CHAPTER 2

Kinematics of Robots: Position Analysis

2.1 Introduction

In this chapter, we will study forward and inverse kinematics of robots. With forward kinematic equations, we can determine where the robot's end (hand) will be if all joint variables are known. Inverse kinematics enables us to calculate what each joint variable must be in order to locate the hand at a particular point and a particular orientation. Using matrices, we will first establish a method of describing objects, locations, orientations, and movements. Then we will study the forward and inverse kinematics of different robot configurations such as Cartesian, cylindrical, and spherical coordinates. Finally, we will use the Denavit–Hartenberg representation to derive forward and inverse kinematic equations of all possible configurations of robots—regardless of number of joints, order of joints, and presence (or lack) of offsets and twists.

It is important to realize that in practice, manipulator-type robots are delivered with no end effector. In most cases, there may be a gripper attached to the robot; however, depending on the actual application, different end effectors are attached to the robot by the user. Obviously, the end effector's size and length determine where the end of the robot will be. For a short end effector, the end will be at a different location compared to a long end effector. In this chapter, we will assume that the end of the robot is a plate to which the end effector can be attached, as necessary. We will call this the "hand" or the "end plate" of the robot. If necessary, we can always add the length of the end effector to the robot for determining the location and orientation of the end effector. It should be mentioned here that a real robot manipulator, for which the length of the end effector is not defined, will calculate its joint values based on the end plate location and orientation, which may be different from the position and orientation perceived by the user.

2.2 Robots as Mechanisms

Manipulator-type robots are multi-degree-of-freedom (DOF), three-dimensional, open loop, chain mechanisms, and are discussed in this section.

Multi-degree-of-freedom means that robots possess many joints, allowing them to move freely within their envelope. In a 1-DOF system, when the variable is set to a particular value, the mechanism is totally set and all its other variables are known. For example, in the 1-DOF 4-bar mechanism of Figure 2.1, when the crank is set to 120°, the angles of the coupler link and the rocker arm are also known, whereas in a multi-DOF mechanism, all input variables must be individually defined in order to know the remaining parameters. Robots are multi-DOF machines, where each joint variable must be known in order to determine the location of the robot's hand.

Robots are three-dimensional machines if they are to move in space. Although it is possible to have a two-dimensional multi-DOF robot, they are not common (or useful).

Robots are open-loop mechanisms. Unlike mechanisms that are closed-loop (e.g., 4-bar mechanisms), even if all joint variables are set to particular values, there is no guarantee that the hand will be at the given location. This is because deflections in any joint or link will change the location of all subsequent links without feedback. For example, in the 4-bar mechanism of Figure 2.2, when link AB deflects as a result of load *F*, link BO_2 will also move; therefore, the deflection can be detected. In an open-loop system such as the robot, the deflections will move all succeeding members without any

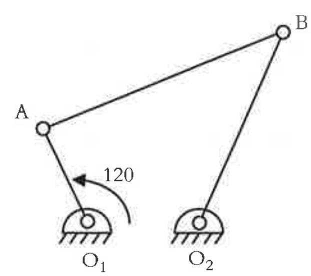


Figure 2.1 A 1-DOF closed-loop 4-bar mechanism.

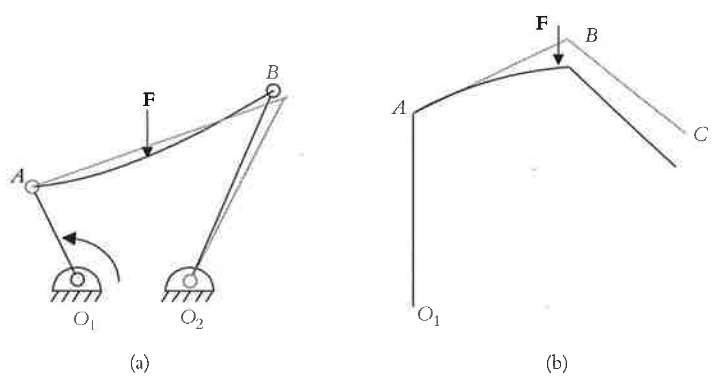


Figure 2.2 Closed-loop (a) versus open-loop (b) mechanisms.

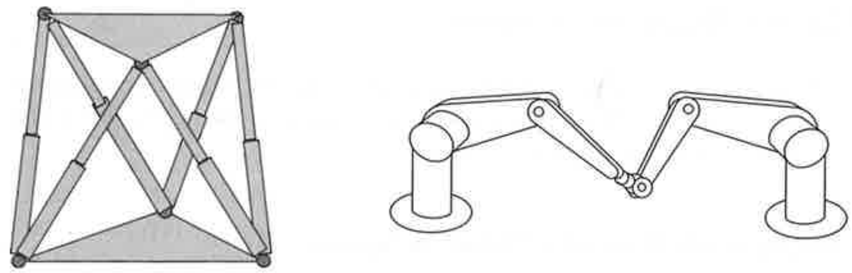


Figure 2.3 Possible parallel manipulator configurations.

feedback. Therefore, in open-loop systems, either all joint and link parameters must continuously be measured, or the end of the system must be monitored; otherwise, the kinematic position of the machine is not completely known. This difference can be expressed by comparing the vector equations describing the relationship between different links of the two mechanisms as follows:

$$\text{For the 4-bar mechanism:} \quad \overline{O_1A} + \overline{AB} = \overline{O_1O_2} + \overline{O_2B} \tag{2.1}$$

$$\text{For the robot:} \quad \overline{O_1A} + \overline{AB} + \overline{BC} = \overline{O_1C} \tag{2.2}$$

As you can see, if there is a deflection in link AB, link O_2B will move accordingly. However, the two sides of Equation (2.1) have changed corresponding to the changes in the links. On the other hand, if link AB of the robot deflects, all subsequent links will move too; however, unless O_1C is measured by other means, the change will not be known. To remedy this problem in open loop robots, either the position of the hand is constantly measured with devices such as a camera, the robot is made into a closed loop system with external means such as the use of secondary arms or laser beams,[1,2,3] or as standard practice, the robot links and joints are made excessively strong to eliminate all deflections. This will render the robot very heavy, massive, and slow, and its specified payload will be very low compared to what it can actually carry.

Alternatives, also called *parallel manipulators*, are based on closed-loop parallel architecture (Figure 2.3). The tradeoff is much-reduced range of motions and workspace.

2.3 Conventions

Throughout this book, we will use the following conventions for describing vectors, frames, transformations, and so on:

Vectors	**i, j, k, x, y, z, n, o, a, p**
Vector components	n_x, n_y, n_z, a_x, a_y, a_z
Frames	F_{xyz}, F_{noa}, xyz, noa, F_{camera}
Transformations	T_1, T_2, $^{''}T$, ^{B}P, $^{U}T_R$ (transformation of robot relative to the Universe, where Universe is a fixed frame)

2.4 Matrix Representation

Matrices can be used to represent points, vectors, frames, translations, rotations, transformations, as well as objects and other kinematic elements. We will use this representation throughout the book.

2.4.1 Representation of a Point in Space

A point P in space (Figure 2.4) can be represented by its three coordinates relative to a reference frame as:

$$P = a_x\mathbf{i} + b_y\mathbf{j} + c_z\mathbf{k} \tag{2.3}$$

where a_x, b_y, and c_z are the three coordinates of the point represented in the reference frame. Obviously, other coordinate representations can also be used to describe the location of a point in space.

2.4.2 Representation of a Vector in Space

A vector can be represented by three coordinates of its tail and its head. If the vector starts at point A and ends at point B, then it can be represented by $\mathbf{P}_{AB} = (B_x - A_x)\mathbf{i} + (B_y - A_y)\mathbf{j} + (B_z - A_z)\mathbf{k}$. Specifically, if the vector starts at the origin (Figure 2.5),

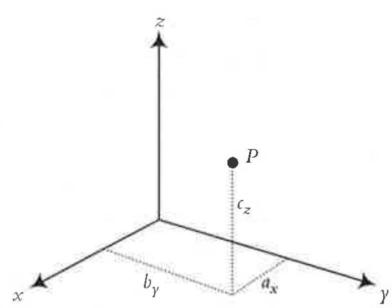


Figure 2.4 Representation of a point in space.

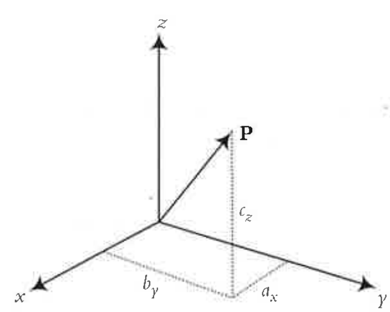


Figure 2.5 Representation of a vector in space.

then:

$$\mathbf{P} = a_x\mathbf{i} + b_y\mathbf{j} + c_z\mathbf{k} \tag{2.4}$$

where a_x, b_y, and c_z are the three components of the vector in the reference frame. In fact, point P in the previous section is in reality represented by a vector connected to it at point P and expressed by the three components of the vector.

The three components of the vector can also be written in matrix form, as in Equation (2.5). This format will be used throughout this book to represent all kinematic elements:

$$\mathbf{P} = \begin{bmatrix} a_x \\ b_y \\ c_z \end{bmatrix} \tag{2.5}$$

This representation can be slightly modified to also include a scale factor w such that if P_x, P_y, and P_z are divided by w, they will yield a_x, b_y, and c_z. Therefore the vector can be written as:

$$\mathbf{P} = \begin{bmatrix} P_x \\ P_y \\ P_z \\ w \end{bmatrix} \quad \text{where } a_x = \frac{P_x}{w},\ b_y = \frac{P_y}{w}, \text{ etc.} \tag{2.6}$$

w may be any number and, as it changes, it can change the overall size of the vector. This is similar to the zooming function in computer graphics. As the value of w changes, the size of the vector changes accordingly. If w is bigger than 1, all vector components enlarge; if w is smaller than 1, all vector components become smaller.

When w is 1, the size of these components remains unchanged. However, if $w = 0$, then a_x, b_y, and c_z will be infinity. In this case, P_x, P_y, and P_z (as well as a_x, b_y, and c_z) will represent a vector whose length is infinite but nonetheless is in the direction represented by the vector. This means that a *direction vector* can be represented by a scale factor of $w = 0$, where the length is not important, but the direction is represented by the three components of the vector. This will be used throughout the book to represent direction vectors.

In computer graphics applications, the addition of a scale factor allows the user to zoom in or out simply by changing this value. Since the scale factor increases or decreases all vector dimensions accordingly, the size of a vector (or drawing) can be easily changed without the need to redraw it. However, our reason for this inclusion is different, and it will become apparent shortly.

Example 2.1

A vector is described as $\mathbf{P} = 3\mathbf{i} + 5\mathbf{j} + 2\mathbf{k}$. Express the vector in matrix form:

(a) With a scale factor of 2.

(b) If it were to describe a direction as a unit vector.

Solution: The vector can be expressed in matrix form with a scale factor of 2 as well as 0 for direction as:

$$\mathbf{P} = \begin{bmatrix} 6 \\ 10 \\ 4 \\ 2 \end{bmatrix} \quad \text{and} \quad \mathbf{P} = \begin{bmatrix} 3 \\ 5 \\ 2 \\ 0 \end{bmatrix}$$

However, in order to make the vector into a unit vector, we normalize the length to be equal to 1. To do this, each component of the vector is divided by the square root of the sum of the squares of the three components:

$$\lambda = \sqrt{P_x^2 + P_y^2 + P_z^2} = 6.16 \text{ and } P_x = {}^3/_{6.16} = 0.487, \text{ etc. Therefore,}$$

$$\mathbf{P}_{unit} = \begin{bmatrix} 0.487 \\ 0.811 \\ 0.324 \\ 0 \end{bmatrix}$$

Note that $\sqrt{0.487^2 + 0.811^2 + 0.324^2} = 1$. ■

Example 2.2

A vector **p** is 5 units long and is in the direction of a unit vector **q** described below. Express the vector in matrix form.

$$\mathbf{q}_{unit} = \begin{bmatrix} 0.371 \\ 0.557 \\ q_z \\ 0 \end{bmatrix}$$

Solution: The unit vector's length must be 1. Therefore,

$$\lambda = \sqrt{q_x^2 + q_y^2 + q_z^2} = \sqrt{0.138 + 0.310 + q_z^2} = 1 \quad \rightarrow \quad q_z = 0.743$$

$$\mathbf{q}_{unit} = \begin{bmatrix} 0.371 \\ 0.557 \\ 0.743 \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{p} = \mathbf{q}_{unit} \times 5 = \begin{bmatrix} 1.855 \\ 2.785 \\ 3.715 \\ 1 \end{bmatrix}$$

■

2.4.3 Representation of a Frame at the Origin of a Fixed Reference Frame

A frame is generally represented by three mutually orthogonal axes (such as x, y, and z). Since we may have more than one frame at any given time, we will use axes x, y, and z to represent the fixed Universe reference frame $F_{x,y,z}$ and a set of axes n, o, and a to represent another (moving) frame $F_{n,o,a}$ relative to the reference frame. This way, there should be no confusion about which frame is referenced.

The letters *n*, *o*, and *a* are derived from the words *normal*, *orientation*, and *approach*. Referring to Figure 2.6, it should be clear that in order to avoid hitting the part while trying to pick it up, the robot would have to approach it along the *z*-axis of the gripper. In robotic nomenclature, this axis is called *approach-axis* and is referred to as the *a*-axis. The orientation with which the gripper frame approaches the part is called *orientation-axis*, and it is referred to as the *o*-axis. Since the *x*-axis is normal to both, it is referred to as *n*-axis. Throughout this book, we will refer to a moving frame as $F_{n,o,a}$ with *normal*, *orientation*, and *approach* axes.

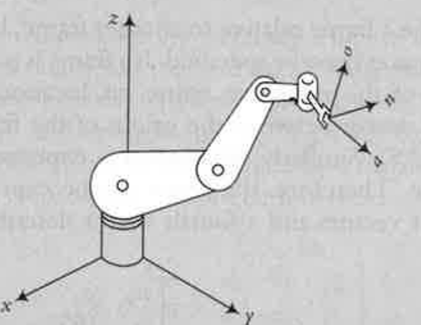


Figure 2.6 The normal-, orientation-, and approach-axis of a moving frame.

Each direction of each axis of a frame $F_{n,o,a}$ located at the origin of a reference frame $F_{x,y,z}$ (Figure 2.7) is represented by its three directional cosines relative to the reference frame as in section 2.4.2. Consequently, the three axes of the frame can be represented by three vectors in matrix form as:

$$F = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix} \tag{2.7}$$

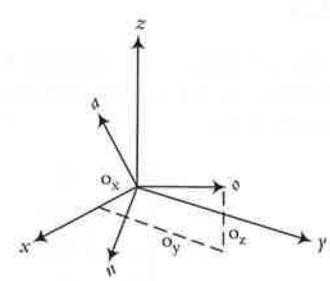


Figure 2.7 Representation of a frame at the origin of the reference frame.

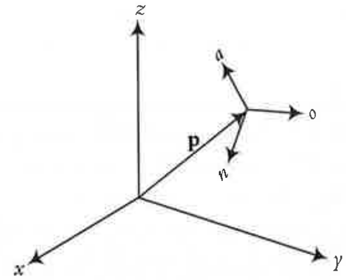


Figure 2.8 Representation of a frame in a frame.

2.4.4 Representation of a Frame Relative to a Fixed Reference Frame

To fully describe a frame relative to another frame, both the location of its origin and the directions of its axes must be specified. If a frame is not at the origin (or, in fact, even if it is at the origin) of the reference frame, its location relative to the reference frame is described by a vector between the origin of the frame and the origin of the reference frame (Figure 2.8). Similarly, this vector is expressed by its components relative to the reference frame. Therefore, the frame can be expressed by three vectors describing its directional unit vectors and a fourth vector describing its location as:

$$F = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.8}$$

As shown in Equation (2.8), the first three vectors are directional vectors with $w = 0$, representing the directions of the three unit vectors of the frame $F_{n,o,a}$, while the fourth vector with $w = 1$ represents the location of the origin of the frame relative to the reference frame. Unlike the unit vectors, the length of vector p is important. Consequently, we use a scale factor of 1.

A frame may also be represented by a 3×4 matrix without the scale factors, but it is not common. Adding the fourth row of scale factors to the matrix makes it a 4×4 or *homogeneous* matrix.

Example 2.3

The frame F shown in Figure 2.9 is located at 3,5,7 units, with its n-axis parallel to x, its o-axis at 45° relative to the y-axis, and its a-axis at 45° relative to the z-axis. The frame can be described by:

$$F = \begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 0.707 & -0.707 & 5 \\ 0 & 0.707 & 0.707 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

■

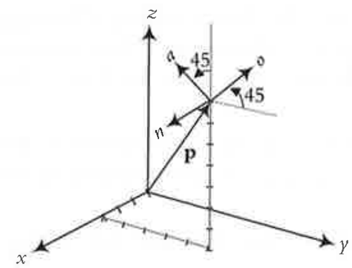


Figure 2.9 An example of representation of a frame.

2.4.5 Representation of a Rigid Body

An object can be represented in space by attaching a frame to it and representing the frame. Since the object is permanently attached to this frame, its position and orientation relative to the frame is always known. As a result, so long as the frame can be described in space, the object's location and orientation relative to the fixed frame will be known (Figure 2.10). As before, a frame can be represented by a matrix, where the origin of the frame and the three vectors representing its orientation relative to the reference frame are expressed. Therefore,

$$F_{object} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.9}$$

As we discussed in Chapter 1, a point in space has only three degrees of freedom; it can only move along the three reference axes. However, a rigid body in space has six degrees of freedom, meaning that not only can it move along x-, y-, and z-axes, it can also rotate about these three axes. Consequently, all that is needed to completely define an object in space is six pieces of information describing the location of the origin of the object in the reference frame and its orientation about the three axes. However, as can be seen in Equation (2.9), twelve pieces of information are given: nine for orientation, and three for position (this excludes the scale factors on the last row of the matrix because they do not add to this information). Obviously, there must be some constraints present in this

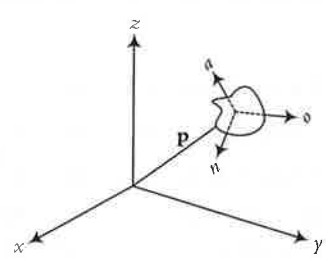


Figure 2.10 Representation of an object in space.

representation to limit the above to six. Therefore, we need 6 constraint equations to reduce the above from twelve to six. The constraints come from the known characteristics of a frame that have not been used yet, that:

- the three unit vectors **n**, **o**, **a** are mutually perpendicular, and
- each unit vector's length, represented by its directional cosines, must be equal to 1.

These constraints translate into the following six constraint equations:

1. $\mathbf{n} \cdot \mathbf{o} = 0$ (the dot-product of **n** and **o** vectors must be zero)
2. $\mathbf{n} \cdot \mathbf{a} = 0$
3. $\mathbf{a} \cdot \mathbf{o} = 0$ (2.10)
4. $|\mathbf{n}| = 1$ (the magnitude of the length of the vector must be 1)
5. $|\mathbf{o}| = 1$
6. $|\mathbf{a}| = 1$

As a result, the values representing a frame in a matrix must be such that the above equations remain true. Otherwise, the frame will not be correct. Alternatively, the first three equations in Equation (2.10) can be replaced by a cross product of the three vectors as:

$$\mathbf{n} \times \mathbf{o} = \mathbf{a} \tag{2.11}$$

Since Equation (2.11) includes the correct right-hand-rule relationship too, it is recommended that this equation be used to determine the correct relationship between the three vectors.

Example 2.4

For the following frame, find the values of the missing elements and complete the matrix representation of the frame:

$$F = \begin{bmatrix} ? & 0 & ? & 5 \\ 0.707 & ? & ? & 3 \\ ? & ? & 0 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Solution: Obviously, the 5,3,2 values representing the position of the origin of the frame do not affect the constraint equations. Please notice that only 3 values for directional vectors are given. This is all that is needed. Using Equation (2.10), we will get:

$$\begin{aligned}
&n_x o_x + n_y o_y + n_z o_z = 0 && \text{or} && n_x(0) + 0.707(o_y) + n_z(o_z) = 0 \\
&n_x a_x + n_y a_y + n_z a_z = 0 && \text{or} && n_x(a_x) + 0.707(a_y) + n_z(0) = 0 \\
&a_x o_x + a_y o_y + a_z o_z = 0 && \text{or} && a_x(0) + a_y(o_y) + 0(o_z) = 0 \\
&n_x^2 + n_y^2 + n_z^2 = 1 && \text{or} && n_x^2 + 0.707^2 + n_z^2 = 1 \\
&o_x^2 + o_y^2 + o_z^2 = 1 && \text{or} && 0^2 + o_y^2 + o_z^2 = 1 \\
&a_x^2 + a_y^2 + a_z^2 = 1 && \text{or} && a_x^2 + a_y^2 + 0^2 = 1
\end{aligned}$$

Simplifying these equations yields:

$$0.707\, o_y + n_z o_z = 0$$
$$n_x a_x + 0.707\, a_y = 0$$
$$a_y o_y = 0$$
$$n_x^2 + n_z^2 = 0.5$$
$$o_y^2 + o_z^2 = 1$$
$$a_x^2 + a_y^2 = 1$$

Solving these six equations will yield $n_x = \pm 0.707$, $n_z = 0$, $o_y = 0$, $o_z = 1$, $a_x = \pm 0.707$, and $a_y = -0.707$. Notice that both n_x and a_x must have the same sign. The reason for multiple solutions is that with the given parameters, it is possible to have two sets of mutually perpendicular vectors in opposite directions. The final matrix will be:

$$F_1 = \begin{bmatrix} 0.707 & 0 & 0.707 & 5 \\ 0.707 & 0 & -0.707 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{or} \quad F_2 = \begin{bmatrix} -0.707 & 0 & -0.707 & 5 \\ 0.707 & 0 & -0.707 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

As you can see, both matrices satisfy all the requirements set by the constraint equations. It is important to realize that the values representing the three direction vectors are not arbitrary but bound by these equations. Therefore, you may not randomly use any desired values in the matrix.

The same problem may be solved using $\mathbf{n} \times \mathbf{o} = \mathbf{a}$, or:

$$\begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ n_x & n_y & n_z \\ o_x & o_y & o_z \end{vmatrix} = a_x\mathbf{i} + a_y\mathbf{j} + a_z\mathbf{k}$$

or $$\mathbf{i}(n_y o_z - n_z o_y) - \mathbf{j}(n_x o_z - n_z o_x) + \mathbf{k}(n_x o_y - n_y o_x) = a_x\mathbf{i} + a_y\mathbf{j} + a_z\mathbf{k} \qquad (2.12)$$

Substituting the values into this equation yields:

$$\mathbf{i}(0.707 o_z - n_z o_y) - \mathbf{j}(n_x o_z) + \mathbf{k}(n_x o_y) = a_x\mathbf{i} + a_y\mathbf{j} + 0\mathbf{k}$$

Solving the three simultaneous equations will result in:

$$0.707\, o_z - n_z o_y = a_x$$
$$-n_x o_z = a_y$$
$$n_x o_y = 0$$

which replace the three equations for the dot products. Together with the three unit-vector length constraint equations, there will be six equations. However, as you will see, only one of the two solutions (F_1) obtained in the first part will satisfy these equations. This is because the dot-product equations are scalar, and therefore, are the same whether the unit vectors are right-handed or left-handed frames, whereas the cross-product equations do indicate the correct right-handed frame configuration. Consequently, it is recommended that the cross-product equation be used. ■

Example 2.5

Find the missing elements of the following frame representation:

$$F = \begin{bmatrix} ? & 0 & ? & 3 \\ 0.5 & ? & ? & 9 \\ 0 & ? & ? & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Solution:

$$n_x^2 + n_y^2 + n_z^2 = 1 \quad \rightarrow \quad n_x^2 + 0.25 = 1 \quad \rightarrow \quad n_x = 0.866$$

$$\mathbf{n} \cdot \mathbf{o} = 0 \quad \rightarrow \quad (0.866)(0) + (0.5)(o_y) + (0)(o_z) = 0 \quad \rightarrow \quad o_y = 0$$

$$|\mathbf{o}| = 1 \quad \rightarrow \quad o_z = 1$$

$$\mathbf{n} \times \mathbf{o} = \mathbf{a} \quad \rightarrow \quad \mathbf{i}(0.5) - \mathbf{j}(0.866) + \mathbf{k}(0) = a_x\mathbf{i} + a_y\mathbf{j} + a_z\mathbf{k}$$

$$a_x = 0.5$$

$$a_y = -0.866$$

$$a_z = 0$$

■

2.5 Homogeneous Transformation Matrices

For a variety of reasons, it is desirable to keep matrices in square form, either 3 × 3 or 4 × 4. First, as we will see later, it is much easier to calculate the inverse of square matrices than rectangular matrices. Second, in order to multiply two matrices, their dimensions must match, such that the number of columns of the first matrix must be the same as the number of rows of the second matrix, as in (m × n) and (n × p), which results in a matrix of (m × p) dimensions. If two matrices, A and B, are square with (m × m) and (m × m) dimensions, we may multiply A by B, or B by A, both resulting in the same (m × m) dimensions. However, if the two matrices are not square, with (m × n) and (n × p) dimensions respectively, A can be multiplied by B, but B may not be multiplied by A, and the result of AB has a dimension different from A and B. Since we will have to multiply many matrices together, in different orders, to find the equations of motion of the robots, we want to have square matrices.

In order to keep representation matrices square, if we represent both orientation and position in the same matrix, we will add the scale factors to the matrix to make it 4 × 4. If we represent the orientation alone, we may either drop the scale factors and use 3 × 3 matrices, or add a fourth column with zeros for position in order to keep the matrix square. Matrices of this form are called homogeneous matrices, and we refer to them as:

$$F = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.13}$$

2.6 Representation of Transformations

A transformation is defined as making a movement in space. When a frame (a vector, an object, or a moving frame) moves in space relative to a fixed reference frame, we represent this motion in a form similar to a frame representation. This is because a transformation is a change in the state of a frame (representing the change in its location and orientation); therefore, it can be represented like a frame. A transformation may be in one of the following forms:

- A pure translation
- A pure rotation about an axis
- A combination of translations and/or rotations

In order to see how these can be represented, we will study each one separately.

2.6.1 Representation of a Pure Translation

If a frame (that may also be representing an object) moves in space without any change in its orientation, the transformation is a pure translation. In this case, the directional unit vectors remain in the same direction, and therefore, do not change. The only thing that changes is the location of the origin of the frame relative to the reference frame, as shown in Figure 2.11. The new location of the frame relative to the fixed reference frame can be found by adding the vector representing the translation to the vector representing the original location of the origin of the frame. In matrix form, the new frame representation may be found by pre-multiplying the frame with a matrix representing the transformation. Since the directional vectors do not change in a pure translation, the transformation T will simply be:

$$T = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.14}$$

where d_x, d_y, and d_z are the three components of a pure translation vector $\mathbf{d}$ relative to the x-, y-, and z-axes of the reference frame. The first three columns represent no rotational movement (equivalent of a 1), while the last column represents the translation. The new

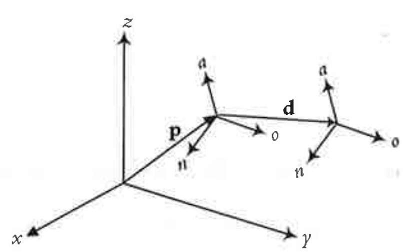


Figure 2.11 Representation of a pure translation in space.

location of the frame will be:

$$F_{new} = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & p_x + d_x \\ n_y & o_y & a_y & p_y + d_y \\ n_z & o_z & a_z & p_z + d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.15)$$

This equation is also symbolically written as:

$$F_{new} = Trans(d_x, d_y, d_z) \times F_{old} \quad (2.16)$$

First, as you can see, pre-multiplying the frame matrix by the transformation matrix will yield the new location of the frame. Second, notice that the directional vectors remain the same after a pure translation, but the new location of the frame is at **d+p**. Third, notice how homogeneous transformation matrices facilitate the multiplication of matrices, resulting in the same dimensions as before.

Example 2.6

A frame F has been moved 10 units along the y-axis and 5 units along the z-axis of the reference frame. Find the new location of the frame.

$$F = \begin{bmatrix} 0.527 & -0.574 & 0.628 & 5 \\ 0.369 & 0.819 & 0.439 & 3 \\ -0.766 & 0 & 0.643 & 8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Solution: Using Equation (2.15) or (2.16), we get:

$$F_{new} = Trans(d_x, d_y, d_z) \times F_{old} = Trans(0, 10, 5) \times F_{old}$$

and

$$F_{new} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 10 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0.527 & -0.574 & 0.628 & 5 \\ 0.369 & 0.819 & 0.439 & 3 \\ -0.766 & 0 & 0.643 & 8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.527 & -0.574 & 0.628 & 5 \\ 0.369 & 0.819 & 0.439 & 13 \\ -0.766 & 0 & 0.643 & 13 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

■

2.6.2 Representation of a Pure Rotation about an Axis

To simplify the derivation of rotations about an axis, let's first assume that the frame is at the origin of the reference frame and is parallel to it. We will later expand the results to other rotations as well as combinations of rotations.

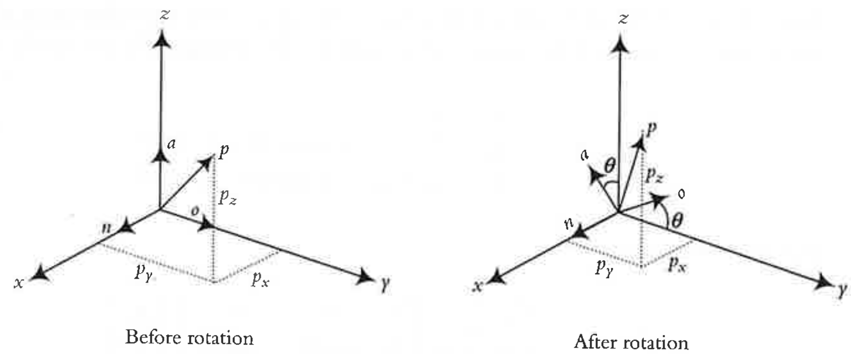


Figure 2.12 Coordinates of a point in a rotating frame before and after rotation.

Let's assume that a frame F_{noa}, located at the origin of the reference frame F_{xyz}, rotates an angle of θ about the x-axis of the reference frame. Let's also assume that attached to the rotating frame F_{noa}, is a point p, with coordinates p_x, p_y, and p_z relative to the reference frame and p_n, p_o, and p_a relative to the moving frame. As the frame rotates about the x-axis, point p attached to the frame will also rotate with it. Before rotation, the coordinates of the point in both frames are the same (remember that the two frames are at the same location and are parallel to each other). After rotation, the p_n, p_o, and p_a coordinates of the point remain the same in the rotating frame F_{noa}, but p_x, p_y, and p_z will be different in the F_{xyz} frame (Figure 2.12). We want to find the new coordinates of the point relative to the fixed reference frame after the moving frame has rotated.

Now let's look at the same coordinates in 2-D as if we were standing on the x-axis. The coordinates of point p are shown before and after rotation in Figure 2.13. The coordinates of point p relative to the reference frame are p_x, p_y, and p_z, while its coordinates relative to the rotating frame (to which the point is attached) remain as p_n, p_o, and p_a.

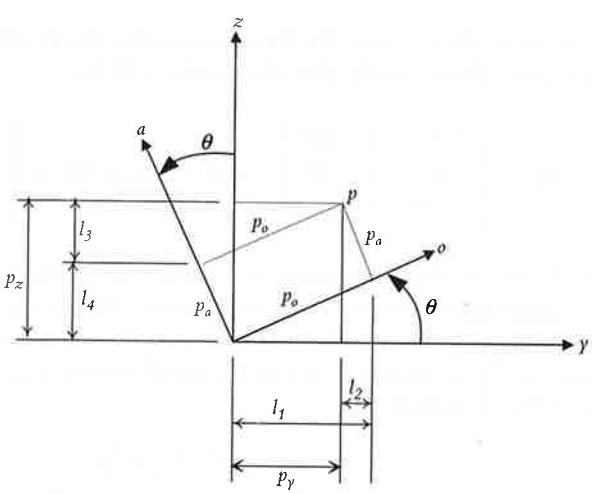


Figure 2.13 Coordinates of a point relative to the reference frame and rotating frame as viewed from the x-axis.

From Figure 2.13, you can see that the value of p_x does not change as the frame rotates about the x-axis, but the values of p_y and p_z do change. Please verify that:

$$\begin{aligned} p_x &= p_n \\ p_y &= l_1 - l_2 = p_o\cos\theta - p_a\sin\theta \\ p_z &= l_3 + l_4 = p_o\sin\theta + p_a\cos\theta \end{aligned} \tag{2.17}$$

and in matrix form:

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} p_n \\ p_o \\ p_a \end{bmatrix} \tag{2.18}$$

This means that the coordinates of the point p (or vector $\mathbf{p}$) in the rotated frame must be pre-multiplied by the rotation matrix, as shown, to get the coordinates in the reference frame. This rotation matrix is only for a pure rotation about the x-axis of the reference frame and is denoted as:

$$p_{xyz} = Rot(x, \theta) \times p_{noa} \tag{2.19}$$

Notice that the first column of the rotation matrix in Equation (2.18)—which expresses the location relative to the x-axis—has 1,0,0 values, indicating that the coordinate along the x-axis has not changed.

To simplify writing these matrices, it is customary to designate $C\theta$ to denote $\cos\theta$ and $S\theta$ to denote $\sin\theta$. Therefore, the rotation matrix may be also written as:

$$Rot(x, \theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\theta & -S\theta \\ 0 & S\theta & C\theta \end{bmatrix} \tag{2.20}$$

You may want to do the same for the rotation of a frame about the y- and z-axes of the reference frame. Please verify that the results will be:

$$Rot(y, \theta) = \begin{bmatrix} C\theta & 0 & S\theta \\ 0 & 1 & 0 \\ -S\theta & 0 & C\theta \end{bmatrix} \quad \text{and} \quad Rot(z, \theta) = \begin{bmatrix} C\theta & -S\theta & 0 \\ S\theta & C\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.21}$$

Equation (2.19) can also be written in a conventional form that assists in easily following the relationship between different frames. Denoting the transformation as ${}^{U}T_{R}$ (and reading it as the transformation of frame R relative to frame U (for Universe)), denoting p_{noa} as ${}^{R}p$ (p relative to frame R), and denoting p_{xyz} as ${}^{U}p$ (p relative to frame U), Equation (2.19) simplifies to:

$${}^{U}p = {}^{U}T_{R} \times {}^{R}p \tag{2.22}$$

As you see, canceling the Rs will yield the coordinates of point p relative to U. The same notation will be used throughout this book to relate to multiple transformations.

Example 2.7

A point $p(2,3,4)^T$ is attached to a rotating frame. The frame rotates 90° about the x-axis of the reference frame. Find the coordinates of the point relative to the reference frame after the rotation, and verify the result graphically.

Solution: Of course, since the point is attached to the rotating frame, the coordinates of the point relative to the rotating frame remain the same after the rotation. The coordinates of the point relative to the reference frame will be:

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\theta & -S\theta \\ 0 & S\theta & C\theta \end{bmatrix} \times \begin{bmatrix} p_n \\ p_o \\ p_a \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 2 \\ -4 \\ 3 \end{bmatrix}$$

As shown in Figure 2.14, the coordinates of point p relative to the reference frame after rotation are 2, −4, 3, as obtained by the above transformation.

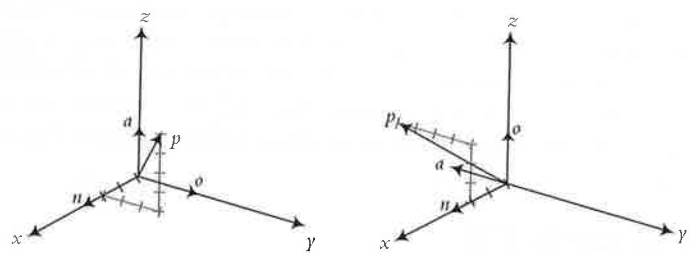


Figure 2.14 Rotation of a frame relative to the x-axis of the reference frame. ■

2.6.3 Representation of Combined Transformations

Combined transformations consist of a number of successive translations and rotations about the fixed reference frame axes or the moving current frame axes. Any transformation can be resolved into a set of translations and rotations in a particular order. For example, we may rotate a frame about the x-axis, then translate about the x-, y-, and z-axes, then rotate about the y-axis in order to accomplish the desired transformation. As we will see later, this order is very important, such that if the order of two successive transformations changes, the result may be completely different.

To see how combined transformations are handled, let's assume that a frame F_{noa} is subjected to the following three successive transformations relative to the reference frame F_{xyz}:

1. Rotation of α degrees about the x-axis,
2. Followed by a translation of $[l_1, l_2, l_3]$ (relative to the x-, y-, and z-axes respectively),
3. Followed by a rotation of β degrees about the y-axis.

Also, let's say that a point p_{noa} is attached to the rotating frame at the origin of the reference frame. As the frame F_{noa} rotates or translates relative to the reference frame, point p within the frame moves as well, and the coordinates of the point relative to the reference frame change. After the first transformation, as we saw in the previous section, the coordinates of point p relative to the reference frame can be calculated by:

$$p_{1,xyz} = Rot(x,\alpha) \times p_{noa} \tag{2.23}$$

where $p_{1,xyz}$ is the coordinates of the point after the first transformation relative to the reference frame. The coordinates of the point relative to the reference frame at the conclusion of the second transformation will be:

$$p_{2,xyz} = Trans(l_1,l_2,l_3) \times p_{1,xyz} = Trans(l_1,l_2,l_3) \times Rot(x,\alpha) \times p_{noa} \tag{2.24}$$

Similarly, after the third transformation, the coordinates of the point relative to the reference frame will be:

$$p_{xyz} = p_{3,xyz} = Rot(y,\beta) \times p_{2,xyz} = Rot(y,\beta) \times Trans(l_1,l_2,l_3) \times Rot(x,\alpha) \times p_{noa}$$

As you can see, the coordinates of the point relative to the reference frame at the conclusion of each transformation is found by pre-multiplying the coordinates of the point by each transformation matrix. Of course, as shown in Appendix A, the order of matrices cannot be changed, therefore this order is very important. You will also notice that for each transformation relative to the reference frame, the matrix is pre-multiplied. Consequently, the order of matrices *written* is the opposite of the order of transformations *performed*.

Example 2.8

A point $p(7,3,1)^T$ is attached to a frame F_{noa} and is subjected to the following transformations. Find the coordinates of the point relative to the reference frame at the conclusion of transformations.

1. Rotation of 90° about the z-axis,
2. Followed by a rotation of 90° about the y-axis,
3. Followed by a translation of [4,−3,7].

Solution: The matrix equation representing the transformation is:

$$p_{xyz} = Trans(4,-3,7)Rot(y,90)Rot(z,90)p_{noa}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 7 \\ 3 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 4 \\ 10 \\ 1 \end{bmatrix}$$

As you can see, the first transformation of 90° about the z-axis rotates the F_{noa} frame as shown in Figure 2.15, followed by the second rotation about the y-axis, followed by the translation relative to the reference frame F_{xyz}. The point p in the frame can then be found relative to the F_{noa} as shown. The final coordinates of the point can be

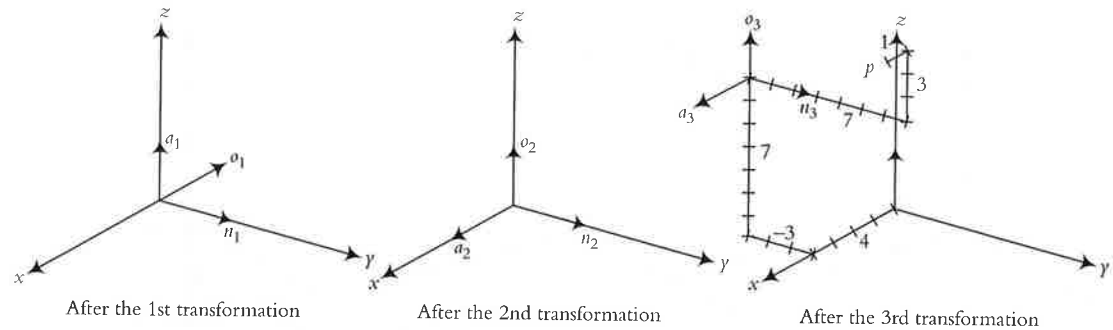


Figure 2.15 Effects of three successive transformations.

traced on the x-, y-, z-axes to be $4+1=5$, $-3+7=4$, and $7+3=10$. Be sure to follow this graphically. ■

Example 2.9

In this case, assume the same point $p(7,3,1)^T$, attached to F_{noa}, is subjected to the same transformations, but the transformations are performed in a different order, as shown. Find the coordinates of the point relative to the reference frame at the conclusion of transformations.

1. A rotation of 90° about the z-axis,
2. Followed by a translation of [4,−3,7],
3. Followed by a rotation of 90° about the y-axis.

Solution: The matrix equation representing the transformation is:

$$p_{xyz} = Rot(y,90)\,Trans(4,-3,7)\,Rot(z,90)\,p_{noa}$$

$$= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 7 \\ 3 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \\ -1 \\ 1 \end{bmatrix}$$

As you can see, although the transformations are exactly the same as in Example 2.8, since the order of transformations is changed, the final coordinates of the point are completely different. This can clearly be demonstrated graphically as in Figure 2.16. In this case, you can see that although the first transformation creates exactly the same change in the frame, the second transformation's result is very different because the translation relative to the reference frame axes will move the rotating frame F_{noa} outwardly. As a result of the third transformation, this frame will rotate about the y-axis, therefore rotating downwardly. The location of point p, attached to the frame is also shown. Please verify that the coordinates of this point

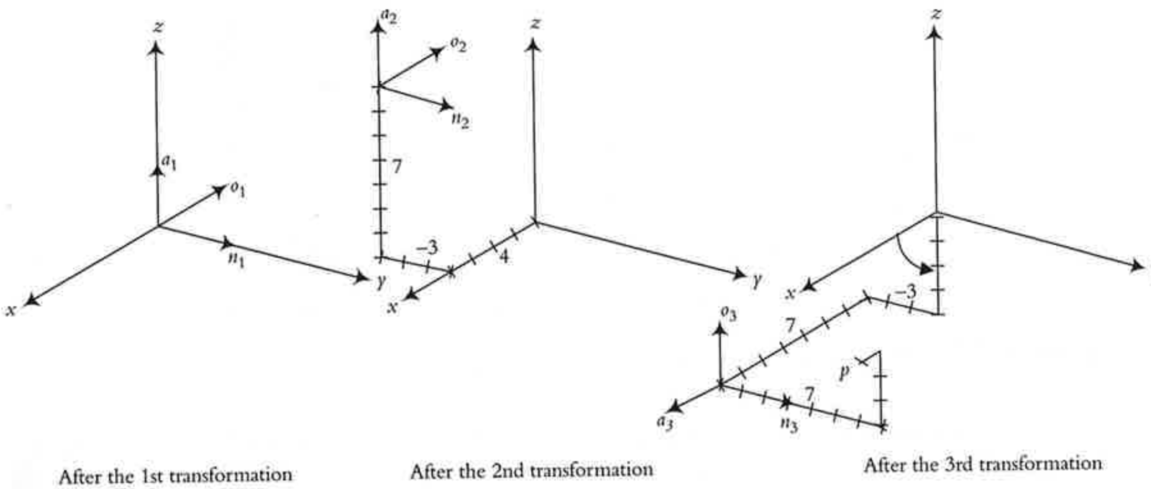


Figure 2.16 Changing the order of transformations will change the final result.

relative to the reference frame are $7 + 1 = 8$, $-3 + 7 = 4$, and $-4 + 3 = -1$, which is the same as the analytical result. ■

2.6.4 Transformations Relative to the Rotating Frame

All transformations we have discussed so far have been relative to the fixed reference frame. This means that all translations, rotations, and distances (except for the location of a point relative to the moving frame) have been measured relative to the reference frame axes. However, it is possible to make transformations relative to the axes of a moving or current frame. This means that, for example, a rotation of 90° may be made relative to the n-axis of the moving frame (also referred to as the current frame), and not the x-axis of the reference frame. To calculate the changes in the coordinates of a point attached to the current frame relative to the reference frame, the transformation matrix is post-multiplied instead. Note that since the position of a point or an object attached to a moving frame is always measured relative to that moving frame, the position matrix describing the point or object is also always post-multiplied.

Example 2.10

Assume that the same point as in Example 2.9 is now subjected to the same transformations, but all relative to the current moving frame, as listed below. Find the coordinates of the point relative to the reference frame after transformations are completed.

1. A rotation of 90° about the a-axis,
2. Then a translation of [4,−3,7] along n-, o-, a-axes
3. Followed by a rotation of 90° about the o-axis.

Solution: In this case, since the transformations are made relative to the current frame, each transformation matrix is post-multiplied. As a result, the equation

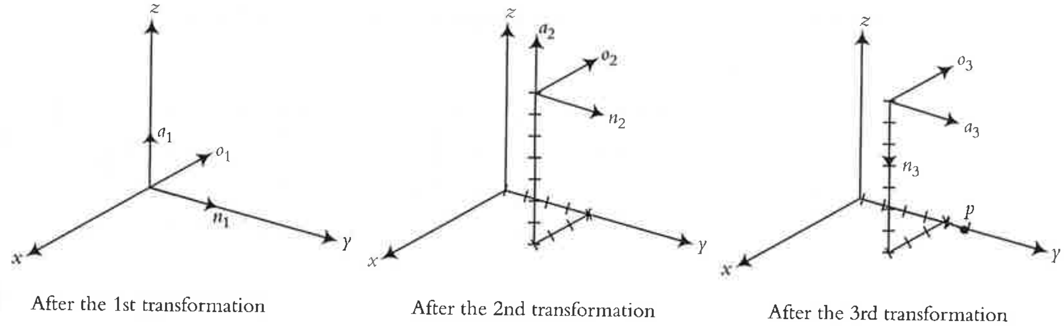


Figure 2.17 Transformations relative to the current frames.

representing the coordinates is:

$$p_{xyz} = Rot(a,90)\,Trans(4,-3,7)\,Rot(o,90)\,p_{noa}$$

$$= \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 7 \\ 3 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 5 \\ 0 \\ 1 \end{bmatrix}$$

As expected, the result is completely different from the other cases, both because the transformations are made relative to the current frame, and because the order of the matrices is now different. Figure 2.17 shows the results graphically. Notice how the transformations are accomplished relative to the current frames.

Notice how the 7,3,1 coordinates of point p in the current frame will result in 0,5,0 coordinates relative to the reference frame. ■

Example 2.11

A frame B was rotated about the x-axis 90°, then it was translated about the current a-axis 3 inches before it was rotated about the z-axis 90°. Finally, it was translated about current o–axis 5 inches.

(a) Write an equation that describes the motions.
(b) Find the final location of a point $p(1,5,4)^T$ attached to the frame relative to the reference frame.

Solution: In this case, motions alternate relative to the reference frame and current frame.

(a) Pre- or post-multiplying each motion's matrix accordingly, we will get:

$${}^{U}T_B = Rot(z,90)\,Rot(x,90)\,Trans(0,0,3)\,Trans(0,5,0)$$

(b) Substituting the matrices and multiplying them, we will get:

$${}^{U}p = {}^{U}T_B \times {}^{B}p$$

$$= \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 5 \\ 4 \\ 1 \end{bmatrix} = \begin{bmatrix} 7 \\ 1 \\ 10 \\ 1 \end{bmatrix}$$

■

Example 2.12

A frame F was rotated about the y-axis 90°, followed by a rotation about the o-axis of 30°, followed by a translation of 5 units along the n-axis, and finally, a translation of 4 units along the x-axis. Find the total transformation matrix.

Solution: The following set of matrices, written in the proper order to represent transformations relative to the reference frame or the current frame describes the total transformation:

$$T = Trans(4,0,0)Rot(y,90)Rot(o,30)Trans(5,0,0)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0.866 & 0 & 0.5 & 0 \\ 0 & 1 & 0 & 0 \\ -0.5 & 0 & 0.866 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -0.5 & 0 & 0.866 & 1.5 \\ 0 & 1 & 0 & 0 \\ -0.866 & 0 & -0.5 & -4.33 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Please verify graphically that this is true. ■

2.7 Inverse of Transformation Matrices

As mentioned earlier, there are many situations where the inverse of a matrix will be needed in robotic analysis. One situation where transformation matrices may be involved can be seen in the following example. Suppose the robot in Figure 2.18 is to be moved toward part P in order to drill a hole in the part. The robot's base position relative to the reference frame U is described by a frame R, the robot's hand is described by frame H, and the end effector (let's say the end of the drill bit that will be used to drill the hole) is described by frame E. The part's position is also described by frame P. The location of the point where the hole will be drilled can be related to the reference frame U through two independent paths: one through the part, one through the robot. Therefore, the

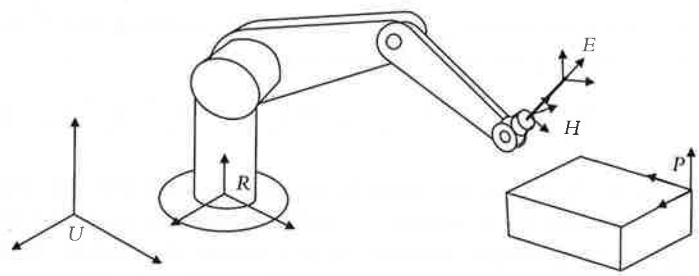


Figure 2.18 The Universe, robot, hand, part, and end effector frames.

following equation can be written:

$$^{U}T_{E} = {^{U}T_{R}}\,{^{R}T_{H}}\,{^{H}T_{E}} = {^{U}T_{P}}\,{^{P}T_{E}} \tag{2.25}$$

The location of point E on the part can be achieved by moving from U to P and from P to E, or it can alternately be achieved by a transformation from U to R, from R to H, and from H to E.

In reality, the transformation of frame R relative to the Universe frame (${^{U}T_{R}}$) is known since the location of the robot's base must be known in any set-up. For example, if a robot is installed in a work cell, the location of the robot's base will be known since it is bolted to a table. Even if the robot is mobile or attached to a conveyor belt, its location at any instant is known because a controller must be following the position of the robot's base at all times. The ${^{H}T_{E}}$, or the transformation of the end effector relative to the robot's hand, is also known since any tool used at the end effector is a known tool and its dimensions and configuration is known. ${^{U}T_{P}}$, or the transformation of the part relative to the universe, is also known since we must know where the part is located if we are to drill a hole in it. This location is known by putting the part in a jig, through the use of a camera and vision system, through the use of a conveyor belt and sensors, or other similar devices. ${^{P}T_{E}}$ is also known since we need to know where the hole is to be drilled on the part. Consequently, the only unknown transformation is ${^{R}T_{H}}$, or the transformation of the robot's hand relative to the robot's base. This means we need to find out what the robot's joint variables—the angle of the revolute joints and the length of the prismatic joints of the robot—must be in order to place the end effector at the hole for drilling. As you can see, it is necessary to calculate this transformation, which will tell us what needs to be accomplished. The transformation will later be used to actually solve for joint angles and link lengths.

To calculate this matrix, unlike in an algebraic equation, we cannot simply divide the right side by the left side of the equation. We need to pre- or post-multiply by inverses of appropriate matrices to eliminate them. As a result, we will have:

$$\left({^{U}T_{R}}\right)^{-1}\left({^{U}T_{R}}\,{^{R}T_{H}}\,{^{H}T_{E}}\right)\left({^{H}T_{E}}\right)^{-1} = \left({^{U}T_{R}}\right)^{-1}\left({^{U}T_{P}}\,{^{P}T_{E}}\right)\left({^{H}T_{E}}\right)^{-1} \tag{2.26}$$

or, since $\left({^{U}T_{R}}\right)^{-1}\left({^{U}T_{R}}\right) = I$ and $\left({^{H}T_{E}}\right)\left({^{H}T_{E}}\right)^{-1} = I$, the left side of Equation (2.26) simplifies to ${^{R}T_{H}}$ and we get:

$$^{R}T_{H} = {^{U}T_{R}^{-1}}\,{^{U}T_{P}}\,{^{P}T_{E}}\,{^{H}T_{E}^{-1}} \tag{2.27}$$

We can check the accuracy of this equation by realizing that $({}^{H}T_{E})^{-1}$ is the same as ${}^{E}T_{H}$. Therefore, the equation can be rewritten as:

$$ {}^{R}T_{H} = {}^{U}T_{R}^{-1}\,{}^{U}T_{P}\,{}^{P}T_{E}\,{}^{H}T_{E}^{-1} = {}^{R}T_{U}\,{}^{U}T_{P}\,{}^{P}T_{E}\,{}^{E}T_{H} = {}^{R}T_{H} \tag{2.28} $$

It is now clear that we need to be able to calculate the inverse of transformation matrices for kinematic analysis as well. In order to see what transpires, let's calculate the inverse of a simple rotation matrix about the x-axis. Please review the process for calculation of square matrices in Appendix A. The rotation matrix about the x-axis is:

$$ Rot(x,\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\theta & -S\theta \\ 0 & S\theta & C\theta \end{bmatrix} \tag{2.29} $$

Recall that the following steps must be taken to calculate the inverse of a matrix:

- Calculate the determinant of the matrix.
- Transpose the matrix.
- Replace each element of the transposed matrix by its own minor (adjoint matrix).
- Divide the converted matrix by the determinant.

Applying the process to the rotation matrix, we will get:

$$ \det[Rot(x,\theta)] = 1(C^2\theta + S^2\theta) + 0 = 1 $$
$$ Rot(x,\theta)^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\theta & S\theta \\ 0 & -S\theta & C\theta \end{bmatrix} $$

Now calculate each minor. As an example, the minor for the 2,2 element will be $C\theta - 0 = C\theta$, the minor for 1,1 element will be $C^2\theta + S^2\theta = 1$, and so on. As you will notice, the minor for each element will be the same as the element itself. Therefore:

$$ \mathrm{Adj}[Rot(x,\theta)] = Rot(x,\theta)^T_{minor} = Rot(x,\theta)^T $$

Since the determinant of the original rotation matrix is 1, dividing the Adj[$Rot\ (x,\ \theta)$] matrix by the determinant will yield the same result. Consequently, the inverse of a rotation matrix about the x-axis is the same as its transpose, or:

$$ Rot(x,\theta)^{-1} = Rot(x,\theta)^T \tag{2.30} $$

Of course, you would get the same result with the second method mentioned in Appendix A. A matrix with this characteristic is called a unitary matrix. It turns out that all rotation matrices are unitary matrices. Therefore, all we need to do to calculate the inverse of a rotation matrix is to transpose it. Please verify that rotation matrices about the

y- and z-axes are also unitary in nature. Beware that only rotation matrices are unitary; if a matrix is not a simple rotation matrix, it may not be unitary.

The preceding result is also true only for a simple 3 × 3 rotation matrix without representation of a location. For a homogenous 4 × 4 transformation matrix, it can be shown that the matrix inverse can be written by dividing the matrix into two portions; the rotation portion of the matrix can be simply transposed, as it is still unitary. The position portion of the homogeneous matrix is the negative of the dot product of the **p**-vector with each of the **n**-, **o**-, and **a**-vectors, as follows:

$$T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad T^{-1} = \begin{bmatrix} n_x & n_y & n_z & -\mathbf{p}\cdot\mathbf{n} \\ o_x & o_y & o_z & -\mathbf{p}\cdot\mathbf{o} \\ a_x & a_y & a_z & -\mathbf{p}\cdot\mathbf{a} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.31}$$

As shown, the rotation portion of the matrix is simply transposed, the position portion is replaced by the negative of the dot products, and the last row (scale factors) is not affected. This is very helpful, since we will need to calculate inverses of transformation matrices, but direct calculation of 4 × 4 matrices is a lengthy process.

Example 2.13

Calculate the matrix representing $Rot(x, 40°)^{-1}$.

Solution: The matrix representing a 40° rotation about the x-axis is:

$$Rot(x, 40°) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.766 & -0.643 & 0 \\ 0 & 0.643 & 0.766 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The inverse of this matrix is:

$$Rot(x, 40°)^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.766 & 0.643 & 0 \\ 0 & -0.643 & 0.766 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

As you can see, since the position vector of the matrix is zero, its dot product with the **n**-, **o**-, and **a**-vectors is also zero. ■

Example 2.14

Calculate the inverse of the given transformation matrix:

$$T = \begin{bmatrix} 0.5 & 0 & 0.866 & 3 \\ 0.866 & 0 & -0.5 & 2 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Solution: Based on the above, the inverse of the transformation will be:

$$T^{-1} = \begin{bmatrix} 0.5 & 0.866 & 0 & -(3\times 0.5 + 2\times 0.866 + 5\times 0) \\ 0 & 0 & 1 & -(3\times 0 + 2\times 0 + 5\times 1) \\ 0.866 & -0.5 & 0 & -(3\times 0.866 + 2\times -0.5 + 5\times 0) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.5 & 0.866 & 0 & -3.23 \\ 0 & 0 & 1 & -5 \\ 0.866 & -0.5 & 0 & -1.598 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

You may want to verify that TT^{-1} will be an identity matrix. ■

Example 2.15

In a robotic set-up, a camera is attached to the fifth link of a 6-DOF robot. It observes an object and determines its frame relative to the camera's frame. Using the following information, determine the necessary motion the end effector must make to get to the object:

$$^5T_{cam} = \begin{bmatrix} 0 & 0 & -1 & 3 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ^5T_H = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{cam}T_{obj} = \begin{bmatrix} 0 & 0 & 1 & 2 \\ 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ^HT_E = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Solution: Referring to Equation (2.25), we can write a similar equation that relates the different transformations and frames together as:

$$^RT_5 \times {}^5T_H \times {}^HT_E \times {}^ET_{obj} = {}^RT_5 \times {}^5T_{cam} \times {}^{cam}T_{obj}$$

Since $^RT_5$ appears on both sides of the equation, we can simply neglect it. All other matrices, with the exception of $^ET_{obj}$, are known. Then:

$$^ET_{obj} = {}^HT_E^{-1} \times {}^5T_H^{-1} \times {}^5T_{cam} \times {}^{cam}T_{obj} = {}^ET_H \times {}^HT_5 \times {}^5T_{cam} \times {}^{cam}T_{obj}$$

where $^HT_E^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ^5T_H^{-1} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Substituting the matrices and the inverses in the above equation will result:

$$ {}^{E}T_{obj} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & -1 & 3 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 2 \\ 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

or

$$ {}^{E}T_{obj} = \begin{bmatrix} -1 & 0 & 0 & -2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & -1 & -4 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

■

2.8 Forward and Inverse Kinematics of Robots

Suppose we have a robot whose configuration is known. This means that all the link lengths and joint angles of the robot are known. Calculating the position and orientation of the hand of the robot is called forward kinematic analysis. In other words, if all robot joint variables are known, using forward kinematic equations, we can calculate where the robot is at any instant. However, if we want to place the hand of the robot at a desired location and orientation, we need to know how much each link length or joint angle of the robot must be such that—at those values—the hand will be at the desired position and orientation. This is called inverse kinematic analysis. This means that instead of substituting the known robot variables in the forward kinematic equations of the robot, we need to find the inverse of these equations to enable us to find the necessary joint values to place the robot at the desired location and orientation. In reality, the inverse kinematic equations are more important since the robot controller will calculate the joint values using these equations and it will run the robot to the desired position and orientation. We will first develop the forward kinematic equations of robots; then, using these equations, we will calculate the inverse kinematic equations.

For forward kinematics, we will have to develop a set of equations that relate to the particular configuration of a robot (the way it is put together) such that by substituting the joint and link variables in these equations, we may calculate the position and orientation of the robot. These equations will then be used to derive the inverse kinematic equations.

You may recall from Chapter 1 that in order to position and orientate a rigid body in space, we attach a frame to the body and then describe the position of the origin of the frame and the orientation of its three axes. This requires a total of 6 DOF, or alternately, six pieces of information, to completely define the position and orientation of the body. Here too, if we want to define or find the position and orientation of the hand of the robot in space, we will attach a frame to it and define the position and orientation of the hand frame of the robot. The means by which the robot accomplishes this determines the forward kinematic equations. In other words, depending on the configuration of the links and joints of the robot, a particular set of equations will relate the hand frame of the robot to the reference frame. Figure 2.19 shows a hand frame, the reference frame, and their relative positions and orientations. The undefined connection between the two frames is related to the configuration of the robot. Of

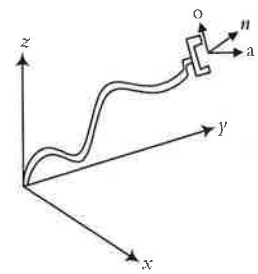


Figure 2.19 The hand frame of the robot relative to the reference frame.

course, there are many different possibilities for this configuration, and we will later see how we can develop the equations relating the two frames, depending on the robot configuration.

In order to simplify the process, we will analyze the position and orientation issues separately. First, we will develop the position equations, then we will do the same for orientation. Later, we will combine the two for a complete set of equations. Finally, we will see about the use of the Denavit-Hartenberg representation, which can model any robot configuration.

2.9 Forward and Inverse Kinematic Equations: Position

In this section, we will study the forward and inverse kinematic equations for position. As was mentioned earlier, the position of the origin of a frame attached to a rigid body has three degrees of freedom, and therefore, can be completely defined by three pieces of information. As a result, the position of the origin of the frame may be defined in any customary coordinates. As an example, we may position a point in space based on Cartesian coordinates, meaning there will be three linear movements relative to the x-, y-, and z-axes. Alternately, it may be accomplished through spherical coordinates, meaning there will be one linear motion and two rotary motions. The following possibilities will be discussed:

(a) Cartesian (gantry, rectangular) coordinates
(b) Cylindrical coordinates
(c) Spherical coordinates
(d) Articulated (anthropomorphic or all-revolute) coordinates

2.9.1 Cartesian (Gantry, Rectangular) Coordinates

In this case, there will be three linear movements along the x-, y-, and z-axes. In this type of robot, all actuators are linear (such as a hydraulic ram or a linear power screw), and the positioning of the hand of the robot is accomplished by moving the three linear joints along the three axes (Figure 2.20). A gantry robot is basically a Cartesian coordinate robot, except that the robot is usually attached to a rectangular frame upside down.

Of course, since there are no rotations, the transformation matrix representing this motion to point p is a simple translation matrix (shown next). Note that here we are only

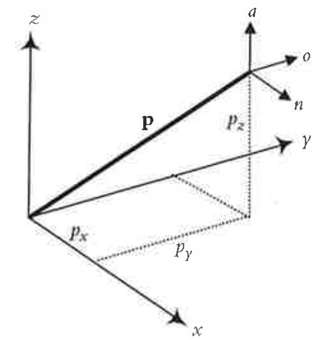


Figure 2.20 Cartesian coordinates.

concerned with the position of the origin of the frame—not its orientation. The transformation matrix representing the forward kinematic equation of the position of the hand of the robot in a Cartesian coordinate system will be:

$$^{R}T_{p} = T_{cart}(p_x, p_y, p_z) = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.32)$$

where $^{R}T_p$ is the transformation between the reference frame and the origin of the hand p, and $T_{cart}(p_x, p_y, p_z)$ denotes Cartesian transformation matrix. For the inverse kinematic solution, simply set the desired position equal to p.

Example 2.16

It is desired to position the origin of the hand frame of a Cartesian robot at point $p = [3,4,7]^T$. Calculate the necessary Cartesian coordinate motions that need to be made.

Solution: Setting the forward kinematic equation, represented by the $^{R}T_p$ matrix of Equation (2.32), equal to the desired position will yield the following result:

$$^{R}T_{p} = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{or} \quad p_x = 3,\ p_y = 4,\ p_z = 7$$

■

2.9.2 Cylindrical Coordinates

A cylindrical coordinate system includes two linear translations and one rotation. The sequence is a translation of r along the x-axis, a rotation of α about the z-axis, and a

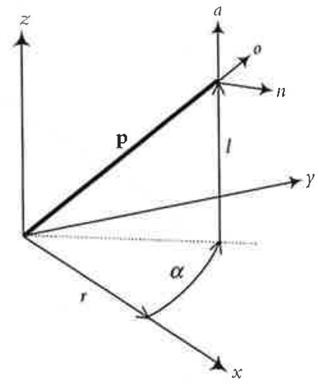


Figure 2.21 Cylindrical coordinates.

translation of l along the z-axis, as shown in Figure 2.21. Since these transformations are all relative to the Universe frame, the total transformation caused by these three transformations is found by pre-multiplying by each matrix, as follows:

$$^{R}T_{p} = T_{cyl}(r,\alpha,l) = Trans(0,0,l)Rot(z,\alpha)Trans(r,0,0) \tag{2.33}$$

$$^{R}T_{p} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} C\alpha & -S\alpha & 0 & 0 \\ S\alpha & C\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & r \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{R}T_{p} = T_{cyl}(r,\alpha,l) = \begin{bmatrix} C\alpha & -S\alpha & 0 & rC\alpha \\ S\alpha & C\alpha & 0 & rS\alpha \\ 0 & 0 & 1 & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.34}$$

The first three columns represent the orientation of the frame after this series of transformations. However, at this point, we are only interested in the position of the origin of the frame, or the last column. Obviously, in cylindrical coordinate movements, due to the rotation of α about the z-axis, the orientation of the moving frame will change. This orientation change will be discussed later.

You may restore the original orientation of the frame by rotating the n,o,a frame about the a-axis an angle of $-\alpha$, which is equivalent of post-multiplying the cylindrical coordinate matrix by a rotation matrix of $Rot(a, -\alpha)$. As a result, the frame will be at the same location but will be parallel to the reference frame again, as follows:

$$T_{cyl} \times Rot(a,-\alpha) = \begin{bmatrix} C\alpha & -S\alpha & 0 & rC\alpha \\ S\alpha & C\alpha & 0 & rS\alpha \\ 0 & 0 & 1 & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} C(-\alpha) & -S(-\alpha) & 0 & 0 \\ S(-\alpha) & C(-\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & rC\alpha \\ 0 & 1 & 0 & rS\alpha \\ 0 & 0 & 1 & l \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

As you can see, the location of the origin of the moving frame has not changed, but it was restored back to being parallel to the reference frame. Notice that the last rotation was performed about the local a-axis in order to not cause any change in the location of the frame, but only in its orientation.

Example 2.17

Suppose we desire to place the origin of the hand frame of a cylindrical robot at $[3,4,7]^T$. Calculate the joint variables of the robot.

Solution: Setting the components of the location of the origin of the frame from the T_{cyl} matrix of Equation (2.34) to the desired values, we get:

$$l = 7$$

$$rC\alpha = 3 \quad \text{and} \quad rS\alpha = 4 \quad \text{and therefore, } \tan\alpha = {}^4/_3 \text{ and } \alpha = 53.1^\circ$$

Substituting α into either equation will yield $r = 5$. The final answer is $r = 5$ units, $\alpha = 53.1^\circ$, and $l = 7$ units. Note: As discussed in Appendix A, it is necessary to ensure that the angles calculated in robot kinematics are in correct quadrants. In this example, $rC\alpha$ and $rS\alpha$ are both positive and the length r is always positive, therefore $S\alpha$ and $C\alpha$ are also both positive. Consequently, the angle α is in quadrant 1 and is correctly 53.1°.

■

Example 2.18

The position and restored orientation of a cylindrical robot are given. Find the matrix representing the original position and orientation of the robot before it was restored.

$$T = \begin{bmatrix} 1 & 0 & 0 & -2.394 \\ 0 & 1 & 0 & 6.578 \\ 0 & 0 & 1 & 9 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Solution: Since r is always positive, it is clear that $S\alpha$ and $C\alpha$ are positive and negative, respectively. Therefore, α is in the second quadrant. From T, we get:

$$l = 9$$

$$\tan(\alpha) = \frac{6.578}{-2.394} = -2.748 \quad \rightarrow \quad \alpha = 180^\circ - 70^\circ = 110^\circ$$

$$r\sin(\alpha) = 6.578 \rightarrow r = 7$$

Substituting these values into Equation (2.34), we will get the original orientation of the robot:

$$^{R}T_{p} = \begin{bmatrix} C\alpha & -S\alpha & 0 & rC\alpha \\ S\alpha & C\alpha & 0 & rS\alpha \\ 0 & 0 & 1 & l \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -0.342 & -0.9397 & 0 & -2.394 \\ 0.9397 & -0.342 & 0 & 6.578 \\ 0 & 0 & 1 & 9 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \blacksquare$$

2.9.3 Spherical Coordinates

A spherical coordinate system consists of one linear motion and two rotations. The sequence is a translation of r along the z-axis, a rotation of β about the y-axis, and a rotation of γ about the z-axis as shown in Figure 2.22. Since these transformations are all relative to the Universe frame, the total transformation caused by these three transformations can be found by pre-multiplying by each matrix, as follows:

$$^{R}T_{p} = T_{sph}(r, \beta, \gamma) = Rot(z, \gamma)Rot(y, \beta)Trans(0, 0, r) \tag{2.35}$$

$$^{R}T_{\mathrm{p}} = \begin{bmatrix} C\gamma & -S\gamma & 0 & 0 \\ S\gamma & C\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} C\beta & 0 & S\beta & 0 \\ 0 & 1 & 0 & 0 \\ -S\beta & 0 & C\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.36}$$

$$^{R}T_{P} = T_{sph}(r, \beta, \gamma) = \begin{bmatrix} C\beta C\gamma & -S\gamma & S\beta C\gamma & rS\beta C\gamma \\ C\beta S\gamma & C\gamma & S\beta S\gamma & rS\beta S\gamma \\ -S\beta & 0 & C\beta & rC\beta \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

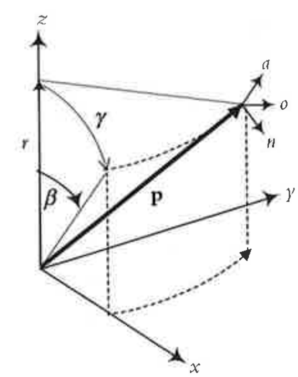


Figure 2.22 Spherical coordinates.

The first three columns represent the orientation of the frame after this series of transformations, while the last column is the position of the origin. We will discuss the orientation part of the matrix later. Note that spherical coordinates may be defined in other orders as well. Therefore, make sure correct equations are used.

Here, too, you may restore the original orientation of the final frame and make it parallel to the reference frame. This exercise is left for you to find the correct sequence of movements to get the right answer.

The inverse kinematic equations for spherical coordinates are more complicated than the simple Cartesian or cylindrical coordinates because the two angles β and γ are coupled. Let's see how this could be done through an example.

Example 2.19

Suppose we now want to place the origin of the hand of a spherical robot at $[3,4,7]^T$. Calculate the joint variables of the robot.

Solution: Setting the components of the location of the origin of the frame from T_{sph} matrix of Equation (2.36) to the desired values, we get:

$$rS\beta C\gamma = 3$$
$$rS\beta S\gamma = 4$$
$$rC\beta = 7$$

From the third equation, we determine that the $C\beta$ is positive, but there is no such information about $S\beta$. Therefore, because we do not know the actual sign of $S\beta$, there are two possible solutions. Later, we will have to check the final results to ensure they are correct.

$\tan\gamma = {}^4\!/_3 \rightarrow$	$\gamma = 53.1°$		or	$233.1°$
then	$S\gamma = 0.8$		or	-0.8
and	$C\gamma = 0.6$		or	-0.6
and	$rS\beta = {}^3\!/_{0.6} = 5$		or	-5
and since	$rC\beta = 7, \rightarrow$	$\beta = 35.5°$	or	$-35.5°$
and	$r = 8.6$			

You may check both answers and verify that they both satisfy all position equations. If you also follow these angles about the given axes in 3-D, you will get to the same point physically. However, you must notice that only one set of answers will also satisfy the orientation equations. In other words, the two answers above will result in the same position, but at different orientations. Since we are not concerned with the orientation of the hand frame at this point, both position answers are correct. In fact, since we cannot specify any orientation for a 3-DOF robot anyway, we cannot determine which of the two answers relates to a desired orientation. ■

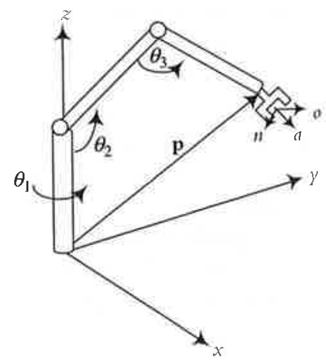


Figure 2.23 Articulated coordinates.

2.9.4 Articulated Coordinates

Articulated coordinates consist of three rotations, as shown in Figure 2.23. We will develop the matrix representation for this later, when we discuss the Denavit-Hartenberg representation.

2.10 Forward and Inverse Kinematic Equations: Orientation

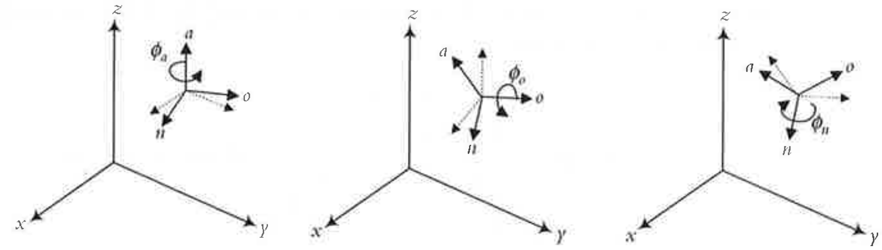
Suppose the moving frame attached to the hand of the robot has already moved to a desired position—in Cartesian, cylindrical, spherical, or articulated coordinates—and is either parallel to the reference frame or is in an orientation other than what is desired. The next step will be to rotate the frame appropriately in order to achieve a desired orientation without changing its position. This can only be accomplished by rotating about the current frame axes; rotations about the reference frame axes will change the position. The appropriate sequence of rotations depends on the design of the wrist of the robot and the way the joints are assembled together. We will consider the following three common configurations:

(a) Roll, Pitch, Yaw (RPY) angles
(b) Euler angles
(c) Articulated joints

2.10.1 Roll, Pitch, Yaw (RPY) Angles

This is a sequence of three rotations about current *a*-, *o*-, and *n*-axes respectively, which will orientate the hand of the robot to a desired orientation. The assumption here is that the current frame is parallel to the reference frame; therefore, its orientation is the same as the reference frame before the application of RPY. If the current moving frame is not parallel to the reference frame, then the final orientation of the robot's hand will be a combination of the previous orientation, post-multiplied by the RPY.

It is very important to realize that since we do not want to cause any change in the position of the origin of the moving frame (we have already placed it at the desired



Rotation ϕ_a Rotation ϕ_o Rotation ϕ_n

Figure 2.24 RPY rotations about the current axes.

location and only want to rotate it to the desired orientation), the movements relating to RPY rotations are relative to the current moving axes. Otherwise, as we saw before, the position of the frame will change. Therefore, all matrices related to the orientation change due to RPY (as well as other rotations) will be post-multiplied. Referring to Figure 2.24, the RPY sequence of rotations consists of:

Rotation of ϕ_a about the a-axis (z-axis of the moving frame) called Roll,
Rotation of ϕ_o about the o-axis (y-axis of the moving frame) called Pitch,
Rotation of ϕ_n about the n-axis (x-axis of the moving frame) called Yaw.

The matrix representing the RPY orientation change will be:

$$\mathrm{RPY}(\phi_a, \phi_o, \phi_n) = Rot(a, \phi_a)Rot(o, \phi_o)Rot(n, \phi_n)$$

$$= \begin{bmatrix} C\phi_a C\phi_o & C\phi_a S\phi_o S\phi_n - S\phi_a C\phi_n & C\phi_a S\phi_o C\phi_n + S\phi_a S\phi_n & 0 \\ S\phi_a C\phi_o & S\phi_a S\phi_o S\phi_n + C\phi_a C\phi_n & S\phi_a S\phi_o C\phi_n - C\phi_a S\phi_n & 0 \\ -S\phi_o & C\phi_o S\phi_n & C\phi_o C\phi_n & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.37)$$

This matrix represents the orientation change caused by the RPY alone. The location and the final orientation of the frame relative to the reference frame will be the product of the two matrices representing the position change and the RPY. For example, suppose that a robot is designed based on spherical coordinates and RPY. Then the robot may be represented by:

$${}^{R}T_H = T_{sph}(r, \beta, \gamma) \times \mathrm{RPY}(\phi_a, \phi_o, \phi_n)$$

The inverse kinematic solution for the RPY is more complicated than the spherical coordinates because here there are three coupled angles, where we need to have information about the sines and the cosines of all three angles individually to solve for the angles. To solve for these sines and cosines, we will have to de-couple these angles. To do this, we will pre-multiply both sides of Equation (2.37) by the inverse of $Rot(a, \phi_a)$:

$$Rot(a, \phi_a)^{-1}\,\mathrm{RPY}(\phi_a, \phi_o, \phi_n) = Rot(o, \phi_o)Rot(n, \phi_n) \quad (2.38)$$

Assuming that the final desired orientation achieved by RPY is represented by the (*n*,*o*,*a*) matrix, we will have:

$$Rot(a,\phi_a)^{-1}\begin{bmatrix} n_x & o_x & a_x & 0 \\ n_y & o_y & a_y & 0 \\ n_z & o_z & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = Rot(o,\phi_o)Rot(n,\phi_n) \tag{2.39}$$

Multiplying the matrices, we will get:

$$\begin{bmatrix} n_xC\phi_a + n_yS\phi_a & o_xC\phi_a + o_yS\phi_a & a_xC\phi_a + a_yS\phi_a & 0 \\ n_yC\phi_a - n_xS\phi_a & o_yC\phi_a - o_xS\phi_a & a_yC\phi_a - a_xS\phi_a & 0 \\ n_z & o_z & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} C\phi_o & S\phi_oS\phi_n & S\phi_oC\phi_n & 0 \\ 0 & C\phi_n & -S\phi_n & 0 \\ -S\phi_o & C\phi_oS\phi_n & C\phi_oC\phi_n & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.40}$$

Remember that the *n*,*o*,*a* components in Equation (2.39) represent the final desired values normally given or known. The values of the RPY angles are the unknown variables. Equating the different elements of the right-hand and left-hand sides of Equation (2.40) will result in the following. Refer to Appendix A for an explanation of *ATAN2* function.

From the 2,1 elements we get:

$$n_yC\phi_a - n_xS\phi_a = 0 \rightarrow \phi_a = ATAN2(n_y, n_x) \text{ and } \phi_a = ATAN2(-n_y, -n_x) \tag{2.41}$$

Note that since we do not know the signs of $\sin(\phi_a)$ or $\cos(\phi_a)$, two complementary solutions are possible. From the 3,1 and 1,1 elements we get:

$$\begin{aligned} &S\phi_o = -n_z \\ &C\phi_o = n_xC\phi_a + n_yS\phi_a \rightarrow \phi_o = ATAN2[-n_z, (n_xC\phi_a + n_yS\phi_a)] \end{aligned} \tag{2.42}$$

And finally, from the 2,2 and 2,3 elements we get:

$$\begin{aligned} &C\phi_n = o_yC\phi_a - o_xS\phi_a \\ &S\phi_n = -a_yC\phi_a + a_xS\phi_a \rightarrow \phi_n = ATAN2[(-a_yC\phi_a + a_xS\phi_a), (o_yC\phi_a - o_xS\phi_a)] \end{aligned} \tag{2.43}$$

Example 2.20

The desired final position and orientation of the hand of a Cartesian-RPY robot is given below. Find the necessary RPY angles and displacements.

$$^{R}T_P = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.354 & -0.674 & 0.649 & 4.33 \\ 0.505 & 0.722 & 0.475 & 2.50 \\ -0.788 & 0.160 & 0.595 & 8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Solution: From the above equations, we find two sets of answers:

$$\phi_a = ATAN2(n_y, n_x) = ATAN2(0.505, 0.354) = 55^\circ \text{ or } 235^\circ$$
$$\phi_o = ATAN2(-n_z, (n_x C\phi_a + n_y S\phi_a)) = ATAN2(0.788, 0.616) = 52^\circ \text{ or } 128^\circ$$
$$\phi_n = ATAN2((-a_y C\phi_a + a_x S\phi_a), (o_y C\phi_a - o_x S\phi_a))$$
$$= ATAN2(0.259, 0.966) = 15^\circ \text{ or } 195^\circ$$
$$p_x = 4.33 \quad p_y = 2.5 \quad p_z = 8 \text{ units.}$$

■

Example 2.21

For the same position and orientation as in Example 2.20, find all necessary joint variables if the robot is cylindrical-RPY.

Solution: In this case, we will use:

$$^{R}T_P = \begin{bmatrix} 0.354 & -0.674 & 0.649 & 4.33 \\ 0.505 & 0.722 & 0.475 & 2.50 \\ -0.788 & 0.160 & 0.595 & 8 \\ 0 & 0 & 0 & 1 \end{bmatrix} = T_{cyl}(r, \alpha, l) \times \text{RPY}(\phi_a, \phi_o, \phi_n)$$

The right-hand side of this equation now involves four coupled angles; as before, these must be de-coupled. However, since the rotation of α about the z-axis for the cylindrical coordinates does not affect the a-axis, it remains parallel to the z-axis. As a result, the rotation of ϕ_a about the a-axis for RPY will simply be added to α. This means that the 55° angle we found for ϕ_a is the summation of $\phi_a + \alpha$ (see Figure 2.25). Using the position information given, the solution of Example 2.20, and referring to Equation (2.34), we get:

$$rC\alpha = 4.33, \quad rS\alpha = 2.5 \rightarrow \alpha = 30^\circ$$
$$\phi_a + \alpha = 55^\circ \quad \rightarrow \phi_a = 25^\circ$$
$$S\alpha = 0.5 \quad \rightarrow r = 5$$
$$p_z = 8 \quad \rightarrow l = 8$$

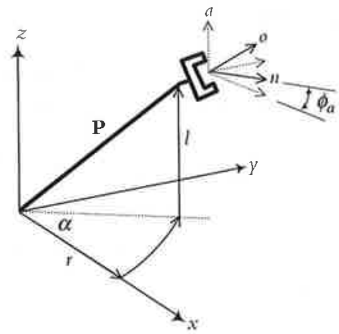


Figure 2.25 Cylindrical and RPY coordinates of Example 2.21.

As in Example 2.16:

$$\rightarrow \phi_o = 52°, \quad \phi_n = 15°$$

Of course, a similar solution may be found for the second set of answers. ■

2.10.2 Euler Angles

Euler angles are very similar to RPY, except that the last rotation is also about the current *a*-axis (Figure 2.26). We still need to make all rotations relative to the current axes to prevent any change in the position of the robot. Therefore, the rotations representing the Euler angles will be:

Rotation of ϕ about the *a*-axis (*z*-axis of the moving frame) followed by,
Rotation of θ about the *o*-axis (*y*-axis of the moving frame) followed by,
Rotation of ψ about the *a*-axis (*z*-axis of the moving frame).

The matrix representing the Euler angles orientation change will be:

$$\text{Euler}(\phi, \theta, \psi) = Rot(a, \phi)Rot(o, \theta), Rot(a, \psi)$$

$$= \begin{bmatrix} C\phi C\theta C\psi - S\phi S\psi & -C\phi C\theta S\psi - S\phi C\psi & C\phi S\theta & 0 \\ S\phi C\theta C\psi + C\phi S\psi & -S\phi C\theta S\psi + C\phi C\psi & S\phi S\theta & 0 \\ -S\theta C\psi & S\theta S\psi & C\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.44)$$

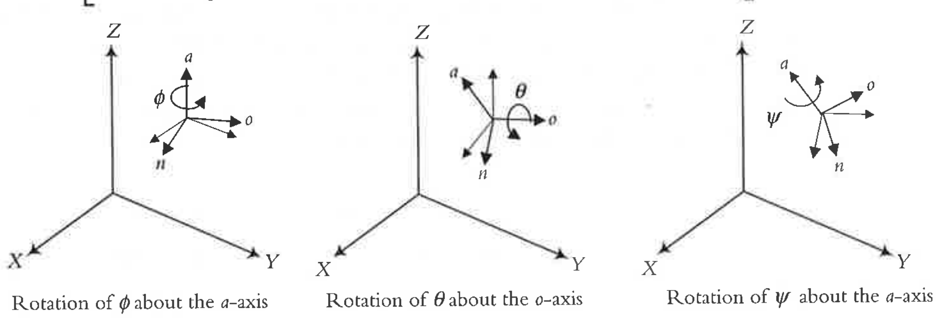


Figure 2.26 Euler rotations about the current axes.

Once again, this matrix represents the orientation change caused by the Euler angles alone. The location and the final orientation of the frame relative to the reference frame will be the product of the two matrices representing the position change and the Euler angles.

The inverse kinematic solution for the Euler angles can be found in a manner very similar to RPY. We will pre-multiply the two sides of the Euler equation by $Rot^{-1}(a,\phi)$ to eliminate ϕ from one side. By equating the elements of the two sides to each other, we will find the following equations. Assuming the final desired orientation achieved by Euler angles is represented by the (n,o,a) matrix:

$$Rot^{-1}(a,\phi) \times \begin{bmatrix} n_x & o_x & a_x & 0 \\ n_y & o_y & a_y & 0 \\ n_z & o_z & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C\theta C\psi & -C\theta S\psi & S\theta & 0 \\ S\psi & C\psi & 0 & 0 \\ -S\theta C\psi & S\theta S\psi & C\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.45}$$

or,

$$\begin{bmatrix} n_x C\phi + n_y S\phi & o_x C\phi + o_y S\phi & a_x C\phi + a_y S\phi & 0 \\ -n_x S\phi + n_y C\phi & -o_x S\phi + o_y C\phi & -a_x S\phi + a_y C\phi & 0 \\ n_z & o_z & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C\theta C\psi & -C\theta S\psi & S\theta & 0 \\ S\psi & C\psi & 0 & 0 \\ -S\theta C\psi & S\theta S\psi & C\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.46}$$

Remember that the n,o,a components in Equation (2.45) represent the final desired values that are normally given or known. The values of the Euler angles are the unknown variables. Equating the different elements of the right-hand and left-hand sides of Equation (2.46) will result in the following.

From the 2,3 elements we get:

$$-a_x S\phi + a_y C\phi = 0 \rightarrow \phi = ATAN2(a_y, a_x) \text{ or } \phi = ATAN2(-a_y, -a_x) \tag{2.47}$$

With ϕ evaluated, all the elements of the left-hand side of Equation (2.46) are known. From the 2,1 and 2,2 elements we get:

$$\begin{aligned} S\psi &= -n_x S\phi + n_y C\phi \\ C\psi &= -o_x S\phi + o_y C\phi \rightarrow \psi = ATAN2\left[(-n_x S\phi + n_y C\phi), (-o_x S\phi + o_y C\phi)\right] \end{aligned} \tag{2.48}$$

And finally, from the 1,3 and 3,3 elements we get:

$$S\theta = a_x C\phi + a_y S\phi$$
$$C\theta = a_z \rightarrow \theta = ATAN2\left[\left(a_x C\phi + a_y S\phi\right), a_z\right] \tag{2.49}$$

Example 2.22

The desired final orientation of the hand of a Cartesian-Euler robot is given. Find the necessary Euler angles.

$$^{R}T_{H} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.579 & -0.548 & -0.604 & 5 \\ 0.540 & 0.813 & -0.220 & 7 \\ 0.611 & -0.199 & 0.766 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Solution: From the above equations, we find:

$$\phi = ATAN2\left(a_y, a_x\right) = ATAN2(-0.220, -0.604) = 20^\circ \text{ or } 200^\circ$$

Realizing that both the sines and cosines of 20° and 200° can be used for the remainder,

$$\psi = ATAN2\left(-n_x S\phi + n_y C\phi, -o_x S\phi + o_y C\phi\right) = (0.31, 0.952) = 18^\circ \text{ or } 198^\circ$$
$$\theta = ATAN2\left(a_x C\phi + a_y S\phi, a_z\right) = ATAN2(-0.643, 0.766) = -40^\circ \text{ or } 40^\circ$$ ■

2.10.3 Articulated Joints

Articulated joints consist of three rotations other than the above. Similar to section 2.9.4., we will develop the matrix representing articulated joints later, when we discuss the Denavit-Hartenberg representation.

2.11 Forward and Inverse Kinematic Equations: Position and Orientation

The matrix representing the final location and orientation of the robot is a combination of the above, depending on which coordinates are used. If a robot is made of a Cartesian and an RPY set of joints, then the location and the final orientation of the frame relative to the reference frame will be the product of the two matrices representing the Cartesian position change and the RPY. The robot may be represented by:

$$^{R}T_{H} = T_{cart}(p_x, p_y, p_z) \times \text{RPY}(\phi_a, \phi_o, \phi_n) \tag{2.50}$$

If the robot is designed based on spherical coordinates for positioning and Euler angles for orientation, then the final answer will be the following equation, where the position is determined by the spherical coordinates, while the final orientation is affected by both the angles in the spherical coordinates as well as the Euler angles:

$$^{R}T_{H} = T_{sph}(r, \beta, \gamma) \times \text{Euler}(\phi, \theta, \psi) \tag{2.51}$$

The forward and inverse kinematic solutions for these cases are not developed here, since many different combinations are possible. Instead, in complicated designs, the Denavit-Hartenberg representation is recommended. We will discuss this next.

2.12 Denavit-Hartenberg Representation of Forward Kinematic Equations of Robots

In 1955, Denavit and Hartenberg[4] published a paper in the *ASME Journal of Applied Mechanics* that was later used to represent and model robots and to derive their equations of motion. This technique has become the standard way of representing robots and modeling their motions, and therefore, is essential to learn. The Denavit-Hartenberg (D-H) model of representation is a very simple way of modeling robot links and joints that can be used for any robot configuration, regardless of its sequence or complexity. It can also be used to represent transformations in any coordinates we have already discussed, such as Cartesian, cylindrical, spherical, Euler, and RPY. Additionally, it can be used for representation of all-revolute articulated robots, SCARA robots, or any possible combinations of joints and links. Although the direct modeling of robots with the previous techniques are faster and more straightforward, the D-H representation has an added benefit; as we will see later, analysis of differential motions and Jacobians, dynamic analysis, force analysis, and others are based on the results obtained from D-H representation.[5–9]

Robots may be made of a succession of joints and links in any order. The joints may be either prismatic (linear) or revolute (rotational), move in different planes, and have offsets. The links may also be of any length, including zero; may be twisted and bent; and may be in any plane. Therefore, any general set of joints and links may create a robot. We need to be able to model and analyze any robot, whether or not it follows any of the preceding coordinates.

To do this, we assign a reference frame to each joint, and later define a general procedure to transform from one joint to the next (one frame to the next). If we combine all the transformations from the base to the first joint, from the first joint to the second joint, and so on, until we get to the last joint, we will have the robot's total transformation matrix. In the following sections, we will define the general procedure, based on the D-H representation, to assign reference frames to each joint. Then we will define how a transformation between any two successive frames may be accomplished. Finally, we will write the total transformation matrix for the robot.

Imagine that a robot may be made of a number of links and joints in any form. Figure 2.27 represents three successive joints and two links. Although these joints and links are not necessarily similar to any real robot joint or link, they are very general and can easily represent any joints in real robots. These joints may be revolute or prismatic, or both. Although in real robots it is customary to only have 1-DOF joints, the joints in Figure 2.27 represent 1- or 2-DOF joints.

Figure 2.27(a) shows three joints. Each joint may both rotate and/or translate. Let's assign joint number n to the first joint, $n + 1$ to the second joint, and $n + 2$ to the third

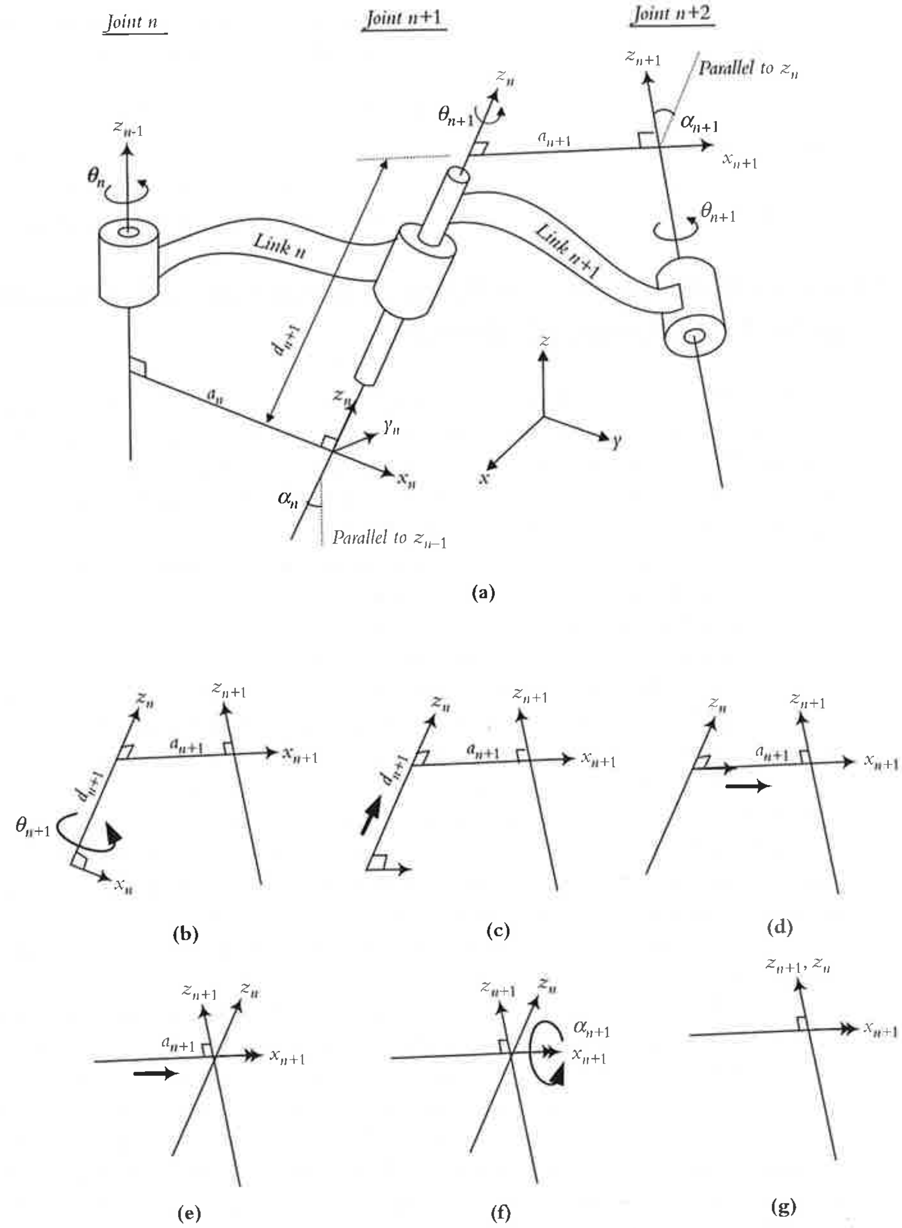


Figure 2.27 A Denavit-Hartenberg representation of a general purpose joint-link combination.

joint shown. There may be other joints before or after these. Each link is also assigned a link number as shown. Link n will be between joints n and $n+1$, and link $n+1$ is between joints $n+1$ and $n+2$.

To model the robot with the D-H representation, the first thing we need to do is assign a local reference frame for each and every joint. Therefore, for each joint, we will have to assign a z-axis and an x-axis. We normally do not need to assign a y-axis, since we always know that y-axes are mutually perpendicular to both x- and z-axes. In addition, the D-H representation does not use the y-axis at all. The following is the procedure for assigning a local reference frame to each joint.

- All joints, without exception, are represented by a z-axis. If the joint is revolute, the z-axis is in the direction of rotation as followed by the right-hand rule for rotations. If the joint is prismatic, the z-axis for the joint is along the direction of the linear movement. In each case, the index number for the z-axis of joint n (as well as the local reference frame for the joint) is $n - 1$. For example, the z-axis representing motions about joint number $n + 1$ is z_n. These simple rules will allow us to quickly assign z-axes to all joints. For revolute joints, the rotation about the z-axis (θ) will be the joint variable. For prismatic joints, the length of the link along the z-axis represented by d will be the joint variable.
- As you can see in Figure 2.27(a), in general, joints may not necessarily be parallel or intersecting. As a result, the z-axes may be skew lines. There is always one line mutually perpendicular to any two skew lines, called common normal, which is the shortest distance between them. We always assign the x-axis of the local reference frame in the direction of the common normal. Therefore, if a_n represents the common normal between z_{n-1} and z_n, the direction of x_n will be along a_n. Similarly, if the common normal between z_n and z_{n+1} is a_{n+1}, the direction of x_{n+1} will be along a_{n+1}. The common normal lines between successive joints are not necessarily intersecting or colinear. As a result, the origins of two successive frames may also not be at the same location. Based on the above, we can assign coordinate frames to all joints, with the following exceptions:
  - If two z-axes are parallel, there are an infinite number of common normals between them. We will pick the common normal that is colinear with the common normal of the previous joint. This will simplify the model.
  - If the z-axes of two successive joints are intersecting, there is no common normal between them (or it has a zero length). We will assign the x-axis along a line perpendicular to the plane formed by the two axes. This means that the common normal is a line perpendicular to the plane containing the two z-axes, which is the equivalent of picking the direction of the cross-product of the two z-axes. This also simplifies the model.

In Figure 2.27(a), θ represents a rotation about the z-axis, d represents the distance on the z-axis between two successive common normals (or joint offset), a represents the length of each common normal (the length of a link), and α represents the angle between two successive z-axes (also called joint twist angle). Commonly, only θ and d are joint variables.

The next step is to follow the necessary motions to transform from one reference frame to the next. Assuming we are at the local reference frame $x_n - z_n$, we will do the following four standard motions to get to the next local reference frame $x_{n+1} - z_{n+1}$:

1. Rotate about the z_n-axis an angle of θ_{n+1} (Figure 2.27(a) and (b)). This will make x_n and x_{n+1} parallel to each other. This is true because a_n and a_{n+1} are both perpendicular to z_n, and rotating z_n an angle of θ_{n+1} will make them parallel (and thus, coplanar).

2. Translate along the z_n-axis a distance of d_{n+1} to make x_n and x_{n+1} colinear (Figure 2.27(c)). Since x_n and x_{n+1} were already parallel and normal to z_n, moving along z_n will lay them over each other.
3. Translate along the (already rotated) x_n-axis a distance of a_{n+1} to bring the origins of x_n and x_{n+1} together (Figure 2.27(d) and (e)). At this point, the origins of the two reference frames will be at the same location.
4. Rotate z_n-axis about x_{n+1}-axis an angle of α_{n+1} to align z_n-axis with z_{n+1}-axis (Figure 2.27(f)). At this point, frames n and $n+1$ will be exactly the same (Figure 2.27(g)), and we have transformed from one to the next.

Continuing with exactly the same sequence of four movements between the $n+1$ and $n+2$ frames will transform one to the next, and by repeating this as necessary, we can transform between successive frames. Starting with the robot's reference frame, we can transform to the first joint, second joint and so on, until the end effector. Note that the above sequence of movements remains the same between any two frames.

The transformation ${}^nT_{n+1}$ (called A_{n+1}) between two successive frames representing the preceding four movements is the product of the four matrices representing them. Since all transformations are relative to the current frame (they are measured and performed relative to the axes of the current local frame), all matrices are post-multiplied. The result is:

$${}^nT_{n+1} = A_{n+1} = Rot(z, \theta_{n+1}) \times Trans(0, 0, d_{n+1}) \times Trans(a_{n+1}, 0, 0) \times Rot(x, \alpha_{n+1})$$

$$= \begin{bmatrix} C\theta_{n+1} & -S\theta_{n+1} & 0 & 0 \\ S\theta_{n+1} & C\theta_{n+1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_{n+1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & a_{n+1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_{n+1} & -S\alpha_{n+1} & 0 \\ 0 & S\alpha_{n+1} & C\alpha_{n+1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.52}$$

$$A_{n+1} = \begin{bmatrix} C\theta_{n+1} & -S\theta_{n+1}C\alpha_{n+1} & S\theta_{n+1}S\alpha_{n+1} & a_{n+1}C\theta_{n+1} \\ S\theta_{n+1} & C\theta_{n+1}C\alpha_{n+1} & -C\theta_{n+1}S\alpha_{n+1} & a_{n+1}S\theta_{n+1} \\ 0 & S\alpha_{n+1} & C\alpha_{n+1} & d_{n+1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.53}$$

Table 2.1 *D-H Parameters Table.*

#	θ	d	a	α
0–1				
1–2				
2–3				
3–4				
4–5				
5–6				

As an example, the transformation between joints 2 and 3 of a generic robot will simply be:

$$^{2}T_{3} = A_3 = \begin{bmatrix} C\theta_3 & -S\theta_3 C\alpha_3 & S\theta_3 S\alpha_3 & a_3 C\theta_3 \\ S\theta_3 & C\theta_3 C\alpha_3 & -C\theta_3 S\alpha_3 & a_3 S\theta_3 \\ 0 & S\alpha_3 & C\alpha_3 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.54}$$

At the base of the robot, we can start with the first joint and transform to the second joint, then to the third, and so on, until the hand of the robot and eventually the end effector. Calling each transformation an A_{n+1}, we will have a number of A matrices that represent the transformations. The total transformation between the base of the robot and the hand will be:

$$^{R}T_{H} = {}^{R}T_{1}\,{}^{1}T_{2}\,{}^{2}T_{3}\ldots{}^{n-1}T_{n} = A_1 A_2 A_3 \ldots A_n \tag{2.55}$$

where n is the joint number. For a 6-DOF robot, there will be six A matrices.

To facilitate the calculation of the A matrices, we will form a table of joint and link parameters, whereby the values representing each link and joint are determined from the schematic drawing of the robot and are substituted into each A matrix. Table 2.1 can be used for this purpose.

In the following examples, we will assign the necessary frames, fill out the parameters tables, and substitute the values into the A matrices. We will start with a simple robot, but will consider more difficult robots later.

Starting with a simple 2-axis robot and moving up to a robot with 6 axes, we will apply the D-H representation in the following examples to derive the forward kinematic equations for each one.

Example 2.23

For the simple 2-axis, planar robot of Figure 2.28, assign the necessary coordinate systems based on the D-H representation, fill out the parameters table, and derive the forward kinematic equations for the robot.

Solution: First, note that both joints rotate in the x-y plane and that a frame $x_H - z_H$ shows the end of the robot. We start by assigning the z-axes for the joints. z_0 will be assigned to joint 1, and z_1 will be assigned to joint 2. Figure 2.28 shows both

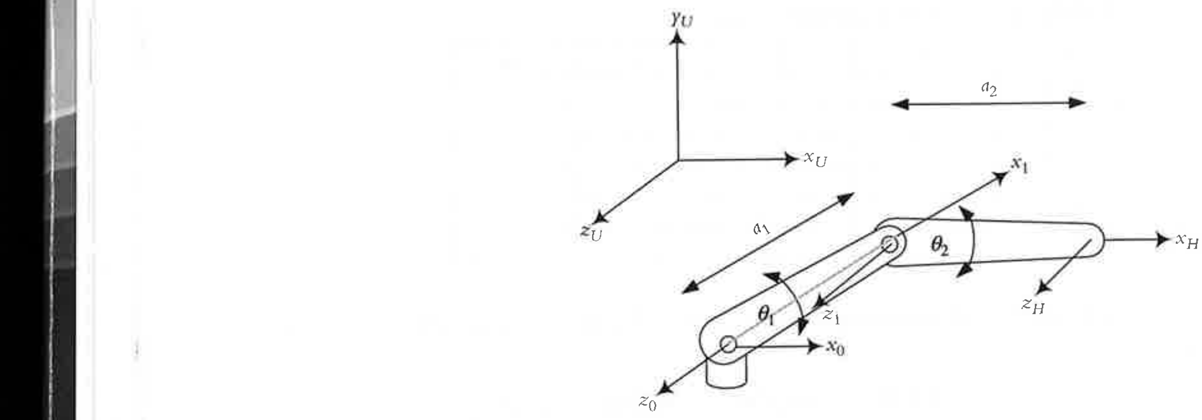


Figure 2.28 A simple 2-axis, articulated robot arm.

z-axes pointing out from the page (as are the z_U- and z_H-axes). Notice that the 0-frame is fixed and does not move. The robot moves relative to it.

Next, we need to assign the x-axes for each frame. Since the first frame (frame 0) is at the base of the robot, and therefore, there are no joints before it, the direction of x_0 is arbitrary. For convenience (only), we may choose to assign it in the same direction as the Universe x-axis. As we will see later, there is no problem if another direction is chosen; all it means is that if we were to specify ${}^U T_H$ instead of ${}^0 T_H$, we would have to include an additional fixed rotation to indicate that x_U- and x_0-axes are not parallel.

Since z_0 and z_1 are parallel, the common normal between them is in the direction between the two, and therefore, the x_1-axis is as shown.

Table 2.2 shows the parameters table for the robot. To identify the values, follow the four necessary transformations required to go from one frame to the next, according to the D-H convention:

1. Rotate about the z_0-axis an angle of θ_1 to make x_0 parallel to x_1.
2. Since x_0 and x_1 are in the same plane, translation d along the z_0-axis is zero.
3. Translate along the (already rotated) x_0-axis a distance of a_1.
4. Since z_0 and z_1-axes are parallel, the necessary rotation α about the x_1-axis is zero.

The same can be repeated for transforming between frames 1 and H.

Note that since there are two revolute joints, the two unknowns are also joint angles θ_1 and θ_2. The forward kinematic equation of the robot can be found by

Table 2.2 *D-H Parameters Table for Example 2.23.*

#	θ	d	a	α
0–1	θ_1	0	a_1	0
1–H	θ_2	0	a_2	0

substituting these parameters into the corresponding A matrices as follows:

$$A_1 = \begin{bmatrix} C_1 & -S_1 & 0 & a_1C_1 \\ S_1 & C_1 & 0 & a_1S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad A_2 = \begin{bmatrix} C_2 & -S_2 & 0 & a_2C_2 \\ S_2 & C_2 & 0 & a_2S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^0T_H = A_1 \times A_2 = \begin{bmatrix} C_1C_2 - S_1S_2 & -C_1S_2 - S_1C_2 & 0 & a_2(C_1C_2 - S_1S_2) + a_1C_1 \\ S_1C_2 + C_1S_2 & -S_1S_2 + C_1C_2 & 0 & a_2(S_1C_2 + C_1S_2) + a_1S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Using functions $C_1C_2 - S_1S_2 = C(\theta_1 + \theta_2) = C_{12}$ and $S_1C_2 + C_1S_2 = S(\theta_1 + \theta_2) = S_{12}$, the transformation simplifies to:

$$^0T_H = \begin{bmatrix} C_{12} & -S_{12} & 0 & a_2C_{12} + a_1C_1 \\ S_{12} & C_{12} & 0 & a_2S_{12} + a_1S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.56}$$

The forward kinematic solution allows us to find the location (and orientation) of the robot's end if values for θ_1, θ_2, a_1, and a_2 are specified. We will find the inverse kinematic solution for this robot later. ■

Example 2.24

Assign the necessary frames to the robot of Figure 2.29 and derive the forward kinematic equation of the robot.

Solution: As you can see, this robot is very similar to the robot of Example 2.23, except that another joint is added to it. The same assignment of frames 0 and 1 are

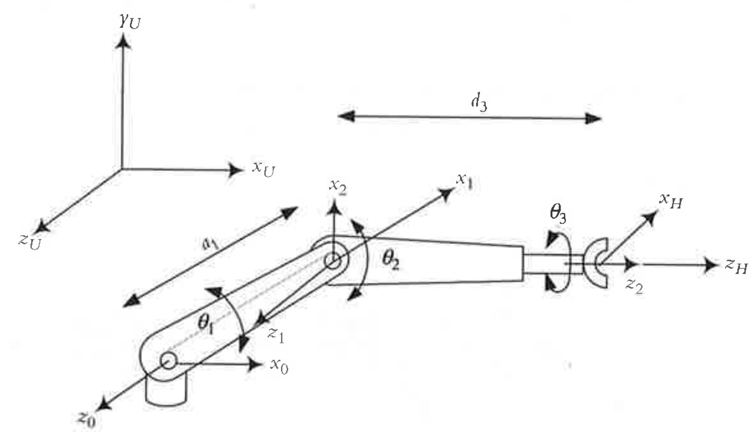


Figure 2.29 The 3-DOF robot of Example 2.24.

Table 2.3 *D-H Parameters Table for Example 2.24.*

#	θ	d	a	α
0–1	θ_1	0	a_1	0
1–2	$90+\theta_2$	0	0	90
2–H	θ_3	d_3	0	0

applicable to this robot, but we need to add another frame for the new joint. Therefore, we will add a z_2-axis perpendicular to the joint, as shown. Since z_1 and z_2 axes intersect at joint 2, x_2-axis will be perpendicular to both at the same location, as shown.

Table 2.3 shows the parameters for the robot. Please follow the four required transformations between every two frames and make sure that you note the following:

- The direction of the H-frame is changed to represent the motions of the gripper.
- The physical length of link 2 is now a "d" and not an "a".
- Joint 3 is shown as a revolute joint. In this case, d_3 is fixed. However, the joint could have been a prismatic joint (in which case, d_3 would be a variable but θ_3 would be fixed), or both (in which case both θ_3 and d_3 would be variables).
- Remember that the rotations are measured with the right-hand rule. The curled fingers of your right hand, rotating in the direction of rotation, determine the direction of the axis of rotation along the thumb.
- Note that the rotation about z_1 is shown to be $90° + \theta_2$ and not θ_2. This is because even when θ_2 is zero, there is a 90° angle between x_1 and x_2 (see Figure 2.30). This is an extremely important factor in real life, when the reset position of the robot must be defined.

Noting that $\sin(90+\theta) = \cos(\theta)$ and $\cos(90+\theta) = -\sin(\theta)$, the matrices representing each joint transformation and the total transformation of the robot are:

$$A_1 = \begin{bmatrix} C_1 & -S_1 & 0 & a_1C_1 \\ S_1 & C_1 & 0 & a_1S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_2 = \begin{bmatrix} -S_2 & 0 & C_2 & 0 \\ C_2 & 0 & S_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_3 = \begin{bmatrix} C_3 & -S_3 & 0 & 0 \\ S_3 & C_3 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0T_H = A_1A_2A_3$$

$$= \begin{bmatrix} (-C_1S_2 - S_1C_2)C_3 & -(-C_1S_2 - S_1C_2)S_3 & C_1C_2 - S_1S_2 & (C_1C_2 - S_1S_2)d_3 + a_1C_1 \\ (C_1C_2 - S_1S_2)C_3 & -(C_1C_2 - S_1S_2)S_3 & C_1S_2 + S_1C_2 & (C_1S_2 + S_1C_2)d_3 + a_1S_1 \\ S_3 & C_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

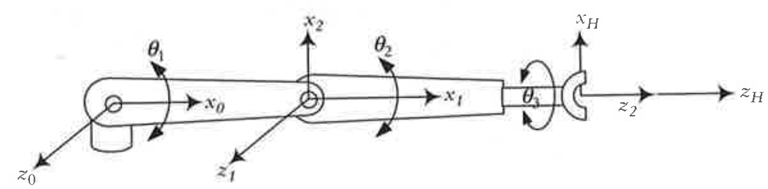


Figure 2.30 Robot of Example 2.24 in reset position.

Simplifying the matrix with $C_1C_2 - S_1S_2 = C_{12}$ and $S_1C_2 + C_1S_2 = S_{12}$, we get:

$$
{}^{0}T_H = A_1A_2A_3 = \begin{bmatrix} -S_{12}C_3 & S_{12}S_3 & C_{12} & C_{12}d_3 + a_1C_1 \\ C_{12}C_3 & -C_{12}S_3 & S_{12} & S_{12}d_3 + a_1S_1 \\ S_3 & C_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
\text{For} \quad \begin{cases} \theta_1 = 0 \\ \theta_2 = 0, \\ \theta_3 = 0 \end{cases} \qquad {}^{0}T_H = \begin{bmatrix} 0 & 0 & 1 & d_3 + a_1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \text{and}
$$

$$
\text{for} \quad \begin{cases} \theta_1 = 90 \\ \theta_2 = 0 \;, \\ \theta_3 = 0 \end{cases} \qquad {}^{0}T_H = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & d_3 + a_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Please verify that these values represent the robot correctly. ■

Example 2.25

For the simple 6-DOF robot of Figure 2.31, assign the necessary coordinate frames based on the D-H representation, fill out the accompanying parameters table, and derive the forward kinematic equation of the robot.

Solution: As you will notice, when the number of joints increases, in this case to six, the analysis of the forward kinematics becomes more complicated. However, all principles apply the same as before. You will also notice that this 6-DOF robot is still simplified with no joint offsets or twist angles. In this example, for simplicity, we are assuming that joints 2, 3, and 4 are in the same plane, which will render their d_n values zero; otherwise, the presence of offsets will make the equations slightly more

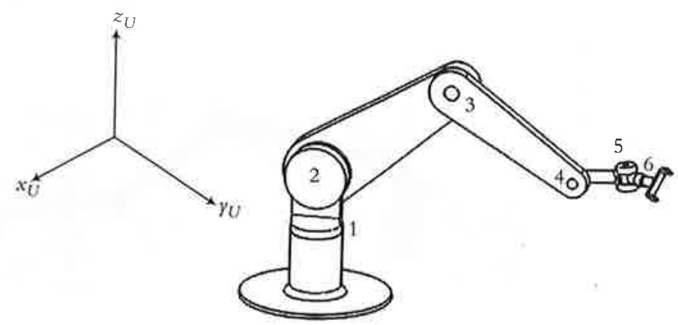


Figure 2.31 A simple 6-DOF articulate robot.

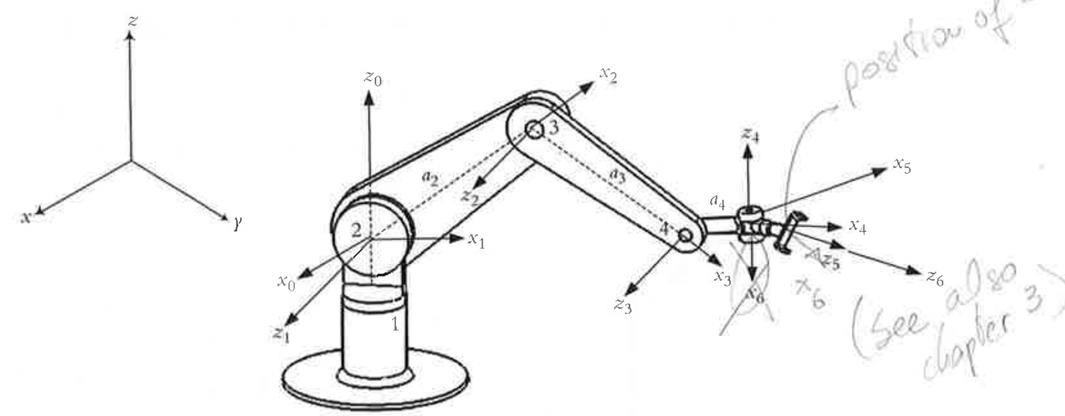


Figure 2.32 Reference frames for the simple 6-DOF articulate robot.

involved. Generally, offsets will change the position terms, but not orientation terms. To assign coordinate frames to the robot, we will first look for the joints (as shown). First, we will assign z-axes to each joint, followed by x-axes. Please follow the coordinates as shown in Figures 2.32 and 2.33. Figure 2.33 is a line drawing of the robot in Figure 2.31 for simplicity. Notice where the origin of each frame is, and why.

Start at joint 1. z_0 represents motions about the first joint. x_0 is chosen to be parallel to the reference frame x-axis. This is done only for convenience. x_0 is a fixed axis, representing the base of the robot, and does not move. The movement of the first joint *occurs around the* $z_0 - x_0$ axes. Next, z_1 is assigned at joint 2. x_1 will be normal to z_0 and z_1 because these two axes are intersecting. x_2 will be in the direction of the common normal between z_1 and z_2. x_3 is in the direction of the common normal between z_2 and z_3. Similarly, x_4 is in the direction of the common normal between z_3 and z_4. Finally, z_5 and z_6 are as shown, because they are parallel and colinear. z_5 represents the motions about joint 6, while z_6 represents the motions of the end effector. Although we normally do not include the end effector in the equations of motion, it is necessary to include the end effector frame because it will allow us to transform out of frame $z_5 - x_5$. Also important to notice is the location of the origins

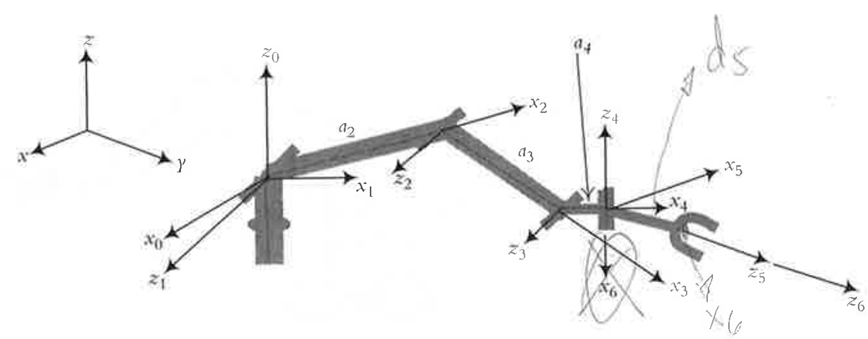


Figure 2.33 Line drawing of the reference frames for the simple 6-DOF articulate robot.

of the first and the last frames. This will determine the total transformation equation of the robot. You may be able to assign other (or different) intermediate coordinate frames between the first and the last, but as long as the first and the last frames are not changed, the total transformation of the robot will be the same. Notice that the origin of the first joint is *not* at the physical location of the joint. You can verify that this is correct, because physically, whether the actual joint is a little higher or lower will not make any difference in the robot's movements. Therefore, the origin can be as shown without regard to the physical location of the joint on the base. Note that we could have chosen to place the 0-frame at the base of the robot. In that case, the total transformation between the base and the end effector of the robot would have included the height of the robot too, whereas the way we have assigned the base frame, our measurements are relative to the present 0-frame. We can simply add the height to our equation later.

Next, we will follow the assigned coordinate frames to fill out the parameters of Table 2.4. Starting with $z_0 - x_0$, there will be a rotation of θ_1 to bring x_0 to x_1, a translation of zero along z_1 and zero along x_1 to align the x's together, and a rotation of $\alpha_1 = +90°$ to bring z_0 to z_1. Remember that the rotations are measured with the right-hand rule. The curled fingers of your right hand, rotating in the direction of rotation, determine the direction of the axis of rotation along the thumb. At this point, we will be at $z_1 - x_1$. Continue with the next joints the same way to fill out the table.

You *must* realize that like any other machine, robots do not stay in one configuration as shown in a drawing. You need to visualize the motions, even though the schematic is in 2-D. This means you must realize that the different links and joints of the robot move, as do the frames attached to them. If in this instant, due to the configuration in which the robot is drawn, the axes are shown to be in a particular position and orientation, they will be at other points and orientations as the robot moves. For example, x_3 is always in the direction of a_3 along the line between joints 3 and 4. As the lower arm of the robot rotates about joint 3, x_3 moves as well, but not x_2. However, x_2 will move as the upper arm rotates about joint 2. This must be kept in mind as we determine the parameters.

θ represents the joint variable for a revolute joint and d represents joint variable for a prismatic joint. Since this is an all-revolute robot, all joint variables are angles.

The transformations between each two successive joints can be written by simply substituting the parameters from the parameters table into the A-matrix.

Table 2.4 *Parameters for the Robot of Example 2.25.*

#	θ	d	a	α
0–1	θ_1	0	0	90
1–2	θ_2	0	a_2	0
2–3	θ_3	0	a_3	0
3–4	θ_4	0	a_4	−90
4–5	θ_5	0	0	90
5–6	θ_6	0	0	0

We get:

$$A_1 = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_2 = \begin{bmatrix} C_2 & -S_2 & 0 & C_2a_2 \\ S_2 & C_2 & 0 & S_2a_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_3 = \begin{bmatrix} C_3 & -S_3 & 0 & C_3a_3 \\ S_3 & C_3 & 0 & S_3a_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} C_4 & 0 & -S_4 & C_4a_4 \\ S_4 & 0 & C_4 & S_4a_4 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_5 = \begin{bmatrix} C_5 & 0 & S_5 & 0 \\ S_5 & 0 & -C_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_6 = \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.57}$$

Once again, to simplify the final solutions, we will use the following trigonometric functions:

$$\begin{aligned} S\theta_1C\theta_2 + C\theta_1S\theta_2 &= S(\theta_1 + \theta_2) = S_{12} \\ C\theta_1C\theta_2 - S\theta_1S\theta_2 &= C(\theta_1 + \theta_2) = C_{12} \end{aligned} \tag{2.58}$$

The total transformation between the base of the robot (where the 0-frame is) and the hand will be:

$$\begin{aligned} {}^{R}T_{H} &= A_1A_2A_3A_4A_5A_6 \\ &= \begin{bmatrix} C_1(C_{234}C_5C_6 - S_{234}S_6) & C_1(-C_{234}C_5C_6 - S_{234}C_6) & C_1(C_{234}S_5) + S_1C_5 & C_1(C_{234}a_4 + C_{23}a_3 + C_2a_2) \\ -S_1S_5C_6 & +S_1S_5S_6 & & \\ S_1(C_{234}C_5C_6 - S_{234}S_6) & S_1(-C_{234}C_5C_6 - S_{234}C_6) & S_1(C_{234}S_5) - C_1C_5 & S_1(C_{234}a_4 + C_{23}a_3 + C_2a_2) \\ +C_1S_5C_6 & -C_1S_5S_6 & & \\ S_{234}C_5C_6 + C_{234}S_6 & -S_{234}C_5C_6 + C_{234}C_6 & S_{234}S_5 & S_{234}a_4 + S_{23}a_3 + S_2a_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{2.59}$$

■

Note the following important insights:

1. In assigning the x- and z-axes, you may choose either direction along the chosen line of action. Ultimately, the result of the total transformation will be the same. However, your individual matrices and parameters are similarly affected.
2. It is acceptable to use additional frames to make things easier to follow. However, you may not have any fewer or more unknown variables than you have joints.
3. The D-H representation does not use a transformation along the y-axis. Therefore, if you find that you need to move along the y-axis to transform from one frame to another, you either have made a mistake, or you need an additional frame in between.
4. In reality, there may be small angles between parallel z-axes due to manufacturing errors or tolerances. To represent these errors between seemingly parallel z-axes, it

will be necessary to make transformations along the y-axis. Therefore, the D-H methodology cannot represent these errors.

5. Note that frame $x_n - z_n$ represents link n before itself. It is attached to link n and moves with it relative to frame $n-1$. Motions about joint n are relative to frame $n-1$.
6. Obviously, you may use other representations to develop the kinematic equations of a robot. However, in order to be able to use subsequent derivations that will be used for differential motions, dynamic analysis, and so on—which are all based on the D-H representation—you may benefit from following this methodology.
7. So far, in all of our examples in this section, we derived the transformation between the base of the robot and the end effector (${}^{0}T_{H}$). It is also possible to desire the transformation between the Universe frame and the end effector (${}^{U}T_{H}$). In that case, we will need to pre-multiply ${}^{0}T_{H}$ by the transformation between the base and the Universe frames, or ${}^{U}T_{H} = {}^{U}T_{0} \times {}^{0}T_{H}$. Since the location of the base of the robot is always known, this will not add to the number of unknowns (or complexity of the problem). The transformation ${}^{U}T_{0}$ usually involves simple translations and rotations about the Universe frame to get to the base frame. This process is not based on the D-H representation; it is a simple set of rotations and translations.
8. As you have probably noticed, the D-H representation can be used for any configuration of joints and links, whether or not they follow known coordinates such as rectangular, spherical, Euler, and so on. Additionally, you cannot use those representations if any twist angles or joint offsets are present. In reality, twist angles and joint offsets are very common. The derivation of kinematic equations based on rectangular, cylindrical, spherical, RPY, and Euler was presented only for teaching purposes. Therefore, you should normally use the D-H for analysis.

Example 2.26

The Stanford Arm: Assign coordinate frames to the Stanford Arm (Figure 2.34) and fill out the parameters table. The Stanford Arm is a spherical coordinate arm: the first two joints are revolute, the third is prismatic, and the last three wrist joints are revolute joints.

Solution: To allow you to work on this before you see the solution, the answer to this problem is included at the end of this chapter. It is recommended that before you look at the assignment of the frames and the solution of the Arm, you try to do this on your own.

The final forward kinematic solution of the Arm[5] is the product of the six matrices representing the transformation between successive joints, as follows:

$$
{}^{R}T_{H\,\mathrm{Stanford}} = {}^{0}T_{6} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

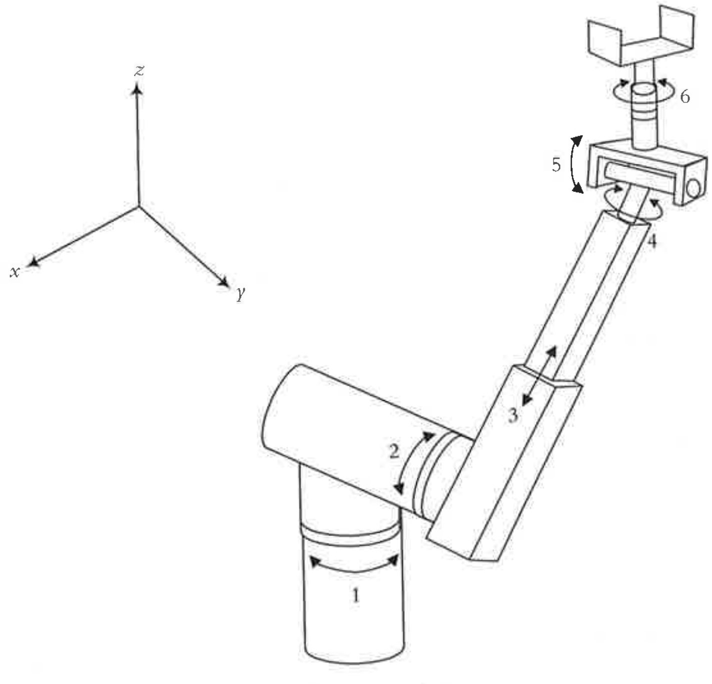


Figure 2.34 Schematic drawing of the Stanford Arm.

where

$$
\begin{aligned}
n_x &= C_1[C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6] - S_1(S_4C_5C_6 + C_4S_6) \\
n_y &= S_1[C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6] + C_1(S_4C_5C_6 + C_4S_6) \\
n_z &= -S_2(C_4C_5C_6 - S_4S_6) - C_2S_5C_6 \\
o_x &= C_1[-C_2(C_4C_5S_6 + S_4C_6) + S_2S_5S_6] - S_1(-S_4C_5S_6 + C_4C_6) \\
o_y &= S_1[-C_2(C_4C_5S_6 + S_4C_6) + S_2S_5S_6] + C_1(-S_4C_5S_6 + C_4C_6) \\
o_z &= S_2(C_4C_5S_6 + S_4C_6) + C_2S_5S_6 \\
a_x &= C_1(C_2C_4S_5 + S_2C_5) - S_1S_4S_5 \\
a_y &= S_1(C_2C_4S_5 + S_2C_5) + C_1S_4S_5 \\
a_z &= -S_2C_4S_5 + C_2C_5 \\
p_x &= C_1S_2d_3 - S_1d_2 \\
p_y &= S_1S_2d_3 + C_1d_2 \\
p_z &= C_2d_3
\end{aligned}
\tag{2.60}
$$

■

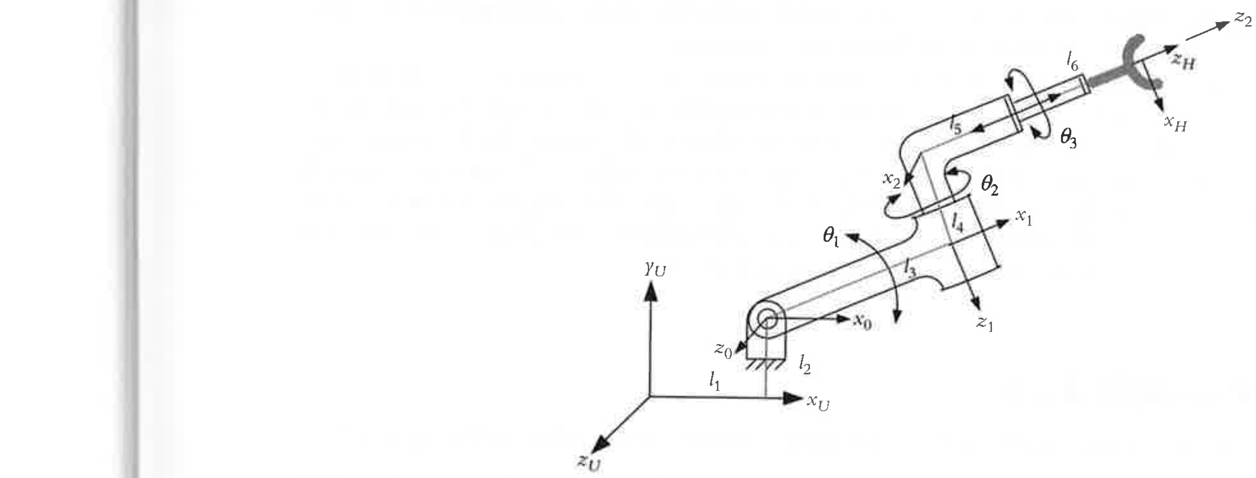


Figure 2.35 4-axis robot of Example 2.27.

Example 2.27

Assign required frames to the 4-axis robot of Figure 2.35 and write an equation describing ${}^{U}T_H$.

Solution: This example shows a robot with a twist angle, a joint offset, and a double-action joint represented by the same z-axis. Applying the standard procedure, we assign the frames. The parameters table is shown in Table 2.5.

The total transformation is:

$$ {}^{U}T_H = {}^{U}T_0 \times {}^{0}T_H = \begin{bmatrix} 1 & 0 & 0 & l_1 \\ 0 & 1 & 0 & l_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times A_1 A_2 A_H $$

Table 2.5 *Parameters for the Robot of Example 2.27.*

#	θ	d	a	α
0–1	θ_1	0	l_3	90
1–2	θ_2	$-l_4$	0	90
2–H	θ_3	$l_5 + l_6$	0	0

■

2.13 The Inverse Kinematic Solution of Robots

As we mentioned earlier, we are actually interested in the inverse kinematic solutions. With inverse kinematic solutions, we will be able to determine the value of each joint in order to place the robot at a desired position and orientation. We have already seen the inverse kinematic solutions of specific coordinate systems. In this section, we will learn a general procedure for solving the kinematic equations.

As you have noticed by now, the forward kinematic equations have a multitude of coupled angles such as C_{234}. This makes it impossible to find enough elements in the matrix to solve for individual sines and cosines to calculate the angles. To de-couple some of the angles, we may multiply the ${}^{R}T_H$ matrix with individual A_n^{-1} matrices. This will yield one side of the equation free of an individual angle, allowing us to find elements that yield sines and cosines of the angle, and subsequently, the angle itself. We will demonstrate the procedure in the following section.

Example 2.28

Find a symbolic expression for the joint variables of the robot of Example 2.23.

Solution: The forward kinematic equation for the robot is shown as Equation (2.56), repeated here. Assume that we desire to place the robot at a position—and consequently, an orientation—given as **n**, **o**, **a**, **p** vectors:

$$ {}^{0}T_H = A_1 \times A_2 = \begin{bmatrix} C_{12} & -S_{12} & 0 & a_2C_{12} + a_1C_1 \\ S_{12} & C_{12} & 0 & a_2S_{12} + a_1S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.56} $$

Since this robot has only two degrees of freedom, its solution is relatively simple. We can solve for the angles either algebraically, or by de-coupling the unknowns. We will do both for comparison. Remember that whenever possible, we should look for values of both the sine and cosine of an angle in order to correctly identify the quadrant in which the angle falls.

I. **Algebraic solution:** Equating elements (2,1), (1,1), (1,4), and (2,4) of the two matrices, we get:

$$ S_{12} = n_y \text{ and } C_{12} = n_x \rightarrow \theta_{12} = ATAN2\,(n_y, n_x) $$

$$ a_2C_{12} + a_1C_1 = p_x \text{ or } a_2n_x + a_1C_1 = p_x \rightarrow C_1 = \frac{p_x - a_2n_x}{a_1} $$

$$ a_2S_{12} + a_1S_1 = p_y \text{ or } a_2n_y + a_1S_1 = p_y \rightarrow S_1 = \frac{p_y - a_2n_y}{a_1} $$

$$ \theta_1 = ATAN2(S_1, C_1) = ATAN2\left(\frac{p_y - a_2n_y}{a_1}, \frac{p_x - a_2n_x}{a_1}\right) $$

Since θ_1 and θ_{12} are known, θ_2 can also be calculated.

II. Alternative solution: In this case, we will post-multiply both sides of Equation (2.56) by A_2^{-1} to de-couple θ_1 from θ_2. We get:

$$A_1 \times A_2 \times A_2^{-1} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times A_2^{-1} \quad \text{or} \quad A_1 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times A_2^{-1}$$

$$\begin{bmatrix} C_1 & -S_1 & 0 & a_1C_1 \\ S_1 & C_1 & 0 & a_1S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} C_2 & -S_2 & 0 & -a_2 \\ S_2 & C_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} C_1 & -S_1 & 0 & a_1C_1 \\ S_1 & C_1 & 0 & a_1S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C_2n_x - S_2o_x & S_2n_x + C_2o_x & a_x & p_x - a_2n_x \\ C_2n_y - S_2o_y & S_2n_y + C_2o_y & a_y & p_y - a_2n_y \\ C_2n_z - S_2o_z & S_2n_z + C_2o_z & a_z & p_z - a_2n_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

From elements 1,4 and 2,4 we get $a_1C_1 = p_x - a_2n_x$ and $a_1S_1 = p_y - a_2n_y$ which is exactly what we got from the other method. Knowing S_1 and C_1, we can find expressions for S_2 and C_2. ■

2.13.1 General Solution for Articulated Robot Arms

In this section, a summary of a technique is presented that may generally be used for inverse kinematic analysis of manipulators.[5] The process is applied to the simple manipulator arm of Example 2.25. Although this solution is for this particular robot with the given configuration, it may similarly be repeated for other robots. As we saw in Example 2.25, the final equation representing the robot, repeated here, is:

$${}^RT_H = A_1A_2A_3A_4A_5A_6$$

$$= \begin{bmatrix} C_1(C_{234}C_5C_6 - S_{234}S_6) & C_1(-C_{234}C_5C_6 - S_{234}C_6) & C_1(C_{234}S_5) + S_1C_5 & C_1(C_{234}a_4 + C_{23}a_3 + C_2a_2) \\ -S_1S_5C_6 & +S_1S_5S_6 & & \\ S_1(C_{234}C_5C_6 - S_{234}S_6) & S_1(-C_{234}C_5C_6 - S_{234}C_6) & S_1(C_{234}S_5) - C_1C_5 & S_1(C_{234}a_4 + C_{23}a_3 + C_2a_2) \\ +C_1S_5C_6 & -C_1S_5S_6 & & \\ S_{234}C_5C_6 + C_{234}S_6 & -S_{234}C_5C_6 + C_{234}C_6 & S_{234}S_5 & S_{234}a_4 + S_{23}a_3 + S_2a_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

We will denote the above matrix as [*RHS*] (Right-Hand Side) for simplicity in writing. Let's, once again, express the desired location and orientation of the robot with:

$$ {}^{R}T_{H} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.61} $$

To solve for the angles, we will pre-multiply these two matrices with selected A_n^{-1} matrices, first with A_1^{-1}:

$$ A_1^{-1} \times \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = A_1^{-1}[\text{RHS}] = A_2A_3A_4A_5A_6 \tag{2.62} $$

$$ \begin{bmatrix} C_1 & S_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ S_1 & -C_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = A_2A_3A_4A_5A_6 $$

$$ \begin{bmatrix} n_xC_1 + n_yS_1 & o_xC_1 + o_yS_1 & a_xC_1 + a_yS_1 & p_xC_1 + p_yS_1 \\ n_z & o_z & a_z & p_z \\ n_xS_1 - n_yC_1 & o_xS_1 - o_yC_1 & a_xS_1 - a_yC_1 & p_xS_1 - p_yC_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ = \begin{bmatrix} C_{234}C_5C_6 - S_{234}S_6 & -C_{234}C_5C_6 - S_{234}C_6 & C_{234}S_5 & C_{234}a_4 + C_{23}a_3 + C_2a_2 \\ S_{234}C_5C_6 + C_{234}S_6 & -S_{234}C_5C_6 + C_{234}C_6 & S_{234}S_5 & S_{234}a_4 + S_{23}a_3 + S_2a_2 \\ -S_5C_6 & S_5S_6 & C_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.63} $$

From the 3,4 elements of Equation (2.63):

$$ p_xS_1 - p_yC_1 = 0 \rightarrow \theta_1 = \tan^{-1}\left(\frac{p_y}{p_x}\right) \text{ and } \theta_1 = \theta_1 + 180^\circ \tag{2.64} $$

From the 1,4 and 2,4 elements, we will get:

$$ \begin{aligned} p_xC_1 + p_yS_1 &= C_{234}a_4 + C_{23}a_3 + C_2a_2 \\ p_z &= S_{234}a_4 + S_{23}a_3 + S_2a_2 \end{aligned} \tag{2.65} $$

We will rearrange the two expressions in Equation (2.65) and square and add them to get:

$$(p_xC_1 + p_yS_1 - C_{234}a_4)^2 = (C_{23}a_3 + C_2a_2)^2$$
$$(p_z - S_{234}a_4)^2 = (S_{23}a_3 + S_2a_2)^2$$
$$(p_xC_1 + p_yS_1 - C_{234}a_4)^2 + (p_z - S_{234}a_4)^2 = a_2^2 + a_3^2 + 2a_2a_3(S_2S_{23} + C_2C_{23})$$

Referring to the trigonometric functions of Equation (2.58):

$$S_2S_{23} + C_2C_{23} = \mathrm{Cos}[(\theta_2 + \theta_3) - \theta_2] = \mathrm{Cos}\theta_3$$

Therefore:

$$C_3 = \frac{(p_xC_1 + p_yS_1 - C_{234}a_4)^2 + (p_z - S_{234}a_4)^2 - a_2^2 - a_3^2}{2a_2a_3} \tag{2.66}$$

In this equation, everything is known except for S_{234} and C_{234}, which we will find next. Knowing that $S_3 = \pm\sqrt{1 - C_3^2}$, we can then say that:

$$\theta_3 = \tan^{-1}\frac{S_3}{C_3} \tag{2.67}$$

Since joints 2, 3, and 4 are parallel, additional pre-multiplications by A_2^{-1} and A_3^{-1} will not yield useful results. The next step is to pre-multiply by the inverses of A_1 through A_4, which results in:

$$A_4^{-1}A_3^{-1}A_2^{-1}A_1^{-1} \times \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = A_4^{-1}A_3^{-1}A_2^{-1}A_1^{-1}[\mathrm{RHS}] = A_5A_6 \tag{2.68}$$

which yields:

$$\begin{bmatrix} C_{234}(C_1n_x + S_1n_y) + S_{234}n_z & C_{234}(C_1o_x + S_1o_y) + S_{234}o_z & C_{234}(C_1a_x + S_1a_y) + S_{234}a_z & C_{234}\left(C_1p_x + S_1p_y\right) + S_{234}p_z - C_{34}a_2 - C_4a_3 - a_4 \\ C_1n_y - S_1n_x & C_1o_y - S_1o_x & C_1a_y - S_1a_x & 0 \\ -S_{234}(C_1n_x + S_1n_y) + C_{234}n_z & -S_{234}(C_1o_x + S_1o_y) + C_{234}o_z & -S_{234}(C_1a_x + S_1a_y) + C_{234}a_z & -S_{234}\left(C_1p_x + S_1p_y\right) + C_{234}p_z + S_{34}a_2 + S_4a_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} C_5C_6 & -C_5S_6 & S_5 & 0 \\ S_5C_6 & -S_5S_6 & -C_5 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.69}$$

From the 3,3 elements of the matrices in Equation (2.69):

$$-S_{234}(C_1a_x+S_1a_y)+C_{234}a_z=0\rightarrow\theta_{234}=\tan^{-1}\left(\frac{a_z}{C_1a_x+S_1a_y}\right)\text{ and }\theta_{234}=\theta_{234}+180^\circ \tag{2.70}$$

and we can calculate S_{234} and C_{234}, which are used to calculate θ_3, as previously discussed.

Now, referring again to Equation (2.65), repeated here, we can calculate the sine and cosine of θ_2 as follows:

$$\begin{cases} p_xC_1+p_yS_1=C_{234}a_4+C_{23}a_3+C_2a_2 \\ p_z=S_{234}a_4+S_{23}a_3+S_2a_2 \end{cases}$$

Since $C_{12}=C_1C_2-S_1S_2$ and $S_{12}=S_1C_2+C_1S_2$, we get:

$$\begin{cases} p_xC_1+p_yS_1-C_{234}a_4=(C_2C_3-S_2S_3)a_3+C_2a_2 \\ p_z-S_{234}a_4=(S_2C_3+C_2S_3)a_3+S_2a_2 \end{cases} \tag{2.71}$$

Treating this as a set of two equations and two unknowns and solving for C_2 and S_2, we get:

$$\begin{cases} S_2=\dfrac{(C_3a_3+a_2)(p_z-S_{234}a_4)-S_3a_3(p_xC_1+p_yS_1-C_{234}a_4)}{(C_3a_3+a_2)^2+S_3^2a_3^2} \\ C_2=\dfrac{(C_3a_3+a_2)(p_xC_1+p_yS_1-C_{234}a_4)+S_3a_3(p_z-S_{234}a_4)}{(C_3a_3+a_2)^2+S_3^2a_3^2} \end{cases} \tag{2.72}$$

Although this is a large equation, all its elements are known and it can be evaluated. Then:

$$\theta_2=\tan^{-1}\frac{(C_3a_3+a_2)(p_z-S_{234}a_4)-S_3a_3(p_xC_1+p_yS_1-C_{234}a_4)}{(C_3a_3+a_2)(p_xC_1+p_yS_1-C_{234}a_4)+S_3a_3(p_z-S_{234}a_4)} \tag{2.73}$$

Now that θ_2 and θ_3 are known:

$$\theta_4=\theta_{234}-\theta_2-\theta_3 \tag{2.74}$$

Remember that since there are two solutions for θ_{234} (Equation (2.70)), there will be two solutions for θ_4 as well. From 1,3 and 2,3 elements of Equation (2.69), we get:

$$\begin{cases} S_5=C_{234}(C_1a_x+S_1a_y)+S_{234}a_z \\ C_5=-C_1a_y+S_1a_x \end{cases} \tag{2.75}$$

$$\text{and}\quad \theta_5=\tan^{-1}\frac{C_{234}(C_1a_x+S_1a_y)+S_{234}a_z}{S_1a_x-C_1a_y} \tag{2.76}$$

As you have probably noticed, there is no de-coupled equation for θ_6. As a result, we have to pre-multiply Equation (2.69) by the inverse of A_5 to de-couple it. We get:

$$\begin{bmatrix} \begin{matrix} C_5\left[C_{234}\left(C_1 n_x + S_1 n_y\right) + S_{234} n_z\right] \\ -S_5\left(S_1 n_x - C_1 n_y\right) \end{matrix} & \begin{matrix} C_5\left[C_{234}\left(C_1 o_x + S_1 o_y\right) + S_{234} o_z\right] \\ -S_5\left(S_1 o_x - C_1 o_y\right) \end{matrix} & 0 & 0 \\ -S_{234}\left(C_1 n_x + S_1 n_y\right) + C_{234} n_z & -S_{234}\left(C_1 o_x + S_1 o_y\right) + C_{234} o_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.77}$$

From 2,1 and 2,2 elements of Equation (2.77) we get:

$$\theta_6 = \tan^{-1} \frac{-S_{234}\left(C_1 n_x + S_1 n_y\right) + C_{234} n_z}{-S_{234}\left(C_1 o_x + S_1 o_y\right) + C_{234} o_z} \tag{2.78}$$

Therefore, we have found six equations that collectively yield the values needed to place and orientate the robot at any desired location. Although this solution is only good for the given robot, a similar approach may be taken for any other robot.

It is important to notice that this solution is only possible because the last three joints of the robot are intersecting at a common point. Otherwise, it will not be possible to solve for this kind of solution, and as a result, we would have to solve the matrices directly or by calculating the inverse of the matrix and solving for the unknowns. Most industrial robots have intersecting wrist joints.

2.14 Inverse Kinematic Programming of Robots

The equations we found for solving the inverse kinematic problem of robots can directly be used to drive the robot to a desired position. In fact, no robot would actually use the forward kinematic equations in order to solve for these results. The only equations that are used are the set of six (or less, depending on the number of joints) equations that calculate the joint values. In other words, the robot designer must calculate the inverse solution and derive these equations and, in turn, use them to drive the robot to position. This is necessary for the practical reason that it takes a long time

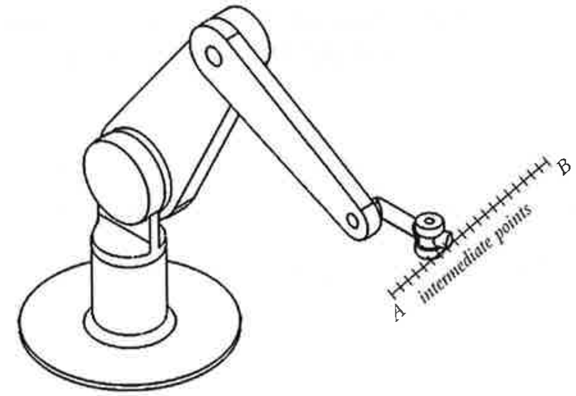


Figure 2.36 Small sections of movement for straight line motions.

for a computer to calculate the inverse of the forward kinematic equations or to substitute values into them and calculate the unknowns (joint variables) by methods such as Gaussian elimination.

For a robot to move in a predictable path, say a straight line, it is necessary to recalculate joint variables many times a second. Imagine that a robot needs to move in a straight line between a starting point A and a destination point B. If no other action is taken and the robot moves from point A to point B, the path is unpredictable. The robot moves all its joints until they are at the final value, which will place the robot at the destination point B. However, depending on the rate of change in each joint, the hand will follow an unknown path in between the two points. To make the robot follow a straight line, it is necessary to break the line into many small sections (Figure 2.36) and make the robot follow those very small sections sequentially between the two points. This means that a new solution must be calculated for each small section. Typically, the location may be recalculated between 50 to 200 times a second. This means that if calculating a solution takes more than 5 to 20 ms, the robot will lose accuracy or will not follow the specified path.[10] The shorter the time it takes to calculate a new solution, the more accurate the robot. As a result, it is vital to eliminate as many unnecessary computations as possible to allow the computer controller to calculate more solutions. This is why the designer must do all mathematical manipulations beforehand and only program the robot controller to calculate the final solutions. This will be discussed in more detail in Chapter 5.

For the 6-axis robot discussed earlier, given the final desired location and orientation as:

$$^{R}T_{H_{Desired}} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

all the controller needs to use to calculate the unknown angles is the set of inverse solutions as summarized below:

$$\begin{aligned}
&\theta_1 = \tan^{-1}\left(\frac{p_y}{p_x}\right) \quad \text{and} \quad \theta_1 = \theta_1 + 180^\circ \\
&\theta_{234} = \tan^{-1}\left(\frac{a_z}{C_1 a_x + S_1 a_y}\right) \quad \text{and} \quad \theta_{234} = \theta_{234} + 180^\circ \\
&C_3 = \frac{(p_x C_1 + p_y S_1 - C_{234} a_4)^2 + (p_z - S_{234} a_4)^2 - a_2^2 - a_3^2}{2 a_2 a_3} \\
&S_3 = \pm\sqrt{1 - C_3^2} \\
&\theta_3 = \tan^{-1}\frac{S_3}{C_3} \\
&\theta_2 = \tan^{-1}\frac{(C_3 a_3 + a_2)(p_z - S_{234} a_4) - S_3 a_3 (p_x C_1 + p_y S_1 - C_{234} a_4)}{(C_3 a_3 + a_2)(p_x C_1 + p_y S_1 - C_{234} a_4) + S_3 a_3 (p_z - S_{234} a_4)} \\
&\theta_4 = \theta_{234} - \theta_2 - \theta_3 \\
&\theta_5 = \tan^{-1}\frac{C_{234}(C_1 a_x + S_1 a_y) + S_{234} a_z}{S_1 a_x - C_1 a_y} \\
&\theta_6 = \tan^{-1}\frac{-S_{234}(C_1 n_x + S_1 n_y) + C_{234} n_z}{-S_{234}(C_1 o_x + S_1 o_y) + C_{234} o_z}
\end{aligned} \tag{2.79}$$

Although this is not trivial, it is much quicker to use these equations and calculate the angles than it is to invert the matrices or do Gaussian elimination. Notice that all operations in this computation are simple arithmetic or trigonometric operations.

2.15 Degeneracy and Dexterity

2.15.1 Degeneracy

Degeneracy occurs when the robot loses a degree of freedom, and therefore, cannot perform as desired.[11] This occurs under two conditions: (1) when the robot's joints reach their physical limits and as a result, cannot move any further; (2) a robot may become degenerate in the middle of its workspace if the z-axes of two similar joints become colinear. This means that, at this instant, whichever joint moves, the same motion will result, and consequently, the controller does not know which joint to move. Since in either case the total number of degrees of freedom available is less than six, there is no solution for the robot. In the case of colinear joints, the determinant of the position matrix is zero as well. Figure 2.37 shows a simple robot in a vertical configuration, where joints 1 and 6 are colinear. As you can see, whether joint 1 or joint 6 rotate, the end effector will rotate the same amount. In practice, it is important to direct the controller to

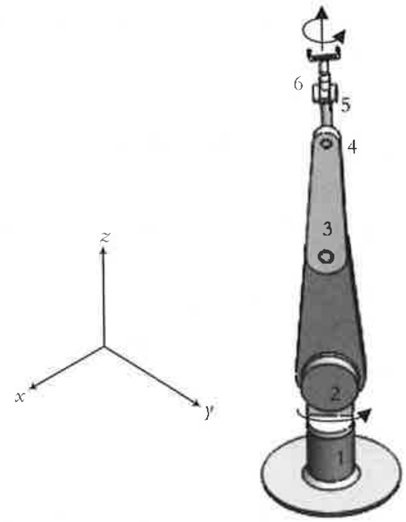


Figure 2.37 An example of a robot in a degenerate position.

take an emergency action; otherwise the robot will stop. Please note that this condition occurs if the two joints are similar. Otherwise, if one joint is prismatic and one is revolute (as in joints 3 and 4 of the Stanford arm), although the z-axes are colinear, the robot will not be in degenerate condition. Paul[11] has shown that if $\sin\alpha_4$, $\sin\alpha_5$ or $\sin\theta_5$ are zero, the robot will be degenerate (this occurs if joints 4 and 5, or 5 and 6 are parallel, and therefore, result in similar motions). Obviously, α_4 and α_5 can be designed to prevent the degeneracy of the robot. However, anytime θ_5 approaches zero or 180°, the robot will become degenerate.

2.15.2 Dexterity

We should be able to position and orientate a 6-DOF robot at any desired location within its work envelope by specifying the position and the orientation of the hand. However, as the robot gets increasingly closer to the limits of its workspace, it will get to a point where, although it is possible to locate it at a desired point, it will be impossible to orientate it at desired orientations. The volume of points where we can position the robot as desired but not orientate it is called nondexterous volume.

2.16 The Fundamental Problem with the Denavit-Hartenberg Representation

Although Denavit-Hartenberg representation has been extensively used in modeling and analysis of robot motions, and although it has become a standard method for doing so, there is a fundamental flaw with this technique, which many researchers have tried to solve by modifying the process.[12] The fundamental problem is that since all motions are about the x- and z-axes, the method cannot represent any motion about the y-axis. Therefore, if there is any motion about the y-axis, the method will fail. This occurs in a number of circumstances. For example, suppose two joint axes that are supposed to be parallel are assembled with a slight deviation. The small angle between the two axes will require a motion about the y-axis. Since all real industrial robots have some degree of inaccuracy in their manufacture, their inaccuracy cannot be modeled with the D-H representation.

Example 2.26 (Continued)

Reference Frames for the Stanford Arm: Figure 2.38 is the solution for the Stanford Arm in Example 2.26 (Figure 2.34). It is simplified for improved visibility. Table 2.6 shows the corresponding parameters.

For the derivation of the inverse kinematic solution of Stanford Arm, refer to References 5 and 13 at the end of the chapter. The following is a summary of the inverse kinematic solution for the Stanford Arm:

$$\theta_1 = \tan^{-1}\left(\frac{p_y}{p_x}\right) - \tan^{-1}\frac{d_2}{\pm\sqrt{r^2 - d_2^2}} \text{ where } r = \sqrt{p_x^2 + p_y^2} \tag{2.80}$$

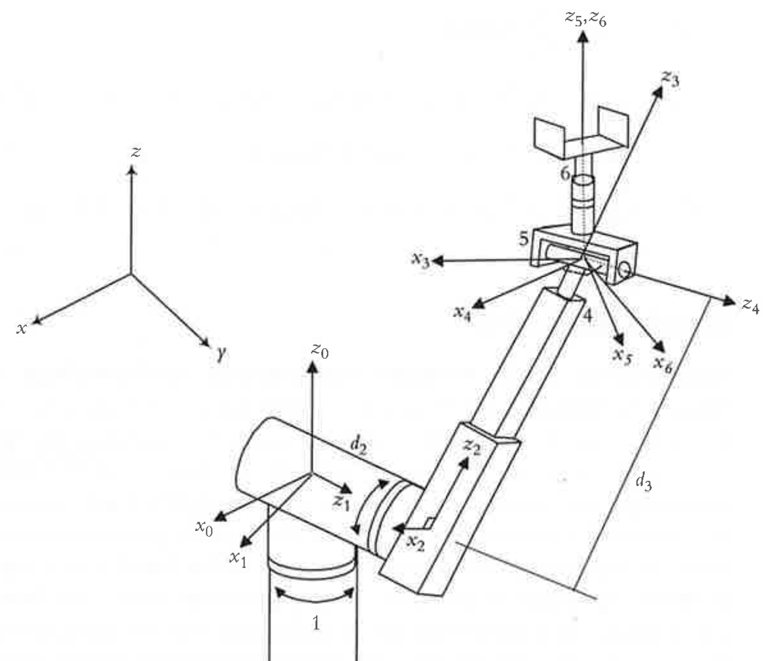


Figure 2.38 The frames of the Stanford Arm.

Table 2.6 *The Parameters Table for the Stanford Arm.*

#	θ	d	a	α
0–1	θ_1	0	0	−90
1–2	θ_2	d_2	0	90
2–3	0	d_3	0	0
3–4	θ_4	0	0	−90
4–5	θ_5	0	0	90
5–6	θ_6	0	0	0

$$\theta_2 = \tan^{-1}\frac{C_1p_x + S_1p_y}{p_z} \tag{2.81}$$

$$d_3 = S_2(C_1p_x + S_1p_y) + C_2p_z \tag{2.82}$$

$$\theta_4 = \tan^{-1}\frac{-S_1a_x + C_1a_y}{C_2(C_1a_x + S_1a_y) - S_2a_z} \quad \text{and} \quad \theta_4 = \theta_4 + 180^\circ \text{ if } \theta_5 < 0 \tag{2.83}$$

$$\theta_5 = \tan^{-1}\frac{C_4[C_2(C_1a_x + S_1a_y) - S_2a_z] + S_4[-S_1a_x + C_1a_y]}{S_2(C_1a_x + S_1a_y) + C_2a_z} \tag{2.84}$$

$$\begin{aligned} \theta_6 &= \tan^{-1}\frac{S_6}{C_6} \text{ where} \\ S_6 &= -C_5\{C_4[C_2(C_1o_x + S_1o_y) - S_2o_z] + S_4[-S_1o_x + C_1o_y]\} \\ &\quad + S_5\{S_2(C_1o_x + S_1o_y) + C_2o_z\} \\ C_6 &= -S_4[C_2(C_1o_x + S_1o_y) - S_2o_z] + C_4[-S_1o_x + C_1o_y] \end{aligned} \tag{2.85}$$

■

Example 2.29

Application of the Denavit-Hartenberg methodology in the design of a finger-spelling hand: A finger-spelling hand[14] was designed at Cal Poly, San Luis Obispo, in order to enable ordinary users to communicate with individuals who are blind and deaf. The hand, with its 17 degrees of freedom, can form all the finger-spelling letters and numbers (Figure 2.39). Each finger-wrist combination was assigned a set of frames based on the D-H representation in order to derive the forward and inverse kinematic equations of the hand. These equations may be used to drive the fingers to position. This application shows that in addition to modeling the motions of a robot, the D-H technique may be used to represent transformations, rotations, and movements between different kinematic elements, regardless of whether or not a robot is involved. You may also find other applications for this representation. ■

Figure 2.39 Cal Poly finger-spelling hand. (Supported by the Smith-Kettlewell Eye Research Institute, San Francisco.)

2.17 Design Projects

Starting with this chapter and continuing with the rest of the book, we will apply the current information in each chapter to the design of simple robots. This will help you to apply the current material to the design of a robot of your own. Common 6-DOF robots are too complicated to be considered simple; therefore, we will use 3-DOF robots. The intention is to design a simple robot that can possibly be built by you from readily available parts from hobby shops, hardware stores, and surplus dealers.

In this section, you may consider the preliminary design of the robot and its configuration, keeping in mind the possible types of actuators you may want to consider later. Although we will study this subject later, it is a good idea to consider the types of actuators now. You should also consider the types of links and joints you may want to use, possible lengths, types of joints, and material (for example, wood dowels, hollow aluminum or brass tubes available in hardware stores, and so on).

2.17.1 A 3-DOF Robot

For this project, you may want to design your own preferred robot with your own preferred configuration. Creativity is always encouraged. However, we will discuss a simple robot as a guideline for you to design and build. After the configuration of the robot is finalized, you should proceed with the derivation of forward and inverse kinematic equations. The final result of this part of the design project will be a set of inverse kinematic equations for the simple 3-DOF robot that can later be used to drive the robot to desired positions. You must realize that the price we pay for this simplicity is that we may only specify the position of the robot, but not its orientation.

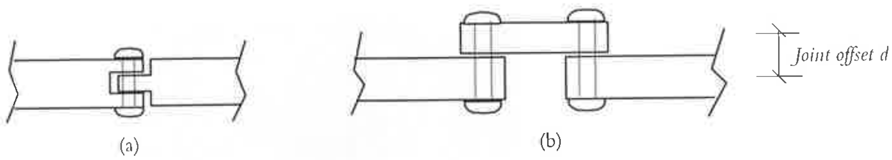


Figure 2.40 Two simple designs for a joint.

One of the important considerations in the design of the robot is its joints. Figure 2.40(a) shows a simple design that has no joint offset d. This would apparently simplify the analysis of the robot, since the A matrix related to the joint would be simpler. However, manufacturing such a joint is not as simple as the design in Figure 2.40(b). The latter allows a larger range of motion too. On the other hand, although we apparently have to deal with a joint offset d with the joint design in Figure 2.40(b), you must remember that in most cases, there will be a second joint with the same joint offset in the opposite direction, which cancels the former in the robot's overall equation. As a result, we will assume that the joints of our robot can be built as in Figure 2.40(b) without having to worry about joint offset d.

We will discuss actuators in Chapter 7. However, for this design project, you should probably consider the use of a servomotor or a stepper motor. While you are designing your robot, consider what type of actuators you will use and how you will connect the actuators to the links and joints. Remember that at this point, you are only designing the robot configuration; you can always change your actuators and adapt the new design to your robot.

When the preliminary sketch of the robot is finished, assign coordinate frames to each joint, fill out the parameters table for the frames, develop the matrices for each frame transformation, and calculate the final ${}^{U}T_{H}$. Then, using the methods learned in this chapter, develop the inverse kinematic equations of the robot. This means that using these equations, if you actually build the robot, you will be able to run it and control its position (since the robot is 3-DOF, you will not be able to control its orientation).

Figure 2.41 shows a simple design for a 3-DOF robot you may use as a guide for your design. In one student design, the lengths were 8, 2, 9, 2, and 9 inches respectively. The links were made of hollow aluminum bars, actuated by three DC gear-motors with encoder feedback and connected to the joint through worm gears.

Figure 2.41 also shows one possible set of frames assigned to the joints. The end of the robot has its own frame. Frame 3 is needed in order to transform from frame 2 to the hand frame. To be able to correctly develop forward and inverse kinematic equations of the robot, it is crucial to define the reset position of the robot, where all joint angles are zero. In this example, the reset position is defined as the robot pointing up and x_0 parallel to x_U. At this point, there is a 90° angle between x_1 and x_2. Therefore, the actual angle for this joint should be $-90 + \theta_2$. The same is true for x_0 and x_1, where a 90° angle exists between the two when θ_1 is zero; therefore, the angle between the two is $90 + \theta_1$. Also notice the permanent angles between other frames.

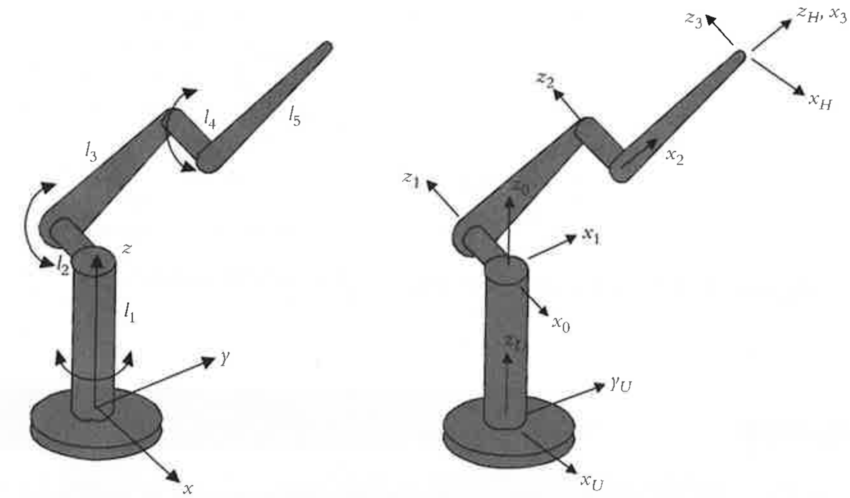


Figure 2.41 A simple 3-DOF robot design that may be used for the design project.

This exercise is left for you to complete. The inverse kinematic equations for the robot relative to frame 0 are:

$$\begin{aligned}
\theta_1 &= \tan^{-1}(-p_x/p_y) \\
\theta_3 &= \cos^{-1}[((p_y/C_1)^2 + (p_z)^2 - 162)/162] \\
\theta_2 &= \cos^{-1}[(p_z C_1(1 + C_3) + p_y S_3)/(18(1 + C_3)C_1)]
\end{aligned} \tag{2.86}$$

Note: If $\cos\theta_1$ is zero, use p_x/S_1 instead of p_y/C_1.

2.17.2 A 3-DOF Mobile Robot

Another project you may consider is a mobile robot. These robots are very common and are used in autonomous navigation and developing artificial intelligence for robots. In general, you may assume the robot is capable of moving in a plane that may be represented by translations along the x- and y-axes or a translation and rotation in a polar form (r, θ). Additionally, the orientation of the robot may be changed by rotating it about the z-axis (α). Therefore, the kinematic equations of the motion of the robot can be developed and used to control its motions. A schematic representation of the robot is shown in Figure 2.42. (See Chapter 7 for a design project involving a single-axis robot that may also be used for this project).

In the next chapters, we will continue with the design of your robots.

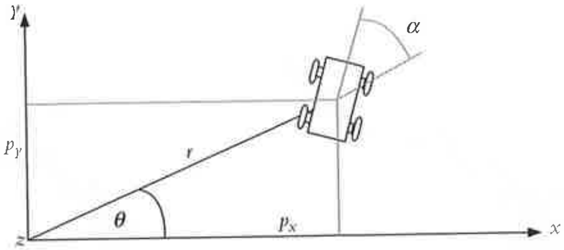


Figure 2.42 Schematic representation of a 3-DOF mobile robot.

Summary

In this chapter, we discussed methods for representation of points, vectors, frames, and transformations by matrices. Using matrices, we discussed forward and inverse kinematic equations for specific types of robots such as Cartesian, cylindrical, and spherical robots as well as Euler and RPY orientation angles. However, the main thrust of this chapter was to learn how to represent the motions of a multi-DOF robot in space and how to derive the forward and inverse kinematic equations of the robot using the Denavit-Hartenberg (D-H) representation technique. This method can be used to represent any type of robot configuration, regardless of the number and type of joints or joint and link offsets or twists.

In the next chapter, we will continue with the differential motions of robots, which in effect is the equivalent of velocity analysis of robots.

References

1. Niku, S., "Scheme for Active Positional Correction of Robot Arms," Proceedings of the 5th International Conference on CAD/CAM, Robotics and Factories of Future, Springer Verlag, pp. 590–593, 1991.
2. Puopolo, Michael G., Saeed B. Niku, "Robot Arm Positional Deflection Control with a Laser Light," Proceedings of the Mechatronics '98 Conference, Skovde, Sweden, Adolfsson and Karlsen, Editors, Pergamon Press, September 1998, pp. 281–286.
3. Ardayfio, D. D., Q. Danwen, "Kinematic Simulation of Novel Robotic Mechanisms Having Closed Kinematic Chains," Paper # 85-DET-81, American Society of Mechanical Engineers, 1985.
4. Denavit, J., R. S. Hartenberg, "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," *ASME Journal of Applied Mechanics*, June 1955, 215–221.
5. Paul, Richard P., "Robot Manipulators, Mathematics, Programming, and Control," MIT Press, 1981.
6. Craig, John J., "Introduction to Robotics: Mechanics and Control", 2nd Edition, Addison Wesley, 1989.
7. Shahinpoor, Mohsen, "A Robot Engineering Textbook," Harper & Row, 1987.
8. Koren, Yoram, "Robotics for Engineers," McGraw-Hill, 1985.

9. Fu, K. S., R.C. Gonzalez, C. S. G. Lee, "Robotics: Control, Sensing, Vision, and Intelligence," McGraw-Hill, 1987.

10. Eman, Kornel F., "Trajectories," International Encyclopedia of Robotics: Applications and Automation," Richard C. Dorf, Editor, John Wiley and Sons, NY, 1988, pp. 1796–1810.

11. Paul, Richard P., C. N. Stevenson, "Kinematics of Robot Wrists," The International Journal of Robotics Research, Vol. 2, No. 1, Spring 1983, pp. 31–38.

12. Barker, Keith, "Improved Robot-Joint Calculations," NASA Tech Briefs, September 1988, p. 79.

13. Ardayfio, D. D., R. Kapur, S. B. Yang, W. A. Watson, "Micras, Microcomputer Interactive Codes for Robot Analysis and Simulation," Mechanisms and Machine Theory, Vol. 20, No. 4, 1985, pp. 271–284.

14. Garcia, Mario, S. B. Niku, "Finger-Spelling Hand," masters thesis, Mechanical Engineering, Cal Poly, San Luis Obispo, CA, 2009.

Problems

The isometric grid (Figure 2.43) is provided to you for use with the problems in this chapter. It is meant to be used as a tracing grid for drawing 3-D shapes and objects such as robots, frames, and transformations. Please make copies of the grid for each problem that requires graphical representation of the results. The grid is also available commercially.

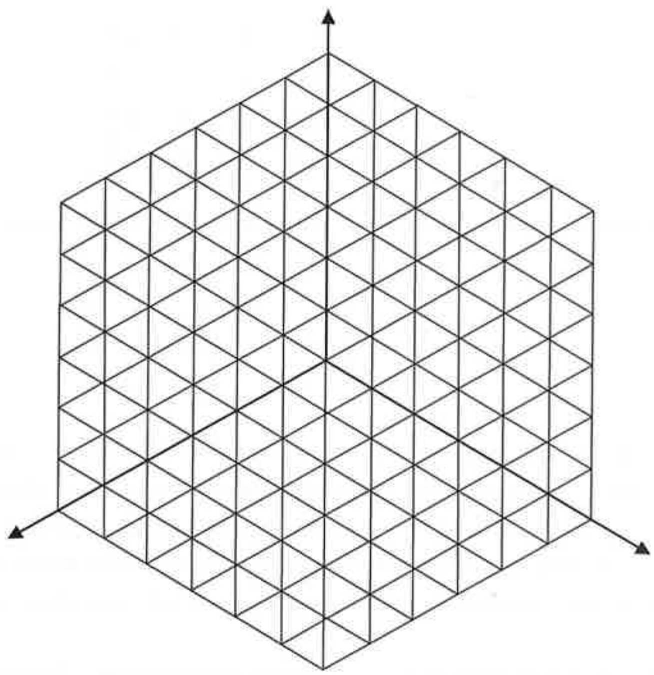


Figure 2.43 Isometric grid.

2.1. Write a unit vector in matrix form that describes the direction of the cross product of $\mathbf{p} = 5\mathbf{i} + 3\mathbf{k}$ and $\mathbf{q} = 3\mathbf{i} + 4\mathbf{j} + 5\mathbf{k}$.

2.2. A vector $\mathbf{p}$ is 8 units long and is perpendicular to vectors $\mathbf{q}$ and $\mathbf{r}$ described below. Express the vector in matrix form.

$$\mathbf{q}_{unit} = \begin{bmatrix} 0.3 \\ q_y \\ 0.4 \\ 0 \end{bmatrix} \quad \mathbf{r}_{unit} = \begin{bmatrix} r_x \\ 0.5 \\ 0.4 \\ 0 \end{bmatrix}$$

2.3. Will the three vectors $\mathbf{p}$, $\mathbf{q}$, and $\mathbf{r}$ in Problem 2.2 form a traditional frame? If not, find the necessary unit vector $\mathbf{s}$ to form a frame between $\mathbf{p}$, $\mathbf{q}$, and $\mathbf{s}$.

2.4. Suppose that instead of a frame, a point $P = (3, 5, 7)^T$ in space was translated a distance of $d = (2, 3, 4)^T$. Find the new location of the point relative to the reference frame.

2.5. The following frame B was moved a distance of $d = (5, 2, 6)^T$. Find the new location of the frame relative to the reference frame.

$$B = \begin{bmatrix} 0 & 1 & 0 & 2 \\ 1 & 0 & 0 & 4 \\ 0 & 0 & -1 & 6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.6. For frame F, find the values of the missing elements and complete the matrix representation of the frame.

$$F = \begin{bmatrix} ? & 0 & -1 & 5 \\ ? & 0 & 0 & 3 \\ ? & -1 & 0 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.7. Find the values of the missing elements of frame B and complete the matrix representation of the frame.

$$B = \begin{bmatrix} 0.707 & ? & 0 & 2 \\ ? & 0 & 1 & 4 \\ ? & -0.707 & 0 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.8. Derive the matrix that represents a pure rotation about the y-axis of the reference frame.

2.9. Derive the matrix that represents a pure rotation about the z-axis of the reference frame.

2.10. Verify that the rotation matrices about the reference frame axes follow the required constraint equations set by orthogonality and length requirements of directional unit vectors.

2.11. Find the coordinates of point $P(2, 3, 4)^T$ relative to the reference frame after a rotation of 45° about the x-axis.

2.12. Find the coordinates of point $P(3, 5, 7)^T$ relative to the reference frame after a rotation of 30° about the z-axis.

2.13. Find the new location of point $P(1,2,3)^T$ relative to the reference frame after a rotation of 30° about the z-axis followed by a rotation of 60° about the y-axis.

2.14. A point P in space is defined as ${}^BP = (5,3,4)^T$ relative to frame B which is attached to the origin of the reference frame A and is parallel to it. Apply the following transformations to frame B and find ${}^AP$. Using the 3-D grid, plot the transformations and the result and verify it. Also verify graphically that you would not get the same results if you apply the transformations relative to the current frame:

- Rotate 90° about x-axis; then
- Translate 3 units about y-axis, 6 units about z-axis, and 5 units about x-axis; then,
- Rotate 90° about z-axis.

2.15. A point P in space is defined as ${}^BP = (2,3,5)^T$ relative to frame B which is attached to the origin of the reference frame A and is parallel to it. Apply the following transformations to frame B and find ${}^AP$. Using the 3-D grid, plot the transformations and the result and verify it:

- Rotate 90° about x-axis, then
- Rotate 90° about local a-axis, then
- Translate 3 units about y-, 6 units about z-, and 5 units about x-axes.

2.16. A frame B is rotated 90° about the z-axis, then translated 3 and 5 units relative to the n- and o-axes respectively, then rotated another 90° about the n-axis, and finally, 90° about the y-axis. Find the new location and orientation of the frame.

$$B = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.17. The frame B of Problem 2.16 is rotated 90° about the a-axis, 90° about the y-axis, then translated 2 and 4 units relative to the x- and y-axes respectively, then rotated another 90° about the n-axis. Find the new location and orientation of the frame.

$$B = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.18. Show that rotation matrices about the y- and the z-axes are unitary.

2.19. Calculate the inverse of the following transformation matrices:

$$T_1 = \begin{bmatrix} 0.527 & -0.574 & 0.628 & 2 \\ 0.369 & 0.819 & 0.439 & 5 \\ -0.766 & 0 & 0.643 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad T_2 = \begin{bmatrix} 0.92 & 0 & 0.39 & 5 \\ 0 & 1 & 0 & 6 \\ -0.39 & 0 & 0.92 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.20. Calculate the inverse of the matrix B of Problem 2.17.

2.21. Write the correct sequence of movements that must be made in order to restore the original orientation of the spherical coordinates and make it parallel to the reference frame. About what axes are these rotations supposed to be?

2.22. A spherical coordinate system is used to position the hand of a robot. In a certain situation, the hand orientation of the frame is later restored in order to be parallel to the reference frame, and the matrix representing it is described as:

$$T_{sph} = \begin{bmatrix} 1 & 0 & 0 & 3.1375 \\ 0 & 1 & 0 & 2.195 \\ 0 & 0 & 1 & 3.214 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Find the necessary values of r, β, γ to achieve this location.
- Find the components of the original matrix **n**, **o**, **a** vectors for the hand before the orientation was restored.

2.23. Suppose that a robot is made of a Cartesian and RPY combination of joints. Find the necessary RPY angles to achieve the following:

$$T = \begin{bmatrix} 0.527 & -0.574 & 0.628 & 4 \\ 0.369 & 0.819 & 0.439 & 6 \\ -0.766 & 0 & 0.643 & 9 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.24. Suppose that a robot is made of a Cartesian and Euler combination of joints. Find the necessary Euler angles to achieve the following:

$$T = \begin{bmatrix} 0.527 & -0.574 & 0.628 & 4 \\ 0.369 & 0.819 & 0.439 & 6 \\ -0.766 & 0 & 0.643 & 9 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.25. Assume that the three Euler angles used with a robot are 30°, 40°, 50° respectively. Determine what angles should be used to achieve the same result if RPY is used instead.

2.26. A frame ${}^{U}B$ was moved along its own o-axis a distance of 6 units, then rotated about its n-axis an angle of 60°, then translated about the z-axis for 3 units, followed by a rotation of 60° about the z-axis, and finally rotated about x-axis for 45°.

- Calculate the total transformation performed.
- What angles and movements would we have to make if we were to create the same location and orientation using Cartesian and Euler configurations?

2.27. A frame ${}^{U}F$ was moved along its own n-axis a distance of 5 units, then rotated about its o-axis an angle of 60°, followed by a rotation of 60° about the z-axis, then translated about its a-axis for 3 units, and finally rotated 45° about the x-axis.

- Calculate the total transformation performed.
- What angles and movements would we have to make if we were to create the same location and orientation using Cartesian and RPY configurations?

2.28. Frames describing the base of a robot and an object are given relative to the Universe frame.

- Find a transformation ${}^{R}T_{H}$ of the robot configuration if the hand of the robot is to be placed on the object.
- By inspection, show whether this robot can be a 3-axis spherical robot, and if so, find α, β, r.
- Assuming that the robot is a 6-axis robot with Cartesian and Euler coordinates, find $p_x, p_y, p_z, \phi, \theta, \psi$.

$$
{}^{U}T_{obj} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^{U}T_{R} = \begin{bmatrix} 0 & -1 & 0 & 2 \\ 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

2.29. A 3-DOF robot arm has been designed for applying paint on flat walls, as shown.

- Assign coordinate frames as necessary based on the D-H representation.
- Fill out the parameters table.
- Find the ${}^{U}T_{H}$ matrix.

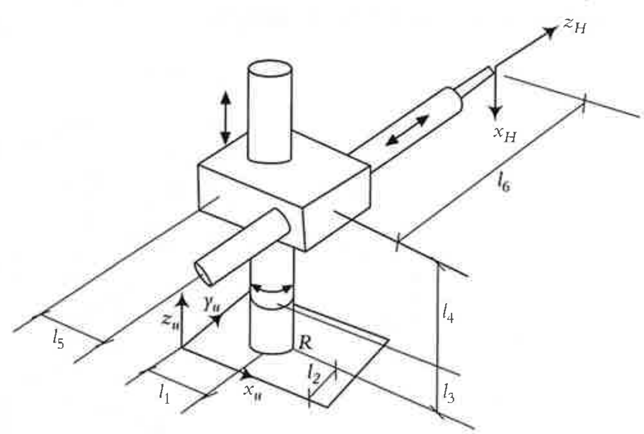


Figure P.2.29

2.30. In the 2-DOF robot shown, the transformation matrix ${}^{0}T_{H}$ is given in symbolic form, as well as in numerical form for a specific location. The length of each link l_1 and l_2 is 1 ft. Calculate the values of θ_1 and θ_2 for the given location.

$$
{}^{0}T_{H} = \begin{bmatrix} C_{12} & -S_{12} & 0 & l_2 C_{12} + l_1 C_1 \\ S_{12} & C_{12} & 0 & l_2 S_{12} + l_1 S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -0.2924 & -0.9563 & 0 & 0.6978 \\ 0.9563 & -0.2924 & 0 & 0.8172 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

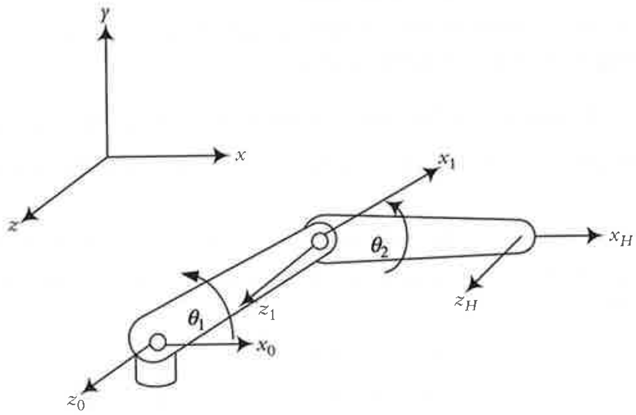


Figure P.2.30

2.31. For the following SCARA-type robot:

- Assign the coordinate frames based on the D-H representation.
- Fill out the parameters table.
- Write all the A matrices.
- Write the $^{U}T_{H}$ matrix in terms of the A matrices.

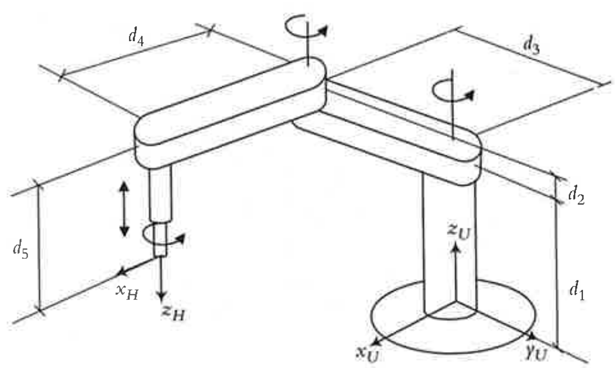


Figure P.2.31

2.32. A special 3-DOF spraying robot has been designed as shown:

- Assign the coordinate frames based on the D-H representation.
- Fill out the parameters table.
- Write all the A matrices.
- Write the $^{U}T_{H}$ matrix in terms of the A matrices.

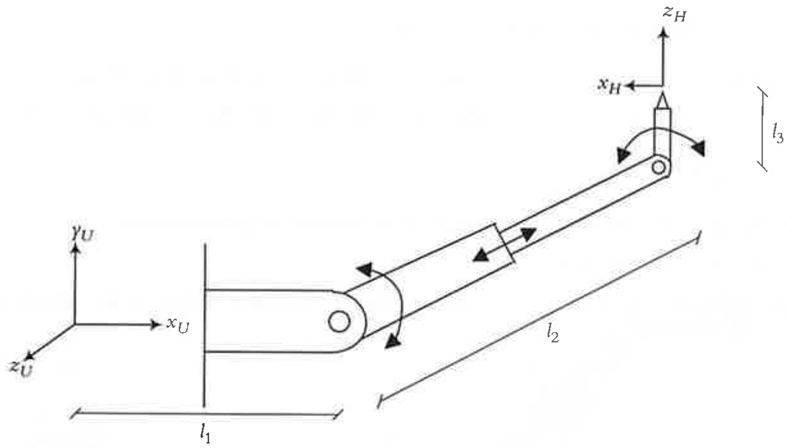
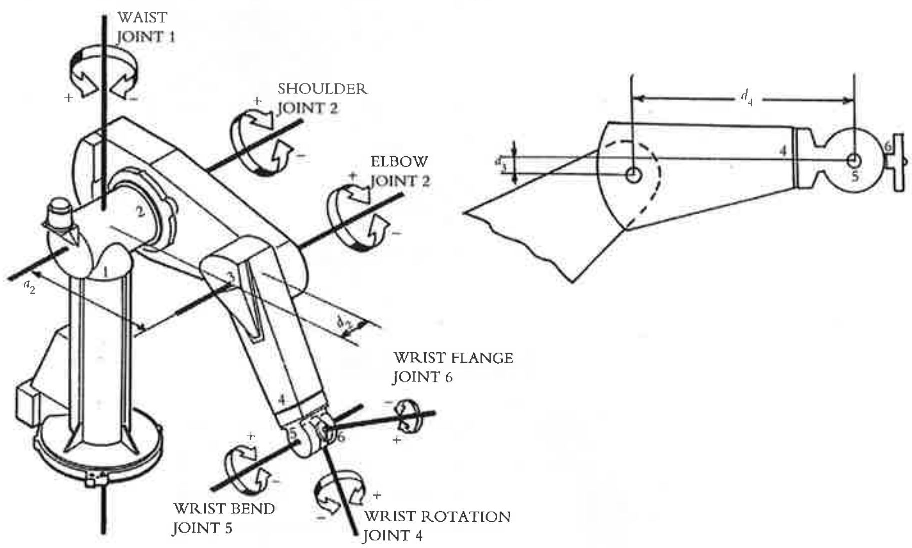


Figure P.2.32

2.33. For the Unimation Puma 562, 6-axis robot shown,



#	θ	d	a	α
0–1				
1–2				
2–3				
3–4				
4–5				
5–6				

Figure P.2.33 Puma 562. (Reprinted with permission from Staubli Robotics.)

- Assign the coordinate frames based on the D-H representation.
- Fill out the parameters table.
- Write all the A matrices.

- Find the ${}^{R}T_{H}$ matrix for the following values:

$$\text{Base height} = 27 \text{ in.}, d_2 = 6 \text{ in.}, a_2 = 15 \text{ in.}, a_3 = 1 \text{ in}, d_4 = 18 \text{ in.}$$
$$\theta_1 = 0°, \theta_2 = 45°, \theta_3 = 0°, \theta_4 = 0°, \theta_5 = -45°, \theta_6 = 0°$$

2.34. For the given 4-DOF robot:

- Assign appropriate frames for the Denavit-Hartenberg representation.
- Fill out the parameters table.
- Write an equation in terms of A matrices that shows how ${}^{U}T_{H}$ can be calculated.

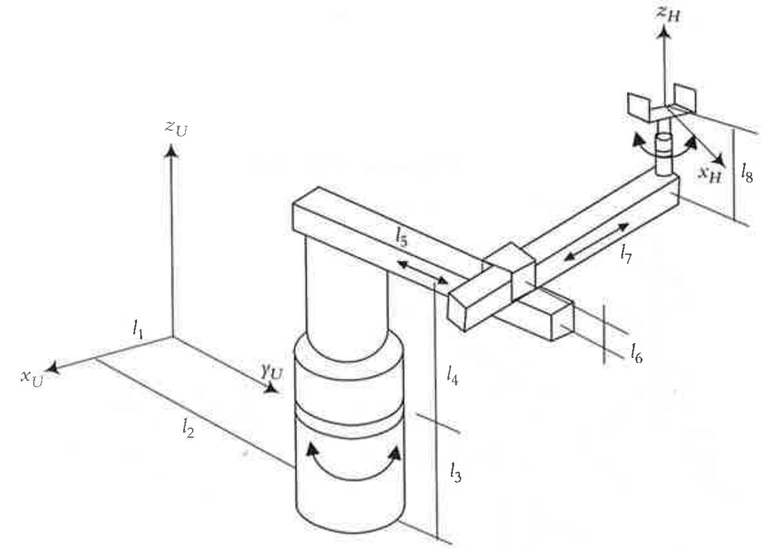


Figure P.2.34

#	θ	d	a	α
0–1				
1–2				
2–3				
3–				

2.35. For the given 4-DOF robot designed for a specific operation:

- Assign appropriate frames for the Denavit-Hartenberg representation.
- Fill out the parameters table.
- Write an equation in terms of A matrices that shows how ${}^{U}T_{H}$ can be calculated.

#	θ	d	a	α
0–1				
1–2				
2–3				
3–				

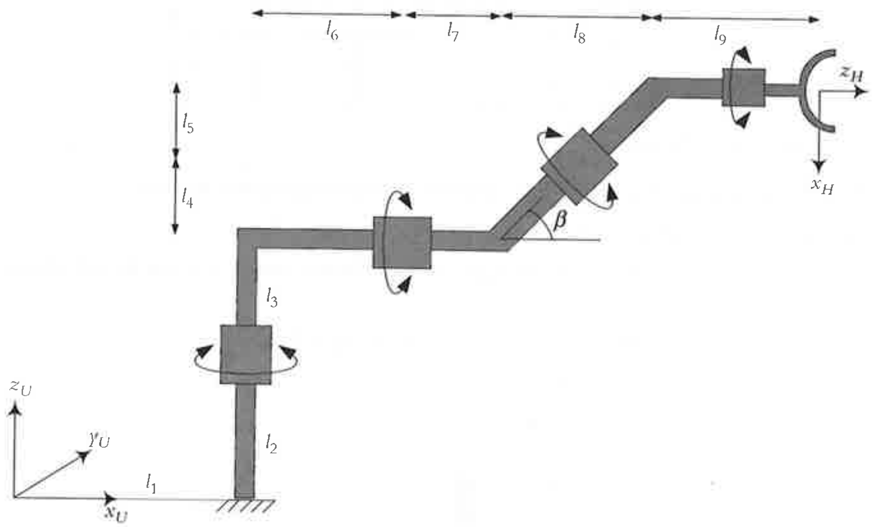


Figure P.2.35

2.36. For the given specialty designed 4-DOF robot:

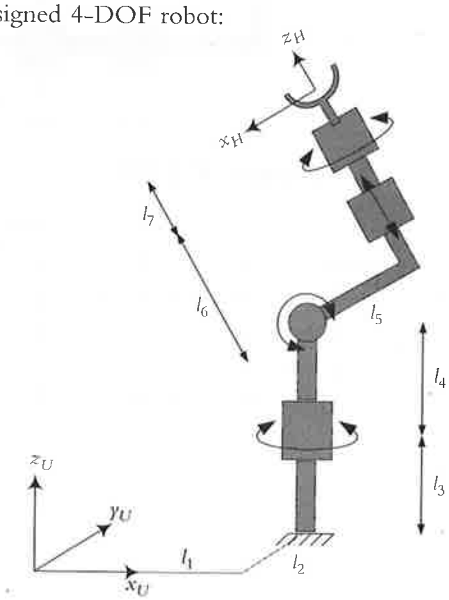


Figure P.2.36

- Assign appropriate frames for the Denavit-Hartenberg representation.
- Fill out the parameters table.
- Write an equation in terms of A matrices that shows how ${}^{U}T_H$ can be calculated.

#	θ	d	a	α
0–1				
1–2				
2–3				
3–				

2.37. For the given 3-DOF robot:

- Assign appropriate frames for the Denavit-Hartenberg representation.
- Fill out the parameters table.
- Write an equation in terms of A matrices that shows how ${}^{U}T_H$ can be calculated.

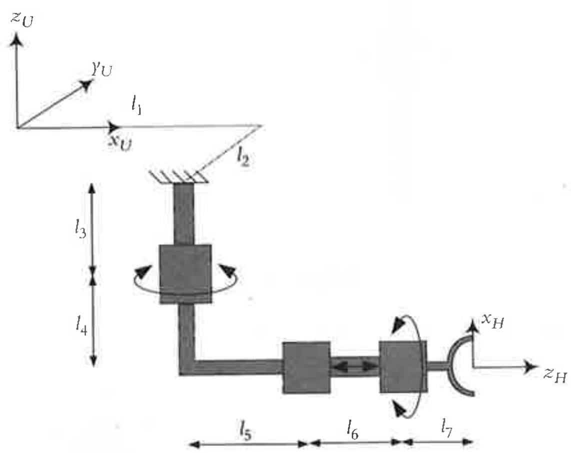


Figure P.2.37

#	θ	d	a	α
0–1				
1–2				
2–				

2.38. For the given 4-DOF robot:

- Assign appropriate frames for the Denavit-Hartenberg representation.
- Fill out the parameters table.
- Write an equation in terms of A matrices that shows how ${}^{U}T_H$ can be calculated.

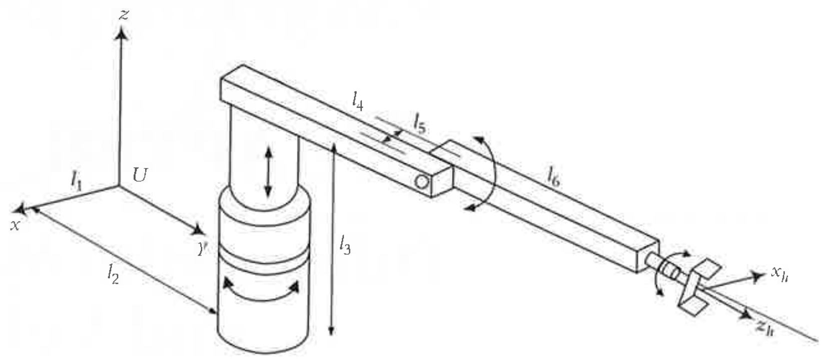


Figure P.2.38

#	θ	d	a	α
0–1				
1–2				
2–3				
3–				

2.39. Derive the inverse kinematic equations for the robot of Problem 2.36.

2.40. Derive the inverse kinematic equations for the robot of Problem 2.37.