26.1 Introduction

As the global marketplace demands higher quality goods and lower costs, factory floor automation has been changing from separate machines with simple hardware-based controls, if any, to an integrated manufacturing enterprise with linked and sophisticated control and data systems. For many organizations the transformation has been gradual, starting with the introduction of programmable logic controllers and personal computers to machines and processes. However, for others the change has been rapid and is still accelerating. This chapter discusses the current state of control and data systems that make up manufacturing automation.

26.2 Process Questions for Control

The appropriate level of control and automation depends on the process to be automated. Before this can be accomplished, questions about the physical process and product requirements must be answered.
1. What types of process and product feedback are required to control the process (e.g., line speed, force, pressure, temperature, length, thickness, moisture, color)?
2. How is the process run (continuous, batch, sequential operations)?
3. What is the current level of automation (none, relay logic, programmable controller, etc.)?
4. What is the process operation schedule (single shift or 24-h operation)?
5. What cost opportunities are available from reduction of waste, improvement of quality, reduction of downtime?

The last question, which is financial, is typically the most important. When applied correctly process control and automation will rapidly pay for itself.

### 26.3 Terminology

- **AS-i** Actuator sensor interface
- **CAN** Controller area network
- **DCS** Distributed control system
- **DDE** Dynamic data exchange
- **FBD** Function block diagram
- **GUI** Graphical user interface
- **HMI** Human machine interface
- **IL** Instruction list
- **I/O** Inputs and outputs (analog or digital)
- **IEC 61158** International standards for controller fieldbuses
- **IEC 61131** International standards for programmable controllers
- **LLD** Ladder logic diagram programming language
- **OLE** Object linking and embedding
- **OPC** OLE for process control
- **P&ID** Piping and instrumentation diagrams
- **PID** Proportional, integral, and derivative
- **PLC** Programmable logic controller
- **SCADA** Supervisory control and data acquisition
- **SFC** Sequential flow chart
- **ST** Structured text
- **TCP/IP** Transmission control protocol/internet protocol
- **UDP/IP** User datagram protocol/internet protocol

### 26.4 Hierarchy of Control and Automation

Figure 26.1 depicts a hierarchy of automation and control. While the desired top level of automation depends on the goals of the project, control typically starts with the first two elements of motor control and event control. The top level of manufacturing automation is enterprise information feedback, commonly referred to as a manufacturing management information system, Figure 26.2. For some manufacturers the desired stage of automation is integrated statistical process control, for others it is feedback for product design. In either case, the design of the industrial control system is critical to overall process success.

#### 26.4.1 History

The basic elements of motor control and event (I/O) control have long been thought of as separate actions. Early process control consisted of speed regulation of asynchronous or DC servo motors via analog or
relatively simple control methods. Event control was often accomplished with relay logic. Automatic control was all hardware-based, and as such it was not easily changed or improved. As microprocessors became more prevalent and accepted in the later part of the 20th century, programmable logic controllers (PLC) were introduced and vastly improved process event control and provided the ability to easily modify a process. A separate and parallel action was programmable motion controllers. With the increasing computational power of successive versions of microprocessors, proportional, integral, and derivative (PID) control was easily implemented in these controllers. This allowed relatively easy tuning of servomechanisms. Communication between the two controller types was initially analog signals, then serial data, and most recently one of several data networks. While the first motion and process controllers were great milestones, integrated process and motion control with real-time process data availability did not appear until the late 1990s. Critical processes, such as high speed drawing of optical fiber, required tightly couple motion and process control to manufacture competitively.

Thus, modern manufacturing automation systems joined motion control and process control together for greater flexibility and control potential. Along with this improvement came newer and faster data buses,
such as 100 Mbit Ethernet, for sharing real-time process information across the factory. This provided the opportunity to arrive at the next level of automation that uses real-time process and product measurement data for statistical process control using product quality metrics. Many controllers integrate web-based features such as TCP/IP or UDP/IP, detailed data structures, allowing easy flow of information between controllers. Often this type of networking is used in supervisory control and data acquisition (SCADA) applications. With this configuration real-time process data and product quality feedback are available to any computer or person requiring the information. Many processes can be programmed and run remotely via intranets or using secure internet logins (although remote programming and operation of processes is not necessarily a safe way of performing these tasks).

The elements of these modern manufacturing automation systems and their usage are given in the following sections. Programming and appropriate system designs for applications are also discussed.

## 26.5 Controllers

There are many different distinctions in the area of industrial automation controllers. The most widely used controllers are motion controllers, programmable logic controllers (PLC), distributed control systems (DCS), and PC-based control. Each controller type has special features that make it the controller of choice for different automation projects. Many of the distinctions of the various controller types are beginning to be less noticeable with each successive product generation, as options expand and pitfalls are addressed.

### 26.5.1 PLC: Programmable Logic Controller

The programmable logic controller (PLC) has been part of manufacturing automation for over two decades, replacing the hard-wired relay logic controllers (Figure 26.3). For smaller-scale, event-driven processes and machines with limited I/O points, stand alone PLCs are the controller of choice. PLCs are rugged, relatively fast, and low cost with excellent sequential control performance.

In the last 10 years the functionality of the PLC and systems using PLCs has been growing rapidly, with integration of networking, peripherals, and expanded programming options. The distinction between PLC

![FIGURE 26.3 PLC-I/O structure.](image-url)
Manufacturing Automation systems and other more complex controllers (e.g., DCS) is diminishing, as PLCs are moving up in function and connectivity. Advanced PLCs and DCSs overlap each other’s controller areas. Indeed, the networked PLC-SCADA systems are virtually equivalent to the larger DCS systems [1]. Programming standards and networks have also removed the limitations of PLCs. In addition to the traditional ladder logic, four other standardized programming languages are available (discussed in software).

26.5.2 DCS: Distributed Control System

Traditionally a DCS is a system of controllers linked by a data network, as a single system. Functionality, physical location, or both separate these controllers. The DCS is typically used in complex process applications where large amounts of I/O and data are required, such as chemical plants or oil refineries. They are well suited to batch processes and have an ability to handle complex interlocks and timing between operations. DCS are multitasking systems able to handle large common databases. A DCS allows for various control loops, using graphical representation of function blocks, and is easier to program than ladder logic-based PLCs. Scan rates can be more predictable than the PLCs. PLCs have scan rates dependent on the amount of I/O [2]. A DCS has safety features, redundancy, and even diagnostics built into its control philosophy to be more robust with less down time [1].

Other advantages of a traditional DCS include

- Integrated system of hardware and software reduces engineering time.
- Database applications are easily integrated with DCS.
- Process simulation and advanced control applications are readily available.
- System is designed for ease of future expansions.

26.5.3 Hybrid Controller

Some controllers define themselves as hybrid controllers, able to perform control of continuous processes (DCS) and discrete events (PLC) as well as batch and logic. The main difference is in the algorithms that are used by each controller. Hybrids can provide a combination of DCS-like infrastructure as well as SCADA functions, all while being used in process/discrete event automation. Hybrid systems are very flexible. They provide HMI/SCADA functionality, along with complex continuous and discrete control, or demanding chemical processing functions in applications with limited I/O to those with thousands of I/O points.

26.5.4 Motion Controller

Motion controllers are implemented as machine controllers when position, velocity, or other servo-control loop critical functions must occur in a process with limited I/O. Machine tools and robotics are the primary uses of these controllers. The motion controllers have built-in servo control algorithms, typically PID. Motion and process I/O data are easily integrated into control loops, as they reside on the same CPU. Some manufacturers allow user-defined interrupt service routines (ISR) for advanced control algorithms.

The motion controllers offer faster update rates for high bandwidth, servo-control loops. Motion controllers can provide servo update rates greater than 1 kHz. However, as integration of motion control into PLCs advances with new control networks, the need of using a motion controller as a process controller is diminishing for processes requiring high speed servo control with large amounts of process I/O. Motion controllers generally provide significantly fewer I/O points than PLCs or DCSs. PLC manufacturers have specific PLC/motor driver/high speed bus configuration for accessing and controlling motion within the PLC [3].

A major limitation of motion controllers as process controllers is their use of unique and proprietary programming languages. Most of these are textual languages, sometimes based on the C programming language. This limits the portability and maintainability of the controller programs.
26.5.5 PC-Based Open Controller

As personal computer (PC) operating systems have become more stable, the open architecture PC systems have made their way into factories. The drive for data automation and electronic commerce along with lower initial and operating costs, flexibility, and built-in networking ability are some of the reasons for pursuing this approach. PCs have an open systems advantage over both DCS and PLC approaches.

PC deployment in control systems occurs in several different ways. In some applications PCs are used as HMIs linked to process controllers. For other applications the PC is the controller and HMI. A PC-based controller is an integrated package of software (operating system, programs, data structures, communication protocols) and hardware (CPUs, communication/network cards, I/O device networks, I/O modules, bus resident controllers, etc.). Another driving reason for PC-based control is the open architecture, allowing the linking and integration of different hardware, software, and networks. While individually excellent components can be purchased and assembled, the success of the system depends on the compatibility of both the hardware and software components [4].

Typically PC-based controls are built using fairly stable operating systems (OS), such as Windows NT/2000/XP/CE or Linux. There are three levels of control/operating systems interactions. First is relying completely on the OS for all levels of control. This is only acceptable in noncritical, low usage systems that do not require real-time control at regular update intervals. Second is the use of a real-time operating system or real-time kernel for Windows/DOS. This provides real-time control with easy access to all the standard PC components. These are sometimes referred to as software PLC applications. Both of these schemes should be avoided in critical applications including life safety systems, to avoid possible shutdowns from suspensions or system reboots. The third PC-based control approach is the use of a separate controller card on the PC bus. Here the real-time control resides in the controller card, which takes its power from the bus. All control I/O are connected through the card. The advantage of this scheme is that the PC may be re-booted without affecting the controller operation. All three PC approaches have cost savings associated with using the PC as a database, communication host, and HMI [5].

26.6 Control Elements

26.6.1 HMI: Human-Machine Interface

An integral part of the modern industrial control system is the human-machine interface (HMI). Before the widespread use of microprocessor-based graphical displays, an HMI consisted of hardwired dial gauges, strip chart and servo recorders, light indicators, and various push buttons. Modern controllers and automation systems interact with people mainly through graphical user interfaces (GUI)[6]. These GUI interfaces most often appear as touch screens with a menu driven display. Minimally HMIs display process information and controls in real-time. Examples of a main menu and subsystem menu are given in Figure 26.4 and Figure 26.5. The graphics also provide some physical system insight, as in the gas flow control menu of Figure 26.5. A low-feature HMI (some are integrated with low cost PLCs) may simply display a current rung of the ladder logic diagram.

With a PC-based HMI, powerful graphics can be used. Historical and real-time data can be displayed, using the databases, file structures, and graphics capabilities of a PC. The use of dynamic data exchange (DDE), multiple open windows, and other standard PC platform software components, such as OPC, augments its functionality [7]. Control parameters and functions are integrated into the HMI screens for easy use by operators and control engineers. Some displays mimic the analog devices they replaced (dial gauges, LED displays, indicator lights, etc.). Other features include charting and trending of data for quality control. Animations are also available for better display process status. System status and other diagnostic tools are also presentable. Using data monitoring with alarms, an HMI may also serve as a supervisory control and data acquisition system (SCADA).
FIGURE 26.4 HMI main menu example.

FIGURE 26.5 HMI gas delivery sub-system menu example.
HMIs can be devices connected directly to the controllers or to database servers that pass information back and forth to the process controller via several network protocols. In fact many HMIs are run in a Web-browser windows, allowing remote viewing and control.

26.6.2 I/O: Inputs and Outputs

To interface with a physical process a PLC, DCS, or PC-based controller must have inputs and outputs. Inputs and outputs (I/O) were initially contact relays and instrument circuits. Today nearly all I/Os are optically isolated, solid-state devices, with many specializations. The optical isolation prevents any device voltage or current from damaging the controller. To verify an event or the presence of an object at a particular location, a digital input is used; this is simply the completion of circuit, typically a threshold voltage. Digital outputs provide a voltage to turn on off devices, such as a solenoid.

The digital I/O are designated by voltage thresholds, AC or DC current (typically 24 VDC or 120 VAC), and whether the current flow to the I/O unit is sinking or sourcing. Sinking and sourcing refer to the current flow into the input or output switch (transistor). Note that this is only relevant for DC voltage and is not associated with AC currents. All inputs require a voltage source and a load to operate. A sinking input (NPN transistor: Figure 26.6) requires the voltage and load to be present before connecting it to the circuit. This means that it is “sinking” the current to ground for the circuit. A sourcing input (PNP transistor: Figure 26.6) must be before the load in the circuit. This means that it is “sourcing” the current to the circuit, from a positive reference voltage. Voltage and a load must be present in either situation to detect a voltage change at the input. A similar logic follows for sinking or sourcing outputs. A sourcing I/O device is generally considered safer than sinking, as supply voltage is not always present at the device. Although, using proper maintenance safety, both types of I/O are safe. Typical digital inputs are push buttons, selector switches, limit switches, level switches, photoelectric sensors, proximity sensors, motor starter contacts, relay contacts, thumbwheel switches. Typical digital outputs are valves, motor starters, solenoids, control relays, alarms, lights, fans, horns, etc.

Analog inputs and outputs are more diverse than digital inputs and outputs. Many additional parameters are used as these devices are used to measure an output voltage and current. For long distances, the current loop devices (generally 4–20 mA) are preferred over the voltage I/O (most commonly 0–5 V) for accurate measurements without line losses. Special purpose analog inputs are available for ease in connecting various types of thermocouples and resistance temperature transducers (RTD).

Apart from the type of analog signal, the resolution of the signal is the most important consideration. Most applications are 8, 16, or 32 bit resolution for scaling the range of the particular I/O unit. So for maximum resolution chose an input or output range that matches the output of the sensor. In 16-bit applications, one bit is used to denote positive or negative leaving 215 bits for data. For example, a −10 to +10 V analog input would result in a resolution of 0.3 mV regardless of the magnitude of the signal.

I/O are linked to the controllers via many routes. All controllers have module connections that are either integral to the controller or connected to the main data bus, for example, the backplane of PLCs. Different I/O modules can be connected for the specific applications (analog, digital, inputs, outputs). There are a limited number of module configuration slots and, thus, limited I/O points with these direct connections, as specified by the manufacturer. Devices are hardwired to the I/O point on the modules and must be configured on the controller. Many controllers allow remote I/O racks with hardwiring back to the controller. Recently device-level networks and field buses have emerged to address problems associated...
with a central controller with detached I/O racks and other devices. This allows the reduction of control wiring with remote I/O and improved diagnostics with intelligent devices. These networks are discussed in the following section.

### 26.7 Networking and Interfacing

As networking information technology has evolved and become more robust, it was a logical step for process automation to take advantage of these technologies. Communication networks abound at many levels in modern industrial automation. Figure 26.7 depicts several levels of networking in a typical manufacturing control system. Several somewhat distinct levels of connections and control can be visualized and coexist: sensor/device-level, process control, controller-level, and information-level networks. (Control networks below the information-only level can be referred to as fieldbuses.) As protocols and fieldbus technologies advance, the differences between the sensor, device, process, and controller levels is becoming fuzzy. This subsection on control networks and communication highlights some of the technology in manufacturing today and is not intended as a comprehensive discussion.

#### 26.7.1 Sensor-Level I/O Protocol

The transition from hardwired I/O to direct network connection is rapidly occurring. The first level of networking in an industrial control system is between the controller and its sensors. Sensor-level networks are generally used in smaller systems with limited I/O.

The highway addressable remote transducer (HART) sensor-level communications protocol is an open protocol developed in the 1980s for communication at the I/O level. It was one of the first protocols for...
Device/sensor networking. It supports two-way digital communications between a HART I/O module and a HART capable device. The protocol is designed to complement a standard 4–20 mA analog signal, requiring no additional wires. HART uses a frequency modulation (e.g., 1200 Hz = “1”, 2200 Hz = “0”) with small amplitude (0.5 mA) on top of a 4–20 mA analog signal for device communication (Figure 26.8). This provides exceptional opportunities for sending device data to the controller, allowing for device identification, status, and diagnostics. While this protocol allows for data communication between devices, it has no more or less wiring than standard 4–20 mA devices. The current HART protocol provides for 15 nodes (devices) per loop [8, 9].

Another current sensor bus protocol present in industry is the actuator sensor interface (AS-i) protocol, [10]. AS-i was developed in Germany in 1994 by a group of industrial automation equipment suppliers to be a low-cost method for addressing discrete sensors in factory automation applications. The physical network layer is a balanced differential voltage (30–24 VDC). Each AS-i 2.0, network can support up to 31 devices or nodes (248 I/O points). The AS-i, version 2.1, specification doubles the maximum allowable nodes to 62 devices per segment for a total network capacity of 434 I/O points.

These sensor networks are cost effective for small systems, using master/slave communication techniques. However, modern manufacturing automation often requires more vertical integration of controllers, sensors, devices, and instruments, using peer-to-peer and publisher-subscriber techniques for higher communication bandwidth. More extensive industrial applications with larger I/O potential and more diverse devices require device networks in lieu of the node-limited sensor level networks.

### 26.7.2 Device-Level Networks

Device networks are more general and expandable than the lower level sensor networks. They connect a wider range of devices, including all I/O, motor drives, and display panels. Devices may be connected into the controller I/O rack or remotely via device-level networks using one or more of the variety of the available data buses. There are two driving forces behind device-level networking of the I/O units. One is the reduction of the wiring limitations of previously hard-wired remote I/O chassis. The second is the ability to have more device-level information to ease configuration and diagnostics. While complex in function, these network compatible devices are more easily set up than the hard-wired devices.

There are numerous device-level networks and fieldbuses already in use. For example, DeviceNet (Allen-Bradley/Rockwell Automation) and Profinet DP (Siemens) are in wide use. Each has its own rationale for use, much of which depends on the selected controller. Many recent efforts have focused on using Ethernet at the device level. While this is not the dominant approach for lower level networks, it is gaining interest. Ethernet does have a large presence at the controller-level and information networks.

The DeviceNet network is an open, low-level network that provides connections between simple industrial devices (sensors and actuators) and higher-level devices (PLCs and PCs). It currently has the largest installed base of all device networks [11]. DeviceNet is an open device network using controller area network (CAN) protocol to provide the control, configuration, and data collection capabilities for
industrial devices. DeviceNet can currently support up to 253 individual nodes at data rates of 125 kb at a maximum distance of 500 m, or 500 kb at shorter distances (without repeaters). The physical layer of CAN protocol uses a differential voltage.

Another device network with a significant installation base is Profinet-DP. Profibus-DP is a device-level bus that supports both analog and discrete signals. Profibus-DP has widespread usage for such items as remote I/O and motor drivers. Profibus-DP communicates at speeds from 94 kbps over distances of 1200 m to 12 Mbps up to 100 m, without repeaters. The physical layer of Profibus-DP is based on RS-485 communication [9,12]. Neither Profibus-DP nor DeviceNet is designed for intrinsically safe applications.

### 26.7.3 Advanced Process Control Fieldbuses

Process control networks are the most advanced fieldbus networks in use today. They provide connectivity of sophisticated process measuring and control equipment. Modern process control networks can be easily deployed for new or existing process equipment and provide more complex functionality than device networks. The advanced characteristics of the host interfaces and connectivity aid in the ease of configuration and start up.

While the MODBUS (Modicon PLCs) was one of the first process control field buses, Foundation Fieldbus and Profinet-PA are two dominant advanced process control fieldbuses/protocols. (Note: Profinet-PA is distinct from Profibus-DP.) Foundation Fieldbus and Profinet-PA are full-function fieldbuses for process-level instrumentation that has a physical layer based on IEC-61158, low-speed voltage mode [13]. Both provide connectivity of sophisticated process measuring and control equipment. Each provides communication at distances up to 1900 m and is intrinsically safe, with low current on the control wiring. Foundation Fieldbus has data transmission rates of 31.25 kbps, while Profinet-PA communicates at 93.75 kbps. Foundation Fieldbus and Profinet-PA (generally linked with Siemens) are market leaders for process fieldbuses, with significant installation bases in the U.S. and Europe, respectively.

When using a device-level network or process control fieldbus, sensors or other I/O devices can be added anywhere by tapping into the trunk line with a multi-wire connection. Devices can be configured over network. Many diagnostics are available along with large amounts of I/O data. Simple sensors to sophisticated devices, all work on the same network.

The key benefits of device and process fieldbuses are

- Less wiring and installation time
- Few wiring mistakes
- Remote and quick setup of devices
- Diagnostics for predictive failure warnings and rapid troubleshooting
- Interoperability of devices from multiple vendors
- Control I/O and information data transmitted together
- Configuration of devices over the network

### 26.7.4 Controller Networks

Interlinking of multiple controllers in a manufacturing process has grown rapidly with the demand for tighter real-time process control. Often several controllers are linked to PLCs, PCs, and motion controllers across several subprocesses. The controller level network requirements include deterministic data flow, repeatable (time) transfers of all operation-critical control data, in addition to supporting transfers of non-time-critical data. I/O updates and controller-to-controller interconnects must always take priority over program file transfers and other communication. Transmission times must be constant and unaffected by devices connecting to, or leaving, the network. In the controller-level arena, several popular fieldbuses are viable, including ControlNet, Foundation Fieldbus, Profinet-FMS, and Ethernet. Independent organizations support and maintain these network standards [12,14,15]. Ethernet using TCP/IP or UDP/IP is a strong contender and is rapidly gaining popularity.
Rockwell Automation conceived ControlNet as a high-level fieldbus network. It uses the control and information protocol (CIP) to provide dual functionality of an I/O network and a peer-to-peer network. Up to 99 nodes can be networked with data transfer rate of 5 Mbps at distances up to 5 km. Multiple PLCs can control unique I/O devices (multimaster) or share input devices on the same wire interlocking of PLCs, and this is accomplished using a producer/consumer network model (a.k.a. publisher/subscriber). ControlNet was designed with a time-slice algorithm that manages these functions on one network without affecting performance or determinism.

ProfiBus-FMS is a control bus generally used for communications between DCS and PLC systems most widely used with Siemens PLCs. It was meant to be a peer-to-peer connection between Profibus masters, exchanging structured data. The Profibus-FMS network is positioned in the architecture similarly to ControlNet. Profibus-FMS has various data rates available, from 9.6 kbps at a maximum distance of 19.2 km to 500 kbps for distances less than 200 m.

### 26.7.5 Information Networks and Ethernet

With several integrated IP protocols, Ethernet is the most popular network at multiple system levels. Control manufacturers select Ethernet to avoid many of the proprietary or custom networks. Ethernet provides interoperability among products from various vendors. 100 Mbps Ethernet is becoming a logical choice for controller networks as well as for information networks. Ethernet provides high standard data transfer rates, efficient error correction, and industrial Ethernet switches for better determinism. A typical architecture is seen in Figure 26.9. Ethernet also brings with it a large available IT expertise. Many control manufacturer have selected Ethernet for connecting remote I/O rack locations to controllers, effectively making Ethernet a process control fieldbus. There is even a push downward for device-level Ethernet.

There are many competitors to Ethernet including Modbus/TCP, ProfiNet, HSE Fieldbus, and many other proprietary protocols. Arguments against using Ethernet in factory floor automation often stress that Ethernet lacks the level of robustness and determinism needed in control applications. However, recent developments of intelligent switches have largely discounted this argument [16].

Along with the high data transfer rates, Ethernet also provides superior technology compared with others networks, in terms of openness, ease of access to and from the Internet, ease of wiring, setup, and maintenance. A new open application layer protocol, Ethernet/IP, has been created for the industrial environment. It is built on the standard TCP/IP protocols and takes advantage of commercial off-the-shelf Ethernet integrated circuits and physical media. IP stands for “industrial protocol.” Ethernet/IP is supported by several networking organizations: ControlNet International (CI), Industrial Ethernet Association (IEA), Open DeviceNet Vendor Association (ODVA), and Industrial Open Ethernet Association (IOANA) [11].
26.7.6 Selection of Controllers and Networks

The selection of controllers and networks is usually a function of the process specifications, existing controller base, and familiarity of personnel with specific controllers. When starting from scratch, the answers to the questions of Section 26.2 will provide guidance. Process familiarity in terms of the physical functions and requirements will greatly augment the piping and instrumentation diagrams (P&I D) and written specifications. Data and information flow is important in determining the appropriate network configuration and hardware (Figure 26.10). Data transmission speed, information complexity, and distance limits are critical design limits for selection of controllers and associated fieldbuses. There is no unique solution. Often there are several viable alternative configurations.

26.8 Programming

Software is the center of great debate and change. Ladder logic, rung ladder logic, or ladder logic diagrams (LLD) have long been the main mode of PLC programming as the diagrams graphically resemble the relay logic they originally replaced. Other styles of programming have evolved to augment computational abilities and ease program development. In addition to LLD, four other major programming paradigms are in use in PLCs and DCS systems. These are structured text (ST), function block diagrams (FBD), sequential flow charts (SFC), and instruction lists (IL). Special purpose motion controllers with augmented processing logic use proprietary text-based languages. While powerful, these unique control languages are not transportable to other controllers and make replacement difficult.

As particular controller vendors often customize their programming functions, it has become more important to have a programming standard for the five prevalent process control languages. One major reason is so one does not need to learn different instruction sets for different PLC manufacturers. The International Standard IEC 61131 is a complete collection of standards on programmable controllers and their associated peripherals. Section 26.4 of the standard defines a minimum instruction set, the basic programming elements, and syntactic and semantic rules for the most commonly used programming languages [17].

The standard was originally published in 1993, with a recent revision published in 2003. Guidelines for application and implementation of programming languages are presented in IEC 61131-8, a related document published in 2000 [18]. Many products already support three of the five languages. Vendors are not required to support all five languages in order to be IEC 1131-3 compliant. Note that the International
Electrotechnical Commission (IEC) is a worldwide standardization body. Nearly all countries over the world have their own national standardization bodies, that have agreed to accept the IEC approved and published standards.

26.8.1 Ladder Logic Diagrams

The ladder logic diagram (LLD) is the most common programming language used in PLC applications. LLD is a graphical language resembling wiring diagrams, as seen in Figure 26.11. Each instruction set is a rung on the ladder. Each rung is executed in sequential order for every control cycle. A rung is a logic statement, reading from left to right (Figure 26.12). Rungs can have more than one branch, as seen in the 4th rung of Figure 26.11.
As ladder logic was the first replacement for relay logic, it is most easily suited to Boolean I/O variables. However, over many years of use and refinement, the current LLD instruction set is quite large, reducing the need for other languages. Many special instructions are available for motion control, array handling, diagnostics, serial port I/O, and character variable manipulation. LLD is used in continuous and batch processes. Both off-line and on-line editing of individual rungs is possible, allowing changes to a running system (on the next cycle). User interfaces allow off-line programming and reviewing of LLD code at either PLC display panels or HMIs, or at remotely via PC software.

26.8.2 Structured Text

Structured text (ST) is a high-level programming language similar to BASIC, Fortran, Pascal, or C. Most engineers are familiar with structured programming languages. Structured text has standard program flow control command statements such as If/Then, Case, Do/While, Do/Until, and For/Next constructs. An example of a structured text is given in Figure 26.13. Most LLD, SFC, and FBD instructions are supported in ST.

While ST provides great flexibility in programming and editing, it is somewhat more difficult to read and follow logic flow, as compared with the more graphical languages. Thus, it is less maintainable over time.

26.8.3 Function Block Diagram

Function block diagram (FBD) is a highly visual language and is easy to understand because it resembles circuit diagrams. The graphical free-form programming environment is easy for manipulating process variables and control. A typical function block diagram is given in Figure 26.14. The foundation of the FBD program is a set of instruction blocks with predefined structures of inputs and outputs from each block. Different blocks are placed and connections are drawn to pass parameters or variables between blocks. Blocks are positioned and organized, based on the specific application, to improve readability.

Although simple instructions can take as much program space as complex instructions, the architecture simplifies program creation and modification. It is ideal for analog control algorithms, as it graphically represents control loops and other signal conditioning devices. This language is commonly found in distributed control systems (DCS) with continuous or batch process control.
26.8.4 Sequential Flow Chart

Sequential flow chart (SFC) is another highly visual programming language yielding applications that are easy to create and read. SFC is a graphical flowchart-based programming environment. Logical steps (blocks) are placed on the visual layout in an organized manner. Connections are drawn to determine execution flow of the program. An example is provided in Figure 26.15. This figure shows two branches from the start block of the program. Note: Blocks are descriptively labeled for case in following program logic. I/O and other functions are embedded within each block. Multiple branches allow for transitional and simultaneous execution flow. Position and organization of blocks are used to increase readability. Floating or linked text boxes provide application documentation (comments). Embedded structured text in action blocks directly improves readability and maintenance, while reducing the number of subroutine calls.

The flow chart structure is ideal for sequencing of machine states (e.g., Idle, Run, Normal, Stop). High-level program/subroutine flow management provides a more flexible approach to developing process sequences. SFC is ideal for machines with repetitive operations and batch processing.

26.8.5 IL: Instruction List

Instruction list (IL) is the most basic-level programming language for PLCs. All languages can be converted to IL, although it is most often used with LLD. IL resembles assembly language, using logical operator codes, operands, and an instruction stack. This language is difficult to follow and is not typically selected. The major application of IL is found in hand-held program for nonnetworked PLCs.

26.8.6 Selection of Languages

The language of choice is typically based on an organizational standard, existing code base, or personal preference. Typically a graphical language is preferred. However, different languages are preferred in specific instances. Ladder logic and function block diagrams are best suited for continuous operations, while sequential flow charts are more suited to operations using states and events. Structure text is suitable for either. When the majority of process variables are Boolean (on/off), ladder logic diagrams, sequential flow charts, and structure text are favored over function block diagrams.

Generally, one language is selected for the entire process control project. However, this is no longer required. Controllers that follow the IEC 61131 standard allow multiple languages to be used within the same programmable controller. This allows the programmer or control engineer to select the language best suited to each particular task.
26.9 Industrial Case Study

While it is impossible to completely detail an industrial control system, this section reviews a system architecture and layout for a specific industry case. The facility discussed is a chemical processing facility for creation of high purity materials. The process line is specified for development of process and product improvement. This requires interfacing of many new sensors and control algorithms on a continual basis. The process uses hazardous (reactive and combustible) chemicals. Thus, chemical leak monitoring must occur 24 hours a day. The control system is separated into two major subsystems: life safety and process control. The architecture is depicted in Figure 26.16.

The life safety monitoring system (LSS) must have a high degree of availability and redundancy as this is classified as a SIL3 (Safety Integrity Level 3) process [18, 19]. This is based on the potential consequence of injury to one or more people with a moderate risk for chemical leaks. A specifically designed life safety PLC (SIL1-3) with redundant processors and networking connections is selected. In the event of a fault in the control system, redundant control components take over and chemical leak monitoring continues.
A touch screen panel HMI is provided for status of alarms (no networking is provided through the HMI). Outputs of the LSS go to several places. In the event of a detected leak, outputs trigger alarms in the process area, starting safety measures (e.g., exhaust fans, bells, lights), removing power from chemical supply valves, signaling the process controller to shutdown, as well as alerting building life safety main controller for additional actions.

The separate controls for process and safety allow flexibility in the design and operation of the process. The process has several hundred I/O points, both analog and digital. Since this process is state-driven, a controller and software based on flow charts (SFC) is used [1]. Ethernet is selected as the process control/device bus for remote I/O controllers and HMI interfaces using a switch for collision control. While a direct parallel bus for I/O can be faster, the use of Ethernet allows for greater future system expansion and longer distances between I/O controllers. Motion control is accomplished via a separate (legacy) controller. Because process motions are not time critical (time constants for the process are in terms of minutes), RS-422 serial communication between the process controller and the motion controller is more than sufficient.

Data are recorded for batch processing locally and via data servers on the WAN/LAN. Local and remote PC-based HMIs allow viewing and operation of the process via the Ethernet connections to the process controller. HMI and process control software are provided from the hardware vendor in this case, yielding an integrated approach. The local HMIs also serve a SCADA function by monitoring and recording events and alarms.

### 26.10 Conclusion

This is only a brief overview of the ever-changing modern manufacturing automation. Other automation aspects include dynamic linking of I/O point data from tag servers and database servers. These tasks require interfacing organizations in a manufacturing enterprise. Manufacturing automation is rapidly integrating control, manufacturing, and business functions as manufacturers pursue improvements of products, processes, and profits in real-time.
References