CHAPTER 5

Automation Technology

Chin-Yin Huang
Tunghai University, Taiwan

Shimon Y. Nof
Purdue University

1. INTRODUCTION **155**
2. PHYSICAL AUTOMATION TECHNOLOGY **156**
3. AUTOMATIC CONTROL SYSTEMS **156**
  3.1. Fundamentals of Control 157
  3.2. Instrumentation of an Automatic Control System 158
  3.3. Basic Control Models 159
    3.3.1. Control Modeling 159
    3.3.2. Control Models 160
  3.4. Advanced Control Models 160
4. TECHNOLOGIES OF ARTIFICIAL INTELLIGENCE **160**
  4.1. Knowledge-Based Systems 160
  4.2. Artificial Neural Networks 162
    4.2.1. Training Stage 163
    4.2.2 Control Stage 163
  4.3. Fuzzy Logic 163
  4.4. Genetic Algorithms 164
  4.5. Hybrid Intelligent Control Models 164
5. INTEGRATION TECHNOLOGY **164**
  5.1. Networking Technologies 165
  5.2. Object Orientation and Petri Net Techniques 166
  5.3 Distributed Control vs. Central Control 166
  5.4. Robot Simulator/Emulator 167
6. EMERGING TRENDS **167**
  6.1. Virtual Machines 168
  6.2. Tool Perspective Environment 168
    6.2.1. Facility Description Language (FDL) 171
    6.2.2 Concurrent Flexible Specifications (CFS) 172
  6.3. Agent-Based Control Systems 174
7. CONCLUSION **175**
ACKNOWLEDGMENTS **175**
REFERENCES **175**

1. INTRODUCTION

Accompanying the evolution of human society, tools have been developed to assist humans to perform all kinds of activities in humans' daily life. Tools not only reduce the effort that men have to put into those activities, they also help men perform activities that would otherwise be impossible due to the limitations of the human body—for example, using a telescope to see objects on the moon. However, humans are not satisfied with developing tools to enhance productivity or conquer their limitations. People have always dreamed of building machines to do their work for them, allowing them more leisure time. This dream can be realized through the help of modern computer and communication technologies. Activities in the manufacturing enterprises that are automated by computer and communication technologies can be summarized into (but not limited by) the following seven categories (Huang and Nof 1999):

1. *Design:* Powerful computation speeds up activities in enterprises. For example, design activities are improved because of powerful CAD workstations.

Handbook of Industrial Engineering: Technology and Operations Management, Third Edition.
Edited by Gavriel Salvendy

2. *Decisions:* Powerful computation allows many simulation trials to find a better solution in decision making. For example, an optimal material handling equipment selection can be obtained through repeated simulation runs.
3. *Sensing:* Input devices (e.g., sensors, bar code readers) can gather and communicate environmental information to computers or humans. A decision may be made by computer systems based on the input information. The decision may also trigger output devices (e.g., robot arms, monitors) to realize the decisions.
4. *Recovery:* Computer systems may apply techniques of artificial intelligence (AI) (e.g., fuzzy rules, knowledge-based logic, neural networks) to improve the quality of activities. For example, a robot system may be recovered automatically from error conditions through decisions made by AI programs.
5. *Collaboration:* Distributed designers can work together on a common design project through a computer supported collaborative work (CSCW) software system.
6. *Partners:* A computer system in an organization may automatically find cooperative partners (e.g., vendors, suppliers, and subcontractors) from the Internet to fulfill a special customer order without any increase in the organization's capacity.
7. *Logistics:* Logistics flows of products and packages are monitored and maintained by networked computers.

Although these seven categories reflect the impact of computer and communication technologies, they are driven by four automation technologies: physical automation systems, automatic control systems, artificial intelligence systems, and integration technology. Physical automation systems and automatic control systems represent two early and ongoing achievements in automation technology. Through automatic control theories, most systems can be controlled by the set points defined by users. With the support of both automatic control theories and modern digital control equipment, such as the programmable logic controller (PLC), physical automation systems that consist of processing machines (e.g., CNCs), transportation equipment (e.g., robots and AGVs), sensing equipment (e.g., bar code readers) can be synchronized and integrated.

Artificial intelligence systems and integration technology are two relatively recent technologies. Many AI techniques, such as artificial neural networks, knowledge-based systems, and genetic algorithms, have been applied to automate the complex decision making processes in design, planning, and managerial activities of enterprises. Additionally, integration techniques, such as electronic data interchange (EDI), client-server systems, and Internet-based transactions, have automated business processes even when the participants are in remote sites.

In this chapter, we will discuss the above four technologies to give readers comprehensive knowledge of automation technology. Section 2 addresses physical automation technologies that are applied in the processing, transportation, and inspection activities. Section 3 introduces classical automatic control theory. The purpose of addressing automatic control theory is to review the traditional methods and explain how a system can be automatically adjusted to the set point given by users. Section 4 addresses artificial intelligence techniques and introduces basic application approaches. Section 5 introduces integration technology, which is based mostly on today's information technology. Section 6 introduces the emerging trends of automation technologies, which include virtual machines, tool perspective environment, and autonomous agents. Section 7 makes some concluding remarks.

2. PHYSICAL AUTOMATION TECHNOLOGY

Since MIT demonstrated the first numerically controlled machine tool in 1952, information technologies have revolutionized and automated the manufacturing processes. See Chapter 12 for physical automation techniques such as robots. In general, physical automation technology can be applied in three areas: processing, transportation/storage, and inspection. Representative examples of automated equipment are:

1. *Automated processing equipment:* CNC machine tools, computer-controlled plastic-injection machines, etc.
2. *Automated transportation/storage equipment:* Industrial robots, automatic guided vehicles (AGV), an automatic storage/retrieval systems (AS/RS), etc.
3. *Automated inspection equipment:* Coordination measuring machines (CMM), machine vision systems, etc.

3. AUTOMATIC CONTROL SYSTEMS

Control is the fundamental engineering and managerial function whose major purpose is to measure, evaluate, and adjust the operation of a process, a machine, or a system under dynamic conditions so that it achieves desired objectives within its planned specifications and subject to cost and safety considerations. A well-planned system can perform effectively without any control only as long as no variations are encountered in its own operation and its environment. In reality, however, many changes occur over time. Machine breakdown, human error, variable material properties, and faulty information are a few examples of why a system must be controlled.

When a system is more complex and there are more potential sources of dynamic variations, a more complicated control is required. Particularly in automatic systems where human operators are replaced by machines and computers, a thorough design of control responsibilities and procedures is necessary. Control activities include automatic control of individual machines, material handling, equipment, manufacturing processes, and production systems, as well as control of operations, inventory, quality, labor performance, and cost. Careful design of correct and adequate controls that continually identify and trace variations and disturbances, evaluate alternative responses, and result in timely and appropriate actions is therefore vital to the successful operation of a system.

3.1. Fundamentals of Control

Automatic control, as the term is commonly used, is "self-correcting," or feedback, control; that is, some control instrument is continuously monitoring certain output variables of a controlled process and is comparing this output with some preestablished desired value. The instrument then compares the actual and desired values of the output variable. Any resulting error obtained from this comparison is used to compute the required correction to the control setting of the equipment being controlled. As a result, the value of the output variable will be adjusted to its desired level and maintained there. This type of control is known as a servomechanism.

The design and use of a servomechanism control system requires a knowledge of every element of the control loop. For example, in Figure 1 the engineer must know the dynamic response, or complete operating characteristics, of each pictured device:

1. The indicator or sampler, which senses and measures the actual output
2. The controller, including both the error detector and the correction computer, which contain the decision making logic
3. The control value and the transmission characteristics of the connecting lines, which communicate and activate the necessary adjustment
4. The operating characteristics of the plant, which is the process or system being controlled

Dynamic response, or *operating characteristics,* refer to a mathematical expression, for example, differential equations, for the transient behavior of the process or its actions during periods of change in operating conditions. From it one can develop the transfer function of the process or prepare an experimental or empirical representation of the same effects.

Because of time lags due to the long communication line (typically pneumatic or hydraulic) from sensor to controller and other delays in the process, some time will elapse before knowledge of changes in an output process variable reaches the controller. When the controller notes a change, it must compare it with the variable value it desires, compute how much and in what direction the control valve must be repositioned, and then activate this correction in the valve opening. Some time is required, of course, to make these decisions and correct the valve position.

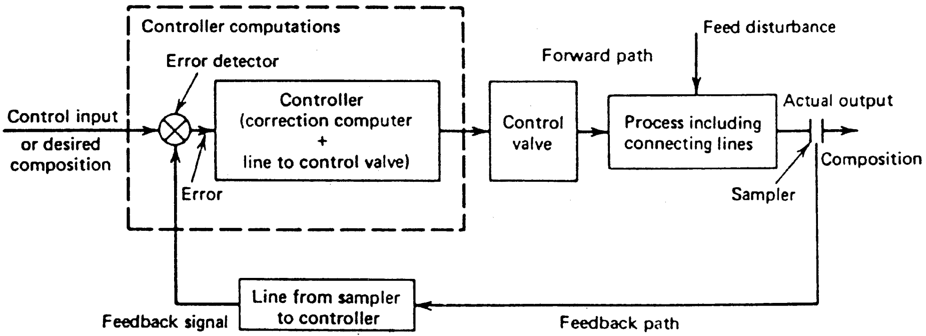


Figure 1 Block Diagram of a Typical Simple, Single Control Loop of a Process Control System. (From Williams and Nof 1992)

Some time will also elapse before the effect of the valve correction on the output variable value can reach the output itself and thus be sensed. Only then will the controller be able to know whether its first correction was too small or too large. At that time it makes a further correction, which will, after a time, cause another output change. The results of this second correction will be observed, a third correction will be made, and so on.

This series of measuring, comparing, computing, and correcting actions will go around and around through the controller and through the process in a closed chain of actions until the actual process valve is finally balanced again at the desired level by the operator. Because from time to time there are disturbances and modifications in the desired level of the output, the series of control actions never ceases. This type of control is aptly termed *feedback control.* Figure 1 shows the direction and path of this closed series of control actions. The closed-loop concept is fundamental to a full understanding of automatic control.

Although the preceding example illustrates the basic principles involved, the actual attainment of automatic control of almost any industrial process or other complicated device will usually be much more difficult because of the speed of response, multivariable interaction, nonlinearity, response limitations, or other difficulties that may be present, as well as the much higher accuracy or degree of control that is usually desired beyond that required for the simple process just mentioned.

As defined here, automatic process control always implies the use of a feedback. This means that the control instrument is continuously monitoring certain output variables of the controlled process, such as a temperature, a pressure, or a composition, and is also comparing this output with some preestablished desired value, which is considered a reference, or a set point, of the controlled variable. An error that is indicated by the comparison is used by the instrument to compute a correction to the setting of the process control valve or other final control element in order to adjust the value of the output variable to its desired level and maintain it there.

If the set point is altered, the response of the control system to bring the process to the new operating level is termed that of a servomechanism or self-correcting device. The action of holding the process at a previously established level of operation in the face of external disturbances operating on the process is termed that of a regulator.

3.2. Instrumentation of an Automatic Control System

The large number of variables of a typical industrial plant constitute a wide variety of flows, levels, temperatures, compositions, positions, and other parameters to be measured by the sensor elements of the control system. Such devices sense some physical, electrical, or other informational property of the variable under consideration and use it to develop an electrical, mechanical, or pneumatic signal representative of the magnitude of the variable in question. The signal is then acted upon by a transducer to convert it to one of the standard signal levels used in industrial plants (3–15 psi for pneumatic systems and 1–4, 4–20, or 10–50 mA or 0–5 V for electrical systems). Signals may also be digitized at this point if the control system is to be digital.

The signals that are developed by many types of sensors are continuous representations of the sensed variables and as such are called analog signals. When analog signals have been operated upon by an analog-to-digital converter, they become a series of bits, or on–off signals, and are then called digital signals. Several bits must always be considered together in order to represent properly the converted analog signal (typically, 10–12 bits).

As stated previously, the resulting sensed variable signal is compared at the controller to a desired level, or set point, for that variable. The set point is established by the plant operator or by an upper-level control system. Any error (difference) between these values is used by the controller to compute the correction to the controller output, which is transmitted to the valve or other actuator of the system's parameters.

A typical algorithm by which the controller computes its correction is as follows (Morriss 1995). Suppose a system includes components that convert inputs to output according to relationships, called gains, of three types: proportional, derivative, and integral gains. Then the controller output is

$$\text{Output} = K_P e(t) + K_I \int e(t)\, dt + K_D \frac{d(e(t))}{dt}$$

where K_P, K_D, and K_I = proportional, derivative, and integral gains, respectively, of the controller.

The error at time t, $e(t)$, is calculated as

$$e(t) = \text{set point} - \text{feedback signal}$$

3.3. Basic Control Models

3.3.1. Control Modeling

Five types of modeling methodologies have been employed to represent physical components and relationships in the study of control systems:

1. Mathematical equations, in particular, differential equations, which are the basis of classical control theory (transfer functions are a common form of these equations)
2. Mathematical equations that are used on state variables of multivariable systems and associated with modern control theory
3. Block diagrams
4. Signal flow graphs
5. Functional analysis representations (data flow diagram and entity relationships)

Mathematical models are employed when detailed relationships are necessary. To simplify the analysis of mathematical equations, we usually approximate them by linear, ordinary differential equations. For instance, a characteristic differential equation of a control loop model may have the form

$$\frac{d^2x}{dt^2} + 2\alpha\frac{dx}{dt} + \beta^2 x = f(t)$$

with initial conditions of the system given as

$$x(0) = X_0$$

$$x'(0) = V_0$$

where $x(t)$ is a time function of the controlled output variable, its first and second derivatives over time specify the temporal nature of the system, α and β are parameters of the system properties, $f(t)$ specifies the input function, and X_0 and V_0 are specified constants.

Mathematical equations such as this example are developed to describe the performance of a given system. Usually an equation or a transfer function is determined for each system component. Then a model is formulated by appropriately combining the individual components. This process is often simplified by applying Laplace and Fourier transforms. A graph representation by block diagrams (see Figure 2) is usually applied to define the connections between components.

Once a mathematical model is formulated, the control system characteristics can be analytically or empirically determined. The basic characteristics that are the object of the control system design are:

1. Response time
2. Relative stability
3. Control accuracy

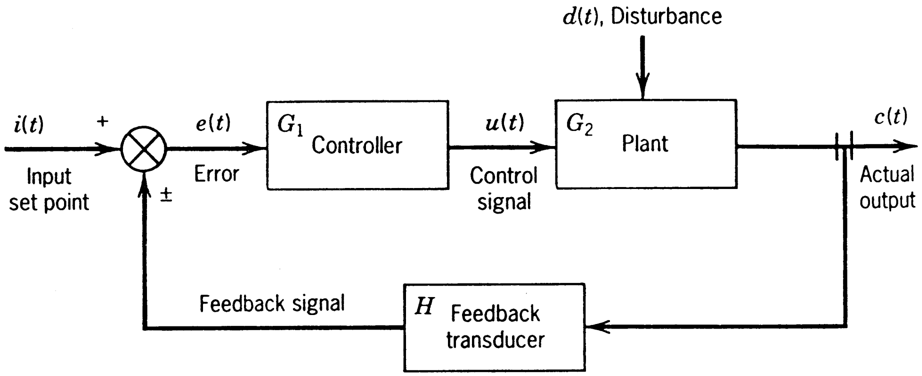


Figure 2 Block Diagram of a Feedback Loop. (From Williams and Nof 1992)

They can be expressed either as functions of frequency, called frequency domain specifications, or as functions of time, called time domain specifications. To develop the specifications the mathematical equations have to be solved. Modern computer software, such as MATLAB (e.g., Kuo 1995), has provided convenient tools for solving the equations.

3.3.2. *Control Models*

Unlike the open-loop control, which basically provides a transfer function for the input signals to actuators, the feedback control systems receive feedback signals from sensors then compare the signals with the set point. The controller can then control the plant to the desired set point according to the feedback signal. There are five basic feedback control models (Morriss 1995):

1. *On/off control:* In on/off control, if the $e(t)$ is smaller than 0, the controller may activate the plant; otherwise the controller stays still. Most household temperature thermostats follow this model.
2. *Proportional (PE) control:* In PE control, the output is proportional to the $e(t)$ value, i.e., $e(t) = K_P e(t)$. In PE, the plant responds as soon as the error signal is non-zero. The output will not stop exactly at the set point. When it approaches the set point, the $e(t)$ becomes smaller. Eventually, the output is too small to overcome opposing force (e.g., friction). Attempts to reduce this small $e(t)$, also called steady state error, by increasing K_P can only cause more overshoot error.
3. *Proportional-integral (PI) control:* PI control tries to solve the problem of steady state error. In PI, output $= K_P e(t) + K_I \int e(t)\, dt$. The integral of the error signal will have grown to a certain value and will continue to grow as soon as the steady state error exists. The plant can thus be drawn to close the steady state error.
4. *Proportional-derivative (PD) control:* PD control modifies the rate of response of the feedback control system in order to prevent overshoot. In PD,

$$\text{Output} = K_P e(t) + K_D \frac{d(e(t))}{dt}$$

   When the $e(t)$ gets smaller, a negative derivative results. Therefore, overshoot is prevented.
5. *Proportional-integral-derivative (PID) control:* PID control takes advantage of PE, PI, and PD controls by finding the gains (K_P, K_I, and K_D) to balance the proportional response, steady state reset ability, and rate of response control, so the plant can be well controlled.

3.4. Advanced Control Models

Based on the control models introduced in Section 3.3, researchers have developed various advanced control models for special needs. Table 1 shows the application domains and examples of the models, rather than the complicated theoretical control diagram. Interested readers can refer to Morriss (1995).

4. TECHNOLOGIES OF ARTIFICIAL INTELLIGENCE

Automation technologies have been bestowed intelligence by the invention of computers and the evolution of artificial intelligence theories. Because of the introduction of the technologies of artificial intelligence, automated systems, from the perspective of control, can intelligently plan, actuate, and control their operations in a reasonable time limit by handling/sensing much more environmental input information (see the horizontal axis in Figure 3). Meanwhile, artificial intelligence increases the decision making complexity of automation technology, but the cost of the system that is automated by the technologies of artificial intelligence is relatively low compared with the system automated by the automatic control theory (see the vertical axis in Figure 3). Figure 3 shows a roadmap of how various automation technologies influence the development of automation systems in two axes. In the following subsections, those technologies of artificial intelligence, including neural networks (NN), genetic algorithms (GA), knowledge-based systems (KBS), fuzzy control, and the hybrid of the above-mentioned technologies will be introduced in general.

4.1. Knowledge-Based Systems

The structure of a knowledge-based system (KBS) in control is presented in Figure 4. The control decisions are achieved by reasoning techniques (inference engine) rather than quantitative computation and can deal with uncertainties and unstructured situations. The knowledge base is updated by continuously acquiring the knowledge from experts, the decisions made by the operator and the feedback from the process. Nowadays, KBSs have been applied in many fields to diagnose causes of problems, such as in medical diagnosis, vehicle troubleshooting, and loan strategy planning in banks.

TABLE 1 System Characteristics and Examples of Advanced Control Models

Control Models	When to Apply (System Characteristics)	Examples
(1) Nested Control Loops	More than one characteristic of a system's output to be controlled at the same time	Robot controllers
(2) Directed Synthesis Control	Time lag between output and feedback reception is long	Satellite guidance systems
(3) Adaptive Directed Synthesis Control	Allow a direct synthesis control system to be adjusted when actual feedback is received	Modern machine tools
(4) Feedforward Control; Cascade Control	Set points are given by the sensed conditions upstream of the process.	Chemical process control
(5) Ratio Control	Use a sensor in one stream to control the processes in one or more parallel streams	Chemical process control
(6) Multiple Output Control	A single control provides the output signal to a group of parallel actuators. Then the result of the multiple actuation is fed back to the controller.	Complex hydraulic cylinders
(7) Constraint Control	More than one controller in the system. Under normal conditions, one of the controllers does the controlling, but if a preset limit is exceeded at a sensor, the second controller overrides the first and takes over control.	Chemical process control

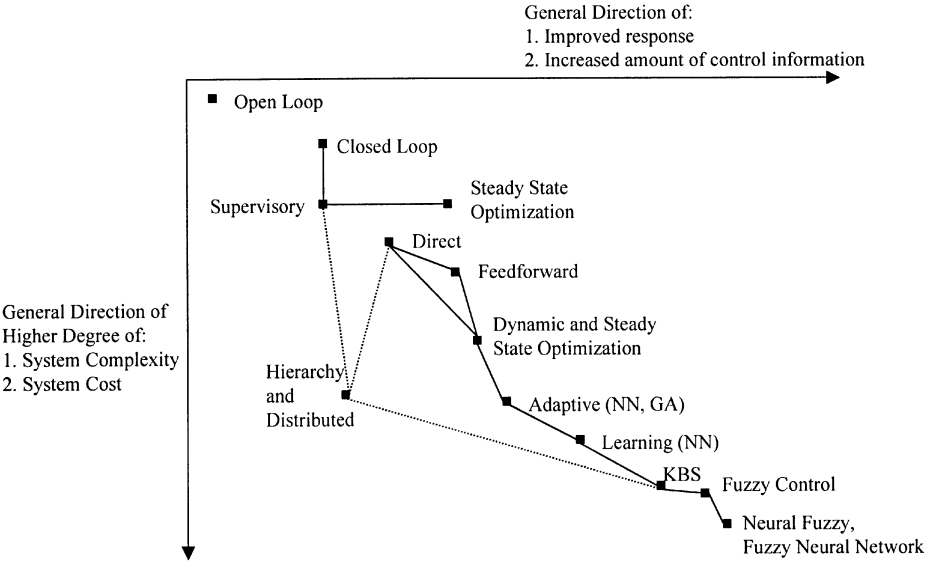


Figure 3 Roadmap of Control Models. (Revised from Williams and Nof 1992)

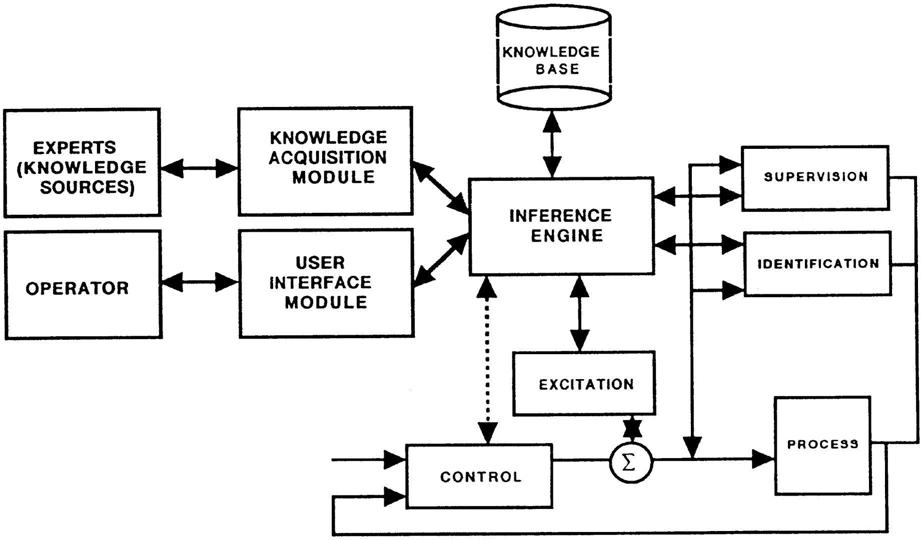


Figure 4 Block Diagram of a Knowledge-Based Control System. (From Williams and Nof 1992)

The numerous industrial engineering applications of control models in computer information systems can be classified into two types: (1) development of decision support systems, information systems that provide the information, and decisions to control operations, and (2) maintenance of internal control over the quality and security of the information itself. Because information systems are usually complex, graphic models are typically used.

Any of the control models can essentially incorporate an information system, as indicated in some of the examples given. The purpose of an information system is to provide useful, high-quality information; therefore, it can be used for sound planning of operations and preparation of realistic standards of performance. Gathering, classifying, sorting, and analyzing large amounts of data can provide timely and accurate measurement of actual performance. This can be compared to reference information and standards that are also stored in the information system in order to immediately establish discrepancies and initiate corrective actions. Thus, an information system can improve the control operation in all its major functions by measuring and collecting actual performance measures, analyzing and comparing the actual to the desired set points, and directing or actuating corrective adjustments.

An increasing number of knowledge-based decision support and control systems have been applied since the mid-1980s. Typical control functions that have been implemented are:

- Scheduling
- Diagnosis
- Alarm interpretation
- Process control
- Planning
- Monitoring

4.2. Artificial Neural Networks

The powerful reasoning and inference capabilities of artificial neural networks (ANN) in control are demonstrated in the areas of

- Adaptive control and learning
- Pattern recognition/classification
- Prediction

To apply ANN in control, the user should first answer the following questions:

1. If there are training data, ANN paradigm with supervised learning may be applied; otherwise, ANN paradigm with unsupervised learning is applied.
2. Select a suitable paradigm, number of network layers, and number of neurons in each layer.
3. Determine the initial weights and parameters for the ANN paradigm.

A widely applied inference paradigm, back propagation (BP), is useful with ANN in control. There are two stages in applying BP to control: the training stage and the control stage (Figure 5).

4.2.1. Training Stage

1. Prepare a set of training data. The training data consist of many pairs of data in the format of input–output.
2. Determine the number of layers, number of neurons in each layer, the initial weights between the neurons, and parameters.
3. Input the training data to the untrained ANN.
4. After the training, the trained ANN provides the associative memory for linking inputs and outputs.

4.2.2. Control Stage

Input new data to the trained ANN to obtain the control decision. For instance, given a currently observed set of physical parameter values, such as noise level and vibration measured on a machine tool, automatically adapt to a new calculated motor speed. The recommended adjustment is based on the previous training of the ANN-based control. When the controller continues to update its training ANN over time, we have what is called learning control.

Other application examples of neural networks in automated systems are as follows:

- Object recognition based on robot vision
- Manufacturing scheduling
- Chemical process control

The ANN could be trained while it is transforming the inputs on line to adapt itself to the environment. Detailed knowledge regarding the architectures of ANN, initial weights, and parameters can be found in Dayhoff (1990), Freeman and Skapura (1991), Fuller (1995), Lin and Lee (1996).

4.3. Fuzzy Logic

Figure 6 shows a structure of applying fuzzy logic in control. First, two types of inputs must be obtained: numerical inputs and human knowledge or rule extraction from data (i.e., fuzzy rules). Then the numerical inputs must be fuzzified into fuzzy numbers. The fuzzy rules consist of the fuzzy membership functions (knowledge model) or so-called fuzzy associative memories (FAMs). Then the

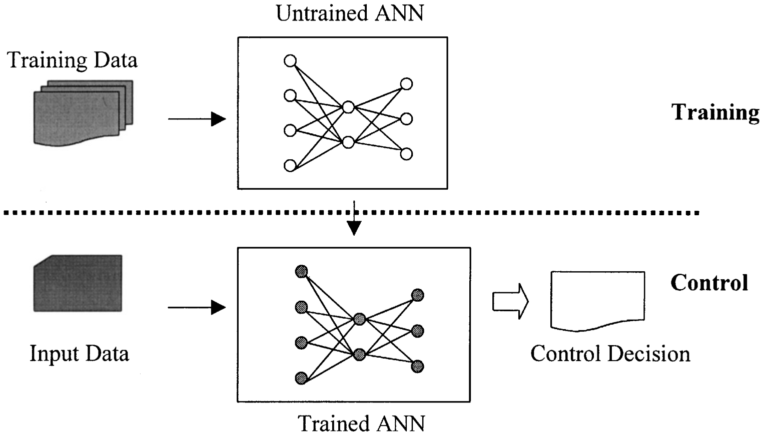


Figure 5 Structure of Applying Back-Propagation ANN in Control.

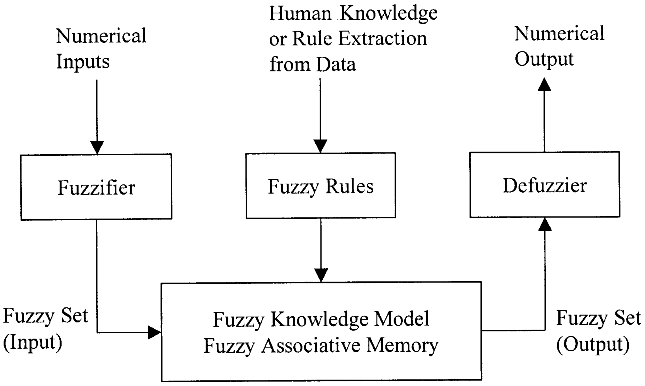


Figure 6 The Structure of Applying Fuzzy Logic in Control.

FAMs map fuzzy sets (inputs) to fuzzy sets (outputs). The output fuzzy sets should be defuzzified into numerical values to control the plant. Due to the powerful ability of fuzzy sets in describing system linguistic and qualitative behavior and imprecise and/or uncertain information, many industrial process behavior and control laws can be modeled by fuzzy logic-based approaches. Fuzzy logic has been applied in a wide range of automated systems, including:

- Chemical process control
- Autofocusing mechanism on camera and camcorder lens
- Temperature and humidity control for buildings, processes, and machines

4.4. Genetic Algorithms

Genetic algorithms (GAs), also referred to as evolutionary computation, are highly suitable for certain types of problems in the areas of optimization, product design, and monitoring of industrial systems. A GA is an automatically improving (evolution) algorithm. First the user must encode solutions of a problem into the form of chromosomes and an evaluation function that would return a measurement of the cost value of any chromosome in the context of the problem. A GA consists of the following steps:

1. Establish a base population of chromosomes.
2. Determine the fitness value of each chromosome.
3. Create new chromosomes by mating current chromosomes; apply mutation and recombination as the parent chromosomes mate.
4. Delete undesirable members of the population.
5. Insert the new chromosomes into the population to form a new population pool.

GA are useful for solving large-scale planning and control problems. Several cases indicate that GA can effectively find an acceptable solution for complex product design, production scheduling, and plant layout planning.

4.5. Hybrid Intelligent Control Models

Intelligent control may be designed in a format combining the techniques introduced above. For example, fuzzy neural networks use computed learning and the adaptive capability of neural networks to improve the computed learning's associative memory. Genetic algorithms can also be applied to find the optimal structure and parameters for neural networks and the membership functions for fuzzy logic systems. In addition, some techniques may be applicable in more than one area. For example, the techniques of knowledge acquisition in KBSs and fuzzy logic systems are similar.

5. INTEGRATION TECHNOLOGY

Recent communication technologies have enabled another revolution in automation technologies. Stand-alone automated systems are integrated via communication technologies. Integration can be identified into three categories:

1. *Networking:* Ability to communicate.
2. *Coordinating:* Ability to synchronize the processes of distributed automated systems. The coordination is realized by either a controller or an arbitrary rule (protocol).
3. *Integration:* Distributed automated systems are able to cooperate or collaborate with other automated systems to fulfill a global goal while satisfying their individual goals. The cooperation/collaboration is normally realized via a protocol that is agreed upon by distributed automated systems. However, the agreement is formed through the intelligence of the automated systems in protocol selection and adjustment.

In this section, automated technologies are introduced based on the above categories. However, truly integrated systems (technologies) are still under development and are mostly designed as agent-based systems, described in Section 6. Only technologies from the first two categories are introduced in this section.

5.1. Networking Technologies

To connect automated systems, the simplest and the most economic method is to wire the switches on the automated systems to the I/O modules of the programmable logic controller (PLC). Ladder diagrams that represent the control logic on the automated systems are popularly applied in PLCs. However, as the automated systems are remotely distributed, automated systems become more intelligent and diversified in the communication standards that they are built in, and the coordination decisions become more complex, the simple messages handled by the PLC-based network are obviously not enough for the control of the automated systems. Hence, some fundamental technologies, such as field bus (Mahalik and Moore 1997) and local area networks (LANs), and some advanced communication standards, such as LonWorks, Profibus, Manufacturing Automation Protocol (MAP), Communications Network for Manufacturing Applications (CNMA), and SEMI* Equipment Communication Standard (SECS) are developed.

In a field bus, automated devices are interconnected. Usually the amount of data transmitted in the bus is not large. In order to deliver message among equipment's timely in a field bus, the seven layers of the open system interconnection (OSI) are simplified into three layers: *physical layer, data link layer,* and *application layer.* Unlike a field bus, which usually handles the connections among devices, office activities are automated and connected by a LAN. Usually more than one file server is connected in a LAN for data storage, retrieval, and sharing among the connected personal computers. Three technological issues have to be designed/defined in a LAN topology, media, and access methods (Cohen and Apte 1997).

For the advanced communication standard, MAP and CNMA are two technologies that are well known and widely applied. Both technologies are based on the Manufacturing Message Specification (MMS) (SISCO 1995), which was developed by the ISO Industrial Automation Technical Committee Number 184. MMS provides a definition to specify automated equipment's external behaviors (Shanmugham et al. 1995). Through the specification, users can control the automated equipment with little knowledge about the internal message conversion within MMS and the equipment.

SECS is currently a popular communication standard in semiconductor industries (SEMI 1997). The standard was developed by SEMI based on two standards, SECS-I (SEMI Equipment Communications Standard Part 1 Message Transfer) and SECS-II (SEMI Equipment Communication Standard 2 Message Content). The relationship between SECS-I and SECS-II, as shown in Figure 7, shows that SECS-I transmits the message that is defined by SECS-II to RS-232. SECS-II's message

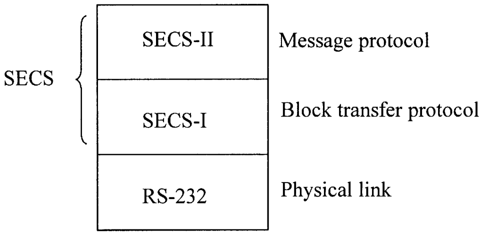


Figure 7 Basic Structure of SECS.

---

* SEMI stands for Semiconductor Equipment and Materials International.

is added as control information by the SECS-I so the transmission message can conform the format of RS-232 for message delivery. For SECS-II, it provides a set of interequipment communication standards under various situations. Hence, engineers only need to know and follow the SECS-II to control the connected automated equipment, rather than taking time to define the detailed message conversion.

5.2. Object Orientation and Petri Net Techniques

Object orientation and Petri net are automation techniques in modeling and analysis levels. Automated systems and their associated information and resources can be modeled by object models. The coordination and communication among the automated systems can then be unified with the message passing among the objects. However, the complex message passing that is used to coordinate behaviors of the automated systems relies on the technique of Petri net. The graphical and mathematically analyzable characteristics make Petri net a very suitable tool for synchronizing the behaviors and preventing deadlocks among automated systems. Combinations of both techniques have been developed and applied in the controllers of flexible manufacturing systems (Wang 1996; Wang and Wu 1998).

5.3. Distributed Control vs. Central Control

The rapid development of microprocessor technology has made distributed control possible and attractive. The use of reliable communication between a large number of individual controllers, each responsible for its own tasks rather than for the complete operation, improves the response of the total system. We can take PLC-based control as a typical example of central control system and LonWorks, developed by Echelon, as an example of distributed control system. In LonWorks, each automated device is controlled by a control module—LTM-10 (Figure 8). The control modules are connected on a LonTalk network that provides an ISO/OSI compatible protocol for communication.

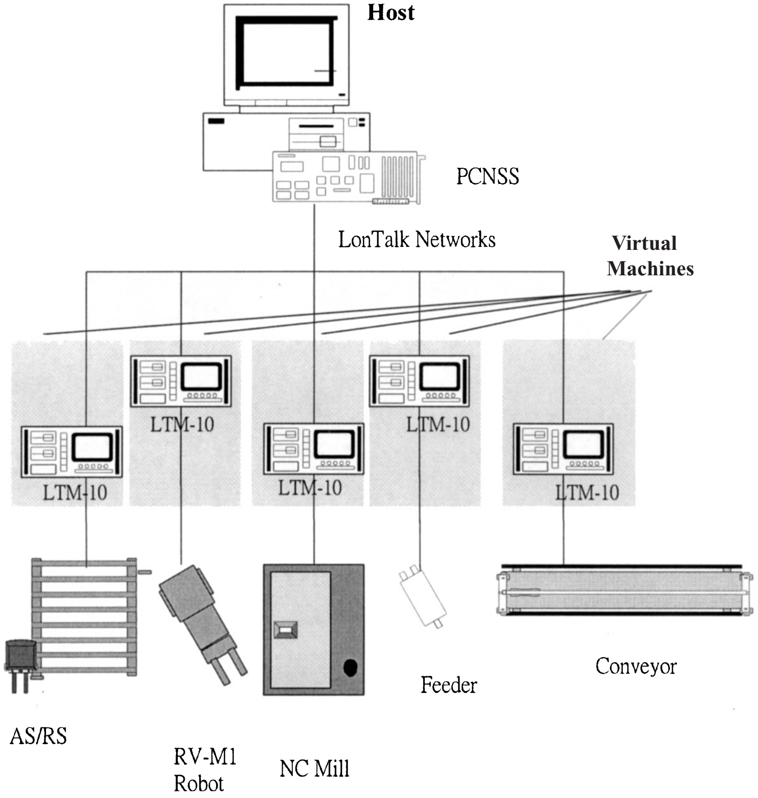


Figure 8 Applying LonWorks to Develop Virtual Machines.

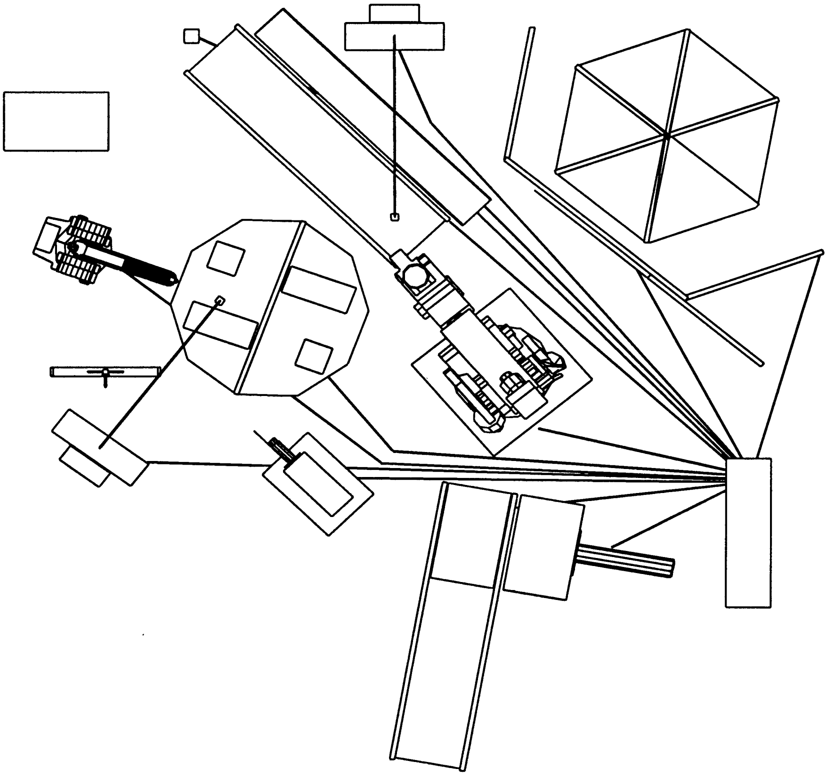


Figure 9 An Example of a Robot Emulator. (From Witzerman and Nof 1995)

Usually the control modules are loaded with Neuron C programs from a host computer that has a PCNSS network card for network management and Neuron C programming. Hence, under running mode the host computer is not necessary and the control modules can work under a purely distributed environment.

Distributed automated systems have the following advantages over centralized automated systems:

- Higher system reliability
- Better response to local demands
- Lower cost in revising the system control programs when automated equipment is added or deleted from the system

5.4. Robot Simulator/Emulator

In recent years, powerful robot simulators/emulators have been developed by several companies (Nof 1999). Examples include ROBCAD by Tecnomatix and RAPID by Adept Technologies. With these highly interactive, graphic software, one can program, model, and analyze both the robots and their integration into a production facility. Furthermore, with the geometric robot models, the emulation can also check for physical reachability and identify potential collisions. Another important feature of the robot simulators/emulators is off-line programming of robotic equipment, which allows designers to compare alternative virtual design before the program is transferred to actual robots (see Figure 9).

6. EMERGING TRENDS

Automation technology has reached a new era. An automation system is automated not only to reach the setting point but to situate itself intelligently in its complex environment. The individual automated systems are also networked to accomplish collaborative tasks. However, networking automated systems is not an easy task. It involves the technologies from the physical levels, which handle the interface and message conversion between automation systems, to the application level, which handles mostly the social interactions between the automation systems. Three typical ongoing research directions are described next to present the trends of automation technology.

6.1. Virtual Machines

Traditionally, a hierarchy structure of the computer control system for a fully automated manufacturing system can be divided into four levels (Figure 10). The supervisory control is responsible for managing the direct digit control by sending control commands, downloading process data, monitoring the process, and handling exception events. From the perspective of message transmission, the hierarchy in Figure 10 can be classified into three standard levels, as shown in Figure 11:

1. *Production message standard:* a standard for obtaining/sending production information from /to low-level computers. The production information can be a routing, a real-time statistics, etc. Usually the information is not directly related to the equipment control.
2. *Control message standard:* a standard for controlling the equipment logically. The control message includes commands, the parameters associated with the command, and data. SECS-II is a standard that can fulfill such a need (Elnakhal and Rzehak 1993).

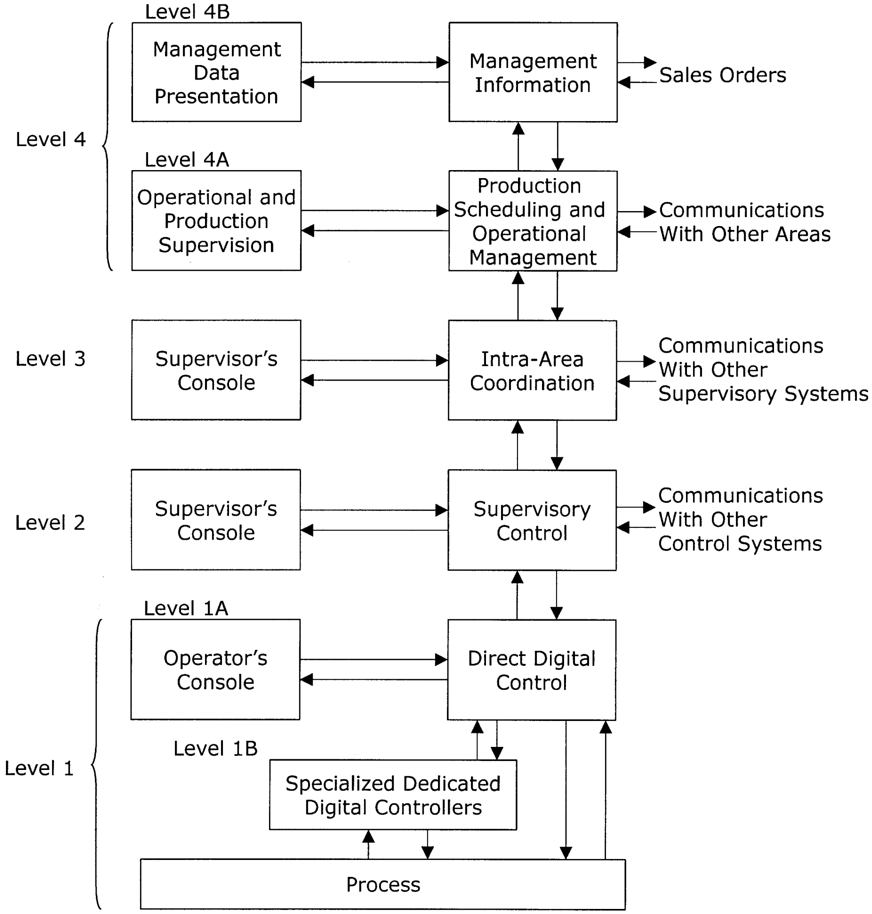


Figure 10 The Hierarchy Structure of the Computer Control System for a Fully Automated Industrial Plant. (From Williams and Nof 1992)

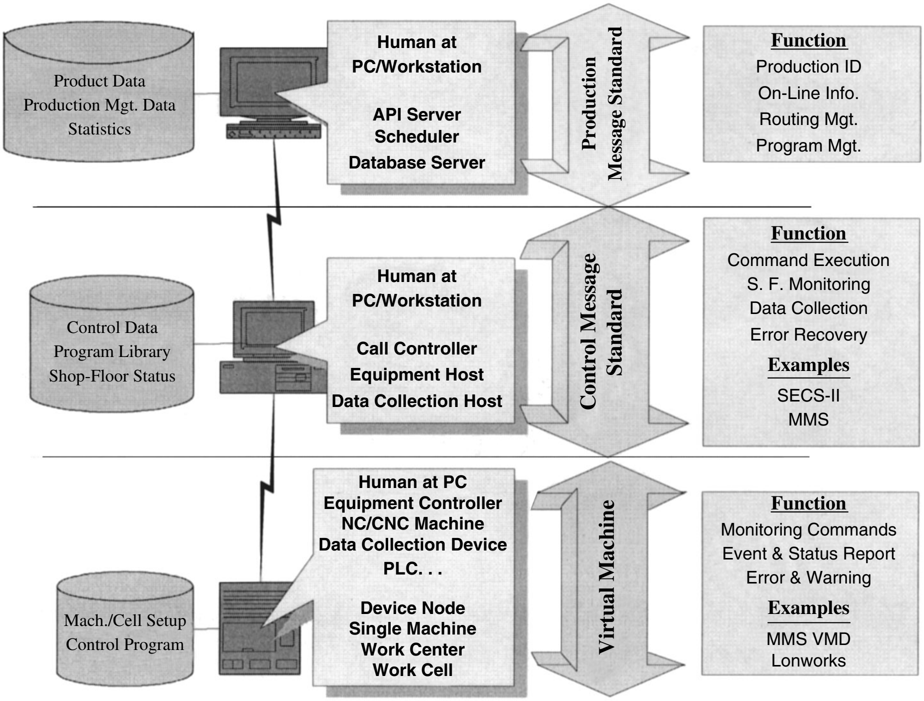


Figure 11 Three Layers of Message Transmission.

3. *Virtual machine:* a mechanism that converts the logical commands based on the control message standard, e.g., SECS-II, to commands format that can be accepted by a physical machine. Figure 12 shows that a virtual machine is laid between a host that delivers SECS-II commands and a real device that receives and executes the commands in its format. A virtual machine is therefore responsible for bridging the message format gap.

The development of virtual machines reduces the problems of incompatibility in an automated system, especially when the automated equipment follows different communication standards and protocols. It also enables the high-level controller to focus on his or her control activities, such as sequencing and scheduling, rather than detailed command incompatibility problems. Further information on virtual machines and automation can be found in Burdea and Coiffet (1999).

6.2. Tool Perspective Environment

Modern computer and communication technologies not only improve the quality, accuracy, and timeliness of design and decisions but also provide tools for automating the design and decision making processes. Under such a tool perspective, human experts focus on developing tools, then users can apply the tools to model and resolve their automation problems (see the right-side diagram in Figure 13). In contrast with traditional approaches, difficulties occur only during the tool development. For traditional approaches, "experts" are the bottlenecks in design projects. They must understand not only the problems but also the applications and limitations in practice. In practice, the costs of looking for and working with the experts are the main expense. For tool perspective, the cost of training the users becomes the major expense. However, frequent environment changes usually result in the design of a flexible system to respond to repeated needs to modify and evaluate the existing system. Modification and evaluation needs can be fulfilled by rapidly applying computerized tools, which provide relatively high flexibility in design. Some researchers have noted the importance of modeling manufacturing systems with the tool perspective. A sample survey is presented in Table 2. The three modeling concerns ((1) conflicts among designers, (2) constraints in physical environment, (3) the information flows in manufacturing systems) are part of the criteria in the survey table. It is found

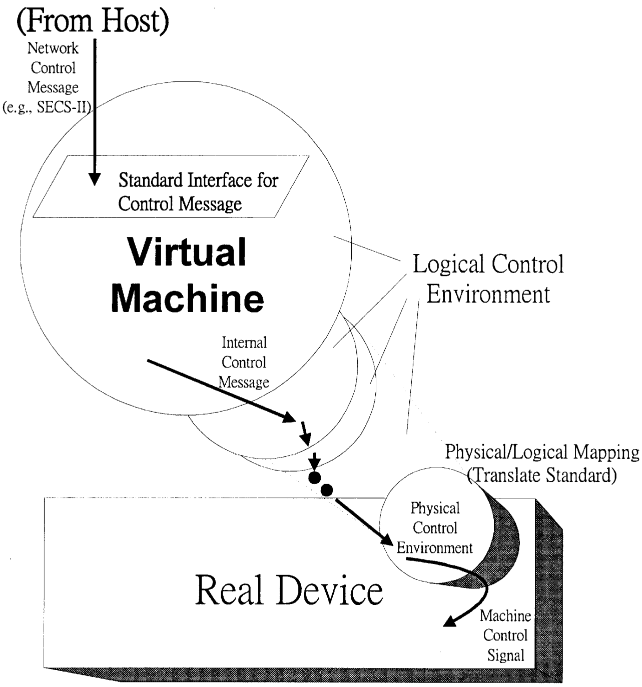


Figure 12 Role of Virtual Machine in Automation Applications.

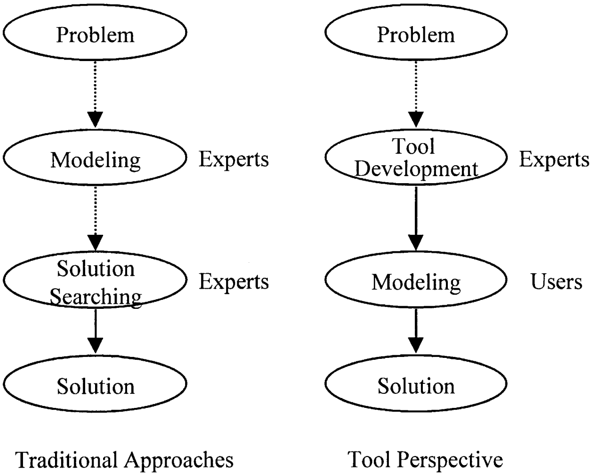


Figure 13 A Comparison of Traditional Approaches and Tool Perspective. (From Huang and Nof 1998)

TABLE 2 A Sample Survey of Models of Material Handling and Robotics with the Tool Perspective

No.	Modeling Approach/Tool, Reference	Methodologies Applied	Measured Criteria or Goal of the Model	Three Modeling Concerns Addressed?
1.	Nadoli and Rangaswami (1993)	Expert system, simulator	Waiting time, congestion level, etc.	No
2.	Chadha et al. (1994)	IDEF0, Data flow diagram, extended entity relationship	Develop an integrated information system	(3) only
3.	Prakash and Chen (1995)	SIMAN IV simulator	Speed of AGV and dispatching rules	No
4.	Sly (1995)	AutoCAD with FactoryFLOW	Minimum distance of material flows	No
5.	*FDL* (Witzerman and Nof 1995, 1996); FDL/CR (Lara et al. 2000)	ROBCAD and simulator	Integrated tool for distributed engineers; computer support for conflict resolution	(1), (2), and (3)
6.	*CFS* (Furtado and Nof 1995)	Data/control flow diagram, Petri net	Integrated tool to incorporate material flows, information flows, and control signals	(1) and (3) only

Adapted from Huang and Nof (1998).

that the concerns are not addressed by the first four models. The approach of the first four models still follows the traditional approaches, except for the second model. In the following paragraphs, two automated, integrated tools for manufacturing system design are introduced to present the above concept: facility description language (FDL) and the other is concurrent flexible specification (CFS).

6.2.1. Facility Description Language (FDL)

FDL provides an integrated modeling tool for distributed engineers working on various aspects of manufacturing system design (Witzerman and Nof 1995, 1996; Lara et al. 2000). FDL is implemented in a 3D emulation environment, ROBCAD (Tecnomatix 1989), which provides a dynamic and physically visible (comparing with ''iconically visible'' in simulation animation) CAD environment. Therefore, all the materials, operations of robots and material handling facilities, and machines are shown by their true physical relationships.

In the framework of FDL, the manufacturing systems are modeled by computer information and graphics. Various information teams are included in the model that are not seen in traditional models (Witzerman and Nof 1995):

1. Organizational relationship among facility components
2. Specification of working locations for robots
3. Flow of parts and materials through the facility
4. Location of personnel and equipment aisles
5. Control relationships between devices
6. Designation of sensors and their targets

These information items are supported by the modeling functions of FDL (Table 3). An example of modeling an aisle by an FDL function is:

TABLE 3 FDL Modeling Functions

Function Class	Function
FDL Definition Input Functions	Aisle, Capabiity, Control, Define (aisle, path, perimeter), Device, Facility, Part, Process, ProcessPart, Sensor, Transfer, Workpoint
FDL Manipulation Input Functions	Attach, Delete, Detach, Display (top, lower, bottom, name), Moveback, ShiftBy, ShiftTo
FDL Utility Input Functions	Comment, Print, Save
FDL Evaluation Function	Reconcile Database, Evaluate Aisles, Evaluate Device Reach, Display Material Flow Paths, Evaluate Fields of View

Aisle	*Redefine aspects pertaining to an aisle.*	
	syntax	***aisle** action path parent size*
	action	char (A, C, D) add, change, delete record (mandatory)
	path	char[16] path name (ROBCAD name)
	parent	char[16] parent name from device table
	size	char "HUMAN", "AGV", "SMALL_FL", or "LARGE_FL"

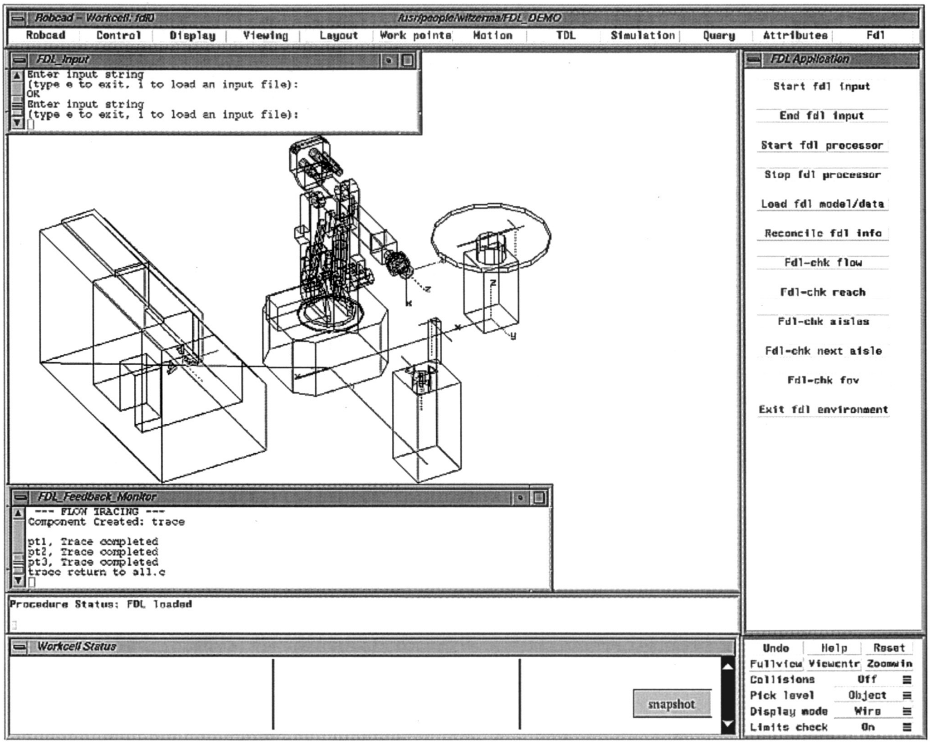
The model in FDL then becomes a list of syntax. The list of syntax triggers the ROBCAD program to construct a 3D emulation model (see the model inside the window of Figure 14). Figure 14 is an example of FDL in a ROBCAD system. The upper left window is used to input the information of the system, including the geometric information of facilities, material flow information, and material flow control information. The lower left window is used to show the output information (e.g., a collision occurring to the robot during the material handling). In addition, FDL provides a reconciliation function (see the right function menu in Figure 14). Therefore, all the control and physical conflicts on the manufacturing systems can be resolved according to the built in algorithm. The reconciliation function may change the positions of robots or machines to avoid the collision or unreachability of material handling. Recently, FDL/CR has been developed to provide knowledge-based computer support for conflict resolution among distributed designers.

Because FDL provides such direct syntax specifications, the designers can use the syntax to model and develop their subsystems. When the designers are in different locations, their subsystems can submit input to the host ROBCAD system to construct the entire system, then use the reconciliation function to adjust the subsystems if conflicts occur. Therefore, the cooperation of designers in different locations for different subsystems can be achieved in FDL. In the FDL working environment, two types of information are exchanged among the designers: (1) the design based on FDL syntax and (2) the operations of the facilities described by a task description language (TDL). TDL represents the control functions of the facilities. Thus, not only the models of the material flow but also the control information are presented and shared among the designers.

6.2.2. Concurrent Flexible Specifications (CFS)

By specification, engineers describe the way that a system should be constructed. Concurrent, flexible specifications for manufacturing systems, in other words, are provided by several engineers to model manufacturing systems with flexibility to design changes. In a manufacturing system, there is a physical flow of raw materials, parts, and subassemblies, together with an information and control flow consisting of status (system state) and control signals. The control and status signals govern the behavior of the physical flows. In order to simultaneously achieve optimal capacity loading with maintained or increased flexibility, an exact definition of material and information flow becomes necessary. This approach is followed by CFS modeling (Furtado and Nof 1995). The specification should represent not only the logical structures of different functions but also their logical connections, such as the structures of material and information flow in a manufacturing cell (Csurgai et al. 1986).

Another important requirement of specification is the ability to define precisely the real-time behavior of the system. In a real-time system, many of the inputs to the system are signals that indicate the occurrence of events. These inputs do not pass data to the system to be processed. Generally they occur in streams over time and their purpose is to trigger some process in the system



```
Enter input string
(type e to exit, i to load an input file):
OK
Enter input string
(type e to exit, i to load an input file):
```

```
 --- FLOW TRACING ---
Component Created: trace

pt1, Trace completed
pt2, Trace completed
pt3, Trace completed
trace return to all.c
```

```
Procedure Status: FDL loaded
```

```
Collisions     Off
Pick level     Object
Display mode   Wire
Limits check   On
```

Figure 14 An Example of Facility Description Language in ROBCAD System.

repeatedly. Furthermore, many systems are made up of subsystems, any of which may be active or nonactive at a particular time during system operation. For this reason, the treatment of timing is an essential element of the specification.

To integrate the above requirements, tools that can incorporate material flows, information flows, and control signals are required. Therefore, different representations of specifications through different tools should not be independent or mutually exclusive but should support each other by forming a concurrent, comprehensive specification of the system. For manufacturing systems, the specification of *functional* and *real-time logic* is important because these attributes describe what and how the system is executing. This information is necessary to determine how the processes are to be implemented with physical equipment. By utilizing two complementary representations, both these aspects of system behavior can be specified concurrently.

Data/control flow diagrams (DFD/CFDs), which are enhanced with real-time extensions when used in conjunction with Petri nets, provide a suitable framework for concurrent specification of functional and real-time state logic. The main reason for this is the ability to maintain identical structural decompositions in both representations at all levels of detail in the specification. This model is accomplished by maintaining identical partitioning of processes in both specifications.

With DFD/CFDs, partitioning is accomplished by hierarchical decomposition of bubbles that represent processes or tasks. An identical hierarchical partitioning can be created with Petri nets by representing processes with subnets at higher levels and then showing the detailed, internal net at lower levels of detail. The DFD/CFDs provide a process definition model of the system, while the Petri nets provide a process analysis model for the study of real-time state behavior. Even though object-oriented modeling is becoming a more popular technique of system design, data/control flow diagrams are still an acceptable technique in our case study. Researchers have proved the possibility of transforming data flow diagrams to object models (Alabiso 1988).

Both these techniques are realized by two software packages: Teamwork and P-NUT. Teamwork is a computer aided software engineering (CASE) tool family that automates standard structured methodologies using interactive computer graphics and multiuser workstation power. P-NUT is a set of tools developed by the Distributed Systems Project in the Information and Computer Science

Department of the University of California at Irvine (Razouk 1987) to assist engineers in applying various Petri net-based analysis methods.

6.3. Agent-Based Control Systems

Early agents were defined for distributed artificial intelligence, which includes two main areas: distributed problem solving (DPS) and multi-agent systems (MASs). DPS focuses on centrally designed systems solving global problems and applying build-in cooperation strategies. In contrast, MAS deals with heterogeneous agents whose goal is to plan their utility-maximizing coexistence. Examples of DPS are mobile robots exploring uncertain terrain, and task scheduling in manufacturing facilities. Both can be operated with centralized programs, but in relatively more distributed environments they are usually more effective with autonomous programs, or agents. Examples of MAS are collaborative product design and group behavior of several mobile robots.

Recent research has explored the concept of autonomous agents in control. An agent is a computing system that can autonomously react and reflex to the impacts from the environment in accordance with its given goal(s). An agent reacts to the environment by executing some preloaded program. Meanwhile, there is an autonomous adjustment mechanism to provide a threshold. When the environmental impacts are higher than the threshold, the agent reflexes; otherwise it is intact. An agent may seek collaboration through communicating with other agents. The communication among agents is regulated by protocols, structure of dialogue, to enhance the effectiveness and efficiency of communication.

An important difference between autonomous agents and other techniques is that an autonomous agent evaluates the rules that it will perform. It may even automatically change its goal to keep itself alive in a harsh environment. Autonomous agents have been applied in many control systems, including air traffic control, manufacturing process control, and patient monitoring.

Usually an agent functions not alone, but as a member of a group of agents or an agent network. The interaction and communication among agents can be explained by the analogy of organizational communication. An organization is an identifiable social pursuing multiple objectives through the coordinated activities and relations among members and objects. Such a social system is open ended and depends for its effectiveness and survival on other individuals and subsystems in the society of all related organizations and individuals. (It is actually similar for both human societies and agent societies.) Following this analogy, three characteristics of an organization and of an agent network can be observed (Weick 1990):

1. Entities and organization
2. Goals and coordinated activities
3. Adaptability and survivability of the organization

Five motivations have been observed for organizational and agent network communication (Jablin 1990):

1. Generate and obtain information
2. Process and integrate information
3. Share information needed for the coordination of interdependent organizational tasks
4. Disseminate decisions
5. Reinforce a group's perspective or consensus

These five motivations can serve as a checklist for developing protocols. One of the most influential factors affecting interpersonal or interagent communication patterns among group members is the characteristic of the task on which they are working. As task certainty increases, the group coordinates itself more through formal rules and plans than through individualized communication modes. Therefore, the interacting behaviors and information exchanges among agents have to follow interaction and communication protocols.

Although different agent applications will require different agent design, five general areas have to be addressed:

1. Goal identification and task assignment
2. Distribution of knowledge
3. Organization of the agents
4. Coordination mechanism and protocols
5. Learning and adaptive schemes

Research into intelligent, collaborative agents is extremely active and in its preliminary stages (Nof 1999; Huang and Nof 2000). While the best-known applications have been in Internet search and remote mobile robot navigation, emerging examples combine agents through computer networks with remote monitoring for security, diagnostics, maintenance and repair, and remote manipulation of robotic equipment. Emerging agent applications will soon revolutionize computer and communication usefulness. Interaction and communication with and among intelligent tools, home appliances, entertainment systems, and highly reliable, safe mobile service robots will change the nature of manufacturing, services, health care, food delivery, transportation, and virtually all equipment-related activities.

7. CONCLUSION

This chapter discusses automation technology in various levels:

1. *Single process:* automatic control theory and technology
2. *Single machine:* artificial intelligence and knowledge-based technology
3. *Distributed machines and distributed systems:* integration theory and technology

Additionally, the trends in future automation technology are classified into another three levels:

1. *Machine integration:* virtual machines
2. *Human–machine integration:* tool-oriented technologies
3. *Machine autonomy:* agent-based technologies

In our postulation, how to automate machines in terms of operations will soon be a routine problem because of the technological maturity of actuators, sensors, and controllers. The application of automation technology will then focus on the issues of intelligence, integration, and autonomy. Another emerging trend involves the incorporation of micro-electromechanical systems (MEMSs) as sensors and controllers of automation and the microscale. However, education and training of workers who interact with intelligent and autonomous machines may be another important research issue in the future.

Acknowledgments

The authors would like to thank Professor Ted J. Williams of Purdue University and Professor Li-Chih Wang, Director of the Automation Laboratory, Tunghai University, Taiwan, for their valuable contributions to this chapter.

REFERENCES

Alabiso, B. (1988), “Transformation of Data Flow Diagram Analysis Models to Object-Oriented Design,” in *OOPSLA ’88: Object-Oriented Programming Systems, Languages and Applications Conference Proceedings* (San Diego, CA, September 25–30, 1988), N. Meyrowitz, Ed., Association for Computing Machinery, New York, pp. 335–353.

Burdea, G. C., and Coiffet, P. (1999), “Virtual Reality and Robotics,” in *Handbook of Industrial Robotics, 2nd Ed.,* S. Y. Nof, Ed., John Wiley & Sons, New York, pp. 325–333.

Chadha, B., Fulton, R. E., and Calhoun, J. C. (1994). “Design and Implementation of a CIM Information System,” *Engineering with Computers,* Vol. 10, No. 1, pp. 1–10.

Cohen, A., and Apte, U. (1997), *Manufacturing Automation,* McGraw-Hill, New York.

Csurgai, G., Kovacs, V., and Laufer, J. (1986), “A Generalized Model for Control and Supervision of Unmanned Manufacturing Cells,” in *Software for Discrete Manufacturing: Proceedings of the 6th International IFIP/IFAC Conference on Software for Discrete Manufacturing, PROLAMAT 85* (Paris, France, June 11–13, 1985), J. P. Crestin and J. F. McWaters, Eds., North-Holland, Amsterdam, pp. 103–112.

Dayhoff, J. E. (1990), *Neural Network Architecture,* Van Nostrand Reinhold, New York.

Elnakhal, A. E., and Rzehak, H. (1993), “Design and Performance Evaluation of Real Time Communication Architectures,” *IEEE Transactions on Industrial Electronics,* Vol. 40, No. 4, pp. 404–411.

Freeman, J. A., and Skapura, D. M. (1991), *Neural Networks: Algorithms, Applications, and Programming Techniques,* Addison-Wesley, Reading, MA.

Fuller, R. (1995), *Neural Fuzzy Systems,* http://www.abo.fi/~rfuller/nfs.html.

Furtado, G. P., and Nof, S. Y. (1995), "Concurrent Flexible Specifications of Functional and Real-Time Logic in Manufacturing," Research Memorandum No. 95-1, School of Industrial Engineering, Purdue University.

Huang, C. Y., and Nof, S. Y. (1998), "Development of Integrated Models for Material Flow Design and Control—A Tool Perspective," *Robots and CIM,* Vol. 14, 441–454.

Huang, C. Y., and Nof, S. Y. (1999), "Enterprise Agility: A View from the PRISM Lab," *International Journal of Agile Management Systems,* Vol. 1, No. 1, pp. 51–59.

Huang, C. Y., and Nof, S. Y. (2000), "Formation of Autonomous Agent Networks for Manufacturing Systems," *International Journal of Production Research,* Vol. 38, No. 3, pp. 607–624.

Jablin, F. M. (1990), "Task/Work Relationships: A Life-Span Perspective," in *Foundations of Organizational Communication: A Reader,* S. R. Corman, S. P. Banks, C. R. Bantz, M. Mayer, and M. E. Mayer, Eds., Longman, New York, pp. 171–196.

Kuo, B. C. (1995), *Automatic Control Systems,* 7th Ed., John Wiley & Sons, New York.

Lara, M. A., Witzerman, J. P., and Nof, S. Y. (2000), "Facility Description Language for Integrating Distributed Designs," *International Journal of Production Research,* Vol. 38, No. 11, pp. 2471–2488.

Lin, C.-T., and Lee, G. (1996), *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems,* Prentice Hall, Upper Saddle River, NJ.

Mahalik, N. P., and Moore, P. R. (1997), "Fieldbus Technology Based, Distributed Control in Process Industries: A Case Study with LonWorks Technology," *Integrated Manufacturing Systems,* Vol. 8, Nos. 3–4, pp. 231–243.

Morriss, S. B. (1995), *Automated Manufacturing Systems: Actuators, Controls, Sensors, and Robotics,* McGraw-Hill, New York.

Nadoli, G., and Rangaswami, M. (1993), "An Integrated Modeling Methodology for Material Handling Systems Design," in *Proceedings of the 1993 Winter Simulation Conference* (Los Angeles, December 12–15, 1993), G. W. Evans, Ed., Institute of Electrical and Electronics Engineers, Piscataway, NJ, pp. 785–789.

Nof, S. Y., Ed. (1999), *Handbook of Industrial Robotics,* 2nd Ed., John Wiley & Sons, New York.

Prakash, A., and Chen, M. (1995), "A Simulation Study of Flexible Manufacturing Systems," *Computers and Industrial Engineering,* Vol. 28, No. 1, pp. 191–199.

Razouk, R. R. (1987), "A Guided Tour of P-NUT (Release 2.2)," ICS-TR-86-25, Department of Information and Computer Science at the University of California, Irvine.

Semiconductor Equipment and Materials International (SEMI) (1997), *SEMI International Standards,* SEMI, Mt View, CA.

Shanmugham, S. G., Beaumariage, T. G., Roberts, C. A., and Rollier, D. A. (1995), "Manufacturing Communication: A Review of the MMS Approach," *Computers and Industrial Engineering,* Vol. 28, No. 1, pp. 1–21.

SISCO Inc. (1995), *Overview and Introduction to the Manufacturing Message Specification* (*MMS*) *Revision 2.*

Sly, D. P. (1995), "Plant Design for Efficiency Using AutoCAD and FactoryFLOW," in *Proceedings of the 1995 Winter Simulation Conference* (Arlington, VA, December 3–6, 1995), C. Alexopoulos, K. Kong, W. R. Lilegdon, and D. Goldsman, Eds., Society for Computer Simulation, International, San Diego, CA, pp. 437–444.

Tecnomatix Technologies (1989), *ROBCAD TDL Reference Manual,* version 2.2.

Wang, L.-C. (1996), "Object-Oriented Petri Nets for Modelling and Analysis of Automated Manufacturing Systems," *Computer Integrated Manufacturing Systems,* Vol. 9, No. 2, pp. 111–125.

Wang, L. C., and Wu, S.-Y. (1998), "Modeling with Colored Timed Object-Oriented Petri Nets for Automated Manufacturing Systems," *Computers and Industrial Engineering,* Vol. 34, No. 2, pp. 463–480.

Weick, K. (1990), "An Introduction to Organization," In *Foundations of Organizational Communication: A Reader,* S. R. Corman, S. P. Banks, C. R. Bantz, M. Mayer, and M. E. Mayer, Eds., Longman, New York, pp. 124–133.

Williams, T. J., and Nof, S. Y. (1992), "Control Models," in *Handbook of Industrial Engineering,* 2nd Ed., G. Salvendy, Ed., John Wiley & Sons, pp. 211–238.

Witzerman, J. P., and Nof, S. Y. (1995), "Facility Description Language," in *Proceedings of the 4th Industrial Engineering Research Conference* (Nashville, TN, May 24–25, 1995), B. Schmeiser and R. Vzsoy, Eds., Institute of Industrial Engineers, Norcross, GA, pp. 449–455.

Witzerman, J. P. and Nof, S. Y. (1996), "Integration of Simulation and Emulation with Graphical Design for the Development of Cell Control Programs," *International Journal of Production Research,* Vol. 33, No. 11, pp. 3193–3206.