of programs are slightly over 3000 orders in length. The program requires about 40 seconds to generate the canonical expansion for each output line of a 12 input-line by 14 output-line problem and from eight to ten minutes to minimize and print the final expression. About 25,000 registers are required to store partial results during the processing. As a result, drum storage is used during the minimization procedure. The procedure is now being programmed for an IBM 709 located at the laboratory, making possible the solution of larger problems and somewhat shortening the programs' running time due to the large core storage of the 709.

The design procedure described here appears very flexible. It can be used to perform automatically the logical design of circuitry which will perform any function which has a unique value of the dependent variable for each value of the independent variable.

To date, networks which yield sine, arc sine, and the square root of the input value have been constructed. The concept of programmed logic as an aid to computer design appears quite attractive for the design of future machines.

ACKNOWLEDGMENT

The computer programs described in this report were written by H. C. Peterson who has also contributed many helpful suggestions. The original problem of designing computer instructions which would yield transcendental operations arose during Lincoln Laboratories' ballistic missile early warning system studies and was suggested to the author by W. I. Wells. The author also wishes to thank V. A. Nedzel, R. W. Sittler, and J. F. Nolan for their helpful comments during the writing of this report.

The Role of Digital Computers in the Dynamic Optimization of Chemical Reactions

R. E. KALMAN† AND R. W. KOEPCKE‡

I. INTRODUCTION

ALONG with the increasing availability of high-speed, large-storage digital computers, there has been growing interest in their utilization for real-time control purposes. A typical problem in this connection and one of long-standing interest is the optimal static and dynamic operation of chemical reactors.[1,2] To our knowledge, no digital computer is being used for this purpose, chiefly because of the many difficulties encountered in utilizing real-time machine computation in reactor control. These difficulties range from the unavailability or inadequacy of hardware (*i.e.*, transducers, measuring instruments, low-level analog-to-digital converters, etc.) to the lack of a well-established body of fundamental theoretical principles. Although a great deal is known about the basic concepts governing control systems,[3,4] present methods cannot be readily applied to designing a program for a real-time digital control computer. This is because the existing design methods are applicable primarily to fairly small-scale systems, whereas the use of a digital computer (in fact the very attractiveness of computer control) arises primarily in connection with large-scale problems.

The role of the digital computer in real-time control consists essentially of "digesting" large amounts of information obtained from the primary measuring instruments and then calculating, as rapidly as possible, the control action to be taken on the basis of these measurements.

One purpose of this report is to provide a broad outline of a new approach to designing control systems for chemical processes which are to be built around a fast, general-purpose digital computer operating in real time. The specific engineering details of the computer will not be of any interest here; rather, we have concentrated on studying the types of computations the computer is to perform. To lend concreteness to the discussion, the chemical process under consideration will be a continuous-flow, stirred reactor. After the fundamental concepts have been established, the detailed analytic equations (in the linear case) leading to the dynamically optimal (and thus also statically optimal) design of the reaction control system are given in Section III. The equations of Section III represent a special case of the new design theory of linear control systems formulated

† Res. Inst. for Advanced Study, Baltimore 12, Md.
‡ IBM Res. Center, Yorktown Heights, N. Y.

[1] T. J. Williams, "Chemical kinetics and the dynamics of chemical reactors," *Control Engrg.*, pp. 100–108; July, 1958.

[2] R. Aris and N. R. Amundson, "An analysis of chemical reactor stability and control," *Chem. Engrg. Sci.*, vol. 7, pp. 121–155; 1958.

[3] J. G. Truxal, "Automatic Feedback Control System Synthesis," McGraw-Hill Book Co., Inc., New York, N. Y.; 1955.

[4] J. R. Ragazzini and G. Franklin, "Sampled-Data Systems," McGraw-Hill Book Co., Inc., New York, N. Y.; 1958.

by the authors.[5,6] The performance of the dynamically optimized control system is illustrated with the aid of a numerical example.

In Section IV the limitations of the linearity assumption or, rather, the additional steps necessary to attack realistic practical problems, are briefly discussed. It is impossible to give more than a rough sketch of these new methods in a short report; however, specific details, mathematical proofs, and discussion of engineering problems may be found in the literature.[5–12]

One of the mathematical tools used in the new approach has been called *dynamic programming* by its developer, Bellman.[13] This is a new method for the solution of problems in the calculus of variations where *dynamic constraints* play the central role. It turns out that our new approach to the description of control-system dynamics leads to concepts which are also the "natural setting" for solving the optimization problem by dynamic programming. A second purpose of this report is to provide a better appreciation of the advantages as well as the limitations of dynamic programming, thereby promoting its use in the solution of engineering problems.

Perhaps the most outstanding advantage of the use of dynamic programming in our problem is that it reveals the intimate connection between the static and the dynamic optimization of the process. In other words, the problem of selecting the operating conditions of the process to obtain optimum yield or optimum product quality cannot be realistically divorced from the problem of providing effective regulation to maintain the process at these conditions. Although these matters are well known to workers skilled in the control art, they are often not clearly understood by others.

In addition to providing some practical means for the solution of reactor control problems, it is hoped that this report will help clarify a number of basic questions.

[5] R. E. Kalman and R. W. Koepcke, "Optimal synthesis of linear sampling control systems using generalized performance indexes," *Trans. ASME*, vol. 80, pp. 1820–1826; 1958.

[6] R. E. Kalman and R. W. Koepcke, "Dynamic optimization of linear control systems. I. Theory. II. Practical aspects and examples." (Scheduled for publication in the *IBM J. Res. Dev.*, vol. 4; 1960.)

[7] R. E. Kalman and J. E. Bertram, "General synthesis procedure for computer control of single and multiloop linear systems," *Trans. AIEE*, vol. 77, pt. 2, pp. 602–609; 1958.

[8] R. E. Kalman, "Optimal nonlinear compensation of saturating systems by intermittent action," 1957 IRE WESCON CONVENTION RECORD, pt. 4, pp. 130–135.

[9] R. E. Kalman and J. E. Bertram, "A unified approach to the theory of sampling systems," *J. Franklin Inst.*, vol. 267, pp. 405–436; 1959.

[10] R. E. Kalman, L. Lapidus, and E. Shapiro, "On the optimal control of dynamic chemical and petroleum processes." (Scheduled for publication in *Chem. Engrg. Progress.*)

[11] P. E. Sarachik, "Cross-coupled multi-dimensional feedback control systems," Ph.D. dissertation, Dept. of Elect. Engrg., Columbia University, New York, N. Y.; 1958.

[12] R. E. Kalman, "On the general theory of control systems," *Proc. Internatl. Congr. on Automatic Control*, Moscow, U.S.S.R., Academic Press, New York, N. Y.; 1960.

[13] R. E. Bellman, "Dynamic Programming," Princeton University, Press, Princeton, N. J.; 1957.

II. FUNDAMENTAL CONCEPTS

A. Description of Chemical Reactor

The continuous-flow, stirred-tank type of chemical reactor with which we shall be concerned here is shown in Fig. 1. The principal inputs to the reactor consist of liquid streams carrying the various raw materials. The volume-flow rates of the input streams in Fig. 1 are denoted by M_1, M_2, M_3. Each stream carries one or more compounds, whose concentrations (measured in terms of moles/unit volume in Fig. 1) are denoted by $U_1, \cdots, U_5$. Other inputs to the reactor may include a catalyst stream (with flow rate M_4 in Fig. 1) and provisions for cooling or heating (with heat-flow rate M_5 in Fig. 1). The numbers $X_1, \cdots, X_n$ denote the concentrations of the various compounds inside the reactor (some of which come from the input streams and some of which are formed chemically inside the reactor); one of the X_i will denote the temperature of the material inside the reactor. Due to agitation, the concentrations of the various compounds as well as the temperature are assumed to be approximately the same at every point inside the reactor and in the output stream. In most cases, it is desirable to keep the amount of material in the reactor constant. This is achieved by means of a level controller which keeps the output stream (F_0 in Fig. 1) at all times approximately equal to $M_1 + \cdots + M_4$.

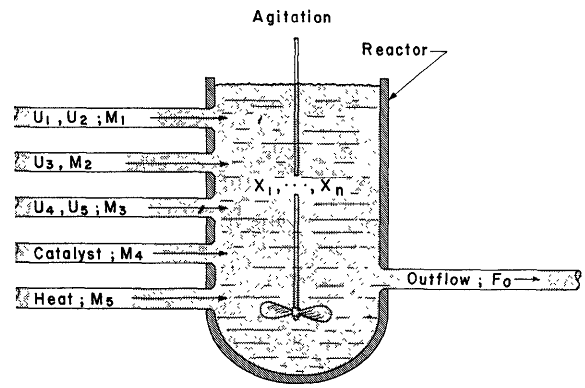

Fig. 1.

The object of the reactor is to produce a certain concentration of chemicals in the output streams. To accomplish this with the given types and concentrations of raw materials in the input streams, one can vary the flow rates $M_1, \cdots, M_5$. Since reactions take place more rapidly as the temperature increases, control can be exerted by changing the temperature in the reactor which, in turn, is achieved (subject to the dynamic lags of heat transfer to the reactor) by changing the heat-input flow-rate M_5. The amount of catalyst present in the reactor also affects the reactions; the amount is concontrolled by changing M_4 (subject to a time constant = reactor volume/F_0 if the amount of catalyst is not affected by the reaction). Similarly, some measure of control can be exerted by changing the flow rates M_1, M_2, M_3; the effect of these changes is complicated and depends on the reaction dynamics.

B. Statement of the Control Problem

The principal objectives in designing a reactor control system may be stated as follows:

Problem: Given the desired values $X_1{}^d, \cdots, X_n{}^d$ of the concentrations in the output stream at time t_0, manipulate the control variables in such a manner as to bring rapidly the actual concentrations existing in the reactor at time t_0 as close as possible to the desired concentrations and then keep the actual concentrations constant at all times despite changes in the concentrations of the input streams, ambient temperature, etc. If, at time $t_1 > t_0$, the desired values of the concentrations are changed, the above process is repeated.

We now examine this problem in more detail. In doing so, we shall specify precisely what is to be meant by "as close as possible" and "rapidly."

C. Reaction Dynamics

Let us assume that p molecules of compound A and q molecules of compound B combine chemically to form a new compound C. If the concentrations X_A, X_B, X_C, of the various compounds are small, the rate of increase of the concentration of compound C is given by the well-known Arrhenius equation.[1,2]

$$dX_C/dt = k_{AB}(T)X_A{}^pX_B{}^q. \tag{1}$$

In (1), the reaction rate coefficient is given by

$$k_{AB}(T) = \alpha_{AB} \exp(-E_{AB}/RT), \tag{2}$$

where α_{AB} is a constant, E_{AB} the activation energy of the reaction, T the absolute temperature, and R the gas constant. Moreover, the rate of decrease of the concentration of compounds A and B resulting from the reaction is equal to p resp. q times the right-hand side of (1).

In qualitative physical terms, the Arrhenius equation has the following interpretation. Consider a small volume with diameter equal to the effective range of intermolecular forces. If p molecules of A and q molecules of B have entered this small volume, a reaction takes place, but not otherwise. In a dilute solution, the probability of a molecule of some compound entering the small volume as a result of thermal agitation is proportional to the thermodynamic factor $\exp(-E_{AB}/RT)$ and the concentration of the compound, but independent of the concentration of the other compounds. The probabilities of independent events multiply, hence (1).

In general, the assumptions which lead to the particular form of (1) are not true, but the reaction rate is still a function of the temperature and concentrations. Thus, in general, one would replace (1) by

$$dX_C/dt = k_{AB}(X_A, X_B, X_C, T), \tag{3}$$

where k_{AB} is some scalar function of the four variables indicated.

It follows that the reaction shown in Fig. 1 can be described by the set of differential equations

$$dX_i/dt = f_i(X_1, \cdots, X_n; M_1, \cdots, M_l; U_1, \cdots, U_k) \tag{4}$$
$$(i = 1, \cdots, n;\ k, l, n = \text{integers}).$$

This is a good place, conceptually and in order to simplify the symbolism, to introduce vector-matrix notation. Thus, let X be a vector ($n \times 1$ matrix) with components $X_1, \cdots, X_n$. Similarly, M and U are defined as a ($l \times 1$) and ($k \times 1$) matrix, respectively; f is a vector function of $k+l+n$ arguments with components $f_1, \cdots, f_n$.

In terms of the new notation, (4) becomes

$$dX/dt = f(X, M, U). \tag{5}$$

The vector X is called the state of the reactor and the components of X are known as the state variables. The reason for this terminology is that if the reactor inputs $M(t)$ and $U(t)$ are specified for all time $t \geq t_0$, then the knowledge of $X(t_0)$ supplies the initial conditions from which the solutions of the differential equation (5) can be uniquely determined (subject to some mild mathematical restrictions) for all future values of time. Thus, the state is a fundamental mathematical concept for describing the reactor dynamics; it is also a physical concept. The temperature and various concentrations can be physically measured (at least in principle); thus the state at time t_0 may be regarded as the information necessary to determine the properties of the material inside the reactor at time t_0.

The behavior of the reactor through time may be visualized as a succession of changes in state. This gives rise to the concept of the state-transition function. In fact, the function f in the differential equation (5) may be regarded as specifying the incremental state transitions taking place during the interval $(t, t+dt)$. For present purposes, it is more convenient to deal with finite-interval state transitions which are obtained by solving the differential equations. Anticipating the later discussion, let us note that for control purposes it is sufficient to sample the state of the process; *i.e.*, observe the state only at discrete instants in time, called sampling instants. Usually, the sampling instants are separated by equal intervals τ of time (τ is called the sampling period), *i.e.*, the sampling instants occur at times

$$t_0,\ t_0 + \tau,\ t_0 + 2\tau,\ \cdots.$$

Now suppose that τ is chosen to be so small that in the interval $(t_0, t_0+\tau)$ the functions $M(t)$, $U(t)$ in (5) may be adequately approximated by the constants $M(t_0)$, $U(t_0)$. Then (5) can be readily integrated (if necessary, by numerical methods) and we get

$$X(t_0 + \tau) = \phi(\tau; X(t_0), M(t_0), U(t_0)), \tag{6}$$

where ϕ is a vector function with n components and $k+l+n$ arguments.

D. Static Optimization

Precisely what is meant by the phrase, "as close as possible to the desired concentrations" in the statement of the basic problem in Section II-B?

The states of the reactor may be represented as points in n-dimensional Euclidean space (the state variables being coordinates of the point) called the *state space*. Suppose we specify r components of the state vector as desired values, with the remaining $n-r$ components being arbitrary. This step may be regarded as essentially a management decision, relating to the question of how one should try to operate the reaction process. The set of states for which the operation of the reactor meets the management requirements is clearly an $(n-r)$-dimensional hyperplane. If the state of the reactor at any instant of time does not lie in the hyperplane, we can measure the "badness" of that state by the distance of the state from the hyperplane of desired states (see Fig. 2). The definition of the distance function (technically, a pseudo-metric) is arbitrary and depends on a management estimate as to what types of deviations from the desired values are more harmful than others. One possible definition of the distance function is

$$\rho(X^d - X) = \left[\sum_{i=1}^{r} (X_i^d - X_i)^2 \right]^{1/2} \qquad (r < n). \quad (7)$$

More generally, if Q is any positive semidefinite matrix, we can define ρ by the quadratic form

$$\rho(X^d - X) = (X^d - X)'Q(X^d - X), \quad (8)$$

where the prime denotes the transpose of the matrix.

By static optimization of the process we mean selecting a set of constant values M^0 of the control variables (subject to some magnitude constraints), so that at equilibrium the actual state lies as close as possible to the hyperplane of desired states. By definition, the equilibrium states X^* of the reactor are given by:

$$dX/dt = f(X^*, M, U) = 0, \quad (9)$$

M, U being constant vectors. Thus the statically optimal control vector M^0 and equilibrium state X^{*0} are determined by solving the minimization problem

$$\min_M \rho(X^d - X^*), \qquad 0 \leq M_i \leq \mu_i. \quad (10)$$

To find the optimal control vector M^0 from (10), X^* has to be expressed as an explicit function of M from (9). This and the amplitude constraints on the control variables lead to great analytic difficulties when f is a nonlinear function. But even in cases where the static optimization problem can be solved, it does not provide a complete answer to the basic problem. This is because:

1) Static optimization does not provide a guide as to how the control variables should be manipulated to bring an arbitrary state as close as possible (in terms of the arbitrarily adopted distance function) to the desired state (dynamic optimization).

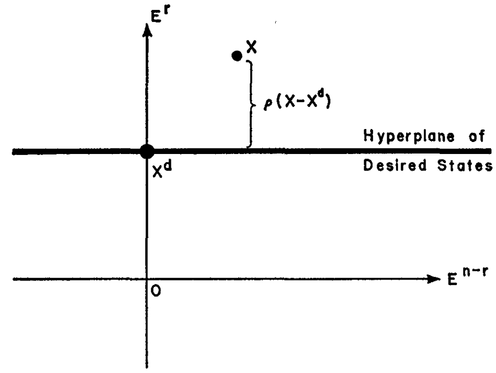

Fig. 2.

2) The equilibrium state closest to the hyperplane of desired states may not be stable.

3) The values of the control variables computed by static optimization will not remain optimal when some of the process parameters (concentrations in the input flows, ambient temperature, etc.) change. In other words, static optimization does not incorporate the important principle of feedback.

In the following it is shown that it is possible to combine both dynamic and static optimization in such a way that the principle of feedback is retained.

E. Dynamic Optimization

In our basic problem statement in Section II-B, the last remaining word to be defined precisely is "rapidly."

A performance index for the reaction under dynamic conditions may be defined as

$$\mathcal{P}[X(t_0)] = \int_{t_0}^{\infty} \rho[X^d - X(t)] \exp[\alpha(t - t_0)]dt, \quad (11)$$

where α is a real constant. We now agree that the phrase, ". . . to bring rapidly the actual concentration as close as possible to the desired concentrations . . ." in the problem statement means that the control variables $M(t)$ are to be chosen, as functions of time, in such a way as to minimize the performance index (11) for any initial state $X(t_0)$. This is called dynamic optimization. Of course, the definition of $\mathcal{P}$, in particular the value of α in (11), is arbitrary and depends on management estimates just as the definitions of X^d and ρ.

Static optimization is evidently a special case of dynamic optimization, as may be seen by setting $X(t_0) = X^{*0}$ in (11). In fact, it may happen that the dynamic optimization leads to the result that, instead of trying to maintain the control variables at constant (equilibrium) values, it is better to vary the control variables continuously, say, in a periodic fashion. In such a case, dynamic optimization will lead to a smaller value of $\mathcal{P}(X^{*0})$ than static optimization.

In order to perform dynamic optimization, we must find a particular vector function $M^0(t)$, defined for all $t \geq t_0$, among the set of all such functions (subject to amplitude constraints) for which the integral (11) assumes its minimum or least upper bound. This is generally a very difficult problem in the calculus of variations and, for all practical purposes, cannot be solved by conventional analytic methods when the number of state variables is large.

So as not to be bothered by certain mathematical niceties, we shall assume from here on that the control variables have constant values over the sampling intervals τ (cf. Section II-C).

Thus instead of minimizing $\mathcal{P}$ with respect to all possible functions $M(t)$, the minimization is to be performed with respect to all possible sequences of constant vectors

$$M(t_0),\, M(t_0+\tau),\, M(t_0+2\tau),\, \cdots \quad (0 \leq M_i(t) \leq \mu_i). \tag{12}$$

Moreover, again for simplicity, the integral in (11) may be replaced by a sum:

$$\mathcal{P}[X(t_0)] = \sum_{k=1}^{\infty} \rho[X^d - X(t_0 + k\tau)]\lambda^k, \tag{13}$$

where $\lambda = \exp \alpha\tau$.

From (6) we see that there is a large number of possible state transformations $X(t_0) \rightarrow X(t_0+\tau)$, depending on the choice of $M(t_0)$ (assuming that $U = \text{const}$). Similarly, the state transformation $X(t_0+\tau) \rightarrow X(t_0+2\tau)$ depends on the choice of $M(t_0+\tau)$ (see Fig. 3). Thus the minimization of $\mathcal{P}$ may be regarded as an infinite-step decision procedure. The optimal choice of $M(t_0+k\tau)$ at the kth step in general depends both on the preceding and succeeding steps. Therefore, at first sight, it would appear that to obtain the optimal sequence of control vectors,

$$M^0(t_0),\, M^0(t_0+\tau),\, M^0(t_0+2\tau),\, \cdots \tag{14}$$

we must simultaneously minimize (13) with respect to all the terms of the sequence (12), which is an impossible job.

Fortunately, at this point we can achieve a decisive simplification by making use of the following intuitively obvious, but powerful, observation due to Bellman.[13]

Principle of Optimality: An optimal sequence of control variables (14) has the property that, whatever the initial state $X(t_0)$ and the initial choice $M^0(t_0)$ of control vector are, the remaining terms $M^0(t_0+\tau)$, $M^0(t_0+2\tau)$, $\cdots$ of *(14) must constitute an optimal sequence with regard to the state $X(t_0+\tau)$ resulting from the choice of $M^0(t_0)$.*

Using the principle of optimality, we can obtain various expressions for the theoretical study and practical determination of the optimal control sequence (14). (Methods derived from the principle of optimality are known by the generic name of dynamic programming.)

We first observe that (13) can be written in the form:

$$\begin{aligned}\mathcal{P}[X(t_0)] &= \lambda\rho[X^d - X(t_0+\tau)] \\ &\quad + \sum_{k=2}^{\infty} \rho[X^d - X(t_0+k\tau)]\lambda^k \\ &= \lambda\{\rho[X^d - X(t_0+\tau)] + \mathcal{P}[X(t_0+\tau)]\}.\end{aligned} \tag{15}$$

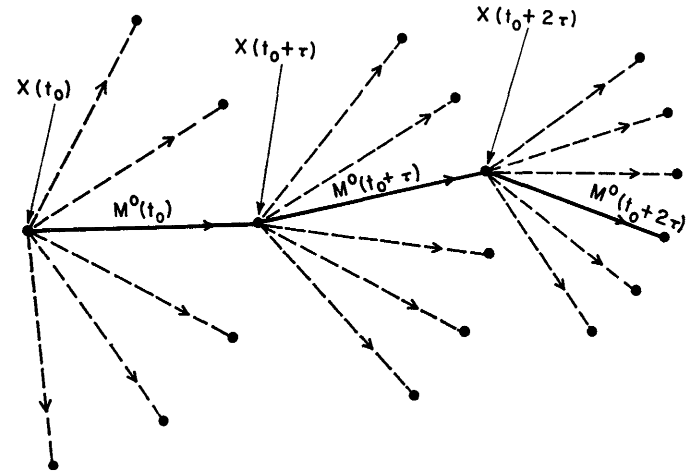

Fig. 3.

Now let $\mathcal{P}^0(X(t_0))$ be the value of the performance index when the optimal sequence (14) is used. Substituting (15) and invoking the principle of optimality, we obtain a functional equation (13) for $\mathcal{P}^0$; *i.e.*,

$$\begin{aligned}\mathcal{P}^0[X(t_0)] &= \min_{M(t_0),M(t_0+\tau),\cdots} \mathcal{P}[X(t_0)] \\ &= \min_{M(t_0)} \lambda\{\rho[X^d - X(t_0+\tau)] + \mathcal{P}^0[X(t_0+\tau)]\}.\end{aligned} \tag{16}$$

Note carefully that the right-hand side of (16) is a function of $M(t_0)$ and $X(t_0)$ through (6). The solution of the functional equation (16) determines $M^0(t_0)$ as some function of $X(t_0)$ which may be denoted by

$$M^0(t_0) = h(X(t_0)). \tag{17}$$

Now at time $t_0+\tau$ we are confronted by the same decision problem as at time t_0. This shows that $\mathcal{P}^0[X(t_0+\tau)]$ also satisfies the functional equation (16), and therefore $M^0(t_0+\tau)$ is the same function of the state at time $t_0+\tau$ as $M^0(t_0)$ was at time t_0. Thus we have arrived at the following result.

If the performance of a dynamic system governed by (6) is optimal in the sense that the performance index (13) is a minimum, then the sequence of optimal control variables is obtained by observing the state of the system at times

$$t_0, \quad t_0+\tau, \quad t_0+2\tau, \cdots$$

and computing the optimal control variables at each sampling instant by means of the formula

$$M^0(t_0+k\tau) = h(X(t_0+k\tau)), \tag{18}$$

h being determined by solving (16).

Eq. (18) shows that the feedback principle can be included in the framework of dynamic optimization. This means that the entire future evolution of the dynamic system, including the values of the optimal control variables at each sampling instant, could be predicted in principle by means of (6) and (18). However, because of the inaccurate knowledge of the state-transition function, unknown disturbances acting on the process, and random effects such as turbulence, etc., the prediction based on (6) will be less and less correct as the prediction interval increases. By re-measuring the state of the system at the sampling instants [assuming that τ has been chosen small enough so that the one-step prediction based on (6) is sufficiently accurate], the prediction errors are corrected so that the control variables assume very nearly their optimal values at all times. This is the conventional use of feedback. As is well known, feedback will also tend to minimize the sensitivity of h to variations in ϕ.

To solve the functional equation (16), it is often convenient to use an iterative procedure. To derive the iteration scheme, we replace the performance index (13) by

$$\mathcal{P}_N[X(t_0)] = \sum_{k=1}^{N} \rho[X^d - X(t_0 + k\tau)]\lambda^k. \qquad (19)$$

In other words, the original infinite-step decision process is converted into a finite-step decision process. Proceeding exactly as in the derivation of (16), we find that the successive optimal performance indexes $\mathcal{P}_N{}^0$ are connected by the recurrence relations:

$$\left.\begin{aligned} \mathcal{P}_1{}^0[X(t_0)] &= \operatorname*{Min}_{M(t_0)} \lambda\rho[X^d - X(t_0+\tau)] \\ \mathcal{P}_{N+1}{}^0[X(t_0)] &= \operatorname*{Min}_{M(t_0)} \lambda\{\rho[X^d - X(t_0+\tau)] + \mathcal{P}_N{}^0[X(t_0+\tau)]\} \end{aligned}\right\} 0 \le M_i(t_0) \le \mu_i. \qquad (20)$$

In each stage of the iteration (20), the optimal control signal $M^0(t_0)$ is determined as some function h_N of $X(t_0)$. As $N \to \infty$, it can be shown under various restrictions[6,13] that $\mathcal{P}_N{}^0$ converges to $\mathcal{P}^0$, and h_N converges to h.

III. Dynamic Programming in the Linear Case

The ease or difficulty of carrying out iterations (20) is determined largely by the complexity of the dynamics of the reaction and by the limits imposed on the control variables. To illustrate these computations concretely, we consider now the very special (but practically important) linear case where

1) The reaction dynamics are governed by an ordinary linear differential equation with constant coefficients

2) There are no amplitude constraints on the control variables.

Linear differential equations arise when the dynamic equations (6) are linearized about some equilibrium state X^* and the corresponding values of the control variables M^*. If we let

$$X = X^* + x, \quad M = M^* + m, \quad \text{and} \quad X^d = X^* + x^d, \qquad (21)$$

and, if the deviations x, m from the equilibrium values are sufficiently small, (6) leads to the linear differential equation with constant coefficients

$$dx/dt = Fx + Dm, \qquad (22)$$

where F is a constant $n \times n$ matrix and D is a constant $n \times l$ matrix. The elements of these matrices are determined from (6) by means of the formulas (assuming U=const)

$$\begin{aligned} F_{ij} &= \partial f_i/\partial X_j\big|_{X=X^*,\ M=M^*} \quad (i, j = 1, \cdots, n) \\ D_{ij} &= \partial f_i/\partial M_j\big|_{X=X^*,\ M=M^*} \\ &\qquad (i = 1, \cdots, n; j = 1, \cdots, l). \end{aligned} \qquad (23)$$

As is well known,[14] the solution of the differential equation (22) has the form:

$$x(t) = \Phi(t - t_0)\, x(t_0) + \int_{t_0}^{t} \Phi(t - \tau) Dm(\tau) d\tau \qquad (24)$$

for any t, t_0. The matrix $\Phi(\tau)$ is called the *transition matrix* of the system (22) and is given by

$$\Phi(\tau) = \exp F\tau = \sum_{k=0}^{\infty} F^k \tau^k / k! \qquad (25)$$

The Taylor series is a convenient way of calculating numerical values of $\Phi(\tau)$ when there are a large number of state variables and when a digital computer is available. There are also analytic ways of computing $\Phi(\tau)$.[5,7]

When $m(t)$ is constant during the intervals between sampling instants, (24) takes the simpler form

$$x(t_0 + \tau) = \Phi(\tau)\, x(t_0) + \Delta(t) m(t_0), \qquad (26)$$

where

$$\Delta(\tau) = \int_0^{\tau} \Phi(\tau - \sigma) D d\sigma. \qquad (27)$$

Eq. (26) is the explicit form of (6) in the linear case.

We now give a formal derivation of the explicit equations for accomplishing the iterations indicated by (20). The various formal steps of the derivation can be justified under mild mathematical restrictions.[6]

If ρ is given by (8) and $\mathcal{P}$ by (13), it can be shown by induction that the optimal performance index may be written in the form

$$\begin{aligned} \mathcal{P}_N{}^0[x(t_0)] &= x'(t_0) P_N x(t_0) - 2x'(t_0) R_N x^d \\ &\quad + x^{d\prime} S_N x^d \qquad (N \ge 0) \end{aligned} \qquad (28)$$

P_N, R_N, S_N being $n \times n$, $n \times l$, and $l \times l$ matrices, respectively, and

$$P_0 = R_0 = S_0 = 0.$$

For simplicity, we now drop the arguments of $\Phi(\tau)$ and $\Delta(\tau)$. Using (20) and (26), we calculate the deriva-

[14] E. A. Coddington and N. Levinson, "Theory of Ordinary Differential Equations," McGraw-Hill Book Co., Inc., New York, N. Y., ch. 3; 1955.

tive of the scalar $\mathcal{P}_{N+1}[x(t_0)]$ with respect to the vector $m(t_0)$. This is a vector (with components $\partial\mathcal{P}_{N+1}/\partial m_i(t_0)$), which is given by:

$$\partial\mathcal{P}_{N+1}/\partial m(t_0) = -2[\Delta'(R_N + Q)x^d - \Delta'(P_N + Q)(\Phi x(t_0) + \Delta m(t_0))] \qquad (N \geq 0). \tag{29}$$

Now $\mathcal{P}_{N+1}[x(t_0)]$ is evidently a quadratic function of each of the incremental control variables $m_i(t_0)$. It follows that $\mathcal{P}_{N+1}$ has a single extremal value [which may be a minimum or a maximum, depending on the value of $x(t_0)$] at that value of $m(t_0)$ which makes the right-hand side of (29) zero. It can be shown[6] that the extremal value is a minimum for every $x(t_0)$. Hence, $m(t_0)$ is found by setting (29) equal to zero, which yields the following expressions for $m^0(t_0)$ and the matrices defining $\mathcal{P}_{N+1}$:

$$m^0(t_0) = -A_N x(t_0) + B_N x^d, \tag{30}$$

where

$$\left.\begin{aligned} A_N &= [\Delta'(P_N + Q)\Delta]^{-1}(P_N + Q)\Phi \\ B_N &= [\Delta'(P_N + Q)\Delta]^{-1}(R_N + Q) \end{aligned}\right\}. \tag{31}$$

With some further calculations, using (30) and (31) we find that:

$$P_{N+1} = \lambda(\Phi - \Delta A_N)'(P_N + Q)\Phi; \tag{32a}$$

$$R_{N+1} = \lambda(\Phi - \Delta A_N)'(R_N + Q); \tag{32b}$$

and

$$S_{N+1} = \lambda(S_N + Q - B_N'\Delta'(R_N + Q)). \tag{32c}$$

The iterations indicated by (32) can be readily performed on a digital computer. Note that S_N need not be computed if only the optimal control vectors are of interest. In the limit $N \to \infty$, all quantities in (32), except S_{N+1}, may be shown to converge under certain restrictions on F, D, and λ.[6]

We get by inspection of (30) the important result: *In the linear case, the optimal control variables are linear functions of the actual and desired states of the reactor.*

Since the control variables are linear functions of the state variables, it follows that under closed-loop control the reactor is a linear dynamic system. It can be shown[6] that the only possible type of limiting behavior in such systems as $t \to \infty$ is for the state $X(t)$ to converge to an equilibrium state X^*. Since dynamic optimization includes static optimization, it follows at once that: *In the linear case, the states of a dynamically optimized system tend asymptotically to the same equilibrium state X^{*0} which is obtained under static optimization.*

Example: As a numerical illustration of the results obtained by the use of dynamic programming in the linear case, let us consider the following hypothetical reactions:

$$\text{(i)} \quad A + B \xrightarrow{k_1(T)} C$$

$$\text{(ii)} \quad 2B + C \xrightarrow{k_2(T)} 2D.$$

The objective is to convert raw materials A and B by means of reaction (i) into C obtaining as much quantity of C as possible. The optimization of the process is complicated by the undesired side reaction (ii) which produces the contamination product D. Under steady-state conditions, the resulting concentrations in the outflow as a function of the "hold-up" time (reactor volume /outflow rate) will have the qualitative shape shown in Fig. 4.

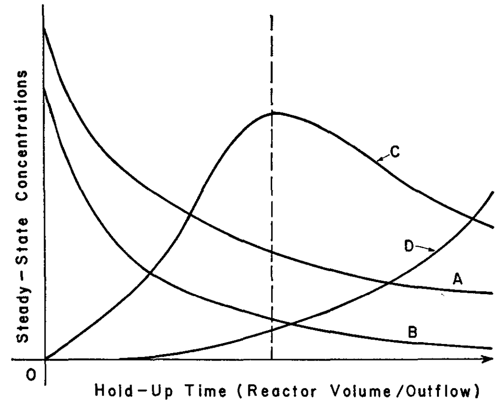

Fig. 4.

We now derive the analytical form of the dynamic equations of the reactor using the assumption that the Arrhenius equation (1) holds. Denoting the concentrations of $A, \cdots, D$ by $X_1, \cdots, X_4$, and the flow rates of A and B by M_1, M_2 we find, using conservation of mass, that:

$$\begin{aligned} dX_1/dt &= -k_1(T)X_1X_2 + (M_1/V)U_1 - [(M_1 + M_2)/V]X_1; \\ dX_2/dt &= -k_1(T)X_1X_2 - 2k_2(T)X_2^2X_3 + (M_2/V)U_2 - [(M_1 + M_2)/V]X_2; \\ dX_3/dt &= k_1(T)X_1X_2 - k_2(T)X_2^2X_3 - [(M_1 + M_2)/V]X_3; \end{aligned} \tag{33}$$

and

$$dX_4/dt = 2k_1(T)X_2^2X_3 - ([M_1 + M_2)/V]X_4.$$

Let T_1 and T_2 denote the temperatures of the input flows M_1, M_2; let T_c be the average cooling water temperature inside the cooling coils of the reactor; and let h be the corresponding average heat transfer coefficient per unit cooling water flow. Furthermore, let H_1 be the heat generated per molecule of the first reaction; H_2 the heat generated per molecule of the second reaction; ρ the average density of the material in the reactor; and c the average heat capacity of the material. Denoting the temperature in the reactor by X_5 and the cooling-water flow rate by M_5, conservation of energy yields

$$\begin{aligned} dX_5/dt = {} & k_1(T)X_1X_2H_1 + k_2(T)X_2^2X_3H_2 \\ & + (M_1/V\rho c)(T_1 - X_5) + (M_2/V\rho c)(T_2 - X_5) \\ & + (h/V\rho c)M_5(T_c - X_5). \end{aligned} \tag{34}$$

At equilibrium, the variables entering (33) and (34) are assumed to have the values shown in Table I. Note that the first reaction is assumed to be exothermic, and the second is assumed to be endothermic.

Using these values, (23) yields the following numerical values for the matrices describing the dynamics of the reactor in the vicinity of equilibrium:

$$F=\begin{bmatrix} -0.325 & -0.5625 & 0 & 0 & -0.200 \\ -0.225 & -0.8125 & -0.014286 & 0 & -0.368 \\ 0.225 & 0.4875 & -0.107143 & 0 & 0.116 \\ 0 & 0.1500 & 0.014286 & -0.1 & 0.168 \\ 0.450 & 0.7500 & -0.035714 & 0 & -0.060 \end{bmatrix}. \quad (35)$$

Assuming that only the flow rates M_1 and M_3 can be changed to effect control, we get:

$$D=\begin{bmatrix} 55 & 0 & 0 \\ -4 & 0 & 0 \\ -21 & 0 & 0 \\ -3 & 0 & 0 \\ -2 & 0 & 35.5 \end{bmatrix}. \quad (36)$$

Using a sampling period $\tau=1$, the transition matrix for the linear system can be obtained using the Taylor series (25).

TABLE I

VALUES OF REACTOR CONSTANTS

U_1	U_2	M_1^*	M_2^*	M_3^*	X_1^*	X_2^*	X_3^*	X_4^*	X_5^*
65	59	0.05	0.05	0.1	10	4	21	3	120

$k_1(X_5^*)$	$k_2(X_5^*)$	$\partial k_1(X_5^*)/\partial X_5$	$k_2(X_5^*)/\partial X_5$
0.05625	0.00044643	0.005	0.00025

T_1	T_2	T_c	H_1	H_2	V	$h/V\rho c$	ρc
100	100	49	2	−5	1	0.5	10

To improve the operation of the reactor, it is desirable to increase the yield of C and cut down the yield of D. Therefore, the desired state of the reactor may be defined as:

$$x_3{}^d = 5 \text{ .and } x_4{}^d = -1. \quad (41)$$

If the reactor starts out at the old equilibrium state ($x_1=x_2=\cdots=x_5=m_1=\cdots=m_3=0$) at time $t_0=0$, the behavior of the state and control variables as a function of time will be as shown in Fig. 5. It is evident from Fig. 5 that it is possible to achieve almost exactly the new desired state and that the new equilibrium state can be reached rather quickly. It should be noted, however, that the results are valid only if the linearized approximation of the dynamics is valid.

$$\Phi(\tau)=\begin{bmatrix} 0.7407 & -0.3581 & 0.00498 & 0 & -0.0909 \\ -0.1793 & 0.4095 & -0.00448 & 0 & -0.2197 \\ 0.1566 & 0.2895 & 0.89510 & 0 & 0.0264 \\ 0.0151 & 0.1373 & 0.00946 & 0.9048 & 0.1288 \\ 0.3004 & 0.3872 & -0.03460 & 0 & 0.8172 \end{bmatrix} \quad (37)$$

$$\Delta(\tau)=\begin{bmatrix} 48.216 & 0 & -2.169 \\ -7.695 & 0 & -4.682 \\ -15.744 & 0 & 0.9124 \\ -3.089 & 0 & 2.5132 \\ 7.056 & 0 & 32.822 \end{bmatrix}. \quad (38)$$

The peformance index is defined as:

$$\mathcal{P} = \sum_{k=1}^{\infty} [x_3{}^d - x_3(t_0+k)]^2 + [x_4{}^d - x_4(t_0+k)]^2. \quad (39)$$

The optimal control variables for this performance index are given by the following functions of the desired and actual state of the reactor:

IV. LIMITATIONS OF THE LINEARITY ASSUMPTION

There are a large number of problems which must be considered before fully automatic dynamic optimization of chemical reactions can take place.

1) If state variables are not physically measurable, they must be generated artificially in order to be able to compute the optimal values of the control variables.

$$m=\begin{bmatrix} 0 & 0 & -0.0680 & 0.0217 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0788 & 0.3932 & 0 \end{bmatrix} x^d + \begin{bmatrix} 0.0102 & 0.0164 & 0.0605 & -0.0197 & -0.0012 \\ 0 & 0 & 0 & 0 & 0 \\ 0.0064 & -0.0327 & 0.0668 & -0.3537 & -0.0504 \end{bmatrix} x. \quad (40)$$

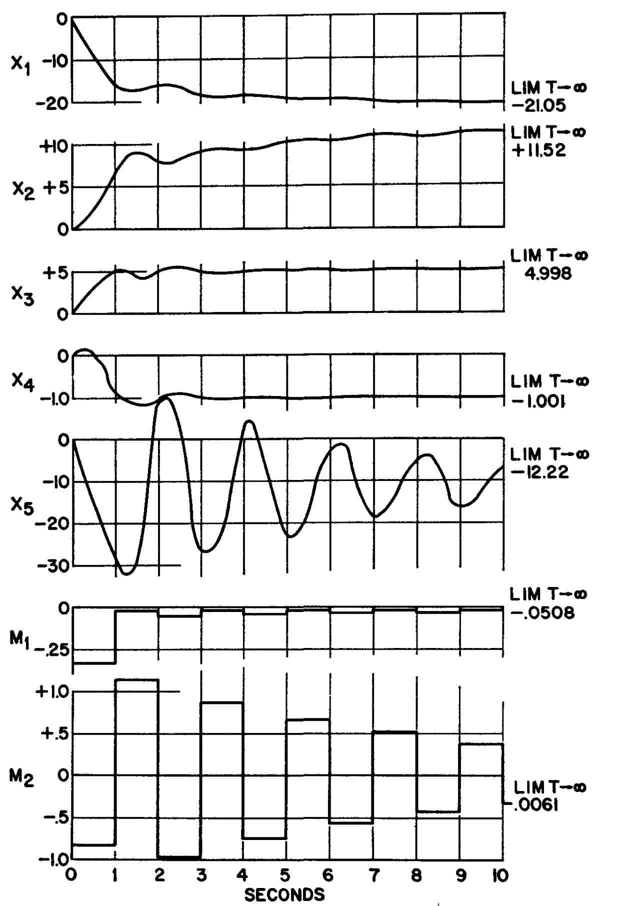

Fig. 5.

This calls for simulating some of the reaction dynamics as an integral part of the control system. This, too, can be done by means of a digital computer. (Most of the analog-type control instruments used at present may be thought of as performing essentially this function.)

2) The dynamic programming equations can be solved, practically speaking, only in the linear case. In reality, of course, the reaction dynamics are nonlinear. Moreover, they may change with time due to uncontrollable or unknown effects. There are essentially two possibilities of attacking these problems.

a) The reaction dynamics are linearized over a certain region in state space. The reaction is then optimized on a linear basis, computing the dynamic programming equations in real time. If, as a result of this optimization, the state moves into another region of the state space, another set of linearized equations is obtained to describe the dynamics in the new region. These equations are then used to obtain a new dynamic optimization, etc. This method of attack is closely related to the problem of designing adaptive or self-optimizing systems[15] about which little is known at present. The chief difficulty is the rapid and accurate determination of the linear dynamics in the presence of measurement noise.

b) The dynamic optimization is solved directly by purely numerical methods. The chief difficulty encountered here is the experimental measurement and representation of the reaction dynamics in a nonlinear form. Very little is known about this problem at present.

c) The control variables cannot be chosen freely but must lie within certain prescribed ranges; in other words, the control variables "saturate." The problem of designing a control system where the dynamic equations of the control object are linear but where the control variables saturate has an extensive literature usually under the subject heading of "Optimal Relay Servo Problem." At present this problem is solved only in the case where 1) the dynamic equations are of the second order and 2) there is only one control variable.[16] Using the point of view of this paper, a rigorous method was recently obtained (which is not subject to the above restrictions, 1 and 2)[8] for the computation of the optimal control variables; however, this method is very inefficient. When the control object has nonlinear dynamics, no method of computing the optimal control variables is known.

Despite these obstacles, much progress can be expected from the utilization of the "state" method of describing reaction dynamics combined with dynamic optimization as presented in this paper. These new ideas will probably be most helpful in attempting to control (by means of real-time digital computation) dynamic systems which have many state variables.

List of Principal Quantities

Sections II-A and II-B

U; U_i = vector denoting concentrations in input streams; its components.

M; M_i = control vector; control variables.

l = number of control variables.

T = temperature.

X; X_i = state vector; state variables (concentrations and temperature inside reactor).

n = number of state variables.

t; t_0 = time; initial time.

Section II-C

f; f_i = infinitesimal state transition function; its components.

τ = sampling period.

ϕ; ϕ_i = (finite-interval) state transition function; its components.

[15] R. E. Kalman, "Design of a self-optimizing system," *Trans. ASME*, vol. 80, pp. 468–478; 1958.

[16] R. E. Kalman, "Analysis and design principles of second and higher-order saturating servomechanisms," App. 2, *Trans. AIEE*, vol. 24, pt. 2, pp. 294–310; 1955.

Section II-D

ρ = distance function in state space (pseudometric).

$'$ = transpose of the matrix.

Q = positive semidefinite matrix.

$(\ )^*$ = equilibrium values.

$(\ )^0$ = optimal values.

Section II-E

$\mathcal{P}$ = performance index.

h; h_i = optimal control function.

Section III.

x; x_i = incremental state vector; incremental state variables.

m; m_i = incremental control vector; incremental control variables.

F = infinitesimal transition matrix in linear case.

D = matrix denoting instantaneous effect of control variables in linear case.

$\Phi(\tau)$ = (finite-interval) transition matrix in linear case.

$\Delta(\tau)$ = matrix denoting effect of control variables in linear case (finite-interval).

A_N, B_N, P_N, R_N, S_N constant matrices.

Simulation of Human Problem-Solving

W. G. BOURICIUS† AND J. M. KELLER†

SIMULATING human problem-solving on a digital computer looks deceptively simple. All one must do is program computers to solve problems in such a manner that the computer employs the identical strategies and tactics that humans do. This will probably prove to be as simple in theory and as hard in actual practice as was the development of reliable digital computers. One of the purposes of this paper is to describe a few of the pitfalls that seem to lie in the path of anyone trying to program machines to "think."

The first pitfall lies in the choice of an experimental problem. Naturally enough the problem chosen should be of the appropriate degree of difficulty, not so difficult that it cannot be done, and not so trivial that nothing is learned. It should also involve symbology and manipulations capable of being handled by digital computers. At this stage of problem consideration, a devious form of reasoning begins to operate. Usually the people engaged in this type of research will have had a thorough grounding in conventional problem-solving on computers. Consequently, they are conversant with the full range of capabilities of computers and have an appreciation of their great speed, reliability, etc. They also know what kinds of manipulations computers do well, and conversely, what kinds of things computers do in a clumsy fashion. All of this hard-earned knowledge and sophistication will tend to lead them astray when the time arrives to choose a problem. They will try to make use of this knowledge and hence choose a problem that will probably involve the simulation of humans solving problems with the aid of computers rather than the simulation of humans solving problems with only paper and pencil. Consequently, the characteristics of present-day computers may confine and constrict the area of research much more than is desirable or requisite. What is liable to happen, and what did happen to us, is that the experimental problem chosen will develop into one of large size and scope. If this always happens, then those human manipulative abilities that are presently clumsy and time-consuming on computers will never get programmed, simulated, or investigated. Fortunately for us, the two experimental problems we chose were of such a nature that they could be easily miniaturized, and this was done as soon as the desirability became apparent.

The second pitfall which must be avoided is the assumption that one knows in detail how one thinks. This delusion is brought about by the following happenstance. People customarily think at various levels of abstraction, and only rarely descend to the abstraction level of computer language. In fact, it seems that a large share of thinking is carried on by the equivalent of "subroutines" which normally operate on the subconscious level. It requires a good deal of introspection over a long period of time in order to dredge up these subroutines and simulate them. We believe people assume that they know the logical steps they pursue when solving problems, primarily because of the fact that when two humans communicate, they do not need to descend to the lower levels of abstraction in order to explain to each other in a perfectly satisfactory way how they themselves solved a particular problem. The fact that they are likely to have very similar "subroutines" is obvious and also very pertinent.

† IBM Res. Center, Yorktown Heights, N. Y.