CHAPTER IX

Computer Process Control

JOHN PESCHON

LUCAS PUN

SANJOY K. MITTER

CONTENTS

1. Introductory Comments 457
1.1 Examples of Computer-Controlled Systems 458
1.2 The Process: Definition and Boundaries 459

2. Computer 462
2.1 The General Purpose Digital Computer 463
2.2 Characteristics of Typical Digital Control Computers 464
2.3 Input and Output Equipment 464
2.4 Programming 466
2.5 The General Purpose Analog Computer 467
2.6 Characteristics of a Typical Analog Control Computer 468
2.7 Input-output Equipment and Programming 468
2.8 Memories and Look-up Tables 469
2.9 Temporal Operation of On-line Computers 470

3. Control 471
3.1 Data Logging 472
3.2 Monitoring and Alarm 472
3.3 Sequence Control 472
3.4 Dispatch of Information 473
3.5 Regulation and Optimization 473

4. Static Optimization 474
4.1 Introduction to Linear Programming 476
4.2 Nonlinear Model Optimization 481
4.3 Optimalizing Control 482
4.4 Optimization by Memory 484

5. Dynamic Control and Optimization 485
5.1 Digital Servomechanisms and Control Systems 486
5.2 Noninteracting Control 487
5.3 Feedforward 489
5.4 Optimum Control of Dynamic Processes 491
5.5 The Calculus of Variations 493
5.6 The Bang-Bang Problem 495
5.7 Dynamic Programming 495
5.8 The Maximum Principle 497
5.9 Control by Fast Model 498

6. Summary and Conclusions 499

Appendix: Digital Instrumentation and Conversion Equipment 501
A.1 Introduction 501
A.2 Transducers 501
A.3 Telemetering 503
A.4 Nature of Transducer Output Signals 503
A.5 Organization of the Conversion Equipment Connecting Plant to Computer 504
A.6 Sampling and Filtering 504
A.7 Scanning and Routing of Information 504
A.8 Amplification and Signal Conditioning 504
A.9 Analog-digital Conversion 507
A.10 Digital-analog Conversion 511
A.11 Amplification 513
A.12 Actuation 513

Bibliography 514

1. Introductory Comments

The expression "computer process control" is being used more and more frequently in the control literature to designate production, transportation, administration, etc., processes which are connected permanently—or on-line—to one or several computers programmed to perform a variety of information processing tasks in order to achieve more desirable process behavior. A few years ago, the meaning of computer process control was much narrower; it primarily referred to certain large production processes of the oil, chemical, steel, and electric utility industries relying on on-line computers to carry out fairly simple data logging, monitoring, and control tasks.

Any discussion of the subject of computer process control as it is being interpreted today remains of necessity superficial. The field of *computers* has become so vast that detailed understanding has become impossible even for the specialist. In addition to the general purpose digital and analog computers, a variety of special purpose, as well as hybrid, machines have been developed. The *processes* under consideration cover the whole range of industrial, commercial, and military activity. Mathematically speaking, these processes might be described by differential, difference, or stochastic equations, assuming that a satisfactory mathematical model can be found at all. Finally, the *control* tasks which the computer is assigned transcend those considered in the classical texts on automatic control. The general purpose digital computer is an information processing machine capable of a wide range of computation and decision operations carried out at high speed and with great accuracy. In an actual system, therefore, it would be a mistake to limit the computer's function to that of conventional controllers; the question which the designer should ask himself is:

How would a group of intelligent humans govern the process assuming that sufficient time were available to perform the required monitoring, computation, information, and rationally based decision tasks in order to ensure optimal performance?

After this question has been answered, the designer can proceed by exploring how these tasks must be reformulated to accord with the special characteristics of the computer, and deciding whether there is sufficient economic incentive to replace an operator or a group of operators by a machine. The important point is that modern computers are capable of a much greater *variety* of assignments than conventional controllers and that this should be taken into account if an economically sound design is to be evolved. A direct consequence of this observation is that the synthesis of computer-based systems not only requires knowledge of the modern theory of automatic control, but also of certain other scientific disciplines developed in order to comprehend and thereafter govern complex situations. Most notable among these are operations research, group dynamics, econometry, and—common sense.

There exist at the present time several hundred processes permanently connected to large digital or analog computers. In many of these installations results have been disappointing in that the savings expected were not, in fact, attained. There is little doubt that large computers are destined to play a major role in making industry and commerce benefit from automation. It is agreed, however, that their installation is economically justified only if imaginative use of their capabilities is made. *It does not suffice to place a computer next to a process to have computer process control;* many man-years of process analysis and system design are required to produce an economically viable installation. Due to the staggering complexity of the subject matter, the present chapter can only provide a broad discussion of trends illustrated by much simplified examples. A few of these examples correspond to systems which are in actual everyday operation. In most cases, however, they correspond to experimental systems which are expected to become operational in the late nineteen sixties. One should carefully distinguish between experimental and operational systems and acknowledge the fact that the latter require economic motivation and guaranties of reliability which may be very difficult to provide, particularly in the intellectually gratifying case of dynamic optimization.

1.1 Examples of Computer-Controlled Systems. Although the expression "computer process control" became widespread only after 1955 when the first general purpose digital computers were permanently connected to certain large industrial processes of the oil, chemical, steel, and electric utility industries, it should be noted that special purpose computers were already in operation before World War II. National telephone systems have relied for a long time on special purpose digital computers called "markers" to replace the manual operator in making and breaking connections between any two of millions of subscribers.

Certain fire-control systems which were operational during World War II employed special purpose analog computers to predict target position at the time of arrival of the shell on the basis of optical or radar measurements and to calculate the appropriate gun deflection.

In the early nineteen fifties, the electric utility industry, which had already developed remarkable control systems before the war, introduced special purpose analog computers to work out the most economic power setting of each of hundreds of interlinked generating units, taking into account their individ ıal efficiencies as well as the cost of transporting electric power from production to consumption centers.

At about the same time, large-scale military systems such as the Continental Air Defense System came into being. Here, the defense weapons were directed with the help of a network of computers and telecommunication links so as to

ensure optimal protection with the available resources in specific tactical situations.

Automobile traffic control systems comprising a centrally located analog or digital computer to direct the traffic lights at some of a city's intersections were introduced in 1952 to improve the flow of traffic. Airline traffic control systems capable of monitoring the positions of thousands of airliners, computing their routes to avoid in-flight collision, and shortening waiting time before landing, are presently being envisaged.

Long-range missiles naturally make use of highly sophisticated computer systems, both ground- and missile-based, to control the position and velocity coordinates during the powered part of the flight. The very stringent accuracy and reliability specifications of some of these systems have not been duplicated as yet in industrial and commercial applications.

1.2 The Process.

Definition and Boundaries. Computer process control is not a sharply defined subject; one reason for this is that the processes envisaged for control by computer vary widely, and frequently cannot be described by the deterministic mathematical models discussed in the previous chapters.

In order to illustrate this most important point, let us consider the major departments of a representative manufacturing company. There will be people concerned with market prediction, customer relations, accounting, inventory, etc., and there will be production machines. It is convenient for the purposes of this discussion to classify these activities into a three-level hierarchy, as shown in Fig. 1. The first or top level comprises the responsibilities of company management, such as decisions pertaining to investment, relations with labor,

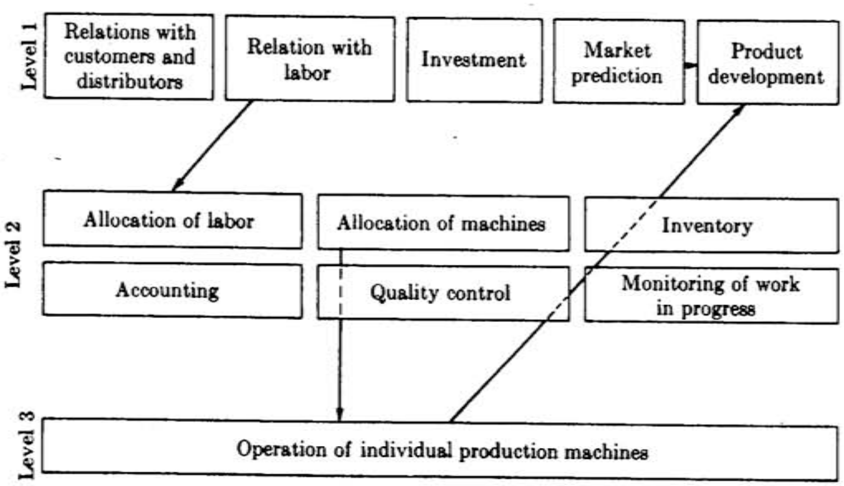

FIGURE 1. Classification of the subprocesses of a manufacturing company.

contract negotiation, market prediction, choice of new lines of product to be developed, etc. At the second level, available labor, material, and production equipment are procured and allocated by the executive staff so as to satisfy customer orders promptly and efficiently. The third and lowest level contains individual machines needed in the over-all production process. It is noted that the proportion of men to machines decreases sharply from the first to the third level: at the first level, there are practically no machines and decision is primarily based on intuition and common sense; at the third level, individual machines play a predominant role. In the manufacturing industries, automation naturally started at this level where controlling action can be automatically generated on the basis of fairly simple considerations pertaining to the process state and to a purely technical objective, such as zero steady-state error. At the second level, the automatic generation of controlling action became possible recently for certain activities, such as inventory control. Digital computers, rather than conventional analog control equipment, are generally required at this level. While at the first level machines are being used to collect and disseminate data, man remains responsible for evolving the proper correcting action.

A similar three-level pattern can be found in almost any undertaking involving a large number of men and machines, and the same comments can be made with regard to the utilization of machines for control. This three-level classification could be and in many countries has been extended upwards to include supervision by regional and national government. For the purposes of this discussion, it suffices to consider three levels of activity; this avoids political argument and circumvents the difficulty of our eventually attempting to optimize the universe by computer.

The three levels cannot function independently from each other and without taking into consideration the perturbing effects due to the "outside world." The body of policies evolved by management, the first level, is based on the state of the subsequent levels whose objectives must in turn be adjusted to fit these policies. Some of these relations are shown in Fig. 1: the allocation of labor at the second level depends on the labor policy postulated by management. Similarly, the choice of a new line of products depends on market prediction as well as on the characteristics of the available production machines. Digital computers are well suited to collect and interpret data and to strengthen coordination between departments by routing messages.

The outside world also influences the decisions taken at each level; at the top, a change in the nation's fiscal policy may provoke correcting action whereas at the third level, such action may be triggered by a change in the composition of raw material. These outside influences are perturbations or disturbances.

The operation of a manufacturing company, or of any other commercial or military entity involving a large number of men and machines, thus

constitutes a single highly complex process which, for reasons of convenience, is subdivided into many interlinked *subprocesses* or departments. Effective control of any one subprocess requires that the relations with other subprocesses be accounted for. Control schemes which take into consideration these relations are referred to as "total" or "integrated." While at the present time, this terminology is ambitious, to say the least, it is likely that the highly automated industrial, commercial, and military systems of the future will make use of a hierarchy of interlinked computers each of which is assigned the task of monitoring and controlling the activities of one subprocess in accordance with the information received from adjacent computers.

The processes of the third level, which primarily comprise machines, are describable by deterministic mathematical models—arrived at either by analytical considerations or by experiment—and are susceptible of being controlled in accordance with the principles of the classical theory of automatic control. The processes of the second level, which comprise men as well as machines, are being characterized in a much more rudimentary and uncertain way by the deterministic and stochastic models of operations research. It is the scope of system engineering (see Chapter X) to automate routine decision at the second level. Mathematical techniques are available to assist the first level, but it would be unreasonable to assume that the activities of this level can be summarized by a mathematical model, since the human element of intuition, personal acquaintances, and original thought plays such a predominant role.

It has been argued that at each of the three levels, the decision processes taking place are conceptually similar in that controlling action is based on an assessment of the difference between the actual state and the desired state. This abstraction is correct, but the reader is cautioned against accepting that the principles of automatic control of demonstrated usefulness at the third level can be easily adapted to the first level.

What is important for the control engineer is the observation that processes of the third level, which are his primary concern, should not be treated separately from each other and from processes of the second and first levels. There is much evidence today that the utilization of computers at the third level, in particular digital computers, can only be economically justified if these relations are accounted for in the statement of the objective function and the treatment of the perturbations.

The present chapter is primarily concerned with the control of processes at the third level, of which the mathematical models are of the general form; see Chapter I, Section 3

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{m}, \mathbf{u}) \\ \mathbf{y} &= \boldsymbol{\varphi}(\mathbf{x}) \end{aligned} \tag{1.1}$$

in the *dynamic* case and

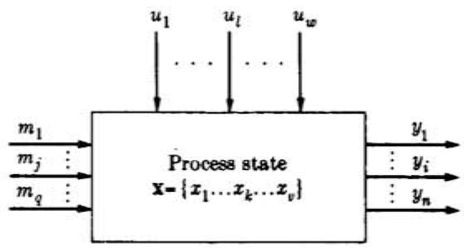

FIGURE 2. Mathematical model of a multivariable process. The x_k are state variables, the y_i are output variables, the m_j are actuating variables, and the u_l are disturbances.

$$\begin{aligned} 0 &= \mathbf{f}(\mathbf{x}, \mathbf{m}, \mathbf{u}) \\ \mathbf{y} &= \boldsymbol{\varphi}(\mathbf{x}) \end{aligned} \tag{1.2}$$

in the *static* case. The notation will be that of Fig. 2.

As a result of the trend to operate machines at ever increasing efficiency, it is generally necessary to add to Eqs. (1.1) and (1.2) a set of constraints of the form

$$\mathbf{m}^L \leqslant \mathbf{m} \leqslant \mathbf{m}^H \tag{1.3}$$

as well as other constraints affecting $\mathbf{x}$, to be discussed later. The restriction (1.3) on $\mathbf{m}$ adds singularly to the difficulties of system synthesis.

The definition of a suitable *objective* to be satisfied by the control system is perhaps more difficult than the elaboration of a mathematical model, especially when it is impossible to consider the complex process as a whole. Subprocesses of the third level are generally controlled for reasons of convenience in such a way that a mathematically tractable function F of the state $\mathbf{x}$ is minimized or maximized.

The desirability of spending much effort on the elaboration of an accurate mathematical model and on the definition of a meaningful objective is emphasized throughout the literature on computer process control. It has been observed that the benefit derived through better process understanding frequently outweighs that resulting from actual control by computer. One of the major advantages of computer process control may thus be the requirement that classical processes be quantitatively assessed and that accepted design and operating practices be reconsidered as a result of this analysis!

2. Computer

The types of computers available for process control introduce almost as much variety as the process itself. In addition to the general purpose digital and analog machines, a large number of special purpose computers as well as

various types of "hybrid" machines have become available for control applications. It is somewhat difficult to decide when a *controller* deserves the qualificative of *computer*, since such conventional gear as PID controllers are already capable of performing certain mathematical operations, namely multiplication by a constant (Proportional control), integration (Integral control), and differentiation (Derivative control). Relay networks have likewise been used for a long time to carry out logical operations and thus relieve the human operator from routine decision. PID controllers and switching networks are not normally considered to be computers although a sufficient number of them could perform most of the control tasks now envisaged for computers. Therefore, price per unit seems to be the only acceptable criterion: if it exceeds several thousand dollars, we may safely speak of a computer. Far more important than this somewhat immaterial question of terminology is the observation that no discontinuity of concept occurred when the first computers were introduced to control military and industrial processes and took over many of the functions previously performed by conventional gear.

Special purpose computers are based upon the same principles as general purpose machines, but are designed with a *specific* application in mind; they cannot be adapted easily to perform other tasks, primarily because their "program" is relatively fixed. *Hybrid* computers, of which there exist at least a dozen different types, were developed in order to combine in the same unit the chief advantages of the digital computer—accuracy, versatility, and memory capability—and of the analog computer—computing speed, ease of programming, and relatively low cost, [17].† Since the special purpose and the hybrid computers are derived from either or both the general purpose digital or the general purpose analog computer, our discussion will be limited to these two basic types of machines. It will be assumed that the reader is familiar with the rudiments of their principles of operation. The purpose of our discussion on computers will be to draw attention to certain factors which are particularly relevant to process control applications.

2.1 The General Purpose Digital Computer.

The general purpose digital computer constitutes one of the most momentous inventions of all time. It makes possible the *processing* or transformation of *information*. The term "computer," which implies capability for calculation, is too narrow, since a variety of other information processing tasks—storage and retrieval of information, logical decision, adaptation, and learning, etc.—are possible.‡ As regards the capability for computation it is helpful to visualize the digital computer as a conventional desk calculator to which an automatic device designed to press the proper keys at the right time in accordance with a prerecorded "program" has been added, though the speed at which the "arith-

† Numbers in brackets refer to the bibliography at the end of this chapter.

‡ In the French language, the terms "machine à traiter l'information" or "ordinateur" are often preferred to "calculateur" for precisely this reason.

metic units" of modern computers performs the basic arithmetic operations of addition and multiplication has introduced a difference of *magnitude* as well as of *kind* in the type of computations one might attempt on a desk calculator.

2.2 Characteristics of Typical Digital Control Computers. Although in principle any sufficiently reliable general purpose digital computer developed for scientific computation or business data processing can be used as a control computer, at least a dozen machines have been developed especially for the control market. Most of these machines fall into the $100,000–300,000 range, and special care is taken to ensure high reliability, 99% of operating time being a commonly accepted figure. Ease of programming is not as important as in the case of the "scientific" computer, since program changes are less frequent.

The chief characteristics of the central part of a digital computer system—this is often called data processor—are computing speed and memory capacity. To define speed, it is customary in the computer industry to quote "add-time" and "access-time." This may suffice for a rough comparison of two data processors, but if a detailed study is to be made, it is necessary to find out what exactly the manufacturer understands by add-time and access-time; usually, the time required to add two randomly chosen numbers and to perform the ancillary operations of retrieval and storage is several times larger than the quoted add-time would lead one to believe. Moreover, it is advisable to find out how fast other arithmetic operations (substraction, multiplication, division, move, and test) can be effected.

The capacity refers to the number of "words" of specified length that can be stored in the fast-access memory, i.e., in the core or drum memory. As before, it is again advisable to inquire as to what exactly is meant by capacity, since word length, nature of the code, etc., determine to what extent the user can put to profit the stated capacity. Whereas speed is a relatively fixed quantity, the capacity can generally be adapted within certain limits to fit the customer's needs.

The following numbers provide an *order of magnitude* of the chief characteristics found in a large number of control computers:

add-time in μsec: 100;
access-time, assuming core storage, in μsec: 20;
word length, in bits: 30;
capacity, in words: 4000.

2.3 Input and Output Equipment. The price of a computer may vary by a factor of 2 depending on the nature and versatility of the "peripheral" equipment which permits the flow of information to take place between the process, the operators, and the computer. The main types of input and output equipment in process control applications are shown in Fig. 3.

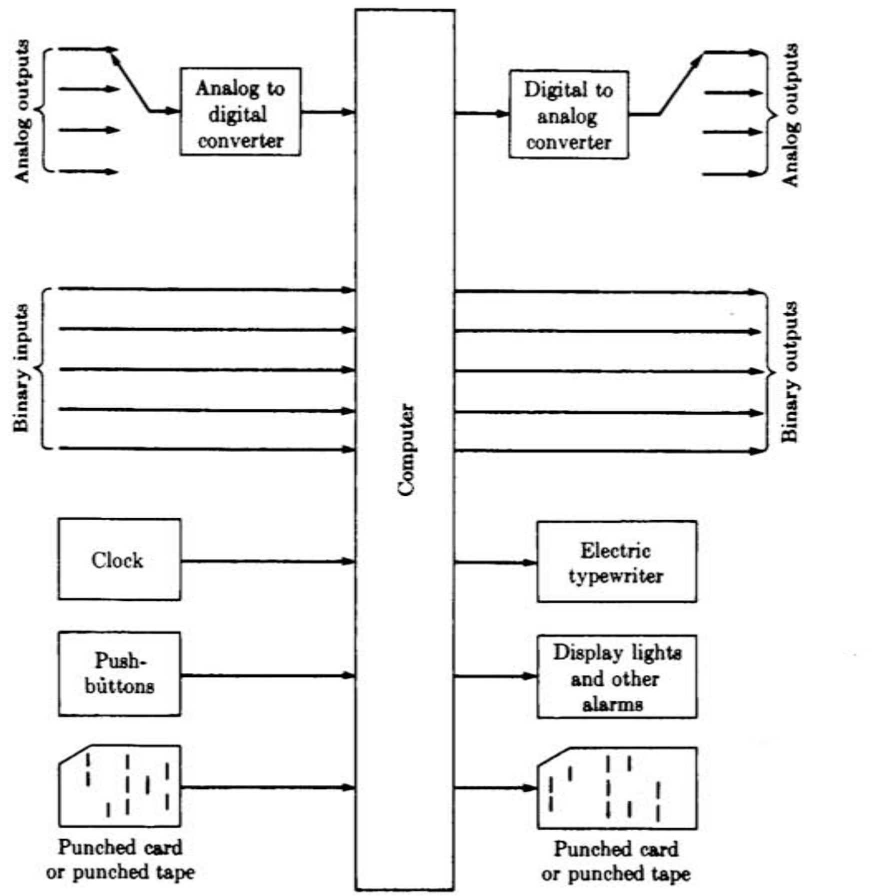

FIGURE 3. Principal inputs and outputs of digital process control computer.

The *analog*† *inputs*, which are assumed to be available under the form of electric voltages, are scanned at a rate compatible with the conversion speed of the analog to digital converter of which the coded output signals are accepted by the computer. There may be several hundred analog input lines; typical scanning rates are 20 lines per second.

† The adjective "analog," as opposed to "digital," is being used quite freely in the literature to denote any variable the magnitude of which is *not* given by a code. The outputs of pulse-frequency, pulse-amplitude, and pulse-position modulators are thus analog signals, whereas the outputs of pulse-code modulators are referred to as digital; in the majority of the computer-based systems, this is also equivalent to binary.

The *binary inputs* either consist of the status (open or closed) of process switches or of the coded signals available from digital process instruments.

In addition to the analog and binary inputs which characterize the state of the process, there may be push buttons, punched card, or punched tape readers to allow the operator to communicate with the computer.

On the output side, the computer provides the possibility of producing up to several hundred *analog outputs*. These signals are piecewise constant since, for reasons of operating safety, analog *holds* are usually inserted between the digital-to-analog converter and the process. If a computer failure were to occur, the process would thus remain in the state resulting from the last values stored in the holds.†

In addition to analog outputs, the computer may operate on-off devices in the process and inform the operator of abnormal conditions via display lights and other alarms.

Better process understanding is gained and operating records are established automatically by use of the electric typewriter, the card and/or paper punch which print out selected data in accordance with the instructions contained in the program.

It is emphasized that these are only the common types of input and output equipment. Extensive research is underway to produce more versatile peripheral equipment, such as machines capable of reading print directly, understanding a limited number of voice commands, and displaying the state of the process in a more comprehensive way. In a somewhat different direction, special equipment is being developed to permit computers of different designs to communicate with one another.

2.4 Programming [19]. The program of a digital computer consists of a set of prerecorded instructions which indicate with great detail the nature of the next partial computation and the precise location in the memory of the data required to perform this computation. The logical, information storage, and retrieval operations are likewise controlled by the program.

Until several years ago, the composition of a program was time-consuming and required a detailed knowledge of the machine's internal operation. The invention in the early fifties of the so-called automatic programming methods contributed much to the widespread utilization of computers by the non-specialist. Some of these methods make it possible for the user to write down the desired sequence of partial computations in a *language* which is almost clear English. The computer manufacturers, on the other hand, make available the programs corresponding to frequently used information processing and computation tasks. Automatic programming methods which have greatly aided scientific machine computation are not, unfortunately, as efficient as the

† These holds are servodriven potentiometers of which the output voltages change in a stepwise fashion at each sampling instant and remain constant in between.

conventional methods, and therefore should not be used when the same computation is to be repeated many times. Since, on the other hand, each process requires a different program, libraries are not of great help. The composition of an efficient program for process control applications thus remains a time-consuming task which may require several man-years of painstaking work. Much thought needs to be given to this task, since the success of a computer control system to a large extent rests on the proper utilization of the computer, that is, on the quality of the program.

In general, several different programs are contained in a process control computer to account for different situations, for example, emergencies. Transitions from one program to another (interrupts) can be provoked automatically, perhaps as a result of an abnormal condition signaled by a process instrument, or manually. Various other refinements, such as omission of a predetermined sequence of steps in the instruction program if certain conditions pertaining to the process state or to the computation itself are realized, may be used to increase speed. It is also customary to have the computer, at prescribed intervals of time, perform "test" computations where the correct results are known, to make certain that no essential part has failed.

At the present time, digital computers require extremely detailed instructions; their achievements are impressive only because of their capability to carry out very simple operations at very high speed. Significant research is underway to make possible more "intelligent" behavior, such as adaptation and learning, and the ability to act upon relatively superficial instructions. Success of this research depends more on imaginative methods of programming than on improved computing elements and systems.

2.5 The General Purpose Analog Computer. The electronic analog computer is based upon the observation that certain electronic components such as high-gain dc amplifiers, resistors, capacitors, etc., can be interconnected in such a way that the resulting network is described by precisely the mathematical equations of which the solution is sought. "Simulation" by these methods of linear, constant coefficient, differential equations is particularly straightforward, but many other types of equations (algebraic, nonlinear, time-varying, and partial differential) can be handled with the nonlinear and logical circuitry now available.

Although clearly the analog computer is conceptually much more limited than the highly versatile digital computer, its value for process control should not be underrated: in a number of specific situations, especially when high accuracy is not required, it surpasses the digital computer in speed at lower equipment and programming cost.

There is much evidence today that the control of complex processes will require the *simultaneous* utilization of analog and digital machines; analog computers appear to be well suited for the more conventional control tasks of the first level whereas digital computers are generally required for the second level. Linkage between the two will become desirable in an ever increasing number of situations.

2.6 Characteristics of a Typical Analog Control Computer. As before, it is difficult to define a "typical" machine, since *any* sufficiently reliable analog computer can be used to control a process. This covers an extremely wide range, from about $1000 for a low-accuracy ten-amplifier linear machine to $200,000 for a high-accuracy computer fitted with several hundred operational amplifiers, nonlinear and logical blocks, the possibility to accurately set the coefficient-potentiometers by tape-controlled servo-mechanism, etc. The characteristics given in the following correspond to the PC-12 which was specially designed by Electronic Associates Inc. for process control purposes.

The most outstanding feature of the PC-12 is its flexibility, obtained by use of *computing modules*, chosen so as to satisfy the computation needs in each individual case. Up to fifty solid-state modules, each of which consists of a computing component and a small patch panel attached to it, can be slipped into standard enclosures and connected to a single power supply. The manufacturer provides modules for multiplication by a constant, integration with respect to time, low-level amplification, multiplication or division of two variables, function generation, comparison of two variables, etc. Some of these modules are available in several grades of accuracy to satisfy computing needs at minimum cost. The program consists of interconnecting the individual patch panels by patch cords and special plugs; minor program changes can be effected easily to account for process alterations or modifications of the objective function. Module accuracies are usually of the order of 0.1%, which means that computing system accuracies are often better than 1%.

2.7 Input-Output Equipment and Programming. The analog computer accepts analog as well as on-off voltages from the process. The operator can change the nature of a computation by pressing push buttons on the computer console or by altering the settings of the coefficient-potentiometers. The outputs are generally dc analog voltages, although on-off signals are also obtained easily from commercially available relay blocks.

Whereas in the digital computer, information is processed *sequentially*, that is, one input line at a time, the analog computer acts upon all the inputs *simultaneously*. Though in general the computations are performed continuously, as in the case of a conventional controller, it is also possible to operate on a set of *sampled* process values, perhaps to predict the future state of the process by performing a "fast" simulation, and to produce a set of sampled actuating signals in accordance with the outcome of the fast simulation. An example of this will be given in Section 5.9.

Programming the analog computer consists in interconnecting the available electronic components in such a way that the resulting network is described by the equation to be solved or, more directly, by the *analog* of the mechanical, electrical, chemical, etc. process to be simulated.† Thus far, this operation has always been performed manually, which frequently led to "scaling" and "patching" errors. Digital computer programs have recently been developed to specify *optimum* patch diagrams and component values directly from the equations to be solved, which considerably reduces the drudgery of complex simulations [9].

2.8 Memories and Look-up Tables. Due to the high cost of computers, it is advisable to explore in each application if the desired system performance cannot be obtained by *storage* of the laws of control into a suitable memory, perhaps a nonlinear network or a magnetic drum. It is recalled here that the law of control relating the manipulating vector $\mathbf{m}$ to the process state $\mathbf{x}$ and the desired output $\mathbf{r}$ can generally be written in the form

$$\mathbf{m}(t) = \mathbf{g}[\mathbf{x}(t), \mathbf{r}(t), \mathbf{u}(t)] \tag{2.1}$$

or

$$\mathbf{m}[(l+1)T_s] = \mathbf{g}[\mathbf{x}(lT_s), \mathbf{r}(lT_s), \mathbf{u}(lT_s)] \tag{2.2}$$

T_s = constant sampling period.

The mathematical operations implied by Eqs. (2.1) and (2.2) are purely algebraic; they can be performed once and for all, either by hand calculation or by off-line machine computation, tabulated in a memory and retrieved for control when required. In order to avoid storing a prohibitive number of points, one may add simple interpolation equipment to the memory.

Example. Consider the classical second-order optimum relay system, of which the law of control is shown dotted in Fig. 4.

It may be sufficient to store a small number of points $M_1, M_2, \ldots, M_1', M_2', \ldots$, and to situate the actual operating point **P** with respect to the broken line $M_3'M_2'M_1'OM_1M_2M_3$. As long as **P** is located to the right of this line, m is positive. At the point Q, where the actual phase trajectory intersects the segment M_1M_2, a reversal of the relay polarity is provoked and nearly optimum performance is accomplished.

It is thus possible, in principle, to control a variety of static and dynamic processes by tabulating a sufficiently fine grid of points $\mathbf{x}$, $\mathbf{r}$, $\mathbf{u}$, and $\mathbf{m}$ and by retrieving the required actuating vector $\mathbf{m}$ corresponding to each "address" defined by the measured values of $\mathbf{x}$, $\mathbf{r}$, and $\mathbf{u}$. If, in a specific situation, the number of points to be stored becomes prohibitive, or if the plant parameters are subject to rapid change, a process computer may be more economic than an oversized memory. In still other cases, it may be best to store the law of control and to have a central computer upgrade the contents of the memory at periodic intervals of time to account for plant parameter and policy changes.

† Suppose that the process to be simulated consists of a cascade of transfer functions of the form $\alpha/(s+\beta)$. In this common situation, it is preferable to interconnect a cascade of integrators, each of which is provided with the required feedback path, rather than to start from the differential equation corresponding to the process.

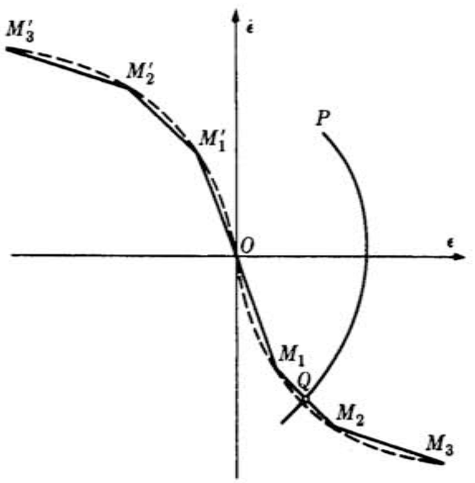

FIGURE 4. Law of control corresponding to optimum second-order relay system. The variables $\epsilon = r - y$ and $\dot{\epsilon} = \frac{d}{dt}(r - y)$ are the state variables.

2.9 Temporal Operation of On-Line Computers. In the majority of the situations where an analog computer is used for control, the actuating vector **m** is produced *continuously* on the basis of continuously available information on the process state **x**. In some analog systems and in the majority of the digital systems, the process state **x** is measured at discrete instants of time $0, T_s, 2T_s, \ldots, lT_s \ldots$ called sampling instants, and the required actuating vector **m** is made to change in a stepwise fashion at the instants $T_s, 2T_s, 3T_s, \ldots, (l + 1)T_s$.

The sum of the intervals Δ and δ which are required for data collection and computation, respectively, must be smaller than the sampling interval T_s. Data collection is generally not instantaneous since measurement of the process variables is performed sequentially and needs to be converted into a suitable code.

The time δ corresponding to each sampling period is not always constant, but may depend on the nature of the computation to be performed. The time required to solve a problem by means of successive iterations, for example, is a function of the numbers selected for the first iteration, i.e., of the process state. See Fig. 5.

The selection of a suitable sampling period T_s demands careful thought. If T_s is chosen too large, the disturbances occurring between consecutive samples are not accounted for, but sufficient time is available to perform detailed calculations on those disturbances actually observed. If, on the

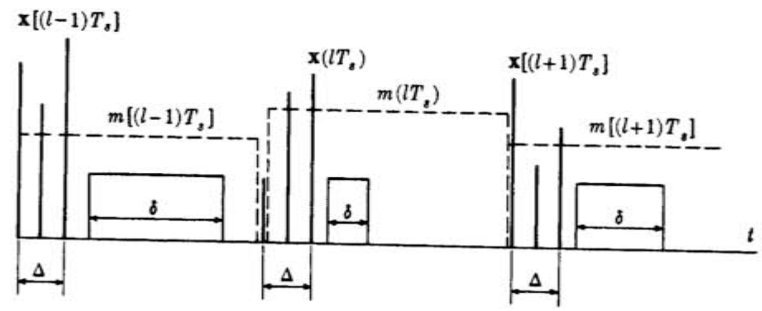

FIGURE 5. Temporal operation of sampled computer control system. Δ represents the time required for data collection and preparation; δ corresponds to actual computing time. The sum $\Delta + \delta$ must be smaller than the sampling period T_s.

other hand, T_s is chosen too small, most of the disturbances will be detected, but optimum control is impossible for lack of computation time within each cycle.

It is also possible to process measured data at irregular intervals of time, perhaps when a disturbance has been detected. If it should happen that several different disturbances, each requiring computer time, occur simultaneously, the machine can be programmed to treat these in their respective orders of priority.

3. Control

We agreed at the beginning of this chapter to take a rather wide view on the subject of computer process control by stressing the importance in the over-all system design of processes incorporating humans as well as machines. It is similarly desirable to interpret the expression "control" rather widely to include not only closed-loop, but also open-loop, as well as certain types of data processing systems. One justification for this is that most of the control computers installed at the present time are programmed to perform a variety of tasks pertaining to open- and closed-loop control and to data processing on a timeshared basis, with closed-loop control remaining a rather new and not yet fully proven application. Another justification is that a specific installation may at the beginning be purely open-loop and subsequently be made closed-loop after sufficient operating experience (and confidence) has been gained.

We shall exclude from our study the numerous situations where information is being processed in an entirely "off-line" fashion and will only consider those situations where a computer is permanently connected to a process, which is referred to as "on-line" operation.

Following closely the exposition given in Adriaenssens [1], we shall first discuss the data processing and open-loop control tasks of "data logging," "monitoring and alarm," "sequence control," and "dispatch of information." Separate sections will thereafter be devoted to the more critical and, for our purposes, more relevant subject of optimization by computer, which usually requires closed-loop control.

3.1 Data Logging. In this application, the measurements of the various process instruments are coordinated by the control computer and transformed into a "format" suitable for off-line data *reduction* by another computer. This involves the selection of the instants of time at which measurements are taken, fairly simple calculations such as are required to derive a nonmeasurable state variable from measurable process variables, and the production of punched cards, punched tape, or other operating records.

The chief purpose of data logging is to gather improved process understanding and to produce the *mathematical model* required before more sophisticated forms of control can be envisaged. Data logging installations are also used for "post-mortem" analysis and "operator surveillance"; if an abnormal condition or emergency arises in the operation of a large-scale process, it is very important to trace the causes by subsequent analysis of the evolution of the process state and the operators' interventions *before* the emergency materialized. This is accomplished conveniently by storage of the complete process history during a specified number of hours and by erasure from the memory of old data while new data are being introduced. The unwieldy masses of paper sharts of conventional analog recorders are thus avoided.

3.2 Monitoring and Alarm. Here, the computer scans the various process instruments in accordance with a predetermined sequence and compares their readings with internally stored low and high limits. If one such limit is exceeded, the computer sounds an alarm and thus informs the operator of an abnormal condition.

In addition to comparing each instrument reading to a predetermined limit the computer can *predict* the process evolution by extrapolating past samples and sound the alarm before the limits are transgressed. It is also possible to *adapt* the predetermined limits to operating conditions; excess values can thus be tolerated during short periods of time without leading to equipment damage.

3.3 Sequence Control. In this mode of operation, which is typical of open-loop control, the computer initiates a sequence of operations in accordance with an internally stored program. The next step of the sequence may be triggered after a specified interval of time has elapsed or when a stated set of process conditions is realized. Complex logical functions can thus be implemented without having to resort to expensive and cumbersome relay equipment. If a computer is used, it is frequently not necessary to store the complete

program in detail: *interpolations* and other simple calculations eliminate the intermediate steps.

The applications of sequence control in industry are extremely important since they prevent equipment damage and inefficient process operation resulting from operator fatigue or negligence, and avoid uncoordinated and ill-timed operator intervention in case of emergency.

3.4 Dispatch of Information. In this application, the (digital) control computer gathers messages and data pertaining to the operation of subprocesses and distributes these to the proper departments at the proper time. Production data might thus be sent to the accounting department, whereas technical data would go to the engineering services. This substantially strengthens the links that should exist between the departments of the first, second, and third level. Dispatch of information, as well as some of the other applications discussed above, do not rely on sophisticated concepts of control and optimization. But the most valuable practical contribution of control computers may well be the reduction of chaos in the over-all process rather than the optimization of subprocesses in accordance with criteria the value of which is subject to question.

3.5 Regulation and Optimization. There are many ways in which the addition of an on-line computer can improve the response of a process to changes in reference input and disturbances and ensure more economic or otherwise more desirable operation. These two areas of application, which are broadly classified as regulation—or control in the conventional sense—and optimization, fall directly within the province of the control engineer

The techniques required to design computer-based regulation and optimization systems can be categorized according to the temporal behavior of the disturbances affecting the process. We distinguish between:

(1) step disturbances of which the mean time of occurrence is appreciably larger than the dominant time constant of the controlled process;
(2) step disturbances of which the mean time of occurrence is of the same order of magnitude as the dominant time constant of the controlled process;
(3) stationary and nonstationary random disturbances.

The detrimental effects of disturbances of the first kind can be minimized by the addition of a computer or controller of which the response time need not be fast. The resulting optimization is referred to as *static* since it is based exclusively upon consideration of the static process model (1.2).

Since disturbances of the second kind constantly prevent the process from attaining its objective, computers are required to improve the regulation capabilities of the system and to determine the optimum trajectory to be followed by the process state $\mathbf{x}$ between consecutive steady-state conditions. Since the design must be based upon consideration of the dynamic model (1.1), optimizations of this kind are referred to as *dynamic*. Whereas there are already several hundred computer systems capable of static optimization throughout the world, only a few dynamic optimization systems have been designed, largely for experimental and demonstration purposes.

If the disturbances are random, but stationary, it is usually possible to optimize an objective function of the form

$$J = \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} \mathbf{x}^T Q \mathbf{x}\, dt \tag{3.1}$$

$Q = n \times n$ constant positive definite matrix

$\mathbf{x}^T =$ transpose of $\mathbf{x}$

by a straightforward extension of Wiener's theory of optimum linear filters. A control computer is not needed in this case, since the required system can be synthesized with linear analog networks; see, for example, Truxal [26], Chapter 8.

If the disturbances are random, but not stationary, or if the objective function (3.1) explicitly includes the parameter time, conventional compensating networks no longer suffice and a computer is required. Optimum stochastic systems are discussed in Chapters IV and VII and will not be mentioned further here.

Various types of computer-based systems capable of static optimization are reviewed in Section 4; dynamic optimization and control is discussed in Section 5. In both cases, it was judged preferable to outline briefly a relatively large number of systems rather than to present the details of a few selected examples. It is hoped that this presentation of the subject matter will emphasize the very broad range of potential control computer applications.

4. Static Optimization

Industrial processes comprise a large number of independent or single-variable feedback control systems, each of which forces an easily measurable

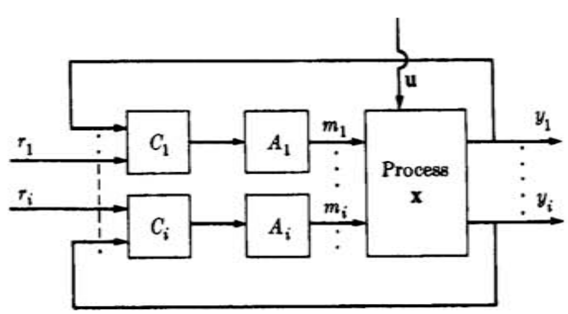

FIGURE 6. Conventional control of a multivariable industrial process by independent feedback systems, comprising actuators A_i and controllers C_i.

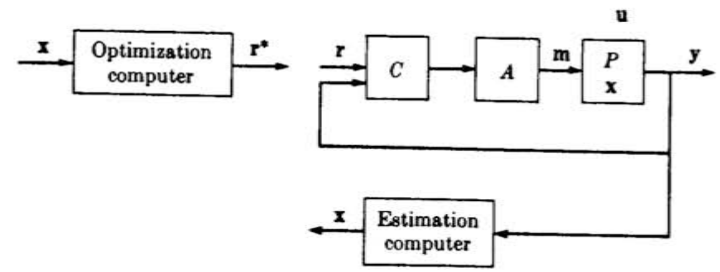

FIGURE 7. Early utilization of computers to estimate $\mathbf{x}$ and to calculate optimum settings $\mathbf{r}^*$.

dependent variable y_i to be equal to the corresponding reference variable, or set-point r_i, as shown in Fig. 6.

It is the responsibility of the human operator to monitor the y_i and to adjust the r_i in such a way that some stated economic objective $F = F(\mathbf{x})$ be optimized, or approximately optimized, in spite of the (infrequent) disturbances $\mathbf{u}$ and subject to certain constraints affecting $\mathbf{m}$. This task often exceeds the capabilities of the human operator for two reasons, namely:

(1) the objective function F generally comprises internal, that is nonmeasurable, process variables $\mathbf{x}$ which must be derived by calculation from $\mathbf{y}$;

(2) the optimum set-points r_i are related in a complex manner to the internal variables x_k and thus to the objective function F.

Digital and analog computers were therefore systematically introduced during the last decade to assist the human operator in deriving the x_k from the measurable y_i and to indicate optimum set-points r_i on the basis of certain known properties of the process. Early computer control installations are thus represented by the diagram of Fig. 7, where, for simplicity, vector notation is used to designate the r_i, y_i, m_j, x_k, and u_l.

The system of Fig. 7 is open-loop as far as the estimation and optimization computers are concerned, since the human operator transfers the estimated values of $\mathbf{x}$ to the optimization computer and adjusts the r_i accordingly if the suggested values r_i^* appear reasonable. Computer reliability thus is not a limiting factor in installations of this kind.

The next step, taken in 1959, was to automatically transfer information between the computers and the system, as shown by the closed-loop configuration of Fig. 8. The estimation and optimization computers of Fig. 7 have been merged in one block since both operations would normally be performed on the same machine. Note also that some of the disturbances $\mathbf{u}$, those that are measurable, may be fed to the computer to facilitate optimization.

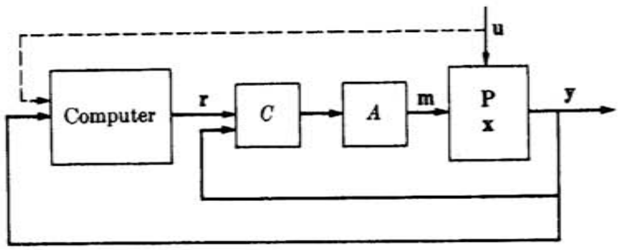

FIGURE 8. Closed-loop computer system for static optimization.

The mathematical techniques needed to design closed-loop systems for static optimization are not the same as those worked out for ordinary feedback controls, since temporal behavior and stability are immaterial. The system is admittedly closed-loop, but the "gain" of the outer path is deliberately kept so small that the state slowly evolves from one steady-state condition to the next. Reliability is not a major consideration either, since in the case of computer failure the process remains under the control of the inner loop.

Minimization of the cost F thus being the primary objective, static systems are designed in accordance with the mathematical methods of *operations research*. Since the control engineer is not, as a rule, familiar with this subject, a short discussion of linear and nonlinear programing which are among the most widely used methods of operations research, is given in the following.

4.1 Introduction to Linear Programming. In the hands of the operations researcher, linear and nonlinear programming are valuable mathematical methods to assist management in making rational and quantitatively based decisions regarding the operations under its control. In the hands of the control engineer, these same mathematical methods are programmed into a control computer, the function of which is to determine optimum set-points, either at regular intervals of time or whenever a disturbance has been detected.

The classical linear programming problem is formulated as follows; see, for example, Kaufmann [15].

Given N variables $z_1, \ldots, z_N$ constrained to be positive or zero and a linear objective function

$$F = \sum_{i=1}^{N} c_i z_i \tag{4.1}$$

$$c_i = \text{arbitrary known constant}$$

it is desired to minimize† F subject to M additional linearly independent constraints of the form

† Since F is usually a cost, we seek a *minimum*; if F were a profit, linear programming could easily be adapted to provide a *maximum*.

$$\begin{aligned} \sum_{i=1}^{N} a_{1i} z_i &= b_1 \\ &\cdots\cdots\cdots \\ \sum_{i=1}^{N} a_{Mi} z_i &= b_M \end{aligned} \tag{4.2}$$

where the a_{ji} and b_j are also arbitrary known constants. For a meaningful solution to exist, it is required that $M < N$.

Example. Let $F = c_1 z_1 + c_2 z_2$

$$z_1 \geqslant 0, z_2 \geqslant 0 \tag{4.3}$$

and

$$a_{11} z_1 + a_{12} z_2 = b_1. \tag{4.4}$$

Equations (4.3) and (4.4) are graphically represented in Fig. 9; the minimum of F is seen to be located at either of the vertices A or B, since the permissible solutions are confined to the first quadrant.

In the classical applications of operations research, this is always the case, since F represents a total cost which, in a realistically formulated problem, must have a nonnegative minimum. It should be noted that at this minimum, either of the two variables z_i is zero (depending on c_1 and c_2), unless the line F = constant is parallel to the constraint (4.4); in this degenerate case, *any* point of Eq. (4.4) constitutes a solution. It can be shown that, generally speaking, $N - M$ variables must be zero at the minimum, which is located at one the vertices of the convex polyedron formed by the surfaces $z_i = 0$ and the M constraint surfaces (4.2).

It frequently happens in practical situations that some or all of the constraint equations (4.2) are replaced by inequalities of the form

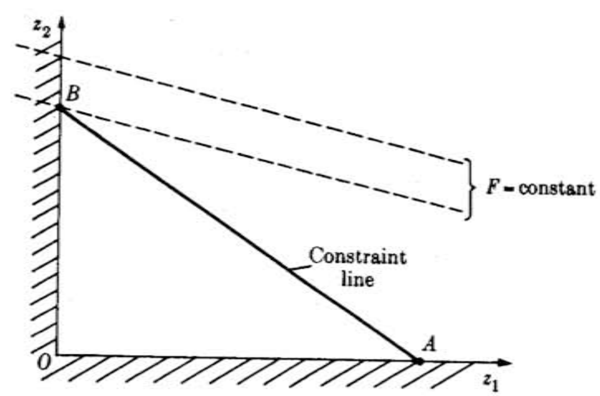

FIGURE 9. Graphical solution of the linear programming problem of the example.

$$\sum_{i=1}^{N} a_{1i} z_i < b_1 \tag{4.5}$$

. . . .

The addition of as many *auxiliary variables* z_k as there are inequalities permits this case to be treated as before. Inequality (4.5), for example, would thus be replaced by equality

$$\sum_{i=1}^{N} a_{1i} z_i + z_{N+1} = b_1. \tag{4.6}$$

The following typical example of application concerned with *production scheduling* in a steel plant is discussed in detail in Kaufmann [15].

There exist three rolling mills A, B, and C. The production schedule, established 1 month in advance, requires that p_j tons of 14 different grades of rolled product be produced ($j = 1, \ldots, 14$). The cost c_{ij} in \$/ton of the jth product in the ith mill and the production rate d_{ij} in tons/hour of the jth product in the ith mill are known. In addition, each of the 3 mills is constrained to operate no more than b_i hours/month. An exact solution is obtained by digital computer in a few minutes; this solution constitutes a saving of 6% as compared to the previous empirical schedule derived from experience.

In any realistic situation, the number N of variables is such that a solution can only be obtained by use of a digital computer. Since it is known that except in the degenerate case previously mentioned, the solution must lie at a vertex, one might attempt to calculate the value of F corresponding to each vertex located in the permissible region of the N-space and thereafter retain the minimum of the finite number of results thus obtained. This may still be prohibitive even for a modern digital computer, because the number of points to be tested is $\dfrac{N!}{M!\,(N-M)!}$. A convenient iterative procedure due to G. B. Dantzig (see, for example, Kaufmann [15]) reduces this number roughly to M and permits a computer solution to be envisaged for most practical situations.

Formulations (4.1) and (4.2) are particularly well suited to solve the classical operations research problem of the optimum *allocation of resources*, where the constraints are usually of the stated type. A variety of control processes can be statically optimized by use of a formulation which is *similar* to that of Eqs. (4.1), (4.2), and (4.5), as will be shown in the following.

We consider the *static* mathematical model

$$0 = \mathbf{f}(\mathbf{x}, \mathbf{m}, \mathbf{u}) \tag{4.7}$$

and we assume, for the time being, that Eq. (4.7) can be approximated by the linear model

$$0 = P\mathbf{x} + Q\mathbf{m} + S\mathbf{u} \tag{4.8}$$

for *large* variations of $\mathbf{x}$, $\mathbf{m}$, and $\mathbf{u}$. Moreover we assume that an objective function involving *some* of the x_i (and possibly m_j) has been defined and that

some of the x_i are constrained to remain within the lower and upper bounds x_i^L and x_i^H. In any practical situation, *all* the actuating variables m_j are furthermore confined to the range m_j^L and m_j^H. It is desired to optimize the objective function

$$F = \sum \alpha_i x_i + \sum \beta_j m_j \tag{4.9}$$

subject to the constraints

$$x_i^L \le x_i \le x_i^H \tag{4.10}$$

$$m_j^L \le m_j \le m_j^H \qquad j = 1, \ldots, q \tag{4.11}$$

and to the static model (4.8).

It is convenient, though not necessary, to express the optimization problem solely in terms of the independent variables m_j. To this end, we make use of Eq. (4.8) to obtain,

$$\mathbf{x} = -P^{-1}Q\mathbf{m} - P^{-1}S\mathbf{u} \tag{4.12}$$

if P is a nonsingular matrix, and we rewrite (4.9), (4.10), and (4.11)

$$F = \sum_{i=1}^{N} c_i m_i + \varphi \tag{4.13}$$

$$m_j^L(\mathbf{u}) \le \sum_{j=1}^{M} a_{ij} m_j \le m_j^H(\mathbf{u}). \quad j = 1, \ldots, M \tag{4.14}$$

The quantities c_i, φ, and a_{ij} are defined for convenience of exposition. Their exact meaning will become clear in the example treated in the following.

Formulations (4.13) and (4.14) are similar to the original formulations (4.1) and (4.2) except that the constraint equations are *inequalities* and that the variables m_j are bounded by the generally nonzero m_j^L. It remains true that except in the degenerate case, the optimum is located at one of the vertices of the convex polyhedron defined by (4.14), but it can no longer be said that $N - M$ variables must be zero.† G. B. Dantzig's algorithm, though less efficient, remains applicable. The presumably known disturbances $\mathbf{u}$ enter into the objective function (4.13) as well as into the constraint equations (4.14), as a result of which the optimum vertex depends on the magnitude of the u_l. The task of the control computer, therefore, consists in finding the optimum vertex each time a change of disturbance has been detected.

Example. In order to illustrate the effect of a disturbance change, we consider the following hypothetical situation. Let

$$0 = -x + m_1 - m_2 - u \tag{4.15}$$

$$F = 2x - m_1 - \tfrac{1}{2}m_2 - u \tag{4.16}$$

$$-10 < x < 10 \tag{4.17}$$

$$-6 < m_1 < 6 \tag{4.18}$$

$$-12 < m_2 < 12. \tag{4.19}$$

† Also, as a result of this different formulation, it is no longer required that $M < N$.

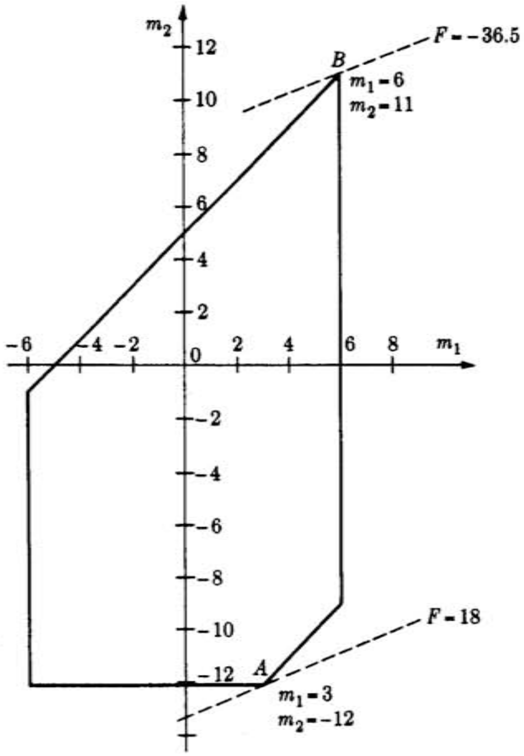

FIGURE 10. $\mu = 5$.

After elimination of the dependent variable x from Eqs. (4.16) and (4.17), the following equations result

$$F = m_1 - 2.5m_2 - 3u \tag{4.20}$$

$$\begin{gathered} -10 + u < m_1 - m_2 < 10 + u \\ -6 < m_1 < 6 \\ -12 < m_2 < 12. \end{gathered} \tag{4.21}$$

Equations (4.20) and (4.21) are plotted in Figs. 10 and 11 for $u = 5$ and $u = -5$. The optima—minima or maxima depending on the definition of F—are seen to be located at one vertex of a convex polygon which changes with the magnitude of u.

If it is unreasonable to approximate the process model (4.7) by the set of linear equations (4.8), the linear programming problem becomes much more difficult. An approximate solution, however, can be obtained as a result of iteration. The procedure consists in expanding the nonlinear process model

(4.7) about a *likely* operating point defined by the variables $\mathbf{m}^*$ and $\mathbf{x}^*$. Thus

$$0 = \mathbf{f}(\mathbf{x}, \mathbf{m}, \mathbf{u}) \approx \mathbf{f}(\mathbf{x}^*, \mathbf{m}^*, \mathbf{u}) + \sum \frac{\partial f_i}{\partial x_j}\, \delta x_j + \sum \frac{\partial f_i}{\partial m_j}\, \delta m_j \tag{4.22}$$

where the $\partial f_i/\partial x_j$ and $\partial f_i/\partial m_j$ are evaluated at the point $\{\mathbf{m}^*, \mathbf{x}^*, u\}$. The increments δx_j and δm_j are thereafter allowed a range of variation compatible with the constraint equations (4.10) and (4.11) and a solution is sought for the *linearized* problem. If this solution, characterized by $\delta\mathbf{m}$, differs too much from the assumed point $\{\mathbf{m}^*, \mathbf{x}^*, \mathbf{u}\}$, new operating points are chosen until a reasonable fit is found. The successive points must be so selected that the iteration converges.

4.2 Nonlinear Model Optimization. (Nonlinear Programming). Let Eq. (4.7)

$$\mathbf{0} = \mathbf{f}(\mathbf{x}, \mathbf{m}, \mathbf{u})$$

be the (nonlinear) process model and consider the two sets of constraints (4.10) and (4.11) where, in the general nonlinear case, the upper and lower bounds

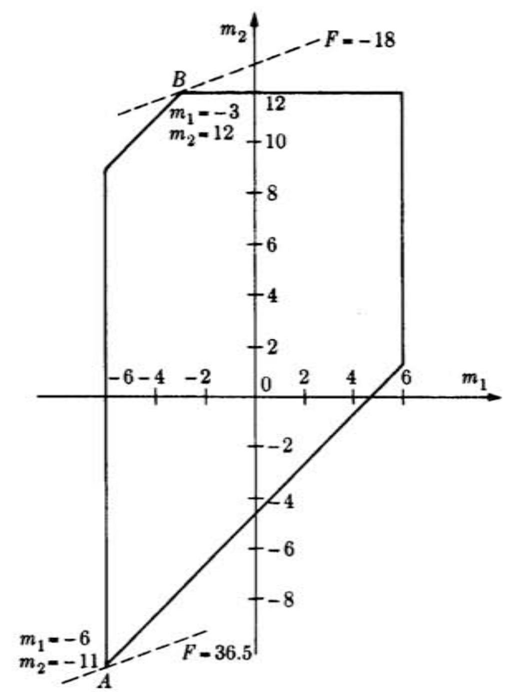

FIGURE 11. $u = -5$.

may not be constants, but may depend on $\mathbf{m}$ or $\mathbf{x}$. To find the optimum of the function

$$F = F(\mathbf{x}, \mathbf{m}, \mathbf{u}) \tag{4.23}$$

for a given $\mathbf{u}$.

In order to illustrate the difficulties arising from nonlinearity, we consider the hypothetical two-variable situation of Fig. 12, where two linear and two nonlinear constraint curves and contours of constant F are represented.

It is noted that a "local" maximum corresponding to $F = 6$ is situated in the permissible zone, whereas another maximum corresponding to $F = 13$ falls outside this zone. The optimum to be retained therefore occurs at point A, which is neither a vertex of the contour of constraints nor such that $\partial F/\partial m_1 = \partial F/\partial m_2 = 0$.

There exist several iterative computer procedures leading to this optimum by trial and error [14]. One of these procedures consists of selecting successive values of $m_1, m_2, \ldots$, in accordance with the sign and magnitude of the variation ΔF observed during the previous step until ΔF becomes sufficiently small. This iteration can be speeded up by proceeding along the line of "steepest ascent," i.e., by choosing the relative magnitude of the increments $\Delta m_1, \Delta m_2, \ldots$, so as to maximize the increment $\Delta F(\Delta m_1, \Delta m_2, \ldots)$.

4.3 Optimalizing Control. Optimalizing control, which was first described in the Western literature by Draper and Li [10], yields the static optimum operating point as a result of trial and error performed on the actual process and not on a model thereof. Let us assume that, in a specific situation, n independent outputs y_i are to be controlled by $n + \alpha$ available process

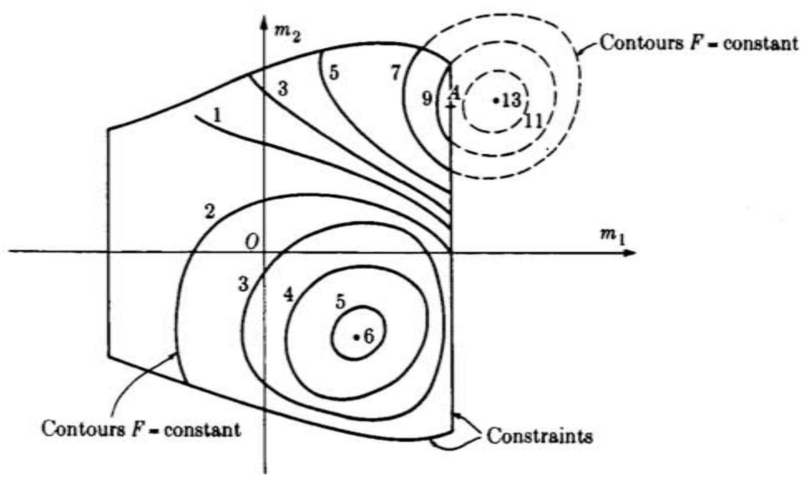

FIGURE 12. Diagram of the general two-variable nonlinear optimization problem.

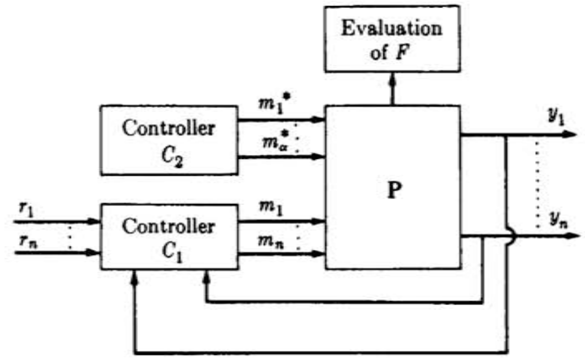

FIGURE 13. Diagram of a multivariable optimalizing system. Controller C_1 realizes the steady-state equilibrium $r_i = y_i$, $i = 1, \ldots, n$. Controller C_2 adjusts the remaining independent variables $m_1^*, \ldots, m_\alpha^*$ to optimize F.

inputs m_j. At least n of the actuating variables m_j are therefore needed to make y_i equal to r_i, $i = 1, \ldots, n$, but the remainder $m_1^*, m_2^*, \ldots, m_\alpha^*$ can be used to optimize the static objective function (4.23) subject to a constraint of the form

$$\mathbf{0} = \mathbf{f}(\mathbf{x}, \mathbf{r}, \mathbf{u}) \tag{4.24}$$

which accounts for the fact that some of the independent variables $\mathbf{x}$ are already related to the reference signals $\mathbf{r}$. Constraints of the form (4.10) and (4.11) are not considered in Draper and Li [10] and in subsequent papers on optimalizing control, although this could be done without great difficulty.

The procedure consists in performing a small change $\Delta m^*(0)$ and in observing the resulting change of F, $\Delta F(0)$. If a maximum of F is desired, the optimalizing controller is programmed in such a way that a positive $\Delta F(0)$ provokes a new increment $\Delta m^*(1)$ of the same sign as $\Delta m^*(0)$ and vice versa. If several actuating variables $m_1^*, m_2^*, \ldots, m_\alpha^*$ are available for optimization, their individual effects upon F are observed by provoking increments of each variable in sequence until finally F remains stationary. See Fig. 13. This signifies that the optimum operating point has been reached.† The set of constraints (4.23) does not need to be considered at all in this optimization scheme, since the trial and error process is performed on the actual plant, not on a model thereof. Constraints of the types (4.10) and (4.11) could be accounted for by scanning the permissible space of $\mathbf{m}^*$ only.

† Provided that the function F has only one maximum. If there were several maxima, the desired "maximum maximorum" would be determined by initiating several sequences of search with different initial conditions $\mathbf{m}^*(0)$. This complication, fortunately, is not of frequent occurrence in practice.

In addition to this *discrete* search technique which is identical in concept to the model search technique of Section 4.2, several other procedures based upon the use of periodic or random excitation of the **m*** are available and will be found in Draper and Li [10].

A serious limitation of the concept of optimalizing control arises from the fact that the increments $\Delta \mathbf{m}^*$ do not provoke an instantaneous response ΔF as a result of the dynamics of the process and of the instrumentation needed to assess ΔF. Furthermore, process and instrument noise make it impossible to reach the exact optimum condition, since the small increments ΔF near the optimum can no longer be discerned, at least not in a straightforward manner.

Optimalizing control thus appears as a very attractive concept to attain optimum operation of fairly simple noise-free and fast-reacting (with respect to F) processes. If these conditions are not met, and this seems to be the case in many practical applications, it is preferable to construct a mathematical model and to determine the optimum either analytically or by search of the model.

4.4 Optimization by Memory. When the process is not subject to frequent parameter changes, and when all the disturbances are measurable, it is possible to store the optimum law of control, calculated off-line, in the controller. A well-known application of this concept is provided by the electric utility industry which is making extensive use of memories to set the power outputs of each of n interconnected power generating units so as to minimize total production cost subject to the constraint that a given customer area be supplied with power at the proper frequency [16].

Let $F_i = F_i(P_i)$ be the known cost-power curve of the ith generating unit and $P = \sum_{i=1}^{n} P_i$ be the power required by the customers. To minimize

$$F = \sum_{i=1}^{n} F_i \tag{4.25}$$

subject to the constraint

$$\sum_{i=1}^{n} P_i = P. \tag{4.26}$$

Minimization problems of this kind are best handled by the well-known method of the *Lagrangian multiplier*, which consists in optimizing the function

$$\mathscr{F} = F + \lambda P = \sum_{i=1}^{n} F_i + \lambda \sum_{i=1}^{n} P_i \tag{4.27}$$

λ = Lagrangian multiplier.

Thus:

$$\frac{\partial \mathscr{F}}{\partial P_i} = 0$$

or:

$$\frac{\partial F_i}{\partial P_i} = -\lambda. \qquad i = 1, \ldots, n \tag{4.28}$$

Equation (4.28) indicates that minimum operating cost F is attained when the incremental costs $\partial F_i/\partial P_i = \partial F_i/\partial P_i(P_i)$ are identical for each of the n-machines. This condition is satisfied *automatically* by storage of the known functions $P_i = P_i(\partial F_i/\partial P_i)$ in an analog or digital memory, as shown in Fig. 14 for $n = 2$.

If the consumers require more power than the sum $P_i + P_2$, the frequency f of the network will drop below the set frequency $f^{(d)}$ of 60 cps and an error ϵ develops. In response to ϵ, the nonlinear controllers generate the *desired* power outputs $P_1^{(d)}$ and $P_2^{(d)}$. The power regulators, not shown in Fig. 14, force the actual power outputs P_1 and P_2 to be equal to $P_1^{(d)}$ and $P_2^{(d)}$, respectively. A little reflexion shows that, at equilibrium, the incremental costs $\partial F_i/\partial P_i$ must be equal and that the frequency error ϵ is in fact the Lagrangian multiplier—λ.

Numerous refinements beyond the scope of this section are discussed in Kirchmayer [16]. In Carpentier [4], significant research toward the optimization of the power generation and distribution system of a whole country, in the presence of voltage, current, and power setting constraints, is reported. The determination of the optimum value of F in the presence of some thousand constraints takes a digital computer, programmed according to the theorem of Kuhn and Tucker, about a quarter of an hour.

5. Dynamic Control and Optimization

In principle, a control computer can be utilized to instrument any of the laws of control discussed in the previous chapters of this book. For reasons of economy, however, one should first examine if the desired law of control cannot be implemented by means of proven and less expensive conventional controllers; this is possible in a large number of practical situations. If the

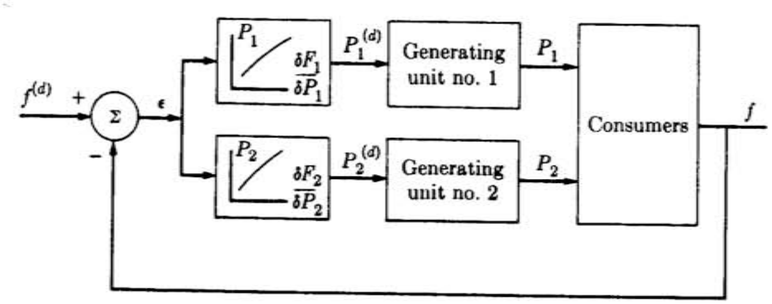

FIGURE 14. Optimization of two interlinked power generating units by storage of the law of control.

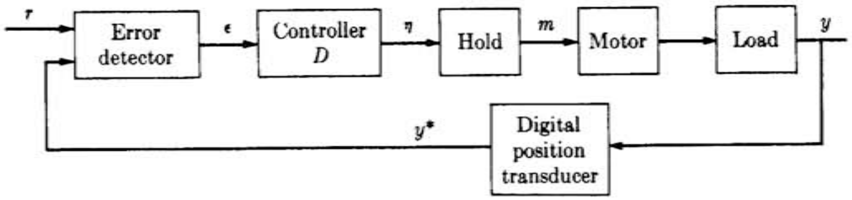

FIGURE 15. Diagram of a digital servomechanism.

purchase of a control computer of either the digital or analog type has already been justified for other reasons, perhaps static optimization, then the dynamic laws of control can often be added at little cost on a *timeshared basis*. There is some feeling that control computers, especially general purpose digital machines, may not be warranted for dynamic control and optimization alone except in a few very special situations.

In the present section, three classes of dynamic computer-based systems will be reviewed. In the first class, emphasis is placed on more efficient control in the conventional sense, i.e., improvement of steady-state accuracy and transient response to changes in set-points and disturbances. In the second class, the minimization of the cost associated with moving the process state between two consecutive equilibria by a *fixed* controller is sought (dynamic optimization). The last class, finally, allows satisfaction of this same objective by means of an *adaptive* controller. Computer control in the conventional sense, the first class, can be considered a proven technique; the second and third class remain experimental, no systematic industrial utilization having been reported as yet.

5.1 Digital Servomechanisms and Control Systems. In automatic *machine tool* applications, the accuracy capabilities of conventional analog position control systems (servomechanisms) are frequently exceeded. Servomechanisms incorporating digital position transducers, digital error comparators, and digital controllers were developed to permit the tool position $y(t)$ to track the reference input $r(t)$, contained on tape under the form of a binary code with the desired accuracy. The diagram corresponding to a digital (or sampled data) servomechanism fitted with a series controller D is shown in Fig. 15. The signals y^*, r, ϵ, and η are binary and therefore relatively immune to the perturbations limiting the accuracy of analog systems. The value of these signals changes at the sampling instants $0, T_s, 2T_s, \ldots, lT_s, \ldots$. The actuating variable m is analog and constant within one sampling interval.

As in the case of continuous control systems, it is possible and often preferable, to use a parallel rather than a series controller.

Proper adjustment of the parameters of the series or parallel controller permits satisfactory transient response to be achieved. It is recalled here that

linear digital control systems can be designed so as to reach equilibrium in a *finite* number of sampling periods [20]. "Finite settling time" or "dead-beat" designs are frequently preferred to other designs which minimize the sum of the squared errors at the sampling instants. In either case, the series controller D must produce a (finite) sequence of samples

$$\eta(lT_s) = \alpha_0\epsilon(lT_s) + \alpha_1\epsilon[(l-1)T_s] + \cdots. \tag{5.1}$$

The law of control (5.1) can be instrumented easily with a drum memory followed by a simple arithmetic unit. In the case of a parallel controller connected to all the state variables, storage of past values of ϵ is unnecessary [13].

If a sufficiently large number of digital servomechanisms and other digital control systems are located in the same plant, a general purpose digital control computer may be more economic than as many digital controllers as there are systems.

Although digital instrumentation and control gear remains more expensive than comparable analog gear, there is a noticeable trend toward such equipment, even in those applications where high accuracy is not required. The reasons for this trend are that *coded* information can be transmitted, displayed, and processed more effectively than analog information. The major types of analog-to-digital and digital-to-analog conversion equipment are discussed in the Appendix to this chapter.

5.2 Noninteracting Control. The design procedures available to decouple the input and output variables of a multivariable process are discussed in detail in Chapter III, Section 3.8. The general subject of noninteraction (or invariance) is relevant to a discussion on computer process control for two reasons, namely the following.

(1) The series or parallel controller required for noninteraction generally cannot be instrumented conveniently by interconnection of conventional process controllers, although this is possible in relatively simple situations; see, for example, Chatterjee [6]. The analog computer is well suited to implement the required cross-coupled dynamic compensating network. Decoupling can also be achieved by means of a digital computer [18].

(2) Since the dynamic optimization of multivariable processes remains laborious except in simple cases, we frequently start out by *decoupling* the process and thereafter optimize each of the now independent loops. The resulting design is not, in general, optimal, since the true relations and constraints between state variables are ignored, but the *suboptimization* thus accomplished is expedient and more satisfactory than no optimization at all.

To familiarize the reader with the use of an analog computer for decoupling, we shall instrument the series compensating network discussed in Chapter III, Section 3.4, and reproduced below:

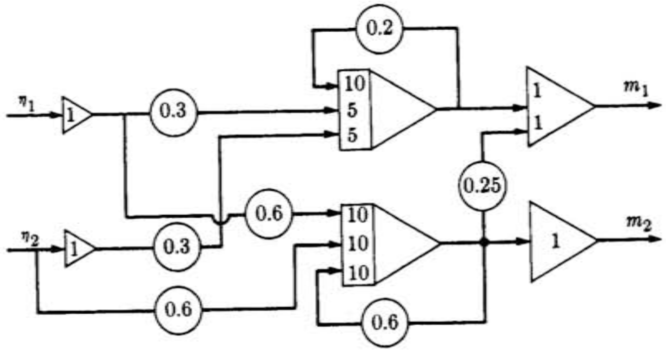

FIGURE 16. Analog computer diagram of the decoupling network (5.2). The symbols ⊳ and ▹ represent integrators and summing amplifiers, respectively. Sign inversion is assumed to take place.

$$D(s) = \begin{bmatrix} -\dfrac{3(s+4)}{(s+2)(s+6)} & \dfrac{-6}{(s+2)(s+6)} \\ -\dfrac{6}{s+6} & \dfrac{6}{s+6} \end{bmatrix}. \tag{5.2}$$

In order to implement the transfer element $d_{11}(s)$, which, in addition to two poles, has a zero, it is convenient to perform a partial fraction expansion; this yields

$$d_{11}(s) = -3\left[\frac{\frac{1}{2}}{s+2} + \frac{\frac{1}{2}}{s+6}\right]. \tag{5.3}$$

The complete compensating network, which requires only two analog integrators, is shown in Fig. 16.

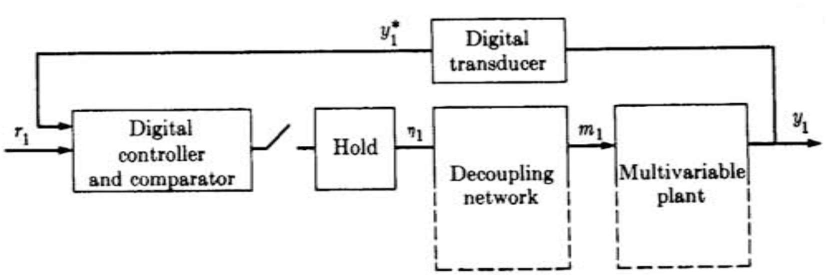

FIGURE 17. Closed-loop system configuration corresponding to the upper loop of the decoupled multivariable plant.

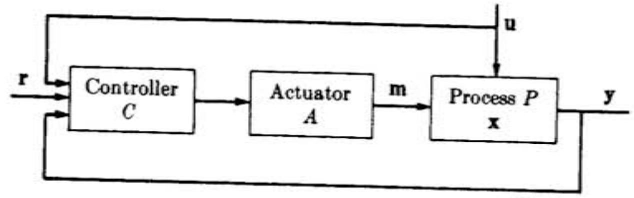

FIGURE 18. Diagram of a combined feedforward feedback system. The measured disturbance **u** is fed to the controller which generates an actuating vector **m** to cancel the effects of **u** upon **x** and **y**.

It is clear that other network configurations are possible and may be more appropriate in specific situations.

The plant considered in Chapter III, Section 3.4, can thus be decoupled by means of an analog computer programmed to satisfy Eq. (5.2). Other requirements, such as steady-state accuracy, optimum steady-state, and (sub-optimum) dynamic performance are satisfied by use of additional controllers, either analog or digital. A possible *system* configuration utilizing a digital controller is thus shown in Fig. 17. The reference input r_1 of Fig. 17 may, in specific situations, be set at regular intervals of time by still another computer (not shown here) which is assigned the task of supervising and optimizing the whole process of which the plant considered in this section is only a part.

5.3 Feedforward. Feedforward is a commonly used procedure to cancel the static and dynamic effects of *measurable* disturbances upon the independent process variables.

This is accomplished by means of a controller programmed to generate an actuating vector **m** such that the effects of perturbations are compensated for, at least approximately, before they materialize. Feedback control alone may not be satisfactory, since correcting action starts only after the state variables have become affected. Feedforward, on the other hand, does not permit accurate disturbance cancellation under steady-state conditions because of incomplete knowledge of the process equations. It is for this reason that feedforward systems generally also include a feedback loop, as shown in Fig. 18.

In practice, it is often possible to improve the response time of the feedforward loop by addition of a second, third ··· actuator which directly acts upon those parts of the process located nearest to the output **y**. This is shown in Fig. 19.

The best configuration in each specific case depends more on the ease with which the successive parts of the process can be influenced by available actuators than on theoretical considerations. As an example of application of the concept of feedforward, we consider a mill stand where plate of non-

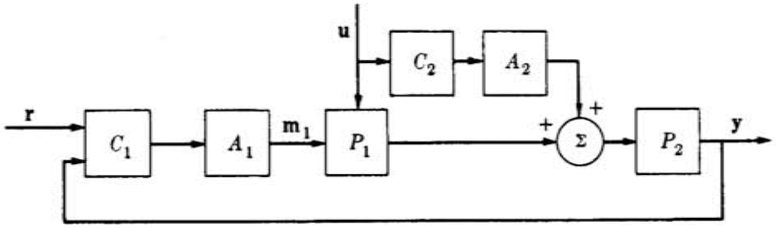

FIGURE 19. Combined feedforward feedback system; A_2 acts upon that part of the process, P_2, located nearest to the output.

uniform gauge—this is the disturbance u—enters the rollers at a known velocity v (see Fig. 20). The gauge of the outgoing product is determined by the spacing of the rollers e, and by the tension T. Since, for practical reasons, the actual gauge cannot be measured immediately after the product leaves the rollers, there is a pure time delay which precludes a fast-acting feedback control. Measurement of gauge variation at the input by g_1 and subsequent control of e and T on the basis of this measurement substantially improves product uniformity.

The analytical techniques to design the feedforward controller are straightforward. In the case of the system of Fig. 18, the following mathematical model

$$\mathbf{Y}(s) = P(s)\mathbf{M}(s) + P_u(s)\mathbf{U}(s) \tag{5.4}$$

is assumed to be known. If the dynamics of the actuator can be neglected, we seek the unknown transfer functions $C'(s)$ and $C''(s)$ of the two parts of the controller corresponding to feedback and feedforward, respectively, by writing that

$$\mathbf{M}(s) = C'(s)[\mathbf{R}(s) - \mathbf{Y}(s)] + C''(s)\mathbf{U}(s). \tag{5.5}$$

Combining Eqs. (5.4) and (5.5), it follows that

$$\mathbf{Y}(s) = P(s)C'(s)[\mathbf{R}(s) - \mathbf{Y}(s)] + P(s)C''(s)\mathbf{U}(s) + P_u(s)\mathbf{U}(s). \tag{5.6}$$

The effects of $\mathbf{U}(s)$ are completely canceled if $C''(s)$ is chosen such that

$$P(s)C''(s) + P_u(s) = \mathbf{0}. \tag{5.7}$$

Suitable transient response of the feedback loop is thereafter obtained by proper selection of $C'(s)$.

The analog computer is well suited to implement $C'(s)$ and $C''(s)$. If the process were highly nonlinear or contained time delays, a digital computer might be preferable. For example, in the case of the mill control previously discussed, it is necessary to delay the measurements of the gauge transducer g_1 until the exact time when the measured part of the plate enters the gap between the rollers. This then requires a pure time delay which can be

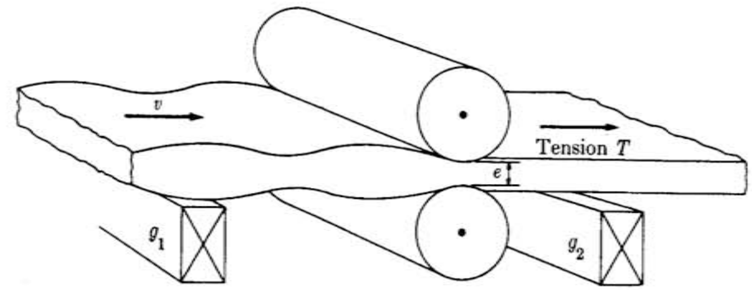

FIGURE 20. Gauge transducer g_1 permits "anticipatory" control of e and T_1 whereas g_2 eliminates possible calibration errors of the mathematical model.

instrumented much more easily on a digital computer. Another reason for preferring the digital computer in this application is that the same mill stand must process a large number of different grades and gauges of plate; therefore, it is necessary to store a mathematical model corresponding to each type of plate.

5.4 Optimum Control of Dynamic Processes. If the disturbances can be assumed to vary in a stepwise fashion, which is frequently the case in practice, the static optimization procedures of Section 4 provide optimum equilibrium points $\mathbf{r}$ for each set of disturbances. It is the function of dynamic optimization to determine the "best" trajectory to be followed by the process between consecutive set-points $\mathbf{r}$. The "best" trajectory is usually defined such that the functional

$$J = \int_0^T F(\mathbf{x}, \mathbf{r}, \mathbf{m}, t)\, dt \tag{5.8}$$

is an extremum, subject to three sets of constraints, namely (1) the dynamic relations between process variables, for example

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{m}) \qquad \mathbf{x}(t)\big|_{t=0} = \mathbf{x}(0) \tag{5.9}$$

(2) the true constraints arising from the fact that some independent variables and all actuating variables have a restricted range of variation

$$\begin{gathered} \mathbf{m}^L \leqslant \mathbf{m} \leqslant \mathbf{m}^H \\ x_k{}^L \leqslant x_k \leqslant x_k{}^H \end{gathered} \tag{5.10}$$

for some of the x_k; (3) the condition that at some time $t = T$ the prescribed equilibrium position be, in fact, attained.†

† It sometimes happens that the state $\mathbf{x}$ at $t = T$ is not specified; see [21] or the example treated in Section 4 of Chapter VII.

The lower integration bound of Eq. (5.8) is arbitrarily set equal to zero in this discussion. The upper integration bound T may be fixed (including $T = \infty$), or else it may be desired to render T as short as possible, in which case (5.8) reduces to

$$J = \int dt = \text{minimum}. \tag{5.11}$$

In the case of a digital (sampled-data) system, the *discrete* versions of Eqs. (5.8) and (5.9) would need to be considered, i.e.,

$$J = \sum_{l=0}^{N} F[\mathbf{x}(lT_s), \mathbf{r}(lT_s), \mathbf{m}(lT_s), lT_s] \tag{5.12}$$

and

$$\mathbf{x}[(l+1)T_s] = \mathbf{f}[\mathbf{x}(lT_s), \mathbf{m}(lT_s)]. \tag{5.13}$$

The dynamic optimization problem thus defined is a typical *variational* problem, which can be approached by the Calculus of Variations, Dynamic Programming, the Principle of the Maximum, and, in the case of Eq. (5.11), by several direct solutions of the bang-bang problem. It is not the purpose of the present section to give an analytical account of these methods, since this has already been done in Chapter VII, but to examine briefly the computational requirements placed upon the optimum systems designed accordingly.

A few words need to be said about the *structure* or configuration of optimum systems. It was pointed out in Chapter I, Section 4, that in *advanced* systems, several compatible objectives $O_1, O_2, \ldots,$ could be defined and that, for reasons of convenience, structures of the first, second, . . . , *degree* could be considered. In Fig. 21, a first and a second degree structure are thus reproduced.

In the case of Fig. 21a, there is only one objective O_1 which can be satisfied optimally, or just adequately, depending on how much care is taken to design C. In the case of Fig. 21b, there are two compatible objectives O_1 and O_2; the controllers C_1 and C_2 can likewise be optimal or just adequate. In principle,

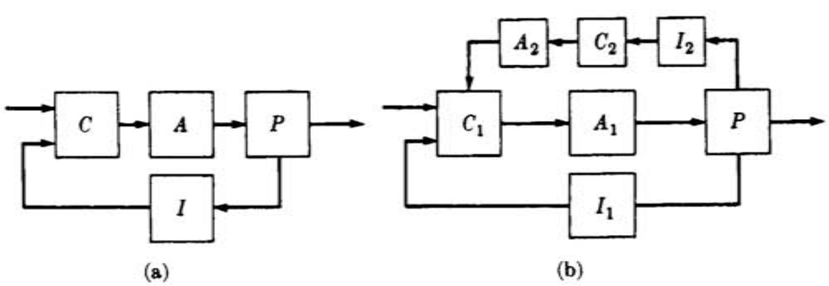

FIGURE 21. (a) First degree structure; (b) Second degree structure.

the variational methods to be reviewed in the following sections can be used to produce an optimum design of either or both of the two channels, though admittedly the mathematical difficulties may become excessive in the case of the second channel.

In what follows, we will primarily discuss the applicability of the variational methods to structures of the first degree. In Section 5.8, a structure of the second degree capable of optimum response to the first objective and of adequate response to the second objective is considered.

5.5 The Calculus of Variations. In its classical formulation, the Calculus of Variations yields that trajectory $x = x(t)$ which optimizes, maximizes or minimizes, the functional

$$J = \int_0^T F(x, \dot{x}; t)\, dt \tag{5.14}$$

for fixed T and given $x(0)$ and $x(T)$. The optimal trajectory satisfies the *Euler* equation

$$\frac{d}{dt}\frac{\partial F}{\partial \dot{x}} - \frac{\partial F}{\partial x} = 0 \tag{5.15}$$

which is a second-order ordinary differential equation of which the integration constants are determined by $x(0)$ and $x(T)$. Extensions to higher order situations, i.e.,

$$J = \int_0^T F(x, \dot{x}, \ddot{x}, \ldots; t)\, dt$$

with the boundary conditions

$$x(t)\big|_{t=0} = x(0); \quad x(t)\big|_{t=T} = x(T)$$
$$\dot{x}(t)\big|_{t=0} = \dot{x}(0); \quad \dot{x}(t)\big|_{t=T} = \dot{x}(T)$$
$$\cdots\cdots\cdots\cdots\cdots\cdots$$

exist; for example, see Weinstock [28].

The *classical* Calculus of Variations thus far has not been applied extensively to the synthesis of optimum controls for two reasons:

(1) The solution $x[x(0), x(T), t]$ of Eq. (5.15) is not suitable for instrumentation in a control system; what we seek is a solution of the form

$$m(t) = m[x(t), x(T), t] \tag{5.16}$$

which permits the desired $x(T)$ to be reached regardless of the perturbations that might occur en route. This point will be illustrated in the example treated in the following.

(2) The constraints (5.10) are not considered.

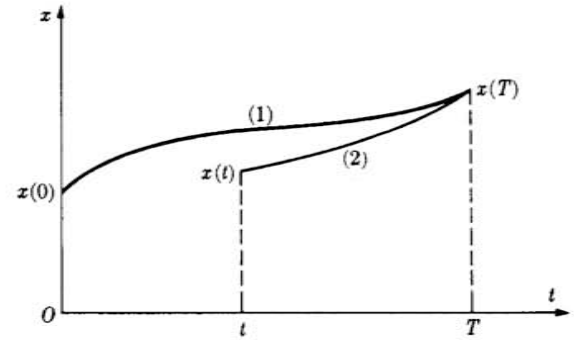

FIGURE 22. Optimal trajectory (1) $x = x[x(0), x(T), t]$, and (2) $x = [x(t), x(T), t]$.

Example. Consider

$$J = \int_0^T [(r - x)^2 + \dot{x}^2]\, dt$$
$$x(t)|_{t=0} = x(0); \quad x(t)|_{t=T} = x(T) \qquad (5.17)$$
$$r = \text{constant}$$

and

$$\dot{x} + 2x = m. \qquad (5.18)$$

There are no constraints on x, $\dot{x}$, and m. It is desired to minimize J. Equation (5.15) yields the condition

$$\ddot{x} - x = -r \qquad (5.19)$$

which is integrated to provide a *mathematically* adequate solution of the form

$$x(t) = C_0 e^t + C_1 e^{-t} + r$$
$$C_0 = C_0[x(0), x(T)]; \quad C_1 = C_1[x(0), x(T)] \qquad (5.20)$$

as shown by curve 1 of Fig. 22.

For *control* purposes, however, it is preferable to consider the present state $x(t)$ which may have been displaced from the original trajectory (5.20) by disturbances and to generate that actuating signal $m(t)$ which moves x to $x(T)$ regardless of further perturbations; see curve 1 of Fig. 22. Integration of Eq. (5.19) between the bounds $x(t)$ and $x(T)$ rather than $x(0)$ and $x(T)$ and subsequent differentiation with respect to t yields the optimal slope

$$\dot{x}(t) = \frac{(r - x)\cosh (T - t)}{\sinh (T - t)} \qquad (5.21)$$

at the present operating point. The desired actuating signal $m(t)$ is obtained by

combination of Eqs. (5.18) and (5.21)

$$m(t) = \frac{r \cosh (T - t) + x[2 \sinh (T - t) - \cosh (T - t)]}{\sinh (T - t)} \qquad (5.22)$$

in terms of r and $x(t)$.

The implementation of Eq. (5.22) requires a time-varying controller, unless $T \to \infty$, in which case m reduces to

$$m(t) = r + x. \qquad (5.23)$$

It is known that a "quadratic" performance index† and a linear plant always yield a linear law of control of the form

$$m(t) = \beta r + \sum_{k=1}^{q} a_k x_k \qquad (5.24)$$

where the "feedback coefficients" β and a_k depend in a known fashion on t, if T is finite, and are constants if T is infinite. In either case, a controller can be constructed with relative case.

In order to take into account the constraints (5.10), it is possible to add to the objective function (5.8) to be minimized a set of additional terms which grow very fast when (5.10) is violated. Suppose, for instance, that m_j must remain within the range $(-m_j^H, m_j^H)$, where m_j^H is a positive constant. If the exponent ρ in the objective function

$$J = \int_0^T F(x, \dot{x}, t)\, dt + \int_0^T \left(\frac{m_j}{m_j^H}\right)^\rho dt \qquad (5.25)$$
$$\rho = \text{even integer}$$

is chosen to be much larger than unity, the Calculus of Variations yields a trajectory which does not (appreciably) violate the stated constraint.

Additional extensions of the classical calculus of variations to the constrained case and with a view on the utilization of control computers are discussed by Carter [5].

5.6 The Bang-Bang Problem. See Chapter V, Section 1 and Chapter VII, Section 2. If the reference signal were a step of amplitude r, the switching surfaces would be *stationary* and would need to be determined only once, either by hand calculation or by off-line machine computation. The coordinates of these switching surfaces could then be stored in the controller as discussed in Section 2.8. If $r(t)$ is not a step, the switching surfaces are (generally) not stationary and it may be more expedient to compute their coordinates on-line in each specific case by application of either of the analytical methods discussed in Chapters V and VII or by use of a fast model; see Section 5.9.

5.7 Dynamic Programming. See Chapter VII, Section 3, and Bellman [2]. Though originally developed to solve the operations research problem of

† The integrand of Eq. (5.14) is often referred to as performance index.

the optimum allocation of resources in multistage processes, Dynamic Programming has since been extended to include the variational problem defined by Eqs. (5.8) and (5.9) for fixed and for minimum T. It was stressed in Chapter VII that in its present form, Dynamic Programming is not only a computational algorithm, as suggested by the term "programming," but a body of knowledge which augments the classical Calculus of Variations.

A direct consequence of this observation is that it may be impossible, even for a very high speed digital computer, to master a specific control problem perfectly manageable by a much smaller machine (or even a passive network), depending on which result of Dynamic Programming is being instrumented. The applications record of Dynamic Programming in its "raw" form [see Eqs. (3.15) to (3.24) of Chapter VII] is expected to remain rather insignificant in the forseeable future, because the speed and memory capabilities of powerful machines are rapidly exceeded.† In its "manipulated" form, however, Dynamic Programming often does not demand great computational capability.

Example. It is desired to design a digital (sampled-data) controller which eliminates any initial perturbation $\mathbf{x}(0)$ of a linear single-input plant by minimizing a quadratic objective function of the form

$$J = \sum_{l=0}^{N} \mathbf{x}^T(lT_s)Q\mathbf{x}(lT_s) \qquad (5.26)$$

N = fixed

x^T = transpose of x

Q = given positive definite square matrix.

It is shown in Kalman and Koepcke [13] with the help of Dynamic Programming that the optimum actuating signal $m(lT_s)$ at the lth sampling instant is linearly related to the actual state $\mathbf{x}(lT_s)$

$$m(lT_s) = \sum_{k=1}^{v} a_k x_k(lT_s) \qquad (5.27)$$

$$a_k = a_k(lT_s).$$

If $N \to \infty$, which is a case of practical importance, the coefficients a_k relating $m(lT_s)$ and $\mathbf{x}(lT_s)$ are independent of l and implementation of the optimal controller only requires as many constant-gain feedback paths as there are state variables. If, on the other hand, N is finite, the a_k depend on l. Implementation in this case requires retrieval, at the lth sampling time, of the proper gain to be inserted into each feedback path. This is still less laborious than on-line computation of $m(lT_s)$ based upon the raw formulation of Dynamic Programming.

The resulting control system is shown in Fig. 23.

It should be mentioned that the example discussed here as well as the much more general linear case considered in Kalman and Koepcke [13] are

† The somewhat limited experience thus far available seems to indicate that, roughly speaking, dynamic systems of the fifth order represent an upper limit.

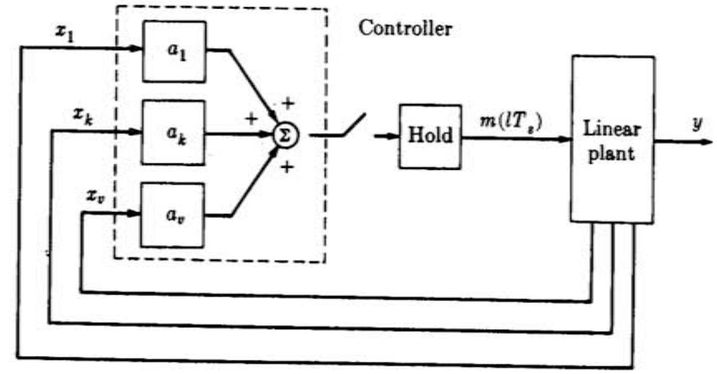

FIGURE 23. Optimum controller of a linear single-input plant to minimize a quadratic performance index.

particularly simple and could also be solved by straightforward application of the classical Calculus of Variations, as was previously done in Section 5.5. But it remains true that in each particular case, one should carefully explore whether results *derived from Dynamic Programming*, such as the pair of partial differential equations (3.36) of Chapter VII, might not alleviate the computational requirements placed upon the controller.

To summarize our discussion on Dynamic Programming, we may say that the optimum systems described in the literature thus far do not exploit the substantial advantages provided by this method, namely, inclusion of constraints and nonequivocal determination of the optimum optimorum when the Calculus of Variations yields several solutions, primarily because the computational requirements are such that there is hardly any economic incentive.

5.8 The Maximum Principle. See Chapter VII, Section 4. A major difficulty arising in the application of the Maximum Principle is the requirement of determining the vector $\tilde{\boldsymbol{\psi}}(t)$ or $\boldsymbol{\psi}(t)$. This necessitates the solution of a set of $2(v+1)$ (or $2v$) ordinary differential equations in $\tilde{\boldsymbol{\psi}}$ and $\tilde{\mathbf{x}}$ (or $\boldsymbol{\psi}$ and $\mathbf{x}$), depending on whether the objective function is of the form (5.8) or (5.11). If the initial conditions $\tilde{\boldsymbol{\psi}}(0)$ (or $\boldsymbol{\psi}(0)$) were given, integration could be carried out in a straightforward fashion by either a digital or an analog computer. Unfortunately, this is never the case and the unknown initial conditions $\tilde{\boldsymbol{\psi}}(0)$ (or $\boldsymbol{\psi}(0)$) must be determined by trial and error. The procedure is to start by assuming a set of initial conditions $\tilde{\boldsymbol{\psi}}(0)$ (or $\boldsymbol{\psi}(0)$); since the final condition will probably not be satisfied at the end of the first trial, a new set of initial conditions is chosen, until the endpoint conditions, which may involve $\mathbf{x}(T)$ or $\boldsymbol{\psi}(T)$ or components of both, are satisfied. The computation speed of analog computers is often sufficient to permit a large number of trial runs in the

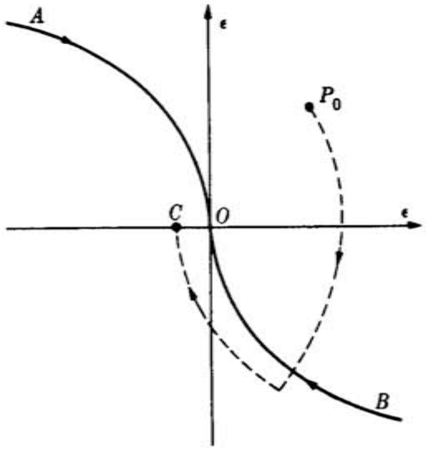

FIGURE 24. Stationary switching curve in $\{\epsilon, \dot{\epsilon}\}$ plane. ϵ is the error $r - y$.

allotted time T_s; the major difficulty in high-order problems is to automatically select successive sets of initial conditions such that the iteration converges. The remarks of the previous section on the importance of carefully determining *which* results of the Maximum Principle should be instrumented and in terms of which process variables the actuating vector $\mathbf{m}$ should be expressed are equally valid here.

5.9 Control by Fast Model. If it is not possible or not practical to store the law of control relating $\mathbf{m}$ to $\mathbf{x}$ and $\mathbf{r}$ in the controller, the desired transient performance can still be obtained in certain cases by simulating the behavior of a fast model of the actual process and by *experimentally* adjusting the actuating vector $\mathbf{m}$ of the fast model until satisfactory performance is observed. The desired actuating vector thus obtained is thereafter used to control the actual process.

The use of a fast model to optimally control a relay system subjected to nonstationary polynomial input is discussed in Coales and Noton [7]. Since in this case the switching surfaces are not stationary, but dependent on the measured coefficients of the polynomial input, storage of the law of control is not practical.

In order to facilitate exposition, we assume that the input r is a step occurring at $t = 0$ and that the plant is linear and of the second order, in which case the switching surfaces reduce to a stationary curve. It is a well-known result that optimum, i.e., minimum time response, is accomplished if the polarity of the relay output m is changed once, at the precise moment when the state $\{\epsilon, \dot{\epsilon}\}$ crosses the switching curve AOB; see Fig. 24.

Since the differential equation of the plant is known, the method consists in resolving on a computer another differential equation obtained from that describing the plant by use of an accelerated time scale and in reversing the model drive signal at some arbitrary time τ_0; the initial conditions of the fast model are identical with those, $\epsilon(0)$ and $\dot{\epsilon}(0)$, of the actual process. The resulting model performance will in all likelihood not be optimal, and an improved value of the first reversal time τ_1 will be used in a second simulation. The initial conditions $\epsilon(1)$ and $\dot{\epsilon}(1)$ are those of the actual state as measured at the time when the second simulation commences. This iterative process is continued until a suitable value for τ has been found. Consecutive values of τ are selected on the basis of the sign and the magnitude of ϵ when $\dot{\epsilon} = 0$; see point C of Fig. 24. For example, if ϵ is negative, the actual system would exhibit overshoot because τ was chosen too large. In the following iteration, τ would therefore be made smaller until ϵ and $\dot{\epsilon}$ become zero simultaneously.†

If the system input r remained constant for $t > 0$, a single sequence of iterations yielding τ would suffice to optimally control the actual process. Since, in the situation discussed in Coales and Noton [7], r changes continuously, it is necessary to compute the reversal time continuously. In order for the system to operate without excessive limit cycles about equilibrium, the model must be many times faster than the process. Analog computers are often capable of the required high-speed operations.

Finally, it should be stressed that the concept of the fast model is applicable not only to the optimum control of relay systems, but, generally speaking, can also serve to improve the transient performance of a wide class of dynamic processes. The obvious limitation is the number of possible choices of $\mathbf{m}$; if this number exceeds the speed capabilities of the computer, the analytical approaches of Sections 5.5, 5.6, 5.7, or 5.8 are necessary.

6. Summary and Conclusions

Computer process control is not a sharply defined subject, but a broad area of applied research and development to utilize computers in estimating the state of the process and in implementing the laws of control derived from the theory of automatic control, operations research, and others; computer control systems thus differ by the nature of the process, in the type of the computer used and in the way the process is influenced. Computer control systems thus far have been designed from either the operations research (static optimization) or the automatic control (dynamic optimization) point of view. In the first case, improved economy of operation of complex

† The policy of adjusting τ is actually somewhat more complicated, as discussed in Coales and Noton [7]. Since no attempt is made to define an objective function for the iterative process other than eventual convergence, the design of the fast loop is not optimum, but just adequate.

man-machine systems is sought; in the second case, better satisfaction of the technical criteria of quality associated with the operation of relatively less complex processes primarily involving machines is the objective. These two points of view now converge, because past experience stresses the importance of considering complex processes as a single entity, particularly when the installation of a digital control computer is envisaged. If so, a host of routine decision tasks can often be added at little cost.

Three groups of tasks are thus handled by the control computer. These are, in increasing order of complexity:

(1) data logging, monitoring, sequence control, and dispatch of information:
(2) static optimization, either by operator guides or closed loop;
(3) dynamic control and optimization.

Although, generally speaking, process control by general-purpose analog or digital computer remains experimental, the first two tasks are already implemented in a sizable number of installations.

In spite of the fact that much progress was recently made to yield a better understanding of the variational problem with a view toward optimal dynamic control, widespread application is not likely to occur in the near future, because there appear to be relatively few practical situations where the utilization of a computer, especially of the general purpose digital type, is economically attractive.

Although it can be asserted that large digital and analog computers constitute the key to genuine automation and will eventually find extensive use, their present applications record is only moderately conclusive. The subject matter has definitely not yet been crystallized into handbook style design methodology. In order to accelerate the introduction of on-line computers into industry and commerce, it is necessary to find ways of predicting economic justification with greater confidence. Automatic control permits the temporal or frequency behavior of a system to be related to the known temporal or frequency behavior of its constituents. Success of computer process control requires that, in addition, the cost, labor-saving, reliability, maintainability, etc., characteristics of the constituents be linked to the corresponding characteristics of the system. This in turn allows economic benefit to be predicted with greater confidence. Such a design methodology, if it were available, would yield answers to the following questions.

Which processes are well suited for computer control? How accurate does the process model need to be? Which are the main process variables? By which model—deterministic or probabilistic, static or dynamic—should they be related? Which type of computer—digital, analog, hybrid—or which system of interlinked computers is required in a given case? Which mode of control is to be programed into the computer in order to fully exploit its capabilities?

APPENDIX

Digital Instrumentation and Conversion Equipment

A.1 Introduction. The object of using a computer for process control is to improve the efficiency of the process. To accomplish this, the relevant information from the plant must be fed into the computer, of which the output is used either to guide the plant operator where manual control is employed, or to control the process automatically. Plant variables—temperature, pressure, flow rate, etc.—are measured by transducers. The computing controller may be a digital or an analog computer, or a hybrid computing system. Analog-digital and digital-analog converters link the transducers to the computer and the computer back to the controlling actuators, since most transducers and actuators are analog devices and the controller is generally digital. Where a hybrid computer is used, i.e., a combination of analog and digital computing techniques, converters are again necessary to transmit the information from one to the other. These two fundamental conversion processes may be represented schematically as shown in Figs. 25a and 25b.

When designing instrumentation for computer control systems operating on-line, certain points must be borne in mind. For control, the computer must operate in real time, while for plant optimization, faster than real time operation may be required. Therefore the conversion equipment must be designed to have the necessary speed. Digital computers are inherently accurate; care must be taken not to lose this accuracy in the conversion equipment. The importance of reliability is well realized, but we should stress here that the reliability of on-line computers is far more critical than that of systems designed for off-line operation.

When specifying the equipment for the plant-to-computer and computer-to-plant links, the following points must be carefully considered.

(1) What type of transducer is necessary to obtain the plant information?
(2) What conversion equipment is required to match transducer output to computing controller input?
(3) Is the transducer located so far from the controller that special data transmission techniques are necessary?
(4) What conversion equipment is required to match the computer output to the actuator?

A.2 Transducers. Most of the transducers available today are of the analog type, although digital transducers do exist. A digital transducer is a measuring device that produces a numerical output signal representing the measured quantity. Most of them are analog devices which have a built-in analog-digital converter. Generally, the choice of a transducer depends on (1) balancing the transducer characteristic against desired system

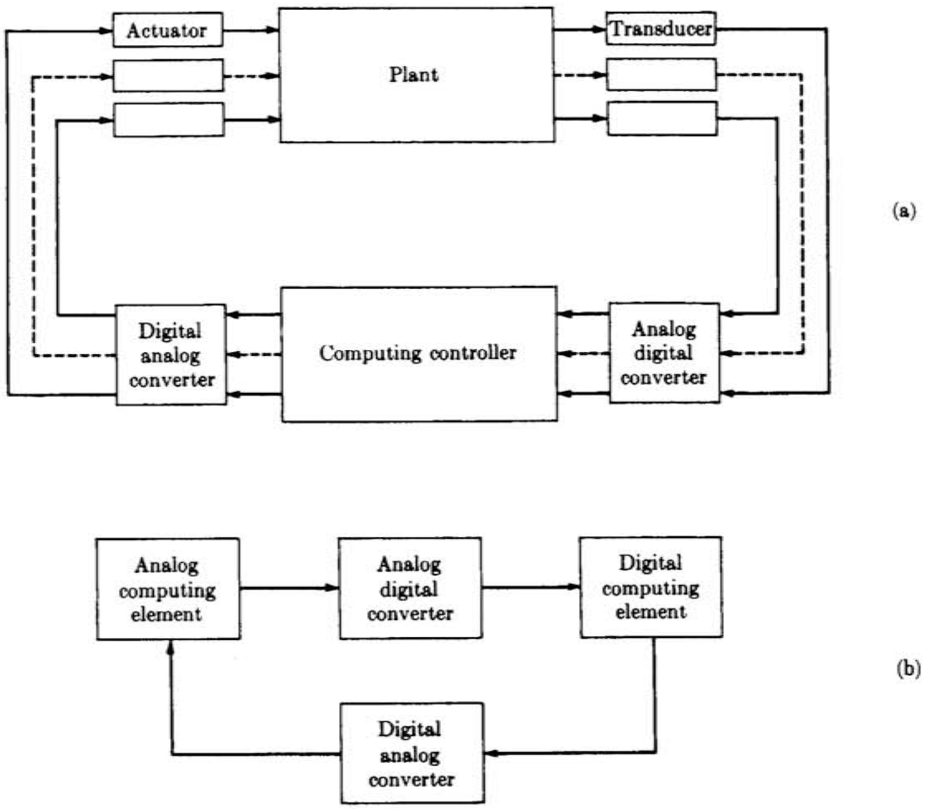

FIGURE 25. Fundamental conversion processes (analog-digital and digital-analog) for connecting (a) Plant to computing controller and back to plant via transducers and actuators; (b) Analog and digital computing elements.

performance, and (2) the input requirements of the controller. The different types of existing transducers have been surveyed in various publications.†

In the domain of digital transducers, several mechanical input devices with integral digital conversion are available for measuring angular displacement and linear motion. Angular displacement is measured by encoders which sense shaft position by means of coded disks and associated brushes. A typical example of a coded disk is shown in Fig. 26. Various methods of coding are used. One method is to photograph the pattern onto a glass disk so as to produce opaque and transparent areas. A light source (which may be pulsed or continuous) is placed on one side of the disk and, depending on the position of the shaft carrying the disk, light will be transmitted or absorbed,

† See, for example, Holzbock, W. G. [12].

giving high- or low-voltage outputs from a set of photocell detectors on the opposite side. Another convenient method is to make the pattern of electrically conducting and nonconducting areas and to detect the difference by means of brush contacts.

A.3 Telemetering. It is not always possible to place the computer near the plant. Economics, physical inaccessibility, or other factors may make it necessary to run long lines from the transducer to the computer. Frequently, it is found that there is a concentration of transducers in one part of an industrial plant, and in such cases it may be advantageous to do the analog-digital conversion locally, using pulse-code modulation techniques to transmit the data to the computer.

A.4 Nature of Transducer Output Signals.

Low-Level Analog Electrical Signals. This is a very large class and is the most difficult to treat because of the very low signal level (0 to ± 10 mv).

High-Level Analog Electrical Signals. Some transducers, e.g., Honeywell Teletransmitter, have a transmitter combined with the transducer. These have standardized outputs in the range 0–5 mA or 0–5 v.

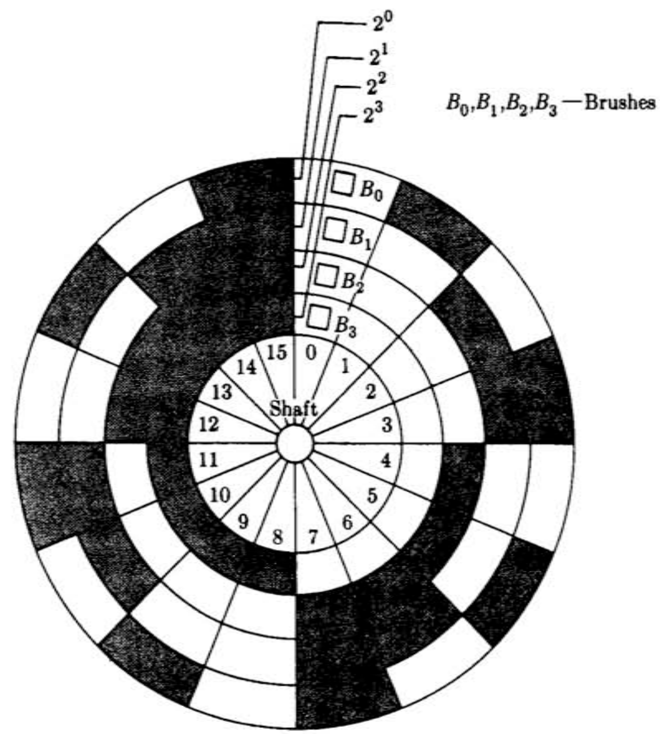

FIGURE 26. Diagram of a coding disk.

Analog Signals, e.g., frequency. Some transducers have outputs in the form of a frequency corresponding to the measured variable. Again, signals from radar or sonar are of pulse-time or frequency-modulated types.

Digital Signals. Signals in the form of a binary code (computer code) of sufficient level for the computer to handle.

Digital Signals. Signals in a code other than the binary, e.g., Gray, Binary Decimal, etc., as in shaft position encoders.

A.5 Organization of the Conversion Equipment Connecting Plant to Computer. A block diagram of the conversion equipment connecting the plant to the computer is shown in Fig. 27. The plant variables may be measured by either analog or digital transducers. Since the computer accepts the signals one at a time, suitable sampling and scanning operations must be carried out to route the information to it. This is done by the input selectors. Before analog-digital conversion can take place, the low-level analog signals must be amplified, this being done by the analog signal processor. Filtering of the low-level signals is also necessary.

A.6 Sampling and Filtering. Filtering is necessary for two reasons. The transducer may be located at a considerable distance from the computer and its signal may be seriously contaminated by noise in transmission. Also, as we shall explain later, since the analog signal is sampled it may be necessary to filter it in some way. It is known that a band-limited message can be transmitted by sampling at a rate two or more times the maximum message frequency f_m. The message can be completely recovered by passing the sampled message through an ideal low-pass filter with a cutoff frequency f_m. In practical systems, however, the signal spectrum is not band-limited and noise is necessarily introduced. These facts prevent the signal from being recovered with zero error. By proper choice of presampling and smoothing filters, the signal may be recovered with negligible error [23].

A.7 Scanning and Routing of Information. Referring to Fig. 27, the various scanning operations may be done according to a fixed clock signal, or a given input may be selected with a program order specifying the input address. This address must be decoded to operate the appropriate input selection switch. Decoding is usually achieved by decoding matrix techniques.

The address decoder sends a signal to operate a particular selection switch, of which the form depends on the input itself.

Where the input is digital, selection may be made with logic elements, such as the AND gate. For analog signals, the switch is either a relay or a semiconductor switch. A relay will generally be used for extremely low-level signals. (Mercury-wetted contacts are used to increase the life of the relay.) Semiconductor switches may be used when the analog signal level is high (100 mv or more). It should be noted, however, that the switching properties of these need careful examination.

A.8 Amplification and Signal Conditioning. Low-level amplifiers are necessary to bring the signals to a sufficiently high level for reasonably

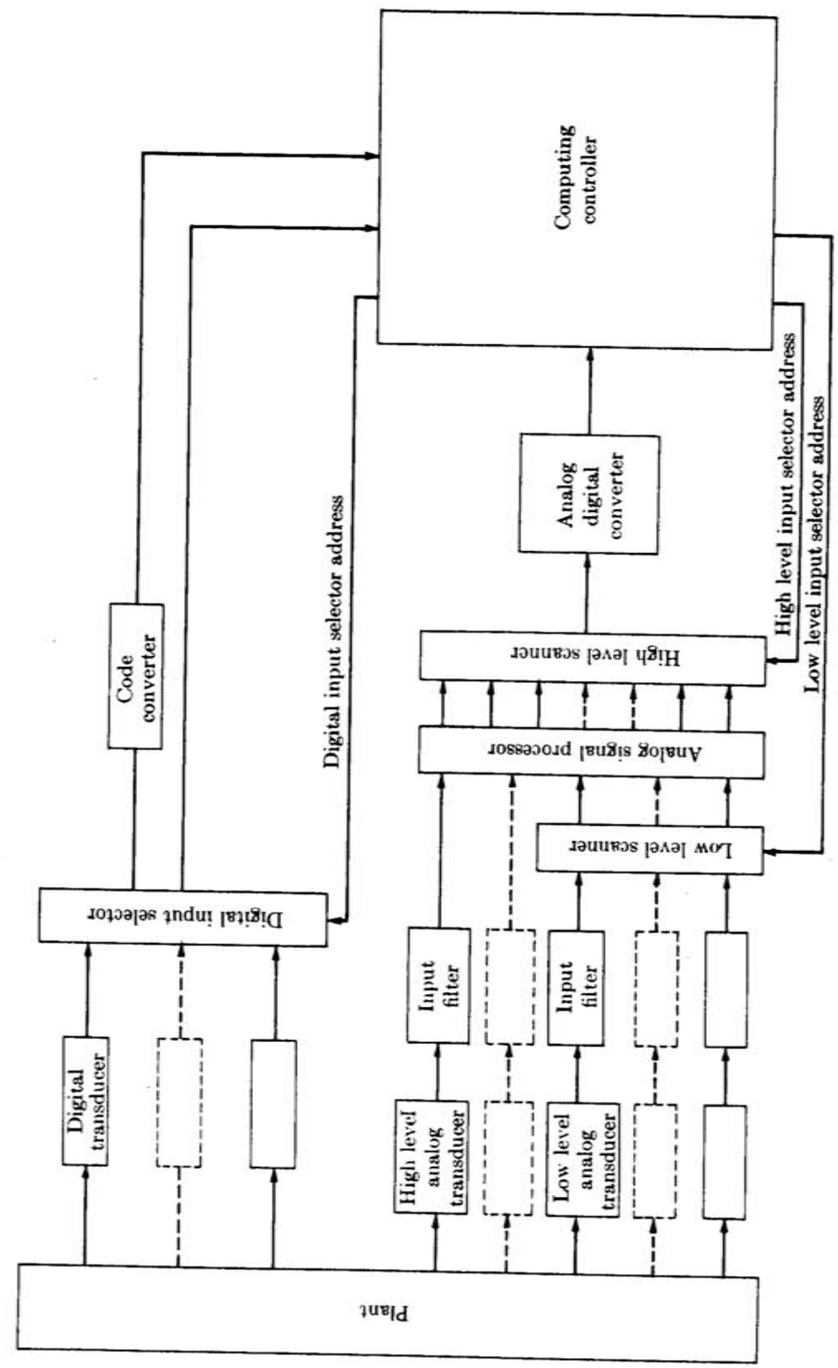

FIGURE 27. Block diagram of the conversion equipment connecting plant to computing controller.

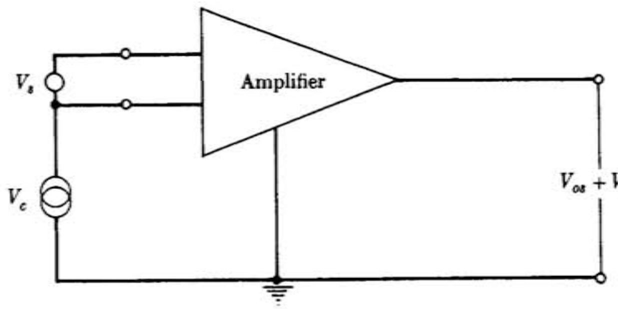

FIGURE 28. Differential and common mode input signals for a differential amplifier.

accurate digitizing in the analog-digital converter. Amplifications up to 1000 are typical.

Attenuation and gain changing are also usually necessary to allow proper scaling of the input channels to accept a wide range of input signal levels.

Low-level amplification is one of the most complex problems of on-line instrumentation. Besides drift, gain, stability, and nonlinearity, the most serious problem the designer has to face is "common-mode rejection."

In most computer control applications, amplification and conversion equipment is some distance from the measuring transducer so the ground potential at the amplifier and conversion equipment will be different from that at the transducer. In order to eliminate ground loops, the input of the amplifier must be "differential." In Fig. 28 V_s is the transducer voltage to be amplified and V_c is the voltage to ground which is common to both the terminals. V_c is termed the "common-mode voltage." Ideally, the output voltage V_0 of the amplifier should be insensitive to the common-mode voltage V_c. However, since the two parts of the differential amplifier are not exactly identical with respect to voltage gain and input resistance, a resultant error signal is produced at the output due to V_c. A figure of merit, often associated with a differential amplifier, is the "common-mode rejection ratio" K_{cm}. It is defined as the ratio between the differential gain and the common-mode gain of the amplifier.

If G_D = differential gain of the amplifier;
G_{cm} = common-mode gain of the amplifier;
V_s = differential input signal;
V_c = common-mode input signal;
V_{os} = output signal due to differential input;
V_{oc} = output signal due to common-mode input;
then,

$$K_{cm} = \frac{G_D}{G_{cm}} = \frac{V_{os}/V_s}{V_{oc}/V_c} \tag{A.1}$$

The higher K_{cm}, the better the amplifier, so far as common-mode signals are concerned. Contemporary designs of low-level differential amplifiers have common-mode rejection ratios of 10^6 to 10^7 at dc.

A.9 Analog-Digital Conversion. Once the signal has been amplified it must be converted into the appropriate digital form for introduction into the computer.

Analog-digital converters may be divided into two major groups according to the encoding process. In one group are those which depend upon a two-dimensional geometric pattern for quantization, while in the other are those which depend on a suitable arrangement of electronic circuitry for encoding. The former group may be referred to as geometric converters and the latter as electronic converters.

Comprehensive reviews of analog-digital converters already exist [3, 24]. In this section we shall describe the basic principles underlying their operation and indicate some trends in their development.

Geometric Converters. The operation of geometric converters is identical to that of coding wheels. Here we add a few words regarding the accuracies obtainable with such converters and indicate a further refinement of the basic device.

It is evident that as the resolution of geometric converters increases the area available for one of the least significant digits decreases, for a given over-all pattern size, so the precision engineering problems are increased. Consequently, one factor limiting the resolution of geometric converters is mechanical engineering practice [11]. Optical methods and etched circuit techniques are now being widely used to alleviate this problem, digit packings of 1000 per inch of track being achieved in present-day converters. In order to overcome the difficulty of producing a large number of small areas on one disk, it is possible to connect two disks through suitable reduction gearing. Work on problems associated with backlash and eccentricity has resulted in the adoption of nonlinear couplings, such as single-toothed gears, between the disks.

At present, geometric shaft position encoders are capable of resolutions up to 1 part in 2^{19}, by using two geared shafts. Patterns on one disk may be made to a resolution of 1 part in 2^{16}. Sampling rates of about 500 per second are possible with these encoders.

The fastest type of converter in this class accepts a voltage input, which becomes the displacement of the electron beam in a cathode-ray tube. A rectangular binary-coded mask is placed on the face of the tube so that a train of light is generated when the beam is deflected at right angles to the signal deflection. Alternatively, the pattern may be permanently photographed on the target of the tube to generate an electron pulse train. Sweep rates of a few million per second are possible, allowing independent conversions to be made at this rate with a suitably fast read-out mechanism. To achieve 10-digit accuracy, however, severe limitations are placed on the geometry of the tube and on the linearity of the wide-band input amplifiers. For a light-obstructing mask on the face of the tube, conversion speed is limited by the response time of the photocells to about 20,000 conversions per second.

In most of these devices, the digital code may be read from the pattern either serially or in parallel, depending on specific requirements. Codes available include the normal binary code, binary-coded decimal, and various forms of cyclic progressive, e.g., Gray, code. A cyclic progressive code is generally considered to be essential in geometric converters. Referring to the normal binary code pattern, (Fig. A26), it can be seen that very small positional differences between the read-out elements can cause large errors in the recorded value when the transition line between numbers happens to be under the read-out position. For the transition from 1000 to 0111, misalignment of the read-out device can give numbers bearing no relation to the true position of the variable. This is the worst case, but there can be ambiguity in the least significant two digits at every second transition. Transitions between adjacent numbers in the Gray code, however, involve a change in only one digit place and therefore misalignments can cause an uncertainty of only one level of quantization.

All-Electronic Converters. The basic principles underlying the design of electronic converters are:

(1) time-encoding;
(2) encoding by comparison;
(3) feedback encoding.

Time-Base Encoder. A block diagram of a time-base encoder is shown in Fig. 29a [22]. The conversion of the voltage signal to a time signal makes use of a saw-tooth voltage which sweeps upward at each conversion interval to an amplitude higher than that of the analog voltage input. From Fig. 29b it can be seen that the times t_1, t_2, t_3 necessary for the saw-tooth voltage to change from a fixed reference voltage to the analog voltage signal to be coded is directly proportional to the analog voltage at that instant. The time intervals t_1, t_2, t_3 are measured by a counting circuit. At the instant corresponding to point A in Fig. 29, the gate is opened to allow a set of pulses to flow into the binary counter. At the instant corresponding to point B, the gate is closed. Thus the counter starts at point A and stops at point B, where the sweep voltage is equal to the analog voltage being coded. The number of pulses from the clock pulse generator passing through the gate during the time interval is the number counted on the binary counter. Some time after that, e.g., at the point C, a pulse is generated to reset the counter to zero. Shortly thereafter, at the instant corresponding to the point A, the gate is opened again and the conversion cycle repeats.

The resolution of such a conversion device depends on the number of clock pulses contained in the full sweep interval. This number is inversely proportional to the size of the quantizing steps of the signal sample. If the frequency of the clock-pulse generator is so chosen that 128 pulses are generated during the full sweep interval, the resolution would be 1 part in 128. Since the maximum number of pulses in the full sweep interval is fixed, the number of pulses passing through the gate indicates not only the amplitude of the analog voltage being encoded, but also the exact time at which it is being measured. Since the signal may

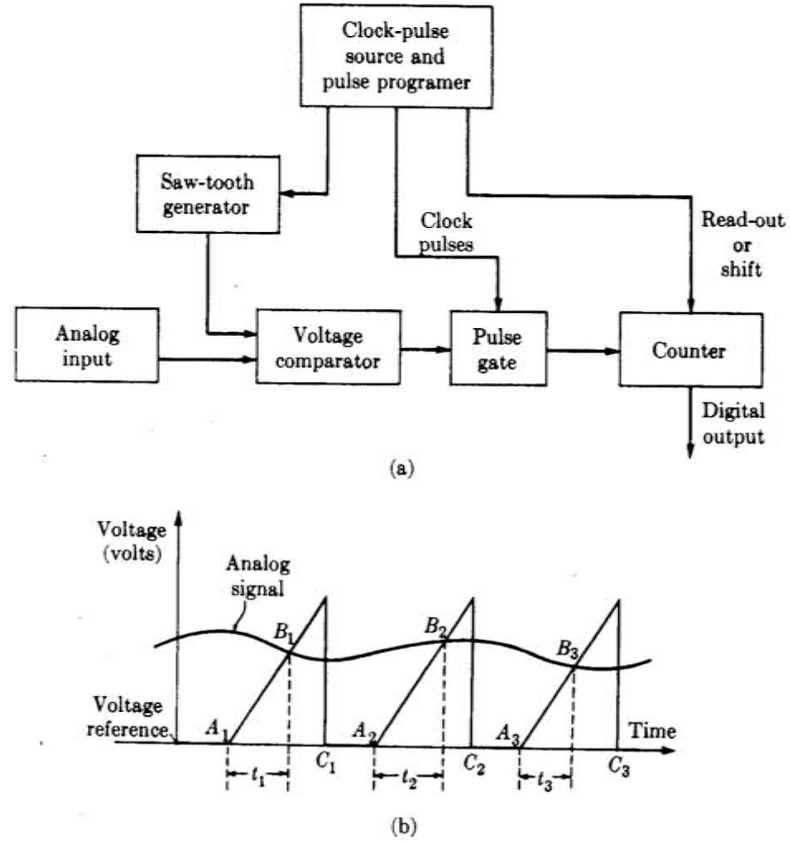

FIGURE 29. (a) Block diagram of time-base encoder. (b) Conversion of analog voltage into time duration.

fluctuate during a sweep interval, a temporary storage or clamping device is generally employed in order to sample the signal at precise intervals and to hold the voltage constant during the sweep period. By this means, the readings can be obtained at constant intervals. The accuracy of the time-base encoder is determined by the linearity of the sweep-signal waveform and the accuracy of the comparison circuit. The conversion time depends on the counting speed of the counter and the maximum number N to be coded. If the time required for reading and clearing the counter is t_r, the conversion time t_c is given by

$$t_c = \frac{N}{v_c} + t_r \tag{A.2}$$

where v_c = counting speed.

The highest operating speed of a practical counter is about 5 mc/sec; it usually takes a minimum of 1 μsec to read and reset the counter. Thus the lowest conversion time which can be attained is

$$t_c = \frac{N + 5}{5}\ \mu\ \text{sec} \tag{A.3}$$

For instance, if the maximum number to be encoded is of two decimal digits, then 7 bits are required in the binary code and the lowest conversion time would be 26.6 μsec. If the maximum number to be coded is of three decimal digits, 10 bits are required in binary notation and the conversion time would not be less than 205.8 μsec. The time-base encoder offers the advantage of simplicity of circuitry, ease of construction, and extremely simple logic circuits. The accuracy depends primarily on the drift of the comparator and the slope and linearity of the sweep generator. Accuracies of the order of 0.1% have been attained with such a converter.

Encoding by Comparison. A block diagram of a typical encoder using the principle of comparison is shown in Fig. 30 [24].

The operation of this type of converter is based on comparing the input voltage with a locally generated voltage which is varied by a control circuit until the two voltages agree. The state of the local voltage generator at the time of agreement is read out in digital form and thus furnishes the desired number. One method of operation is as follows. The most significant source S_n is turned on first by the control circuit and the voltage to be converted compared with S_n in the comparator. If the input voltage is greater than S_n, S_n is left on; if less, S_n is turned off. The control circuit then switches on the next most significant source S_{n-1}. Again, if the input signal is greater than S_{n-1}, it is left on and if less, S_{n-1} is switched off and so on. If a source is left on, a "1" is read out and if it is switched off, a "0" is read out.

As an example, let us consider a 4-digit binary coder capable of handling inputs ranging from 0 to 15 v. The most significant source S_3 is $2^3 = 8$ v. The next is 4 v, the following is 2v, and the least significant source, 1 v. For an input of 9 v, agreement is reached when the most and least significant sources are on, causing an output read-out of 1001.

Encoders of the comparison type are limited in accuracy by the accuracy of the trial voltage and the voltage comparison circuit. When the trial voltages are

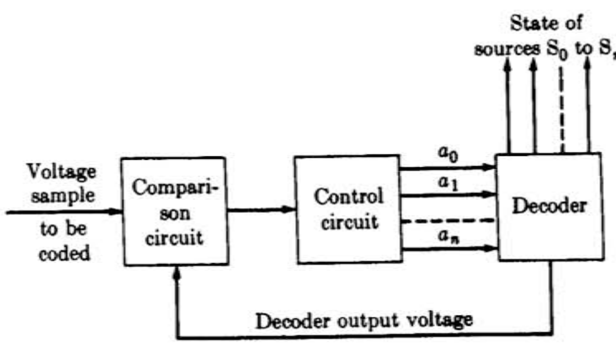

FIGURE 30. Block diagram of comparison type coder.

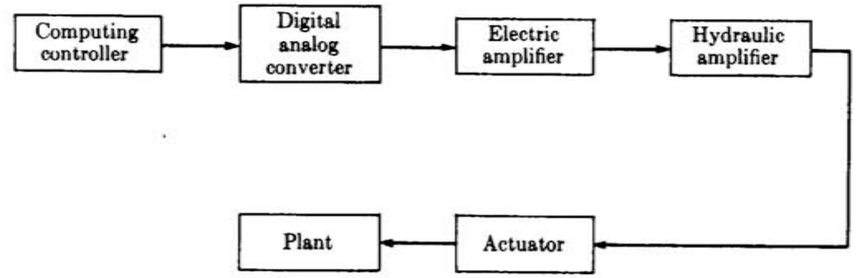

FIGURE 31. Typical link connecting computing controller back to the plant.

obtained from voltage dividers, the conversion accuracy also depends upon the stability and accuracy of the reference voltage and the resistors of the voltage dividers. The conversion speed of this type is usually faster than that of time-base encoding.

Recent Developments in Analog-Digital Conversion. Recent developments in analog-digital conversion appear to be following two trends:

(1) development of very high speed converters;
(2) low-level converters.

In the field of high-speed converters, conversion times of 25 μsec and sampling rates of 5×10^6 samples per second have recently been achieved [8]. Direct conversion of low-level analog signals (0—10 mv) is also being attempted. This eliminates the complicated problem of dc amplification. A magnetic core device which converts the low-level signals to a corresponding time duration has recently been developed [27].

Connection between Computer Output and Process. A typical link connecting the computing controller back to the plant is shown in Fig. 31. We shall examine the various elements of the link.

A.10 Digital-Analog Conversion.

If the computing controller is a digital computer, the signals from it have to be converted into a suitable analog form for the actuator. This conversion is necessary since most of the actuating devices are analog.

Digital-analog converters may be broadly divided into two types:

(1) those which convert the binary number into an equivalent electrical signal;
(2) those which convert the binary number into an equivalent physical motion.

Converters of Type (1). The method consists of using the numerical representation to switch electrical sources into a network in accordance with the number to be decoded. The principles involved in the design of the various possible network configurations are more or less the same. We shall describe one scheme to illustrate these principles. A simplified diagram of a decoder of this type is shown in Fig. 32 [24]. When the jth digit is One, switch S_j is closed and the current

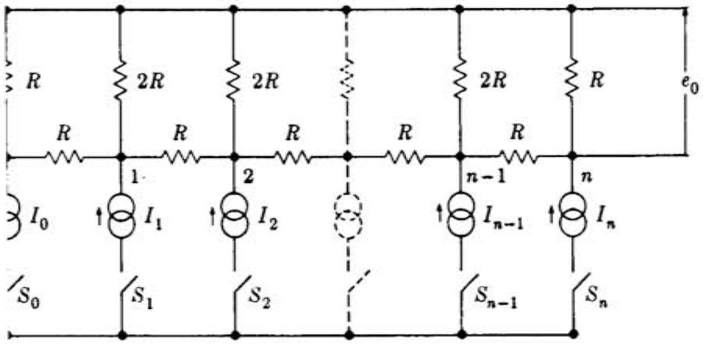

FIGURE 32. Voltage decoder with ladder network.

ɔntributes to the output voltage e_0. When the jth digit is Zero, switch ɜnd I_j does not contribute to the output. All current sources I_0 to I_n l magnitude I and, by definition, a current source has infinite imped- lows then that the presence of absence of all the other sources does not oading of I_j when S_j is closed; the only loading on I_j is that repre- :he resistive network consisting of R and $2R$. The configuration is so ıat the load on every interior stage ($j = 1$ to $j = n - 1$) is $2R$ looking of node j, $2R$ looking to the right of node j, and $2R$ looking up. The ›ry interior stage is, therefore

$$\frac{1}{R_L} = \frac{1}{2R} + \frac{1}{2R} + \frac{1}{2R} = \frac{3}{2R} \tag{A.4}$$

$$R_L = \frac{2R}{3}. \tag{A.5}$$

shown that each source j sets up between its node j and ground a ich is given by $E = \dfrac{2R}{3} I$

$$e_0 = \frac{E_0}{2^n} + \frac{E_1}{2^{n-1}} + \cdots + \frac{E_n}{2^0}. \tag{A.6}$$

$$E_j = \frac{2R}{3} I_j \text{ where } I_j = 0 \text{ or } I.$$

ber to be decoded is given by

$$p = a_n 2^n + a_{n-1} 2^{n-1} + \cdots + a_0 2^0 \tag{A.7}$$

ite:

$$e_0 = \frac{2}{3} RI \left(\frac{a_0}{2^n} + \frac{a_1}{2^{n-1}} + \cdots + \frac{a_n}{2^0} \right)$$

$$= \frac{2}{3} RI \frac{1}{2^n} p. \tag{A.8}$$

Therefore the output voltage is a linear function of the number to be decoded.

Incremental Position Decoder. In an incremental position decoder the number is first converted into a train of pulses. These pulses may then be used to drive a quantized electromechanical transducer. The number of pulses in the train is a measure of the digital number, and each pulse represents one position increment. An incremental feedback decoder is shown in Fig. 33.

A.11 Amplification. The command signal generated in a computing controller, after being converted into the correct form in the digital-analog converter, may require several stages of amplification before it can actuate the final mechanism.

On-off control is the simplest form of amplification. A low-power signal can energize a relay capable of handling a much higher power level, and relays may be cascaded to achieve high gains. Electronic amplifiers, both tube type and transistor type, may be used to raise the computer output signal to a level capable of operating relays, solenoids, electric motors, etc.

Another important element for electronic amplification is the thyratron and its modern form, the semiconductor-controlled rectifier.

When reliability and the ability to withstand shock and vibration are essential, magnetic amplifiers are often preferred to electron tube amplifiers.

Electric motors are also used. They not only amplify, but at the same time convert the signal into a mechanical form.

Hydraulic amplifier systems are efficient, extremely compact, fast, and capable of high torque output.

A.12 Actuation. The whole range of final plant variables—temperature, pressure, liquid level, flow-rate, machined shape, humidity, etc.—can be controlled by liquid flow, linear or angular displacement, or a rate of displacement. As flow is regulated by the motion of a valve or pump, so the final controlled output is a movement or a rate of movement of an actuating

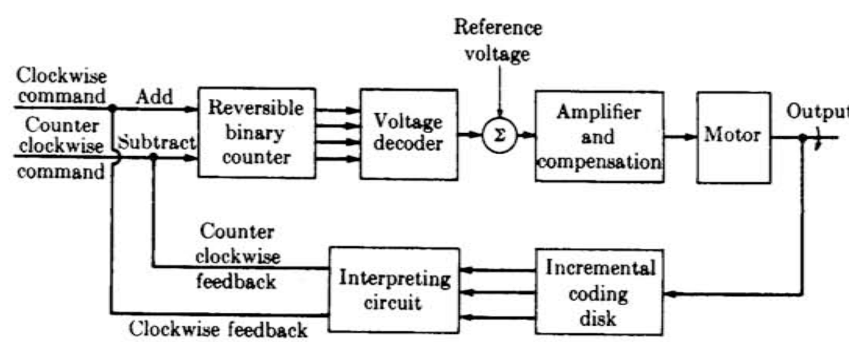

FIGURE 33. Block diagram of incremental feedback decoder servo.

element. The only obvious exception occurs when electric power is modulated and even here the final control is often by means of a displacement.

Liquid or gas flow may be used to control many physical and chemical processes. For liquids under pressure valve motion can act as the passive actuating element. Pumps can be used to actively control the flow. Electric motors are most widely used to control displacement and speed.

Where linear displacement involves large forces and high performance requirements, hydraulic or pneumatic devices are often preferred as actuators because of their direct linear motion output and their compactness and high response characteristics.

In some processes, actuation is carried out by the controlled flow of electric power rather than by mechanical motion or fluid flow.

Although actuators today are mostly analog devices, more and more attention is being paid to direct digital actuation. Thus the signal from the digital computer may be used to actuate the process without undergoing digital-analog conversion, although some amplification may be necessary. One such device exists commercially—the power stepping motor [25]. These motors respond to any source of pulses where the frequency of the pulses is proportional to the desired motor speed and the number of pulses is proportional to the desired angular displacement. Motors with speeds up to 100,000 steps/min and torque outputs to 3000 in.-lb have been built. Another type of digital actuator is a cylinder with a piston whose stroke varies as 2^n.

BIBLIOGRAPHY

1. Adriaenssens, G. A., *Applications Industrielles des Calculateurs Numeriques*, IBM-Belgium, 1962, 50 pp.
2. Bellman, R., *Adaptive Control Processes: a Guided Tour*. Princeton, 1961. 256 pp.
3. Bower, G. C., "Analog to digital converters," Control Eng. (April, 1959).
4. Carpentier, J., "Contribution à l'Etude du Dispatching Economique," Electricité de France (February 15, 1952), 78 pp.
5. Carter, L. V. P., "Optimization Techniques," TN 18.048. IBM Nordiska Laboratorier (June 12, 1961), 28 pp.
6. Chatterjee, H. K., "Multivariable Process Control," *Proceedings of the IFAC. Moscow, 1960*, Butterworths, 1961.
7. Coales, J. F., and Noton, A. R. M., "An on-off servomechanism with predicted change-over," Proc. IEE Paper No. 1895 M (July, 1956).
8. Crocker, C., and Prager, M., "A Technique for Converting Analog Voltages to Digital Codes at Sampling Rates above 5 Million Samples per Second with Accuracies of Seven Bits," IRE Natl. Convention Record, Globecom (May, 1961).

9. Debroux, A., *et al.*, "Code Apache destiné a la Programmation d'un Problème Analogique au moyen d'un Calculateur Digital," Rapport CETIS 30, Euratom (September, 1961).
10. Draper, C. S., and Li, Y. T., "Principles of Optimalizing Control Systems and an Application to the Internal Combustion Engine," ASME Publication (1951).
11. Herring, G. J., *Analogue-to-Digital Conversion Techniques: Progress in Automation*, Butterworths, 1960.
12. Holzbock, W. G., *Instruments for Measurement and Control*, Reinhold, 1955.
13. Kalman, R. E., and Koepcke, R. W., "Optimal Synthesis of Linear Sampling Control Systems Using Generalized Performance Indexes," ASME Paper No. 58-IRD-6, 7 pp.
14. Karlin, S., *Mathematical Methods and Theory in Games, Programming, and Economics*, Vol. 1, Addison-Wesley, 1959.
15. Kaufmann, A., *Méthodes et Modèles de la Recherche Opérationnelle*, Dunod, 1962, 534 pp.
16. Kirchmayer, L. K., "Optimalizing Computer Control in the Electric Utility Industry," *Proceedings of the IFAC, Moscow, 1960*, Butterworths, 1961.
17. Korn, G. A., "The impact of hybrid analog digital techniques on the analog computer art." Proc. Inst. Radio Engrs. 1077–1087 (May, 1962).
18. Nishida, F., "Synthesis of Multivariable Control Systems by Means of Sampled Data Compensations," *Proceedings of the IFAC, Moscow, 1960*, Butterworths, 1961.
19. Orchard-Hays, W., "The evolution of programming systems." Proc. Inst. Radio Engrs. 283–295 (January, 1961).
20. Ragazzini, J. R., and Franklin, F. G., *Sampled Data Control Systems*, McGraw-Hill, 1958.
21. Rozonoer, L. I., "The maximum principle of L. S. Pontriagin in optimal system theory, I, II, III," Automation and Remote Control Nos. 10, 11, and 12 (1959).
22. Slaughter, D. W., "An analog-to-digital converter with an improved linear sweep generator," IRE Natl. Convention Record Part 7, 7–12 (1953).
23. Spilker, J. J., "Theoretical bounds on the performance of sampled-data communication systems," IRE Trans. on Circuit Theory (1960).
24. Susskind, A. K., *Notes on Analog/Digital Conversion Techniques*, Technology and Wiley.
25. Thomas, A., and Fleischauer, F., "The power stepping motor—a new digital actuator," Control Eng. 74–81 (October, 1957).
26. Truxal, J. G., *Automatic Feedback Control System Synthesis*, McGraw-Hill. 1955, 675 pp.
27. van Praag, V. A., Stanks, William, and van Minders, David, "Magnetic core converts voltage to pulse duration," Control Eng. (1961).
28. Weinstock, R., *Calculus of Variations*, McGraw-Hill, 1952.