Control of Robot Manipulators

F. L. Lewis
University of Texas at Arlington

C. T. Abdallah
University of New Mexico

D. M. Dawson
Clemson University

1993

MACMILLAN PUBLISHING COMPANY
New York

MAXWELL MACMILLAN CANADA
Toronto

MAXWELL MACMILLAN INTERNATIONAL
New York Oxford Singapore Sydney

APPENDIX **A**

Review of Robot Kinematics and Jacobians

We review here the basic information from a first course in robotics that is needed for this book. This includes robot kinematics, the arm Jacobian, and the issue of specifying the Cartesian position. The review is detailed since those with a background in system theory and controls may not yet have seen this material. Several examples are given which are used throughout the book for design and simulation purposes.

A.1 Basic Manipulator Geometries

In this section we look at some basic arm geometries. A robot arm or manipulator is composed of a set of *joints* separated in space by the arm *links*. The joints are where the motion in the arm occurs (cf. our own wrist and elbow), while the links are of fixed construction (cf. our own forearm). Thus the links maintain a fixed relationship between the joints. [Although the links may be *flexible* (i.e., they may bend); we ignore flexibility effects here.]

The joints may be actuated by motors or hydraulic actuators. There are two sorts of robot joints, involving two sorts of motion. A *revolute joint* (denoted R) is one that allows rotary motion about an axis of rotation. An example is the human elbow. A *prismatic joint* (denoted P) is one that allows extension or telescopic motion. An example is a telescoping automobile antenna. There is no anthropomorphous analogy to the prismatic link. The *joint variables* of a manipulator are the variable parameters of the joints. For a revolute joint the variable is an angle, denoted θ. For a prismatic joint it is a length, denoted d.

Some basic arm geometries are shown in Fig. A.1-1. The RRR *articulated arm* in Fig. A.1-1a is like the human arm, while the PPP *Cartesian arm* in Fig. A.1-1e is closely tied to the coordinates used in the manipulator *workspace,* where the Cartesian coordinates (x,y,z) are often used to describe tasks to be performed. The workspace is the total volume swept out by the end effector as the robot executes all possible motions.

The *joint axis* of a revolute joint is the axis about which the rotation θ occurs. (The sense of rotation is determined using the right-handed screw rule: If the curled fingers of the right hand indicate the direction of rotation, the thumb indicates the direction of the axis of rotation.) For a prismatic joint, it is the axis along which the telescoping action d occurs. The relative orientations of the joint axes of an arm determine its fundamental properties. Figure A.1-1c shows a RRP manipulator known as the SCARA (selected

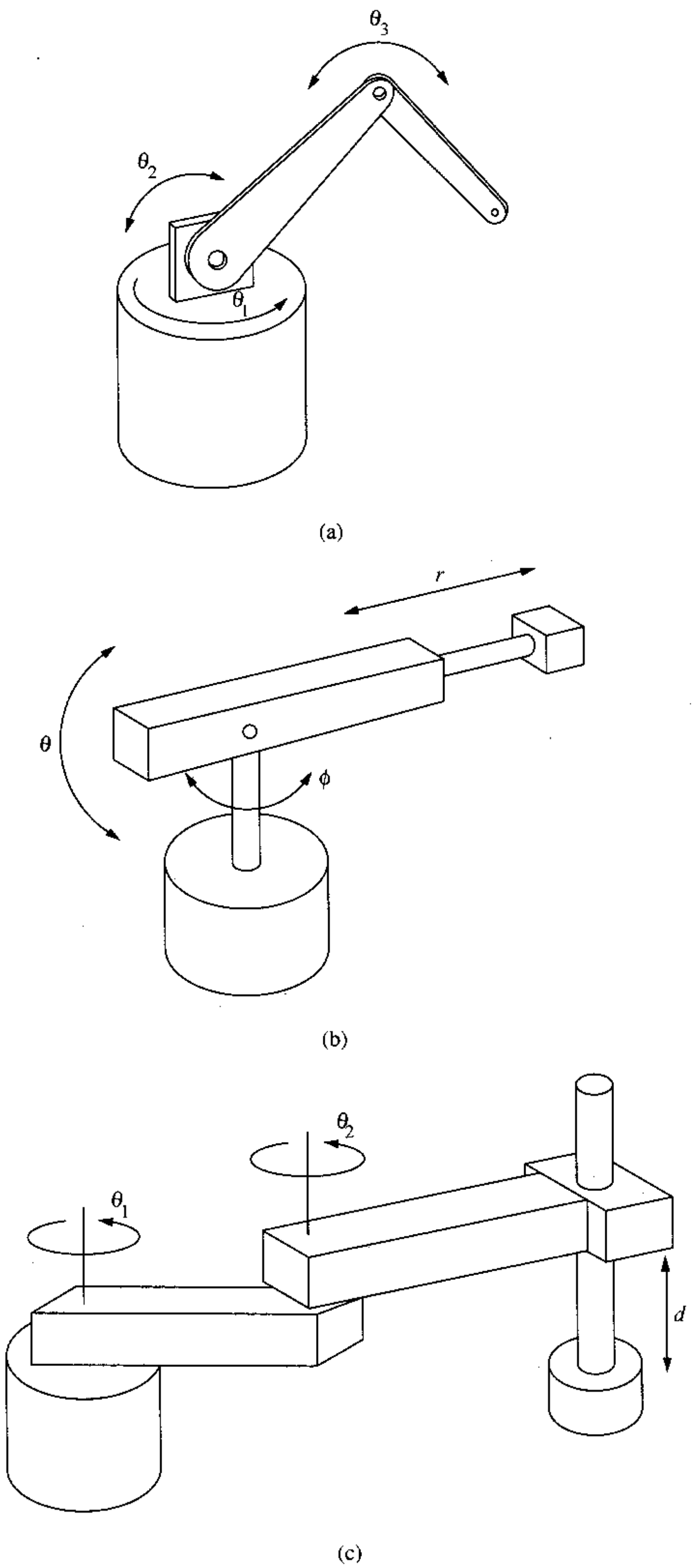


FIGURE A.1-1 Basic robot arm geometries: (a) articulated arm, revolute coordinates (RRR); (b) spherical coordinates (RRP); (c) SCARA arm (RRP).

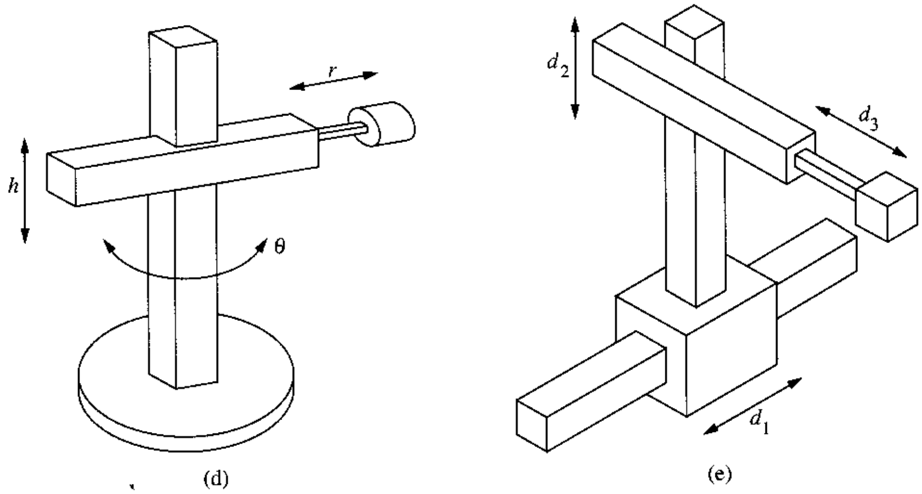


FIGURE A.1-1 (*Cont.*) (d) cylindrical coordinates (RPP); (e) Cartesian arm, rectangular coordinates (PPP).

compliant articulated robot for assembly) arm. It has quite a different structure than the RRP spherical arm shown in Fig. A.1-1b, since its joint axes are all parallel. On the other hand, the joint axes of the spherical arm intersect at a point.

Industrial examples of the RRR arm are the PUMA and Cincinnati-Milacron T^3 735 manipulators. The Stanford manipulator is a spherical RRP arm; the AdeptOne is a SCARA RRP arm. An example of the RPP arm is the GMF M-100. The Cincinnati-Milacron T^3 gantry robot is a PPP arm.

Many industrial robots are *serial link* manipulators since they consist of a series of links connected together by actuated joints. The base is called link 0, and the last link is terminated by the tool or *end effector.* Many robots have six joints, corresponding to the *six degrees of freedom* needed to obtain arbitrary position and orientation of the end effector in three-dimensional space.

Arms like the PUMA 560 have six revolute joints. In such an arm, the joints may be grouped into two sets of three joints each. The first three joints may be used to place the end effector at an arbitrary position within the three-dimensional workspace. The last three joints may be used to obtain an arbitrary orientation of the end effector at that position. In the PUMA 560, the axes of joints 4, 5, and 6 intersect at a common point and are mutually orthogonal. This makes orienting the end-effector convenient. The last three joints are known as the *wrist mechanism* (see Example A.2-4).

A.2 Robot Kinematics

Here we review the kinematics of robot manipulators, including the arm A matrices, homogeneous transformations, the T matrix, forward and inverse kinematics, and joint-space and Cartesian coordinates. Several illustrative examples are given.

A Matrices

For given values of the joint variables, it is important to be able to specifiy the locations of the links with respect to each other. This is accomplished by using the manipulator *kinematic equations.*

We may associate with each link i a coordinate frame (x_i, y_i, z_i) fixed to that link. See Fig. A.2-1. A standard and consistent paradigm for so doing is the *Denavit–Hartenberg (D-H) representation* [Paul 1981, Spong and Vidyasagar 1989]. The frame attached to link 0 (i.e., the base of the manipulator) is called the *base frame* or *inertial frame.*

The relation between coordinate frame $i-1$ and coordinate frame i is given by the transformation matrix

$$A_i = \left[\begin{array}{ccc|c} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ \hline 0 & 0 & 0 & 1 \end{array}\right]. \qquad \text{(A.2-1)}$$

Most of the parameters in this *A matrix for link i* are fixed. The *link parameters* are α_i, the twist of the link i, and a_i, the length of link i. These parameters are tabulated for each link in the arm manufacturer's specifications. The *joint parameters* are the joint angel θ_i and the joint offset d_i. If joint i is revolute, the joint variable is θ_i and d_i is a constant tabulated in the specs. On the other hand, if the link is prismatic, then d_i is the joint variable and θ_i is a constant provided in the specs. The parameter a_i for a prismatic joint is defined to be zero, since the link length is variable and described by d_i.

By the D-H convention, for a revolute joint, the rotation θ_i occurs about axis z_{i-1}. For a prismatic joint d_i occurs along axis z_{i-1}. Thus the link coordinate frame is considered to be attached to the *outer end* of the link. See the examples.

The A matrix A_i is a function of only a single variable, namely the joint variable θ_i or d_i, since all the other parameters in A_i are fixed for a specific

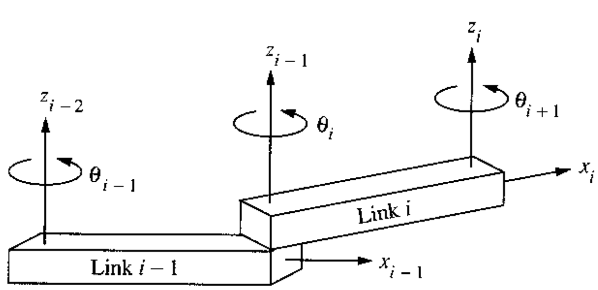


FIGURE A.2-1 Link kinematic relations in a manipulator.

joint. If a manipulator has n links, the *joint-variable vector* q is an n-vector composed of the individual joint variables. Thus q is in general a combination of angles θ_i and lengths d_i. For instance, for an RRP arm,

$$q = [\theta_1 \quad \theta_2 \quad d_3]^T.$$

The components of q are denoted by q_i; that is, the general joint variable q_i can represent either an angle θ_i or a length d_i as appropriate.

Homogeneous Transformations

The A matrix is a *homogeneous transformation* matrix of the form

$$A_i = \begin{bmatrix} R_i & p_i \\ 0 & 1 \end{bmatrix}, \tag{A.2-2}$$

where R_i is a *rotation matrix* and p_i is a *translation vector.* Thus if ${}^ir$ is a point described with respect to the coordinate frame of link i, the same point has coordinates ${}^{i-1}r$ with respect to the frame of link $i-1$ given by

$${}^{i-1}r = A_i{}^ir. \tag{A.2-3}$$

The homogeneous transformation is a 4×4 matrix, so that it can describe both rotations and translations; therefore, the vectors describing position in a given coordinate frame are 4-vectors. They are of the form

$${}^ir = \begin{bmatrix} {}^ix \\ {}^iy \\ {}^iz \\ 1 \end{bmatrix}, \tag{A.2-4}$$

where $({}^ix, {}^iy, {}^iz)$ are the coordinates of the point in frame i. Thus, according to (A.2-3) and (A.2-2),

$$\begin{bmatrix} {}^{i-1}x \\ {}^{i-1}y \\ {}^{i-1}z \end{bmatrix} = R_i \begin{bmatrix} {}^ix \\ {}^iy \\ {}^iz \end{bmatrix} + p_i, \tag{A.2-5}$$

which is just a rotation of R_i applied to the coordinates in frame i plus a translation of p_i.

We may interpret frame $i-1$ as the fixed (i.e., "original") frame and frame i as the rotated and translated (i.e., "new") frame due to the following considerations. According to (A.2-5), there is an easy way to find the rotation matrix R_i that rotates a given coordinate frame $i-1$ into another given frame i. Set p_i equal to zero and $({}^ix, {}^iy, {}^iz) = (1, 0, 0)$. Then (A.2-5) is equal to the first column of R_i. That is, the first column of R_i is nothing but *the repre-*

sentation of the new x-axis ^{i}x *in terms of the original coordinates* $({}^{i-1}x, {}^{i-1}y, {}^{i-1}z)$. Similarly, the second (respectively third) column of R_i is the representation of the rotated y-axis ^{i}y (respectively, z-axis and ^{i}z) in the fixed frame $i-1$. We shall illustrate this in Example A.2-1.

Therefore, A_i may be interpreted from several points of view. It is the transformation that takes a representation ^{i}r of a vector in frame i to its representation ^{i-1}r in frame $i-1$. On the other hand, it is the description of frame i in terms of frame $i-1$; in fact, R_i describes the orientation of the axes of frame i in terms of frame $i-1$, while p_i describes the origin of frame i in terms of the coordinates of frame $i-1$.

A rotation matrix R enjoys the property of *orthogonality;* that is, $R^{\mathrm{T}} = R^{-1}$. It has one eigenvalue at $\lambda = 1$, whose eigenvector is the axis of rotation. A rotation of θ about the x axis, for instance, looks like

$$R_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}. \tag{A.2-6}$$

The last entry of "1" in (A.2-4) represents a *scaling factor* for the length of the vector. By using a homogeneous transformation whose (4,4) entry is not unity, vectors may be scaled. By using entries in positions, (4,1), (4,2), (4,3) of the transformation matrix, *perspective transformations* may be performed. These ideas are important, for instance, in camera-frame transformations but will not be useful in the transformations associated with the manipulator arm.

Arm T Matrix

To obtain the coordinates of a point in terms of the base (i.e., link 0) frame, we may use the matrices

$$T_i = A_1 A_2 \cdot \cdot \cdot A_i. \tag{A.2-7}$$

Then, given the coordinates ^{i}r of a point expressed in the frame attached to link i, the coordinates of the same point in the base frame are given by

$$^{0}r = T_i\, {}^{i}r. \tag{A.2-8}$$

We call T_i a *kinematic chain* of transformations.

We define the *arm T matrix* as

$$T \equiv T_n = A_1 \cdot \cdot \cdot A_n, \tag{A.2-9}$$

with n the number of links in the manipulator. Then, if ^{n}r are the coordinates of a point referred to the last link, the base coordinates of the point are

$$^{0}r = T\, {}^{n}r. \tag{A.2-10}$$

This is an important relation since ${}^n r$, the coordinates of an object in the nth frame, can represent the location of the object *with respect to the tool or end effector.* It is thus important in specifying tasks to be accomplished. On the other hand, ${}^0 r$ represents the location of the object with respect to the *base frame,* which is the object's absolute position with respect to the manipulator base.

Forward Kinematics

The position and orientation of the end effector with respect to the manipulator base frame are given by evaluating the arm T matrix. It is conventional to symbolize this homogeneous transformation as

$$T = \begin{bmatrix} n & o & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}. \tag{A.2-11}$$

Thus the orientations of the axes of the end-effector reference frame are described with respect to base coordinates by the rotation matrix $R = [n \quad o \quad a]$, and the origin of the end-effector frame has a position of p in base coordinates.

The 3-vectors n, o, a, and p are defined as in Fig. A.2-2. The *approach* vector of the end effector is "a"; the *orientation* vector "o" is the direction specifying the orientation of the hand, from fingertip to fingertip. The *normal* vector "n" is chosen to complete the definition of a right-handed coordinate system using

$$n = o \times a.$$

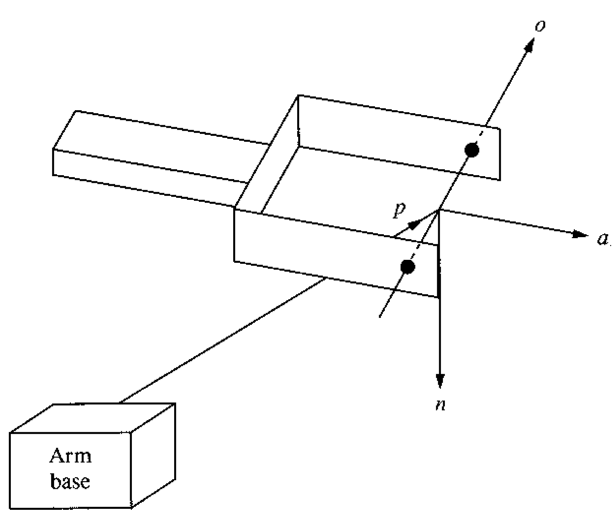


FIGURE A.2-2 Robot end effector, showing the definition of (n, o, a, p).

Thus (n,o,a) are the base coordinates of an (x,y,z) Cartesian coordinate system attached to the end effector. The *position* vector p specifies the location of the origin of the (n,o,a) frame with respect to the base frame.

The representation (n,o,a) for the orientation of the end effector is inefficient. Note that $[n \quad o \quad a]$ is a 3×3 matrix, so that it has nine entries. However, it does not take nine degress of freedom to specify orientation. Indeed, $[n \quad o \quad a]$ is a rotation matrix, so that its columns are orthogonal; this orthogonality requirement imposes extra constraints on the elements of $[n \quad o \quad a]$, so that the nine entries of $[n \quad o \quad a]$ are not independent. Alternative more efficient methods of specifying the orientation of the end effector in base coordinates are the roll–pitch–yaw, Euler angle, quaternion, and tool-configuration vector descriptions. We discuss some of these later. The reason we use (n,o,a) is that the arm T matrix is easily computed using the A matrices in terms of the arm parameters and join variables. Then n, o, a, and p are simply read off by examining the T matrix, as we shall see in the examples.

At this point we should like to distinguish between two sets of coordinates describing the end effector. The *joint variable* coordinates of the end effector are given by the n-vector.

$$q = [q_1 \quad q_2 \quad \cdots \quad q_n]^T,$$

where q_i can represent angles or lengths, depending on whether the links are revolute or prismatic. Generally, $n = 6$, so that the arm has six degrees of freedom. The end-effector *Cartesian coordinates* are the description of end-effector orientation and position in terms of the arm base coordinates. According to (A.2-11), where T may be computed knowing q, the joint variable and Cartesian coordinates are equivalent, for both specify the location of the end effector.

We say that q is the *joint-space* description of the position and orientation of the end effector, while (n,o,a,p) is the *Cartesian* or *task space* description. This terminology derives from the fact that descriptions of tasks to be performed by an arm are generally given in Cartesian coordinates, not in joint coordinates.

The robot arm *kinematics problem* is as follows. Given the joint variables q, find the Cartesian position and orientation of the end of the manipulator. Thus the kinematics problem amounts to converting given joint variables into the Cartesian position and orientation of the end effector expressed in base coordinates. Let us illustrate the solution of this problem for several simple robot arms which we use as examples throughout the book. It amounts to computing T.

It should be mentioned that there are several software packages commercially available for computing the A matrices and the T matrix for a given robot arm. See, for instance, [MATMAN 1986].

EXAMPLE A.2-1 Kinematics for Three-Link Cylindrical Arm

A simple RPP manipulator is shown in Fig. A.2-3a. It may be interpreted as the first three joints of an arm much as the GMF M-100. These are the joints used to position the end effector. A wrist mechanism consisting of three joints may be added to the end of the RPP arm to orient the end effector in space (see Example A.2-4). The joint variables of the three-link arm shown are θ, h, r, which correspond to the coordinates of a cylindrical coordinate system, so that the joint-variable vector is

$$q = [\theta \quad h \quad r]^T. \tag{1}$$

a. A Matrices

Coordinate frames may be attached to links 1, 2, and 3 using any technique desired. The D-H frames, to which (A.2.1) corresponds, are shown in Fig. A.2-3b. For this choice of frames, the A matrices may be determined as follows.

Frame 1 is related to the base frame 0 by a simple rotation of θ degrees about the axis z_0. A z-axis rotation, $R_{z,\theta}$, with no translation is described by

$$A_1 = \left[\begin{array}{ccc|c} c\theta & -s\theta & 0 & 0 \\ s\theta & c\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right], \tag{2}$$

where $c\theta$ represents $\cos\theta$ and $s\theta$ represents $\sin\theta$.

To find A_2 we may use (A.2-1). Since link 2 is prismatic with extension h, the length a_2 is zero. In this example the rotation θ_2 is also zero. The twist α_2 of link 2 is the angle of rotation about axis x_2 required to align z_1 with z_2 – that is, $-90°$.

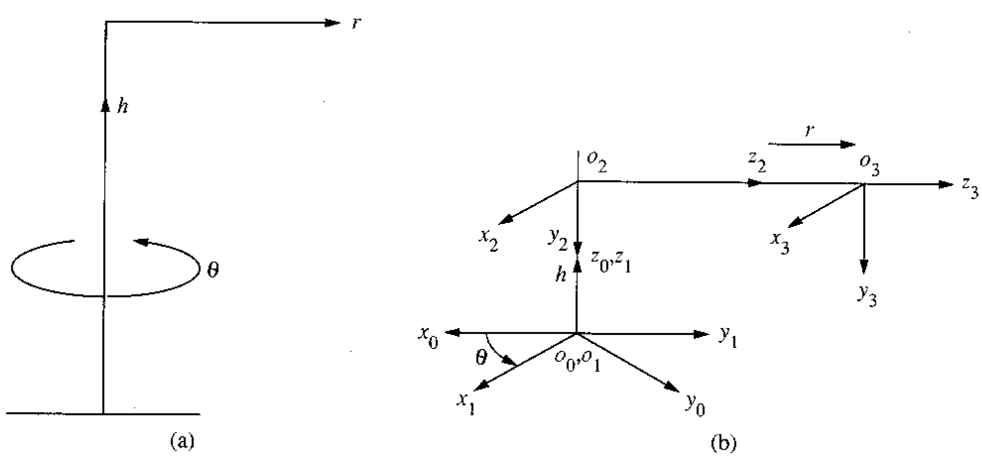


FIGURE A.2-3 Three-link cylindrical manipulator: (a) arm schematic; (b) D-H coordinate frames.

Therefore,

$$A_2 = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & h \\ \hline 0 & 0 & 0 & 1 \end{array}\right]. \tag{3}$$

There is an attractive alternative to (A.2.1) for determining the link A matrix in simple cases. Recall that the first column of A_2 is the representation of x_2 in the coordinates (x_1,y_1,z_1). According to the figure, this is just $[1 \quad 0 \quad 0]^T$. The second column of A_2 is the representation of y_2 in the coordinates (x_1,y_1,z_1), which the figure shows is $[0 \quad 0 \quad -1]^T$. The third column of A_2 is the representation of z_2 in frame o_1, which is just $[0 \quad 1 \quad 0]^T$. Thus (3) is obtained by inspection.

In similar fashion we obtain,

$$A_3 = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r \\ \hline 0 & 0 & 0 & 1 \end{array}\right]. \tag{4}$$

b. *T* Matrix and Arm Kinematics

The arm T matrix is now obtained as

$$T = A_1A_2A_3 = \left[\begin{array}{ccc|c} c\theta & 0 & -s\theta & -r\,s\theta \\ s\theta & 0 & c\theta & r\,c\theta \\ 0 & -1 & 0 & h \\ \hline 0 & 0 & 0 & 1 \end{array}\right]. \tag{5}$$

To interpret the T matrix, examine (A.2-11). The position in base coordinates of the end of the manipulator, that is, the origin of frame 3, is

$$p = [-r\,s\theta \quad r\,c\theta \quad h]^T. \tag{6}$$

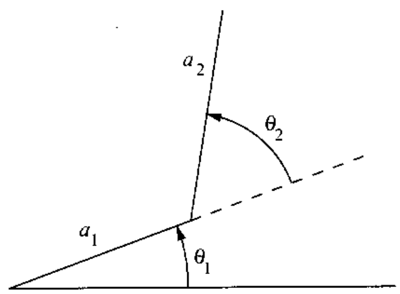
The orientation of frame 3 described in base coordinates is expressed in (n,o,a) form by giving the coordinates of the normal, orientation, and approach vectors in terms of the base frame. These are given by

$$\begin{aligned} n &= [c\theta \quad s\theta \quad 0]^T \\ o &= [0 \quad 0 \quad -1]^T \\ a &= [-s\theta \quad c\theta \quad 0]^T. \end{aligned} \tag{7}$$

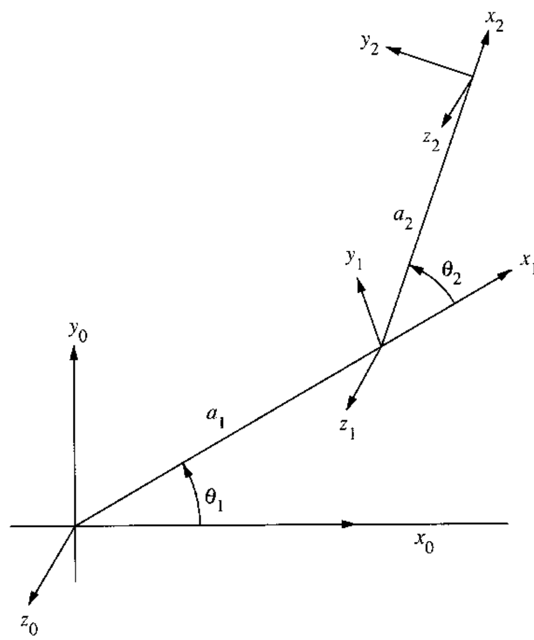
A glance at Fig. A.2.3b verifies these expressions.

EXAMPLE A.2-2 Kinematics for Two-Link Planar Elbow Arm

A two-link planar RR arm is shown in Fig. A.2-4a where a_1 and a_2 are the fixed and known link length parameters. The link coordinate frames are shown in Fig. A.2-4b. We have taken the z-axes perpendicular to the page to conform to the con-



(a)



(b)

FIGURE A.2-4 Two-link planar RR arm: (a) arm schematic; (b) D-H coordinate frames.

vention of specifying points in a plane by (x,y) coordinates. Therefore, the frames are not defined quite as in the D-H convention.

The arm A matrices may be written by inspection as

$$A_1 = \left[\begin{array}{ccc|c} c_1 & -s_1 & 0 & a_1c_1 \\ s_1 & c_1 & 0 & a_1s_1 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]. \tag{1}$$

where c_1, s_1 represent respectively $\cos\theta_1$, $\sin\theta_1$, and

$$A_2 = \left[\begin{array}{ccc|c} c_2 & -s_2 & 0 & a_2c_2 \\ s_2 & c_2 & 0 & a_2s_2 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]. \tag{2}$$

The arm T matrix is given by

$$T = A_1A_2 = \left[\begin{array}{ccc|c} c_{12} & -s_{12} & 0 & a_1c_1+a_2c_{12} \\ s_{12} & c_{12} & 0 & a_1s_1+a_2s_{12} \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]. \tag{3}$$

where $c_{12} \equiv \cos(\theta_1 + \theta_2)$ and $s_{12} \equiv \sin(\theta_1 + \theta_2)$.

Therefore, the origin o_2 of frame 2 in terms of base coordinates is located at

$$p = [a_1c_1+a_2c_{12} \quad a_1s_1+a_2s_{12} \quad 0]^T. \tag{4}$$

This represents the kinematic solution, which converts the joint variable coordinates (θ_1,θ_2) into the base-frame Cartesian coordinates of the end of the arm. The reader should examine Fig. A.2.4a to verify this expression.

EXAMPLE A.2-3 Kinematics for Two-Link Polar Arm

A two-link planar RP arm is shown in Fig. A.2-5a where ℓ is the fixed known length of the base link. The link frames, with the z–axes perpendicular to the page, are shown in Fig. A.2-5b. The joint vector is

$$q = [\theta \quad r]^T, \tag{1}$$

which corresponds to polar coordinates in the plane.

By inspection, the A matrices are found to be

$$A_1 = \left[\begin{array}{ccc|c} c\theta & -s\theta & 0 & 0 \\ s\theta & c\theta & 0 & \ell \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 1 & 1 \end{array}\right] \tag{2}$$

$$A_2 = \left[\begin{array}{ccc|c} 1 & 0 & 0 & r \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]. \tag{3}$$

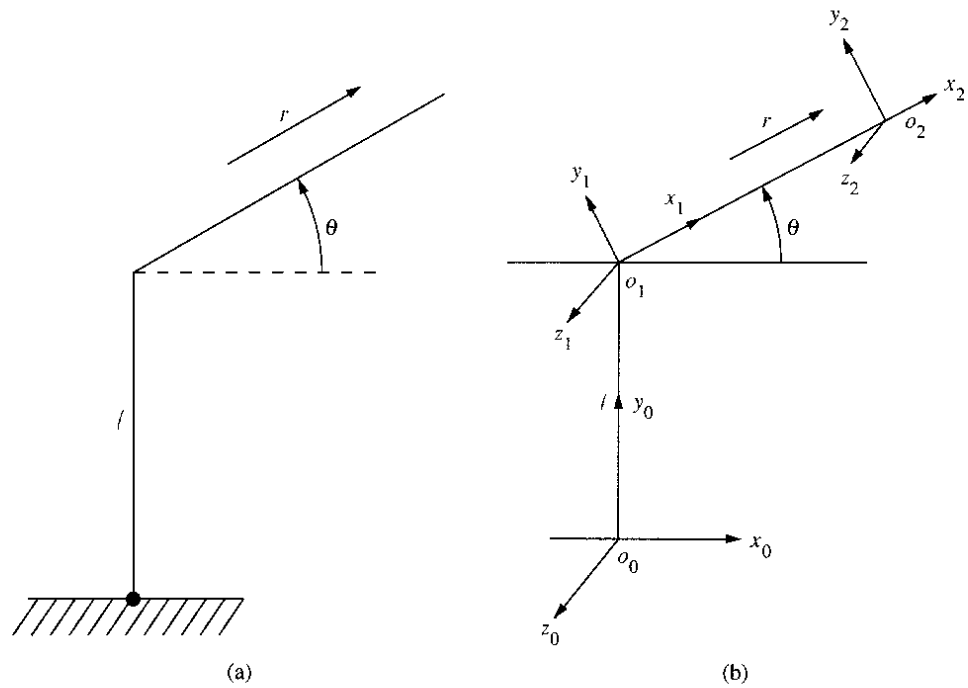


FIGURE A.2-5 Two-link planar RP arm: (a) arm schematic; (b) D-H coordinate frames.

The T matrix is

$$T = A_1A_2 = \left[\begin{array}{ccc|c} c\theta & -s\theta & 0 & rc\theta \\ s\theta & c\theta & 0 & \ell + rs\theta \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]. \tag{4}$$

Therefore, the base coordinates of the end of the arm are

$$p = [r\,c\theta \quad \ell + rs\theta \quad 0]^T, \tag{5}$$

which should be verified by examining Fig. A.2-5a. This may be interpreted as the forward kinematics solution for the arm.

This example illustrates the freedom we have to select the base frame origin o_0. We could have selected o_0 coincident with o_1. However, we have chosen to include the length ℓ of link 0 by placing o_0 at the bottom of the base link.

EXAMPLE A.2-4 Kinematics for Spherical Wrist

A spherical wrist mechanism is shown in Fig. A.2-6. See [Paul 1981, Spong and Vidyasagar 1989]. For convenient orientational control, all three joint axes intersect at a common point. Since many six-link industrial arms, including the Stanford arm, end in such a configuration, we have labeled the three joint varia-

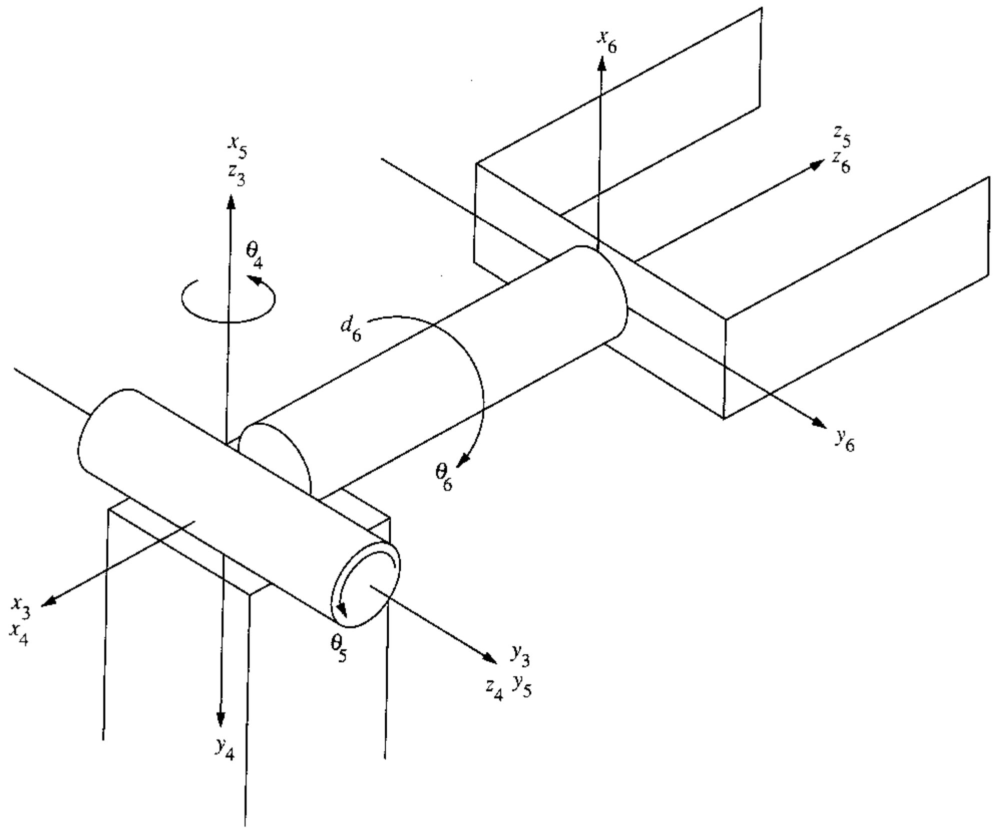


FIGURE A.2-6 Spherical wrist.

bles θ_4, θ_5, θ_6. The D-H frames are also shown in the figure. Recall that the rotation θ_i occurs about axis z_{i-1} in the D-H convention. The origin of frame 6 has been chosen at the base of the fingered gripper. The length d_6 is a fixed known parameter.

By expressing the axes x_4, y_4, z_4 in terms of the coordinates (x_3,y_3,z_3), we are able to directly determine the columns of A_4 to obtain

$$A_4 = \left[\begin{array}{ccc|c} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]. \tag{1}$$

Determining x_5, y_5, z_5 in terms of (x_4,y_4,z_4) yields

$$A_5 = \left[\begin{array}{ccc|c} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]. \tag{2}$$

In writing down A_5, it is important to note that θ_5 is defined to be zero when the end effector is in an upright position, that is, when the arm is fully extended. We have drawn the figure with $\theta_5 = -90°$ to show more clearly the different coordinate systems.

In similar fashion, we obtain

$$A_6 = \left[\begin{array}{ccc|c} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]. \tag{3}$$

The wrist T matrix is

$$T = A_4A_5A_6 = \left[\begin{array}{ccc|c} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & c_4s_5 & c_4s_5d_6 \\ s_4c_5c_6 + c_4s_6 & -s_4c_5s_6 + c_4c_6 & s_4s_5 & s_4s_5d_6 \\ -s_5c_6 & s_5s_6 & c_5 & c_5d_6 \\ \hline 0 & 0 & 0 & 1 \end{array}\right], \tag{3}$$

where $c_4 \equiv \cos\theta_4$, and so on. It is quite interesting to note that the rotational part of this T matrix corresponds to the Euler angle transformation [Spong and Vidyasagar 1989]. Therefore, θ_4, θ_5, θ_6 may be interpreted as the Euler angles with respect to frame 3.

Suppose that we would like to determine the kinematic transformations of the 3-link cylindrical arm in Example A.2-1 terminated with a spherical wrist. Then it is only necessary to multiply the T matrix in Example A.2-1 and (4), in that order, to obtain the overall manipulator T matrix.

Inverse Kinematics

The location of the end effector is specified in base coordinates by the arm T matrix. On the other hand, the location of the end effector is specified in joint coordinates by giving the values of the joint variables q_i. We may compute the T matrix knowing the joint-variable vector q, as in the examples just shown. Finding T from q is the kinematics problem.

The *inverse-kinematics* problem is as follows. Given (n,o,a,p) for the end effector in base coordinates, determine the joint variables q_i in the T matrix

$$T = \begin{bmatrix} n & o & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.2-12}$$

that yield the specified (n,o,a,p). This problem is important in manipulator control, for the desired Cartesian orientation and position of the end effector are specified by the task. Then the solution to the inverse kinematics problems gives the joint variables q_i required to achieve that orientation and position.

Due to the functions involved in the T matrix, the relations between q_i and (n,o,a,p) are highly nonlinear; these must be inverted to obtain the inverse kinematic solution q. The solutions for q_i are generally not unique. If the last three arm axes intersect at a point (e.g., the wrist mechanism in Example A.2-4), it is common to split the inverse kinematics problem into two parts. In a six-link arm, for instance, one first determines q_1, q_2, and q_3 required to obtain the desired Cartesian position p. Then the wrist variables q_4, q_5, and q_6 that give the desired orientation (n,o,a) are determined. Some techniques for solving the inverse kinematics problem are given in [Paul 1981, Craig 1989, Spong and Vidyasagar 1989]. An example is now given.

EXAMPLE A.2-5 Inverse Kinematics for Two-Link Planar Elbow Arm

For the planar RR arm of Example A.2-2, the inverse kinematics problem amounts to finding the joint variable θ_1 and θ_2 given a desired Cartesian position (x,y) of the end of the arm. See Fig. A.2-7.

The first thing that is evident is that, as long as $a_1^2 + a_2^2 < r \equiv x^2 + y^2$, there are two solutions. The one shown in Fig. A.2-7 is the "elbow down" solution. Another solution may be determined for the "elbow up" configuration, where both links are above the vector $[x \quad y]^T$. Thus *the inverse kinematics problem generally has a nonunique solution.* This may often be taken advantage of to obtain end-effector positioning with *collision avoidance.*

Given the T matrix from Example A.2-2, we see that θ_1 and θ_2 may be found by solving

$$a_1 c_1 + a_2 c_{12} = x \tag{1}$$

$$a_1 s_1 + a_2 s_{12} = y. \tag{2}$$

Determining θ_1 and θ_2 based on algebraic manipulations of such equations is called the *algebraic* inverse kinematics solution technique [Paul 1981]. Let us show a *geometric technique* here.

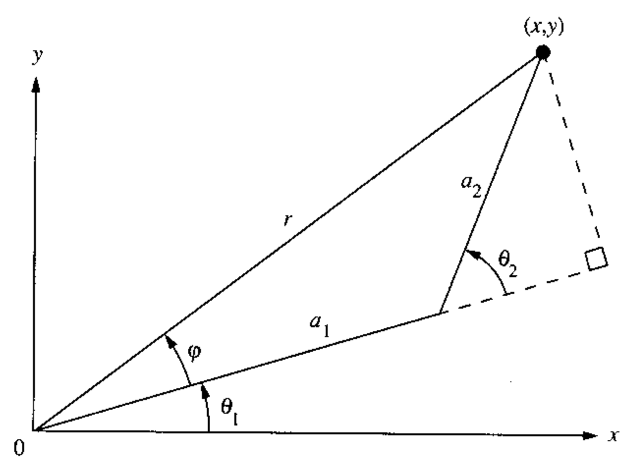


FIGURE A.2-7 Inverse kinematics for two-link planar RR arm.

Referring to Fig. A.2-7, define

$$r^2 = x^2 + y^2 \tag{3}$$

and use the law of cosines to obtain

$$\begin{aligned} r^2 &= a_1^2 + a_2^2 - 2a_1a_2 \cos(\pi - \theta_2) \\ &= a_1^2 + a_2^2 + 2a_1a_2 \cos\theta_2. \end{aligned} \tag{4}$$

We could now solve for θ_2 using the $\cos^{-1}$ function. However, it is better to use $\tan^{-1}$ for reasons of numerical accuracy. The FORTRAN function implementing $\tan^{-1}(b/c)$ is ATAN2(b,c). This function has a uniform accuracy over the range of its arguments, returns a unique value for the angle depending on the signs of b and c, and gives the correct solution if b and/or c is zero.

Therefore, we proceed by computing

$$\cos\theta_2 = \frac{r^2 - a_1^2 - a_2^2}{2a_1a_2} \equiv C \tag{5}$$

$$\sin\theta_2 = \pm\sqrt{1 - \cos^2\theta_2} = \pm\sqrt{1 - C^2} \equiv D \tag{6}$$

$$\theta_2 = \text{ATAN2(D,C)}. \tag{7}$$

An additional advantage of using the arctangent is that the multiple solutions of the inverse kinematics problem are explicitly revealed by the choice of negative or positive sign in (6).

To determine θ_1, define the auxiliary angle φ in the figure. By inspection of the right triangle shown,

$$\tan\varphi = \frac{a_2 \sin\theta_2}{a_1 + a_2\cos\theta_2}. \tag{8}$$

Moreover,

$$\tan(\varphi + \theta_1) = \frac{y}{x}, \tag{9}$$

so that

$$\theta_1 = \text{ATAN2}(y,x) - \text{ATAN2}(a_2 \sin\theta_2, a_1 + a_2 \cos\theta_2). \tag{10}$$

Note that θ_1 depends on θ_2.

A.3 The Manipulator Jacobian

Given a generally nonlinear transformation from the joint variable $q(t) \in \mathbb{R}^n$ to $y(t) \in \mathbb{R}^p$,

$$y = h(q) \tag{A.3-1}$$

we define the Jacobian associated with $h(q)$ as

$$J(q) \equiv \frac{\partial \mathrm{h}(q)}{\partial q}. \tag{A.3-2}$$

As we have just seen, the Jacobian is useful in feedback linearization, and therefore in robot manipulator control. It is also the means by which we transform velocity, acceleration, and force between coordinate frames.

Transformation of Velocity and Acceleration

Since

$$\dot{y} = \frac{\partial h}{\partial q}\,\dot{q} = J\dot{q}, \tag{A.3-3}$$

the Jacobian allows us to transform velocity from joint space to "y-space." Let us discuss the special case where $\dot{y}(t)$ is the Cartesian velocity. Then $J(q)$ is called the *manipulator Jacobian.*

It is usual to define the generalized Cartesian velocity as

$$\dot{y} = \begin{bmatrix} v \\ \omega \end{bmatrix}, \tag{A.3-4}$$

with $v = [v_x \quad v_x \quad v_z]^T$ the linear velocity and $\omega = [\omega_x \quad \omega_y \quad \omega_z]^T$ the angular velocity. For instance, ω_x represents angular velocity about the x-axis. Thus $\dot{y}$ has six components and the arm Jacobian J is a $6 \times n$ matrix, with n the number of joints in the manipulator. If $n = 6$, the Jacobian is square. We shall soon show how to compute the manipulator Jacobian.

Using (A.3-3), we can obtain an expression for the transformation of *differential motion.* Let $dq = [dq_1 \; \cdots \; dq_n]^T$ be a differential motion in joint space, with dq_i a small rotation if joint i is revolute, and a small linear displacement if joint i is prismatic. Let

$$\mathbf{dy} = \begin{bmatrix} dx \\ dy \\ dz \\ \delta x \\ \delta y \\ \delta z \end{bmatrix} \tag{A.3-5}$$

describe the same differential motion in Cartesian coordinates, with $[d_x \quad d_y \quad d_z]^T$ the differential linear motion and $[\delta x \quad \delta y \quad \delta z]^T$ representing the differential rotation. Here δx, for instance, means a small rotation about the x-axis. We have written the Cartesian differential motion vector **dy** in boldface to distinguish it from the small change dy in the second coordinate of Cartesian position.

According to (A.3-3), where $\dot{y} = dy/dt$ and $\dot{q} = dq/dt$, we see that

$$\mathbf{dy} = J\,dq, \tag{A.3-6}$$

with J the Jacobian relating joint space and Cartesian space.

The transformation of acceleration is found by differentiating (A.3-3) to be

$$\ddot{y} = J\ddot{q} + \dot{J}\dot{q}. \tag{A.3-7}$$

Transformation of Force

To discover the transformation of static force between coordinate frames, consider the following.

The *virtual work* resulting from the application of a generalized force/torque in joint space that results in a differential motion dq is

$$\delta W = \tau^T dq, \tag{A.3-8}$$

with τ the n-vector of arm control torques/forces, and $dq = [dq_1 \cdots dq_n]^T$ the differential change in the joint variable. If the description in another coordinate frame of the force is F and the description there of position is y, we must also have

$$\delta W = F^T \mathbf{dy}. \tag{A.3-9}$$

Now taking (A.3-6) into account, we may write

$$\delta W = \tau^T dq = F^T J dq,$$

so that the transformation from force to torque is given by

$$\tau = J^T(q)F. \tag{A.3-10}$$

In the case that $y(t)$ is Cartesian position, we define the Cartesian generalized force to be the 6-vector

$$F = \begin{bmatrix} f_c \\ \tau_c \end{bmatrix}, \tag{A.3-11}$$

with $\mathrm{f_c} = [f_x \;\; f_y \;\; f_z]^T$ a Cartesian force 3-vector, and $\tau_c = [\tau_x \;\; \tau_y \;\; \tau_z]^{\mathrm{T}}$ a 3-vector representing Cartesian torques. For instance, τ_x represents torque exerted about the x-axis.

If the arm has six links and the Jacobian is nonsingular, the transformation from generalized torque to generalized force is given by

$$F = J^{-T}(q)\tau. \tag{A.3-12}$$

The singularities in the Jacobian generally occur at the extremities of the manipulator workspace.

Specification of Cartesian Position

It is now necessary to confront an issue that is a source of confusion in robotics. Consider equation (A.3-2). Unfortunately, when discussing the transformation $h(q)$ from joint space to Cartesian space, this equation may only be interpreted as a convenience of notation, not as a rigorous mathematical formula. The reason is that although the generalized Cartesian velocity (A.3-4) and acceleration, and the generalized Cartesian force (A.3-11) are bona fide 6-vectors, there is a problem with conveniently specifying the generalized Cartesian position $y(t)$. We now discuss several ways to specify the Cartesian position.

Representing Generalized Cartesian Position as (*n,o,a,p*). In our context, both the *location of the origin* of the end-effector frame as well as its *orientation* must be specified in base coordinates. It is easy to specify the origin of the end-effector frame in base coordinates (x,y,z) using a 3-vector $p(t) = [p_x \quad p_y \quad p_z]^T$. However, specification of orientation is not so easy. This is because it takes more than three independent variables to specify uniquely the orientation of one frame with respect to another.

It should be mentioned that conventions such as the Euler angles and roll–pitch–yaw [Paul 1981, Spong and Vidyasagar 1989] involve only three variables. However, they may not be used to uniquely specify the *absolute* orientation of one frame with respect to another, but only relative changes in orientation. We have already seen in (A.3-5) that only three variables are needed to describe *differential rotations.*

In our work we have specified the Cartesian position $y(t)$ of the end effector in base coordinates by using the (n,o,a,p) approach (see Section A.2). There we define the generalized Cartesian position as

$$y(t) \equiv \begin{bmatrix} n(t) & o(t) & a(t) & p(t) \\ 0 & 0 & 0 & 1 \end{bmatrix} = T(t), \qquad \text{(A.3-13)}$$

with $T(t)$ the arm T matrix, (n,o,a) the vectors needed to describe the end-effector orientation, and $p = [p_x \quad p_y \quad p_z]^T$ the positional portion of the Cartesian description that specifies the origin of the end-effector frame.

It is clear that the selection of the transformation $h(q)$ depends on how we wish to describe the Cartesian position $y(t)$. If we use (A.3-13), then $h(q)$ is just the arm kinematics transformation discussed in Section A.2. However, then $y(t)$ is not a 6-vector but a 4×4 matrix. Therefore, the definition of the arm Jacobian as $J(q) = \partial h/\partial q$ is a loose one purely for notational convenience. Therefore, we are faced with the problem of providing a suitable definition for $J(q)$. Before we do this, let us discuss some more techniques for representing the generalized Cartesian position.

Representing Cartesian Orientation Using Euler's Theorem. The positional portion of y is easy to specify; it is just the 3-vector $p = [p_x \quad p_y \quad p_z]^T$ found from the last column of the T matrix in (A.3-13). However, since $[n \quad o \quad a]$ is a rotation matrix having nine elements, it does not afford an efficient representation of orientation. The orthogonality of the rotation matrix imposes some constraints among its elements, meaning that it actually has only four degrees of freedom.

The orientation of one coordinate frame (frame 1) with respect to another (frame 0) may be specified uniquely using a 3-vector k representing the axis of rotation of frame 1 with respect to frame 0, and an angle φ specifying the amount of rotation about that axis. We call (k,φ) the *Euler rotation parameters.* The (k,φ) convention involves four variables, which may be found as follows. Suppose that the rotation matrix

$$R = [n \quad o \quad a] \tag{A.3-14}$$

describes the orientation of frame 1 with respect to frame 0. Note that R is the upper left 3×3 submatrix of the arm T matrix, so that it may be computed given the joint vector q.

A rotation matrix is orthogonal, so that it has one eigenvalue equal to 1. Then the axis of rotation k is the eigenvector of the eigenvalue. That is,

$$(R - I)k = 0. \tag{A.3-15}$$

This is known as *Euler's theorem.* It says that $Rk = k$, or that the rotation axis is not rotated by R. There are many good routines for computing eigenvectors (e.g., [IMSL]), which may be used to find k from R. See the problems to find an explicit formula for k in terms of (n,o,a) [Paul 1981].

The angle of rotation φ about the axis k may easily be found, for the trace of a rotation matrix is equal to $1 + 2\cos\varphi$. Therefore,

$$\cos\varphi = \tfrac{1}{2}(n_x + o_y + a_z - 1), \tag{A.3-16}$$

where n_x, for instance, denotes the x component of the normal vector n.

Using Euler's rotation parameters we may specify the generalized Cartesian position as a 7-vector, with three components for position (i.e., the last column p of the T matrix) and four components (i.e., k and φ) for orientation.

Representing Cartesian Orientation Using Quaternions. The *quaternion* representation is a 4-vector which is often used to describe the end-effector frame orientation [Ickes 1970, Shepperd 1978, Yuan 1988, Stevens and Lewis 1991].

The quaternions are four numbers represented as (q_0,q), with $q = [q_1 \quad q_2 \quad q_3]^T$ a 3-vector. They are given in terms of the Euler parameters (k,φ) by

$$\begin{aligned} q_0 &= \cos(\varphi/2) \\ q &= k\sin(\varphi/2). \end{aligned} \tag{A.3-17}$$

Using quaternions, the generalized Cartesian position may be represented as the 7-vector $y = [p_x \quad p_y \quad p_z \quad q_0 \quad q_1 \quad q_2 \quad q_3]^T$. An advantage of this representation is that no angles are involved.

Tool-Configuration Vector. A technique for specifying $y(t)$ as a 6-vector is given in [Schilling 1990]. It depends on adopting a convention for encoding orientational information.

Consider the arm T matrix in (A.3.13) whose rotational portion is R in (A.3-14). Let us select the 3-vector p to represent Cartesian linear position. The approach vector a is a 3-vector specifying the direction in which the end effector points in terms of base coordinates. Recall from the discussion on kinematics in Section A.2 that this vector specifies in base coordinates the axis z_n of the frame attached to the end of link n. Since the rotation matrix R is orthogonal, the approach vector has a length of one.

The two vectors p and a almost completely specify the Cartesian position of the end effector. The only piece of information missing is the roll angle of the end effector. However, this angle is nothing by $q_n = \theta_n$, the last joint variable of the manipulator.

To capture the information q_n, let us propose scaling the approach vector by the exponential function $e^{q_n/\pi}$ to define the *tool-configuration vector* as

$$w = \begin{bmatrix} p \\ ae^{q_n/\pi} \end{bmatrix}. \tag{A.3-18}$$

This is a 6-vector, which uniquely specifies the Cartesian position of the end effector in base coordinates.

Note that there is a unique mapping between w and (p,a,q_n), since, given w and the fact that $\|a\| = 1$, we may compute

$$q_n = \pi \ \ln(w_4^2 + w_5^2 + w_6^2)^{1/2} \tag{A.3-19}$$

$$a = \frac{1}{(w_4^2 + w_5^2 + w_6^2)^{1/2}} \begin{bmatrix} w_4 \\ w_5 \\ w_6 \end{bmatrix}, \tag{A.3-20}$$

with w_i the ith component of w.

Finding Cartesian Velocity from the Arm T Matrix. Note that using the definition (A.3-4), the Cartesian velocity $\dot{y}$ is a 6-vector that is not strictly speaking the derivative of $y(t)$ in (A.3-13), which is a 4×4 matrix. To compute $\dot{y}$, we may find the Jacobian $J(q)$ and then use (A.3-3). Alternatively, the Cartesian velocity may be found directly from the arm T matrix (A.2-11) as follows.

Setting $\dot{y} = [v^T \quad \omega^T]^T$, with v the linear velocity and ω the angular velocity, it is clear that

$$v = \dot{p}, \tag{A.3-21}$$

with p found from the last column of T.

To find ω, proceed as follows. Since R in (A.2-11) is orthogonal, we have

$$RR^T = I, \tag{A.3-22}$$

which on differentiation yields

$$\dot{R}R^T + R\dot{R}^T = 0.$$

Therefore, the matrix defined as

$$\Omega \equiv \dot{R}R^T \tag{A.3-23}$$

satisfies $\Omega + \Omega^T = 0$, so it is skew symmetric. It may therefore be represented as

$$\Omega = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \tag{A.3-24}$$

The relation between the *cross-product matrix* Ω and the angular velocity vector

$$\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{A.3-25}$$

is an interesting one, for one may easily demonstrate that

$$\omega \times w = \Omega w \tag{A.3-26}$$

for any 3-vector w. That is, the cross product may be replaced by a matrix multiplication in terms of the cross-product matrix. We denote by $\Omega(\omega)$ the cross-product matrix associated with a vector ω.

The complete procedure for determining $\dot{y}$ from the T matrix is then as follows. First, compute v using the p vector as in (A.3-21). Then, compute $\Omega(\omega)$ from the rotation matrix R using (A.3-23), and hence find the last three components ω of $\dot{y}$ from the definition of $\Omega(\omega)$. Computing $\dot{y}$ from q thus requires a *computer subroutine.* Robot controllers generally have block diagrams in which some of the blocks are standard components like integrators, but some of the blocks are implemented in software.

Using (A.3-23), we may write the *strapdown equation*

$$\dot{R} = \Omega R, \tag{A.3-27}$$

which is of fundamental importance in inertial navigation [Stevens and Lewis 1991].

Computing the Arm Jacobian

Let us now return to the problem of defining and computing the arm Jacobian $J(q)$. If the Cartesian position is defined by (A.3-13), then (A.3-2) does not afford an appropriate definition. Therefore, let us select (A.3-3) as the definition of $J(q)$. That is, define $J(q)$ by

$$\dot{y} = J(q)\dot{q}, \tag{A.3-28}$$

so that it maps the joint velocity $\dot{q}$ to the Cartesian velocity $\dot{y}$ as defined by (A.3-4). Then $J(q)$ is a $6 \times n$ matrix.

Let us consider a few examples showing how to compute the arm Jacobian to demonstrate that the procedure is not complicated. A methodical technique for computing $J(q)$ will then be given.

EXAMPLE A.3-1: Arm Jacobian for Three-Link Cylindrical Arm

Let us use the kinematics derived in Example A.2-1 to compute the Jacobian for the three-link cylindrical arm. We shall call the joint variables $(\theta,z,r,)$ instead of (θ,h,r) to avoid confusion with the nonlinear function $h(q)$.

Since there is no wrist on this three-link arm, the orientation of the last coordinate frame is fixed in terms of (θ,z,r), so that is not necessary to specify Cartesian orientation. This simplifies things for this example.

Denote the linear Cartesian position of the end effector in base coordinates (i.e., the first three components of y) as y_p. Then, according to Example A.2-1, y_p is the last column of the arm T matrix, so that

$$y_p = \begin{bmatrix} -r\sin\theta \\ r\cos\theta \\ z \end{bmatrix} \equiv h_p(q). \tag{1}$$

Omitting the orientational information has allowed us to define a (positional) transformation function $h_p(q)$ simply as the last column p of the T matrix.

Using (A.3-2) and the definition of $h_p(q)$ given in (1), the Jacobian is

$$J(q) = \frac{\partial h_p}{\partial q} = \begin{bmatrix} \frac{\partial h_p}{\partial \theta} & \frac{\partial h_p}{\partial z} & \frac{\partial h_p}{\partial r} \end{bmatrix} = \begin{bmatrix} -r\cos\theta & 0 & -\sin\theta \\ -r\sin\theta & 0 & \cos\theta \\ 0 & 1 & 0 \end{bmatrix}. \tag{2}$$

Therefore, the velocities of the joints are converted into Cartesian velocities using

$$\frac{dy_p}{dt} = \begin{bmatrix} -r\cos\theta & 0 & -\sin\theta \\ -r\sin\theta & 0 & \cos\theta \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{z} \\ \dot{r} \end{bmatrix}. \tag{3}$$

The determinant of J is $(r \cos^2\theta + r \sin^2\theta) = r$, so that J is nonsingular as long as $r \neq 0$.

Note that the definition (1) of the positional function $h_p(q)$ means that the Jacobian is equal to $\partial h_p/\partial q$. This is a result of neglecting the orientation portion of y.

EXAMPLE A.3-2: Arm Jacobian for Two-Link Planar Elbow Arm

In Example A.2-2 the joint variable was $q = [\theta_1 \quad \theta_2]^T$. As in Example A.3-1, the orientation of the last coordinate frame is fixed once q is given, so that we are not concerned with the orientational portion of y, but only its linear position portion y_p.

The linear Cartesian position of the end effector in the (x_1,x_2)-plane is given by the last column of the arm T matrix as

$$y_p = \begin{bmatrix} a_1 \cos \theta_1 + a_2 \cos (\theta_1 + \theta_2) \\ a_1 \sin \theta_1 + a_2 \sin (\theta_1 + \theta_2) \end{bmatrix} \equiv h_p(q). \tag{1}$$

Since the motion is constrained to the plane, we have suppressed the x_3 component of y_p.

The Jacobian is

$$J = \frac{\partial h_p}{\partial q} = \begin{bmatrix} -a_1 \sin \theta_1 - a_2 \sin (\theta_1 + \theta_2) & -a_2 \sin (\theta_1 + \theta_2) \\ a_1 \cos \theta_1 + a_2 \cos (\theta_1 + \theta_2) & a_2 \cos (\theta_1 + \theta_2) \end{bmatrix}. \tag{2}$$

Note that the definition of $h_p(q)$ in (1) means that the Jacobian is equal to $\partial h_p/\partial q$.

The determinant of J is (verify!) equal to $a_1a_2 \sin \theta_2$, so that J is nonsingular unless $\theta_2 = 0$ or π; that is, unless the arm is fully extended or link 2 is folded back on top of link 1.

EXAMPLE A.3-3: Jacobian for Transformation to Camera Coordinates

Consider the three-link cylindrical arm of Examples A.2-1 and A.3-1 with a camera mounted vertically above the arm that measures only the position (x_1,x_2) of the end effector in the horizontal plane (in base coordinates). Call this position y. Then, according to those examples, the transformation from joint coordinates (θ,z,r) to camera coordinates (x_1,x_2) is

$$y = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -r \sin \theta \\ r \cos \theta \end{bmatrix} \equiv h(q) \tag{1}$$

and the associated Jacobian is

$$J = \begin{bmatrix} -r\cos\theta & 0 & -\sin\theta \\ -r\sin\theta & 0 & \cos\theta \end{bmatrix}. \tag{2}$$

The determinant of J with the second column deleted is equal to $-r$, so that J has full row rank as long as $r \neq 0$. The pseudo-inverse of J is

$$J^+ = \begin{bmatrix} -(\cos\theta)/r & -(\sin\theta)/r \\ 0 & 0 \\ -\sin\theta & \cos\theta \end{bmatrix}. \tag{3}$$

Since the camera cannot measure the distance x_3 perpendicular to the plane, that coordinate must be selected in any control scheme using an independent means that does not involve trajectory following control in the horizontal plane.

Algorithm for Computing the Arm Jacobian. We are finally in a position to show how to compute the arm Jacobian given the joint variables q_i. The procedure follows.

Given q_i, compute the matrices T_i defined in Section A.2 as

$$T_i = A_1 A_2 \cdots A_i \equiv \begin{bmatrix} R_i & p_i \\ 0 & 1 \end{bmatrix}, \tag{A.3-29}$$

whose corresponding rotation matrices will be denoted

$$R_i = [x_i \quad y_i \quad z_i]. \tag{A.3-30}$$

Define $T_o = I$, $R_o = I$, and the arm T matrix

$$T = T_n = \begin{bmatrix} R_n & p_n \\ 0 & 1 \end{bmatrix} \equiv \begin{bmatrix} n & o & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{A.3-31}$$

The vector z_i represents the z-axis of frame i in base coordinates. The vector p represents the location of the origin of link frame n (the end-effector frame) in terms of base coordinates. The Jacobian is computed using the vectors p and z_i as follows.

The generalized Cartesian velocity is $\dot{y} = [v^T \quad \omega^T]^T$. Therefore, we may partition the Jacobian matrix into a linear and an orientational part by writing

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = J(q)\dot{q} = \begin{bmatrix} J_p(q) \\ J_o(q) \end{bmatrix} \dot{q}, \tag{A.3-32}$$

with $J_p(q)$ the first three rows of $J(q)$ and $J_o(q)$ its last three rows.

First, consider the computation of the linear position Jacobian $J_p(q)$. Given that the linear portion of the generalized Cartesian position y is just p, we may write

$$v = \sum_{i=1}^{n} \frac{\partial p}{\partial q_I} \dot{q}_i = \left[\frac{\partial p}{\partial q_1} \quad \frac{\partial p}{\partial q_2} \cdots \frac{\partial p}{\partial q_n} \right] \dot{q}. \tag{A.3-33}$$

Therefore, $J_p(q)$ is given by

$$J_p(q) = \left[\frac{\partial p}{\partial q_1} \quad \frac{\partial p}{\partial q_2} \cdots \frac{\partial p}{\partial q_n} \right]. \tag{A.3-34}$$

It should be clearly understood that this is exactly what we used in Examples A.3-1 and A.3-2.

Let us now turn to the orientational portion $J_o(q)$ of the Jacobian. Exactly as for linear velocities, one may add angular velocities as long as they are represented in the same coordinate frame. Let us therefore add the individual angular velocities of the links in the arm to obtain the angular velocity of the end effector. A prismatic joint does not contribute to the angular velocity of the end effector.

For a revolute joint, the joint rotation $q_i = \theta_i$ occurs about joint axis $z_{i\text{-}1}$ (see Section A.2, especially Fig. A.2-1). The angular velocity for joint variable i is therefore given by $z_{i-1}\dot{q}_i$. To add the effects of all the links, it is necessary to express $z_{i\text{-}1}$ in a common frame; we select base coordinates. However, the last column of $R_{i\text{-}i}$ is exactly $z_{i\text{-}1}$ in base coordinates. Therefore, we may write

$$\omega = \sum_{i=1}^{n} \kappa_i z_{i-1} \dot{q}_i = [\kappa_1 z_0 \quad \kappa_2 z_1 \quad \cdots \quad \kappa_n z_{n-1}]\, \dot{q}. \tag{A.3-35}$$

where $z_0 = [0 \quad 0 \quad 1]^T$ and the selection parameter κ_i is zero if q_i is prismatic and 1 if q_i is revolute. Thus

$$J_o(q) = [\kappa_1 z_0 \quad \kappa_2 z_1 \quad \cdots \quad \kappa_n z_{n-1}]. \tag{A.3-36}$$

The complete Jacobian is now given by stacking $J_l(q)$ on top of $J_o(q)$. It should now be clear that the arm Jacobian must be found from the joint vector q using a significant amount of computation. Therefore, whenever the Jacobian is required in a robot arm control scheme, it should be computed using a *computer subroutine*.

It is a good exercise to rework Examples A.3-1 and A.3-2 to determine the complete Jacobian, including the angular portion (see the problems).

EXAMPLE A.3-4: Jacobian for Spherical Wrist

In Example A.2-4 we derived the kinematics for a spherical wrist. Although the wrist would normally terminate an arm, for illustration let us take link 3 as the base link in this example. See Fig. A.2-6. Thus we shall determine the Jacobain in the coordinates of the link 3 frame of reference.

To find the Jacobian, we compute the matrices

$$T_3 = I$$

$$T_4 = A_4 = \left[\begin{array}{ccc|c} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

$$T_5 = A_4A_5 = \left[\begin{array}{ccc|c} c_4c_5 & -s_4 & c_4s_5 & 0 \\ s_4c_5 & c_4 & s_4s_5 & 0 \\ -s_5 & 0 & c_5 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

$$T = T_6 = A_4A_5A_6 = \left[\begin{array}{ccc|c} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & c_4s_5 & c_4s_5d_6 \\ s_4c_5c_6 + c_4s_6 & -s_4c_5s_6 + c_4c_6 & s_4s_5 & s_4s_5d_6 \\ -s_5c_6 & s_5s_6 & c_5 & c_5 6_6 \\ \hline 0 & 0 & 0 & 1 \end{array}\right].$$

Using the approach just derived, we compute directly that

$$J_p = \begin{bmatrix} \dfrac{\partial p}{\partial \theta_4} & \dfrac{\partial p}{\partial \theta_5} & \dfrac{\partial p}{\partial \theta_6} \end{bmatrix} = \begin{bmatrix} -s_4s_5d_6 & c_4c_5d_6 & 0 \\ c_4s_5d_6 & s_4c_5d_6 & 0 \\ 0 & -s_5d_6 & 0 \end{bmatrix}$$

$$J_o = [z_3 \quad z_4 \quad z_5] = \begin{bmatrix} 0 & -s_4 & c_4s_5 \\ 0 & c_4 & s_4s_5 \\ 1 & 0 & c_5 \end{bmatrix}.$$

REFERENCES

Craig, J. J., *Introduction to Robotics.* Reading, MA: Addison-Wesley, 1989.

Ickes, B. P., "A new method for performing digital control system altitude computations using quaternions," *AIAA J.*, vol. 8, no. 1, pp. 13–17, Jan. 1970.

IMSL, *Library Contents Document,* 8th ed. Houston, TX: International Mathematical and Statistical Libraries.

MATMAN, *Symbolic Matrix Manipulation,* M. R. Driels, Pembroke, MA: Kern International, 1986.

Paul, R. P., *Robot Manipulators,* Cambridge, MA: MIT Press, 1981.

Schilling, R. J., *Fundamentals of Robotics,* Englewood Cliffs, NJ: Prentice Hall, 1990.

Sheppard, S. W., "Quaternion from rotation matrix," *Eng. Notes,* pp. 223–224, May/June 1978.

Spong, M. W., and M. Vidyasagar, *Robot Dynamics and Control.* New York: Wiley, 1989.

Stevens, B. L., and F. L. Lewis, *Aircraft Modeling, Dynamics, and Control.* New York: Wiley, 1992.

Yuan, J. S.-C., "Closed-loop manipulator control using quaternion feedback," *IEEE J. Robot. Autom.,* vol. 4, no. 4, pp. 434–440, Aug. 1988.