

THE COMPUTER SCIENCE AND ENGINEERING HANDBOOK

EDITOR-IN-CHIEF
ALLEN B. TUCKER, JR.

CRC PRESS

A CRC Handbook Published in Cooperation with ACM, The Association for Computing

33

Robotics

33.1 Introduction 734

33.2 Robot Workcells 735

33.3 Workcell Command and Information Organization 737
Intelligent Control Architectures • Behaviors and Hybrid Systems Design • Workcell Planning, Coordination, and Control Structure

33.4 Commercial Robot Configurations and Types 740
Manipulator Performance • Common Kinematic Configurations • Drive Types of Commercial Robots • Commercial Robot Controllers

33.5 Robot Kinematics, Dynamics, and Servo-level Control 744
Kinematics and Jacobians • Robot Dynamics and Properties • Robot Servo-level Motion Control • Robot Force/Torque Servocontrol • Motion Trajectory Generation

33.6 End Effectors and End-of-Arm Tooling 752
Part Fixtures and Robot Tooling • Grippers and Fingers • Robot Wrist Mechanisms • Robot/Tooling Process Integration and Coordination

33.7 Sensors 756
The Philosophy of Robotic Workcell Sensors • Types of Sensors • Sensor Data Processing • Vision for Robotics

33.8 Workcell Planning 762
Workcell Behaviors and Agents • Task Decomposition and Planning • Task Matrix Approach to Workcell Planning • Path Planning

33.9 Job and Activity Coordination 772
Matrix Rule-Based Job Coordination Controller • Process Integration, Digital I/O, and Job Coordination Controller Implementation • Coordination of Multiple Robots

33.10 Error Detection and Recovery 775
Error Detection • Error Recovery

33.11 Human Operator Interfaces 776
Levels of User Interface • Mechanisms for User Interface

33.12 Robot Workcell Programming 777
Robot Programming Languages • V+, A Representative Robot Language

33.13 Mobile Robots and Automated Guided Vehicles 780
Mobile Robots • Automated Guided Vehicle Systems

F. L. Lewis
University of Texas at Arlington

M. Fitzgerald
University of Texas at Arlington

K. Liu
University of Texas at Arlington

33.1 Introduction

The word *robot* was introduced by the Czech playright Karel Čapek in his 1920 play *Rossum's Universal Robots*. The word *robota* in Czech means simply work. In spite of such practical beginnings, science fiction writers and early Hollywood movies have given us a romantic notion of robots and expectations that they will revolutionize several walks of life including industry. However, many of the more far-fetched expectations from robots have failed to materialize. For instance, in underwater assembly and

0-8493-2909-4/97/$0.00+$.50
© 1997 by CRC Press, Inc.

oil mining, teleoperated robots are very difficult to manipulate due to sea currents and low visibility, and have largely been replaced or augmented by automated smart quick-fit couplings that simplify the assembly task. However, through good design practices and painstaking attention to detail, engineers have succeeded in applying robotic systems to a wide variety of industrial and manufacturing situations where the environment is *structured* or predictable. Thus, the first successful commercial implementation of process robotics was in the U.S. automobile industry; the word *automation* was coined in the 1940s at Ford Motor Company, a contraction of automatic motivation.

As machines, robots have precise motion capabilities, repeatability, and endurance. On a practical level, robots are distinguished from other electromechanical motion equipment by their dexterous manipulation capability in that robots can work, position, and move tools and other objects with far greater dexterity than other machines found in the factory. The capabilities of robots are extended by using them as a basis for *robotic workcells.* Process robotic workcells are integrated functional systems with grippers, **end effectors**, sensors, and process equipment organized to perform a controlled sequence of jobs to execute a process. Robots must coordinate with other devices in the workcell such as machine tools, conveyors, part feeders, cameras, and so on. Sequencing jobs to correctly perform automated tasks in such circumstances is not a trivial matter, and robotic workcells require sophisticated planning, sequencing, and control systems.

Today, through developments in computers and artificial intelligence (AI) techniques (and often motivated by the space program), we are on the verge of another breakthrough in robotics that will afford some levels of autonomy in *unstructured environments.* For applications requiring increased autonomy it is particularly important to focus on the design of the data structures and command-and-control information flow in the robotic system. Therefore, this chapter focuses on the design of robotic workcell *systems.* A distinguishing feature of robotics is its multidisciplinary nature: to successfully design robotic systems one must have a grasp of electrical, mechanical, industrial, and computer engineering, as well as economics and business practices. The purpose of this chapter is to provide a background in these areas so that design of robotic systems may be approached from a position of rigor, insight, and confidence.

The chapter begins by discussing layouts and architectures for robotic workcell design. Then, components of the workcell are discussed from the bottom up, beginning with robots, sensors, and conveyors/part feeders, and progressing upwards in abstraction through task coordination, job sequencing, and resource dispatching, to task planning, assignment, and decomposition. Concepts of user interface and exception handling/fault recovery are included.

33.2 Robot Workcells

In factory automation and elsewhere it was once common to use layouts such as the one in Fig. 33.1, which shows an assembly line with distinct workstations, each performing a dedicated function. Robots have been used at the workstation level to perform operations such as assembly, drilling, surface finishing, welding, palletizing, and so on. In the assembly line, parts are routed sequentially to the workstations by conveyors. Such systems are very expensive to install, require a cadre of engineering experts to design and program, and are extremely difficult to modify or reprogram as needs change. In today's high-mix low-volume (HMLV) manufacturing scenario, these characteristics tolled the death knell for such rigid antiquated designs.

In the assembly line, the robot is *restricted* by placing it into a rigid sequential system. Robots are versatile machines with many capabilities, and their potential can be significantly increased by using them as a basis for robotic workcells such as the one in Fig. 33.2 [Decelle 1988, Jamshidi et al. 1992, Pugh 1983]. In the robotic workcell, robots are used for part handling, assembly, and other process operations. The workcell is designed to make full use of the workspace of the robots, and components such as milling machines, drilling machines, vibratory part feeders, and so on are placed within the robots' workspaces to allow servicing by the robots. Contrary to the assembly line, the physical layout does not impose a priori a fixed sequencing of the operations or jobs. Thus, as product requirements change, all that is required is to reprogram the workcell in software. The workcell is ideally suited to emerging HMLV conditions in manufacturing and elsewhere.

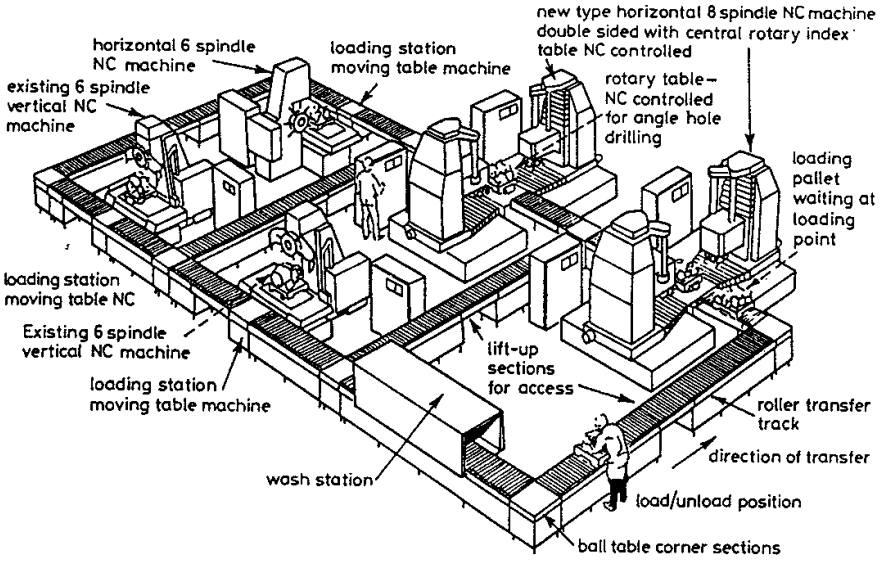


FIGURE 33.1 Antiquated sequential assembly line with dedicated workstations. (*Source:* Courtesy of Edkins, M. 1983. Linking industrial robots and machine tools. In *Robotic Technology.* A. Pugh, ed. Peregrinus, London. With permission.)

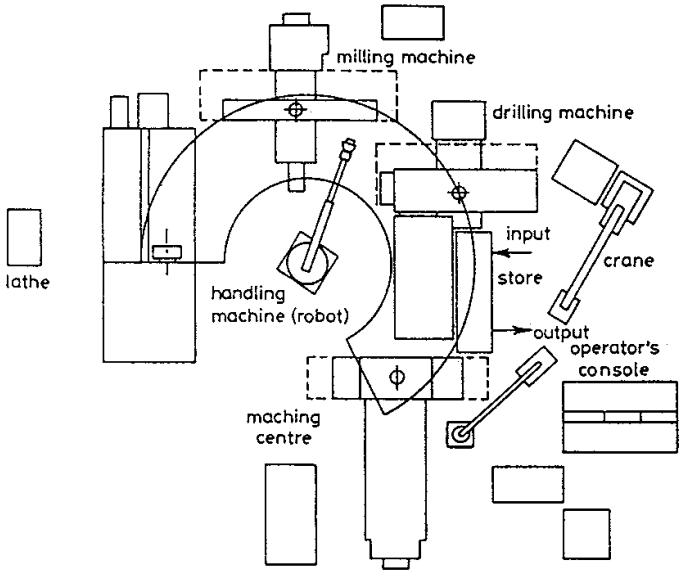


FIGURE 33.2 Robot workcell. (*Source:* Courtesy of Edkins, M. 1983. Linking industrial robots and machine tools. In *Robotic Technology.* A. Pugh, ed. Peregrinus, London. With permission.)

The rising popularity of robotic workcells has taken emphasis away from hardware design and placed new emphasis on innovative *software techniques and architectures* that include planning, coordination, and control (PC&C) functions. Research into individual robotic devices is becoming less useful; what is needed are rigorous design and analysis techniques for integrated multirobotic systems.

33.3 Workcell Command and Information Organization

In this section we define some terms, discuss the design of intelligent control systems, and specify a planning, coordination, and control structure for robotic workcells. The remainder of the chapter is organized around that structure. The various architectures used for modeling AI systems are relevant to this discussion, although here we specialize the discussion to intelligent control architecture.

Intelligent Control Architectures

Many structures have been proposed under the general aegis of the so-called intelligent control (IC) architectures [Antsaklis and Passino 1992]. Despite frequent heated philosophical discussions, it is now becoming clear that most of the architectures have much in common, with apparent major differences due to the fact that different architectures focus on different aspects of intelligent control or different levels of abstraction. A general IC architecture based on work by Saridis is given in Fig. 33.3, which illustrates the principle of decreasing precision with increasing abstraction [Saridis 1996]. In this figure, the organization level performs as a manager that schedules and assigns tasks, performs task decomposition and planning, does path planning, and determines for each task the required job sequencing and assignment of resources. The coordination level performs the prescribed job sequencing, coordinating the workcell agents or resources; in the case of shared resources it must execute dispatching and conflict resolution.

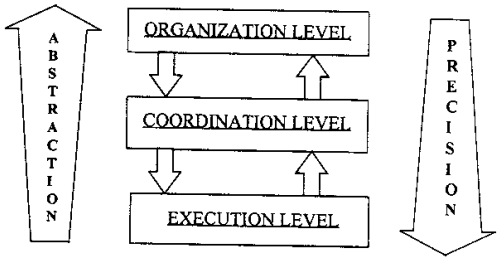


FIGURE 33.3 Three-level intelligent control architecture from work by Saridis.

The *agents* or *resources* of the workcell include robot manipulators, grippers and tools, conveyors and part feeders, sensors (e.g., cameras), mobile robots, and so on. The execution level contains a closed-loop controller for each agent that is responsible for the real-time performance of that resource, including trajectory generation, motion and force feedback servo-level control, and so on. Some permanent built-in motion sequencing may be included (e.g., stop robot motion prior to opening the gripper).

At each level of this hierarchical IC architecture, there may be several systems or nodes. That is, the architecture is not strictly hierarchical. For instance, at the execution level there is a real-time controller for each workcell agent. Several of these may be coordinated by the coordination level to sequence the jobs needed for a given task. At each level, each node is required to sense conditions, make decisions, and give commands or status signals. This is captured in the sense/world-model/execute (SWE) paradigm of Albus [1992], shown in the NASREM configuration in Fig. 33.4; each node has the SWE structure.

Behaviors and Hybrid Systems Design

In any properly designed IC system, the supervisory levels should not destroy the capabilities of the systems supervised. Thus, design should proceed in the manner specified by Brooks [1986], where *behaviors* are built in at lower levels, then selected, activated, or modified by upper-level supervisors. From the point of view of still higher level nodes, the composite performance appears in terms of new more complex or emergent behaviors. Such *subsumption* design proceeds in the manner of adding layers to an onion, as depicted loosely in Fig. 33.5.

Near or slightly below the interfaces between the coordination level and the execution level one must face the transition between two fundamentally distinct worlds. Real-time servo-level controller design and control may be accomplished in terms of *state-space systems*, which are time-varying dynamical systems (either continuous time or discrete time) having continuous-valued states such as temperatures, pressures, motions, velocities, forces, and so on. On the other hand, the coordinator is not concerned about such

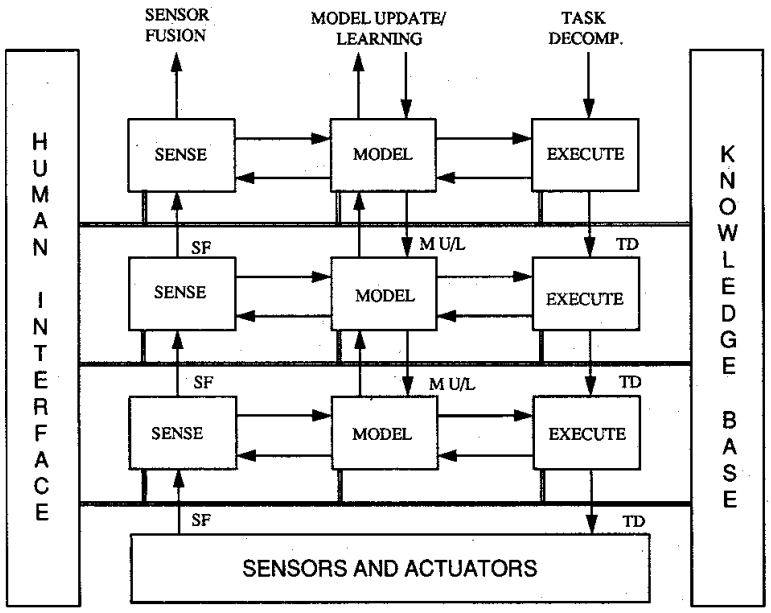


FIGURE 33.4 Three-element structure at all levels of the IC architecture: the NASREM paradigm.

details, but speaks in terms of *discrete events* such as "perform this job" or "check this condition." The theory of *hybrid systems* is concerned with the interface between continuous-state systems and discrete event systems.

These concepts are conveniently illustrated by figures such as Fig. 33.6, where a closed-loop real-time feedback controller for the plant having dynamics $\dot{x} = f(x, u)$ is shown at the execution level. The function of the coordinator is to select the details of this real-time feedback control structure; that is, the outputs $z(t)$, control inputs $u(t)$, prescribed reference trajectories $r(t)$, and controllers K to be switched in at the low level. Selecting the outputs amounts to selecting which sensors to read; selecting the control inputs amounts to selecting to which actuators the command signals computed by the controller should be sent. The controller K is selected from a library of stored predesigned controllers.

A specific combination of (z, u, r, K) defines a *behavior* of the closed-loop system. For instance, in a mobile robot, for path-following behavior one may select: as outputs, the vehicle speed and heading; as controls, the speed and steering inputs; as the controller, an adaptive **proportional-integral-derivative (PID) controller**; and as reference input, the prescribed path. For wall-following behavior, for instance, one

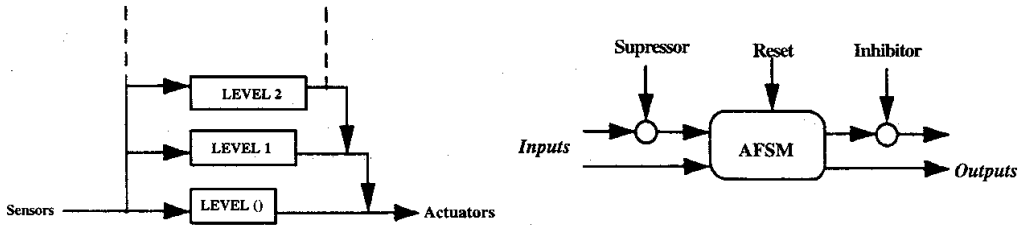


FIGURE 33.5 Behavior-based design after the subsumption technique of Brooks.

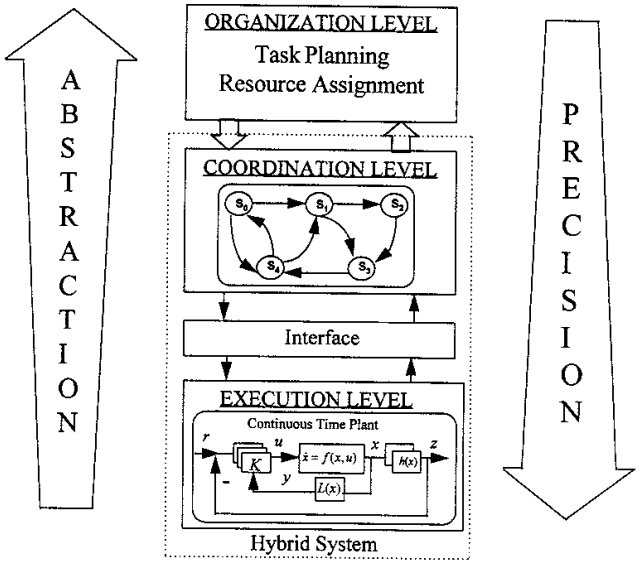


FIGURE 33.6 Hybrid systems approach to defining and sequencing the plant behaviors.

simply selects as output the sonar distance from the wall, as input the steering command, and as reference input the prescribed distance to be maintained. These distinct closed-loop behaviors are sequenced by the coordinator to perform the prescribed job sequence.

Workcell Planning, Coordination, and Control Structure

A convenient planning, coordination, and control structure for robotic workcell design and operation is given in Fig. 33.7, which is modified from the next generation controller (NGC) paradigm. This is an operational PC&C architecture fully consistent with the previous IC structures. In this figure, the term *virtual agent* denotes the agent plus its low-level servocontroller and any required built-in sequencing coordinators. For instance, a *virtual robot* includes the manipulator, its commercial controller with servo-level joint controllers and trajectory generator, and in some applications the gripper controller plus an agent internal coordinator to sequence manipulator and gripper activities. A *virtual camera* might include the camera(s) and framegrabber board, plus software algorithms to perform basic vision processing such as edge detection, segmentation, and so on; thus, the virtual camera could include a *data abstraction*, which is a set of data plus manipulations on that data.

The remainder of the chapter is structured after this PC&C architecture, beginning at the execution level to discuss robot manipulator kinematics, dynamics and control; end effectors and tooling; sensors; and other workcell components such as conveyors and part feeders. Next considered is the coordination level including sequencing control and dispatching of resources. Finally, the organization level is treated including task planning, path planning, workcell management, task assignment, and scheduling.

Three areas are particularly problematic. At each level there may be *human operator interfaces*; this complex topic is discussed in a separate section. An equally complex topic is *error detection and recovery*, also allotted a separate section, which occurs at several levels in the hierarchy. Finally, the strict NGC architecture has a component known as the *information* or *knowledge base*; however, in view of the fact that all nodes in the architecture have the SWE structure shown in Fig. 33.4, it is clear that the knowledge base is distributed throughout the system in the world models of the nodes. Thus, a separate discussion on this component is not included.

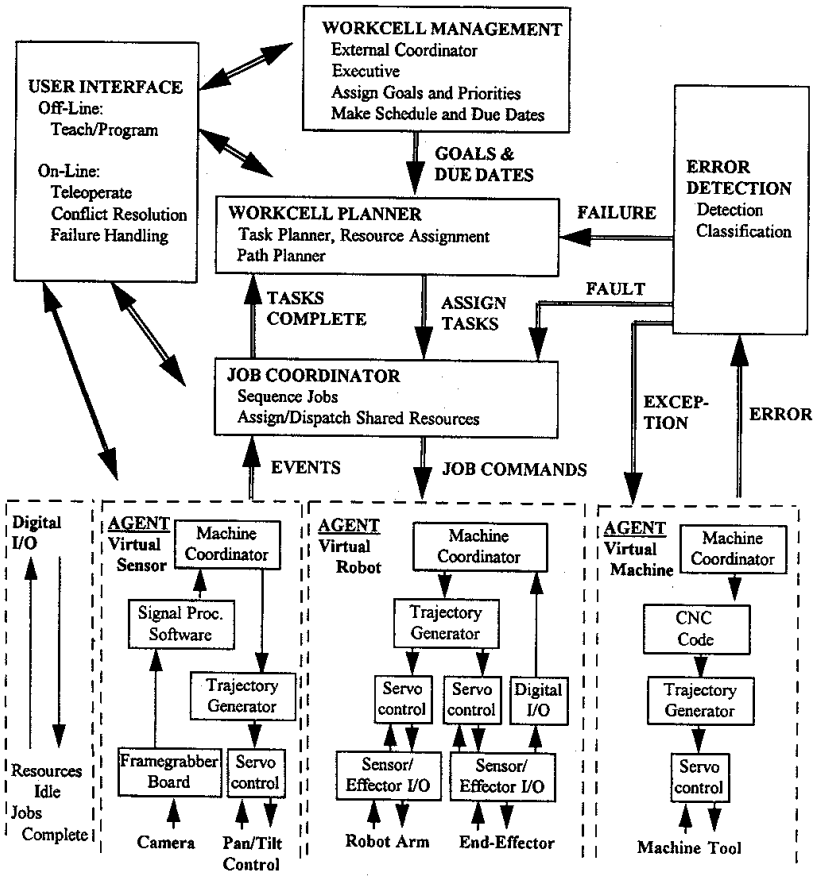


FIGURE 33.7 Robotic workcell planning, coordination, and control operational architecture.

33.4 Commercial Robot Configurations and Types

Robots are highly reliable, dependable, and technologically advanced factory equipment. The majority of the world's robots are supplied by established companies using reliable off-the-shelf component technologies. All commercial industrial robots have two physically separate basic elements, the manipulator arm and the controller. The basic architecture of all commercial robots is fundamentally the same, and consists of digital servocontrolled electrical motor drives on serial-link kinematic machines, usually with no more than six axes (degrees of freedom). All are supplied with a proprietary controller. Virtually all robot applications require significant design and implementation effort by engineers and technicians. What makes each robot unique is how the components are put together to achieve performance that yields a competitive product. The most important considerations in the application of an industrial robot center on two issues: manipulation and integration.

Manipulator Performance

The combined effects of kinematic structure, axis drive mechanism design, and real-time motion control determine the major manipulation performance characteristics: reach and dexterity, payload, quickness, and precision. Caution must be used when making decisions and comparisons based on manufacturers' published performance specifications because the methodsfor measuring and reporting them are not

standardized across the industry. Usually motion testing, simulations, or other analysis techniques are used to verify performance for each application.

Reach is characterized by measuring the extent of the *workspace* described by the robot motion and *dexterity* by the angular displacement of the individual joints. Some robots will have unusable spaces such as dead zones, singular poses, and wrist-wrap poses inside of the boundaries of their reach.

Payload weight is specified by the manufacturers of all industrial robots. Some manufacturers also specify inertial loading for rotational wrist axes. It is common for the payload to be given for extreme velocity and reach conditions. Weight and inertia of all tooling, workpieces, cables and hoses must be included as part of the payload.

Quickness is critical in determining throughput but difficult to determine from published robot specifications. Most manufacturers will specify a maximum speed of either individual joints or for a specific kinematic tool point. However, *average speed* in a working cycle is the quickness characteristic of interest.

Precision is usually characterized by measuring **repeatability**. Virtually all robot manufacturers specify static position repeatability. **Accuracy** is rarely specified, but it is likely to be at least four times larger than repeatability. Dynamic precision, or the repeatability and accuracy in tracking position, velocity, and acceleration over a continuous path, is not usually specified.

Common Kinematic Configurations

All common commercial industrial robots are serial-link manipulators, usually with no more than six kinematically coupled axes of motion. By convention, the axes of motion are numbered in sequence as they are encountered from the base on out to the wrist. The first three axes account for the spatial positioning motion of the robot; their configuration determines the shape of the space through which the robot can be positioned. Any subsequent axes in the kinematic chain generally provide rotational motions to orient the end of the robot arm and are referred to as *wrist axes.* There are two primary types of motion that a **robot axis** can produce in its driven link—either **revolute** or **prismatic.** It is often useful to classify robots according to the orientation and type of their first three axes. There are four very common commercial robot configurations: articulated, type I **selectively compliant assembly robot arm** (**SCARA**), type II SCARA, and Cartesian. Two other configurations, cylindrical and spherical, are now much less common.

Articulated Arms. The variety of commercial articulated arms, most of which have six axes, is very large (Fig. 33.8). All of these robot's axes are revolute. The second and third axes are parallel and work together

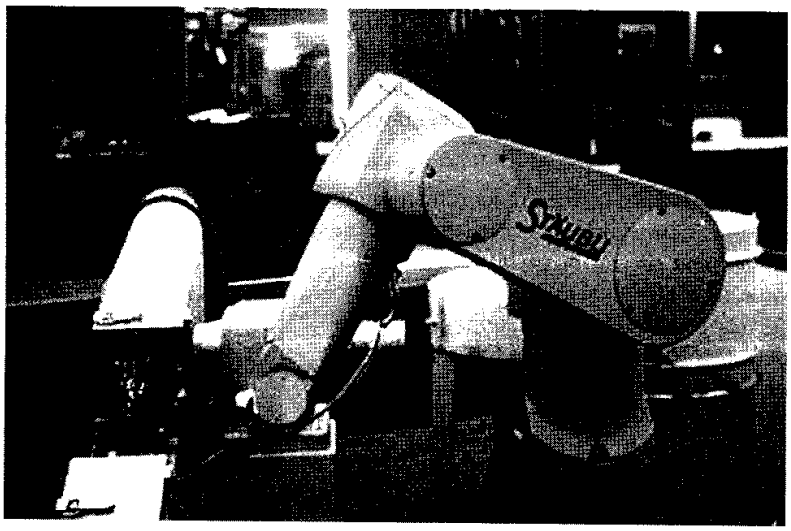


FIGURE 33.8 Articulated arm; six-axis arm grinding from a casting. (*Source:* Courtesy of Staubli Unimation, Inc.).

to produce motion in a vertical plane. The first axis in the base is vertical and revolves the arm to sweep out a large work volume. Many different types of drive mechanisms have been devised to allow wrist and forearm drive motors and gearboxes to be mounted close to the first and second axis of rotation, thus minimizing the extended mass of the arm. The workspace efficiency of well-designed articulated arms, which is the degree of quick dexterous reach with respect to arm size, is unsurpassed by other arm configurations when five or more degrees of freedom are needed. A major limiting factor in articulated arm performance is that the second axis has to work to lift both the subsequent arm structure and the payload. Historically, articulated arms have not been capable of achieving accuracy as well as other arm configurations, as all axes have joint angle position errors which are multiplied by link radius and accumulated for the entire arm.

Type I SCARA. The type I SCARA (selectively compliant assembly robot arm) uses two parallel revolute joints to produce motion in the horizontal plane (Fig. 33.9). The arm structure is weight-bearing but the first and second axes do no lifting. The third axis of the type I SCARA provides work volume by adding a vertical or z axis. A fourth revolute axis will add rotation about the z axis to control orientation in the horizontal plane. This type of robot is rarely found with more than four axes. The type I SCARA is used extensively in the assembly of electronic components and devices, and it is used broadly for the assembly of small- and medium-sized mechanical assemblies.

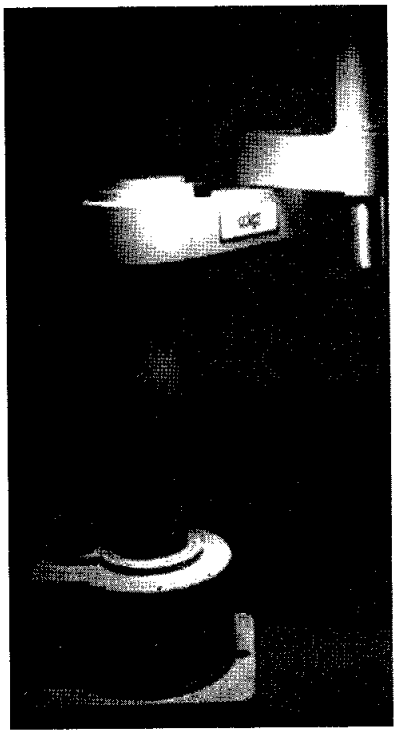


FIGURE 33.9 Type I SCARA arm. High-precision, high-speed midsized SCARA I. (*Source:* Courtesy of Adept Technologies, Inc. With permission.)

Type II SCARA. The type II SCARA, also a four-axis configuration, differs from type I in that the first axis is a long vertical prismatic z stroke, which lifts the two parallel revolute axes and their links (Fig. 33.10). For quickly moving heavier loads (over approximately 75 lb) over longer distance (more than about 3 ft), the type II SCARA configuration is more efficient than the type I.

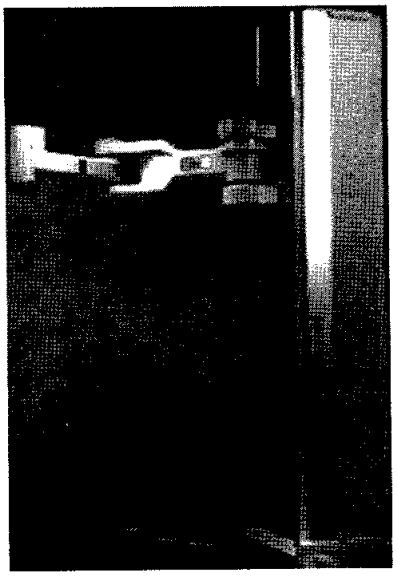


FIGURE 33.10 Type II SCARA. (*Source:* Courtesy of Adept Technologies, Inc. With permission.)

Cartesian Coordinate Robots. Cartesian coordinate robots use orthogonal prismatic axes, usually referred to as x, y, and z, to translate their end effector or payload through their rectangular workspace (Fig. 33.11). One, two, or three revolute wrist axes may be added for orientation. Commercial robot companies supply several types of Cartesian coordinate robots with workspace sizes ranging from a few cubic inches to tens of thousands of cubic feet, and payloads ranging to several hundred pounds. Gantry robots, which have an elevated bridge structure, are the most common Cartesian style and are well suited to material handling applications where large areas and/or large loads must be serviced. They are particularly useful in applications such as arc welding,

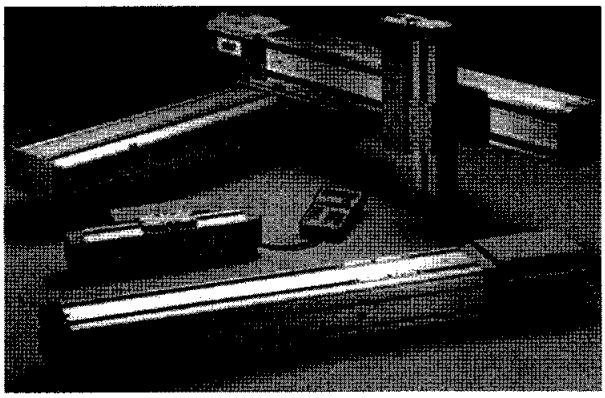


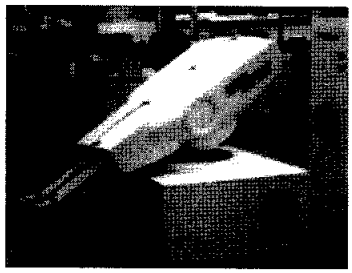
FIGURE 33.11 Cartesian robot. Three-axis robot constructed from modular single-axis motion modules. (*Source:* Courtesy of Adept Technologies, Inc. With permission.)

waterjet cutting, and inspection of large complex precision parts. Modular Cartesian robots are also commonly available from several commercial sources. Each module is a self-contained completely functional single-axis actuator; the modules may be custom assembled for special-purpose applications.

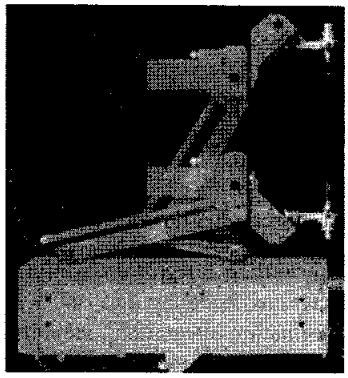
Spherical and Cylindrical Coordinate Robots. The first two axes of the spherical coordinate robot are revolute and orthogonal to one another, and the third axis provides prismatic radial extension. The result is a natural spherical coordinate system with a spherical work volume. The first axis of cylindrical coordinate robots is a revolute base rotation. The second and third are prismatic, resulting in a natural cylindrical motion. Commercial models of spherical and cylindrical robots (Fig. 33.12) were originally very common and popular in machine tending and material handling applications. Hundreds are still in use but now there are only a few commercially available models. The decline in use of these two configurations is attributed to problems arising from use of the prismatic link for radial extension/retraction motion; a solid boom requires clearance to fully retract.

Drive Types of Commercial Robots

The vast majority of commercial industrial robots use electric servomotor drives with speed reducing transmissions. Both ac and dc motors are popular. Some servohydraulic articulated arm robots are available now for painting applications. It is rare to find robots with servopneumatic drive axes. All types of mechanical transmissions are used, but the tendency is toward low- and zero-backlash type drives. Some robots use direct drive methods to eliminate the amplification of inertia and mechanical backlash associated with other drives. Joint angle position sensors, required for real-time servo-level control, are generally considered an important part of the drive train. Less often, velocity feedback sensors are provided.



(a)



(b)

FIGURE 33.12 Spherical and cylindrical robots. (a) Hydraulic powered spherical robot. (*Source:* Courtesy of Kohol Systems, Inc. With permission.) (b) Cylindrical arm using scissor mechanism for radial prismatic motion. (*Source:* Courtesy of Yamaha Robotics. With permission.)

Commercial Robot Controllers

Commercial robot controllers are specialized multiprocessor computing systems that provide four basic processes allowing integration of the robot into an automation system: motion trajectory generation and following, motion/process integration and sequencing, human user integration, and information integration.

Motion Trajectory Generation and Following. There are two important controller related aspects of industrial robot motion generation. One is the extent of manipulation that can be programmed, the other is the ability to execute controlled programmed motion. A unique aspect of each robot system is its real-time servo-level motion control. The details of real-time control are typically not revealed to the user due to safety and proprietary information secrecy reasons. Each robot controller, through its operating system programs, converts digital data from higher level coordinators into coordinated arm motion through precise computation and high-speed distribution and communication of the individual axis motion commands, which are executed by individual joint servocontrollers. Most commercial robot controllers operate at a sample period of 16 ms. The real-time motion controller invariably uses classical independent-joint proportional-integral-derivative control or simple modifications of PID. This makes commercially available controllers suitable for point-to-point motion, but most are not suitable for following continuous position/velocity profiles or exerting prescribed forces without considerable programming effort, if at all.

Motion/Process Integration and Sequencing. Motion/process integration involves coordinating manipulator motion with process sensors or other process controller devices. The most primitive process integration is through discrete digital input/output (I/O). For example, a machine controller external to the robot controller might send a 1-b signal indicating that it is ready to be loaded by the robot. The robot controller must have the ability to read the signal and to perform logical operations (if then, wait until, do until, etc.) using the signal. Coordination with sensors (e.g., vision) is also often provided.

Human Integration. The controller's human interfaces are critical to the expeditious setup and programming of robot systems. Most robot controllers have two types of human interface available: computer style CRT/keyboard terminals for writing and editing program code off line, and *teach pendants*, which are portable manual input terminals used to command motion in a telerobotic fashion via touch keys or joy sticks. Teach pendants are usually the most efficient means available for positioning the robot, and a memory in the controller makes it possible to play back the taught positions to execute motion trajectories. With practice, human operators can quickly teach a series of points which are chained together in playback mode. Most robot applications currently depend on the integration of human expertise during the programming phase for the successful planning and coordination of robot motion. These interface mechanisms are effective in unobstructed workspaces where no changes occur between programming and exceution. They do not allow human interface during execution or adaptation to changing environments.

Information Integration. Information integration is becoming more important as the trend toward increasing flexibility and agility impacts robotics. Many commercial robot controllers now support information integration functions by employing integrated personal computer (PC) interfaces through the communications ports (e.g., RS-232), or in some through direct connections to the robot controller data bus.

33.5 Robot Kinematics, Dynamics, and Servo-level Control

In this section we shall study the kinematics, dynamics, and servocontrol of robot manipulators; for more details see Lewis et al. [1993]. The objective is to turn the manipulator, by proper design of the control system and trajectory generator, into an *agent with desirable behaviors*, which behaviors can then be selected by the job coordinator to perform specific jobs to achieve some assigned task. This agent, composed of