1

INTRODUCTION

Robotics is a relatively young field of modern technology that crosses traditional engineering boundaries. Understanding the complexity of robots and their applications requires knowledge of electrical engineering, mechanical engineering, systems and industrial engineering, computer science, economics, and mathematics. New disciplines of engineering, such as manufacturing engineering, applications engineering, and knowledge engineering have emerged to deal with the complexity of the field of robotics and factory automation.

This book is concerned with fundamentals of robotics, including **kinematics**, **dynamics**, **motion planning**, **computer vision**, and **control**. Our goal is to provide a complete introduction to the most important concepts in these subjects as applied to industrial robot manipulators, mobile robots, and other mechanical systems. A complete treatment of the discipline of robotics would require several volumes. Nevertheless, at the present time, the majority of robot applications deal with industrial robot arms operating in structured factory environments so that a first introduction to the subject of robotics must include a rigorous treatment of the topics in this text.

The term **robot** was first introduced into our vocabulary by the Czech playwright Karel Capek in his 1920 play *Rossum's Universal Robots*, the word *robota* being the Czech word for work. Since then the term has been applied to a great variety of mechanical devices, such as teleoperators, underwater vehicles, autonomous land rovers, etc. Virtually anything that operates with some degree of autonomy, usually under computer control, has at some point been called a robot. In this text the term robot will mean a computer controlled industrial manipulator of the type shown in Figure 1.1. This type

Fig. 1.1 The ABB IRB6600 Robot. Photo courtesy of ABB.

of robot is essentially a mechanical arm operating under computer control. Such devices, though far from the robots of science fiction, are nevertheless extremely complex electro-mechanical systems whose analytical description requires advanced methods, presenting many challenging and interesting research problems.

An official definition of such a robot comes from the **Robot Institute of America** (RIA): *A robot is a reprogrammable multifunctional manipulator designed to move material, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks.*

The key element in the above definition is the reprogrammability of robots. It is the computer brain that gives the robot its utility and adaptability. The so-called robotics revolution is, in fact, part of the larger computer revolution.

Even this restricted version of a robot has several features that make it attractive in an industrial environment. Among the advantages often cited in favor of the introduction of robots are decreased labor costs, increased precision and productivity, increased flexibility compared with specialized machines, and more humane working conditions as dull, repetitive, or hazardous jobs are performed by robots.

The robot, as we have defined it, was born out of the marriage of two earlier technologies: **teleoperators** and **numerically controlled milling machines**. Teleoperators, or master-slave devices, were developed during the second world war to handle radioactive materials. Computer numerical control (CNC) was developed because of the high precision required in the machining of certain items, such as components of high performance aircraft. The first robots essentially combined the mechanical linkages of the teleoperator with the autonomy and programmability of CNC machines.

The first successful applications of robot manipulators generally involved some sort of material transfer, such as injection molding or stamping, where the robot merely attends a press to unload and either transfer or stack the finished parts. These first robots could be programmed to execute a sequence of movements, such as moving to a location A, closing a gripper, moving to a location B, etc., but had no external sensor capability. More complex applications, such as welding, grinding, deburring, and assembly require not only more complex motion but also some form of external sensing such as vision, tactile, or force-sensing, due to the increased interaction of the robot with its environment.

It should be pointed out that the important applications of robots are by no means limited to those industrial jobs where the robot is directly replacing a human worker. There are many other applications of robotics in areas where the use of humans is impractical or undesirable. Among these are undersea and planetary exploration, satellite retrieval and repair, the defusing of explosive devices, and work in radioactive environments. Finally, prostheses, such as artificial limbs, are themselves robotic devices requiring methods of analysis and design similar to those of industrial manipulators.

1.1 MATHEMATICAL MODELING OF ROBOTS

While robots are themselves mechanical systems, in this text we will be primarily concerned with developing and manipulating mathematical models for robots. In particular, we will develop methods to represent basic geometric aspects of robotic manipulation, dynamic aspects of manipulation, and the various sensors available in modern robotic systems. Equipped with these mathematical models, we will be able to develop methods for planning and controlling robot motions to perform specified tasks. Here we describe some of the basic ideas that are common in developing mathematical models for robot manipulators.

1.1.1 Symbolic Representation of Robots

Robot Manipulators are composed of **links** connected by **joints** to form a **kinematic chain**. Joints are typically rotary (revolute) or linear (prismatic). A **revolute** joint is like a hinge and allows relative rotation between two links. A **prismatic** joint allows a linear relative motion between two links. We denote revolute joints by R and prismatic joints by P, and draw them as shown in Figure 1.2. For example, a three-link arm with three revolute joints is an RRR arm.

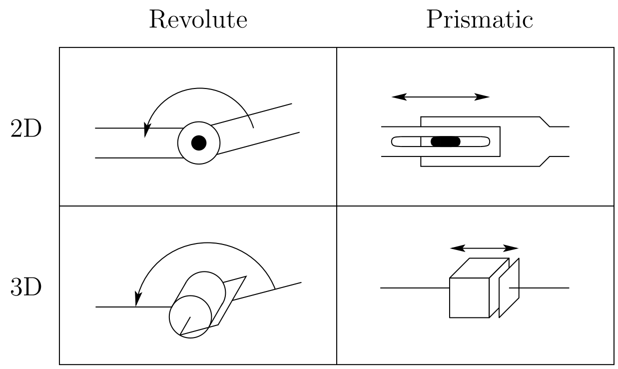

Fig. 1.2 Symbolic representation of robot joints.

Each joint represents the interconnection between two links. We denote the axis of rotation of a revolute joint, or the axis along which a prismatic joint translates by z_i if the joint is the interconnection of links i and $i+1$. The **joint variables**, denoted by θ for a revolute joint and d for the prismatic joint, represent the relative displacement between adjacent links. We will make this precise in Chapter 3.

1.1.2 The Configuration Space

A **configuration** of a manipulator is a complete specification of the location of every point on the manipulator. The set of all possible configurations is called the **configuration space**. In our case, if we know the values for the joint variables (i.e., the joint angle for revolute joints, or the joint offset for prismatic joints), then it is straightforward to infer the position of any point on the manipulator, since the individual links of the manipulator are assumed to be rigid, and the base of the manipulator is assumed to be fixed. Therefore, in this text, we will represent a configuration by a set of values for the joint variables. We will denote this vector of values by q, and say that the robot is in configuration q when the joint variables take on the values $q_1 \cdots q_n$, with $q_i = \theta_i$ for a revolute joint and $q_i = d_1$ for a prismatic joint.

An object is said to have n **degrees-of-freedom** (DOF) if its configuration can be minimally specified by n parameters. Thus, the number of DOF is equal to the dimension of the configuration space. For a robot manipulator, the number of joints determines the number DOF. A rigid object in three-dimensional space has six DOF: three for **positioning** and three for **orientation** (e.g., roll, pitch and yaw angles). Therefore, a manipulator should typically possess at least six independent DOF. With fewer than six DOF the arm cannot reach every point in its work environment with arbitrary orientation. Certain applications such as reaching around or behind obstacles may require more than six DOF. A manipulator having more than six links is referred to as a **kinematically redundant** manipulator. The difficulty of controlling a manipulator increases rapidly with the number of links.

1.1.3 The State Space

A configuration provides an instantaneous description of the geometry of a manipulator, but says nothing about its dynamic response. In contrast, the **state** of the manipulator is a set of variables that, together with a description of the manipulator's dynamics and input, are sufficient to determine any future state of the manipulator. The **state space** is the set of all possible states. In the case of a manipulator arm, the dynamics are Newtonian, and can be specified by generalizing the familiar equation $F = ma$. Thus, a state of the manipulator can be specified by giving the values for the joint variables q and for joint velocities $\dot{q}$ (acceleration is related to the derivative of joint velocities). We typically represent the state as a vector $x = (q, \dot{q})^T$. The dimension of the state space is thus $2n$ if the system has n DOF.

1.1.4 The Workspace

The **workspace** of a manipulator is the total volume swept out by the end-effector as the manipulator executes all possible motions. The workspace is constrained by the geometry of the manipulator as well as mechanical constraints on the joints. For example, a revolute joint may be limited to less than a full $360°$ of motion. The workspace is often broken down into a **reachable workspace** and a **dexterous workspace**. The reachable workspace is the entire set of points reachable by the manipulator, whereas the dexterous workspace consists of those points that the manipulator can reach with an arbitrary orientation of the end-effector. Obviously the dexterous workspace is a subset of the reachable workspace. The workspaces of several robots are shown later in this chapter.

1.2 ROBOTS AS MECHANICAL DEVICES

There are a number of physical aspects of robotic manipulators that we will not necessarily consider when developing our mathematical models. These include mechanical aspects (e.g., how are the joints actually implemented), accuracy and repeatability, and the tooling attached at the end effector. In this section, we briefly describe some of these.

1.2.1 Classification of Robotic Manipulators

Robot manipulators can be classified by several criteria, such as their **power source**, or way in which the joints are actuated, their **geometry**, or kinematic structure, their intended **application area**, or their **method of control**. Such classification is useful primarily in order to determine which robot is right for a given task. For example, a hydraulic robot would not be suitable for food handling or clean room applications. We explain this in more detail below.

Power Source. Typically, robots are either electrically, hydraulically, or pneumatically powered. Hydraulic actuators are unrivaled in their speed of response and torque producing capability. Therefore hydraulic robots are used primarily for lifting heavy loads. The drawbacks of hydraulic robots are that they tend to leak hydraulic fluid, require much more peripheral equipment (such as pumps, which require more maintenance), and they are noisy. Robots driven by DC- or AC-servo motors are increasingly popular since they are cheaper, cleaner and quieter. Pneumatic robots are inexpensive and simple but cannot be controlled precisely. As a result, pneumatic robots are limited in their range of applications and popularity.

Application Area. Robots are often classified by application into **assembly** and **non-assembly robots**. Assembly robots tend to be small, electrically driven and either revolute or SCARA (described below) in design. The main nonassembly application areas to date have been in welding, spray painting, material handling, and machine loading and unloading.

Method of Control. Robots are classified by control method into **servo** and **non-servo** robots. The earliest robots were non-servo robots. These robots are essentially open-loop devices whose movement is limited to predetermined mechanical stops, and they are useful primarily for materials transfer. In fact, according to the definition given previously, fixed stop robots hardly qualify as robots. Servo robots use closed-loop computer control to determine their motion and are thus capable of being truly multifunctional, reprogrammable devices.

Servo controlled robots are further classified according to the method that the controller uses to guide the end-effector. The simplest type of robot in this class is the **point-to-point** robot. A point-to-point robot can be taught a discrete set of points but there is no control on the path of the end-effector in between taught points. Such robots are usually taught a series of points with a teach pendant. The points are then stored and played back. Point-to-point robots are severely limited in their range of applications. In **continuous path** robots, on the other hand, the entire path of the end-effector can be controlled. For example, the robot end-effector can be taught to follow a straight line between two points or even to follow a contour such as a welding seam. In addition, the velocity and/or acceleration of the end-effector can often be controlled. These are the most advanced robots and require the most sophisticated computer controllers and software development.

Geometry. Most industrial manipulators at the present time have six or fewer degrees-of-freedom. These manipulators are usually classified kinematically on the basis of the first three joints of the arm, with the wrist being described separately. The majority of these manipulators fall into one of five geometric types: **articulated (RRR)**, **spherical (RRP)**, **SCARA (RRP)**, **cylindrical (RPP)**, or **Cartesian (PPP)**. We discuss each of these below.

Each of these five manipulator arms are **serial link** robots. A sixth distinct class of manipulators consists of the so-called **parallel robot**. In a parallel manipulator the links are arranged in a closed rather than open kinematic chain. Although we include a brief discussion of parallel robots in this chapter, their kinematics and dynamics are more difficult to derive than those of serial link robots and hence are usually treated only in more advanced texts.

1.2.2 Robotic Systems

A robot manipulator should be viewed as more than just a series of mechanical linkages. The mechanical arm is just one component in an overall **Robotic System**, illustrated in Figure 1.3, which consists of the **arm**, **external power**

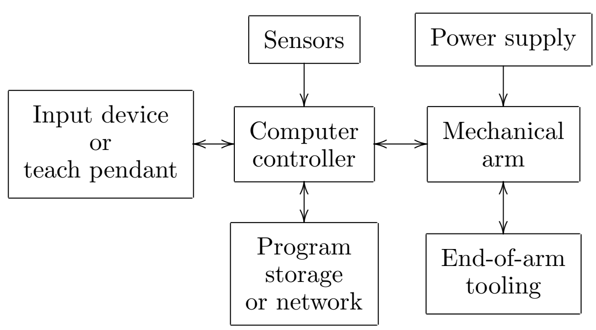

Fig. 1.3 Components of a robotic system.

source, **end-of-arm tooling**, **external and internal sensors**, **computer interface**, and **control computer**. Even the programmed software should be considered as an integral part of the overall system, since the manner in which the robot is programmed and controlled can have a major impact on its performance and subsequent range of applications.

1.2.3 Accuracy and Repeatability

The **accuracy** of a manipulator is a measure of how close the manipulator can come to a given point within its workspace. **Repeatability** is a measure of how close a manipulator can return to a previously taught point. The primary method of sensing positioning errors in most cases is with position encoders located at the joints, either on the shaft of the motor that actuates the joint or on the joint itself. There is typically no direct measurement of the end-effector position and orientation. One must rely on the assumed geometry of the manipulator and its rigidity to infer (i.e., to calculate) the end-effector position from the measured joint positions. Accuracy is affected therefore by computational errors, machining accuracy in the construction of the manipulator, flexibility effects such as the bending of the links under gravitational and other loads, gear backlash, and a host of other static and dynamic effects. It is primarily for this reason that robots are designed with extremely high rigidity. Without high rigidity, accuracy can only be improved by some sort of direct sensing of the end-effector position, such as with vision.

Once a point is taught to the manipulator, however, say with a teach pendant, the above effects are taken into account and the proper encoder values necessary to return to the given point are stored by the controlling computer. Repeatability therefore is affected primarily by the controller resolution. **Controller resolution** means the smallest increment of motion that the controller can sense. The resolution is computed as the total distance traveled by the tip divided by 2^n, where n is the number of bits of encoder accuracy. In this context, linear axes, that is, prismatic joints, typically have higher resolution than revolute joints, since the straight line distance traversed by the tip of a linear axis between two points is less than the corresponding arc length traced by the tip of a rotational link.

In addition, as we will see in later chapters, rotational axes usually result in a large amount of kinematic and dynamic coupling among the links with a resultant accumulation of errors and a more difficult control problem. One may wonder then what the advantages of revolute joints are in manipulator design. The answer lies primarily in the increased dexterity and compactness of revolute joint designs. For example, Figure 1.4 shows that for the same

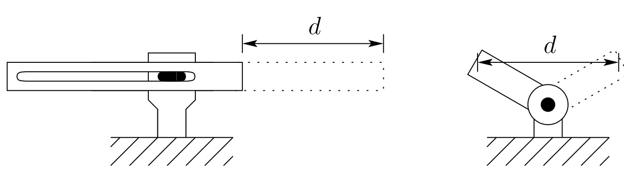

Fig. 1.4 Linear vs. rotational link motion.

range of motion, a rotational link can be made much smaller than a link with linear motion. Thus manipulators made from revolute joints occupy a smaller working volume than manipulators with linear axes. This increases the ability of the manipulator to work in the same space with other robots, machines, and people. At the same time revolute joint manipulators are better able to maneuver around obstacles and have a wider range of possible applications.

1.2.4 Wrists and End-Effectors

The joints in the kinematic chain between the arm and end effector are referred to as the **wrist**. The wrist joints are nearly always all revolute. It is increasingly common to design manipulators with **spherical wrists**, by which we mean wrists whose three joint axes intersect at a common point. The spherical wrist is represented symbolically in Figure 1.5.

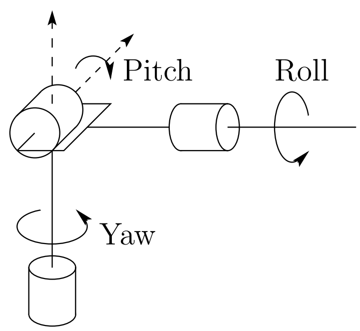

Fig. 1.5 Structure of a spherical wrist.

The spherical wrist greatly simplifies the kinematic analysis, effectively allowing one to decouple the positioning and orientation of the end effector. Typically therefore, the manipulator will possess three degrees-of-freedom for position, which are produced by three or more joints in the arm. The number of degrees-of-freedom for orientation will then depend on the degrees-of-freedom of the wrist. It is common to find wrists having one, two, or three degrees-of-freedom depending of the application. For example, the SCARA robot shown in Figure 1.14 has four degrees-of-freedom: three for the arm, and one for the wrist, which has only a rotation about the final z-axis.

It has been said that a robot is only as good as its **hand** or **end-effector**. The arm and wrist assemblies of a robot are used primarily for positioning the end-effector and any tool it may carry. It is the end-effector or tool that actually performs the work. The simplest type of end-effectors are grippers, which usually are capable of only two actions, **opening** and **closing**. While this is adequate for materials transfer, some parts handling, or gripping simple tools, it is not adequate for other tasks such as welding, assembly, grinding, etc. A great deal of research is therefore devoted to the design of special purpose end-effectors as well as to tools that can be rapidly changed as the task dictates. There is also much research on the development of anthropomorphic hands. Such hands have been developed both for prosthetic use and for use in manufacturing. Since we are concerned with the analysis and control of the manipulator itself and not in the particular application or end-effector, we will not discuss end-effector design or the study of grasping and manipulation.

1.3 COMMON KINEMATIC ARRANGEMENTS OF MANIPULATORS

Although there are many possible ways use prismatic and revolute joints to construct kinematic chains, in practice only a few of these are commonly used. Here we briefly describe several arrangements that are most typical.

1.3.1 Articulated manipulator (RRR)

The articulated manipulator is also called a **revolute**, or **anthropomorphic** manipulator. The ABB IRB1400 articulated arm is shown in Figure 1.6. A common revolute joint design is the **parallelogram linkage** such as the

Fig. 1.6 The ABB IRB1400 Robot. Photo courtesy of ABB.

Motoman SK16, shown in Figure 1.7. In both of these arrangements joint axis z_2 is parallel to z_1 and both z_1 and z_2 are perpendicular to z_0. This kind of manipulator is known as an elbow manipulator. The structure and terminology associated with the elbow manipulator are shown in Figure 1.8. Its workspace is shown in Figure 1.9.

The revolute manipulator provides for relatively large freedom of movement in a compact space. The parallelogram linkage, although typically less dexterous than the elbow manipulator manipulator, nevertheless has several advantages that make it an attractive and popular design. The most notable feature of the parallelogram linkage manipulator is that the actuator for joint 3 is located on link 1. Since the weight of the motor is born by link 1, links 2

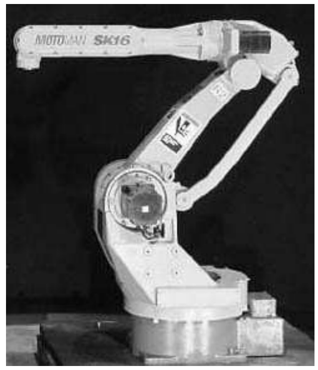

Fig. 1.7 The Motoman SK16 manipulator.

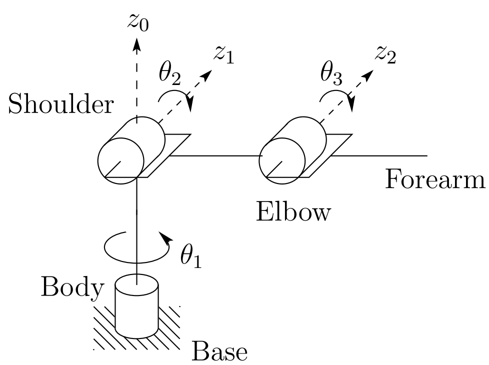

Fig. 1.8 Structure of the elbow manipulator.

and 3 can be made more lightweight and the motors themselves can be less powerful. Also the dynamics of the parallelogram manipulator are simpler than those of the elbow manipulator, thus making it easier to control.

1.3.2 Spherical Manipulator (RRP)

By replacing the third or elbow joint in the revolute manipulator by a prismatic joint one obtains the spherical manipulator shown in Figure 1.10. The term **spherical manipulator** derives from the fact that the spherical coordinates defining the position of the end-effector with respect to a frame whose origin lies at the intersection of the three z axes are the same as the first three joint variables. Figure 1.11 shows the Stanford Arm, one of the most well-known spherical robots. The workspace of a spherical manipulator is shown in Figure 1.12.

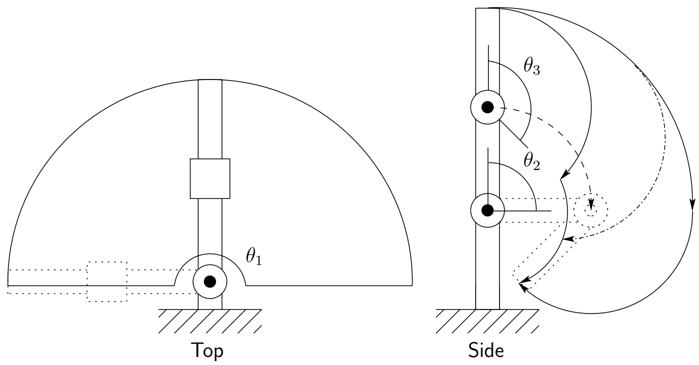

Fig. 1.9 Workspace of the elbow manipulator.

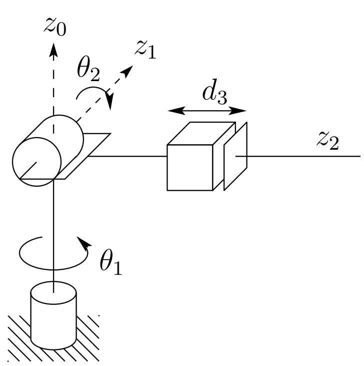

Fig. 1.10 The spherical manipulator.

1.3.3 SCARA Manipulator (RRP)

The **SCARA** arm (for **S**elective **C**ompliant **A**rticulated **R**obot for **A**ssembly) shown in Figure 1.13 is a popular manipulator, which, as its name suggests, is tailored for assembly operations. Although the SCARA has an RRP structure, it is quite different from the spherical manipulator in both appearance and in its range of applications. Unlike the spherical design, which has z_0 perpendicular to z_1, and z_1 perpendicular to z_2, the SCARA has z_0, z_1, and z_2 mutually parallel. Figure 1.14 shows the Epson E2L653S, a manipulator of this type. The SCARA manipulator workspace is shown in Figure 1.15.

1.3.4 Cylindrical Manipulator (RPP)

The cylindrical manipulator is shown in Figure 1.16. The first joint is rev-

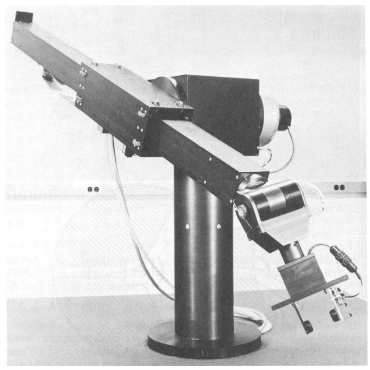

Fig. 1.11 The Stanford Arm. Photo courtesy of the Coordinated Science Lab, University of Illinois at Urbana-Champaign.

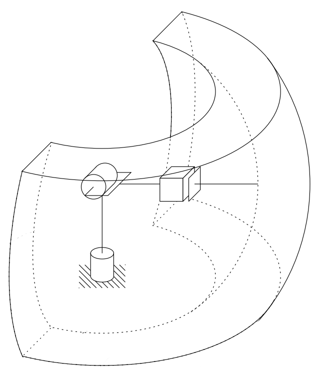

Fig. 1.12 Workspace of the spherical manipulator.

olute and produces a rotation about the base, while the second and third joints are prismatic. As the name suggests, the joint variables are the cylindrical coordinates of the end-effector with respect to the base. A cylindrical robot, the Seiko RT3300, is shown in Figure 1.17, with its workspace shown in Figure 1.18.

1.3.5 Cartesian manipulator (PPP)

A manipulator whose first three joints are prismatic is known as a Cartesian manipulator, shown in Figure 1.19.

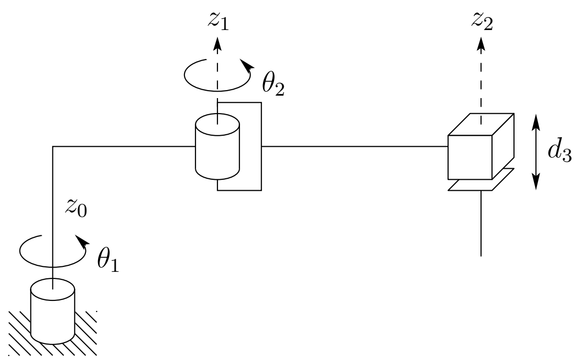

Fig. 1.13 The SCARA (Selective Compliant Articulated Robot for Assembly).

Fig. 1.14 The Epson E2L653S SCARA Robot. Photo Courtesy of Epson.

For the Cartesian manipulator the joint variables are the Cartesian coordinates of the end-effector with respect to the base. As might be expected the kinematic description of this manipulator is the simplest of all manipulators. Cartesian manipulators are useful for table-top assembly applications and, as gantry robots, for transfer of material or cargo. An example of a Cartesian robot, from Epson-Seiko, is shown in Figure 1.20. The workspace of a Cartesian manipulator is shown in Figure 1.21.

1.3.6 Parallel Manipulator

A **parallel manipulator** is one in which some subset of the links form a closed chain. More specifically, a parallel manipulator has two or more independent kinematic chains connecting the base to the end-effector. Figure 1.22 shows the ABB IRB 940 Tricept robot, which is a parallel manipulator. The closed

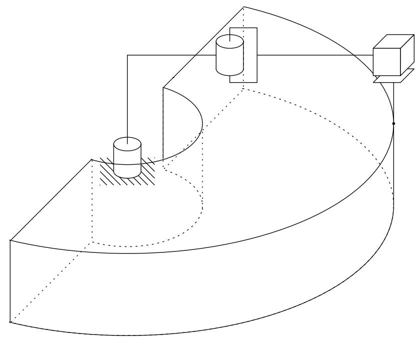

Fig. 1.15 Workspace of the SCARA manipulator.

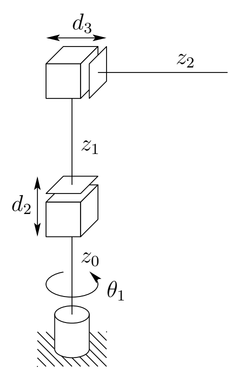

Fig. 1.16 The cylindrical manipulator.

chain kinematics of parallel robots can result in greater structural rigidity, and hence greater accuracy, than open chain robots. The kinematic description of parallel robots is fundamentally different from that of serial link robots and therefore requires different methods of analysis.

1.4 OUTLINE OF THE TEXT

A typical application involving an industrial manipulator is shown in Figure 1.23. The manipulator is shown with a grinding tool that it must use to remove a certain amount of metal from a surface. In the present text we are concerned with the following question: *What are the basic issues to be resolved and what must we learn in order to be able to program a robot to perform such tasks?*

Fig. 1.17 The Seiko RT3300 Robot. Photo courtesy of Seiko.

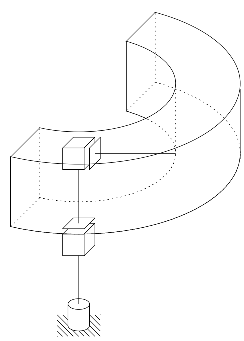

Fig. 1.18 Workspace of the cylindrical manipulator.

The ability to answer this question for a full six degree-of-freedom manipulator represents the goal of the present text. The answer is too complicated to be presented at this point. We can, however, use the simple two-link planar mechanism to illustrate some of the major issues involved and to preview the topics covered in this text.

Suppose we wish to move the manipulator from its **home** position to position A, from which point the robot is to follow the contour of the surface S to the point B, at constant velocity, while maintaining a prescribed force F normal to the surface. In so doing the robot will cut or grind the surface according to a predetermined specification. To accomplish this and even more general tasks, a we must solve a number of problems. Below we give examples of these problems, all of which will be treated in more detail in the remainder of the text.

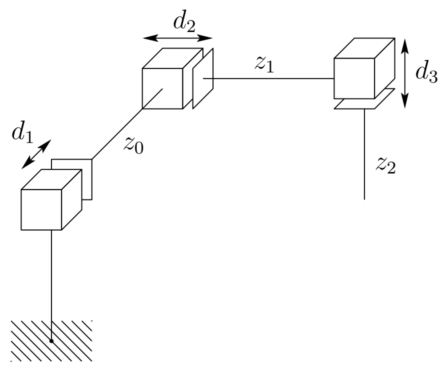

Fig. 1.19 The Cartesian manipulator.

Fig. 1.20 The Epson Cartesian Robot. Photo courtesy of Epson.

Forward Kinematics

The first problem encountered is to describe both the position of the tool and the locations A and B (and most likely the entire surface S) with respect to a common coordinate system. In Chapter 2 we give some background on representations of coordinate systems and transformations among various coordinate systems.

Typically, the manipulator will be able to sense its own position in some manner using internal sensors (position encoders located at joints 1 and 2) that can measure directly the joint angles θ_1 and θ_2. We also need therefore to express the positions A and B in terms of these joint angles. This leads to the **forward kinematics problem** studied in Chapter 3, which is to determine the position and orientation of the end-effector or tool in terms of the joint variables.

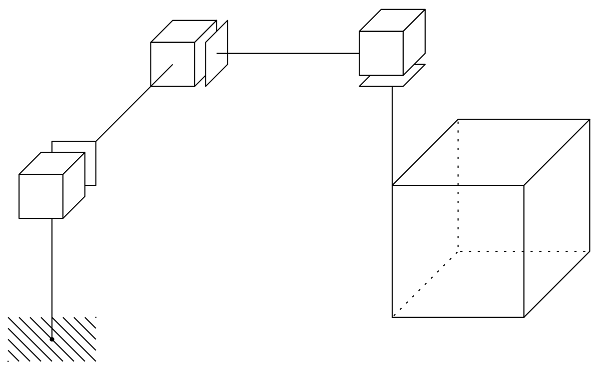

Fig. 1.21 Workspace of the Cartesian manipulator.

Fig. 1.22 The ABB IRB940 Tricept Parallel Robot. Photo courtesy of ABB.

It is customary to establish a fixed coordinate system, called the **world** or **base** frame to which all objects including the manipulator are referenced. In this case we establish the base coordinate frame $o_0x_0y_0$ at the base of the robot, as shown in Figure 1.24. The coordinates (x, y) of the tool are expressed in this coordinate frame as

$$x \;=\; x_2 \;=\; \alpha_1 \cos\theta_1 + \alpha_2 \cos(\theta_1 + \theta_2) \tag{1.1}$$

$$y \;=\; y_2 \;=\; \alpha_1 \sin\theta_1 + \alpha_2 \sin(\theta_1 + \theta_2) \tag{1.2}$$

in which α_1 and α_2 are the lengths of the two links, respectively. Also the **orientation of the tool frame** relative to the base frame is given by the direction cosines of the x_2 and y_2 axes relative to the x_0 and y_0 axes, that is,

$$\begin{aligned} x_2 \cdot x_0 &= \cos(\theta_1 + \theta_2); \qquad x_2 \cdot y_0 = -\sin(\theta_1 + \theta_2) \\ y_2 \cdot x_0 &= \sin(\theta_1 + \theta_2); \qquad y_2 \cdot y_0 = \cos(\theta_1 + \theta_2) \end{aligned}$$

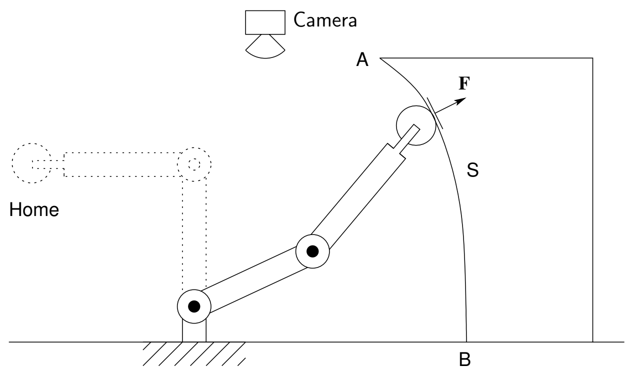

Fig. 1.23 Two-link planar robot example.

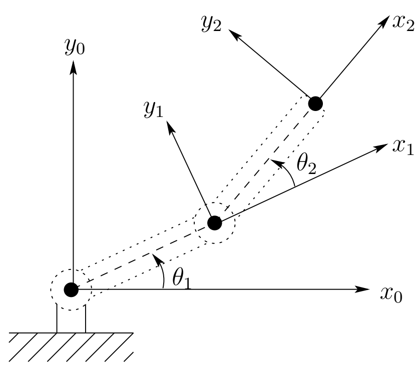

Fig. 1.24 Coordinate frames for two-link planar robot.

which we may combine into an **orientation matrix**

$$\begin{bmatrix} x_2 \cdot x_0 & y_2 \cdot x_0 \\ x_2 \cdot y_0 & y_2 \cdot y_0 \end{bmatrix} = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) \end{bmatrix} \tag{1.3}$$

Equations (1.1), (1.2) and (1.3) are called the **forward kinematic equations** for this arm. For a six degree-of-freedom robot these equations are quite complex and cannot be written down as easily as for the two-link manipulator. The general procedure that we discuss in Chapter 3 establishes coordinate frames at each joint and allows one to transform systematically among these frames using matrix transformations. The procedure that we use is referred to as the **Denavit-Hartenberg** convention. We then use **homogeneous coordinates** and **homogeneous transformations** to simplify the transformation among coordinate frames.

Inverse Kinematics

Now, given the joint angles θ_1, θ_2 we can determine the end-effector coordinates x and y. In order to command the robot to move to location A we need the inverse; that is, we need the joint variables θ_1, θ_2 in terms of the x and y coordinates of A. This is the problem of **inverse kinematics**. In other words, given x and y in the forward kinematic Equations (1.1) and (1.2), we wish to solve for the joint angles. Since the forward kinematic equations are nonlinear, a solution may not be easy to find, nor is there a unique solution in general. We can see in the case of a two-link planar mechanism that there may be no solution, for example if the given (x, y) coordinates are out of reach of the manipulator. If the given (x, y) coordinates are within the manipulator's reach there may be two solutions as shown in Figure 1.25, the so-called **elbow

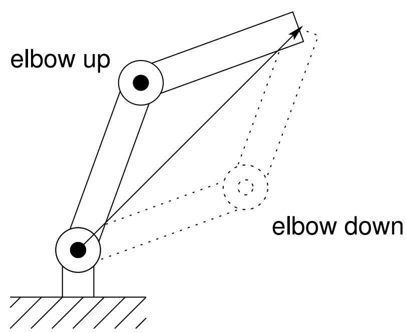

Fig. 1.25 Multiple inverse kinematic solutions.

up** and **elbow down** configurations, or there may be exactly one solution if the manipulator must be fully extended to reach the point. There may even be an infinite number of solutions in some cases (Problem 1-25).

Consider the diagram of Figure 1.26. Using the **Law of Cosines** we see that the angle θ_2 is given by

$$\cos\theta_2 = \frac{x^2 + y^2 - \alpha_1^2 - \alpha_2^2}{2\alpha_1\alpha_2} := D \tag{1.4}$$

We could now determine θ_2 as

$$\theta_2 = \cos^{-1}(D) \tag{1.5}$$

However, a better way to find θ_2 is to notice that if $\cos(\theta_2)$ is given by Equation (1.4) then $\sin(\theta_2)$ is given as

$$\sin(\theta_2) = \pm\sqrt{1 - D^2} \tag{1.6}$$

and, hence, θ_2 can be found by

$$\theta_2 = \tan^{-1} \frac{\pm\sqrt{1 - D^2}}{D} \tag{1.7}$$

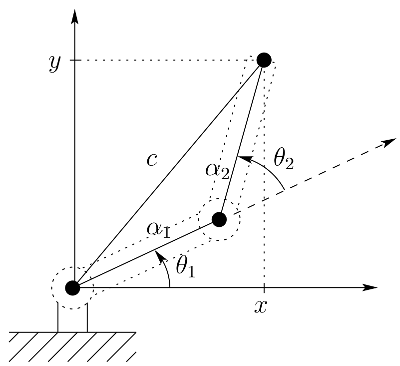

Fig. 1.26 Solving for the joint angles of a two-link planar arm.

The advantage of this latter approach is that both the elbow-up and elbow-down solutions are recovered by choosing the positive and negative signs in Equation (1.7), respectively.

It is left as an exercise (Problem 1-19) to show that θ_1 is now given as

$$\theta_1 \quad = \quad \tan^{-1}(y/x) - \tan^{-1}\left(\frac{\alpha_2 \sin\theta_2}{\alpha_1 + \alpha_2 \cos\theta_2}\right) \tag{1.8}$$

Notice that the angle θ_1 depends on θ_2. This makes sense physically since we would expect to require a different value for θ_1, depending on which solution is chosen for θ_2.

Velocity Kinematics

In order to follow a contour at constant velocity, or at any prescribed velocity, we must know the relationship between the velocity of the tool and the joint velocities. In this case we can differentiate Equations (1.1) and (1.2) to obtain

$$\begin{aligned} \dot{x} &= -\alpha_1 \sin\theta_1 \cdot \dot{\theta}_1 - \alpha_2 \sin(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2) \\ \dot{y} &= \alpha_1 \cos\theta_1 \cdot \dot{\theta}_1 + \alpha_2 \cos(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2) \end{aligned} \tag{1.9}$$

Using the vector notation $x = \begin{bmatrix} x \\ y \end{bmatrix}$ and $\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$ we may write these equations as

$$\begin{aligned} \dot{x} &= \begin{bmatrix} -\alpha_1 \sin\theta_1 - \alpha_2 \sin(\theta_1 + \theta_2) & -\alpha_2 \sin(\theta_1 + \theta_2) \\ \alpha_1 \cos\theta_1 + \alpha_2 \cos(\theta_1 + \theta_2) & \alpha_2 \cos(\theta_1 + \theta_2) \end{bmatrix} \dot{\theta} \\ &= J\dot{\theta} \end{aligned} \tag{1.10}$$

The matrix J defined by Equation (1.10) is called the **Jacobian** of the manipulator and is a fundamental object to determine for any manipulator.

In Chapter 4 we present a systematic procedure for deriving the Jacobian for any manipulator in the so-called **cross-product form**.

The determination of the joint velocities from the end-effector velocities is conceptually simple since the velocity relationship is linear. Thus the joint velocities are found from the end-effector velocities via the inverse Jacobian

$$\dot{\theta} \;=\; J^{-1}\dot{x} \tag{1.11}$$

where J^{-1} is given by

$$J^{-1} \;=\; \frac{1}{\alpha_1\alpha_2 s_{\theta_2}}\begin{bmatrix} \alpha_2 c_{\theta_1+\theta_2} & \alpha_2 s_{\theta_1+\theta_2} \\ -\alpha_1 c_{\theta_1} - \alpha_2 c_{\theta_1+\theta_2} & -\alpha_1 s_{\theta_1} - \alpha_2 s_{\theta_1+\theta_2} \end{bmatrix} \tag{1.12}$$

in which c_θ and s_θ denote respectively $\cos\theta$ and $\sin\theta$. The determinant of the Jacobian in Equation (1.10) is $\alpha_1\alpha_2\sin\theta_2$. The Jacobian does not have an inverse, therefore, when $\theta_2 = 0$ or π, in which case the manipulator is said to be in a **singular configuration**, such as shown in Figure 1.27 for $\theta_2 = 0$.

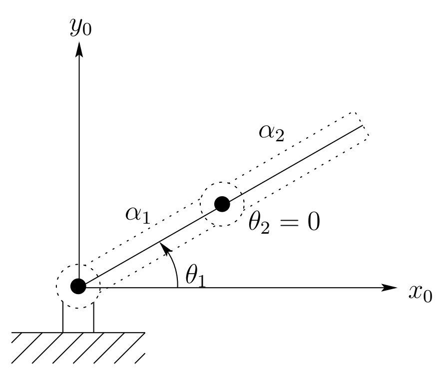

Fig. 1.27 A singular configuration.

The determination of such singular configurations is important for several reasons. At singular configurations there are infinitesimal motions that are unachievable; that is, the manipulator end-effector cannot move in certain directions. In the above cases the end effector cannot move in the positive x_2 direction when $\theta_2 = 0$. Singular configurations are also related to the nonuniqueness of solutions of the inverse kinematics. For example, for a given end-effector position, there are in general two possible solutions to the inverse kinematics. Note that a singular configuration separates these two solutions in the sense that the manipulator cannot go from one configuration to the other without passing through a singularity. For many applications it is important to plan manipulator motions in such a way that singular configurations are avoided.

Path Planning and Trajectory Generation

The robot control problem is typically decomposed hierarchically into three tasks: path planning, trajectory generation, and trajectory tracking. The path planning problem, considered in Chapter 5, is to determine a path in task space (or configuration space) to move the robot to a goal position while avoiding collisions with objects in its workspace. These paths encode position and orientation information without timing considerations, i.e. without considering velocities and accelerations along the planned paths. The trajectory generation problem, also considered in Chapter 5, is to generate reference trajectories that determine the time history of the manipulator along a given path or between initial and final configurations.

Dynamics

A robot manipulator is primarily a positioning device. To control the position we must know the dynamic properties of the manipulator in order to know how much force to exert on it to cause it to move: too little force and the manipulator is slow to react; too much force and the arm may crash into objects or oscillate about its desired position.

Deriving the dynamic equations of motion for robots is not a simple task due to the large number of degrees of freedom and nonlinearities present in the system. In Chapter 6 we develop techniques based on Lagrangian dynamics for systematically deriving the equations of motion of such a system. In addition to the rigid links, the complete description of robot dynamics includes the dynamics of the actuators that produce the forces and torques to drive the robot, and the dynamics of the drive trains that transmit the power from the actuators to the links. Thus, in Chapter 7 we also discuss actuator and drive train dynamics and their effects on the control problem.

Position Control

In Chapters 7 and 8 we discuss the design of control algorithms for the execution of programmed tasks. The motion control problem consists of the **Tracking and Disturbance Rejection Problem**, which is the problem of determining the control inputs necessary to follow, or **track**, a desired trajectory that has been planned for the manipulator, while simultaneously **rejecting** disturbances due to unmodeled dynamic effects such as friction and noise. We detail the standard approaches to robot control based on frequency domain techniques. We also introduce the notion of **feedforward control** and the techniques of **computed torque** and **inverse dynamics** as a means for compensating the complex nonlinear interaction forces among the links of the manipulator. Robust and adaptive control are introduced in Chapter 8 using the **Second Method of Lyapunov**. Chapter 10 provides some additional advanced techniques from nonlinear control theory that are useful for controlling high performance robots.

Force Control

Once the manipulator has reached location A. it must follow the contour S maintaining a constant force normal to the surface. Conceivably, knowing the location of the object and the shape of the contour, one could carry out this task using position control alone. This would be quite difficult to accomplish in practice, however. Since the manipulator itself possesses high rigidity, any errors in position due to uncertainty in the exact location of the surface or tool would give rise to extremely large forces at the end-effector that could damage the tool, the surface, or the robot. A better approach is to measure the forces of interaction directly and use a **force control** scheme to accomplish the task. In Chapter 9 we discuss force control and compliance along with common approaches to force control, namely **hybrid control** and **impedance control**.

Vision

Cameras have become reliable and relatively inexpensive sensors in many robotic applications. Unlike joint sensors, which give information about the internal configuration of the robot, cameras can be used not only to measure the position of the robot but also to locate objects external to the robot in its workspace. In Chapter 11 we discuss the use of computer vision to determine position and orientation of objects.

Vision-based Control

In some cases, we may wish to control the motion of the manipulator relative to some target as the end-effector moves through free space. Here, force control cannot be used. Instead, we can use computer vision to close the control loop around the vision sensor. This is the topic of Chapter 12. There are several approaches to vision-based control, but we will focus on the method of Image-Based Visual Servo (IBVS). This method has become very popular in recent years, and it relies on mathematical development analogous to that given in Chapter 4.

1.5 CHAPTER SUMMARY

In this chapter, we have given an introductory overview of some of the basic concepts required to develop mathematical models for robot arms. We have also discussed a few of the relevant mechanical aspects of robotic systems. In the remainder of the text, we will address the basic problems confronted in sensor-based robotic manipulation.

Many books have been written about these and more advance topics, including [1][3] [6][10][15][16][20] [22][31][34][43] [46][51][52][53] [61][65][69][70][77] [44][13]. There is a great deal of ongoing research in robotics. Current research results can be found in journals such as *IEEE Transactions on Robotics* (previously *IEEE Transactions on Robotics and Automation*), *IEEE Robotics and Automation Magazine*, *International Journal of Robotics Research*, *Robotics and Autonomous Systems*, *Journal of Robotic Systems*, *Robotica*, *Journal of Intelligent and Robotic Systems*, *Autonomous Robots*, *Advanced Robotics*. and in proceedings from conferences such as *IEEE International Conference on Robotics and Automation*, *IEEE International Conference on Intelligent Robots and Systems*, *Workshop on the Algorithmic Foundations of Robotics*, *International Symposium on Experimental Robotics*, and *International Symposium on Robotics Research*.

Problems

1-1 What are the key features that distinguish robots from other forms of automation such as CNC milling machines?

1-2 Briefly define each of the following terms: forward kinematics, inverse kinematics, trajectory planning, workspace, accuracy, repeatability, resolution, joint variable, spherical wrist, end effector.

1-3 What are the main ways to classify robots?

1-4 Make a list of robotics related magazines and journals carried by the university library.

1-5 Make a list of 10 robot applications. For each application discuss which type of manipulator would be best suited; which least suited. Justify your choices in each case.

1-6 List several applications for non-servo robots; for point-to point robots, for continuous path robots.

1-7 List five applications that a continuous path robot could do that a point-to-point robot could not do.

1-8 List five applications where computer vision would be useful in robotics.

1-9 List five applications where either tactile sensing or force feedback control would be useful in robotics.

1-10 Find out how many industrial robots are currently in operation in the United States. How many are in operation in Japan? What country ranks third in the number of industrial robots in use?

1-11 Suppose we could close every factory today and reopen then tomorrow fully automated with robots. What would be some of the economic and social consequences of such a development?

1-12 Suppose a law were passed banning all future use of industrial robots. What would be some of the economic and social consequences of such an act?

1-13 Discuss possible applications where redundant manipulators would be useful.

1-14 Referring to Figure 1.28, suppose that the tip of a single link travels a distance d between two points. A linear axis would travel the distance d

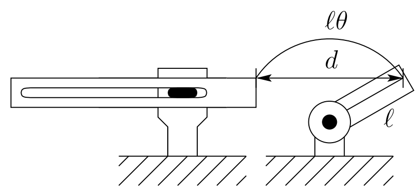

Fig. 1.28 Diagram for Problem 1-15

while a rotational link would travel through an arc length $\ell\theta$ as shown. Using the law of cosines show that the distance d is given by

$$d = \ell\sqrt{2(1 - \cos(\theta))}$$

which is of course less than $\ell\theta$. With 10-bit accuracy and $\ell = 1$m, $\theta = 90°$ what is the resolution of the linear link? of the rotational link?

1-15 A single-link revolute arm is shown in Figure 1.28. If the length of the link is 50 cm and the arm travels 180? what is the control resolution obtained with an 8-bit encoder?

1-16 Repeat Problem 1.15 assuming that the 8-bit encoder is located on the motor shaft that is connected to the link through a 50:1 gear reduction. Assume perfect gears.

1-17 Why is accuracy generally less than repeatability?

1-18 How could manipulator accuracy be improved using direct endpoint sensing? What other difficulties might direct endpoint sensing introduce into the control problem?

1-19 Derive Equation (1.8).

1-20 For the two-link manipulator of Figure 1.24 suppose $\alpha_1 = \alpha_2 = 1$. Find the coordinates of the tool when $\theta_1 = \frac{\pi}{6}$ and $\theta_2 = \frac{\pi}{2}$.

1-21 Find the joint angles θ_1, θ_2 when the tool is located at coordinates $\left(\frac{1}{2}, \frac{1}{2}\right)$.

1-22 If the joint velocities are constant at $\dot{\theta}_1 = 1$, $\dot{\theta}_2 = 2$, what is the velocity of the tool? What is the instantaneous tool velocity when $\theta_1 = \theta_2 = \frac{\pi}{4}$?

1-23 Write a computer program to plot the joint angles as a function of time given the tool locations and velocities as a function of time in Cartesian coordinates.

1-24 Suppose we desire that the tool follow a straight line between the points (0,2) and (2,0) at constant speed s. Plot the time history of joint angles.

1-25 For the two-link planar manipulator of Figure 1.24 is it possible for there to be an infinite number of solutions to the inverse kinematic equations? If so, explain how this can occur.