2

Systems Overview of a Robot

2.0 OBJECTIVES

In this chapter we define the components of a robot from a systems approach, expanding on some of the ideas presented in Chapter 1. Some of the functions (and/or features) that a robot should be capable of performing whether alone or as part of a more extensive system are described. Upon completion, the reader should be able to identify the major system components of a robot and should understand the required functionality of a robot and its controller necessary for the unit to be properly integrated and utilized in a real-world environment. In addition, the reader should have an appreciation for interpreting robot system specifications. Specifically, the topics that will be covered are:

- The basic components of a robot system
- The robot as part of a workcell
- The functions required of a robot system
- The specifications of robot systems

2.1 MOTIVATION

The field of robotics draws on a multitude of engineering disciplines. Obviously, there are mechanical, electrical, and software considerations. However, the interaction among these and other disciplines is quite complex. Consider, for example, some of the design criteria required for a multijointed arm so that it is capable of moving along a straight path.

- The geometry of the manipulator must be such that it can position its tool along the path.
- The required positions (*set points**) for the servos that drive each joint must be generated in real time, usually by a computer.
- The servo system must be capable of responding to the set points and of driving each joint so that the tool traces out a straight trajectory (note that this is related to bandwidth and linear operation of the servos).
- The joint actuators (motors) must be sized properly to provide the torques needed as the arm moves (note that the inertias reflected into each joint may be a function of position).
- The feedback transducers must have the proper resolution so that the servos can control the joint positions within some defined error.
- The mechanical system itself must meet some predefined degree of stiffness, accuracy, and repeatability. (Thus the proper materials must be chosen for its construction to meet these requirements. Also, consideration must be given to the thermal properties of these materials.)

This brief discussion demonstrates the interdisciplinary nature of a robot and points out the need for good communication among the various engineers involved in the design of a robot system.

The following sections use a "top-down" description of a relatively sophisticated computer-controlled robot system so as to acquaint the reader with its many subsystems or "functional blocks" and give some indication of where and why interactions among them occur.

2.2 BASIC COMPONENTS OF A ROBOT SYSTEM

Recall from Chapter 1 that the four basic components of a robot system are:

- Manipulator
- Sensory devices
- Controller
- Power conversion unit

Figure 2.2.1 shows these components connected as a system. It is important to note that the sensory devices are spread throughout the system. For example, in

---

*Set points are commands given to a servo system. The system attempts to adjust its output so that it coincides with a given set point.

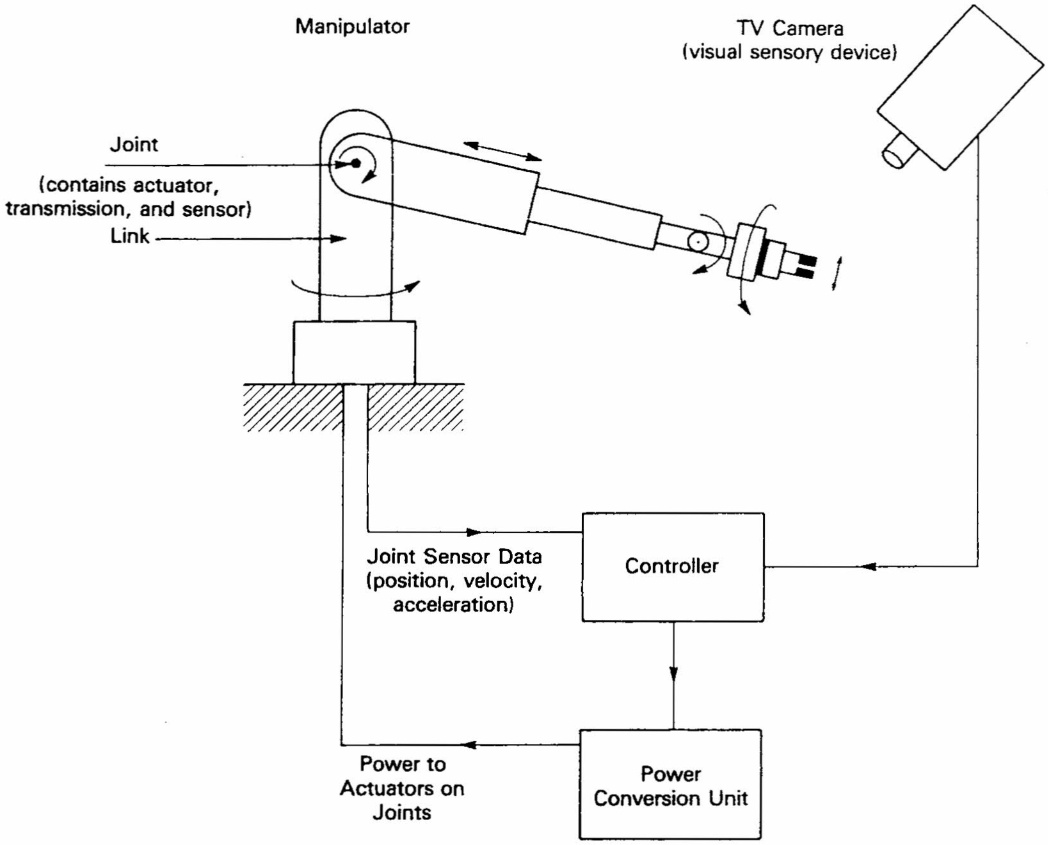

Figure 2.2.1. Components of a robot system.

addition to the TV camera (a visual sensor), each joint contains sensors for position, velocity, and/or acceleration. In addition, some of the power conversion hardware may be located inside the manipulator. The following discussion defines the functions and subdivisions of these major components.

2.2.1 Manipulator

The manipulator consists of a series of rigid members, called *links*, connected by *joints*. Motion of a particular joint causes subsequent links attached to it to move. The motion of the joint is accomplished by an actuator mechanism. The actuator can be connected directly to the next link or through some mechanical transmission (in order to produce a torque or speed advantage or "gain"). The manipulator ends with a link on which a tool can be mounted. The interface between the last link and the tool or end effector is called the *tool mounting plate* or *tool flange*.

The manipulator itself may be thought of as being composed of three divisions:

- The major linkages
- The minor linkages (wrist components)
- The end effector (gripper or tool)

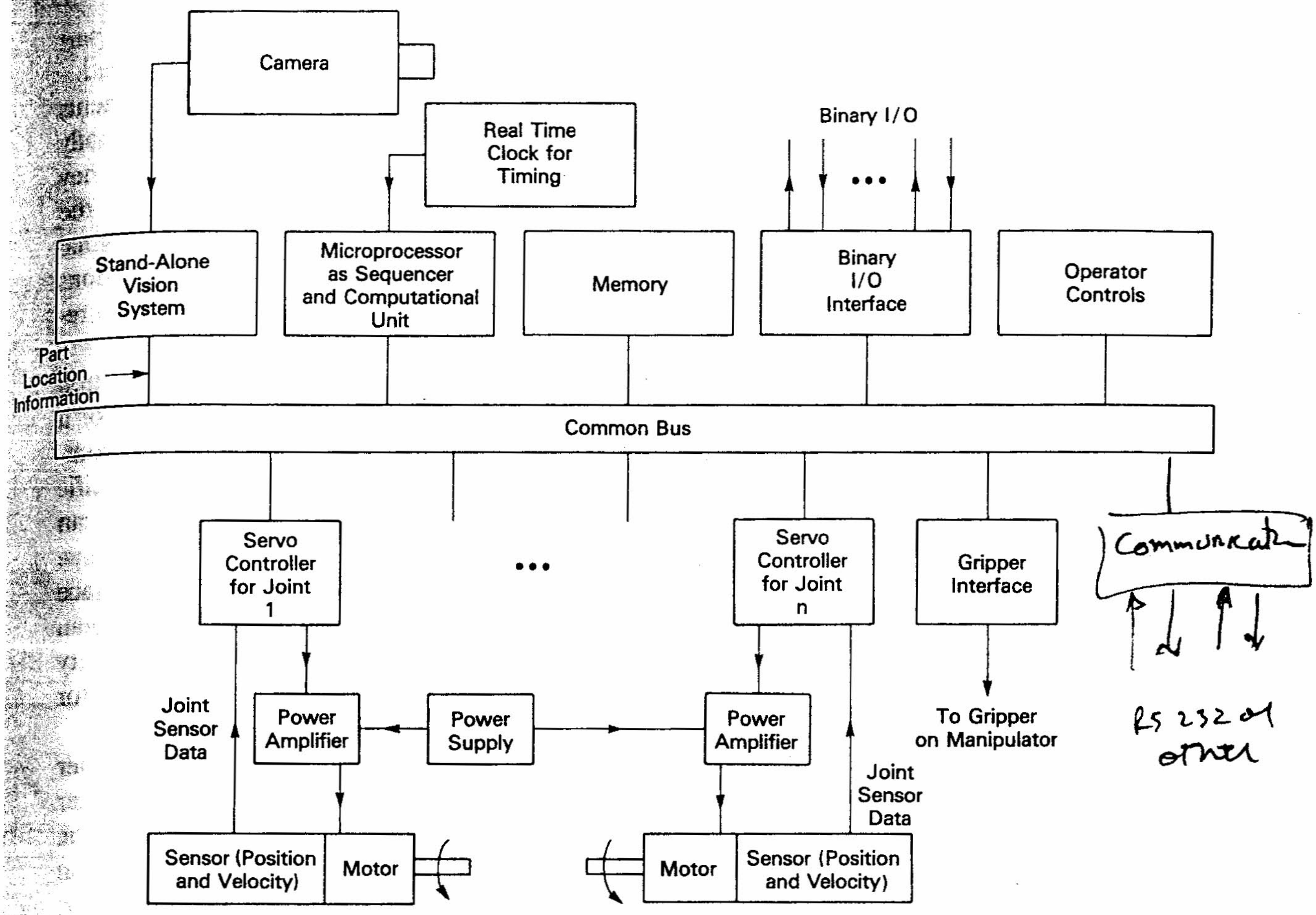

Figure 2.2.3. Possible implementation of a robot controller.

As opposed to the functionality approach just described, this organization is based on the physical packaging of the components and as a matter of fact, most industrial robots are packaged this way. Clearly, such a description is not as meaningful to the user in terms of the functionality of each subunit. However, it has the advantage of corresponding directly to the actual pieces of hardware.

2.3 THE ROBOT SYSTEM IN AN APPLICATION

By itself, a robot system has limited utility. Normally, it must be integrated with other components so that it can be programmed or trained to do some useful task. The term "workcell" is used to describe a collection of automated equipment and controls dedicated to performing one or more specific tasks. The workcell may contain several robots in addition to fixed automation devices (e.g., part feeders and conveyors), control devices (e.g., computers or programmable controllers), or

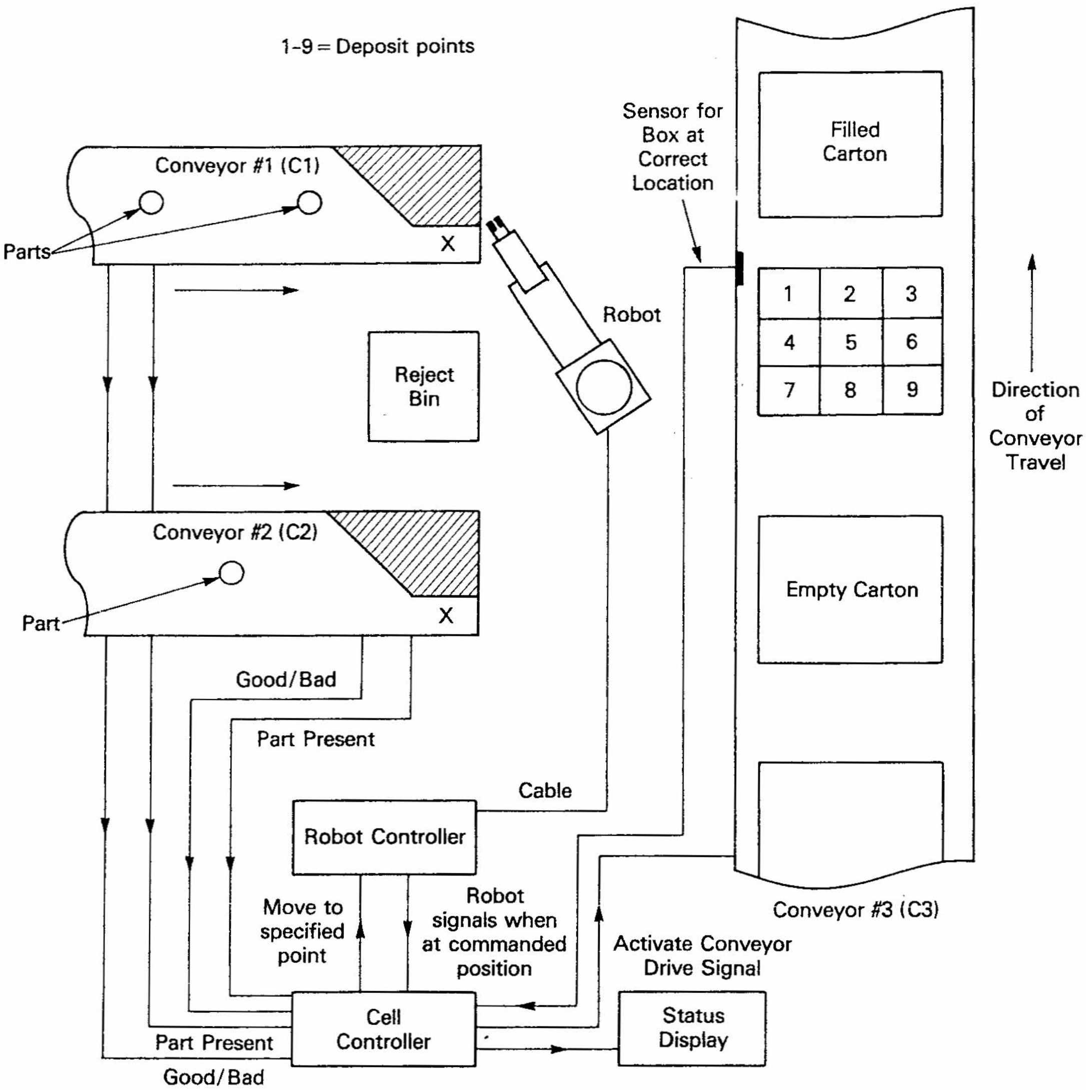

Figure 2.3.2. The robot as a peripheral device.

FUNCTIONS

FOR the training (programming) of the robot and its use with ancillary equipment. In addition, some features of the manipulator will also be noted. As will be seen, these features and capabilities have a visible impact on both the architecture and complexity of a robot system. In particular, the following capabilities are desirable:

Ability to Define Points (or Locations) in Space to Which the Robot Is to Go

- Demonstration (trainer moves the robot to a particular point and then "teaches" this location)
- Computation (controller computes offset distances from the current location to the new location, e.g., "move forward 3.6 inches")

- Interfacing to external sensors that define locations to which the robot should position itself; for example, in order to acquire a part (vision interface)
- Accepting off-line data (e.g., from a CAD/CAM system) which define points the robot is to move to as part of its program sequence

Ability to Move between Points in Various Ways

- Joint interpolated (all joints start and stop at the same time). This is sometimes called *coordinated motion*.
- Straight line (tool tip moves in a straight line while maintaining the same tool orientation).
- Continuous path (tool tip passes as close as possible to a series of taught or computed points while maintaining a constant velocity).
- Contouring (ability to draw circles or arcs, or move along a specified curve).
- Path profile specified whereby the acceleration, deceleration, and speed can be selected.

Program Control

- Delay before the next instruction is executed
- Ability to set, increment, or decrement counters or registers
- Ability to test the numeric condition of registers and branch to an instruction based on the result of the test
- Ability to display data (register values or positions)
- Ability to input data (for loop control, e.g., perform the same subroutine 20 times since 20 parts are present)
- Subroutine capability so that previously taught programs can be utilized in other larger programs

Control for the End Effector

- Command to open or close (simple pneumatic "bang-bang"-type gripper)
- Close or open a certain distance (position servo-controlled gripper)
- Close exerting a certain force (force servo-controlled gripper)

Provide Interfaces to Ancillary Equipment Such As Parts Feeders, Other Robots, and/or Vision Systems

- Interface via binary inputs and outputs so as to permit use of simple sensors, or turn-on or turn-off ancillary equipment. Typically, the interface should provide for various types of voltage levels (i.e., ac, dc, 24, 120, 240V, etc.). This interface may also be used to provide *handshaking** between the robot controller and the cell controller or even another robot.

---

*Handshaking is a technique whereby two or more systems exchange information to coordinate operations.

- Interface via serial link. This reduces the number of interconnects and permits communications to other devices using standard serial communications ports. Typically, this interface would be used to talk to a "host" computer. Various types of software protocols are necessary for maximum flexibility.

External Robot Control and Communications

- Remote specification of which program the controller is to run and an indication of when execution is completed. This provides the hook for the "cell controller" to tell the robot what to do.
- Remote control for safety features such as the provisions to halt the robot (regardless of what it is doing) or completely remove power from the unit.
- The ability to interface the robot with a factory network using a protocol such as GM's MAP (Manufacturing Automation Protocol) or SEMI's (Semiconductor Equipment and Materials Institute) SECS (Semiconductor Equipment Communication Standard). Protocols such as these are the basis of a standard for interconnecting various robots and other equipment to each other or to computers so that they may work in unison either as part of a workcell or within an entire factory.

Housekeeping Features —>SAFETY

- Being able to store, retrieve, and delete program and point data
- Having editing capabilities (such as inserting or deleting program steps, reteaching a point, etc.)
- Specifying which program is to run
- Providing statistics for use by the trainer, such as how much memory is required for a program, how much mass storage space is available, or the date on which a particular program was created

Program Debug and Simulation

- Debugging facilities (such as single step and back step)
- Ability to run a program but to ignore input signals and/or prevent outputs from occurring
- Ability to get a trace of which instructions or steps were executed
- Ability to set "breakpoints"
- Ability to check the state of the inputs without running a program
- Ability to set the state of outputs outside program control

System Parameters

- Reliable calibration method, so that when power is removed, the same joint positions can be obtained when the system is restarted. This implies a way of absolutely defining each joint angle.
- Ability to move back to a point (i.e., one previously shown or demonstrated

to the manipulator) within a certain error. A measure of this is called *repeatability*.

- Ability to move to a computed point (never before attained) within a certain absolute error. A measure of this is called accuracy.
- Ability to follow a curve or move in a straight line within some known error envelope.
- Definition of payload versus performance (i.e., maximum motion speed or acceleration with a certain payload).
- Definition of "settling time" versus speed and payload (e.g., how long it takes the loaded tool to reach the desired location with a specified maximum acceptable deviation about this point).

Serviceability

- Ease of maintenance of the manipulator (change of actuators, links, sensors, etc.)
- Run-time diagnostics (to monitor that the system is operating properly, pinpoint general errors, and immediately stop the robot if a detectable error occurs, such as the jamming of an axis or loss of some functionality of the controller)
- Invoked self-diagnostics (to pinpoint faulty components, subassemblies, or subsystems easily)

From this list we observe that a robot system can be quite complicated. Besides acting as a sequencer to guide the manipulator through some predefined activity, it may be sophisticated enough to be able to alter its program sequence or performance in real time as dictated by its environment. That is, the robot must have the capability to interface to the world around it, either through the use of simple binary sensors or by a more complicated scheme such as vision.

Unlike fixed automation, its sequence of operations may be altered by its trainer or programmer. Since the sequence can be modified, provisions must exist to make the changes relatively simply. One possible approach is the use of a robot-specific programming language (Chapter 7 discusses some robot programming languages). Another consideration, which stems from the "flexible automation" concept, is that since the robot system may be easily reprogrammed, the designers cannot anticipate all the possible permutations of how the device will be used or even what it will be asked to carry. Thus specifications must be given so that the applications engineer or user can successfully predict what a particular robot system can and cannot do without a costly trial-and-error procedure.

Not all the features identified above are implemented in every robot controller. Certain subsets may be chosen to make a simpler or more "cost-effective" implementation. Regardless of its features, the bottom line for the successful use of a robot depends on:

- Meeting performance specifications
- Reliability
- Maintainability

SAFETY/
CERTIFICATIONS

Reliability is sometimes measured as "mean time between failures" (MTBF), and maintainability is indicated by "mean time to repair" (MTTR). Manufacturers are faced with a trade-off among cost, sophistication, ease of use, salability, flexibility, and market demand. A robot must be a relatively cost-effective solution to a manufacturing problem; otherwise, its use may not be justifiable compared to fixed automation or human labor. A robot having the most innovative controller or programming language which is mechanically unreliable becomes nothing more than an expensive laboratory toy.