CHAPTER 6

KINEMATICS AND DYNAMICS OF ROBOT MANIPULATORS

Andrew A. Goldenberg
Mohammad R. Emami
University of Toronto
Toronto, Ontario

1 INTRODUCTION

This chapter reviews current practical methodologies for kinematics and dynamics modeling and calculations. A kinematics model is a representation of the motion of the robot manipulator without considering masses and moments of inertia; a dynamics model is a representation of the balancing of external and internal loads acting on the manipulator whether it is stationary or moving. Both models are used widely in design, simulation, and, more recently, real-time control.

These topics are considered fundamental to the study and use of robotics. In the early development of this branch of science and engineering, kinematics and dynamics modeling were the main topics treated in the literature. Over the years, kinematics and dynamics modeling have generated the greatest number of publications related to robotics. This chapter attempts to extract some of the most relevant issues, but does not provide a summary of all the published work. Its aim is to present the standard tools for kinematics and dynamics modeling without prerequisites. The reader is also referred to the corresponding chapter in the first edition of this handbook (Walker, 1985).

TERMINOLOGY

n	number of degrees of freedom of the manipulator
$\mathbf{q}$	vector of joint variable displacements
$\dot{\mathbf{q}}$	vector of joint variable velocities
$\ddot{\mathbf{q}}$	vector of joint variable accelerations
$\mathbb{Q}$	joint space
$\mathbb{T}$	task space
$\mathbf{F}_i$	coordinate frame attached to link i
$R_{i-1,i}$	3×3 rotation matrix between frames $(i-1)$ and i
$\mathbf{p}_{i-1,i}$	position vector of the origin of frame i with respect to frame $(i-1)$ expressed in frame $(i-1)$
$H_{i-1,i}$	4×4 homogeneous transformation matrix between frames $(i-1)$ and i
$\boldsymbol{\epsilon}$	axis of general rigid body rotation
ν	axis of general rigid body translation
Ψ	4×4 twist matrix
$W(P_w)$	wrist workspace
$W(P_e)$	end-effector workspace
$\boldsymbol{\omega}_i$	angular velocity vector of frame i
$\dot{\boldsymbol{\omega}}_i$	angular acceleration vector of frame i
$\boldsymbol{\nu}_i$	linear velocity vector of the origin of frame i
$\dot{\boldsymbol{\nu}}_i$	linear acceleration vector of the origin of frame i
$\mathbf{v}_{ci}$	linear velocity vector of the center of mass of link i
$\dot{\boldsymbol{\nu}}_{ci}$	linear acceleration vector of the center of mass of link i
$\mathbf{p}_{ci}$	position vector of the center of mass of link i with respect to frame i

Handbook of Industrial Robotics, Second Edition. Edited by Shimon Y. Nof

I_{ci} 3×3 moment of inertia matrix of link i about its center of mass expressed in frame i
m_i mass of link i
J $6 \times n$ Jacobian matrix
$\boldsymbol{\tau}$ vector of joint torques and forces
$\mathbf{f}_e$ vector of resulting reaction forces at the end-effector
$\mathbf{g}_e$ vector of resulting reaction moments at the end-effector
$\mathbf{f}_{ext}$ vector of external forces acting at the end-effector
$\mathbf{g}_{ext}$ vector of external moments acting at the end-effector
$\mathbf{G}_e$ wrench vector
M $n \times n$ manipulator inertia matrix
$\mathcal{G}$ gravity acceleration vector

2 KINEMATICS

This section considers the motion of the robot manipulator irrespective of inertial and external forces. A study of the geometry of motion is essential in manipulator design and control in order to obtain the mapping between the end-effector location (position and orientation) and the movement of manipulator links, as well as the mapping between the end-effector velocity and the speed of manipulator links. The final goal is to use these mappings to relate the end-effector (or gripper, or tool mounted on the end-effector) motion to joint displacements (generalized coordinates) and velocities.

2.1 Forward Kinematics

The objective of forward kinematics is to determine the location of the end-effector with respect to a reference coordinate frame as a result of the relative motion of each pair of adjacent links. Attention is restricted to the case of an *open-chain* manipulator, a serial link of rigid bodies connected in pairs by *revolute* and/or *prismatic* joints, for relative rotation and relative translation, respectively.

2.1.1 Different Configuration Spaces for Robot Manipulators

The configuration of a robot manipulator can be specified using either of the following algebraic spaces:

1. The *joint space* $\mathcal{Q}$ is the set of all possible vectors of joint variables. The dimension of the joint vector is equal to the number of joints (or degrees of freedom), i.e., $\mathcal{Q} \subset \mathbb{R}^n$. Each joint variable is defined as an angle $\theta \in \mathcal{S} = [0, 2\pi)$ for a revolute joint, or a linear translation $d \in \mathbb{R}$ for a prismatic joint. Let $\mathbf{q} \in \mathcal{Q}$ denote the vector of generalized coordinates.
2. The *task space* $\mathcal{T}$ is the set of pairs $(\mathbf{p}, R)$, where $\mathbf{p} \in \mathbb{R}^3$ is the position vector of the origin of link coordinate frame and $R \in SO(3)$ represents the orientation of the link frame, both with respect to a general reference frame. Here, $SO(3)$ denotes the group of 3×3 proper rotation matrices. Thus, the task space is a *Special Euclidean* group $SE(3)$, defined as follows:

$$SE(3) = \{(\mathbf{p}, R): \mathbf{p} \in \mathbb{R}^3, R \in SO(3)\} = \mathbb{R}^3 \times SO(3). \quad (1)$$

Using the above notation, the forward kinematics is a mapping H, defined as follows:

$$H: \mathcal{Q} \rightarrow SE(3). \quad (2)$$

This mapping can be represented by a 4×4 *homogeneous transformation* matrix, defined as

$$\left[\begin{array}{c|c} R & \mathbf{p} \\ \hline [0] & 1 \end{array}\right].$$

2.1.2 The End-Effector Position and Orientation

In order to obtain the forward kinematics mapping H, suitable coordinate frames should be assigned to the manipulator base, end-effector, and intermediate links. One standard method attributed to Denavit and Hartenberg (1965) is based on the homogeneous transformation H. The Denavit–Hartenberg (DH) convention uses the minimum number of parameters to completely describe the geometric relationship between adjacent robot links.

Each link and joint of the manipulator is numbered, as illustrated in Figure 1. The frame $\mathbf{F}_i$ attached to link i is defined with the $\mathbf{z}_i$ along the axis of joint $(i + 1)$; the origin is located at the intersection of $\mathbf{z}_i$ and the common normal to $\mathbf{z}_{i-1}$ and $\mathbf{z}_i$, and $\mathbf{x}_i$ is along the common normal, as illustrated in Figure 1. The homogeneous transformation matrix between links i and $(i - 1)$ is then expressed as (Walker, 1985):

$$H_{i-1,i} = \left[\begin{array}{c|c} R_{i-1,i} & \mathbf{p}_{i-1,i} \\ \hline [0] & 1 \end{array}\right] = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \cos\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

where

$R_{i-1,i}$ = relative rotation of frame $\mathbf{F}_i$ with respect to $\mathbf{F}_{i-1}$;
$\mathbf{p}_{i-1,i}$ = position vector of the origin of $\mathbf{F}_i$ with respect to $\mathbf{F}_{i-1}$, expressed in $\mathbf{F}_{i-1}$;
$[0]$ = 1×3 null matrix.

The three link parameters a_i, α_i, d_i and one joint variable θ_i required to specify the transformation (3) are defined as follows:

$a_i \equiv$ the length of the common normal between $\mathbf{z}_{i-1}$ and $\mathbf{z}_i$ (link length)
$\alpha_i \equiv$ the angle between $\mathbf{z}_{i-1}$ and $\mathbf{z}_i$ measured about $\mathbf{x}_i$ (twist angle)
$d_i \equiv$ the distance from $\mathbf{x}_{i-1}$ to $\mathbf{x}_i$ measured along $\mathbf{z}_{i-1}$ (link offset or distance)
$\theta_i \equiv$ the angle between $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$ measured about $\mathbf{z}_{i-1}$.

The homogeneous transformation between the base frame $\mathbf{F}_0$ and the end-effector frame $\mathbf{F}_n$ (for an n-d.o.f. manipulator) can then be systematically determined by successive multiplication of the intermediate transformations, namely:

$$H_{0n} = H_{01}H_{12} \cdots H_{i-1,i} \cdots H_{n-2,n-1}H_{n-1,n} \tag{4}$$

The matrix H_{0n} contains the rotation matrix between frames $\mathbf{F}_0$ and $\mathbf{F}_n$ (R_{0n}), and the location of the origin of $\mathbf{F}_n$ with respect to $\mathbf{F}_0$, expressed in $\mathbf{F}_0$:

$$H_{0n} = \left[\begin{array}{c|c} R_{0n} & p_{0n} \\ \hline [0] & 1 \end{array}\right] = \left[\begin{array}{ccc|c} n_{0n}^x & o_{0n}^x & a_{0n}^x & p_{0n}^x \\ n_{0n}^y & o_{0n}^y & a_{0n}^y & p_{0n}^y \\ n_{0n}^z & o_{0n}^z & a_{0n}^z & p_{0n}^z \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \tag{5}$$

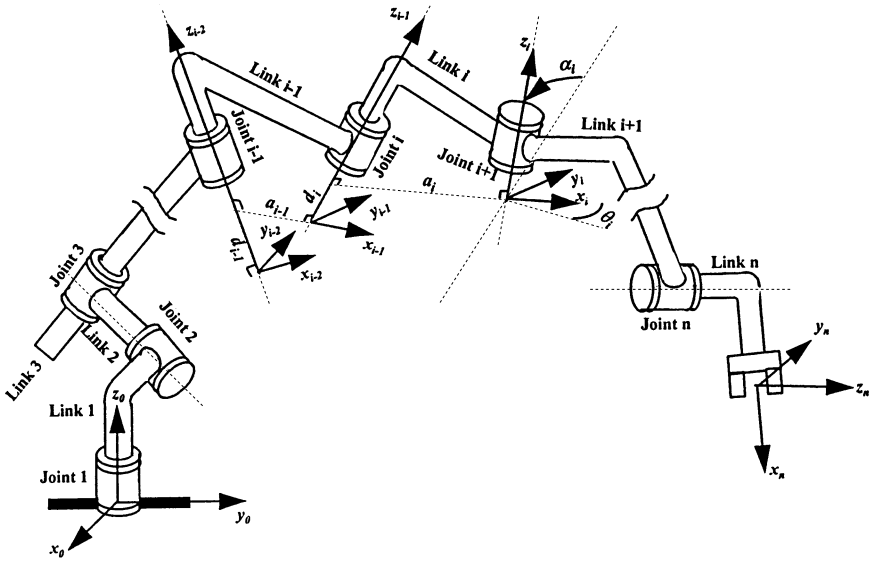

Figure 1 DH link frames and parameters.

where vectors $\mathbf{n}_{0n}$, $\mathbf{o}_{0n}$ and $\mathbf{a}_{0n}$ specify the orientation of the $\mathbf{x}_n$, $\mathbf{y}_n$ and $\mathbf{z}_n$ axes, respectively, of frame $\mathbf{F}_n$ with respect to frame $\mathbf{F}_0$.

2.1.3 Standard Coordinate Frames

Figure 2 shows some of the standard frames commonly used in industrial applications. The position of the origin and the orientation of each frame with respect to the base frame is obtained by successive multiplications of the intermediate homogeneous transformation matrices. For example, the representation of the tool frame with respect to the base frame is determined by

$$H_{0t} = H_{0n}H_{nt} \tag{6}$$

where H_{nt} and H_{0n} are the homogeneous transformation matrices between the end-effector and the tool frames and between the end-effector and the base frames, respectively.

2.1.4 Computational Considerations and Symbolic Formulation

In practical applications it is always effective to minimize the computational time required to perform the kinematic calculations. The calculations can be performed recursively because the open-chain manipulator can be seen as being constructed by adding a link to the previous links. This reduces the number of multiplications and additions at the cost of creating local variables in order to avoid the use of common terms throughout the computation. Algorithm 1 illustrates a backward recursive formulation for calculating the forward kinematics (Hoy and Sriwattanathamma, 1989).

Symbolic kinematic equations that describe the end-effector (or tool) position and orientation as explicit functions of joint coordinates can be derived in advance of real-time computation. If suitable trigonometric simplifications are implemented, symbolic representation helps to reduce the number of arithmetic operations. Either general-purpose (such as MATHEMATICA and MAPLE) or special-purpose (such as SD/FAST (Westmacott, 1993)) symbolic modeling software can replace manual derivation to generate symbolic kinematic relationships automatically (Vukobratovic, 1986).

Transcendental functions are a major computational expense in forward kinematics calculations when standard software is used. Instead, lookup table implementations of these functions may reduce the required calculation time by a factor of two to three, or more (Ruoff, 1981). Moreover, using fixed-point instead of floating-point representation can speed up the operations. A 24-bit representation of joint variables is adequate due to the typically small dynamic range of these variables (Turner, Craig, and Gruver, 1984).

2.1.5 Manipulator Workspace

Evaluation of the manipulator workspace is a subject of interest for purposes of both analysis and synthesis. The workspace of a manipulator is defined as the set of all end-effector locations (positions and orientations of the end-effector frame) that can be reached by arbitrary choices of joint variables within the corresponding ranges. If both end-effector

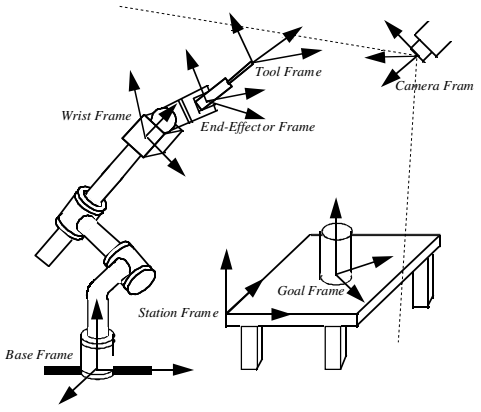

Figure 2 Standard coordinate frames.

LOOP: FOR $i = n - 1$ to 1

1. SET: $R_{i,n} = \begin{bmatrix} n^x_{i,n} & o^x_{i,n} & a^x_{i,n} \\ n^y_{i,n} & o^y_{i,n} & a^y_{i,n} \\ n^z_{i,n} & o^z_{i,n} & a^z_{i,n} \end{bmatrix} = \begin{bmatrix} [R^x_{i,n}] \\ [R^y_{i,n}] \\ [R^z_{i,n}] \end{bmatrix}$

2. CALCULATE:
$$\begin{cases} [M_{i,n}] = \cos\alpha_i[R^y_{i,n}] - \sin\alpha_i[R^z_{i,n}] \\ r_{i,n} = \cos\alpha_i p^y_{i,n} - \sin\alpha_i p^z_{i,n} \\ s_{i,n} = \begin{cases} p^x_{i,n} + a_i & \text{: if joint } i \text{ is revolute} \\ p^x_{i,n} & \text{: if joint } i \text{ is prismatic} \end{cases} \end{cases}$$

3. CALCULATE:
$$\begin{cases} [R^x_{i-1,n}] = \cos\theta_i[R^x_{i,n}] - \sin\theta_i[M_{i,n}] \\ [R^y_{i-1,n}] = \sin\theta_i[R^x_{i,n}] + \cos\theta_i[M_{i,n}] \\ [R^z_{i-1,n}] = \sin\alpha_i[R^y_{i,n}] + \cos\alpha_i[R^z_{i,n}] \\ p^x_{i-1,n} = \cos\theta_i s_{i,n} - \sin\theta_i r_{i,n} \\ p^y_{i-1,n} = \sin\theta_i s_{i,n} + \cos\theta_i r_{i,n} \\ p^z_{i-1,n} = \sin\alpha_i p^y_{i,n} + \cos\alpha_i p^z_{i,n} + d_i \end{cases}$$

NEXT i

Algorithm 1 Backward recursive formulation of the forward kinematics problem.

position and orientation are considered, the workspace is the *complete workspace;* disregarding the orientation of the end-effector gives the *reachable workspace.* The subset of the reachable workspace that can be attained with arbitrary orientations of the end-effector is the *dexterous workspace.* Most industrial manipulators have spherical wrists; therefore, for a 6-d.o.f. manipulator, the wrist is positioned using the first three joints. If the wrist point P_w and the end-effector point of interest P_e are different, then, after the workspace of the wrist point $W(P_w)$ is determined, a sphere of radius $\overline{P_wP_e}$ is moved so that its center is on the boundary of the wrist workspace $W(P_w)$. The inner and outer envelopes are the boundaries of the dexterous and reachable workspaces, respectively. Nevertheless, due to machining tolerances, assembly errors, and other limitations, it is impossible to build a perfect wrist with three orthogonal revolute axes intersecting at one point. Thus a general methodology is required for determining the manipulator workspace (Ceccarelli, 1994).

2.1.6 The Product of Exponentials (PE) Formula

A geometric description of the robot kinematics can be obtained based on classical *screw theory* (Yuan, 1971). The fundamental fact is that any arbitrary rigid body motion is equivalent to a rotation θ about a certain line ε ($\|\varepsilon\| = 1$) combined with a translation l parallel to that line. The homogeneous transformation of the rigid body motion with respect to a reference frame can then be represented as

$$H(\theta, l) = e^{\Psi\theta} H(0, 0) \tag{7}$$

where $H(0, 0)$ is the initial homogeneous representation of the rigid body with respect to the same reference frame when $\theta = 0$ and $l = 0$. In Equation (7), Ψ is a 4×4 matrix called the *twist* and is defined as

$$\Psi = \left[\begin{array}{c|c} E & -E\mathbf{p} + (l/\theta)\boldsymbol{\epsilon} \\ \hline [0] & 0 \end{array}\right] \tag{8}$$

where E is a 3×3 skew-symmetric matrix of the rotation axis $\boldsymbol{\epsilon} = [\epsilon_x \ \epsilon_y \ \epsilon_z]^T$ such that

$$E = \begin{bmatrix} 0 & -\epsilon_z & \epsilon_y \\ \epsilon_z & 0 & -\epsilon_x \\ -\epsilon_y & \epsilon_x & 0 \end{bmatrix} \tag{9}$$

and $\mathbf{p} = [p_x \ p_y \ p_z]^T \in \mathbb{R}^3$ is the position vector of an arbitrary point located on the

rotation axis ϵ and expressed in the same reference frame. The matrix exponential mapping $e^{\Psi\theta}$ is based on the following formulation (Park and Okamura, 1994).

$$e^{\Psi\theta} = \left[\begin{array}{c|c} e^{E\theta} & A(-E\mathbf{p} + (l/\theta)\epsilon) \\ \hline [0] & 1 \end{array}\right] \tag{10}$$

where
$$e^{E\theta} = [I] + E \sin\theta + E^2(1 - \cos\theta) \tag{11}$$

and
$$A = [I]\theta + E(1 - \cos\theta) + E^2(\theta - \sin\theta) \tag{12}$$

For a pure rigid body translation ($\theta = 0$) along some axis v ($\|v\| = 1$), the twist is defined as

$$\Psi_{\text{trans}} = \left[\begin{array}{c|c} [[0]] & lv \\ \hline [0] & 0 \end{array}\right] \tag{13}$$

where [[0]] and [0] are 3×3 and 1×3 null matrices, respectively. In this case the matrix exponential mapping becomes

$$e^{\Psi_{\text{trans}}l} = \left[\begin{array}{c|c} [I] & lv \\ \hline [0] & 1 \end{array}\right] \tag{14}$$

with $[I]$ as the 3×3 identity matrix.

Based on the above formulation of rigid body motion, for an open-chain manipulator, the homogeneous transformation of each link *i with respect to the base frame* is obtained by an exponential mapping

$$H_{0i}(q_i) = e^{\Psi_i q_i} H_{0i}(0) \tag{15}$$

where

q_i = joint variable;
Ψ_i = twist of link i.

The homogeneous representation of the end-effector with respect to the base frame is obtained by combining a sequence of mappings into the so-called *product of exponentials* (PE) formula (Brockett, 1983).

$$H_{0n}(q_1, q_2, \ldots, q_n) = e^{\Psi_1 q_1} e^{\Psi_2 q_2} \cdots e^{\Psi_n q_n} H_{0n}(0, 0, \ldots, 0). \tag{16}$$

The matrix $H_{0n}(0, 0, \ldots, 0)$ represents the homogeneous transformation of the end-effector frame with respect to the base frame when the manipulator is in its *reference configuration,* i.e., all joint variables are zero. The twist Ψ_i corresponds to the screw motion of the i^{th} link as a result of moving joint i with all other joint variables held fixed at $q_j = 0$ ($j \neq i$).

One of the features of the PE formula is that, in contrast to the DH representation, there is no need to attach a frame to each link: once the base, end-effector, and a reference configuration frame have been chosen, a unique set of link twists is obtained that describes the forward kinematics of the robot. This property and the geometric representation make the PE formula a superior alternative to the DH convention.

2.2 Inverse Kinematics

Inverse kinematics is used to find the values of the joint variables that will place the end-effector at a desired location, i.e., desired position and orientation relative to the base, given the manipulator geometry (link lengths, offsets, twist angles, and the location of the base). Formally, for an n-d.o.f. manipulator, given the homogeneous matrix H_{0n} (5), the values of $q_1, q_2, \ldots, q_n$ are calculated. In general, the matrix equation (4) corresponds to 12 scalar equations; because the rotation matrix R_{0n} is orthonormal, only 6 of the 12 equations are independent. Therefore, the problem of inverse kinematics of a general 6-d.o.f. manipulator corresponds to solving a set of six nonlinear, transcendental equations with six unknowns (joint variables). There may be no solution, a unique solution, or multiple solutions to the inverse kinematics problem.

2.2.1 Solvability and Number of Solutions

A general approach for systematically solving the inverse kinematics problem (Lee and Liang, 1988) is to consider the set of nonlinear equations as a set of multivariate polynomials in $s_i \equiv \sin \theta_i$ and $c_i \equiv \cos \theta_i$ for $i = 1, 2, \ldots, n$. This is possible since the entries of each homogeneous transformation matrix (3) are unary (i.e., of degree one or less) in s_i and c_i. Then, by elimination of variables in a systematic way (Salmon, 1964), $n - 1$ variables are eliminated in a system of n polynomials in n variables, and a single polynomial in one variable is obtained. This method is called *dyalitic elimination,* and the resultant polynomial is called the *characteristic polynomial.* Once the roots of this polynomial are found, the eliminated variables can be determined from a set of linear equations. This general algorithm is presented in the next subsection, which addresses the existence and number of solutions of the inverse kinematics problem.

Generally, at least six joints are required to attain arbitrary three-dimensional task positions and orientations. The necessary condition for the existence of a solution is that the desired end-effector location lie in the reachable workspace. If the desired location is inside the workspace, then the existence of at least one solution is guaranteed. The existence of an analytical, closed-form solution to the inverse kinematics problem depends on the order of the characteristic polynomial. If the characteristic polynomial is of order 4 or less, since the roots can be obtained as algebraic functions of the polynomial coefficients, the corresponding inverse kinematics problem can be solved analytically. Otherwise, iterative numerical methods must be relied upon to obtain the roots of the polynomial. In this case, the problem is considered numerically solvable when: 1) an upper bound on the number of solutions exists. 2) an efficient algorithm for computing all solutions is available. Based on recent results in kinematics (Selfridge, 1989), all 6-d.o.f. open-chain manipulators with revolute and prismatic joints are solvable. The number of solutions depends on the number of prismatic joints and kinematic parameters. For the general case of six revolute joints (6R manipulator) or one prismatic and five revolute joints (5R1P manipulator), there are at most 16 different configurations for each end-effector location. For 4R2P manipulators the number of possible configurations drops to 8, and for the 3R3P the number is 2. These numbers are independent of the physical order of revolute and prismatic joints in the chain. In all of the above cases, the number of *real* roots of the characteristic polynomials (and hence the number of real configurations) may be less than the numbers cited above by any multiple of 2. Certain values of the kinematic parameters may also reduce the number of possible configurations. A detailed investigation can be found in Mavroidis and Roth (1994). As an example, a 6R manipulator with three consecutive joint axes intersecting in a common point (Pieper and Roth, 1969) or with three parallel joint axes (Duffy, 1980) has at most 8 configurations, and the characteristic polynomial is of order 4 with repeated roots; therefore analytical solutions exist. A 6R manipulator with a spherical wrist is very common in industry. The analytical technique for this case is first to solve for the first three joint variables to satisfy the desired wrist point location and then to find the last three joint variables to achieve the required hand orientation (Pieper and Roth, 1969).

2.2.2 A General Solution for Six-Degree-of-Freedom Manipulators

A systematic method of solving the inverse kinematics of 6-d.o.f. manipulators is to arrange the set of nonlinear equations as a set of multivariate polynomials in s_i and c_i and then eliminate all variables except θ_3, thus obtaining a polynomial of order 16 in $\tan(\theta_3/2)$ such that the joint angle θ_3 can be computed as its roots. The remaining joint variables are obtained by substituting and solving for some intermediate equations. In this section, the procedure is presented for general 6R manipulators. The extension to manipulators with prismatic joints is also discussed. The following algorithm is a summary of the algorithm presented in Raghavan and Roth (1993).

Step 1

Determine the DH parameters and homogeneous transformation matrices $H_{i-1,i}$ and then rewrite the forward kinematics matrix equation in the following form:

$$H_{23}H_{34}H_{45} = H_{12}^{-1}H_{01}^{-1}H_{06}H_{56}^{-1} \tag{17}$$

Step 2

Equate each of the first three elements of the third and fourth columns of both sides of Equation (17). This gives two sets of three scalar equations, from which all the other

equations are formed. These sets are written as two three-dimensional vector equations, denoted P (corresponding to the third column) and Q (corresponding to the fourth column):

$$P \equiv \begin{bmatrix} P_{1l} = P_{1r} \\ P_{2l} = P_{2r} \\ P_{3l} = P_{3r} \end{bmatrix}; \qquad Q \equiv \begin{bmatrix} Q_{1l} = Q_{1r} \\ Q_{2l} = Q_{2r} \\ Q_{3l} = Q_{3r} \end{bmatrix} \tag{18}$$

where P_{il} and Q_{il} refer to the left-hand side and P_{ir} and Q_{ir} refer to the right-hand side of the equations.

The set of all six equations can be written in the following matrix form:

$$A\mathbf{X}_1 = B\mathbf{Y} \tag{19}$$

For a 6R manipulator, A is a 6×9 matrix whose elements are linear combinations of s_3, c_3, B is a 6×8 matrix with constant elements, and $\mathbf{X}_1$ and $\mathbf{Y}$ are 9×1 and 8×1 matrices, respectively, defined as

$$\mathbf{X}_1 = [s_4s_5 \quad s_4c_5 \quad c_4s_5 \quad c_4c_5 \quad s_4 \quad c_4 \quad s_5 \quad c_5 \quad 1]^T \tag{20}$$

$$\mathbf{Y} = [s_1s_2 \quad s_1c_2 \quad c_1s_2 \quad c_1c_2 \quad s_1 \quad c_1 \quad s_2 \quad c_2]^T \tag{21}$$

Step 3

Construct the following scalar and vector equations to obtain eight new scalar equations:

$$Q \cdot Q \equiv [Q_{1l}^2 + Q_{2l}^2 + Q_{3l}^2 = Q_{1r}^2 + Q_{2r}^2 + Q_{3r}^2] \tag{22}$$

$$P \cdot Q \equiv [P_{1l}Q_{1l} + P_{2l}Q_{2l} + P_{3l}Q_{3l} = P_{1r}Q_{1r} + P_{2r}Q_{2r} + P_{3r}Q_{3r}] \tag{23}$$

$$P \times Q \equiv \begin{bmatrix} P_{2l}Q_{3l} - P_{3l}Q_{2l} = P_{2r}Q_{3r} - P_{3r}Q_{2r} \\ P_{3l}Q_{1l} - P_{1l}Q_{3l} = P_{3r}Q_{1r} - P_{1r}Q_{3r} \\ P_{1l}Q_{2l} - P_{2l}Q_{1l} = P_{1r}Q_{2r} - P_{2r}Q_{1r} \end{bmatrix} \tag{24}$$

$$P(Q \cdot Q) - 2Q(P \cdot Q) \equiv \begin{bmatrix} P_{1l}\sum_{i=1}^{3} Q_{il}^2 - 2Q_{1l}\sum_{i=1}^{3} P_{il}Q_{il} = P_{1r}\sum_{i=1}^{3} Q_{ir}^2 - 2Q_{1r}\sum_{i=1}^{3} P_{ir}Q_{ir} \\ P_{2l}\sum_{i=1}^{3} Q_{il}^2 - 2Q_{2l}\sum_{i=1}^{3} P_{il}Q_{il} = P_{2r}\sum_{i=1}^{3} Q_{ir}^2 - 2Q_{2r}\sum_{i=1}^{3} P_{ir}Q_{ir} \\ P_{3l}\sum_{i=1}^{3} Q_{il}^2 - 2Q_{3l}\sum_{i=1}^{3} P_{il}Q_{il} = P_{3r}\sum_{i=1}^{3} Q_{ir}^2 - 2Q_{3r}\sum_{i=1}^{3} P_{ir}Q_{ir} \end{bmatrix} \tag{25}$$

These eight equations have the same functional form as P and Q. Therefore, combining all the equations generates a set of 14 nonlinear equations of the form

$$\overline{A}\mathbf{X}_1 = \overline{B}\mathbf{Y} \tag{26}$$

where $\overline{A}$ is a 14×9 matrix whose elements are linear combinations of s_3, c_3, and $\overline{B}$ is a 14×8 constant matrix.

Step 4

Use any 8 of the 14 equations in (26) to solve for $\mathbf{Y}$ in terms of $\mathbf{X_1}$. The resulting system of 6 equations takes the form

$$\Gamma_1\mathbf{X}_1 = 0 \tag{27}$$

where Γ_1 is a 6×9 matrix. As a result, joint variables θ_1 and θ_2 are eliminated from the set of equations.

Step 5

Change Equation (27) into polynomial form by the following substitutions:

$$s_i = \frac{2x_i}{1 + x_i^2}; \qquad c_i = \frac{1 - x_i^2}{1 + x_i^2} \qquad \text{for} \quad i = 3, 4, 5 \tag{28}$$

where $x_i = \tan(\theta_i/2)$. Then multiply each equation by $(1 + x_4^2)$ and $(1 + x_5^2)$ to clear the denominators, and multiply the first four equations by $(1 + x_3^2)$ to obtain the following form:

$$\Gamma_2 \mathbf{X}_2 = 0 \tag{29}$$

where Γ_2 is a 6×9 matrix whose entries are linear combinations of s_3, c_3. For a general 6R manipulator, the vector $\mathbf{X_2}$ is

$$\mathbf{X}_2 = [x_4^2 x_5^2 \quad x_4^2 x_5 \quad x_4^2 \quad x_4 x_5^2 \quad x_4 x_5 \quad x_5^2 \quad x_5 \quad 1]^T \tag{30}$$

Step 6

Multiply the 6 equations in Equation (29) by x_4 to obtain 6 more equations. The set of all 12 equations forms the following homogeneous system:

$$\Gamma \mathbf{X} = \left[\begin{array}{c|c} \Gamma_2 & [0] \\ \hline [0] & \Gamma_2 \end{array}\right] \mathbf{X} = 0 \tag{31}$$

where [0] is the 6×9 null matrix and $\mathbf{X}$ is the vector of power products, which for a 6R manipulator is obtained as follows:

$$\mathbf{X} = [x_4^3 x_5^2 \quad x_4^3 x_5 \quad x_4^3 \quad x_4^2 x_5^2 \quad x_4^2 x_5 \quad x_4^2 \quad x_4 x_5^2 \quad x_4 x_5 \quad x_4 \quad x_5^2 \quad x_5 \quad 1]^T \tag{32}$$

Step 7

Apply the condition of having nontrivial solutions to the homogeneous system in order to obtain the characteristic equation in x_3, i.e.,

$$\det(\Gamma) = 0 \tag{33}$$

which is a polynomial of order 16 in the case of a general 6R manipulator.

Step 8

Obtain the roots of the characteristic polynomial by numerical methods. The real roots correspond to the real configurations of the inverse kinematics problem. For each value of x_3 thus obtained, the corresponding joint variable θ_3 may be computed from the formula $\theta_3 = 2 \tan^{-1}(x_3)$.

Step 9

Substitute each real value of x_3 into the coefficient matrix of Equation (31), and then solve for X to obtain unique values for x_4 and x_5, and hence θ_4 and θ_5, using $\theta_i = 2 \tan^{-1}(x_i)$.

Step 10

Substitute the values of θ_3, θ_4, and θ_5 into Equation (26) and use a subset of 8 equations to solve for $\mathbf{Y}$; then use the numerical values of s_1, c_1 and s_2, c_2 to obtain θ_1 and θ_2, respectively.

Step 11

Substitute values of θ_1, θ_2, θ_3, θ_4, and θ_5 into the first and second entries of the first column of the following kinematics relationship:

$$H_{56} = H_{45}^{-1} H_{34}^{-1} H_{23}^{-1} H_{12}^{-1} H_{01}^{-1} H_{06} \tag{34}$$

to obtain two linear equations in s_6 and c_6 from which a unique value for θ_6 is obtained.

For 5R1P manipulators, the above algorithm remains unchanged. However for a prismatic joint k, $\sin \theta_k$ is replaced by d_k, and $\cos \theta_k$ is replaced by $(d_k)^2$. For 4R2P manipulators, there are fewer power products and therefore fewer equations are required, leading to a characteristic polynomial of order 8. However, the procedure is essentially the same as above (Kohli and Osvatic, 1992). For 3R3P manipulators, the procedure simplifies considerably, leading to a characteristic polynomial of order 2 (Kohli and Osvatic, 1992).

2.2.3 Repeatability, Accuracy, and Computational Considerations

Industrial robots are rated on the basis of their ability to return to the same location when repetitive motion is required. The locations attained for any number of repeated motions

may not be identical. The *repeatability* of the manipulator is the expected maximum error at any attained location with respect to an *average* attained location of repeated motions. Manipulators with low joint backlash, friction, and flexibility usually have high repeatability.

Another robot rating criterion is the precision with which a desired location can be attained. This is called the *accuracy* of the manipulator, and it is the error between the desired and attained locations. The accuracy can be enhanced if two major obstacles are overcome. First, due to manufacturing errors in the machining and assembly of manipulators, the physical kinematics parameters may differ from the design parameters, which can produce significant errors between the actual and predicted locations of the end-effector. To solve this problem, calibration techniques are devised to improve accuracy through estimation of the individual kinematics parameters (Hollerbach, 1989). The second difficulty is that the numerical algorithms for solving the inverse kinematics problem are not efficient for practical implementations. For instance, for a general 6R manipulator the algorithm illustrated in Section 2.2.2 takes on average 10 seconds of CPU on an IBM 370-3090 using double precision arithmetic (Wampler and Morgan, 1991), while a speed on the order of milliseconds would be required in real-time applications. Furthermore, the problem of computing the roots of a characteristic polynomial of degree 16 can be ill-conditioned (Wilkinson, 1959). Closed-form solutions are quite efficient, but they exist only for a few special manipulators. This is one reason most industrial manipulators are limited to simple configurations and inverse kinematics calculations are not performed on-line. Recently some efficient algorithms have been suggested for the general solution of the inverse kinematics problem for 6-d.o.f. manipulators (Manocha and Canny, 1994). These algorithms require that the following operations be performed.

Off-line Symbolic Formulation and Numeric Substitution

For any class of manipulators, symbolic preprocessing can be performed to obtain the entries of matrices $\overline{A}$, $\overline{B}$, and Γ in Equations (26) and (31) as functions of kinematic parameters and elements of the end-effector homogeneous matrix H_{06}. The symbolic derivation and simplification can be performed using MATHEMATICA or MAPLE. Then, the numerical values of the kinematic parameters of a particular manipulator can be substituted in advance into the functions representing the elements of matrices $\overline{A}$, $\overline{B}$, and Γ. Given the desired location of the end-effector, the remaining numerical substitutions are performed on-line.

Changing the Problem of Finding the Roots of the Characteristic Polynomial to an Eigenvalue Problem

A more efficient method of finding nontrivial solutions for the matrix Equation (31) is to set up an equivalent eigenvalue problem (Ghasvini, 1993), rather than expanding the determinant of the coefficient matrix. The matrix Γ in Equation (31) can be expressed as (Ghasvini, 1993)

$$\Gamma = Lx_3^2 + Nx_3 + K \tag{35}$$

where L, N, and K are 12×12 matrices. For a nonsingular matrix L, the following 24×24 matrix is constructed:

$$\Pi = \left[\begin{array}{c|c} [0] & [I] \\ \hline -L^{-1}K & -L^{-1}N \end{array}\right] \tag{36}$$

where [0] and [I] are 12×12 null and identity matrices, respectively. The eigenvalues of Π are equal to the roots of Equation (33). Moreover, the eigenvector of Π corresponding to the eigenvalue x_3 has the structure

$$\mathbf{V} = \begin{bmatrix} \mathbf{X} \\ x_3\mathbf{X} \end{bmatrix} \tag{37}$$

which can be used to compute the solutions of Equation (27). If the matrix L is singular, then the following two matrices are constructed:

$$\Pi_1 = \begin{bmatrix} [I] & [0] \\ [0] & L \end{bmatrix}; \qquad \Pi_2 = \begin{bmatrix} [0] & [I] \\ -K & -N \end{bmatrix} \tag{38}$$

The roots of Equation (33) are equal to the eigenvalues of the generalized eigenvalue problem ($\Pi_1 - x_3\Pi_2$), which has the same eigenvalues as Equation (36).

2.3 Velocity Kinematics

2.3.1 Link Velocity, Jacobian

In addition to the kinematic relationship between joint displacements and the end-effector location, the relationship between the joint velocity vector and end-effector linear and angular velocities is also useful. The absolute velocities (i.e., relative to the base coordinate frame) of the link i coordinate frame are computed from the absolute velocities of the link $i - 1$ frame as follows:

If the i^{th} joint is revolute, i.e., $\dot{q}_i = \dot{\theta}_i$,

$$\begin{aligned} \boldsymbol{v}_i &= \boldsymbol{v}_{i-1} + \omega_i \times \mathbf{p}_{i-1,i} \\ \boldsymbol{\omega}_i &= \boldsymbol{\omega}_{i-1} + \mathbf{z}_{i-1}\dot{q}_i \end{aligned} \tag{39}$$

and if the i^{th} joint is prismatic, i.e., $\dot{q}_i = \dot{d}_i$,

$$\begin{aligned} \boldsymbol{v}_i &= \boldsymbol{v}_{i-1} + \omega_i \times \mathbf{p}_{i-1,i} + \mathbf{z}_{i-1}\dot{q}_i \\ \omega_i &= \omega_{i-1} \end{aligned} \tag{40}$$

where ω_i is the angular velocity vector of frame F_i (attached to link i), $\boldsymbol{v}_i$ is the linear velocity vector of its origin, and $\mathbf{z}_i$ is the unit vector along the axis of joint i.

Combining Equations (39) and (40), the relationship between the link frame and joint velocities is obtained as follows (Whitney, 1969):

$$\mathbf{V}_i = \begin{bmatrix} \boldsymbol{v}_i \\ \boldsymbol{\omega}_i \end{bmatrix} = J_i(q_1, q_2, \ldots, q_{i-1})\dot{\mathbf{q}} \tag{41}$$

The $n \times 1$ matrix $\dot{\mathbf{q}} = [\dot{q}_1\ \dot{q}_2 \cdots \dot{q}_n]^T$ consists of joint velocities, and J_i is a $6 \times n$ "Jacobian" of link i defined as follows:

$$J_i(q_1, q_2, \ldots, q_{i-1}) = \begin{bmatrix} \mathbf{t}_0\ \mathbf{t}_1 \cdots \mathbf{t}_{i-1} & \underbrace{[0]\ [0] \cdots [0]}_{(n\ -\ i)\ \text{columns}} \end{bmatrix} \tag{42}$$

where [0] is the 6×1 null matrix and other columns of the Jacobian are of the form

$$\begin{cases} \mathbf{t}_j = \begin{bmatrix} \mathbf{z}_j \times \mathbf{p}_{j,i} \\ \mathbf{z}_j \end{bmatrix} & \text{for joint } j \text{ revolute} \\ \mathbf{t}_j = \begin{bmatrix} \mathbf{z}_j \\ [0] \end{bmatrix} & \text{for joint } j \text{ prismatic} \end{cases} ; \qquad j = 0, 1, 2, \ldots, i - 1 \tag{43}$$

with [0] as the 3×1 null matrix.

The robot Jacobian J_n (or simply the Jacobian J) is obtained from Equations (42) and (43) with i equal to n (the total number of joints):

$$J(\mathbf{q}) = [\mathbf{t}_0\ \mathbf{t}_1 \cdots \mathbf{t}_{n-2}\ \mathbf{t}_{n-1}] \tag{44}$$

and therefore the end-effector linear and angular velocities are obtained from the linear mapping

$$\mathbf{V} = J(\mathbf{q})\dot{\mathbf{q}} \tag{45}$$

Numerical computation of the Jacobian depends on the frame in which the vectors in Equation (43) are expressed. If all vectors are taken in the base frame, the resulting Jacobian matrix denoted by J^0 is represented in the base frame, and so are the absolute velocities of the end-effector from Equation (45). A recursive formulation of J^0 is readily obtained from the recursive formulation of the forward kinematics by adding the following step to the backward loop of Algorithm 1:

4. CALCULATE: $\mathbf{t}_{i-1} =$

$$\begin{cases} \begin{bmatrix} a^y_{i-1,n}p^z_{i-1,n} - a^z_{i-1,n}p^y_{i-1,n} \\ a^z_{i-1,n}p^x_{i-1,n} - a^x_{i-1,n}p^z_{i-1,n} \\ a^x_{i-1,n}p^y_{i-1,n} - a^y_{i-1,n}p^x_{i-1,n} \\ a^x_{i-1,n} \\ a^y_{i-1,n} \\ a^z_{i-1,n} \end{bmatrix} & \text{if joint } i-1 \text{ is revolute} \\ \begin{bmatrix} a^x_{i-1,n} \\ a^y_{i-1,n} \\ a^z_{i-1,n} \\ 0 \\ 0 \\ 0 \end{bmatrix} & \text{if joint } i-1 \text{ is prismatic} \end{cases} \tag{46}$$

The absolute velocities of the end-effector can also be expressed in its own frame, with the Jacobian being computed in this frame (denoted by J^n). The Jacobian matrix J^n can be directly obtained from the following formulation:

$$J^n = \left[\begin{array}{c|c} R^T_{0n} & [0] \\ \hline [0] & R^T_{0n} \end{array}\right] J^0 \tag{47}$$

As in the forward kinematics problem, a symbolic formulation is recommended for real-time tasks.

2.3.2 Link Acceleration

Link accelerations are computed the same way as link velocities. The absolute linear and angular accelerations (i.e., relative to the base coordinate frame) of the link i coordinate frame are obtained from the absolute accelerations of the link $i-1$ frame as follows:

for joint i revolute, i.e., $\dot{q}_i = \dot{\theta}_i$ and $\ddot{q}_i = \ddot{\theta}_i$,

$$\begin{aligned} \dot{\boldsymbol{\nu}}_i &= \dot{\boldsymbol{\nu}}_{i-1} + \dot{\boldsymbol{\omega}}_i \times \mathbf{p}_{i-1,i} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \mathbf{p}_{i-1,i}) \\ \dot{\boldsymbol{\omega}}_i &= \dot{\boldsymbol{\omega}}_{i-1} + \mathbf{z}_{i-1}\ddot{q}_i + \boldsymbol{\omega}_i \times (\mathbf{z}_{i-1}\dot{q}_i) \end{aligned} \tag{48}$$

and for joint i prismatic, i.e., $\dot{q}_i = \dot{d}_i$ and $\ddot{q}_i = \ddot{d}_i$,

$$\begin{aligned} \dot{\boldsymbol{\nu}}_i &= \dot{\boldsymbol{\nu}}_{i-1} + \dot{\boldsymbol{\omega}}_i \times \mathbf{p}_{i-1,i} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \mathbf{p}_{i-1,i}) + \mathbf{z}_{i-1}\ddot{q}_i + 2\boldsymbol{\omega}_i \times (\mathbf{z}_{i-1}\dot{q}_i) \\ \dot{\boldsymbol{\omega}}_i &= \dot{\boldsymbol{\omega}}_{i-1} \end{aligned} \tag{49}$$

The above equations can be written in compact form as

$$\dot{\mathbf{V}}_i = \begin{bmatrix} \dot{\boldsymbol{\nu}}_i \\ \boldsymbol{\omega}_i \end{bmatrix} = J_i(q_1, q_2, \ldots, q_{i-1})\ddot{\mathbf{q}} + \dot{J}_i(q_1, \ldots, q_{i-1}, \dot{q}_1, \ldots, \dot{q}_{i-1})\dot{\mathbf{q}} \tag{50}$$

2.3.3 Singularity

Generally, the link frames velocities corresponding to a particular set of end-effector linear and angular velocities can be obtained from the inverse of the linear mapping (45) as

$$\dot{\mathbf{q}} = J^{-1}(\mathbf{q})\mathbf{V} \tag{51}$$

The linear mapping (51) exists only for configurations at which the inverse of the Jacobian matrix exists, i.e., *nonsingular* configurations. In a *singular configuration,* the end-effector cannot move in certain direction(s); thus the manipulator loses one or more degrees of freedom. In a singular configuration, the Jacobian rank decreases, i.e., two or more columns of J become linearly dependent; thus the determinant of the Jacobian becomes zero. This is a computational test for the existence of singular configurations. Singular configurations should usually be avoided because most of the manipulators are designed for tasks in which all degrees of freedom are required. Furthermore, near sin-

gular configurations the joint velocities required to maintain the desired end-effector velocity in certain directions may become extremely large. The most common singular configurations for 6-d.o.f. manipulators are listed below (Murray, Li, and Sastry, 1994):

1. *Two collinear revolute joint axes* occur in spherical wrist assemblies that have three mutually perpendicular axes intersecting at one point. Rotating the second joint may align the first and third joints, and then the Jacobian will have two linearly dependent columns. Mechanical restrictions are usually imposed on the wrist design to prevent the wrist axes from generating a singularity of the wrist.
2. *Three parallel coplanar revolute joint axes* occur, for instance, in an elbow manipulator (Murray, Li, and Sastry, 1994, p. 90) that consists of a 3-d.o.f. manipulator with a spherical wrist when it is fully extended or fully retracted.
3. *Four revolute joint axes intersecting in one point.*
4. *Four coplanar revolute joints.*
5. *Six revolute joints intersecting along a line.*
6. *A prismatic joint axis perpendicular to two parallel coplanar revolute joints.*

In addition to the Jacobian singularities, the motion of the manipulator is restricted if the joint variables are constrained to a certain interval. In this case, a reduction in the number of degrees of freedom may occur when one or more joints reach the limit of their allowed motion.

2.3.4 Redundant Manipulators and Multiarm Robots

A *kinematically redundant* manipulator is one that has more than the minimum number of degrees of freedom required to attain a desired location. In this case, an infinite number of configurations can be obtained for a desired end-effector location. Multi-arm robots are a special class of redundant manipulators. When two or more robot arms are used to perform a certain task cooperatively, an increased load-carrying capacity and manipulation capability are achieved. For a general redundant manipulator with n degrees of freedom ($n > 6$), the Jacobian is not square, and there are only $(n - 6)$ arbitrary variables in the general solution of mapping (51), assuming that the Jacobian is full rank. Additional constraints are needed to limit the solution to a unique one. Therefore, redundancy provides the opportunity for choice or decision. It is typically used to optimize some secondary criteria while achieving the primary goal of following a specified end-effector trajectory. The secondary criteria considered so far in the literature are robot singularity and obstacle avoidance, minimization of joint velocity and torque, increasing the system precision by an optimal distribution of arm compliance, and improving the load-carrying capacity by optimizing the transmission ratio between the input torque and output forces. Hayward (1988) gives a review of the different criteria. As an example, one common approach is to choose the minimum (in the least squares sense) joint velocities that provide the desired end-effector motion. This is achieved by choosing (Hollerbach, 1984)

$$\dot{\mathbf{q}} = J^*(\mathbf{q})\mathbf{V} \tag{52}$$

where

$$J^*(\mathbf{q}) = J^T(JJ^T)^{-1} \tag{53}$$

is the *pseudo-inverse* of the Jacobian matrix.

2.4 Parallel Manipulators

Parallel manipulators are mechanisms that contain two or more serial chains connecting the end-effector to the base. Generally, parallel manipulators can offer more accuracy in positioning and orienting objects than open-chain manipulators. They can also possess a high payload/weight ratio, and they are easily adaptable to position and force control. On the other hand, the workspace of parallel manipulators is usually smaller than that of serial manipulators. A typical configuration of parallel manipulators is the so-called *in-parallel* manipulators, in which each serial chain has the same structure and one joint is actuated in each chain (Duffy, 1996). A popular example of this type is the Stewart platform, shown in Figure 3. The mechanism consists of a rigid plate connected to the base by a set of prismatic joints, each of which is connected to the plate and the base by spherical joints, allowing complete rotational motion. Only the prismatic joints are actuated.

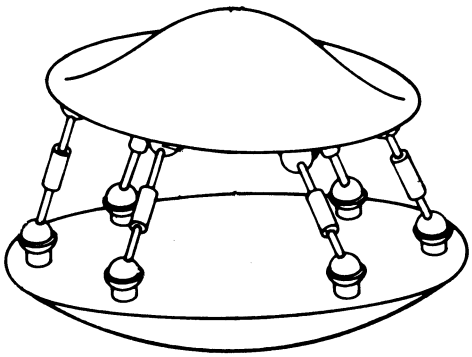

Figure 3 A general form of the Stewart platform.

The forward kinematics of a parallel manipulator can be expressed by equating the end-effector location to each individual chain. Consider a parallel manipulator with m chains so that each chain contains n_i ($i = 1, 2, \ldots, m$) joints. The forward kinematics model can then be described as

$$H_{0e} = H_{01}H_{12}^{1} \cdots H_{n_1-1,n_1}^{1} = H_{01}^{2}H_{12}^{2} \cdots H_{n_2-1,n_2}^{2} = \cdots = H_{01}^{m}H_{12}^{m} \cdots H_{n_m-1,n_m}^{m} \quad (54)$$

where H_{ij}^{k} is the homogeneous transformation matrix between joints i and j in the chain k. All quantities are specified with respect to a unique base and end-effector coordinate frame. Equation (54), called the *structure equation,* introduces constraints between the joint displacements of the manipulator. As a result, unlike for serial manipulators, the joint space for a parallel manipulator is not the Cartesian product of the individual joint spaces but a subset of it that satisfies Equation (54). In a parallel manipulator, if N and L are the number of joints and links, respectively, and l_i is the number of degrees of freedom of the i^{th} joint, then the number of degrees of freedom of the manipulator can be obtained by taking the total number of degrees of freedom for all links and subtracting the number of constraints imposed by the joints attached to the links. For the specific case where all joints apply independent constraints, the number of degrees of freedom F can be calculated as

$$F = 6L - \sum_{i=1}^{N} (6 - l_i) = 6(L - N) + \sum_{i=1}^{N} l_i \quad (55)$$

The inverse kinematics problem is no more difficult for a parallel manipulator than for the open-chain case, as each chain can be analyzed separately.

The end-effector velocity of a parallel manipulator can be obtained from each chain equivalently:

$$\mathbf{V} = J_1\dot{\mathbf{q}}_1 = J_2\dot{\mathbf{q}}_2 = \cdots = J_m\dot{\mathbf{q}}_m \quad (56)$$

Obviously not all joint velocities can be specified independently. The relationship between joint torques and end-effector forces is more complex in a parallel manipulator than in a serial manipulator, as there are internal interactions between forces produced by the different chains in a parallel manipulator. The reader is referred to more detailed sources, such as Duffy (1996), and open-chain manipulators will be considered in the sequel.

3 STATICS

This section discusses the relationship between the forces and moments that act on the manipulator when it is at rest. In open-chain manipulators, each joint is usually driven by an individual actuator. The corresponding input joint torque (for revolute joints) or force (for prismatic joints) is transmitted through the manipulator arm linkages to the end-effector, where the resultant force and moment balance an external load. The relationship between the actuator drive torques (or forces) and the end-effector resultant force and moment is determined using the manipulator Jacobian (Asada and Slotine, 1986):

$$\tau = (J^e)^T \begin{bmatrix} \mathbf{f}_e \\ \mathbf{g}_e \end{bmatrix} = (J^e)^T \mathbf{G}_e \tag{57}$$

where $\tau \in \mathbb{R}^n$ is the vector of joint torques (and forces) and $\mathbf{f}_e \in \mathbb{R}^3$ and $\mathbf{g}_e \in \mathbb{R}^3$ are vectors of the resulting reaction force and moment, respectively, of the external loads acting on the end-effector and expressed in the end-effector frame:

$$\mathbf{f}_e = -\mathbf{f}_{ext}; \qquad \mathbf{g}_e = -\mathbf{g}_{ext} \tag{58}$$

The *generalized force* $\mathbf{G}_e \in \mathbb{R}^6$, which consists of the force/moment pair, is called the *wrench* vector.

3.1 Duality Between the Velocity Kinematics and Statics

The statics (57) is closely related to the velocity kinematics (45) because the manipulator Jacobian is being used for both mappings. In a specific configuration, the kinematics and statics linear mappings can be represented by the diagram shown in Figure 4 (Asada and Slotine, 1986). For the velocity kinematics, the Jacobian is a linear mapping from the n-dimensional vector space $\mathbb{R}^n$ to the six-dimensional space $\mathbb{R}^6$. Note that n (number of joints) can be more than six in the case of redundant manipulators. The range subspace $R(J)$ represents all possible end-effector velocities that can be generated by the n joint velocities in the present configuration. If $n > 6$, there exists a null space $N(J)$ of the Jacobian mapping that corresponds to all joint velocity vectors $\dot{\mathbf{q}}$ that produce no net velocity at the end-effector. In a singular configuration, J is not full rank, and the subspace $R(J)$ does not cover the entire vector space $\mathbb{R}^6$, i.e., there exists at least one direction in which the end-effector can not be moved.

The statics relationship is also a linear mapping from $\mathbb{R}^6$ to $\mathbb{R}^n$ provided by the transpose of the Jacobian. The range subspace $R(J^T)$ and null subspace $N(J^T)$ can be identified from the Jacobian mapping. The null subspace $N(J^T)$ corresponds to the end-effector wrenches that can be balanced without input torques or forces at the joints, as the load is borne entirely by the structure of the arm linkages. For a redundant manipulator, the range subspace $R(J^T)$ does not cover the entire space $\mathbb{R}^n$, and there are some sets of input joint torques or forces that cannot be balanced by any end-effector wrench. These configurations correspond to the null space of the kinematics mapping $N(J)$ that contains the joint velocity vectors that produce no end-effector motion.

The velocity kinematics and statics are dual concepts that can be stated as follows:

1. In a singular configuration, there exists at least one direction in which the end-effector can not be moved. In this direction, the end-effector wrench is entirely balanced by the manipulator structure and does not require any input joint torque or force.

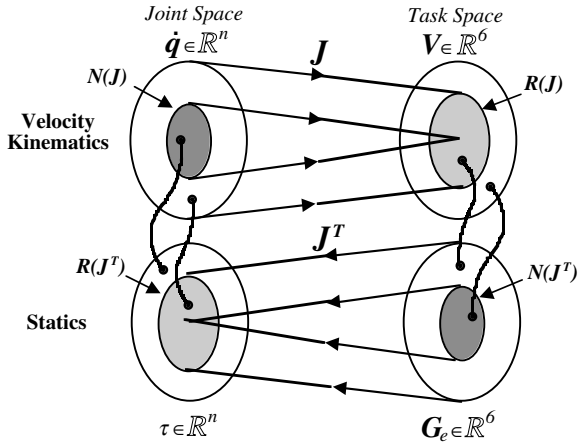

Figure 4 The duality relation between statics and velocity kinematics linear mappings (from H. Asada and J. J. E. Slotine, *Robot Analysis and Control.* New York: John Wiley & Sons, Inc., 1986).

2. In each configuration of a redundant manipulator, there is at least one direction in which joint velocities produce no end-effector velocity. In this direction, the joint torques and forces cannot be balanced by any end-effector wrench. Therefore, in order to maintain a stationary arm configuration, no input joint torque or force that generates end-effector wrench should be applied.
3. For a general manipulator, in each configuration the directions of possible motion of the end-effector also define the directions in which wrenches that are applied to the end-effector can be entirely balanced by the joint torques and forces.

4 DYNAMICS

The dynamics model describes the balance between internal and external loads applied to the manipulator. The input joint torques and forces generated by the actuators balance the other external forces and moments and the internal loads of the manipulator. The dynamics model is fundamental in mechanical and control system design and simulation of motion and also for real-time control (model-based control). The dynamics of a manipulator with n degrees of freedom can be expressed using the following equation of motion of each link i (Spong and Vidyasagar, 1989):

$$\tau_i + [J^T \mathbf{G}_{ext}]_i - N_i(\mathbf{q}, \dot{\mathbf{q}}) = \sum_{j=1}^{n} M_{ij}(\mathbf{q})\ddot{q}_j + \sum_{k=1}^{n}\sum_{j=1}^{n} h_{ijk}(\mathbf{q})\dot{q}_j\dot{q}_k; \qquad i = 1, 2, \ldots, n \tag{59}$$

The left-hand side of Equation (59) consists of all external forces and moments applied to the i^{th} link, which are decomposed into three parts. The first part τ_i is the input torque (or force) applied by the actuator of joint i. The second part $[J^T\mathbf{G}_{ext}]_i$ is the mapping of the external wrench $\mathbf{G}_{ext}$ to the joint space, in particular the effect of the external wrench on link i. The third part $-N_i(\mathbf{q}, \dot{\mathbf{q}})$ represents any other external force and moment that act on the i^{th} link, including gravity torque, friction, etc.

The right-hand side of (59) contains the reaction (internal) loads of the manipulator. The first term represents the inertial load on the i^{th} link, while the second term accounts for the Coriolis effect (for $j \neq k$) and centrifugal effect (for $j = k$). The entries $[M_{ij}]$ of the manipulator inertia matrix are configuration-dependent, and the off-diagonal elements ($i \neq j$) generate the coupling inertial loads. Coriolis and centrifugal terms $h_{ijk}(\mathbf{q})$ also depend on configuration and introduce further interaction effects in the manipulator dynamics. The components of $h_{ijk}(\mathbf{q})$ are defined as follows (Spong and Vidyasagar, 1989):

$$h_{ijk} = \frac{\partial M_{ij}}{\partial q_k} - \frac{1}{2}\frac{\partial M_{jk}}{\partial q_i}. \tag{60}$$

Equation (59) can be rewritten in compact form as

$$\tau + J^T \mathbf{G}_{ext} = M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + N(\mathbf{q}, \dot{\mathbf{q}}) \tag{61}$$

where the elements of matrix C are defined as

$$C_{ij}(\mathbf{q}, \dot{\mathbf{q}}) = \sum_{k=1}^{n} h_{ijk}\dot{q}_k \qquad i, j = 1, 2, \ldots, n \tag{62}$$

The matrices M and C express the inertial characteristics of the manipulator and have the following important properties (Spong and Vidyasagar, 1989):

1. $M(\mathbf{q})$ is symmetric and bounded positive definite.
2. $(\dot{M} - 2C) \in \mathbb{R}^{n\times n}$ is a skew-symmetric matrix.

These properties are useful in reducing the number of operations in the calculation of the dynamics model, and they are also used in proofs of stability of many control laws for robot manipulators.

The manipulator dynamics model can be used in two modes:

1. To simulate the motion of the manipulator (joint displacements, velocities, and accelerations) for given input joint torques and forces; this is referred to as the *forward dynamics* problem

2. To calculate the joint torques and forces required to attain specific joint displacements, with given velocities and accelerations; this is called the *inverse dynamics* problem.

The computational algorithms of the above dynamics problems and related considerations are discussed in the sequel.

4.1 Inverse Dynamics

4.1.1 The Recursive Algorithm and Computational Considerations

Given the joint displacements, velocities, and accelerations, the joint torques can be directly computed from Equation (61). However, since at each trajectory point the configuration-dependent coefficients M, C, and N must be computed, the amount of computation required is extremely high, and it increases very rapidly as the number of degrees of freedom n increases. Calculating Equation (61) requires $(32n^4 + 86n^3 + 171n^2 + 53n - 128)$ multiplications and $(25n^4 + 66n^3 + 129n^2 + 42n - 96)$ additions, or more than 100,000 arithmetic operations for a 6-d.o.f. manipulator. This heavy computational load is a bottleneck for the use of the inverse dynamics model in real-time control, especially since the calculation would have to be repeated at the rate of 60 Hz or higher.

If the dynamics equations are formulated in a *recursive* form, the computational complexity can be significantly reduced from $O(n^4)$ to $O(n)$ so that the required number of operations would vary linearly with the number of degrees of freedom. Most of the fast algorithms for the inverse dynamics problem are based on the Newton–Euler approach, and consist of two main steps. In the first step, the angular velocity and acceleration of each link of the manipulator and the linear velocity and acceleration of its center of mass are calculated starting with the first link and continuing through the last. In the second step, the force and moment exerted on each link are calculated backward starting from the last link. Furthermore, if the dynamic equations are expressed in a tensor form and tensor properties are used, the computational complexity of the inverse dynamics algorithm can be further reduced. Algorithm 2 presents an efficient inverse dynamics formulation that requires $(104n - 77)$ multiplications and $(92n - 70)$ additions for computing the joint torques or forces for each trajectory point. The basic formulation of this algorithm was originally developed in Balafoutis, Patel, and Misra (1988). Although this algorithm is not the fastest available formulation, it is simple and effective for real-time implementations. See He and Goldenberg (1990) for more details on computational considerations.

While the current recursive formulations are quite efficient, symbolic closed-form equations derived for a particular manipulator are likely to be the most efficient formulation (Burdick, 1986). If symbolic operations are used simplified formulations can easily be obtained for real-time applications.

4.2 Forward Dynamics

In forward dynamics, joint displacements and velocities are obtained as a result of applied joint torques (forces) and other external forces and moments. From Equation (61), given the current joint positions and velocities and the current external wrenches, the vector of joint accelerations can be calculated. From this calculation the corresponding joint velocities and positions are computed by subsequent integrations. In this way, the problem of forward dynamics is divided into two phases:

Phase 1 Obtain joint accelerations from Equation (61).

Phase 2 Integrate the joint accelerations and obtain the new joint velocities; then integrate again and obtain the new joint displacements.

In Phase 2 any suitable numerical integration method can be used. In Phase 1 the following form of Equation (61) is used:

$$M(\mathbf{q})\ddot{\mathbf{q}} = [C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + N(\mathbf{q}, \dot{\mathbf{q}}) - J^T \mathbf{G}_{ext}] - \tau \tag{63}$$

Phase 1 can be performed by completing the following three steps:

Step 1.1. Calculate the elements of $[C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + N(\mathbf{q}, \dot{\mathbf{q}}) - J^T\mathbf{G}_{ext}]$

Step 1.2. Calculate the elements of the inertia matrix $M(\mathbf{q})$.

Step 1: Forward Kinematics

OBTAIN link rotation matrices $R_{i-1,i}$ for $i = 1, 2, \ldots, n$

SET $R_{n,n+1} = [I]_{3\times3}$

Step 2: Initialization

SET $\begin{cases} \boldsymbol{\omega}_0 = \dot{\boldsymbol{\omega}}_0 = \mathbf{v}_0 = [0 \quad 0 \quad 0]^T; \quad \dot{\mathbf{v}}_0 = -\mathbf{g}; \quad \tilde{\omega}_0 = [0]_{3\times3} \\ \mathbf{f}_{n+1} = -\mathbf{f}_{ext}; \quad \mathbf{g}_{n+1} = -\mathbf{g}_{ext}; \quad m_i; \quad I_{ci}; \quad \mathbf{p}_{ci} \end{cases}$

Step 3: Forward Loop

FOR $i = 1$ to n

CALCULATE: $\begin{cases} \begin{cases} \boldsymbol{\omega}_i = R_{i-1,i}^T(\boldsymbol{\omega}_{i-1} + [0 \quad 0 \quad \dot{q}_i]^T) \\ \dot{\boldsymbol{\omega}}_i = R_{i-1,i}^T(\dot{\boldsymbol{\omega}}_{i-1} + \tilde{\omega}_{i-1}[0 \quad 0 \quad \dot{q}_i]^T + [0 \quad 0 \quad \ddot{q}_i]^T) \end{cases} & \text{if joint } i \text{ is revolute} \\ \begin{cases} \boldsymbol{\omega}_i = R_{i-1,i}^T\boldsymbol{\omega}_{i-1} \\ \dot{\boldsymbol{\omega}}_i = R_{i-1,i}^T\dot{\boldsymbol{\omega}}_{i-1} \end{cases} & \text{if joint } i \text{ is prismatic} \end{cases}$

SET: $\tilde{\omega}_i = \begin{bmatrix} 0 & -\omega_i^z & \omega_i^y \\ \omega_i^z & 0 & -\omega_i^x \\ -\omega_i^y & \omega_i^x & 0 \end{bmatrix}; \quad \tilde{\dot{\omega}}_i = \begin{bmatrix} 0 & -\dot{\omega}_i^z & \dot{\omega}_i^y \\ \dot{\omega}_i^z & 0 & -\dot{\omega}_i^x \\ -\dot{\omega}_i^y & \dot{\omega}_i^x & 0 \end{bmatrix}$

CALCULATE: $\begin{cases} \boldsymbol{\omega}_i = \tilde{\boldsymbol{\omega}}_i + \tilde{\boldsymbol{\omega}}_i\tilde{\boldsymbol{\omega}}_i \\ \begin{cases} \dot{\mathbf{v}}_i = R_{i-1,i}^T(\dot{\mathbf{v}}_{i-1} + \Omega_i\mathbf{p}_{i-1,i}) & \text{if joint } i \text{ is revolute} \\ \dot{\mathbf{v}}_i = R_{i-1,i}^T(\dot{\mathbf{v}}_{i-1} + \Omega_i\mathbf{p}_{i-1,i} + 2\tilde{\omega}_i[0 \quad 0 \quad \dot{q}_i]^T + [0 \quad 0 \quad \ddot{q}_i]^T) & \text{if joint } i \text{ is prismatic} \end{cases} \\ \dot{\mathbf{v}}_{ci} = \dot{\mathbf{v}}_i + \boldsymbol{\omega}_i\mathbf{p}_{ci} \\ \tilde{g}_{ci} = -(\Omega_i I_{ci}) + (\Omega_i I_{ci})^T \\ \mathbf{f}_{ci} = m_i\dot{\mathbf{v}}_{ci} \end{cases}$

SET: $\mathbf{g}_{ci} = [-\tilde{g}_{ci}(1, 2) \quad \tilde{g}_{ci}(1, 3) \quad -\tilde{g}_{ci}(2, 3)]^T$

NEXT i

Step 4: Backward Loop

FOR $i = n$ to 1

SET: $\tilde{P}_{i-1,i} = \begin{bmatrix} 0 & -p_{i-1,i}^z & p_{i-1,i}^y \\ p_{i-1,i}^z & 0 & -p_{i-1,i}^x \\ -p_{i-1,i}^y & p_{i-1,i}^x & 0 \end{bmatrix}; \quad \tilde{P}_{ci} = \begin{bmatrix} 0 & -p_{ci}^z - p_{i-1,i}^z & p_{ci}^y + p_{i-1,i}^y \\ p_{ci}^z + p_{i-1,i}^z & 0 & -p_{ci}^x - p_{i-1,i}^x \\ -p_{ci}^y - p_{i-1,i}^y & p_{ci}^x + p_{i-1,i}^x & 0 \end{bmatrix}$

CALCULATE: $\begin{cases} \mathbf{f}_i = R_{i,i+1}\mathbf{f}_{i+1} + \mathbf{f}_{ci} \\ \mathbf{g}_i = R_{i,i+1}\mathbf{g}_{i+1} + \mathbf{g}_{ci} + \tilde{P}_{ci}\mathbf{f}_{ci} + \tilde{P}_{i+1,i}\mathbf{f}_{i+1} \end{cases}$

CALCULATE: $\begin{cases} \tau_i = [0 \quad 0 \quad 1]R_{i-1,i}\mathbf{g}_i & \text{if joint } i \text{ is revolute} \\ \tau_i = [0 \quad 0 \quad 1]R_{i-1,i}\mathbf{f}_i & \text{if joint } i \text{ is prismatic} \end{cases}$

NEXT i

Algorithm 2 Tensor recursive formulation of the inverse dynamics problem.

Step 1.3. Solve the set of simultaneous linear equations (63) for $\ddot{\mathbf{q}}$.

Step 1.1 can be directly computed using an inverse dynamics algorithm with the joint accelerations set to zero. If the inverse dynamics algorithm is represented (Walker, 1985) as the function

$$\tau = INVDYN(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \dot{\mathbf{v}}_0, \mathbf{G}_{ext}) \tag{64}$$

then Step 1.1 can be performed as

$$[C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + N(\mathbf{q}, \dot{\mathbf{q}}) - J^T\mathbf{G}_{ext}] = INVDYN(\mathbf{q}, \dot{\mathbf{q}}, 0, \dot{\mathbf{v}}_0, \mathbf{G}_{ext}) \tag{65}$$

Using the inverse dynamics algorithm, the complexity of this step is O(n).

The computation of the matrix M can also be performed using the inverse dynamics formulation. First $\dot{\mathbf{q}}$, $\dot{\mathbf{v}}_0$ and $\mathbf{G}_{ext}$ are set to be zero. Then the i^{th} column of M, denoted by $[M]_i$, can be obtained by setting the joint acceleration in the inverse dynamics algorithm to be $\boldsymbol{\delta}_i$ which is a vector with all elements equal to zero except for the i^{th} element being equal to one. Thus

$$[M]_i = INVDYN(\mathbf{q}, 0, \boldsymbol{\delta}_i, 0, 0) \qquad i = 1, 2, \ldots, n \tag{66}$$

By repeating this procedure for n columns of M, the entire inertia matrix is obtained. This step requires n applications of the inverse dynamics with the order of complexity $O(n)$.

M and the right-hand side of Equation (63) having been computed, the final step is to solve the set of simultaneous linear equations for $\ddot{\mathbf{q}}$. The order of complexity of this step is $O(n^3)$, thus making the order of complexity of the overall forward dynamics algorithm $O(n^3)$. The symmetry and positive definiteness of M can be exploited in Step 1.2 to calculate only the lower triangle of the matrix and in Step 1.3 to use specialized factorization techniques in order to improve the computational efficiency of the algorithm (Featherstone, 1987). New and efficient algorithms have recently been suggested (Lilly, 1993).

5 CONCLUSIONS

This chapter presents basic kinematics, statics, and dynamics modeling algorithms. These questions have been addressed extensively in the literature, but the approach taken here attempts to extract only the fundamentals needed by a reader wishing to obtain a complete set of tools to generate models of robot manipulators. The tools suggested could be computerized without great effort, and they could be useful in the design of new manipulators or workcells, the analysis of existing manipulators, simulations, and real-time control. The procedures and algorithms suggested here are considered the most computationally effective for fulfilling the basic tasks of modeling.

Some of the issues discussed are not commonly encountered in the literature. For example, the use of screw theory and products of exponentials in kinematics modeling, and even symbolic formulations in kinematics and dynamics modeling, have limited applicability, but can be very effective.

Further study is needed to investigate the numerical accuracy and robustness of inverse kinematics and forward dynamics calculations. In particular, there are strong indications that model-based control is more effective; thus incorporation of kinematics and dynamics models into real-time control is recommended. The computational overhead resulting from this methodology requires extensive work to minimize the number of arithmetic operations and to exploit parallel processing. In addition, the accuracy of on-line calculations must be addressed. In a different direction, special-configuration manipulators not describable as standard open-loop kinematic chains must also be investigated and specific models generated for them.

REFERENCES

Asada, H., and J. J. E. Slotine. 1986. *Robot Analysis and Control.* New York: John Wiley & Sons.

Balafoutis, C. A., R. V. Patel, and P. Misra. 1988. ''Efficient Modeling and Computation of Manipulator Dynamics Using Orthogonal Cartesian Tensors.'' *IEEE Journal of Robotics and Automation* **4**(6), 665–676.

Brockett, R. W. 1983. ''Robotic Manipulators and Product of Exponentials Formula.'' In *Proceedings of International Symposium of Mathematical Theory of Networks and Systems,* Beer Sheba, Israel.

Burdick, J. W. 1986. ''An Algorithm for Generation of Efficient Manipulator Dynamic Equations.'' In *Proceedings of the 1986 IEEE International Conference on Robotics and Automation,* San Francisco. 212–218.

Ceccarelli, M. 1994. ''Determination of the Workspace Boundary of a General n-Revolute Manipulator.'' In *Advances in Robot Kinematics and Computational Geometry.* Ed. A. J. Lenarcic and B. B. Ravani. Dordrecht: Kluwer Academic. 39–48.

Denavit, J., and R. S. Hartenberg. 1965. ''A Kinematic Notation for Low-Pair Mechanisms Based on Matrices.'' *ASME Journal of Applied Mechanics* (June), 215–221.

Duffy, J. 1980. *Analysis of Mechanisms and Manipulators.* New York: John Wiley & Sons.

———. 1996. *Statics and Kinematics with Applications to Robotics.* Cambridge: Cambridge University Press.

Featherstone, R. 1987. *Robot Dynamics Algorithms.* Dordrecht: Kluwer Academic.

Ghasvini, M. 1993. ''Reducing the Inverse Kinematics of Manipulators to the Solution of a Generalized Eigenproblem.'' In *Computational Kinematics.* Ed. J. Angeles, G. Hommel, and P. Kovacs. Dordrecht: Kluwer Academic. 15–26.

Hayward, V. 1988. ''An Analysis of Redundant Manipulators from Several Viewpoints.'' In *Proceedings of the NATO Advanced Research Workshop on Robots with Redundancy: Design Sensing and Control,* Salo, Italy, June 27–July 1.

He, X., and A. A. Goldenberg. 1990. ''An Algorithm for Efficient Manipulator Dynamic Equations.'' *Journal of Robotic Systems* **7,** 689–702.

Ho, C. Y., and J. Sriwattanathamma. 1989. ''Symbolically Automated Direct Kinematics Equations Solver for Robotic Manipulators.'' *Robotica* **7,** 243–254.

Hollerbach, J. M. 1984. ''Optimal Kinematic Design for a Seven Degree of Freedom Manipulator.'' Paper presented at International Symposium of Robotics Research, Kyoto.

———. 1989. ''A Survey of Kinematic Calibration.'' In *The Robotics Review I.* Ed. I. Khatib, J. J. Craig, and T. Lozano-Pérez. Cambridge: MIT Press.

Kohli, D., and M. Osvatic. 1992. ''Inverse Kinetics of General 4R2P, 3R3P, 4R1c, 4R2C, and 3C Manipulators.'' *Proceedings of 22nd ASME Mechanisms Conference,* Scottsdale, DE. Vol. 45. 129–137

Lee, H. Y., and C. G. Liang. 1988. ''Displacement Analysis of the General Spatial 7-link 7R Mechanism.'' *Mechanism and Machine Theory* **23,** 219–226.

Lilly, K. W. 1993. *Efficient Dynamic Simulation of Robotic Mechanisms.* Dordrecht: Kluwer Academic.

Manocha, D., and J. F. Canny. 1994. ''Efficient Inverse Kinematics for General 6R Manipulators.'' *IEEE Transactions on Robotics and Automation* **10**(5), 648–657.

Mavroidis, C., and B. Roth. 1994. ''Structural Parameters Which Reduce the Number of Manipulator Configurations.'' *Journal of Mechanical Design* **116,** 3–10.

Murray, R. M., Z. Li, and S. S. Sastry. 1994. *A Mathematical Introduction to Robotic Manipulation.* London: CRC Press. Chapter 3.

Park, F. C., and K. Okamura. 1994. ''Kinematic Calibration and the Product of Exponentials Formula.'' In *Advances in Robot Kinematics and Computational Geometry.* Ed. A. J. Lenarcic and B. B. Ravani. Dordrecht: Kluwer Academic. 119–128.

Pieper, D. L., and B. Roth. 1969. ''The Kinematics of Manipulators under Computer Control.'' In *Proceedings of 2nd International Congress for the Theory of Machines and Mechanisms,* Zakopane, Poland. Vol. 2. 159–168.

Raghavan, M., and B. Roth. 1993. ''Inverse Kinematics of the General 6R Manipulator and Related Linkages.'' *Transactions of the ASME* **115,** 502–508.

Ruoff, C. 1981. ''Fast Trigonometric Functions for Robot Control.'' *Robotics Age* (November).

Salmon, G. 1964. *Lessons Introductory to the Modern Higher Algebra.* New York: Chelsea.

Selfridge, R. G. 1989. ''Analysis of 6-Link Revolute Arms.'' *Mechanism and Machine Theory* **24**(1), 1–8.

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

Turner, T., J. Craig, and W. Gruver. 1984. ''A Microprocessor Architecture for Advanced Robot Control.'' Paper presented at 14th ISIR, Stockholm, October.

Vukobratovic, M., and M. Kircanski. 1986. *Kinematics and Trajectory Synthesis of Manipulation Robot.* New York: Springer-Verlag. 53–77.

Walker, M. W. 1985. ''Kinematics and Dynamics.'' In *Handbook of Industrial Robots.* 1st ed. Ed. S. Y. Nof. New York: John Wiley & Sons. Chapter 6.

Wampler, C., and A. P. Morgan. 1991. ''Solving the 6R Inverse Position Problem Using a Generic-Case Solution Methodology.'' *Mechanisms and Machine Theory* **26**(1), 91–106.

Westmacott, G. D. 1993. ''The Application of Symbolic Equation Manipulation to High Performance Multibody Simulation.'' In *Robotics: Applied Mathematics and Computational Aspects.* Ed. Kevin Warwick. Oxford: Oxford University Press. 527–540.

Whitney, D. E. 1969. ''Resolved Motion Rate Control of Manipulators and Human Prosthesis.'' *IEEE Transactions on Man–Machine Systems* **MMS-10**(2), 47–53.

Wilkinson, J. H. 1959. ''The Evaluation of the Zeros of Ill-Conditioned Polynomials, Parts I and II.'' *Numerical Methods* **1,** 150–180.

Yuan, M. S. C., and F. Freudenstein. 1971. ''Kinematic Analysis of Spatial Mechanisms by Means of Screw Coordinates, Part 1—Screw Coordinates.'' *Transactions of the ASME* **93,** 61–66.