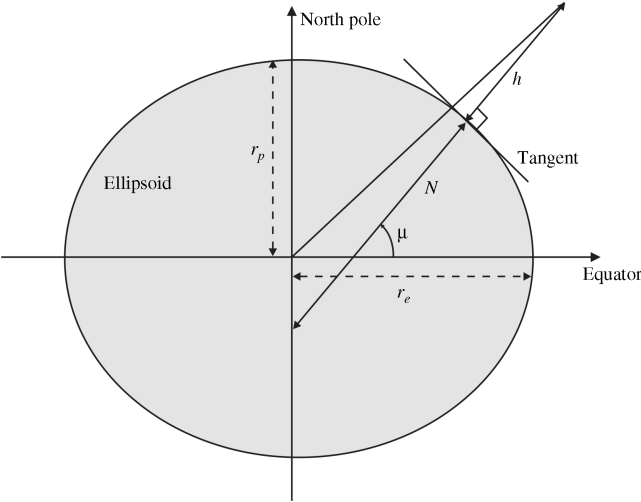
For local flat Earth navigation it can be assumed that the NED tangent plane is fixed on the surface of the Earth, that is  $l$  and  $\mu$  are constants, by assuming that the operating radius of the vessel is limited. This suggests that the NED position vector

$$\dot{\mathbf{p}}_{b/n}^n = \mathbf{R}_b^n(\Theta_{nb}) \mathbf{v}_{b/n}^b \quad (2.85)$$

is used for control design. When designing dynamic positioning (DP) systems for offshore vessels this is particularly useful. The ECEF coordinates for flat Earth navigation are found by requiring that  $\Theta_{en} = \text{constant}$ , such that

$$\mathbf{R}_n^e(\Theta_{en}) = \mathbf{R}_o = \text{constant} \quad (2.86)$$

$$\dot{\mathbf{p}}_{b/e}^e = \mathbf{R}_o \mathbf{R}_b^n(\Theta_{nb}) \mathbf{v}_{b/e}^b \quad (2.87)$$



**Figure 2.6** Definitions of the ellipsoidal parameters.

When designing global waypoint tracking control systems for ships, “flat Earth” is not a good approximation since  $(l, \mu)$  will vary largely for ships in transit between the different continents. Hence, the more general expression (2.82) should be used for global navigation.

### 2.3.2 Longitude and Latitude from ECEF Coordinates

The measurements of satellite navigation systems (GPS, GLONASS and Galileo) are given in the Cartesian ECEF frame, but there are measurements that do not make much sense to the user. The presentation of terrestrial position data  $\mathbf{p}_{b/e}^e = [x, y, z]^T$  is therefore made in terms of the ellipsoidal parameters longitude  $l$ , latitude  $\mu$  and height  $h$ .

Figure 2.6 shows the definitions of parameters needed for the transformations. The reference ellipsoid used for satellite navigation systems, WGS-84, is found by rotating an ellipse around the polar axis. Because of symmetry about the polar axis, it is only necessary to look at the meridian plane (latitude) equations. The origin of the ellipsoid coincides with the mass center of the Earth. The most important parameters of the WGS-84 ellipsoid are listed in Table 2.2.

In Figure 2.6,  $\mu$  is the *geodetic* latitude,  $h$  is the ellipsoidal height and  $N$  is the radius of curvature in the prime vertical.  $N$  is calculated by

$$N = \frac{r_e^2}{\sqrt{r_e^2 \cos^2(\mu) + r_p^2 \sin^2(\mu)}} \quad (2.88)$$

where the equatorial and polar earth radii,  $r_e$  and  $r_p$ , are the semi-axes of the ellipsoid.

**Table 2.2** WGS-84 parameters

Parameters	Comments
$r_e = 6\,378\,137\text{ m}$	Equatorial radius of ellipsoid (semi-major axis)
$r_p = 6\,356\,752\text{ m}$	Polar axis radius of ellipsoid (semi-minor axis)
$\omega_e = 7.292115 \times 10^{-5}\text{ rad/s}$	Angular velocity of the Earth
$e = 0.0818$	Eccentricity of ellipsoid

Longitude  $l$  is easily computed as

$$l = \text{atan} \left( \frac{y}{x} \right) \quad (2.89)$$

while latitude  $\mu$  and height  $h$  are implicitly computed by

$$\tan(\mu) = \frac{z}{p} \left( 1 - e^2 \frac{N}{N+h} \right)^{-1} \quad (2.90)$$

$$h = \frac{p}{\cos(\mu)} - N \quad (2.91)$$

where  $e$  is the eccentricity of the Earth given by

$$e = \sqrt{1 - \left( \frac{r_p}{r_e} \right)^2} \quad (2.92)$$

Since these two equations are implicit, they can be solved iteratively by using Algorithm 2.4 (Hofmann-Wellenhof *et al.*, 1994).

**Algorithm 2.4 (Transformation of  $(x, y, z)$  to  $(l, \mu, h)$ )**

1. Compute  $p = \sqrt{x^2 + y^2}$ .
2. Compute the approximate value  $\mu_{(0)}$  from

$$\tan(\mu_{(0)}) = \frac{z}{p} (1 - e^2)^{-1}$$

3. Compute an approximate value  $N$  from

$$N = \frac{r_e^2}{\sqrt{r_e^2 \cos^2(\mu_{(0)}) + r_p^2 \sin^2(\mu_{(0)})}}$$

4. Compute the ellipsoidal height by

$$h = \frac{p}{\cos(\mu_{(0)})} - N_{(0)}$$

5. Compute an improved value for the latitude by

$$\tan(\mu) = \frac{z}{p} \left( 1 - e^2 \frac{N_{(0)}}{N_{(0)} + h} \right)^{-1}$$

6. Check for another iteration step: if  $|\mu - \mu_{(0)}| < \text{tol}$ , where  $\text{tol}$  is a small number, then the iteration is complete. Otherwise set  $\mu_{(0)} = \mu$  and continue with Step 3.

**Matlab**

Algorithm 2.4 is programmed in the MSS toolbox as a function

$$[l, \mu, h] = \text{ecef211h}(x, y, z)$$

Several other algorithms can be used for this purpose; see Farrell and Barth (1998) and references therein. An approximate solution can also be found in Hofmann-Wellenhof *et al.* (1994) and an exact explicit solution is given by Zhu (1993).

**Height Transformation**

The WGS-84 ellipsoid is a global ellipsoid, which is only an approximation of the mean sea level of the Earth. It can deviate from the real mean sea level by as much as 100 meters at certain locations. The Earth's geoid, on the other hand, is defined physically and its center is coincident with the center of the Earth. It is an equipotential surface so that it has the same gravitational magnitude all over the surface, and the gravity vector is always perpendicular to the geoid.

The geoid is the surface chosen as a zero level reference. The ellipsoidal height  $h$  in Figure 2.7 must therefore be transformed to the *orthometric* height  $H$  in Figure 2.7 through the relation

$$h \approx H + M$$

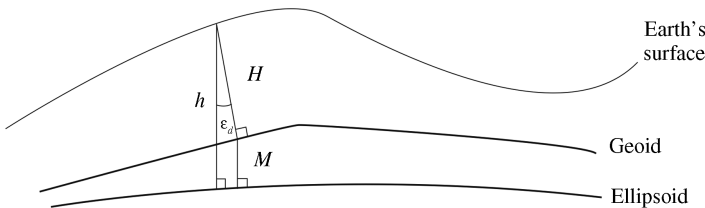
where  $M$  is called the *geoidal* height. The angle  $\varepsilon_d$  is small enough for the above approximation to be sufficiently accurate for all practical purposes. The angle  $\varepsilon_d$  is known as the deflection of the vertical, and does not exceed 30 arcseconds in most of the world. In fact the largest deflection encountered over the entire earth is in the order of 1 arcminutes (Britting, 1971). The geoidal height  $M$  is found through a *datum* transformation (Hofmann-Wellenhof *et al.*, 1994).

**2.3.3 ECEF Coordinates from Longitude and Latitude**

The transformation from  $\Theta_{en} = [l, \mu]^\top$  for given heights  $h$  to  $p_{b/e}^e = [x, y, z]^\top$  is given by (Heiskanen and Moritz, 1967)

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} (N + h) \cos(\mu) \cos(l) \\ (N + h) \cos(\mu) \sin(l) \\ \left( \frac{r_e^2}{r_e^2} N + h \right) \sin(\mu) \end{bmatrix} \quad (2.93)$$

For a ship  $h$  is the vertical distance from the sea level to the coordinate origin of  $\{b\}$ .



**Figure 2.7** Illustration of ellipsoidal and orthometric heights  $h$  and  $H$  where  $\varepsilon_d$  is the deflection of gravity and  $M$  is the geoidal height (undulation).



**Matlab**

The transformation from  $\Theta_{en} = [l, \mu]^\top$  to  $\mathbf{p}_{b/e}^e = [x, y, z]^\top$ , Equation (2.93), is programmed in the MSS toolbox function

$$[x, y, z] = \text{llh2ecef}(l, \mu, h)$$

**Example 2.4 (ECEF Coordinates from  $l$  and  $\mu$ )**

Assume that  $l = 10.3^\circ$ ,  $\mu = 63.0^\circ$  and  $h = 0$  m. Hence, the ECEF coordinates are computed to be

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 2\,856\,552 \text{ m} \\ 519\,123 \text{ m} \\ 5\,659\,978 \text{ m} \end{bmatrix}$$

using the MSS Matlab command

$$[x, y, z] = \text{llh2ecef}(10.3 * (\pi/180), 63.0 * (\pi/180), 0)$$

## 2.4 Transformations between BODY and FLOW

*Flow axes* are often used to express hydrodynamic data. The FLOW axes are found by rotating the BODY axis system such that resulting  $x$  axis is parallel to the freestream flow. Moreover, in FLOW axes, the  $x$  axis points directly into the relative flow while the  $z$  axis remains in the reference plane, but rotates so that it remains perpendicular to the  $x$  axis. The  $y$  axis completes the right-handed system. The transformation is outlined in Section 2.4.2.

The main reason for the FLOW axis system is that it is more convenient for calculating hydrodynamic forces. For instance, lift is, by definition, perpendicular to the relative flow, while drag is parallel. With FLOW axes, both lift and drag resolve into a force that is parallel to one of the axes.

### 2.4.1 Definitions of Course, Heading and Sideslip Angles

The relationship between the angular variables *course*, *heading* and *sideslip* is important for maneuvering of a marine craft in the horizontal plane (3 DOF). The terms course and heading are used interchangeably in much of the literature on guidance, navigation and control of marine craft, and this leads to confusion. Consequently, definitions utilizing a consistent symbolic notation will now be established.

The speed of a marine craft moving in the horizontal plane is

$$U = \sqrt{u^2 + v^2} \quad (2.94)$$

In the presence of ocean currents, the *relative speed* becomes

$$U_r = \sqrt{(u - u_c)^2 + (v - v_c)^2} \quad (2.95)$$

where the velocity components of the current are  $u_c$  and  $v_c$ .

The relationship between the angular variables is shown in Figure 2.8. The following definitions for motions in the horizontal plane are adopted from Breivik and Fossen (2004b):

**Definition 2.4 (Course Angle  $\chi$ )**

The angle from the  $x_n$  axis of  $\{n\}$  to the velocity vector of the craft, positive rotation about the  $z_n$  axis of  $\{n\}$  by the right-hand screw convention.

**Definition 2.5 (Heading (Yaw) Angle  $\psi$ )**

The angle from the  $x_n$  axis of  $\{n\}$  to the  $x_b$  axis of  $\{b\}$ , positive rotation about the  $z_n$  axis of  $\{n\}$  by the right-hand screw convention.

**Definition 2.6 (Sideslip (Drift) Angle  $\beta$ )**

The angle from the  $x_b$  axis of  $\{b\}$  to the velocity vector of the vehicle, positive rotation about the  $z_b$  axis of  $\{b\}$  by the right-hand screw convention.

By these definitions, it is apparent that the course angle satisfies

$$\chi = \psi + \beta \quad (2.96)$$

where

$$\beta = \sin^{-1} \left( \frac{v}{U} \right) \xrightarrow{\beta \text{ small}} \beta \approx \frac{v}{U} \quad (2.97)$$

This relationship is easily verified from Figure 2.8. The sideslip angle can be extended to include the effect of ocean currents by letting

$$\beta_r = \sin^{-1} \left( \frac{v_r}{U_r} \right) \xrightarrow{\beta_r \text{ small}} \beta_r \approx \frac{v_r}{U_r} \quad (2.98)$$

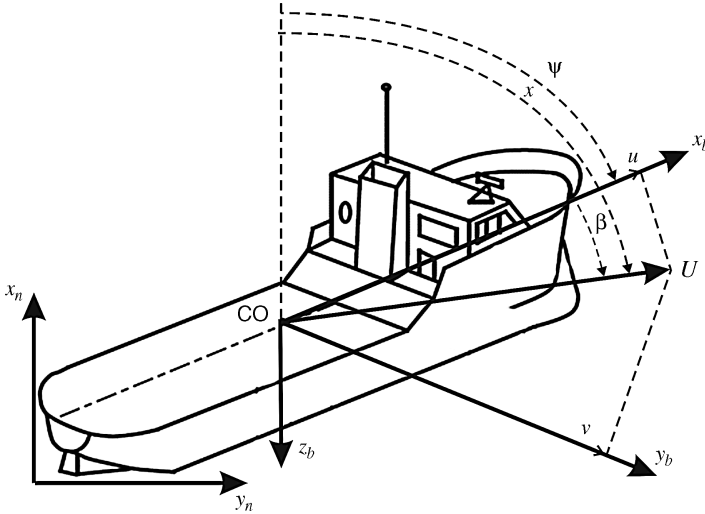
where  $v_r = v - v_c$  is the relative sway velocity.

**Remark 2.1**

In SNAME (1950) and Lewis (1989) the sideslip angle for marine craft is defined according to:

$$\beta_{\text{SNAME}} := -\beta$$

Note that the sideslip definition in this section follows the sign convention used by the aircraft community, for instance as in Nelson (1998) and Stevens and Lewis (1992). This definition is more intuitive from a guidance point of view, as shown in Figure 2.8.



**Figure 2.8** The geometrical relationship  $\chi = \psi + \beta$  between course  $\chi$ , heading angle  $\psi$  and sideslip angle  $\beta$ .

**Example 2.5 (Sideslip Angle: No Ocean Currents)**

Consider a marine craft moving at  $U = 10$  m/s under the assumption of no ocean currents and zero sway velocity ( $v = 0$ ). Hence, the sideslip angle is

$$\beta = \sin^{-1} \left( \frac{v}{U} \right) = 0 \quad (2.99)$$

For this case the heading angle equals the course angle, that is

$$\chi \equiv \psi \quad (2.100)$$

**Example 2.6 (Sideslip Angle: Ocean Currents)**

Consider a marine craft at rest and exposed for an ocean current  $u_c = v_c = 0.5$  m/s. Since  $u = v = 0$  it follows that  $U_r = \sqrt{u_c^2 + v_c^2}$  and

$$\beta_r = \sin^{-1} \left( \frac{-v_c}{\sqrt{u_c^2 + v_c^2}} \right) = -0.36 \quad (2.101)$$

corresponding to  $-20.7^\circ$ . In this case the heading and the course angles satisfy

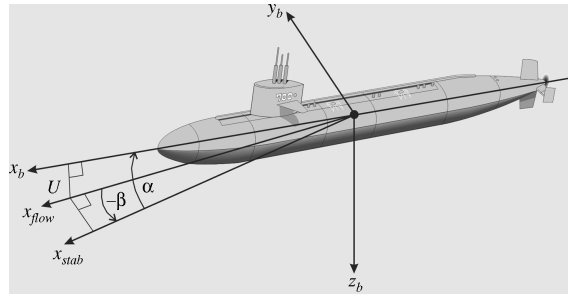
$$\chi = \psi - 20.7^\circ \quad (2.102)$$

## 2.4.2 Sideslip and Angle of Attack

The transformation from FLOW to BODY axes is defined by two principal rotations. First, the flow axes are rotated by a *negative* sideslip angle  $-\beta$  about the  $z$  axis and the new coordinate system is called *stability axes*. Second, the stability axes are rotated by a *positive* angle  $\alpha$  about the new  $y$  axis. This angle  $\alpha$  is called the *angle of attack*.

The names *stability* and *wind axes* are commonly used in aerodynamics to model lift and drag forces, which both are nonlinear functions of  $\alpha$ ,  $\beta$  and  $U$ . This convention has been adopted by the marine community and SNAME to describe lift and drag forces on submerged vehicles (SNAME, 1950). For marine craft, *wind axes* correspond to *flow axes*.

Stability and flow axes are also used in path following. For instance, a ship equipped with a single rudder and a main propeller can follow a path even though only two controls are available by simply steering the vessel to the path using the rudder. The speed is controlled by an independent propeller feedback loop (Fossen *et al.*, 2003a). This means that we control the  $xy$  coordinates and yaw angle  $\psi$  of the ship (3 DOF). When doing this, it is optimal to have a zero sideslip angle when there are no ocean currents, wave and wind loads. If the environmental forces are nonzero, it is optimal to have a nonzero sideslip angle, as shown in Figure 2.9 (Breivik and Fossen, 2005a). This is referred to as weathervaning.



**Figure 2.9** Illustration of stability and flow axes in terms of the angle of attack  $\alpha$  and the sideslip angle  $\beta$ .

However, it is extremely difficult to track the desired path given by  $x$  and  $y$ , and at the same time maintain a constant heading angle  $\psi$  unless three controls are available for feedback since this is an underactuated control problem.

The transformation between BODY, STABILITY and FLOW axes can be mathematically expressed as

$$\mathbf{v}^{\text{stab}} = \mathbf{R}_{y,\alpha} \mathbf{v}^b \quad (2.103)$$

$$\mathbf{v}^{\text{flow}} = \mathbf{R}_{z,-\beta} \mathbf{v}^{\text{stab}} \quad (2.104)$$

where

$$\mathbf{R}_{y,\alpha} = \begin{bmatrix} \cos(\alpha) & 0 & \sin(\alpha) \\ 0 & 1 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix} \quad (2.105)$$

$$\mathbf{R}_{z,-\beta} = \mathbf{R}_{z,\beta}^\top = \begin{bmatrix} \cos(\beta) & \sin(\beta) & 0 \\ -\sin(\beta) & \cos(\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.106)$$

The transformation matrix from BODY to FLOW axes then becomes

$$\begin{aligned} \mathbf{R}_b^{\text{flow}} &= \mathbf{R}_{z,-\beta} \mathbf{R}_{y,\alpha} \\ &= \begin{bmatrix} \cos(\beta) \cos(\alpha) & \sin(\beta) & \cos(\beta) \sin(\alpha) \\ -\sin(\beta) \cos(\alpha) & \cos(\beta) & -\sin(\beta) \sin(\alpha) \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix} \end{aligned} \quad (2.107)$$

The velocity transformation

$$\mathbf{v}^{\text{flow}} = \mathbf{R}_b^{\text{flow}} \mathbf{v}^b \quad (2.108)$$

can now be rewritten as

$$\mathbf{v}^b = (\mathbf{R}_b^{\text{flow}})^\top \mathbf{v}^{\text{flow}} \quad (2.109)$$

$\Updownarrow$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{R}_{y,\alpha}^\top \mathbf{R}_{z,-\beta}^\top \begin{bmatrix} U \\ 0 \\ 0 \end{bmatrix} \quad (2.110)$$

Writing this expression in component form yields

$$u = U \cos(\alpha) \cos(\beta) \quad (2.111)$$

$$v = U \sin(\beta) \quad (2.112)$$

$$w = U \sin(\alpha) \cos(\beta) \quad (2.113)$$

For a marine craft moving at constant forward speed  $U > 0$ , the angle of attack and sideslip angle become

$$\alpha = \tan^{-1} \left( \frac{w}{u} \right) \quad (2.114)$$

$$\beta = \sin^{-1} \left( \frac{v}{U} \right) \quad (2.115)$$

For small angles of  $\alpha$  and  $\beta$  (linear theory), it follows that

$$u \approx U, \quad v \approx \beta U, \quad w \approx \alpha U \quad (2.116)$$

and

$$\alpha \approx \frac{w}{U}, \quad \beta \approx \frac{v}{U} \quad (2.117)$$

Time differentiation of (2.112) under the assumption that  $U$  and the ocean current in  $\{n\}$  are constant gives the sway rate

$$\dot{v} = U \cos(\beta) \dot{\beta} \quad (2.118)$$

Consequently, the sideslip rate for  $U > 0$  becomes

$$\dot{\beta} = \frac{1}{U \cos(\beta)} \dot{v} \quad (2.119)$$

This relationship is exploited when designing path-following control systems (see Section 10.4).

### Extension to Ocean Currents

For a marine craft exposed to ocean currents, the concept of relative velocities is introduced; see Section 8.3. Let the current velocities expressed in  $\{b\}$  be denoted  $u_c$ ,  $v_c$  and  $w_c$ . The relative velocities are

$$u_r = u - u_c \quad (2.120)$$

$$v_r = v - v_c \quad (2.121)$$

$$w_r = w - w_c \quad (2.122)$$

such that the relative speed becomes

$$U_r = \sqrt{u_r^2 + v_r^2 + w_r^2} \quad (2.123)$$

Angle of attack and sideslip angle as given by (2.111)–(2.113) are modified in terms of the relative velocities according to

$$u_r = U_r \cos(\alpha_r) \cos(\beta_r) \quad (2.124)$$

$$v_r = U_r \sin(\beta_r) \quad (2.125)$$

$$w_r = U_r \sin(\alpha_r) \cos(\beta_r) \quad (2.126)$$

such that

$$\alpha_r = \tan^{-1} \left( \frac{w_r}{u_r} \right) \quad (2.127)$$

$$\beta_r = \sin^{-1} \left( \frac{v_r}{U_r} \right) \quad (2.128)$$

For small angles  $\alpha_r$  and  $\beta_r$ , it is seen that

$$u_r \approx U_r, \quad v_r \approx \beta_r U_r, \quad w_r \approx \alpha_r U_r \quad (2.129)$$

such that

$$\alpha_r \approx \frac{w - w_c}{U_r}, \quad \beta_r \approx \frac{v - v_c}{U_r} \quad (2.130)$$

### State-Space Transformation

The state-space model of a marine craft can be transformed to FLOW axes using a transformation matrix depending on speed. Let

$$\mathbf{v} = [u, v, w, p, q, r]^\top \quad (2.131)$$

and

$$\mathbf{v}^{\text{flow}} = [U, \beta, \alpha, p, q, r]^\top \quad (2.132)$$

The latter representation is often more intuitive to use from a hydrodynamic point of view, while control engineers prefer the former. Both representations are, however, equivalent since there exists a nonlinear transformation between  $(u, v, w)$  and  $(U, \beta, \alpha)$ , given by (2.111)–(2.113). For small angles, the following approximative transformation:

$$\mathbf{v}^{\text{flow}} = \mathbf{T}(U)\mathbf{v} \quad (2.133)$$

with

$$\mathbf{T}(U) = \text{diag} \left\{ 1, \frac{1}{U}, \frac{1}{U}, 1, 1, 1 \right\} \quad (2.134)$$

can be used. If we assume that  $U = \text{constant}$  such that  $\dot{\mathbf{T}}(U) = \mathbf{0}$ , the linear model

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{N}\mathbf{v} = \boldsymbol{\tau} \quad (2.135)$$

transforms to FLOW axes according to

$$\mathbf{T}(U)\mathbf{M}\mathbf{T}(U)^{-1}\mathbf{v}^{\text{flow}} + \mathbf{T}(U)\mathbf{N}\mathbf{T}(U)^{-1}\mathbf{v}^{\text{flow}} = \mathbf{T}(U)\boldsymbol{\tau} \quad (2.136)$$