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-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-1} in a common frame; we select base coordinates. However, the last column of R_{i-1} is exactly z_{i-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 $J_i(q)$ 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

$$T_3 = I$$

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

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

$$T = T_6 = A_4A_5A_6 = \left[\begin{array}{ccc|c} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & c_4s_5 & c_4s_5d_6 \\ s_4c_5c_6 + c_4s_6 & -s_4c_5s_6 + c_4c_6 & s_4s_5 & s_4s_5d_6 \\ -s_5c_6 & s_5s_6 & c_5 & c_56_6 \\ \hline 0 & 0 & 0 & 1 \end{array}\right].$$

Using the approach just derived, we compute directly that

$$J_p = \left[\frac{\partial p}{\partial \theta_4} \quad \frac{\partial p}{\partial \theta_5} \quad \frac{\partial p}{\partial \theta_6}\right] = \begin{bmatrix} -s_4s_5d_6 & c_4c_5d_6 & 0 \\ c_4s_5d_6 & s_4c_5d_6 & 0 \\ 0 & -s_5d_6 & 0 \end{bmatrix}$$

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

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.