Forming ${}^{A}_{D}R$ from its three constituents requires 54 multiplications and 36 additions. Performing the final matrix-vector multiplication of (2.96) requires an additional 9 multiplications and 6 additions, bringing the totals to 63 multiplications and 42 additions.

If, instead, we transform the vector through the matrices one at a time, that is,

$$\begin{aligned} {}^{A}P &= {}^{A}_{B}R\, {}^{B}_{C}R\, {}^{C}_{D}R\, {}^{D}P \\ {}^{A}P &= {}^{A}_{B}R\, {}^{B}_{C}R\, {}^{C}P \\ {}^{A}P &= {}^{A}_{B}R\, {}^{B}P \\ {}^{A}P &= {}^{A}P, \end{aligned} \tag{2.97}$$

then the total computation requires only 27 multiplications and 18 additions, fewer than half the computations required by the other method.

Of course, in some cases, the relationships ${}^{A}_{B}R$, ${}^{B}_{C}R$, and ${}^{C}_{D}R$ are constant, while there are many ${}^{D}P_i$ that need to be transformed into ${}^{A}P_i$. In such a case, it is more efficient to calculate ${}^{A}_{D}R$ once, and then use it for all future mappings (see also Exercise 2.16).

EXAMPLE 2.10

Give a method of computing the product of two rotation matrices, ${}^{A}_{B}R\, {}^{B}_{C}R$, that uses fewer than 27 multiplications and 18 additions.

Where $\hat{L}_i$ are the columns of ${}^{B}_{C}R$ and $\hat{C}_i$ are the three columns of the result, compute

$$\begin{aligned} \hat{C}_1 &= {}^{A}_{B}R\hat{L}_i, \\ \hat{C}_2 &= {}^{A}_{B}R\hat{L}_2, \\ \hat{C}_3 &= \hat{C}_1 \times \hat{C}_2, \end{aligned} \tag{2.98}$$

which requires 24 multiplications and 15 additions.

BIBLIOGRAPHY

[1] B. Noble, *Applied Linear Algebra*, Prentice-Hall, Englewood Cliffs, NJ, 1969.

[2] D. Ballard and C. Brown, *Computer Vision*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[3] O. Bottema and B. Roth, *Theoretical Kinematics*, North Holland, Amsterdam, 1979.

[4] R.P. Paul, *Robot Manipulators*, MIT Press, Cambridge, MA, 1981.

[5] I. Shames, *Engineering Mechanics*, 2nd edition, Prentice-Hall, Englewood Cliffs, NJ, 1967.

[6] Symon, *Mechanics*, 3rd edition, Addison-Wesley, Reading, MA, 1971.

[7] B. Gorla and M. Renaud, *Robots Manipulateurs*, Cepadues-Editions, Toulouse, 1984.

EXERCISES

2.1 [15] A vector ${}^{A}P$ is rotated about $\hat{Z}_A$ by θ degrees and is subsequently rotated about $\hat{X}_A$ by ϕ degrees. Give the rotation matrix that accomplishes these rotations in the given order.

2.2 [15] A vector ${}^AP$ is rotated about $\hat{Y}_A$ by 30 degrees and is subsequently rotated about $\hat{X}_A$ by 45 degrees. Give the rotation matrix that accomplishes these rotations in the given order.

2.3 [16] A frame $\{B\}$ is located initially coincident with a frame $\{A\}$. We rotate $\{B\}$ about $\hat{Z}_B$ by θ degrees, then we rotate the resulting frame about $\hat{X}_B$ by ϕ degrees. Give the rotation matrix that will change the descriptions of vectors from ${}^BP$ to ${}^AP$.

2.4 [16] A frame $\{B\}$ is located initially coincident with a frame $\{A\}$. We rotate $\{B\}$ about $\hat{Z}_B$ by 30 degrees, then we rotate the resulting frame about $\hat{X}_B$ by 45 degrees. Give the rotation matrix that will change the description of vectors from ${}^BP$ to ${}^AP$.

2.5 [13] ${}^A_BR$ is a 3×3 matrix with eigenvalues 1, e^{+ai}, and e^{-ai}, where $i = \sqrt{-1}$. What is the physical meaning of the eigenvector of ${}^A_BR$ associated with the eigenvalue 1?

2.6 [21] Derive equation (2.80).

2.7 [24] Describe (or program) an algorithm that extracts the equivalent angle and axis of a rotation matrix. Equation (2.82) is a good start, but make sure that your algorithm handles the special cases $\theta = 0°$ and $\theta = 180°$.

2.8 [29] Write a subroutine that changes representation of orientation from rotation-matrix form to equivalent angle–axis form. A Pascal-style procedure declaration would begin

```
Procedure RMTOAA (VAR R:mat33; VAR K:vec3; VAR theta: real);
```

Write another subroutine that changes from equivalent angle–axis representation to rotation-matrix representation:

```
Procedure AATORM(VAR K:vec3; VAR theta: real: VAR R:mat33);
```

Write the routines in C if you prefer. Run these procedures on several cases of test data back-to-back and verify that you get back what you put in. Include some of the difficult cases!

2.9 [27] Do Exercise 2.8 for roll, pitch, yaw angles about fixed axes.

2.10 [27] Do Exercise 2.8 for Z–Y–Z Euler angles.

2.11 [10] Under what condition do two rotation matrices representing finite rotations commute? A proof is not required.

2.12 [14] A velocity vector is given by

$$ {}^BV = \begin{bmatrix} 10.0 \\ 20.0 \\ 30.0 \end{bmatrix}. $$

Given

$$ {}^A_BT = \begin{bmatrix} 0.866 & -0.500 & 0.000 & 11.0 \\ 0.500 & 0.866 & 0.000 & -3.0 \\ 0.000 & 0.000 & 1.000 & 9.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, $$

compute ${}^AV$.

2.13 [21] The following frame definitions are given as known:

$$ {}^U_AT = \begin{bmatrix} 0.866 & -0.500 & 0.000 & 11.0 \\ 0.500 & 0.866 & 0.000 & -1.0 \\ 0.000 & 0.000 & 1.000 & 8.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, $$

$$ {}^{B}_{A}T = \begin{bmatrix} 1.000 & 0.000 & 0.000 & 0.0 \\ 0.000 & 0.866 & -0.500 & 10.0 \\ 0.000 & 0.500 & 0.866 & -20.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, $$

$$ {}^{C}_{U}T = \begin{bmatrix} 0.866 & -0.500 & 0.000 & -3.0 \\ 0.433 & 0.750 & -0.500 & -3.0 \\ 0.250 & 0.433 & 0.866 & 3.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. $$

Draw a frame diagram (like that of Fig. 2.15) to show their arrangement qualitatively, and solve for ${}^{B}_{C}T$.

2.14 [31] Develop a general formula to obtain ${}^{A}_{B}T$, where, starting from initial coincidence, $\{B\}$ is rotated by θ about $\hat{K}$ where $\hat{K}$ passes through the point ${}^{A}P$ (not through the origin of $\{A\}$ in general).

2.15 [34] $\{A\}$ and $\{B\}$ are frames differing only in orientation. $\{B\}$ is attained as follows: starting coincident with $\{A\}$, $\{B\}$ is rotated by θ radians about unit vector $\hat{K}$—that is,

$$ {}^{A}_{B}R = {}^{A}_{B}R_{K}(\theta). $$

Show that

$$ {}^{A}_{B}R = e^{k\theta}, $$

where

$$ K = \begin{bmatrix} 0 & -k_x & ky \\ k_z & 0 & -k_x \\ -k_y & k_x & 0 \end{bmatrix}. $$

2.16 [22] A vector must be mapped through three rotation matrices:

$$ {}^{A}P = {}^{A}_{B}R\ {}^{B}_{C}R\ {}^{C}_{D}R\ {}^{D}P. $$

One choice is to first multiply the three rotation matrices together, to form ${}^{A}_{D}R$ in the expression

$$ {}^{A}P = {}^{A}_{D}R\ {}^{D}P. $$

Another choice is to transform the vector through the matrices one at a time—that is,

$$ {}^{A}P = {}^{A}_{B}R\ {}^{B}_{C}R\ {}^{C}_{D}R\ {}^{D}P, $$

$$ {}^{A}P = {}^{A}_{B}R\ {}^{B}_{C}R\ {}^{C}P, $$

$$ {}^{A}P = {}^{A}_{B}R\ {}^{B}P, $$

$$ {}^{A}P = {}^{A}P. $$

If ${}^{D}P$ is changing at 100 Hz, we would have to recalculate ${}^{A}P$ at the same rate. However, the three rotation matrices are also changing, as reported by a vision system that gives us new values for ${}^{A}_{B}R$, ${}^{B}_{C}R$, and ${}^{C}_{D}R$ at 30 Hz. What is the best way to organize the computation to minimize the calculation effort (multiplications and additions)?

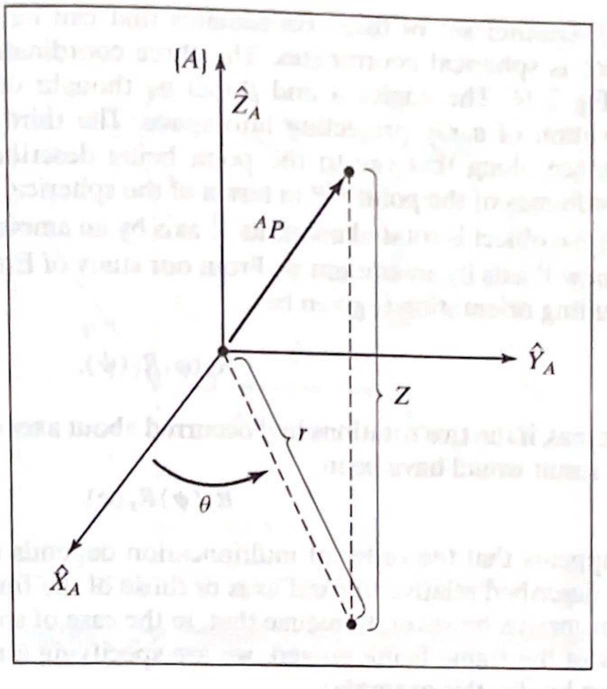

FIGURE 2.23: Cylindrical coordinates.

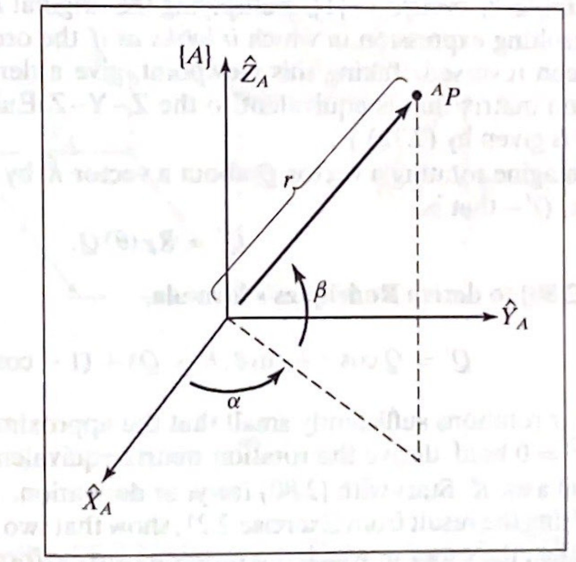

FIGURE 2.24: Spherical coordinates.

2.17 [16] Another familiar set of three coordinates that can be used to describe a point in space is cylindrical coordinates. The three coordinates are defined as illustrated in Fig. 2.23. The coordinate θ gives a direction in the xy plane along which to translate radially by an amount r. Finally, z is given to specify the height above the xy plane. Compute the Cartesian coordinates of the point ^AP in terms of the cylindrical coordinates θ, r, and z.

2.18 [18] Another set of three coordinates that can be used to describe a point in space is spherical coordinates. The three coordinates are defined as illustrated in Fig. 2.24. The angles α and β can be thought of as describing azimuth and elevation of a ray projecting into space. The third coordinate, r, is the radial distance along that ray to the point being described. Calculate the Cartesian coordinates of the point ${}^{A}P$ in terms of the spherical coordinates α, β, and r.

2.19 [24] An object is rotated about its $\hat{X}$ axis by an amount ϕ, then it is rotated about its new $\hat{Y}$ axis by an amount ψ. From our study of Euler angles, we know that the resulting orientation is given by

$$R_x(\phi)R_y(\psi),$$

whereas, if the two rotations had occurred about axes of the fixed reference frame, the result would have been

$$R_y(\psi)R_x(\phi).$$

It appears that the order of multiplication depends upon whether the rotations are described relative to fixed axes or those of the frame being moved. It is more appropriate, however, to realize that, in the case of specifying a rotation about an axis of the frame being moved, we are specifying a rotation in the fixed system given by (for this example)

$$R_x(\phi)R_y(\psi)R_x^{-1}(\phi).$$

This *similarity transform* [1], multiplying the original $R_x(\phi)$ on the left, reduces to the resulting expression in which *it looks as if* the order of matrix multiplication has been reversed. Taking this viewpoint, give a derivation for the form of the rotation matrix that is equivalent to the Z–Y–Z Euler angle set (α, β, γ). (The result is given by (2.72).)

2.20 [20] Imagine rotating a vector Q about a vector $\hat{K}$ by an amount θ to form a new vector, Q'—that is,

$$Q' = R_K(\theta)Q.$$

Use (2.80) to derive **Rodriques's formula**,

$$Q' = Q\cos\theta + \sin\theta(\hat{K} \times Q) + (1 - \cos\theta)(\hat{K} \cdot Q)\hat{K}.$$

2.21 [15] For rotations sufficiently small that the approximations $\sin\theta = \theta$, $\cos\theta = 1$, and $\theta^2 = 0$ hold, derive the rotation-matrix equivalent to a rotation of θ about a general axis, $\hat{K}$. Start with (2.80) for your derivation.

2.22 [20] Using the result from Exercise 2.21, show that two infinitesimal rotations commute (i.e., the order in which the rotations are performed is not important).

2.23 [25] Give an algorithm to construct the definition of a frame ${}^{U}_{A}T$ from three points ${}^{U}P_1$, ${}^{U}P_2$, and ${}^{U}P_3$, where the following is known about these points:

1. ${}^{U}P_1$ is at the origin of $\{A\}$;
2. ${}^{U}P_2$ lies somewhere on the positive $\hat{X}$ axis of $\{A\}$;
3. ${}^{U}P_3$ lies near the positive $\hat{Y}$ axis in the XY plane of $\{A\}$.

2.24 [45] Prove Cayley's formula for proper orthonormal matrices.

2.25 [30] Show that the eigenvalues of a rotation matrix are 1, $e^{\alpha i}$, and $e^{-\alpha i}$, where $i = \sqrt{-1}$.

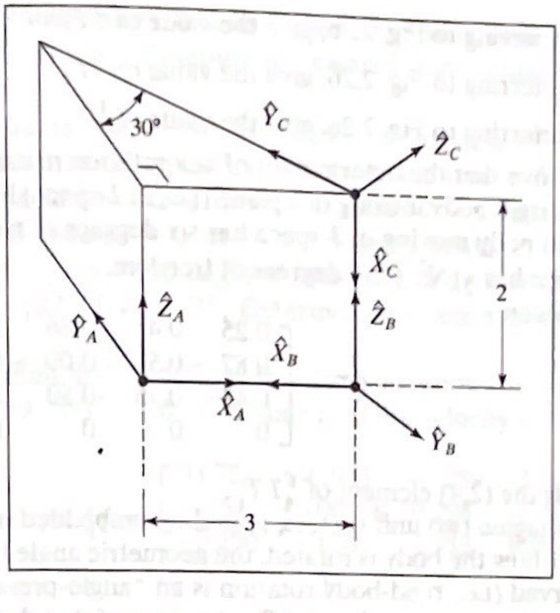

FIGURE 2.25: Frames at the corners of a wedge.

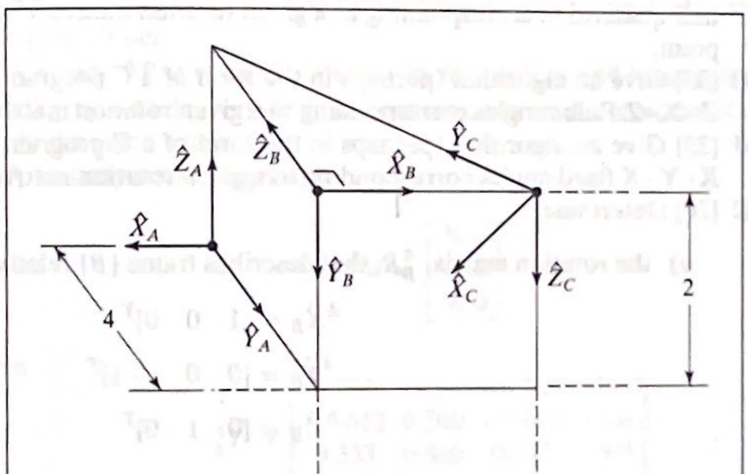

FIGURE 2.26: Frames at the corners of a wedge.

2.26 [33] Prove that any Euler angle set is sufficient to express all possible rotation matrices.

2.27 [15] Referring to Fig. 2.25, give the value of ${}^A_BT$.

2.28 [15] Referring to Fig. 2.25, give the value of ${}^A_CT$.

2.29 [15] Referring to Fig. 2.25, give the value of ${}^B_CT$.

2.30 [15] Referring to Fig. 2.25, give the value of ${}^C_AT$.

2.31 [15] Referring to Fig. 2.26, give the value of ${}^A_BT$.

2.32 [15] Referring to Fig. 2.26, give the value of ${}^A_CT$.

2.33 [15] Referring to Fig. 2.26, give the value of ${}^B_CT$.

2.34 [15] Referring to Fig. 2.26, give the value of ${}^C_AT$.

2.35 [20] Prove that the determinant of any rotation matrix is always equal to 1.

2.36 [36] A rigid body moving in a plane (i.e., in 2-space) has three degrees of freedom. A rigid body moving in 3-space has six degrees of freedom. Show that a body in N-space has $\frac{1}{2}(N^2 + N)$ degrees of freedom.

2.37 [15] Given

$$
{}^A_BT = \begin{bmatrix} 0.25 & 0.43 & 0.86 & 5.0 \\ 0.87 & -0.50 & 0.00 & -4.0 \\ 0.43 & 0.75 & -0.50 & 3.0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

what is the (2,4) element of ${}^B_AT$?

2.38 [25] Imagine two unit vectors, v_1 and v_2, embedded in a rigid body. Note that, no matter how the body is rotated, the geometric angle between these two vectors is preserved (i.e., rigid-body rotation is an "angle-preserving" operation). Use this fact to give a concise (four- or five-line) proof that the inverse of a rotation matrix must equal its transpose, and that a rotation matrix is orthonormal.

2.39 [37] Give an algorithm (perhaps in the form of a C program) that computes the unit quaternion corresponding to a given rotation matrix. Use (2.91) as a starting point.

2.40 [33] Give an algorithm (perhaps in the form of a C program) that computes the Z–X–Z Euler angles corresponding to a given rotation matrix (see Appendix B).

2.41 [33] Give an algorithm (perhaps in the form of a C program) that computes the X–Y–X fixed angles corresponding to a given rotation matrix (see Appendix B).

2.42 [20] Determine

a) the rotation matrix, ${}^A_BR$, that describes frame $\{B\}$ relative to $\{A\}$ if

$$
{}^A\hat{X}_B = [1 \quad 0 \quad 0]^T
$$
$$
{}^A\hat{Y}_B = [0 \quad 0 \quad -1]^T
$$
$$
{}^A\hat{Z}_B = [0 \quad 1 \quad 0]^T
$$

b) the Z–Y–Z Euler angles required for such a rotation.

2.43 [15] Calculate the rotation matrix, ${}^A_BR$, if frames $\{A\}$ and $\{B\}$ are originally coincident then

a) frame $\{B\}$ is rotated about $\hat{X}_A$ by 30 degrees, then about $\hat{Y}_A$ by 15 degrees and, finally, about $\hat{Z}_A$ by 70 degrees.

b) frame $\{B\}$ is rotated about $\hat{Z}_B$ by 70 degrees, then about $\hat{Y}_B$ by 15 degrees and, finally, about $\hat{Y}_B$ by 70 degrees.

2.44 [20] A coordinate frame, $\{B\}$, is located at the base of a robot manipulator. $\hat{Z}_B$ points upward. Three cameras are used to view the manipulator. Coordinate frames $\{C\}$, $\{D\}$, and $\{E\}$ describe the camera positions and orientations. The origin of $\{C\}$ is on the $\hat{X}_B - \hat{Z}_B$ plane. The cameras are mounted on tripods 1.5 units tall placed at the vertices of an equilateral triangle having $\{B\}$ at the incenter. The robot frame is on the focal axis of each camera (the camera's $\hat{Z}$-axis), and the

Euclidean distance from $\{B\}$ to any camera is 5 units, thus ${}^{C}P_{BORG} = {}^{D}P_{BORG} = {}^{E}P_{BORG} = [0 \quad 0 \quad 5]^T$. Compute the camera transformation matrices: ${}^{B}_{C}T$, ${}^{B}_{D}T$, and ${}^{B}_{E}T$.

2.45 [20] A coordinate frame, $\{B\}$, is located at the base of a robot manipulator. Frame $\{C\}$ describes the position and orientation of a depth camera that was originally coincident with $\{B\}$ then translated 7 units in $\hat{X}_B$, translated -2 units in $\hat{Y}_B$, translated 5 units in $\hat{Z}_B$, rotated about $\hat{Z}_C$ by -20 degrees, and rotated about $\hat{Y}_C$ by -110 degrees. The camera detects an object having coordinates ${}^{C}P = [0.5 \quad 0.2 \quad 3.2]^T$. Determine the object coordinates in frame $\{B\}$, that is, ${}^{B}P$.

2.46 [20] The position and velocity of an object are known to be ${}^{B}P_0 = [0 \quad 0.5 \quad 0]^T$ and ${}^{B}V_0 = [1.9 \quad 0.1 \quad -0.3]^T$ at time t_0. If the velocity is constant and

$$
{}^{A}_{B}T = \begin{bmatrix} 0.0722 & -0.963 & -0.259 & -5.00 \\ 0.954 & -0.00868 & 0.298 & -6.50 \\ -0.290 & -0.269 & 0.919 & 8.00 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

then what is ${}^{A}P$ after 5 units of time?

2.47 [15] A vector ${}^{A}P$ is rotated about $\hat{Z}_A$ by θ degrees, and is subsequently rotated about $\hat{Y}_A$ by ϕ degrees. Give the rotation matrix that accomplishes these rotations in the given order.

2.48 [15] A vector ${}^{A}P$ is rotated about $\hat{Y}_A$ by 60 degrees, and is subsequently rotated about $\hat{X}_A$ by -45 degrees. Give the rotation matrix that accomplishes these rotations in the given order.

2.49 [14] A velocity vector is given by

$$
{}^{B}V = \begin{bmatrix} 30.0 \\ 40.0 \\ 50.0 \end{bmatrix}.
$$

Given

$$
{}^{A}_{B}T = \begin{bmatrix} 0.707 & 0 & -0.707 & 11.0 \\ -0.612 & 0.500 & -0.612 & -3.0 \\ 0.353 & 0.866 & 0.353 & -9.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

compute ${}^{A}V$.

2.50 [15] Referring to Fig. 2.26, give the value of ${}^{C}_{B}T$.

2.51 [15] Given

$$
{}^{A}_{B}T = \begin{bmatrix} 0.25 & 0.43 & 0.86 & 5.0 \\ 0.87 & -0.50 & 0.00 & -4.0 \\ 0.43 & 0.75 & -0.50 & 3.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

what is ${}^{B}P_{AORG}$?

PROGRAMMING EXERCISE (PART 2)

1. If your function library does not include an Atan2 function subroutine, write one.
2. To make a friendly user interface, we wish to describe orientations in the planar world by a single angle, θ, instead of by a 2×2 rotation matrix. The user will always