

Chapter 3

Rotation Matrices



3.1 Introduction

Six parameters (degrees of freedom) are required and sufficient to completely describe the movement of an object in space: three describe the 3-D position of the object, and three the 3-D orientation, often referred to as “attitude” in aeronautics. When describing movements that are less than a few kilometers, we often use space-fixed, Cartesian coordinate systems. In these systems, the orientation of each axis is the same for each point in space, and for all time. For typical industrial applications and for movement measurements, a system fixed with respect to the surface of the earth can be regarded as space-fixed, as a convenient substitute of a true “inertial system”.¹ These coordinate systems are sometimes called local-level-local-North, assuming a flat earth. For larger distances, when the curvature of the earth becomes significant, the direction of “up” starts to depend on the location, and Cartesian coordinate systems are no longer useful. An example of such a non-cartesian coordinate system is the geodetic latitude/longitude/height coordinate system.

After choosing an arbitrary point in space as the *reference position*, the *position* of each point is defined by three translations away from the coordinate center, e.g., forward, left, and up. Here, it is worth pointing out a seemingly obvious fact: the final location of the object is independent of the sequence of these translations. If we move first 10 m right and then 15 m forward, we end up in the same location as if we had moved first 15 m forward, and then 10 m right. This property is referred to as the *commutativity* of translations (see also Fig. 5.1).

The description of orientation is done in a similar way (Fig. 3.1). First, an arbitrarily chosen orientation is defined as *reference orientation*. Once that is done, any other orientation can be described by three parameters: an object can not only be

¹An inertial frame is a frame of reference in which a body remains at rest or moves with a constant linear velocity unless acted upon by forces. An inertial reference frame does not have a single, universal coordinate system attached to it: positional values in an inertial frame can be expressed in any convenient coordinate system. In other words, an inertial frame is a frame of reference where the laws of inertia apply—there is no requirement for specific coordinates.

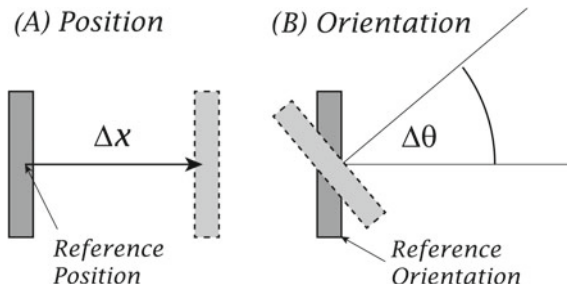


Fig. 3.1 Similarity between one-dimensional translations (left) and single-axis rotations (right): Both require the selection of a reference, and both are characterized by a single parameter

translated along each of the three coordinate axes, but it can also be rotated about each of these axes. To describe the three-dimensional orientation, two inherently different approaches can be taken. The first approach is based on *Euler's Theorem*, which states that for every two orientations of an object, the object can always move from one to the other by a single rotation about a fixed axis (Euler 1775). In that case, the axis of the rotation is defined by two parameters, and the magnitude of the rotation defines the third parameter. The second approach is to describe the rotation from the reference orientation to the current orientation through three consecutive rotations about well-defined, hierarchically nested coordinate axes (e.g., Goldstein 1980). For a long time, this has been the most common approach to characterize orientation in three dimensions. A detailed analysis of all the rotation angle sequences is given by (Diebel 2006).

The following section will deal with this three-rotation description of 3-D orientation, while the approach based on Euler's Theorem will be explained in detail in Chap. 4 ("Quaternions").

3.2 Rotations in a Plane

A simple rotation of a point \mathbf{p} in a plane can be uniquely described in two ways:

- In *Cartesian coordinates* through a rotation matrix.
- In *polar coordinates*, through an angle θ characterizing the rotation about an axis perpendicular to the plane.

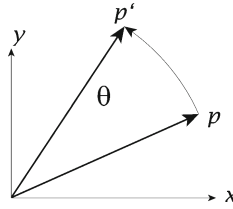


Fig. 3.2 Rotation in a plane

3.2.1 Rotation in Cartesian Coordinates

In two-dimensional Cartesian coordinates, a point \mathbf{p} is defined by its two coordinate components $\mathbf{p} = \begin{pmatrix} x \\ y \end{pmatrix}$. When that point is rotated by an angle θ into a new point $\mathbf{p}' = \begin{pmatrix} x' \\ y' \end{pmatrix}$ (Fig. 3.2), the coordinates of \mathbf{p}' are given by²

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}. \quad (3.1)$$

Defining the “rotation matrix” \mathbf{R} as

$$\mathbf{R} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}, \quad (3.2)$$

Equation (3.1) can be rewritten as

$$\mathbf{p}' = \mathbf{R} \cdot \mathbf{p}. \quad (3.3)$$

Note that the columns of the rotation matrix are equivalent to the basis vectors of the space-fixed coordinate system ($\mathbf{n}_x, \mathbf{n}_y$) rotated by the angle θ (Fig. 3.3)! Or in other words, the rotation matrix is the projection of the rotated unit vectors onto the coordinate axes. The rotation matrix is therefore sometimes also referred to as the “direction cosine matrix (DCM)”.

$$\mathbf{R} = \mathbf{R} \cdot [\mathbf{n}_x \ \mathbf{n}_y] = [\mathbf{n}'_x \ \mathbf{n}'_y]. \quad (3.4)$$

²Note for Matlab users: here and in the following, the dash in \mathbf{p}' does NOT mean the vector \mathbf{p} transposed, but rather the vector \mathbf{p} rotated!

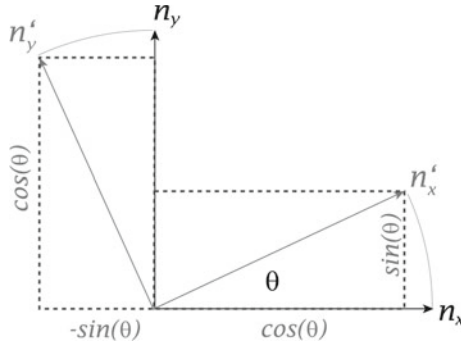


Fig. 3.3 Rotation matrix: projection in 2-D

3.2.2 Rotation in Polar Coordinates

Every complex number has a real and an imaginary part

$$c = x + j * y$$

and therefore can be represented by a vector in the (x, y) -plane (Fig. 3.4). And since

$$e^{j\theta} = \cos(\theta) + j * \sin(\theta), \tag{3.5}$$

every complex number can also be represented by a magnitude r and an angle θ :

$$c = r * e^{j\theta} = r * (\cos(\theta) + j * \sin(\theta)), \tag{3.6}$$

where

$$r = \sqrt{Re^2 + Im^2} \tag{3.7}$$

$$\theta = \arctan\left(\frac{Im}{Re}\right). \tag{3.8}$$

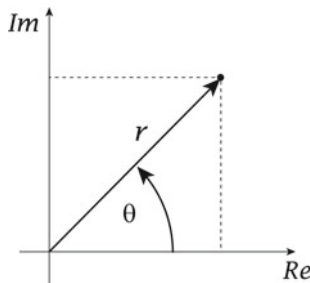


Fig. 3.4 Complex number, in polar coordinates (r, θ)

As a result, a rotation of a 2-D-vector, expressed as a complex number c , by an angle ϕ , can be written as

$$c' = e^{j\phi} * c = e^{j\phi} * (r * e^{j\theta}) = r * e^{j(\phi+\theta)}. \tag{3.9}$$

Note: In mathematics and physics, the square root of -1 is typically denoted with i , whereas in many technical areas j is used. In both Python and Matlab, j can be used:

```
x = -1+0j
np.sqrt(x)
>>> 1j
```

In polar coordinates, the similarity between one-dimensional translations and single-axis rotations becomes obvious (Fig. 3.1).

3.2.3 Application: Orienting an Object in a Plane

Task: If a gun originally pointing straight ahead along the $+x$ axis is to shoot at a target at $\mathbf{P} = (x, y)$, by which amount does the gun have to rotate to point at that target (Fig. 3.5)?

Solution:

The gun barrel originally points straight ahead, so the direction of the bullet aligns with \mathbf{n}_x .

The rotation of the gun is described by the rotation matrix $\mathbf{R} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} = [\mathbf{n}'_x \ \mathbf{n}'_y]$. The direction of the gun barrel after the rotation is given by $\mathbf{n}'_x = \frac{\mathbf{p}}{|\mathbf{p}|}$, which is also the first column of the rotation matrix \mathbf{R} .

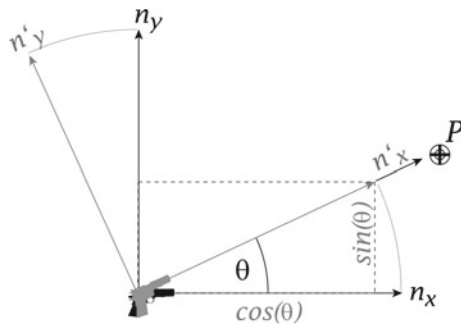


Fig. 3.5 Targeting with one degree-of-freedom (DOF)

Combining these two equations leads to the vector equation

$$\begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix} = \frac{1}{\sqrt{x^2 + y^2}} \begin{pmatrix} x \\ y \end{pmatrix}.$$

The y-component of this vector equation is $\sin \theta = \frac{y}{\sqrt{x^2 + y^2}} \rightarrow \theta = \arcsin \frac{y}{\sqrt{x^2 + y^2}}$

Note: For small angles ($\theta \ll 1$), $\sin(\theta)$ and $\cos(\theta)$ can be expanded with a Taylor series, and one obtains in a linear approximation

$$\begin{aligned} \sin(\theta) &\approx \theta + 0(\theta^2), \text{ and} \\ \cos(\theta) &\approx 1 + 0(\theta^2). \end{aligned} \tag{3.10}$$

As a result, for small angles, numerical errors are minimized by calculating the angle from measurements related to $\sin(\theta)$ as shown above. For angles around 90° the component proportional to $\cos(\theta)$ should be used, to minimize numerical errors in the computation of θ .

3.3 Rotations About Coordinate Axes in 3-D

3.3.1 3-D Rotations About Coordinate Axes

The same ideas as described above for rotations in a plane can be applied to three dimensions, leading to the 3-D rotation matrix

$$\mathbf{R} = [\mathbf{n}'_x \ \mathbf{n}'_y \ \mathbf{n}'_z], \tag{3.11}$$

where the \mathbf{n}'_i are column vectors. In the following, eye movements will often be used as an example, as they can be easily visualized. In that case, the *reference coordinate system* or *space-fixed coordinate system* will be the coordinate system provided by the head, and the *eye(body)-fixed coordinate system* will here be a coordinate system fixed to the body of the eye, with the x -axis aligned with the line of sight (often referred to as *gaze direction*).

In order to define single-axis rotations in three dimensions about coordinate axes, first an external, space-fixed coordinate system has to be defined and a body-fixed coordinate system to describe the three-dimensional orientation of the object with respect to space. Let $\mathbf{S} = [\mathbf{s}_x \ \mathbf{s}_y \ \mathbf{s}_z]$ be a right-handed, space-fixed coordinate system such that \mathbf{s}_x coincides with the line of sight when the eye is in the reference position, \mathbf{s}_y with the interaural axis (i.e., left-right), and \mathbf{s}_z with earth vertical (Fig. 3.6a).

Let $\mathbf{B} = [\mathbf{b}_x \ \mathbf{b}_y \ \mathbf{b}_z]$ (note: \mathbf{b}_i $i = x, y, z$ are column vectors!) denote a right-handed body-fixed coordinate system (i.e., it moves with the object, here the eye) such that \mathbf{B} coincides with the space-fixed coordinate system \mathbf{S} when the eye/body

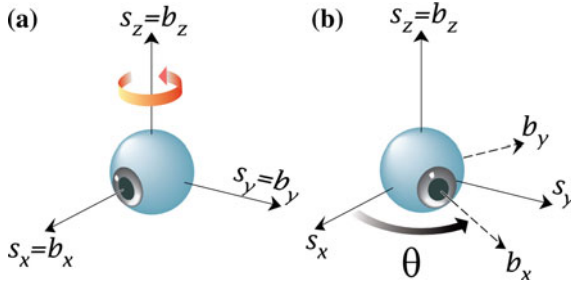


Fig. 3.6 Horizontal rotation of the eye about the space-fixed axis s_z by an angle θ from the reference orientation (a) to a new orientation (b)

is in the reference orientation. Any horizontal rotation of the body-fixed coordinate system (and thus of the object) from the reference orientation to a new orientation, as indicated in Fig. 3.6b, can be described by

$$\mathbf{b}_i = \mathbf{R} \cdot \mathbf{s}_i, \quad i = x, y, z, \tag{3.12}$$

or, equivalently

$$\mathbf{B} = \mathbf{R} \cdot \mathbf{S}. \tag{3.13}$$

The rotation matrix \mathbf{R} describes the orientation of the eye/body (\mathbf{B}) relative to the head (\mathbf{S}).

For a rotation of a point about a vertical axis in a space-fixed coordinate system, as indicated in Fig. 3.6b, the matrix \mathbf{R} describes a rotation about s_z by an angle of θ . This matrix, which we here call $\mathbf{R}_z(\theta)$, is given by

$$\mathbf{R}_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.14}$$

In the same way, vertical rotation of a point about s_y in a space-fixed coordinate system by an angle of ϕ can be described by

$$\mathbf{R}_y(\phi) = \begin{bmatrix} \cos \phi & 0 & \sin \phi \\ 0 & 1 & 0 \\ -\sin \phi & 0 & \cos \phi \end{bmatrix}, \tag{3.15}$$

and torsional rotation of a point about s_x in a space-fixed coordinate system by an angle of ψ by

$$\mathbf{R}_x(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & -\sin \psi \\ 0 & \sin \psi & \cos \psi \end{bmatrix}. \tag{3.16}$$

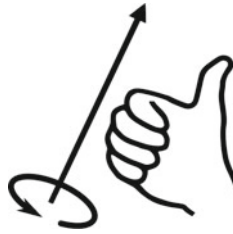


Fig. 3.7 Right-hand rule for rotations

With these definitions, and with the positive (x, y, z) -axis pointing forward/left-/up, respectively, positive θ , ϕ , and ψ values correspond to leftward, downward, and clockwise movements. Depending on the context, these three rotations are either called *yaw*, *pitch* and *roll* angles (in nautical applications, or in eye movement research), or—equivalently—*heading*, *elevation* and *banking* angles (in aerospace engineering). In the latter case, the term *attitude* is used to characterize 3-D orientation.

Notes:

- The direction of positive rotations can easily be remembered with the right-hand rule (Fig. 3.7): if a body is gripped with the right hand and rotated in the direction of the curled fingers, the direction of the thumb determines the sign of the rotation. With the coordinate system as defined in Fig. 3.6, rotations to the left, downward(!), and clockwise (as seen from the user) are positive.
- Care has to be taken with the implementation of rotations, such as in Eq. (3.13), if the data are in row format. For example, if data are stored as

$$\mathbf{Data} = \begin{bmatrix} x_0 & y_0 & z_0 \\ x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} \quad (3.17)$$

then using the matrix notation, the rotation of these data has to be implemented as

$$\mathbf{Data}' = [\mathbf{R} \cdot \mathbf{Data}^T]^T = \mathbf{Data} \cdot \mathbf{R}^T, \quad (3.18)$$

because

$$[\mathbf{A} \cdot \mathbf{B}]^T = \mathbf{B}^T \cdot \mathbf{A}^T. \quad (3.19)$$

3.3.2 Rotations of Objects Versus Rotations of Coordinate Systems

The next step is conceptually trivial, but its ramifications have caused consternation among generations of scientists.

Figure 3.8a shows the reference setup, where a picture is taken of a cat, with the cat's nose in the center of the image. In Fig. 3.8b, *the cat* has been rotated by 25° , and its nose is now at the lower edge of the captured image. In Fig. 3.8c, the cat remains stationary, but now *the camera* coordinate system is rotated by 25° , this time in the opposite direction. The image of the cat looks exactly the same as in Fig. 3.8b.

From the definition of our coordinate system (forward, left, and up are the positive directions), and the choice of the right-hand rule, a “downward” rotation has to be positive. And in both cases (Fig. 3.8b, c), the final relative orientation between camera and cat is the same. But now the catch is: should the relative movement shown in Fig. 3.8b and c be labeled a *downward* rotation, because the cat is rotated downward relative to the camera? Or should it be labeled an *upward* rotation, because the camera is rotated up with respect to the surroundings? In situations where the measurement setup is stationary, such as in Fig. 3.8b, it makes sense to call this relative rotation a *downward* rotation. (This is the convention used in this book, where \mathbf{R} describes the rotation of an object relative to a space-fixed coordinate system.)! But when a constant environment is observed from a moving object, as in Fig. 3.8c, it makes more sense to call this relative rotation an *upward* rotation, because the camera rotates *upward*. (That is the convention often used in theoretical physics, theoretical mechanics, and aeronautics, where a fixed world and fixed events are observed from different, moving reference systems.) It would be appropriate to describe, for example, the orientation of a moving aeroplane relative to a fixed-world environment.)

There exists no *correct* choice here, only two different options. Which one is chosen depends on the field of research and the application.

When comparing the equations from this book with other literature, carefully check if the definition of the rotation matrices is consistent with Eqs. (3.14)–(3.16). If this is the case, everything is fine. But if the definition of the rotation matrices

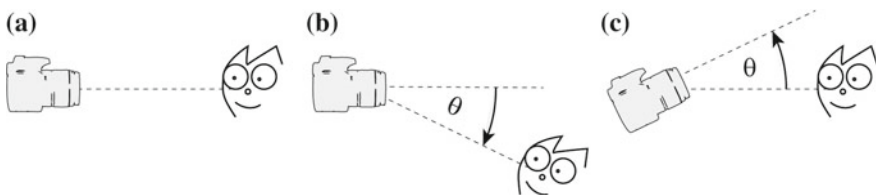


Fig. 3.8 a Reference picture. b “Object” rotated by 25° . c Camera coordinate system rotated by -25° , i.e., in the opposite direction

is the transposed of Eqs. (3.14)–(3.16), the alternate choice has been made, which affects all subsequent equations!³

Important Note

Even though I am repeating myself: Care has to be taken, because the exact form of the rotation matrices depends on the definition of \mathbf{R} . Technical applications often use rotations of the coordinate system for the definition of the rotation matrix, and the signs of the angles are inverted compared to our definitions in Eqs. (3.14)–(3.16). In those applications, for example in the excellent summary by (Diebel 2006), the rotation matrices are exactly the transposed versions of the matrices used here!!

3.4 Combined Rotations

For rotations about a single axis, no distinction has to be made between rotations about body-fixed or space-fixed axes. Since the body-fixed and space-fixed coordinate systems coincide when the object is in the reference position, the axis about which the object rotates is the same in the body-fixed and space-fixed system. But this is no longer the case for combined rotations about different axes. For such rotations the elements of \mathbf{R} are no longer determined by the relatively simple formulas in Eqs. (3.14)–(3.16).

The example in Fig. 3.9 may help to better understand the problem: how should we distinguish between a downward movement of the object by a rotation about the space-fixed axis \mathbf{s}_y (as shown in Fig. 3.9a) and a downward movement by a rotation about the rotated, body-fixed axis \mathbf{b}_y (Fig. 3.9b)?

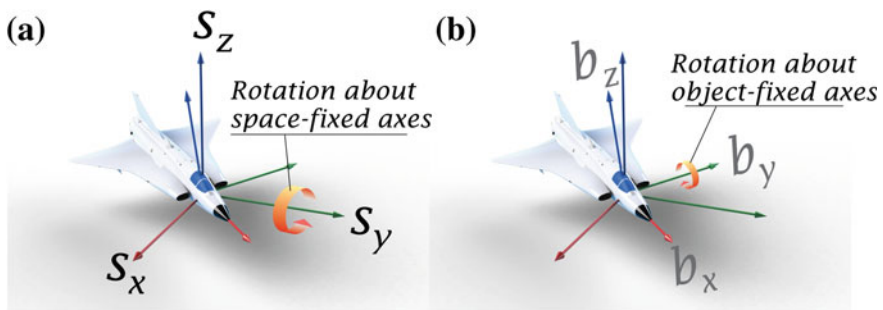


Fig. 3.9 In describing a combined horizontal–vertical movement, one has to distinguish clearly if the vertical movement is (a) a rotation about the space-fixed y -axis \mathbf{s}_y , which remains fixed, or (b) a rotation about the object-fixed y -axis \mathbf{b}_y , which moves with the object

³Appendix A.3.3 contains the proof that the body-fixed representation of rotations uses the inverse (i.e., the transpose) rotation matrix compared to the space-fixed representation.

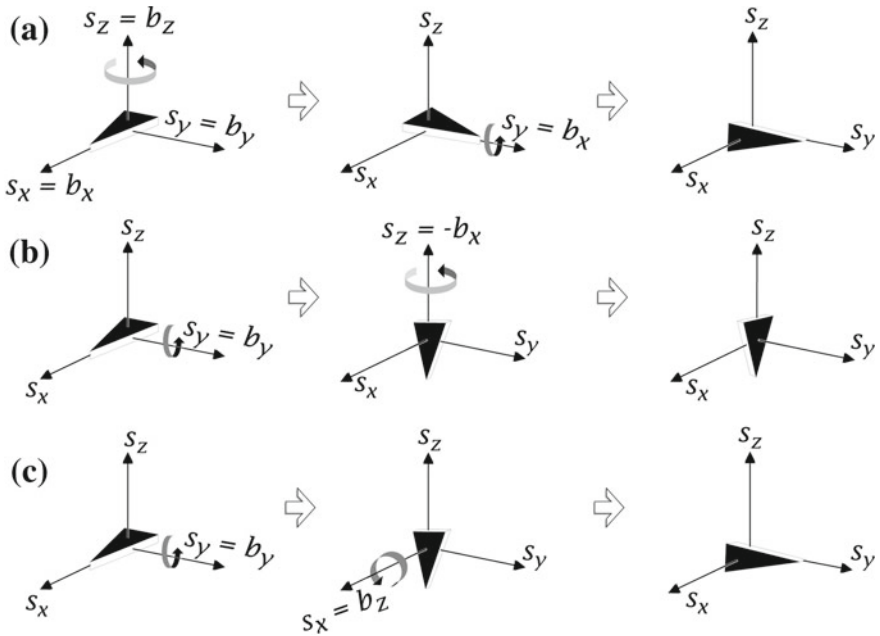


Fig. 3.10 Sequences of two rotations. (a) **Space-fixed**, $s_z :: s_y : 90^\circ$ rotation about the vertical axis s_z , followed by a 90° rotation about the horizontal axis s_y . (b) **Space-fixed**, $s_y :: s_z : 90^\circ$ rotation about the horizontal axis s_y , followed by a 90° rotation about the vertical axis s_z . (c) **Body-fixed**, $b_y :: b_z : 90^\circ$ rotation about the body-fixed axis b_y , followed by a 90° rotation about the body-fixed axis b_z . The final orientation is the same as in (a). Body-fixed axes and space-fixed axes are superposed because the size of the rotations in this example is exactly 90°

Mathematically, the difference between rotations in space-fixed coordinates and body-fixed coordinates lies in the sequence in which the rotations are executed. This is illustrated in Fig. 3.10. The upper column (Fig. 3.10a) shows a rotation of an object about s_z by $\theta = 90^\circ$, followed by a rotation about the space-fixed axis s_y by $\phi = 90^\circ$. Mathematically, this is described by

$$\mathbf{b}_i = \mathbf{R}_y(\phi) \cdot \mathbf{R}_z(\theta) \cdot \mathbf{s}_i \tag{3.20}$$

with $\theta = \phi = 90^\circ$.

Note: The rotation that is executed first is on the *right-hand side*, because this is the first matrix to act on the object to be rotated:

$$\mathbf{R}_y(\phi) \cdot (\mathbf{R}_z(\theta) \cdot \mathbf{s}_i) = (\mathbf{R}_y(\phi) \cdot \mathbf{R}_z(\theta)) \cdot \mathbf{s}_i \tag{3.21}$$

This leads to

Rule 1: Subsequent rotations are written *right-to-left*.

Inverting the sequence of two rotations about space-fixed axes changes the final orientation of the object. This can be seen in Fig. 3.10b, where the sequence of rotations is inverted: the first rotation is about the space-fixed axis \mathbf{s}_y , and the second rotation about the space-fixed \mathbf{s}_z . This sequence is mathematically described by

$$\mathbf{b}_i = \mathbf{R}_z(\theta) \cdot \mathbf{R}_y(\phi) \cdot \mathbf{s}_i. \quad (3.22)$$

Equations (3.20) and (3.22) both describe rotations about space-fixed axes. However, they can also be re-interpreted as rotations about body-fixed axes in the reverse sequence: Eq. (3.20) can be re-interpreted as a rotation about the axis \mathbf{b}_y by ϕ , followed by a rotation about the body-fixed axis \mathbf{b}_z by θ (Fig. 3.10c). Figures 3.10a and c demonstrate that rotations about space-fixed axes and rotations about object-fixed axes in the reverse sequence lead to the same final orientation. And Eq. (3.22) is equivalent to a rotation about \mathbf{b}_z by θ , followed by a rotation about the body-fixed axis \mathbf{b}_y by ϕ . A mathematical analysis of this problem can be found in (Altmann 1986).

This can be summarized as

Rule 2: A switch from a representation of subsequent rotations from space-fixed axes to body-fixed axes has to be accompanied by an inversion of the sequence of the rotation matrices.

This also gives the answer to the problem raised by Fig. 3.9: the combination of two rotations about the space-fixed axes \mathbf{s}_z and \mathbf{s}_y , as shown in Fig. 3.9a, is mathematically described by Eq. (3.20), while the combination of two rotations about the object-fixed axes \mathbf{b}_z and \mathbf{b}_y , as shown in Fig. 3.9b, is described by Eq. (3.22).

Rotations about space-fixed axes are often called “rotations of the object” or “active rotations”, since in successive rotations only the object is rotated, and the axes of the successive rotations are unaffected by the preceding rotations of the object. Rotations about object-fixed axes are often referred to as “rotations of the coordinate system” or “passive rotations”, since each rotation changes the coordinate axes about which the next rotations will be performed.

3.4.1 3-D Orientation with Sequential Rotations

Systems that use such a combination of three rotations for the description of the orientation can be demonstrated with *gimbal systems*. A gimbal is a ring or a frame that is suspended so it can rotate about an axis. Gimbals are typically nested one within another to accommodate rotation about multiple axes, and the hierarchy of sequential rotations is automatically implemented. As pointed out in the Chap. 1, we will by default use an inertial coordinate system where the positive x-, y-, and z-axes point forward, left, and up.

3.4.1.1 Nautical Sequence

In aeronautics (Kuipers 1999) and maritime applications, the yaw–pitch–roll sequence of rotations, which we will refer to as “nautical sequence”, is very common.

The **first** rotation is about the (vertical) z -axis. This movement is called a “yaw movement”, and the corresponding angle is sometimes referred to as “heading angle”. According to the right-hand rule, a positive rotation rotates the nose of the airplane to the left.

The **second** movement is about the once-rotated, body-fixed y -axis (when the object is an airplane, about the line connecting the wings). The movement is called a “pitch movement”, and the corresponding angle is referred to as “elevation angle”.

The **last** rotation is about the twice-rotated x -axis (when the object is an airplane, about the longitudinal axis). This movement is called “roll movement”, and the corresponding angle is the “banking angle”. If the rotated object is a camera or an eye, this roll rotation will not change the line of sight or gaze direction, but it will rotate the image.

This sequence has first been used by the German doctor and physiologist Adolf Fick (Fick 1854), who worked on eye movements and who also invented the first contact lenses worn by patients. In eye movement research, the yaw, pitch, and roll angles for this sequence are therefore often referred to as “Fick angles”. The yaw and pitch angles together determine the line of sight, and the corresponding direction is called the “gaze direction”.

The left illustration in Fig. 3.11 shows a gimbal which corresponds to the nautical sequence of rotations. The angles of the nautical sequence will be denoted by the subscript “N” (θ_N, ϕ_N, ψ_N). The rotation matrix corresponding to the nautical sequence of rotations is

$$\mathbf{R}_{\text{nautical}} = \mathbf{R}_z(\theta_N) \cdot \mathbf{R}_y(\phi_N) \cdot \mathbf{R}_x(\psi_N), \quad (3.23)$$

where the rotation matrices $\mathbf{R}_x, \mathbf{R}_y, \mathbf{R}_z$ describe per definition rotations about space-fixed axes. The discussion of Eqs. (3.20) and (3.22) defines the sequence in which nested rotations have to be written down: the first rotation (i.e., the one on the right-hand side) has to be the rotation of the innermost axis, since this is the only rotation that does not affect the other ones. This can be formulated as

Rule 3: For nested rotations, the sequence of rotations has to be written *from the inside out*, in order to ensure rotations about the correct axes.

Inserting Eqs. (3.14)–(3.16) into Eq. (3.23) leads to⁴

⁴The required work to find those matrices is much reduced using the *symbolic computation* packages offered by many scripting languages. For Python, the implementation is shown in Appendix C.4.2.

$$\mathbf{R}_{nautical} = \begin{bmatrix} \cos \theta_N \cos \phi_N & \cos \theta_N \sin \phi_N \sin \psi_N - \sin \theta_N \cos \psi_N & \cos \theta_N \sin \phi_N \cos \psi_N + \sin \theta_N \sin \psi_N \\ \sin \theta_N \cos \phi_N & \sin \theta_N \sin \phi_N \sin \psi_N + \cos \theta_N \cos \psi_N & \sin \theta_N \sin \phi_N \cos \psi_N - \cos \theta_N \sin \psi_N \\ -\sin \phi_N & \cos \phi_N \sin \psi_N & \cos \phi_N \cos \psi_N \end{bmatrix}. \quad (3.24)$$

This provides a convenient way to obtain the angles $(\theta_N, \phi_N, \psi_N)$ from the rotation matrix \mathbf{R}

$$\begin{aligned} \phi_N &= -\arcsin(R_{zx}) \\ \theta_N &= \arcsin\left(\frac{R_{yx}}{\cos \phi_N}\right) \\ \psi_N &= \arcsin\left(\frac{R_{zy}}{\cos \phi_N}\right). \end{aligned} \quad (3.25)$$

Helmholtz Sequence

The nautical sequence is not the only sequence to describe the 3-D orientation of an object. Helmholtz (1867), another German physicist and physiologist from the nineteenth century, thought it would be better to start with a rotation about a horizontal axis. He characterized eye positions by a rotation about the horizontal interaural axis (i.e., the y -axis), followed by a rotation about the vertical axis, and then by a rotation about the line of sight, as shown in the right gimbal in Fig. 3.11:

$$\mathbf{R}_{Helm} = \mathbf{R}_y(\phi_H) \cdot \mathbf{R}_z(\theta_H) \cdot \mathbf{R}_x(\psi_H). \quad (3.26)$$



Fig. 3.11 In gimbal systems, the axes of rotation are determined by the geometry of system. Both gimbals in this figure are in the reference orientation. Let \mathbf{b}_x , \mathbf{b}_y , \mathbf{b}_z describe a body-fixed coordinate system. **Left** In a *nautical (Fick) gimbal*, the orientation of the object (the turn on the inner dial) is completely characterized by a rotation about the vertical axis \mathbf{b}_z by θ_N , followed by a rotation about the (rotated) horizontal axis \mathbf{b}_y by ϕ_N , and a rotation about the (twice-rotated) dial-axis \mathbf{b}_x by ψ_N . **Right** In a *Helmholtz gimbal*, the orientation of the inner dial is characterized by a rotation first about the horizontal axis \mathbf{b}_y by ϕ_H , followed by a rotation about the (rotated) \mathbf{b}_z axis by θ_H , and then a rotation about the dial-axis \mathbf{b}_x by ψ_H

The subscript ‘‘H’’ indicates that the angles refer to the *Helmholtz sequence* of rotations. One should keep in mind that the orientation of the object is characterized by the values of the rotation matrix \mathbf{R} , and $\mathbf{R}_{nautical}$ and \mathbf{R}_{Helm} only give different sequence parameterizations for the rotation matrix. But once constructed, the matrix is used in the same manner. Using Eqs. (3.14)–(3.16) and matrix multiplication, we get

$$\mathbf{R}_{Helm} = \begin{bmatrix} \cos \theta_H \cos \phi_H & -\sin \theta_H \cos \phi_H \cos \psi_H + \sin \phi_H \sin \psi_H & \sin \theta_H \cos \phi_H \sin \psi_H + \sin \phi_H \cos \psi_H \\ \sin \theta_H & \cos \theta_H \cos \psi_H & -\cos \theta_H \sin \psi_H \\ -\cos \theta_H \sin \phi_H & \sin \theta_H \sin \phi_H \cos \psi_H + \cos \phi_H \sin \psi_H & -\sin \theta_H \sin \phi_H \sin \psi_H + \cos \phi_H \cos \psi_H \end{bmatrix} \quad (3.27)$$

When \mathbf{R} is given, the Helmholtz angles $(\theta_H, \phi_H, \psi_H)$ can be using

$$\begin{aligned} \theta_H &= \arcsin(R_{yx}) \\ \phi_H &= -\arcsin\left(\frac{R_{zx}}{\cos \theta_H}\right) \\ \psi_H &= -\arcsin\left(\frac{R_{yz}}{\cos \theta_H}\right). \end{aligned} \quad (3.28)$$

Euler Sequence

Yet another sequence to describe 3-D orientation is common in theoretical physics and mechanics and in other technical literature, and often referred to as *Euler sequence*.⁵

In order to describe the movement of a spinning top rotating on a table, or of the earth during its rotation around the sun (see Fig. 3.12), three angles are needed: the intrinsic *rotation* (γ), *nutation* (β), and *precession* (α). Using these three angles, the orientation of the spinning object is described by (see Fig. 3.13)

- a rotation about the z -axis, by an angle α ,
- followed by a rotation about the rotated x -axis, by an angle β , and
- followed by a rotation about the twice-rotated z -axis, by an angle γ .

$$\mathbf{R}_{Euler} = \mathbf{R}_z(\alpha) \cdot \mathbf{R}_x(\beta) \cdot \mathbf{R}_z(\gamma). \quad (3.29)$$

This leads to the parametrization

$$\mathbf{R}_{Euler} = \begin{bmatrix} -\sin \alpha_E \cos \beta_E \sin \gamma_E + \cos \alpha_E \cos \gamma_E & -\sin \alpha_E \cos \beta_E \cos \gamma_E - \cos \alpha_E \sin \gamma_E & \sin \alpha_E \sin \beta_E \\ \sin \alpha_E \cos \gamma_E + \cos \alpha_E \cos \beta_E \sin \gamma_E & -\sin \alpha_E \sin \gamma_E + \cos \alpha_E \cos \beta_E \cos \gamma_E & -\cos \alpha_E \sin \beta_E \\ \sin \beta_E \sin \gamma_E & \sin \beta_E \cos \gamma_E & \cos \beta_E \end{bmatrix} \quad (3.30)$$

⁵The expression *Euler angles* should be used very carefully: sometimes, these angles represent the *Euler sequence*, but often that expression is also applied when the nautical sequence is actually used!

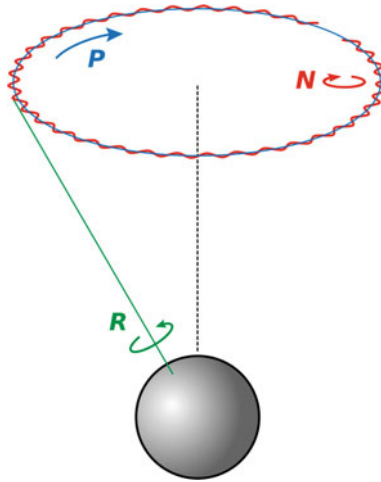


Fig. 3.12 Basic Euler motions of the earth. Intrinsic rotation (green “R”), precession (blue “P”), and nutation (red “N”). (From Wikipedia, original design by Dr. H. Sulzer)

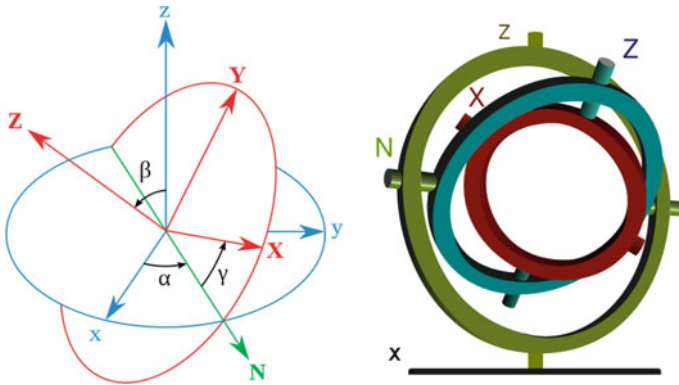


Fig. 3.13 Euler sequence: **Left**—The xyz (fixed) system is shown in blue, and the XYZ (rotated) system is shown in red. The line of nodes, labeled N, is shown in green (from Wikipedia). **Right** The corresponding gimbal

The corresponding angles can be calculated with

$$\begin{aligned}
 \gamma_E &= \text{atan2}(R_{zx}, R_{zy}) \\
 \beta_E &= \arccos(R_{zz}) \\
 \alpha_E &= -\text{atan2}(R_{xz}, R_{yz}).
 \end{aligned}
 \tag{3.31}$$

Notes:

- $\text{atan2}(a, b)$ is equivalent to $\arctan(\frac{a}{b})$ where it also takes into account the quadrant that the point (a, b) is in.
- For β , there are generally two solutions in the interval $(\pi, \pi]$. The above formula works only when β is within the interval $[0, \pi)$.

Other Sequences

Depending on the application, sometimes yet other sequences can be used. For example, the most important angles in gait analysis are the knee and hip angles, which correspond approximately to rotations about the y -axis. Therefore, in gait analysis, the sequence $\mathbf{R}_z \cdot \mathbf{R}_x \cdot \mathbf{R}_y$ is common.

The names for different gimbal systems may differ significantly, depending on the area of application. In some contexts, angles of any type of gimbal system are referred to as “Euler angles”. Angles from rotation sequences that involve all three axes (x - y - z , y - z - x , z - x - y , x - z - y , z - y - x , y - x - z) can be either called “Tait–Bryan angles”, in honor of the Scottish mathematical physicist Peter Tait (1832–1901), who was— together with Hamilton—the leading exponent of quaternions, and the Welshman George Bryan (1864–1928), the originator of the equations of airplane motion or “Cardan angles”, after the Renaissance mathematician, physician, astrologer, and gambler Jerome Cardan (1501–1576), who first described the cardan joint which can transmit rotary motion. And angles that have the same axis for the first and the last rotation (like the *Euler sequence* above) are called “proper Euler angles” (z - x - z , x - y - x , y - z - y , z - y - z , x - z - x , y - x - y).

3.4.2 Gimbal Lock

Consider tracking a helicopter flying from the horizon toward an aerial gun, as indicated in Fig. 3.16. The helicopter flies toward the gun site and is tracked by the gun in elevation (Φ) and azimuth (Θ). When the helicopter is immediately above the gun site, the aerial gun is in the orientation indicated in Fig. 3.14. If the helicopter now changes direction and flies at 90° to its previous course, the gun cannot track this maneuver without a discontinuous jump in one or both of the gimbal orientations. There is no continuous motion that allows it to follow the target—it is in “gimbal lock”. Note that even if the helicopter does not pass through the gimbal’s zenith, but only near it, so that gimbal lock does not occur, the system must still move exceptionally rapidly to track the helicopter if it changed direction, as it rapidly passes from one bearing to the other. The closer to zenith the nearest point is, the faster this must be done, and if it actually goes through zenith, the limit of these “increasingly rapid” movements becomes infinitely fast, i.e., discontinuous. Another way to describe gimbal lock is to consider the inverse of matrices Eqs. (3.24, 3.27, and 3.30). The mathematical equivalent of a gimbal lock is if the calculation of the inverse of the matrices in Eqs. (3.24, 3.27, and 3.30) contains a divide by zero condition.



Fig. 3.14 Gimbal lock: the orientation of the innermost axis cannot be reoriented in the direction of the dotted arrows

3.5 Homogeneous Coordinates

3.5.1 Definition

An application where a tremendous amount of 3-D calculations have to be performed is the rendering of 3-D scenes in computer graphics. With every change in position and orientation of the observer the appearance of each 3-D surface element of the scene has to be re-calculated. The calculations require not only translations and rotations but also scaling and perspective distortions (McConnell 2005).

The mathematical discipline of *projective geometry* has found a way to perform all the required calculations efficiently in one step (see Fig. 3.15). For this approach to work, the number of coordinates for each point has to be increased from three to four, and the extended coordinates are called “homogeneous coordinates” or “projective coordinates”:

$$\mathbf{p} \rightarrow \begin{pmatrix} \mathbf{p} \\ 1 \end{pmatrix}. \quad (3.32)$$

The additional, fourth element in the coordinates is essentially a scaling factor.

The matrix required to execute a generalized perspective transformation is a 4×4 matrix. For example, a rotation and translation can be “homogenized”, i.e., executed in one step, as described in the following. A point \mathbf{p} can be rotated and translated by

$$\mathbf{p}' = \mathbf{R} \cdot \mathbf{p} + \mathbf{t}, \quad (3.33)$$

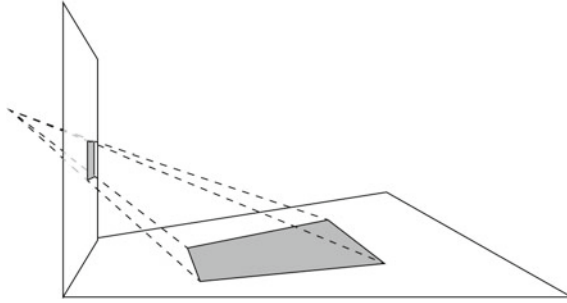


Fig. 3.15 Projection from one plane into another

where \mathbf{p} indicates the starting point, \mathbf{R} is the rotation matrix, \mathbf{t} is the translation, and \mathbf{p}' is the new location. Using the “homogeneous” coordinates defined in Eq. (3.32), this can be written in one step as

$$\begin{pmatrix} \mathbf{p}' \\ 1 \end{pmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} \mathbf{p} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{R} \cdot \mathbf{p} + \mathbf{t} \\ 1 \end{pmatrix}. \quad (3.34)$$

The resulting matrix $\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix}$ is a 4×4 matrix (since \mathbf{R} is a 3×3 matrix, and one row and one column have been added) and is called “spatial transformation matrix”. Similarly, geometric projections can be implemented by allowing general 4×4 matrices for the transformation.

By moving from 3-D Euclidian coordinates to 4-D homogeneous coordinates, all those transformations can be achieved with the same type of operation. The disadvantage: to represent a point, we now need four instead of the previous three numbers. The advantage: most image manipulations can now be performed in the same way. This homogeneous treatment of the different operations is reflected in the name, “homogeneous coordinates”. This has become a huge advantage in computer vision: graphics cards can perform massive matrix multiplications. For comparison, modern multi-core CPUs (central processing units, i.e., the processor in a typical PC) have on the order of 16 pipelines, while GPUs (graphical processing units, the processors that are used for graphics cards) have on the order of 1024 pipelines. Thereby, instructions can be executed in parallel and can be dramatically optimized.

An important application of homogeneous coordinates are the “Denavit Hartenberg parameters”. These are used in mechanical engineering to denote position and orientation of an end-link in robot manipulators (see Appendix B).

3.6 Applications

A few examples of the application of rotation matrices may help to show how to use them in practical applications.

3.6.1 Two DOF—Targeting an Object in 3-D

An aerial gun is mounted like a nautical gimbal: the outermost rotation is always about an earth-vertical axis (Fig. 3.16).

In the starting orientation ($\theta/\phi = 0/0$), the barrel of the aerial gun points straight ahead, i.e., along \mathbf{b}_x .

Task: When a target appears at $\mathbf{p} = (x, y, z)$, we want to reorient the gun such that the rotated gun barrel, which after the rotation points in the direction of \mathbf{b}'_x , points at the target.

Solution: Taking the rotation matrix in the nautical sequence (Eq. 3.24), with $\psi = 0$ since a rotation about the line of the gun barrel is not relevant, we get

$$\mathbf{R}_{\text{nautical}}(\psi_N = 0) = [\mathbf{b}'_x \ \mathbf{b}'_y \ \mathbf{b}'_z] = \begin{bmatrix} \cos \theta_N \cos \phi_N & -\sin \theta_N \cos \theta_N \sin \phi_N & \\ \sin \theta_N \cos \phi_N & \cos \theta_N \sin \theta_N \sin \phi_N & \\ -\sin \phi_N & 0 & \cos \phi_N \end{bmatrix}.$$

With the first column $\mathbf{b}'_x = \frac{\mathbf{p}}{|\mathbf{p}|}$, this leads to

$$\begin{aligned} \phi_N &= -\arcsin\left(\frac{p_z}{\sqrt{p_x^2 + p_y^2 + p_z^2}}\right) \\ \theta_N &= \arcsin\left(\frac{p_y}{\sqrt{p_x^2 + p_y^2 + p_z^2}} \cdot \frac{1}{\cos \phi_N}\right). \end{aligned} \quad (3.35)$$

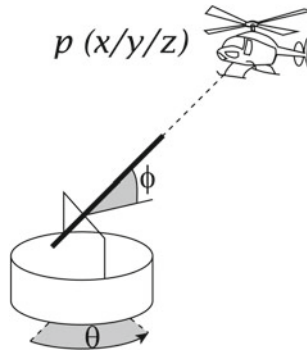


Fig. 3.16 Aerial Gun, tracking a target at $\mathbf{p} = (x/y/z)$

Note that a combination of a horizontal and a vertical rotation of the object in a well-defined sequence uniquely characterizes the direction of the forward direction. With eye movements, this is the line of sight, or gaze direction; with a gun turret on a ship this is the direction of the gun barrel (see Fig. 3.16). However, this does not completely determine the three-dimensional orientation of the object, since the rotation about the forward direction is still unspecified. A third rotation is needed to completely determine the orientation of the object. This third rotation ψ would not affect the direction in which the gun is pointing, it would only rotate around the pointing vector. For a quaternion solution to the targeting problem, see also Sect. 4.5.1.

3.6.2 Two DOF—Projection onto a Flat Surface

Another frequent paradigm is a projection onto a flat surface. Consider the following practical problem (Fig. 3.17).

Two projection systems are mounted at a distance d in front of a flat surface, and should both project a point at P , located on the screen at the location (*hor/ver*), where the positive horizontal direction on the screen is to the right, and the positive vertical direction up. The lower system is mounted like an aerial gun: it can rotate about a vertical axis (θ_N) and swivel about a (rotating) horizontal axis (ϕ_N). The system is mounted *below* cm lower than the center of the screen-based coordinate system, and when $\theta_N = \phi_N = 0$ the projector is pointing straight ahead toward the screen.

The upper projection system is mounted on a disk that hangs on the ceiling, *above* cm higher than the center of the screen-based coordinate system. It is mounted on a horizontal hinge which is parallel to the projection screen, so that it can rotate up and down (ϕ_H). The projector can swivel on the disk left/right (θ_H). And again, when $\theta_H = \phi_H = 0$ the projector is pointing straight ahead toward the screen.

Task: What are the projector angles for the lower projector (θ_N , ϕ_N), and for the upper projector (θ_H , ϕ_H), if both should point at the target $P = (\textit{hor/ver})$ on the screen?

Solution: The rotation sequence for the lower projector corresponds to the two outer rotations of a nautical gimbal, and the sequence of the upper projector to the two outer rotations of a Helmholtz gimbal, respectively (see also Fig. 3.11). In both cases, the rotation about the target direction is unimportant, and ψ in the equations for the nautical- and Helmholtz-rotation matrix can be set to zero. The direction to the target corresponds to the \mathbf{b}_x axis after the rotation, and the target point is the intersection of this axis with a plane parallel to the $\mathbf{s}_y/\mathbf{s}_z$ -plane at a distance d . So for the lower projector, the target is at

$$\mathbf{p} = (d / -\textit{hor/ver} + \textit{below})$$

$$\mathbf{b}'_x = \frac{\mathbf{p}}{|\mathbf{p}|}.$$

(The sign before *hor* is negative, because the positive direction for “horizontal” on the screen is to the right, but the corresponding positive direction for “horizontal” for the projector is to the left.) From that the corresponding nautical angles can be determined with Eq. (3.35) from the previous example with the aerial gun. For the upper projector, the target is at

$$\mathbf{p} = (d / -hor / ver - above)$$

$$\mathbf{b}'_{\mathbf{x}} = \frac{\mathbf{p}}{|\mathbf{p}|}$$

and the projector angles can be found by applying the first two equations of Eq. (3.28):

$$\theta_H = \arcsin\left(\frac{p_y}{\sqrt{p_x^2 + p_y^2 + p_z^2}}\right)$$

$$\phi_H = -\arcsin\left(\frac{p_z}{\sqrt{p_x^2 + p_y^2 + p_z^2}} \cdot \frac{1}{\cos\theta_H}\right).$$
(3.36)

A Python solution, with numbers approximating a setup such as Fig. 3.17, would be

```
from skinematics import vector
(d, hor, ver, above, below) = (1.5, 0.3, 0.2, 0.7, 1.4)
p_lower = [d, -hor, ver+below]
p_upper = [d, -hor, ver-above]
lower_projector = vector.target2orient(p_lower, orient_type='nautical')
upper_projector = vector.target2orient(p_upper, orient_type='Helmholtz')
```

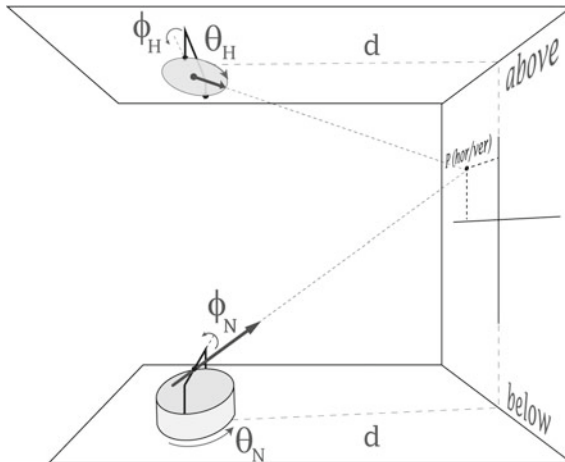


Fig. 3.17 Projection onto a flat surface, with different projection systems

Section 4.5.3 shows the solution of a somewhat more complex, but conceptually similar problem: orienting a camera in a missile such that it is directed on a selected target.

3.6.3 Three DOF—3-D Orientation Measurements with Search Coils

An interpretation of the values of the rotation matrix can be found by looking at Eq. (3.11): the columns of the rotation matrix \mathbf{R} are equivalent to the vectors of the body-fixed coordinate system $[\mathbf{b}_x \mathbf{b}_y \mathbf{b}_z]$ expressed in the space-fixed coordinate system $[\mathbf{s}_x \mathbf{s}_y \mathbf{s}_z]$. Thus, for eye movement measurements with the search-coil method illustrated in Fig. 2.18, different values in the rotation matrix \mathbf{R} indicate a different orientation of the eye-fixed coordinate system, i.e., a different orientation of the eye ball.

Task: What is the orientation of an eye on a gimbal that is rotated 15° to the left and 25° down, if it is (i) a nautical gimbal or (ii) a Helmholtz gimbal?

Solution: If an artificial eye ball on a nautical gimbal (Fig. 3.11a) is first turned 15° to the left and then (about the rotated axis \mathbf{b}_y) 25° down, i.e., $(\theta_N, \phi_N, \psi_N) = (15, 25, 0)$, its orientation will be given by the matrix

$$\mathbf{R}_{nautical} = \begin{bmatrix} 0.88 & -0.26 & 0.41 \\ 0.23 & 0.97 & 0.11 \\ -0.42 & 0 & 0.91 \end{bmatrix}.$$

Putting an eye on a Helmholtz gimbal (Fig. 3.11b), and turning it first 25° down and then 15° to the left (about the rotated axis \mathbf{b}_z), i.e., $(\phi_H, \theta_H, \psi_H) = (25, 15, 0)$, leads to a different orientation of the eye:

$$\mathbf{R}_{Helm} = \begin{bmatrix} 0.88 & -0.23 & 0.42 \\ 0.26 & 0.97 & 0 \\ -0.41 & 0.11 & 0.91 \end{bmatrix}.$$

The orientation of the eye in the two examples is clearly different: on the nautical gimbal \mathbf{b}_z is given by (0.41, 0.11, 0.91), whereas on the Helmholtz gimbal it points in a different direction, (0.42, 0, 0.91).

Interpretation: Experimentally, the three-dimensional orientation of the eye in space can be measured, for example, with induction coils (see Figs. 2.17 and 2.18). When an induction coil is put into an oscillating magnetic field \mathbf{m} , a voltage is induced in the coil (Robinson 1963). If the coil is characterized by a coil vector \mathbf{c} , which is perpendicular to the coil and has a length proportional to the surface surrounded by the coil, the voltage induced is proportional to the cosine of the angle between \mathbf{m} and \mathbf{c} (Fig. 2.16). As pointed out by (Tweed et al. 1990), this leads to a simple correspondence between the values of the rotation matrix and the voltages induced

in search coils. This connection can be demonstrated with the experimental setup shown in Fig. 2.18. Let

$$\mathbf{m}_i = \mathbf{s}_i m_i \sin(\omega_i t), \quad i \in x, y, z$$

be three homogeneous orthogonal magnetic fields. They are parallel to the axes of the space-fixed coordinate system $[\mathbf{s}_x \mathbf{s}_y \mathbf{s}_z]$, have amplitudes m_i , and oscillate at frequencies ω_i . Further, let $[\mathbf{c}_x \mathbf{c}_y \mathbf{c}_z]$ denote three orthogonal coils parallel to the body-fixed coordinate system $[\mathbf{b}_x \mathbf{b}_y \mathbf{b}_z]$ and firmly attached to the object (here the eye). Then, the voltage induced by the magnetic field \mathbf{m}_i in coil \mathbf{c}_j , V_{ij} , is given by

$$V_{ij} = R_{ij} * m_i * \omega_i * \cos(\omega_i * t) * c_j \text{ with } i, j \in x, y, z$$

where $c_j = |\mathbf{c}_j|$ indicates the length of the vector \mathbf{c}_j . This gives a direct interpretation of the elements of the rotation matrix \mathbf{R} : the voltage induced by the magnetic field \mathbf{m}_i in the coil \mathbf{c}_j is proportional to the element R_{ij} of the rotation matrix \mathbf{R} , which describes the rotation from the reference position, where the coils $[\mathbf{c}_x \mathbf{c}_y \mathbf{c}_z]$ line up with the magnetic fields $[\mathbf{m}_x \mathbf{m}_y \mathbf{m}_z]$, to the current position.

Note that for the determination of the 3-D orientation, three matrix elements suffice: H, V, and T indicate the signals that represent approximately the horizontal, vertical, and torsional orientation of the object:

$$\mathbf{R} = \begin{bmatrix} - & - & - \\ H & - & - \\ V & T & - \end{bmatrix}.$$

3.6.4 Nested or Cascaded 3-D Rotation Sequences

Nested or cascaded sequences are commonly found where one 3-D rotation follows another 3-D rotation. For example, the moving eye is placed inside a moving head, or a gimbaled camera is mounted on a moving vehicle or missile. How can the formulas given above be used to derive the composite rotation of the two 3-D rotations combined? Furthermore, given the location of a point in space-fixed coordinates and eye coordinates, would it be possible to calculate the required rotation to ensure that the eye/camera looks at the point?

Camera on Moving Base

To describe the orientation of a camera-in-space (described by $\mathbf{R}_{camera}^{space}$), as shown in Fig. 3.18, one has to combine the orientation of the tilted base, e.g., a Google maps car (described by $\mathbf{R}_{base}^{space}$) and the orientation of the camera with respect to this base ($\mathbf{R}_{camera}^{base}$). To implement this mathematically, one has to use rotations of the coordinate system. This determines the sequence of the two rotations, and the rotation matrix describing the orientation of the camera-in-space is—according to the

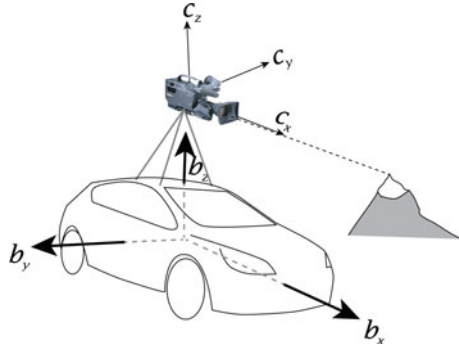


Fig. 3.18 A nested or cascaded rotation can, for example, be the orientation of a camera $[c_x c_y c_z]$ on a moving base $[b_x b_y b_z]$ fixating a target, here the top of a mountain

discussion following Eqs. (3.20) and (3.22)-given by

$$\mathbf{R}_{camera}^{space} = \mathbf{R}_{base}^{space} \cdot \mathbf{R}_{camera}^{base} \tag{3.37}$$

From this, the orientation of the camera with respect to the base can be determined as

$$\mathbf{R}_{camera}^{base} = (\mathbf{R}_{base}^{space})^{-1} \cdot \mathbf{R}_{camera}^{space} \tag{3.38}$$

The way I personally remember these sequences: Take, for example, the following equation, which determines the orientation of the line of sight (LOS) of the camera

$$\mathbf{c}' = \mathbf{R}_{base}^{space} \cdot (\mathbf{R}_{camera}^{base} \cdot \mathbf{c}) \tag{3.39}$$

With base and camera in the reference orientation, \mathbf{c} indicates the line of sight (LOS) of the camera. To find the current LOS, I first rotate the LOS of the camera on the base $(\mathbf{R}_{camera}^{base} \cdot \mathbf{c})$. Then, in the second step I rotate the base, with the rotated camera already on it, to obtain the current orientation of the LOS: $\mathbf{R}_{base}^{space} \cdot (\mathbf{R}_{camera}^{base} \cdot \mathbf{c})$.

Eye in Head

Similarly, let $\mathbf{R}_{head}^{space}$ be the rotation matrix describing the rotation of the reference frame with respect to a space-fixed coordinate system, and \mathbf{R}_{eye}^{head} describe the rotation of the object in the reference frame (e.g., eye in head). Then

$$\mathbf{R}_{eye}^{space} = \mathbf{R}_{head}^{space} \cdot \mathbf{R}_{eye}^{head} \tag{3.40}$$

and

$$\mathbf{R}_{eye}^{head} = (\mathbf{R}_{head}^{space})^{-1} \cdot \mathbf{R}_{eye}^{space} \tag{3.41}$$

3.6.5 Camera Images

Imagine the gaze vector of two cameras mounted on two different-sequenced two-axis gimbals following the same point in space. As the point moves around, the gaze vector of the two cameras traces the same locus in space. However, each camera will view differently rolled images of the world (i.e., the image “up” vector in Camera 1 will be at different angles than in Camera 2). The different gimbal’s camera images will be rolled with respect to each other. The gimbaled cameras require all two degrees of freedom to follow the target and have no freedom to orientate their images to the upright orientation. If three-axis gimbals are used, there are no remaining degrees of freedom, and the images can always be rotated such that the required direction is pointing “up”.

3.7 Exercises

Exercise 3.1: CT Scanner

A good example of a device that requires 3-D kinematics is a modern CT scanner.

Task: For the exercise, label the rotations about the three axes with α , β , and γ as shown in Fig. 3.19. With the CT scanner in the starting orientation, the outermost axis aligns with the z-axis, the middle axis with x, and the inner axis with y.

A patient has been attacked on the streets and has suffered from two gunshot wounds. When the patient is in a supine orientation (as indicated in Fig. 3.19), the first shot went through the left eye socket, into the direction $\vec{bullet}_1 = [5 \ 2 \ 2]$; the second shot also went through the left eye socket, but in the direction $\vec{bullet}_2 = [5 \ -2 \ -4]$.

Find out which settings for α , β , and γ bring the scanner into such an orientation that (i) \vec{bullet}_1 in the image is oriented along the y-axis of the scan and (ii) the trajectories of both bullets lie in the scanned plane (Fig. 3.20).

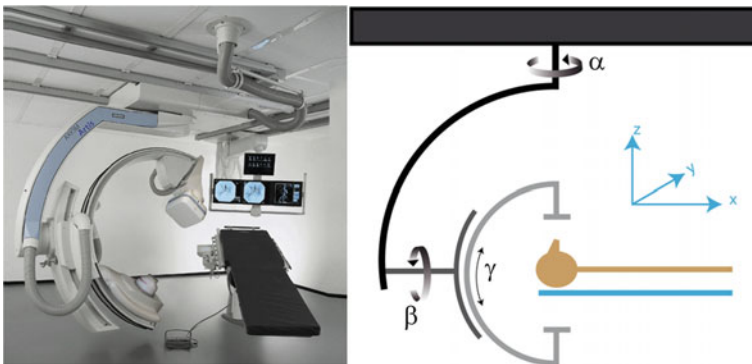


Fig. 3.19 (Left) An image of the Siemens Axiom Artis dTC scanner. (Right) Cartoon indicating the mathematical representation of this scanner

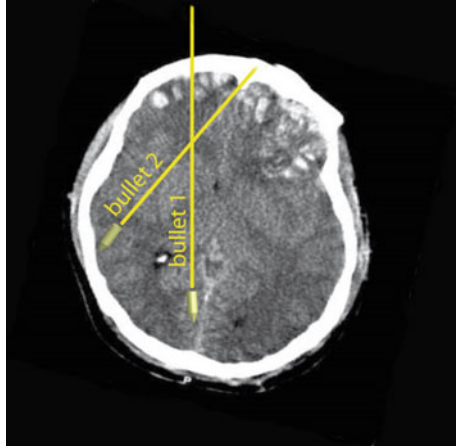


Fig. 3.20 CT scan of the head, with the bullet paths from two gunshot wounds schematically indicated. Note that the path of bullet 1 aligns with the y-axis

Exercise 3.2: Combining Rotation and Translation

This example will provide the first step in the measurement of movements with video-based systems. In the example, the movement of a comet that moves in a plane in space is observed with a camera, as shown in Fig. 3.22. The object is a comet that circles around a planet. The data units are 10^7 km. The tasks for this exercise are as follows:

- Read in the data from the file `planet_trajectory_2D.txt`, and write a program to calculate the planet velocity in the x- and y-directions. The data can be found in the *scikit-kinematics* package for Python users, and in the *Kinematics* toolbox for Matlab users, and are shown in Fig. 3.21.

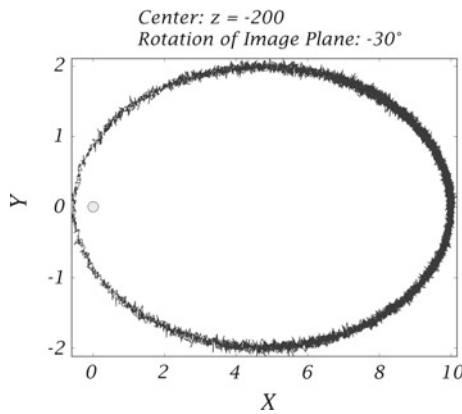


Fig. 3.21 Observed trajectory

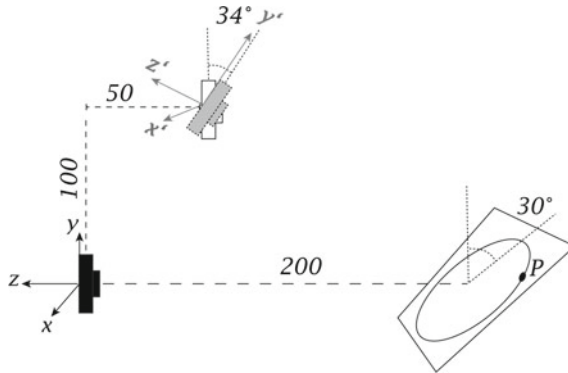



Fig. 3.22 Experimental setup: a particle moving in a tilted plane is observed from two different camera positions orientations

- External information is provided, which tells us that the data center is 200×10^7 km in front of the camera, and the trajectory lies in a plane that is tilted by 30° about the x-axis. Calculate the 3-D position of the trajectory of the particle (see Fig. 3.22).
- Calculate the 3-D position p_{shifted} of the comet in camera coordinates, observed from a satellite which has traveled 50×10^7 km toward the planet, and 100×10^7 km orthogonally to it.
- Calculate the position p_{shiftRot} (with respect to the camera), if the satellite is rotated 34° downward.

 **python**™ **Code:** [C3_examples_rotmat.py](#): Python examples of different operations with rotation matrices, such as generating symbolic and numeric rotation matrices and calculating the corresponding rotation sequences. (p. 141)