CHAPTER

TWO

INTRODUCTION TO ON-LINE DATA ACQUISITION AND COMPUTER CONTROL

2.1 INTRODUCTION

In order to effectively design a modern process control scheme, one must have a basic understanding of the practical means of implementation. For example, computational algorithms suitable for on-line application must be carried out faster than real time on a typical process control minicomputer. Thus sampling frequency, computer computation times, and other factors must be considered in order to design a workable system. The successful installation of a modern control scheme also involves considerations of data links and computer command transmissions. Process variables must be measured and these data provided to the computer, while desired control actions must be computed and transmitted to the process. Figure 2.1 gives the simplified structure of such a computer control scheme. The measured variables are transduced, filtered, and sent by cable to a computer interface for data inputs. This measurement information is then incorporated into the computer control algorithm and control signal outputs generated. These signals are transmitted to local transducers and local controllers (such as valves and heaters), which cause changes in the manipulated variables of the process. The technology which accomplishes these tasks must be generally understood by the control engineer so that the designs are realistic and effective. Thus this chapter shall be devoted to providing the reader an overview of the considerations necessary in actually implementing a computer control scheme.

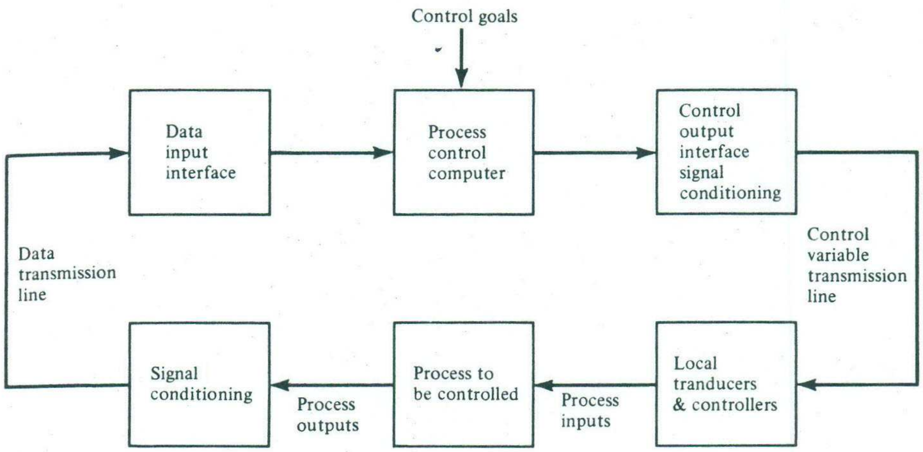


Figure 2.1 Structure of a computer process control scheme.

We shall begin by describing typical minicomputer architecture, capabilities, and peripherals. This should give one an idea of the computer resources which might be typically present in an industrial computer control installation. Secondly, we shall discuss the types of interfaces normally used for data acquisition and control-signal generation. These interfaces put constraints and specifications on the types of signals needed from the measuring devices. Finally, we consider commonly used sensing devices as well as techniques for transducing, multiplexing, and conditioning the data and control signals. This latter material should provide the control system designer with a grasp of the quality of the signals which must be dealt with in design, and can guide the choice of signal mode (e.g., analog voltage, analog current, or digital signal).

While this chapter is important in providing a well-rounded perspective for the reader, it is not crucial to the theory which follows; thus it is possible, if desired, to bypass this material for the moment and return to it later.

2.2 COMPUTER SYSTEM ARCHITECTURE

There are many different manufacturers of minicomputers designed for process control applications (a few of these are listed in Table 2.1) and many different types of architecture. However, the functions of the computer itself are much the same, and peripherals have been standardized to be compatible for most minicomputer mainframes. Thus it is possible to describe generally a typical minicomputer. We choose as our example the minicomputer in the Department of Chemical Engineering at the University of Wisconsin. This system, shown in Figs. 2.2 and 2.3, consists of a PDP 11/55 minicomputer with associated peripherals, interfaces, etc. We shall briefly describe each component of the system.

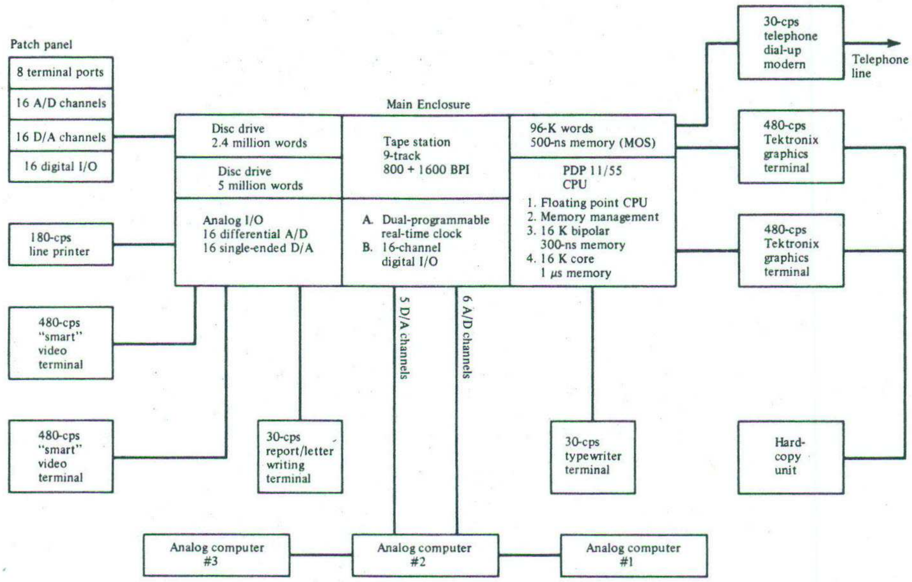


Figure 2.2 Chemical Engineering Minicomputer System, University of Wisconsin.

1. *Central processing unit* (*CPU*)—This is the heart of the minicomputer and has overall control of the system. The computer uses the *binary digits* (0 or 1) as the basic computational and communication code. This is accomplished electronically by manipulating and transmitting pulses which need be interpreted only as a 0 or a 1. The resolution of a machine is dependent on the number of binary digits (abbreviated *bits*) in the basic unit of information, called the *word*. Common minicomputers have 8-, 12-, 16-, or 18-bit words (see Table 2.1). Usually a smaller unit, a *byte* consisting of 8 bits, is used as well, because a byte suffices to specify characters (alphabet letters, numbers, control characters, etc.) according to industry standards (e.g., ASCII Code).

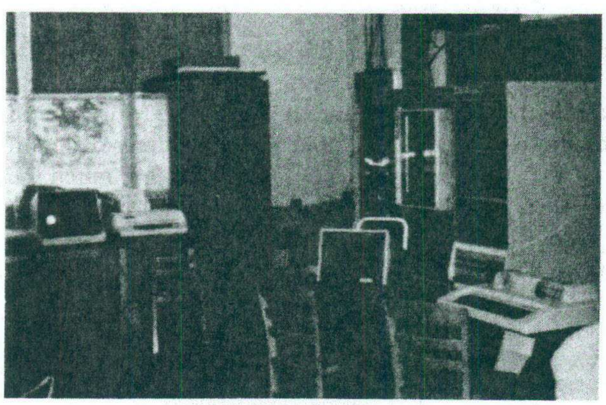


Figure 2.3 PDP 11/55 minicomputer at the University of Wisconsin.

Table 2.1 Some of the commonly encountered minicomputers for real-time applications*

Manufacturer	Computer (word length)	Maximum memory (words)	Speed: fixed-point addition, memory cycle time (μs)	Approximate price†
Data General Corp.	1. Nova (16 bits)	32 K	0.7, 0.7	$10 K
	2. Eclipse S, C (16 bits)	128–512 K	0.6, 0.7	$30 K
Digital Equipment Corp.	1. PDP 8 (12 bits)	128 K	3.0, 1.5	$10 K
	2. PDP 11/03 (16 bits)	32 K	3.5, 1.2	$ 4 K
	3. PDP 11/34 (16 bits)	128 K	2.0, 0.7	$ 9 K
	4. PDP 11/55 (16 bits)	128 K	0.3, 0.3	$40 K
Hewlett-Packard Corp.	HP 1000 (16 bits)	1024 K	0.9, 0.6	$10 K

* Current and comprehensive listings of available minicomputers may be found in trade journals, surveys, etc. (e.g., references [9, 10]).

† Price includes CPU and 32 K words of memory.

Normally 7 bits are used to specify the character and one is used for error checking (parity). Thus $2^7 = 128$ characters may be defined by a byte. Normally the bits in a word will be arranged in *octal* form by grouping them in 3s, e.g.,

$$110 \quad 010 \quad 100 \quad 001 \quad 101$$

Since 3 bits allows one to obtain the numbers 0 to 7, as shown in Table 2.2, octal integer notation is a convenient shorthand for addresses, machine instructions, etc. For example, the 15 bits shown above represent the octal number 62415, which is a much more efficient notation than the original binary form. A 16-bit machine (the most common type) thus can use 1 bit for sign (+ or −) and 15 bits to represent integer numbers. The maximum integer range then is from −77777 to +77777 octal, which corresponds to −32,767 to +32,768 decimal. These octal numbers are sometimes used to enter elementary computer instructions via switches on the CPU console (Fig. 2.4 shows a photograph of a typical console). Associated with the CPU, and working in parallel, many computers have a *hardware floating-point processor* which performs all floating-point arithmetic operations. For the PDP 11/55 machine shown in Figs. 2.2 and 2.3, which has 16-bit words, these floating-point operations involve numbers whose length is two words (32 bits). If double precision is called for, the floating-point processor uses 4 words (64 bits); however, double-precision operations require substantially more time to execute. The precision of an arithmetic operation depends on the number of bits carried along. For a 32-bit single-precision operation on the PDP 11/55, 24 bits are used for the fractional number and 8 bits for the exponent. In each case 1 bit is reserved for the sign, so that the exponent is limited to floating-point numbers between 2^{-2^7} and 2^{2^7} (approximately 10^{-38} to 10^{+38}

Table 2.2 Binary, octal, and decimal conversion

Binary	Octal	Decimal
000 000	0	0
000 001	1	1
000 010	2	2
000 011	3	3
000 100	4	4
000 101	5	5
000 110	6	6
000 111	7	7
001 000	10	8
001 001	11	9
001 010	12	10
001 011	13	11
001 100	14	12
001 101	15	13
001 110	16	14
001 111	17	15
010 000	20	16
011 000	30	24
100 000	40	32
101 000	50	40
110 000	60	48
111 000	70	56

Octal—binary conversion For 3 bits octal = $b_0 + 2b_1 + 4b_2$, where b_0, b_1, b_2 are binary digits.
Decimal-octal conversion Decimal = $a_0 + 8a_1 + 64a_2 + \cdots + 8^n a_n$, where $a_0, a_1, \ldots, a_n$ are the octal digits.

decimal). The fractional part, of 24 bits, has an error corresponding approximately to the least significant bit. With 1 bit for the sign and 23 for the fractional number, the relative error is approximately $2^{-23} \cong 10^{-7}$, so that a maximum of about 6 decimal digits are reliable in single-precision operations. For double precision, 8 bits are used for the exponent and 56 bits for the fraction, giving an approximate relative error of $2^{-55} \cong 10^{-17}$. This corresponds to a maximum reliability of about 16 decimal digits. For the floating-point processor used on the PDP 11/55, these operations are quite fast (~ 1 μs for addition, ~ 4 μs for division). Such high-performance floating-point processors, coupled with logic which allows operation in parallel with the CPU, make modern, inexpensive minicomputers rather fast in carrying out numerical calculations.

2. *Memory*—The memory devices associated with minicomputers may be of several different types (see Table 2.3). *Core memory*, which uses electromagnetic storage techniques, has a typical cycle time of ~ 1 μs—rather slow in the present context. However, because of its ferromagnetic character, core memory retains stored information even when the electric power fails. *Metal-oxide silicon* (*MOS*) *memory*, on the other hand, consists of simple

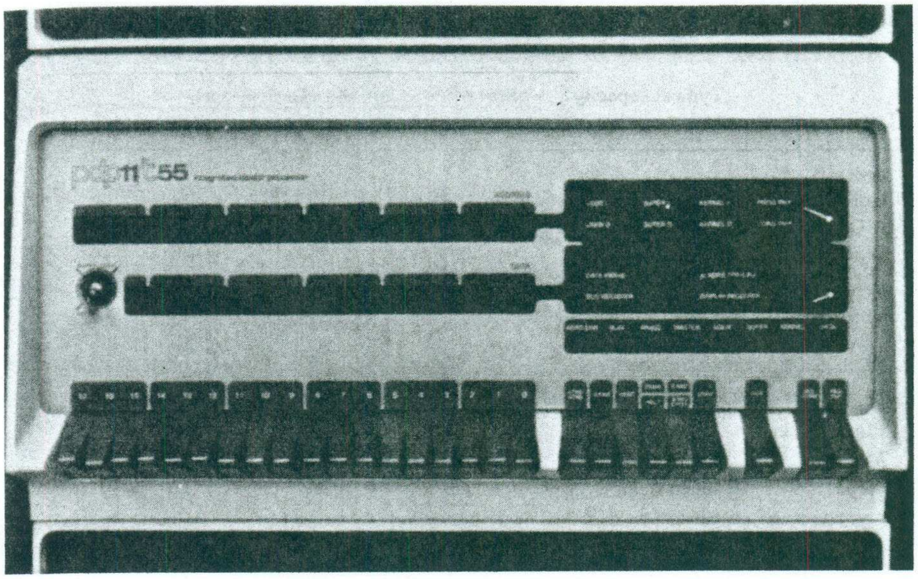


Figure 2.4 The PDP 11/55 CPU operator console (Reproduced by permission of Digital Equipment Corporation).

circuit semiconductor devices, which are both cheaper and faster (~ 500 n s cycle time) than core memory. *Bipolar transistor memory* makes use of more complex integrated circuits which are still faster (~ 300 n s cycle time). Both of these devices, however, lose the stored information in a power failure; thus battery power backup is often available. Special-purpose memory such as *read only memory* (*ROM*) is usually used to permanently store basic instructions for starting up the machine, input/output commands, etc., or for crucial, frequently used data files. These devices require special programming hardware in order to make any changes in the stored information. Memory can be purchased with and without extra *parity bits*. These extra bits are used to double check each byte transfer operation for transmission errors. This feature is useful in crucial computations and in diagnosing the specific locations of any memory failures. For the PDP 11/55 system shown in Figs. 2.2 and 2.3, all four types of memory devices are in use. Although only 64 K bytes (32 K words) of memory can be routinely addressed by a 16-bit machine, the PDP 11 series can make use of as much as 128 K words of memory by using many memory partitions, each of which may be as much as 32 K words in size.

3. *Mass storage devices*—There are a number of different types of mass storage devices available for minicomputers (see Table 2.3). These are used for inexpensive high-capacity storage with access times and transfer rates much slower than memory. *Magnetic tape* is the cheapest bulk storage device, but has the disadvantage that it has very slow access times (due to the time

Table 2.3 Capabilities of storage devices*

Device	Typical capacity/ 16-bit words	Typical access times	Typical transfer rates (words/s)	Approx. cost, $/1000 words
Memory:			(cycle time)	
Core	128 K	0.4 μs	1 μs	50–100
MOS	128 K	0.2 μs	0.5 μs	50–100
Bipolar	128 K	0.1 μs	0.3 μs	500–1000
Mass storage:				
Fixed-head disks	~1 million	~8 μs	10^5	20–50
Moving-head disks	1–100 million	35–70 μs	10^5	0.5–2
Floppy disks	256–512 K	500 ms	3×10^4	5–10
Magnetic tape	10–15 million	Several seconds	2×10^4 to 4×10^4	1

* Figures are for 1980.

required to move back and forth on the tape drive to find the desired files). For sequentially stored files, magnetic tape has reasonably fast transfer rates and is thus suitable for backup storage of large programs, data banks, etc. A much faster mass storage device is the *rotating disk*, which comes in several types (fixed-head, moving-head, and floppy). Most of the disks have removable packs or cartridges so that interchangeable data files, program libraries, etc., are possible. *Fixed-head disks* usually have a small capacity but short access times and high transfer rates. These devices are used for fast swapping of files between memory and disk. *Moving-head disks* have much slower access times but transfer rates comparable to those of fixed-head disks and much higher capacity. *Floppy disks* are low-capital-cost, small-capacity, low-performance mass storage devices in common use with microcomputers. Figure 2.5 shows a photograph of the magnetic tape unit, the movable-head disk and drive, and the floppy-disk drive used with the PDP 11 systems at the University of Wisconsin.

4. *Real-time clock*—The scheduler and timekeeper of the computer system is the *real-time clock*. This allows the computer to schedule tasks to be performed on a regular basis, such as taking data or changing control variables at specified time intervals, and timesharing the CPU's attention between various terminals and other peripherals. Each time the computer is powered up, the correct date and time are entered, and the real-time clock keeps track of chronological time after this initialization. A different means of sharing the CPU's attention is through *interrupts*. Each task running on the computer has a certain *priority* level; when it needs the attention of the CPU (for example, to record a data point), the task generates an *interrupt*, and if this task has a higher priority than the task currently being serviced by the CPU, the new

Figure 2.5a Mass storage devices used on PDP 11/55 minicomputer. A: Magnetic tape, B: Moving head disks.

Figure 2.5b Mass storage devices used on LSI 11 minicomputer. Floppy disks plus disk drive.

task is immediately serviced. Thus one may establish a hierarchy of priorities and interrupts in the computer and have the most urgent tasks (such as data taking) serviced immediately while those less pressing (such as printing on the line printer) must wait. Typical waiting times are a fraction of a second, so that the user is not usually aware of waiting in a queue.

5. *Input/output interfaces for data*—These interfacing devices deal with either *digital* or *analog* signals. *Digital I/O* interfaces are usually capable of either (*a*) *parallel* two-way simultaneous transmission (for example, 16 bits input and 16 bits output for a 16-bit machine) at speeds comparable to those within the computer system, or (*b*) one-way *serial* transmission (16 bits in only one direction at once for a 16-bit machine) at specified transmission rates. *Parallel-transmission* interfaces are typically used for high-speed communication between minicomputers, to multiplexers, or to other high-speed digital logic devices. *Serial-transmission* interfaces are used for communication with slower digital devices, such as terminals or graphical display units. Serial-transmission rates are indicated as *baud rates*, where baud rate = 10 × (# characters/s). Typical rates range from 30 to 960 characters/s, depending on the device being interfaced (see Table 2.4). Both types of digital interface devices can be put under program control and the information analyzed by individual bits, bytes, or words, depending on the application. Table 2.5 lists some typical applications of digital I/O interfaces.

   Analog I/O interfaces involve *analog-to-digital* (A/D) and *digital-to-analog* (D/A) *conversion*. A/D converters convert an analog voltage signal lying within a specific range (such as ±10 V, 0 to 5 V, etc.) to an integer number. The maximum size of the integer number, and thus the resolution of the conversion, depends on the number of bits handled by the converter. The range of the integer I is given by $I_{\text{range}} = 2^N$, where N is the number of bits. A 12-bit converter with an input voltage range of $-10 \text{ V} \le V \le +10 \text{ V}$ *differential* is shown in Fig. 2.6. The converter handles a *differential* signal because the reference voltage is also measured. A *single-ended input* would have only one voltage converted referenced to the computer ground voltage. The maximum integer range for the 12-bit converter is $2^{12} = 4096$. This

Table 2.4 Some communications peripherals

Peripheral	Communication rates	Typical capital cost
Line printers	150 characters/s–1200 lines/min	\$3000–\$20,000
Typewriter terminals	10–30 characters/s	\$1000–\$4000
Video terminals	240–960 characters/s	\$1000–\$10,000
Storage scope terminals	480–960 characters/s	\$3000–\$8000
Punched cards	250–1200 cards/min	\$3000–\$10,000
Paper tape	10–300 characters/s	\$3000–\$5000
Magnetic tape	10^4–10^5 characters/s	\$5000–\$15,000

Table 2.5 Some typical applications of data I/O interfaces

Type of interface	Typical connections and data
Digital inputs	From other computers Digital instrumentation Communications terminals Switch settings Relay positions Multiplexer status Alarms Counters Logic devices
Digital outputs	To other computers Commands to relays and solenoids Commands to switches Alarm messages Stepping motors Sequencing of multiplexers Logic devices Commands to digital instrumentation Communications terminals Hard copy and graphical output devices Various control signals
Analog inputs	From analog computers Analog instrumentation: Temperatures Pressures Flows Composition
Analog outputs	To analog computers Graphical plotters Oscilloscopes Process parameter changes Control signals: Heater controls Set-point changes Valve drivers

number of integers is chosen to run between $-2048 \leq I \leq +2047$, where $I = 0$ falls between the negative and positive integers. For the positive range (2047), the *resolution* of the converter is the voltage range divided by the intervals between integers, or

$$\text{Resolution} = \frac{10\text{ V}}{2047} = 0.00489\text{ V}$$

This gives an *expected error* of $\pm\frac{1}{2}$ the resolution, or ± 0.00244 V. In this case the *relative error* in conversion is approximately given by

$$\text{Relative error} \cong \frac{0.00244\ \text{V}}{|\text{measured voltage}|}$$

so that more precision is obtained if the measured voltages are close to the full-scale voltage (10 V in this example). This can be accomplished through *signal conditioning*, which will be discussed in the next section. The throughput speeds of A/D converters are routinely 50,000–100,000 conversions/s, and high-performance models can achieve even higher rates. As an illustration of A/D conversion, suppose the 12-bit converter shown in Fig. 2.6 measures a voltage and reports the integer 1261 as a result of the conversion. To determine the actual measured voltage, one uses the formula

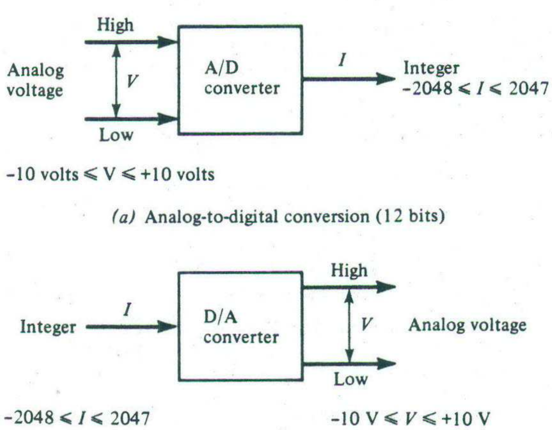
$$V_{\text{measured}} = \frac{10\,I}{2047} = \frac{12{,}610}{2047} = 6.16023\ \text{V}$$

where for this input voltage, the relative error is

$$\text{Relative error} = \frac{\pm 0.00244}{6.16023} = \pm 0.000397 = \pm 0.0397\%$$

a rather small value.

D/A converters operate as just the reverse of A/D conversion. An integer with range 2^N is placed in a register and converted to an analog voltage output. The 12-bit D/A converter shown in Fig. 2.6 has a voltage output range of $-10\ \text{V} \le V \le +10\ \text{V}$, corresponding to input integers ranging from $-2048 \le I \le +2047$. Thus the resolution and expected error of this D/A converter are the same as those of the A/D converter discussed above. To demonstrate how one programs the converter to obtain a desired voltage output, let us assume we wish to output 3.5 V. To choose the input



(a) Analog-to-digital conversion (12 bits)

(b) Digital-to-analog conversion (12 bits)

Figure 2.6 12-bit A/D and D/A conversion.

integer, one uses the formula

$$I_{\text{input}} = \frac{2047\,V_{\text{output}}}{10} = \frac{(2047)(3.5)}{10} = 716.45$$

which is rounded to $I_{\text{input}} = 716$. This gives an actual output voltage of

$$V_{\text{output}} = \frac{716}{2047} \times 10 = 3.49780 \text{ V}$$

very close to the desired voltage. Throughput rates for D/A converters are also very high, typically on the order of a million conversions/s.

Most computer installations will have multiple channels of A/D and D/A converters in order to allow several different projects to interface with the computer at any one time. For the computer system at the University of Wisconsin, there are 16 channels of 12-bit differential A/D and 16 channels of 12-bit D/A.

6. *Communications peripherals*—There are many different types of devices for communicating between the user and the computer system (see Table 2.4 and Fig. 2.7). For printed output, there are *line printers*, *typewriter terminals*, and *report-writing terminals*. Line printers can have very high rates of output, while typewriter and report-writing terminals are usually much slower. Graphical output is possible on many devices, but the most common types are *video terminals*, *storage scope terminals*, and *x-y plotters*. Video terminals (or cathode-ray tubes) require either a microprocessor or CPU attention for constantly refreshing the screen in graphical mode, but allow real-time motion of the graphics and even multicolored graphs. Storage scope terminals require no CPU-screen refreshing, but require that all or most of the screen be erased before a new graph can be drawn. Both types of graphics devices

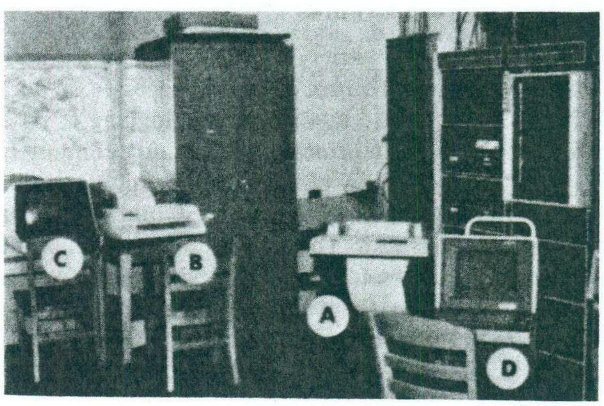


Figure 2.7 Some typical communication peripherals. A: Lineprinter, B: Report-writing terminal, C: Video Terminal, D: Storage scope graphics terminal.

can be linked to hard copy units to provide a paper copy of the screen image. Plotters are of many different types, but most provide report-quality finished graphs.

Input communications devices include *typewriter*, *video*, and *storage scope terminals* as well as *punched cards*, *paper tape*, and *magnetic tape*. Removable disk cartridges, which were discussed above, may also be considered to be I/O communications devices. For long-range I/O, it is possible to communicate with the computer over telephone lines, usually through a terminal at a remote location.

The rates of communication of these various devices are shown in Table 2.4. Modern minicomputers are bypassing the requirements for punched cards (and keypunching) by having large numbers of low-cost terminals on-line in a timeshared mode. In this way, all program input can be made directly into the computer with very small load on the CPU. Similarly, paper tape is losing ground to magnetic tape and disks as an input/output and storage medium, largely due to its very slow transfer rates and mechanical problems.

2.3 DATA ACQUISITION AND CONTROL

In order to use a real-time computer to implement a process control algorithm, one must be concerned with the hardware realization of data acquisition and control. There are many different computer configurations which will accomplish these goals. One may use a large central real-time computer linked to many labs or processes, or have a digital computer network involving many smaller microcomputers, each dedicated to a specific process or lab. We shall begin this section with a discussion of such computer structures.

Computer Data Acquisition and Control Networks

Depending on the specific application, a wide variety of tasks may be required from a data acquisition and control network: control of plant processes or pilot plants, servicing of terminals in various labs and control rooms, logging of data from analytical and/or research laboratories, interfacing with analog computers for hybrid computation, etc. To illustrate some of these applications, let us consider the network used within the department of chemical engineering at the University of Wisconsin and depicted in Fig. 2.8. The hub of the network is the PDP 11/55 minicomputer described earlier, functioning under a multiuser, time-sharing operating system. The links to the various remote stations are basically of two types, either with or without a remote microcomputer. At present the Mössbauer spectroscopy lab, the bioengineering lab, the three pilot plant locations, and the remote terminal links all come directly to the central minicomputer without involvement of a microcomputer. The polymer physics lab and the Raman spectroscopy lab, on the other hand, have PDP 11/LSI

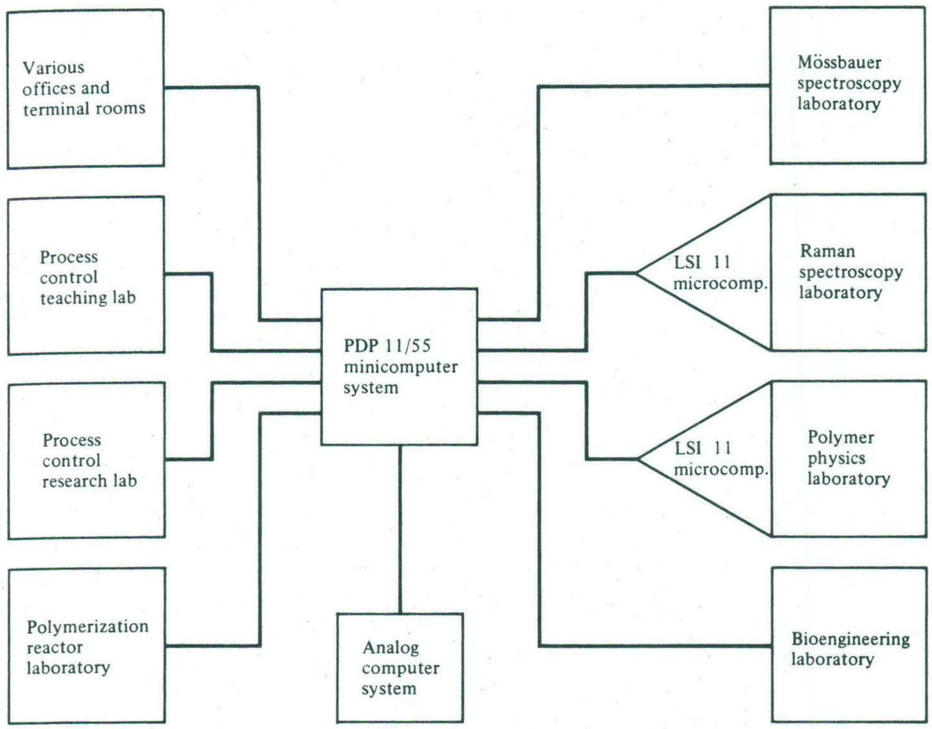


Figure 2.8 The Computer Data Acquisition and Control Network at the University of Wisconsin.

Table 2.6 Capabilities of the computers in the computer network shown in Fig. 2.8

Computer	Memory (words)	Disk storage (words)	Magnetic tape storage (words/tape)	Computing speed	Approximate cost
PDP 11/55 system shown in Fig. 2.2	128 K	2.5 million	15 million	Floating point: add, 1 μs; divide, 4 μs	$90,000
PDP 11/LSI system shown in Fig. 2.9	32 K	256 K	—	Floating point: add, 60 μs; divide, 160 μs	$ 9,000

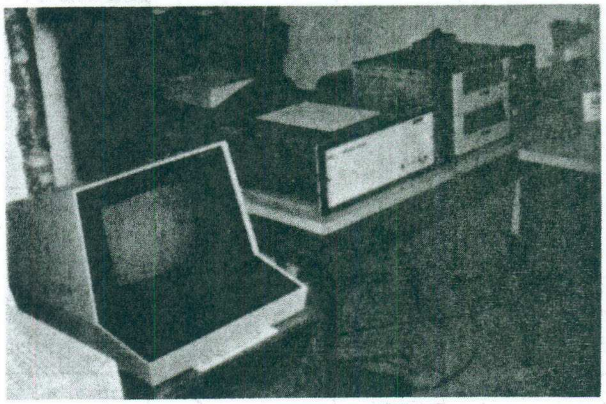


Figure 2.9 LSI 11 microcomputer with video terminal, twin floppy disks, 32 K words memory, A/D and D/A converters, and real time clock.

microcomputers, which provide local control of the instruments and acquire and store the raw data. Through the links to the central minicomputer, these microcomputers are able to call upon the large file storage (disks), permanent data storage (magnetic tape), and computing power of the larger machine for data manipulation. Table 2.6 illustrates the capabilities of the various computers in the network, while Fig. 2.9 shows one of these PDP 11/LSI microcomputers.* Note that while the microcomputer is relatively inexpensive and has a moderate amount of memory and mass storage, computing speed for the LSI 11 is 40 to 60 times slower than that for the 11/55. Thus it is better to use the 11/55 for any substantial numerical calculation. Through different, but compatible, operating systems, the computers in the network are able to transfer programs, data, calculated results, etc., back and forth so as to optimize the performance of the entire computer system.

Through a link between the 11/55 and the analog computer system, hybrid computation may be carried out. As noted in Fig. 2.10, this requires A/D and D/A conversion to handle the analog computer inputs and outputs. Some scaling may also be involved because the voltage ranges of the two computers may be different. For example, the University of Wisconsin network has an analog computer operating over the range -100 V to $+100$ V, while the A/D and D/A converters have the range -10 V to $+10$ V. This means that the analog input signals must be amplified by a factor of 10 and the analog output signals divided by the same factor. The most common application of this hybrid computation facility is for control algorithm testing. A process (distillation column, chemical reactor, etc.) may be readily simulated on the analog computer and will respond in real time in much the same way as the actual process. This allows one to program the computer control algorithm on the digital

* Microcomputers similar to the LSI 11 are often used as industrial process control computers where relatively small numbers of control loops are involved.

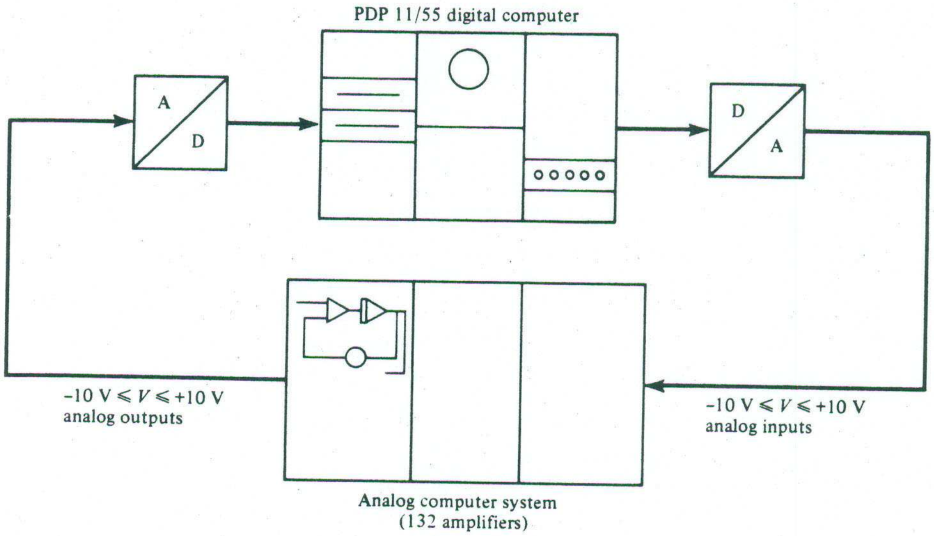


Figure 2.10 Analog-digital links for hybrid computation.

computer and test the performance in real time with the help of the analog computer. Examples of this approach will be given in forthcoming chapters.

Transducers for Data Acquisition and Control Signal Output

It is useful to consider some of the transducers more frequently encountered in the process industries. A representative sample of data acquisition sensors for some of the more common process variables is listed in Table 2.7. Depending on the sensor chosen, the primary output can be either digital, pneumatic (air pressure), current, or voltage. Because the digital computer can receive only voltage or digital signals, the pneumatic and current signals must be converted to one of these acceptable inputs. In practice this involves special transducers for pressure/voltage conversion (P/E transducer) and current/voltage conversion (I/E transducer, which is usually just a high-precision resistor).

In a similar way control system actuators come in several different forms. Table 2.8 includes some of the most common types. Notice that in the process industries, the most frequently seen actuators are for flow control. This is because heating and cooling processes often involve such factors as coolant flow-rate adjustment, steam injection rates, or furnace fuel flow rates. Similarly, pressure adjustments usually require controlling inflow or outflow from a vessel, as does liquid level control. Although the digital computer outputs are limited to digital or voltage signals, some of the control actuators require current or pneumatic inputs. Thus special transducers are needed to provide for voltage/pressure (E/P) and voltage/current (E/I) conversion.

Table 2.7 Some common data acquisition sensors

Process variable	Device	Output	Comments
1. Temperature	Thermocouple	mV	Cheap, rugged
	Resistance sensors (e.g., thermistors)	Resistivity	High sensitivity
	Oscillating quartz crystal	Oscillating frequency	High sensitivity
	Radiation pyrometer	mV	For high-temperature applications
2. Pressure	Diaphragm	Capacitance	Low pressure
	Bellows		
	Bourdon tube	Inductance	Both low and high pressure
		Photoelectric	↓
		Resistance	↓
		Reluctance	↓
		Strain gauge	From low to very high pressures
		Piezoelectric	
	Ionization gauge	Emf	Ultra-low pressure
3. Flow	Pitot tube	Differential pressure	
	Orifice	Differential pressure	
	Impaction	Mechanical deflection	
	Turbine	Rotational speed	
	Electromagnetic field	Generated emf	For difficult flow measurements
	Neutron bombardment	Nuclear radiation	For difficult flow measurements
	Ultrasound	Doppler shift	
	Hot-wire anemometry	Resistivity	High precision
4. Liquid level	Float	Mechanical deflection	
	Sonic resonance	Resonant frequency	
	Conductivity	Resistance	
	Dielectric variability	Capacitance	
	Thermoelectric probe	Resistivity	
	Nuclear radiation	Radiation intensity	
	Photoelectric cell	Voltage	
	Head pressure	Differential pressure	

Table 2.7 Some common data acquisition senors (Continued)

Process variable	Device	Output	Comments
5. Composition*	Potentiometry	mV	
	Moisture content:		
	Hygrometry	Resistivity	
	Psychrometry	Wet- and dry-bulb temperatures	
	Chromatography	mV	Long analysis times
	Refractive index	mV	
	Ultrasound	Sound velocity	
	Spectroscopy: UV, visible, IR, Mössbauer, Raman, atomic-emission, x-ray, electron, ion, magnetic resonance	Intensity	Many of these techniques are not yet suitable for on-line applications
	Polarography	mV	
	Conductimetry	Resistance	
	Mass spectrometry	mV	
	Differential thermal analysis	Heat transferred	
	Thermogravimetric analysis	Mass changes	

* There are so many composition detection devices that only a few are noted here.

Table 2.8 Some typical control system actuations

Control action	Device	Input
Electrical heating	Electric furnace	Voltage or current
Flow adjustment	Pneumatic valve	Air pressure
	Solenoid valve	Voltage, current, or digital pulse
	Motor-driven valve	Digital pulses
	Variable-speed pump	Voltage or current
	Fluidic control valve	Pressure
Alarms	Lights, bells	Digital pulse
On-off signals	Relays, switches	Digital pulse

Signal Conditioning

An additional consideration in data acquisition and control is conditioning the sensor or actuator signal so that it will have high accuracy, have low noise levels, and makes efficient use of the interface equipment. Usually the steps involved in signal conditioning are multiplexing and amplification of the signals, suppression of noise on the signal, and selection of a transmitting mode for the signal. We shall discuss each of these considerations in turn.

In some applications, there will be many similar measurements at a remote location, and it is usually most efficient *to multiplex* these at the source and transmit them sequentially over only a few lines. As an example, consider 10 thermocouple measurements from a process, each with a voltage signal of ~ 10 mV. First of all, one would not wish to run 10 lines and use 10 channels of our A/D converters just for these signals, so multiplexing is necessary. In addition, such low-voltage signals will be heavily corrupted by noise in transmission and will have very low resolution when they reach the A/D converter; thus *amplification* as well as *multiplexing* is needed. Figure 2.11 illustrates two possible approaches to solving this problem. Scheme (*a*) involves amplification of each low-level signal (requiring N high-gain amplifiers) up to ~ 5 V, then multiplexing with solid-state relays. By multiplexing high-level signals, one minimizes the effects of noise introduced by the relays; however, this is at the cost of an expensive amplifier for each signal. Scheme (*b*), on the other hand, multiplexes first and then requires only a single high-gain amplifier. This is more efficient, but requires care in selecting a low-noise-level multiplexer. Such a device usually requires relays with mercury-wetted contacts in order to avoid corrupting the low-level voltage signal. Both types of multiplexing schemes are known to be successful, and one of them will be illustrated by an example given below.

Noise suppression, either before or after signal transmission to the computer, is also an important part of signal conditioning. There are times when relatively high-frequency noise, such as 60 Hz coming from power lines or motors, appears on the signal. Fortunately, in the process industries, such high frequencies are very rarely contained in the desired signal; thus *filtering* to remove these frequencies can be a very successful means of noise suppression. A simple *low-pass filter* with frequency response characteristics like those shown in Fig. 2.12*a* will often suffice. Other common types of filters include *high-pass filters*, which reject low frequencies, and *notch filters*, which pass only signals within a specified frequency band. Figure 2.13 shows an example of the performance of a low-pass filter applied to a thermocouple signal afflicted with motor noise.

The last part of signal conditioning involves a decision on how the signal is to be transmitted to the computer. There are several alternatives available for signal transmission:

1. **Voltage signals**—These work well over short distance cables (up to ~ 300 ft) where voltage losses and cable capacitance do not cause signal deterioration.

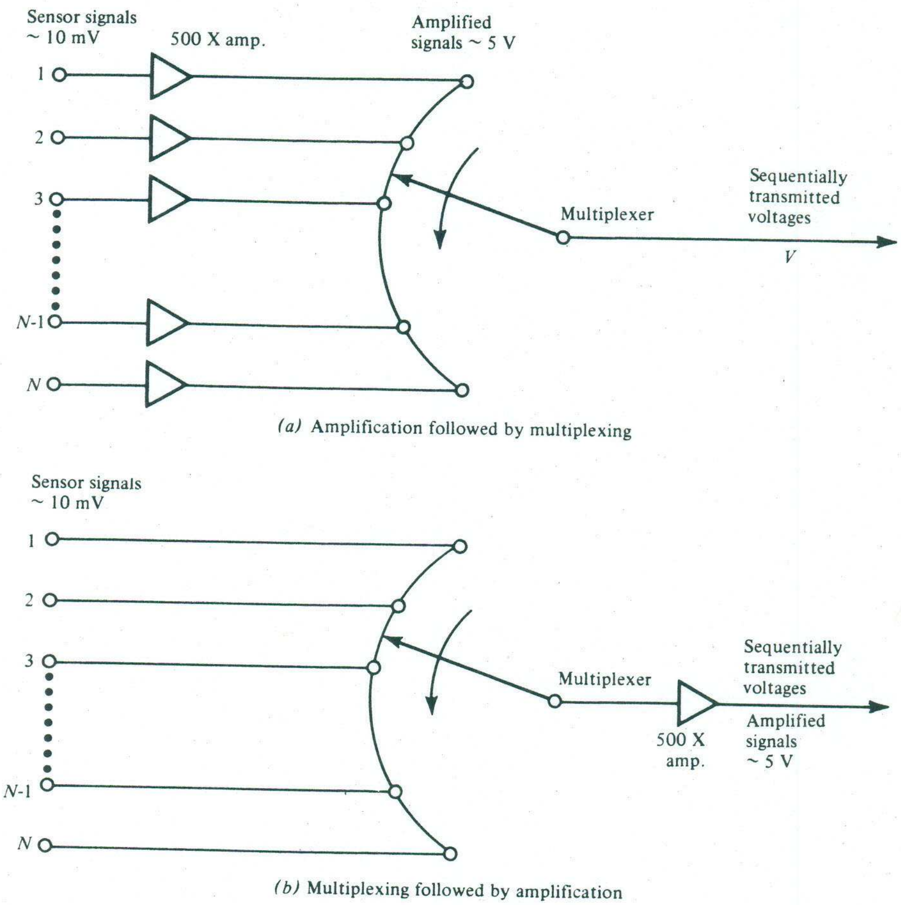


Figure 2.11 Two different strategies for multiplexing and amplifying low voltage signals.

2. **Current signals**—This method works well with hardwire connections over longer distances but requires special transmitters and I/E conversion at the receiving end.
3. **Digital signals**—These signals can be transmitted most easily and are the natural choice when the original signal is digital. Even voltage signals may be transmitted in a digital mode through D/A conversion at the sending location and A/D conversion at the receiving end. Digital signals have the advantage that they are less susceptible to noise problems and can be transmitted over extremely long distances through telephone connections.

Of course, the selection of transmission mode should be made based on the particular application, noise levels to be encountered, and distances to be covered.

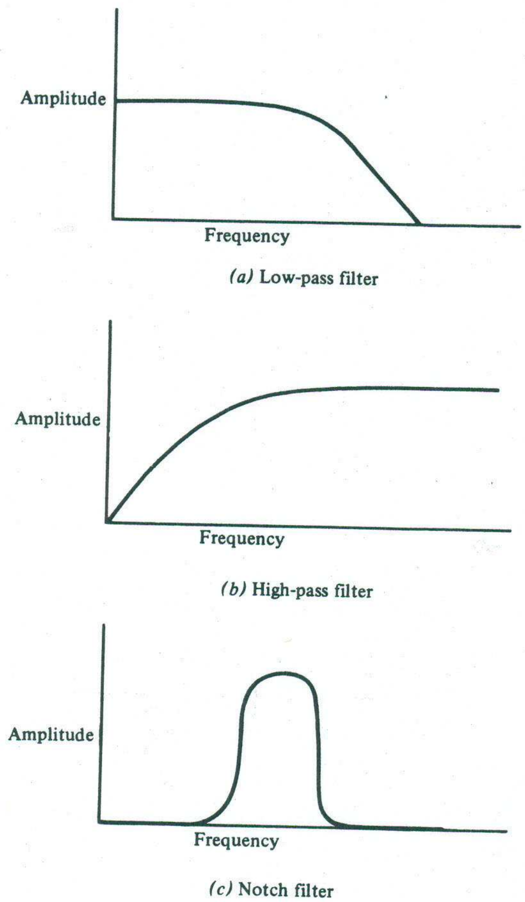


Figure 2.12 Frequency response characteristics of some common filters.

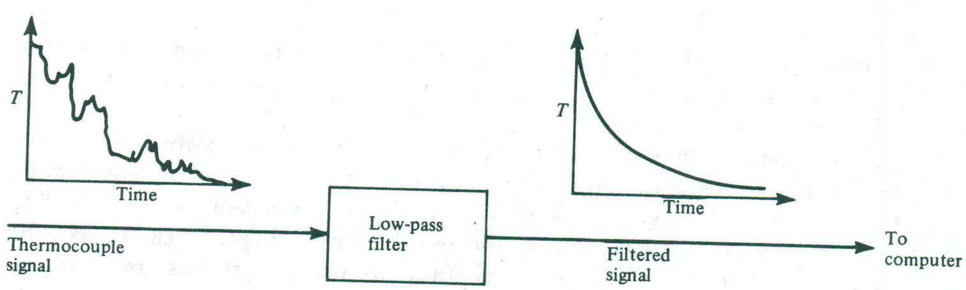
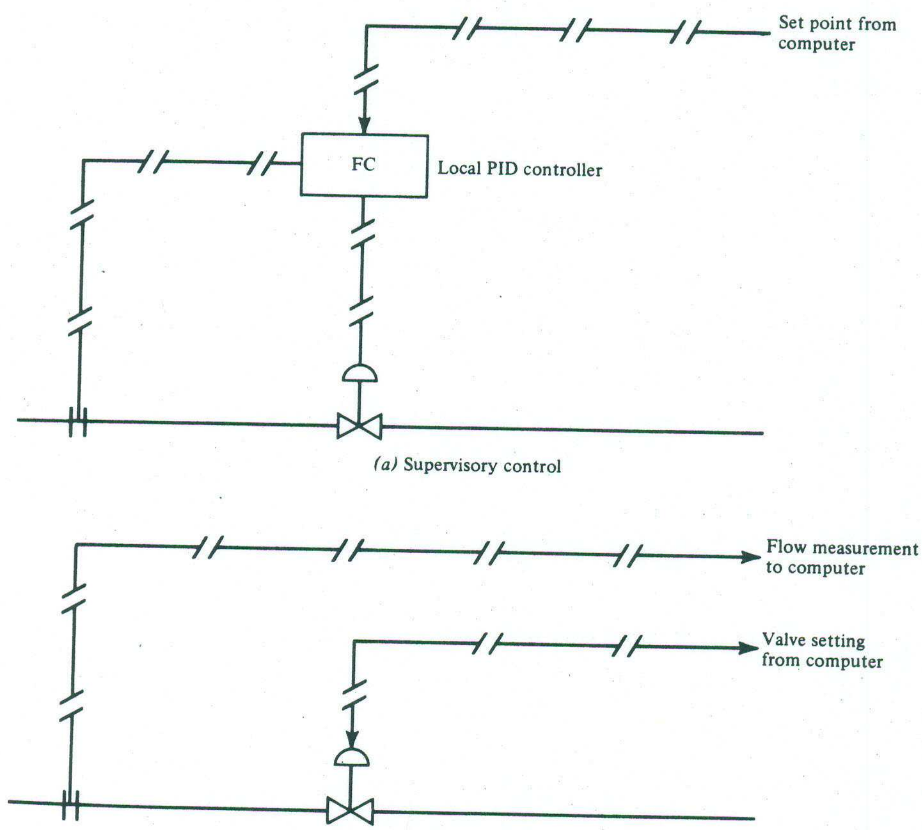


Figure 2.13 Performance of a low-pass filter.

Modes of Computer Control

Computer control is usually carried out in one of two modes: *supervisory control* or *direct digital control* (*DDC*). Supervisory control, illustrated in Fig. 2.14*a*, involves resetting the set point of the local controller according to some computer algorithm. Thus the computer control scheme need only supervise and coordinate the actions of the local controllers. Direct digital control, by contrast, requires that all the controller action be carried out by the digital computer. Measurements are sent to the computer and compared with the set point; then the computed control action is transmitted to the actuator. This is illustrated in Fig. 2.14*b*. For DDC the computer samples the flow measurement at discrete instants of time and sends as control signals step changes in the control valve stem position. The time interval Δt between samples and controller changes must be chosen with care. If Δt is taken too large, the controller performance will deteriorate, while having Δt too small will put an unnecessarily high



(*a*) Supervisory control

(*b*) Direct digital control

Figure 2.14 Supervisory and direct digital control of a fluid flowrate.

computational load on the computer. Both supervisory control and DDC are in wide use in industrial applications, and both may be readily used to implement modern control strategies. In most processes, the local controller dynamics (whether analog or digital) are short compared with the time constants of the process under control; thus most sophisticated control strategies are implemented in a cascade fashion—involving programmed set-point changes in the local controllers. Hence, it matters little whether these local controllers are analog or digital. As an example, suppose the flow-rate control loop in Fig. 2.14 is a steam flow into a heat exchanger. Suppose further that an optimization algorithm calculates the optimal steam flow-rate program $F(t)$. Clearly this program could be implemented by either supervisory control or DDC when the steam-valve dynamics are fast compared with the heat exchanger time constants. In either case the flow-loop set point would be programmed to follow the function $F(t)$.

Before proceeding to more detailed examples, it is useful to point out that we have provided here only a brief summary of the considerations important in on-line data acquisition and computer control. For more detail, the reader should consult the references at the end of the chapter.

2.4 SOME EXAMPLES

To illustrate some of the principles discussed in the earlier sections, several example case studies shall be presented. It is hoped that these will give the reader a practical grasp of what is involved in interfacing processes to real-time computers.

Computer Control of the Pressure in a Gas Storage Tank

A simple interfacing problem arises for a gas storage tank system which is used in the teaching laboratory at the University of Wisconsin.

A drawing of the interfacing scheme is shown in Fig. 2.15. A bourdon tube transducer is used to measure the pressure in the tank. This transducer is linear and transforms a pressure range of 0 to 60 psig to a voltage range of 0 to 50 mV. This low-voltage signal is then amplified to a voltage range of 0 to 10 V. This high-level voltage is then sent to the computer room (~ 300 ft away) by shielded cable.

The digital computer must be programmed to receive the pressure measurements and calculate control signals based on some controller algorithm. For example, a simple PID controller in discrete form would be

$$u(t_k) = u_0 + K_c\left\{[y_d - y(t_k)] + \frac{\Delta t}{\tau_I}\sum_{n=1}^{k}[y_d - y(t_n)] + \frac{\tau_D}{\Delta t}[y(t_k) - y(t_{k-1})]\right\} \tag{2.4.1}$$

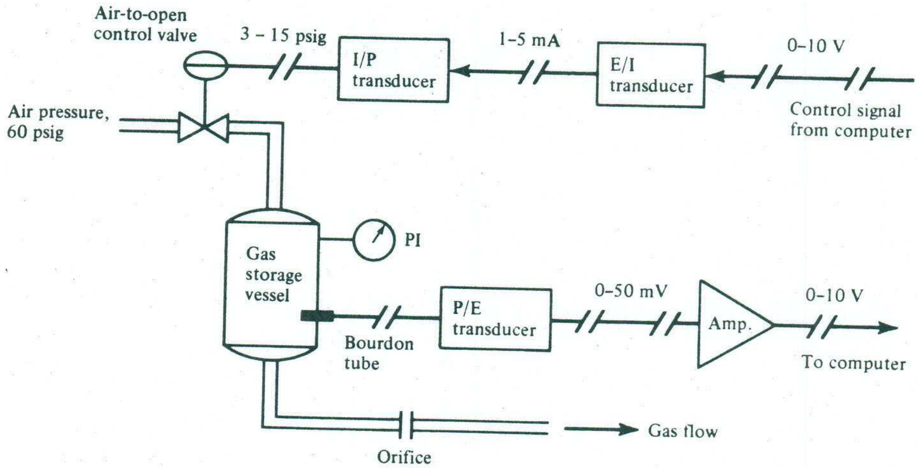


Figure 2.15 Interfacing for the gas storage vessel.

where $u(t_k)$ is the control signal at time t_k, u_0 is the zero-error control signal, K_c is the proportional controller gain, τ_I is the integration time, and τ_D is the derivative time. The measured output (pressure in this case) at time t_k is denoted by $y(t_k)$, and y_d is the controller set point. The quantity Δt is the sampling/control time interval, and is usually selected to be short compared with the time constants of the process.

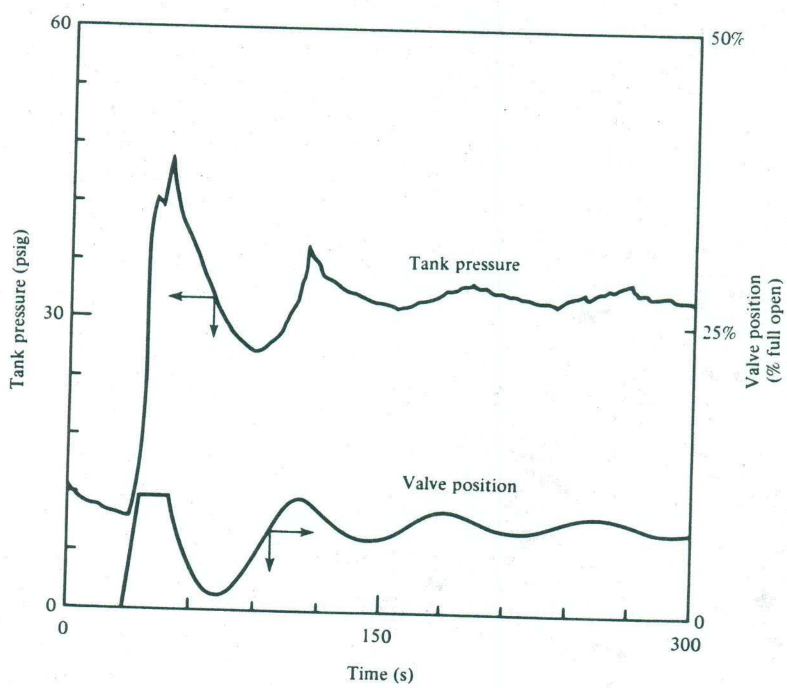


Figure 2.16 Dynamic behavior of the gas storage vessel under DDC control.

The control signal, which is transmitted as an analog voltage, must be converted to a 3- to 15-psig air-pressure signal in order to drive the pneumatic valve. This could be done in one step with an E/P transducer; however, to make use of existing equipment, this conversion is done in two steps, as shown in Fig. 2.15.

After installation, the transducers must be calibrated. For example, 0 to 60 psig corresponds to 0 to 10 V analog voltage for our system, while 0 to 10 V analog control signal corresponds to 3 (full shut) to 15 psig (full open) for the control valve. Because the analog signals were sent over shielded cables for relatively short distances and were not in the vicinity of power lines or motors, no noise suppression devices were needed in this application. Figure 2.16 illustrates the dynamic behavior of the implemented control scheme, showing both the measured tank pressure and the computer-specified control valve position. The control time interval Δt was taken to be only a few seconds, so that the DDC step changes in valve position appear as a continuous curve in the plot. The actual valve stem position does not exactly follow the specified control signal due to the dynamics of the valve itself. Nevertheless, the control system performs adequately, as might be expected.

Computer Control of an Instrumented Steel Ingot

A more complicated scheme of process interfacing can be seen in the following example. It is desired to control the temperature distribution of a cylindrical steel ingot by manipulating the power input to each zone of a three-zone furnace. This is a laboratory model of a metallurgical heating furnace; it is much more heavily instrumented with thermocouples than is possible in practice. This instrumentation is to allow an absolute measure of the temperature distribution in order to evaluate the performance of proposed control algorithms. A photograph of the furnace system is shown in Fig. 2.17. Thirty-two thermocouples are placed axially and radially in the ingot, and each has a 5- to 10-mV voltage

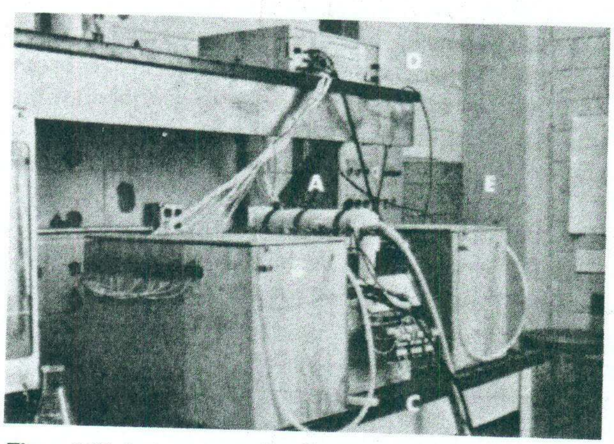


Figure 2.17 An instrumented steel ingot in a three-zone furnace: A, furnace; B, thermocouples; C, furnace heater relays; D, multiplexer and heater controller; E, link to computer.

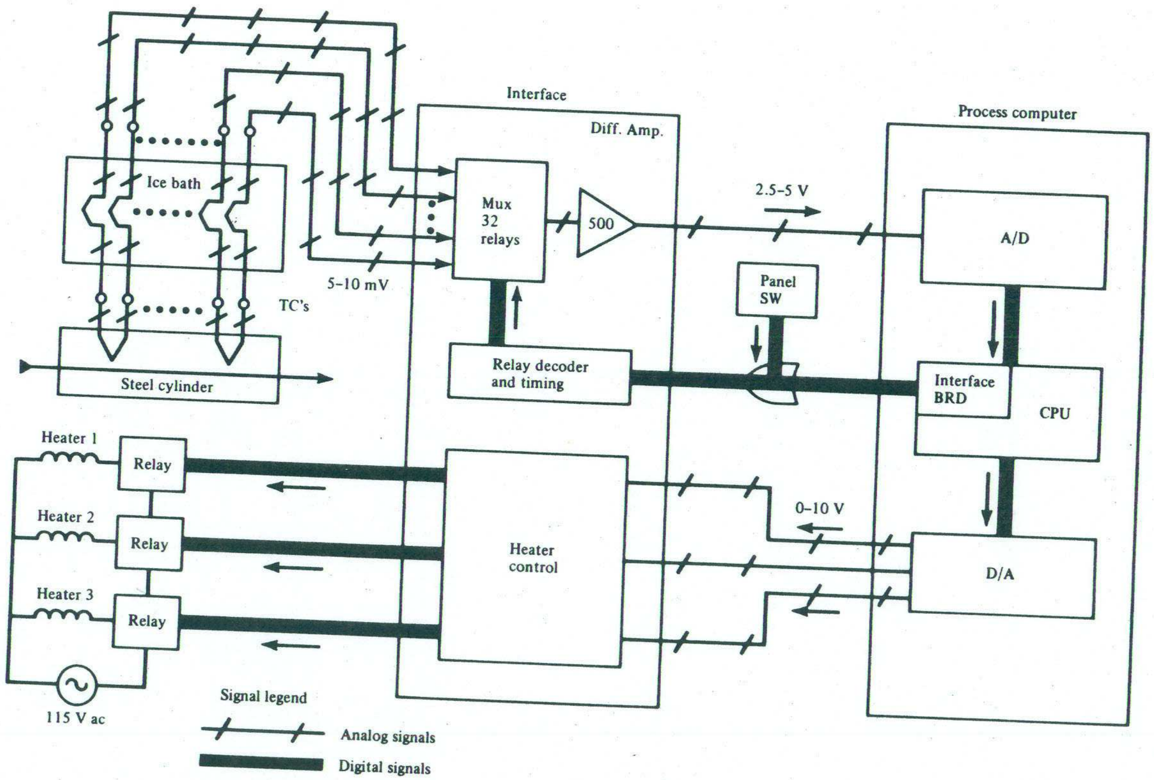


Figure 2.18 Data acquisition and control signal structure for the instrumented steel ingot.

signal. These signals must be multiplexed and amplified before they are sent to the computer. Similarly, the control signal for each zone of the three-zone furnace must be converted to a power input for the electrical heaters.

Figure 2.18 shows the data acquisition and control signal design used for the system. The 32 thermocouples are passed through a reference temperature bath* and then to an interface device. The interface device multiplexes the 32 signals through 32 mercury-wetted relays, which introduce very little noise to the signal. The multiplexing is carried out through digital logic and under digital signal control from the computer. Thus the multiplexing rate and sampling interval are under program control. After multiplexing, the signal passes through a high-quality amplifier with a gain of 500. This brings the 5- to 10-mV signal up to 2.5 to 5 V, and this may be transmitted and processed by the A/D converter without undue error. Through multiplexing, the 32 thermocouple signals require only one channel of the A/D converter—a very efficient use of computing resources. To allow each relay of the multiplexer to properly settle before reading the signal, one must not switch the multiplexer too fast. In the present application it was necessary to allow 1.5 ms per relay for reliable multiplexing. Nevertheless, this allowed all 32 thermocouples to be scanned in less than a tenth of a second. For most process control problems, where the process time constants are minutes to hours, such multiplexing times are negligible.

The control signals for the three-zone furnace are sent through three channels of the D/A converter as analog voltages in the range 0 to 10 V. The interface device has digital logic, which converts the input voltage to an arcless relay controller which controls the power input to the heaters. The conversion is linear, with 0 V control signal corresponding to zero power input and 10 V control signal causing full heater power input.

The interfacing scheme shown in Fig. 2.18 performed well and allowed a detailed study of state estimation and control algorithms for the system. One of the case histories discussed in Chap. 6 demonstrates the performance of this interface design.

REFERENCES

1. Andrew, W. G.: *Applied Instrumentation in the Process Industries*, vols. I, II, III, Gulf Publishing, Houston, 1974.
2. Considine, D. M.: *Process Instruments and Controls Handbook*, 2d ed., McGraw-Hill, New York, 1974.
3. Ewing, G. W.: *Instrumental Methods of Chemical Analysis*, McGraw-Hill, New York, 1975.
4. Finkel, Jules: *Computer-Aided Experimentation*, Wiley, New York, 1975.

*As an alternative to ice baths, there are inexpensive electronic compensators which work equally well.

5. Holman, J. P.: *Experimental Methods for Engineers*, McGraw-Hill, New York, 1971.
6. Korn, G. A.: *Minicomputers for Engineers and Scientists*, McGraw-Hill, New York, 1973.
7. Norton, H. P.: *Handbook of Transducers for Electronic Measuring Systems*, Prentice-Hall, Englewood Cliffs, N.J., 1969.
8. Perone, S. P., and D. O. Jones: *Digital Computers in Scientific Instrumentation*, McGraw-Hill, New York, 1973.
9. Theis, D. J.: *Datamation*, p. 73, (Feb. 1977).
10. *DataPro Reports on Minicomputers*, DataPro Research Corp., October 1978.
11. Harrison, T. J.: *Handbook on Industrial Computer Control*, Wiley, New York, 1972.
12. Harrison, T. J.: *Minicomputers in Industrial Control*, ISA Publ., Pittsburgh, 1978.

PROBLEMS

2.1 Convert the following binary numbers to octal numbers:

(*a*) 100 111 000 111 101
(*b*) 101 010 101 110 011
(*c*) 111 001 110 100 011

2.2 Convert the following octal numbers to decimal numbers:

(*a*) 747 (*c*) 100
(*b*) 440 (*d*) 556

2.3 A Fortran program consists of 100 lines with approximately 25 characters per line. Estimate the number of words of memory in a 16-bit machine required to store this Fortran code.

2.4 In responding to interrupts from various peripheral devices, a computer manufacturer recommends the following priority levels:

Terminal—4 Line printer—4
Real-time clock—7 A/D converter—6
Disk drive—7
Magnetic tape unit—7

Discuss why some of these peripherals are assigned higher priorities than others.

2.5 A computer has a 12-bit A/D converter with −10-V to +10-V span. Determine the analog input voltages and relative error for the following integers coming from the converter:

(*a*) 570 (*c*) −960
(*b*) 2000 (*d*) 25

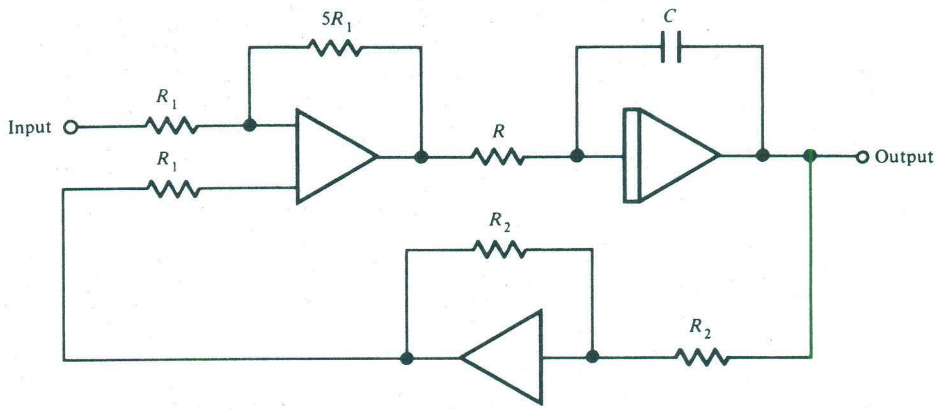
2.6 Provide the answers to Prob. 2.5 for a 10-bit A/D converter with a −5-V to +5-V span and for the following integers:

(*a*) 450 (*b*) −200 (*c*) 100 (*d*) 25

2.7 It is desired to output the following voltages through a 12-bit D/A converter with a −10-V to +10-V span. Determine the integer that should be loaded into the D/A in each case.

(*a*) −1 V
(*b*) +2.7 V
(*c*) +8.7 V
(*d*) −6 V

2.8 The following analog circuit has been suggested as a simple low-pass filter. Write down the differential equations corresponding to this circuit and indicate how this circuit might accomplish the filtering action. Can you suggest values for R and C that will eliminate 60-Hz noise without disturbing the 1-Hz signal?



2.9 Two computers are linked over a 300-baud telephone line. One computer would like to transfer a Fortran program to the other computer over this telephone link. The Fortran program in question has about 500 lines of code with approximately 30 characters per line. Estimate how long it will take to transfer this program. An engineer suggests connecting the two computers via a 9600-baud cable link. How long would the program transfer take in this case?