LITERATURE CITED

1. Vliet, G. C., and G. Leppert, *J. Heat Transfer*, 163 (May, 1961).
2. Yuge, T., *ibid.*, **82**, 214 (1960).
3. Kramers, H., *Physica*, **12**, 61 (1946).
4. Briggs, D., W. Brown, M. Larson, P. McCuen, and J. Mitchell, Mech. Eng. 232 Lab., Stanford Univ., (1958), unpublished.
5. Williams, C. C., D.Sc. thesis, Mass. Inst. Tech., Cambridge, Mass., (1942).
6. Garner, F. H., and R. B. Keey, *Chem. Eng. Sci.*, **6**, 119 (1958).
7. Steele, L. R., and C. J. Geankoplis, *AIChE J.*, **5**, 178 (1959).
8. El-Wakil, M. M., G. E. Myers, and R. J. Schilling, *J. Heat Transfer*, 399 (Nov., 1966).
9. Ranz, W. E., and W. R. Marshall, Jr., *Chem. Eng. Prog.*, **48**, 141 (1952).
10. Friedlander, S. K., *AIChE J.*, **3**, 43 (1957).
11. *Ibid.*, **7**, 347 (1961).
12. Yuge, T., *Rept. Inst. High Speed Mechanics*, Tohoku Univ., **57**, 148, (1956).
13. Griffith, R. M., *Chem. Eng. Sci.*, **12**, 198 (1960).
14. Acrivos, Andreas, *AIChE J.*, **4**, 285 (1958).
15. Szewczyk, A. A., *J. Heat Transfer*, 501 (Nov., 1964).
16. Garner, F. H., and R. B. Keey, *Chem. Eng. Sci.*, **9**, 218 (1958).
17. Steinberger, R. L., and R. E. Treybal, *AIChE J.*, **6**, 227 (1960).
18. Ranz, W. E., and W. H. Marshall, Jr., *Chem. Eng. Progr.*, **48**, 173 (1952).
19. Bird, R. B., W. E. Stewart, and E. N. Lightfoot, "Transport Phenomena," p. 515, John Wiley, New York (1964).
20. Sherwood, T. K., and R. C. Reid, "The Properties of Gases and Liquids," p. 51, McGraw-Hill, Pennsylvania, (1958).
21. Garner, F. H., and J. M. Hoffman, *AIChE J.*, **6**, 579 (1960).

Manuscript received October 9, 1967; revision received January 9, 1968; paper accepted January 13, 1968.

Iterative Techniques in Optimization

I. Dynamic Programming and Quasilinearization

E. STANLEY LEE

Kansas State University, Manhattan, Kansas

The quasilinearization technique is used to overcome the dimensionality difficulties of dynamic programming. The approach is based on the fact that if the difference or differential equations are linear, their closed forms of solution can be obtained. This solution permits us to separate the effects due to the initial state from the effects due to the control variables. By using this separation combined with quasilinearization, the dimensionality of the functional equation of dynamic programming can be reduced to one in most cases. First, the optimization problem in cross-current extraction with discontinuous objection function is used to illustrate the technique. Then the technique is generalized to systems of difference and differential equations with fairly general objective functions.

Multistage optimization techniques can be divided into two general classes. The first class of techniques are the classical methods which include the calculus of variations and both the continuous and the discrete versions of the maximum principle. Various difficulties are encountered in solving optimization problems numerically by the classical methods. In the first place, considerable computational difficulties exist for the solution of the optimization problem if the number of the state variables are large. For continuous optimization problems these difficulties are known as the boundary-value difficulties, since they are caused by the numerical solution of large dimensional two-point boundary-value problems. A second difficulty in applying the classical methods arises from the inequality constraints which involve the state variables only. Inequality constraints involving the control variables, or involving both the control and the state variables, are fairly easy to handle numerically (*1*). However, this is not the case for inequality constraints involving the state variables only. Control variable is, in a sense, an independent variable and is not subject to differential or algebraic equation constraints. State variables are not independent. A trial and error or iterative procedure is generally used to treat inequality constraints on state variables. The third difficulty concerns the problem of true optimum. Both the calculus of variations and the maximum principle do not guarantee a true optimum. In fact, the Euler-Lagrange equations only give a stationary point. Whether this stationary point is a maximum, minimum, or just a saddle point, it must be determined from physical or other considerations. A fourth difficulty is the inability of the classical method to handle nonanalytic objective functions easily.

The second class of optimization technique is the dynamic programming technique. This technique can handle state variable inequality constraints. It also can obtain the true optimum if proper search techniques are used. Furthermore, it is most suited for treating problems with nonanalytic objective functions. However, severe difficulties exist when the number of state variables is moderately large. These difficulties are known as the dimensionality difficulties which are caused principally by the limited rapid-access memory of current computers. Although the dynamic programming technique does not have the other three difficulties, the dimensionality difficulties limit this technique to the optimization of problems with two, or to the maximum, three state variables.

Although various iterative techniques such as the functional gradient technique (*2* to *4*), the second variational method (*5, 6*), and the quasilinearization technique (*1, 7*) have been devised to overcome the boundary-value difficulties, little work has been done in using these iterative techniques to overcome the dimensionality difficulties in dynamic programming. In this paper, the quasilinearization technique, or the Newton-Raphson method for discrete systems, will be used to overcome these dimensionality difficulties. A set of recurrence relations are first obtained by the quasilinearization technique from the origi-

nal nonlinear differential or nonlinear difference equations. Since these recurrence relations are essentially linear differential or linear difference equations, the closed form of the solutions of these equations is known. By using certain properties of these solutions, the dimensionality of the functional equation of the dynamic programming technique can be reduced to one in most cases.

Since the functional equation of dynamic programming is most suited for the numerical solution of discrete systems, most of the discussions in this paper are concerned with discrete systems. The approach is introduced by considering the optimization problem in cross-current extraction with a discontinuous objective function. Using the general solution of a system of simultaneous first-order difference equations, the approach is generalized to systems of difference equations. The optimization of continuous systems is also discussed.

It should be mentioned that various techniques have been proposed to reduce the dimensionality difficulties. Some of these techniques are the use of Lagrange multipliers, the use of orthogonal polynomials, the use of successive approximations, and the state increment dynamic programming (*8, 17, 18*). Except the last technique which has been applied effectively to optimal control problems with continuous independent variables, (*18*) all the other techniques have proven to be satisfactory in only a few cases.

A SIMPLE OPTIMIZATION PROBLEM

To illustrate the technique, let us consider an optimization problem in cross-current extraction. A simpler version of this problem has been studied in a previous paper (*4*) by the functional gradient technique. The flow sheet for this process is shown in Figure 1. We wish to produce a certain product A, whose market or sale value depends upon the purity of the product. The undesirable impurity is C which is to be extracted by an extract solvent B. The concentrations of C and A in the original mixture and raffinate are represented by x_R and y_R, respectively. The corresponding components in the extract streams are represented by x_E and y_E. The extract stream has an inlet flow rate, w, of pure solvent B and an outlet rate, u. The raffinate flow rate is represented by q. The streams q, u, and w are expressed in total weights and all concentrations are in weight fractions.

Mass balances on the total material in each stream, the impurity x and the product y respectively, give

$$q(n-1) + w(n) = q(n) + u(n) \tag{1}$$

$$q(n-1)x_R(n-1) = q(n)x_R(n) + u(n)x_E(n) \tag{2}$$

$$q(n-1)y_R(n-1) = q(n)y_R(n) + u(n)y_E(n) \tag{3}$$

In order to make the problem meaningful, the cost of the extract solvent must be considered. One way to accomplish this is to introduce a Lagrange multiplier, λ, and consider the maximization of the quantity,

$$\phi = \xi[x_R(N)]q(N)y_R(N) - \lambda \sum_{n=1}^{N} w(n) \tag{4}$$

where λ can be considered as the cost of the extract solvent divided by the value of the product. The term $q(N)y_R(N)$ represents the total amount of the product A produced. The function $\xi[x_R(N)]$ represents the unit sale value of A. This sale value is a function of the impurity concentration $x_R(N)$. In most practical cases this function is not continuous in $x_R(N)$. A typical sale value curve of A is shown in Figure 2. We shall assume that the value of ξ as a discontinuous function of $x_R(N)$ is known and is given in tabulated form such as

$$\begin{aligned} &\xi[x_R(N)] = \xi_1, \quad 0 \leq x_R(N) \leq x^{(1)} \\ &\xi[x_R(N)] = \xi_2, \quad x^{(1)} \leq x_R(N) \leq x^{(2)} \\ &\qquad \cdots\cdots\cdots\cdots \\ &\xi[x_R(N)] = \xi_l, \quad x^{(l-1)} \leq x_R(N) \leq 1 \end{aligned} \tag{5}$$

Our problem is to find $w(n)$, $n = 1, 2, \ldots, N$ so that Equation (4) is maximized. There are six unknowns: q, u, x_R, x_E, y_R, and y_E. In addition to Equations (1) to (3), three more equations can be obtained from the liquid-liquid equilibrium data which are usually represented by triangular equilibrium diagram. These equilibrium data can be correlated into equations of the following form:

$$x_R = f_1(y_R) \tag{6}$$

$$x_E = f_2(y_R) \tag{7}$$

$$y_E = f_3(y_R) \tag{8}$$

Using these three equations, Equations (1) to (3) become

$$q(n-1) + w(n) = q(n) + u(n) \tag{9}$$

$$q(n-1)f_1[y_R(n-1)] = q(n)f_1[y_R(n)] + u(n)f_2[y_R(n)] \tag{10}$$

$$q(n-1)y_R(n-1) = q(n)y_R(n) + u(n)f_3[y_R(n)] \tag{11}$$

The unknowns q, u, and y_R are state variables and w is the control variable. Equation (4) becomes

$$\phi = \xi\{f_1[y_R(N)]\}q(N)y_R(N) - \lambda \sum_{n=1}^{N} w(n) \tag{12}$$

The initial conditions for the difference equations, Equations (9) to (11), are

$$q(0) = q^o$$

$$x_R(0) = x^o$$

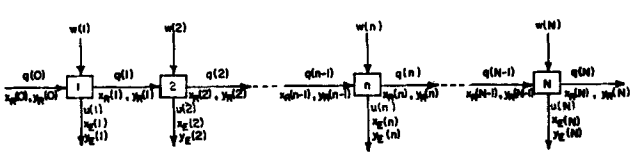

Fig. 1. Cross-current extraction.

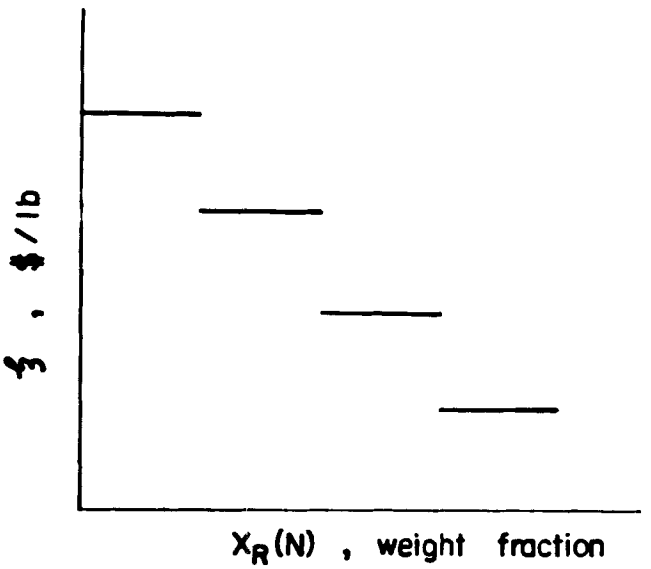

Fig. 2. A typical product value curve.

$$y_R(0) = y^o \tag{13}$$

With these given initial conditions and with a known control policy the three first-order nonlinear difference equations, Equations (9) to (11), can be solved by a trial and error or iterative procedure.

Our problem is to maximize Equation (12) subject to the constraints of Equations (9) to (11). Because of the presence of the jump discontinuity in the function ξ, both the functional gradient technique and the discrete maximum principle cannot be used conveniently. Although dynamic programming can handle optimization problems with nonanalytic objection functions, the presence of three state variables makes the dynamic programming approach not routine. To obtain a reasonably accurate solution a computer with a fairly large rapid-access memory must be used. We shall show that by the use of quasilinearization or the Newton-Raphson type linearization technique an optimization problem with only two state variables can be obtained. Furthermore, for a large number of problems the quasilinearization approach can reduce an optimization problem with m state variables to a problem with only one state variable.

RECURRENCE RELATIONS

Bellman and Kalaba (9) have shown that if the equations governing the transformation of the process are linear, a reduction in the dimensionality of the problem can be obtained. These authors treated continuous problems where the transformation is governed by differential equations. We shall apply the same idea to difference equations.

In order to obtain a reduction in the dimensionality of the problem, Equations (9) to (11) must be linearized. This can be achieved by the quasilinearization technique. Since the control variable w appears linearly in Equations (9) to (11), only the state variables need to be considered. Equations (9) to (11) can be linearized simultaneously by using the following vector equation

$$\begin{aligned}\mathbf{F}[\mathbf{v}_{k+1}(n), \mathbf{v}_{k+1}(n-1)] &= \mathbf{F}[\mathbf{v}_k(n), \mathbf{v}_k(n-1)] \\ &+ \mathbf{J}_{v(n)}(\mathbf{v}_k)[\mathbf{v}_{k+1}(n) - \mathbf{v}_k(n)] \\ &+ \mathbf{J}_{v(n-1)}(\mathbf{v}_k)[\mathbf{v}_{k+1}(n-1) - \mathbf{v}_k(n-1)]\end{aligned} \tag{14}$$

which is obtained from Taylor series with second and higher order terms neglected. $\mathbf{F}$, $\mathbf{v}_{k+1}$, and $\mathbf{v}_k$ are in vector form and represent the m-dimensional vectors with elements $F_1, F_2, \ldots, F_m$; $v_{1,k+1}, v_{2,k+1}, \ldots, v_{m,k+1}$; and $v_{1,k}, v_{2,k}, \ldots, v_{m,k}$ respectively. The Jacobian matrix, $\mathbf{J}_{v(n)}(\mathbf{v}_k)$, is defined by

$$\mathbf{J}_{v(n)}(\mathbf{v}_k) = \begin{bmatrix} \dfrac{\partial F_1}{\partial v_{1,k}(n)} & \dfrac{\partial F_1}{\partial v_{2,k}(n)} & \cdots & \dfrac{\partial F_1}{\partial v_{m,k}(n)} \\ \cdots & \cdots & \cdots & \cdots \\ \dfrac{\partial F_m}{\partial v_{1,k}(n)} & \dfrac{\partial F_m}{\partial v_{2,k}(n)} & \cdots & \dfrac{\partial F_m}{\partial v_{m,k}(n)} \end{bmatrix} \tag{15}$$

If we consider the values of the variables with the subscript k as known and as obtained from the previous kth iteration, the right-hand side of Equation (14) is always linear in the unknown variables with the subscript $(k+1)$.

Applying Equation (14) to Equations (9) to (11), a set of complex recurrence relations can be obtained. Symbolically, these relations can be written as

$$\begin{aligned} &p_1 q_{k+1}(n) + p_2 u_{k+1}(n) + p_3 y_{R(k+1)}(n) \\ &\quad + p_4 q_{k+1}(n-1) + p_5 y_{R(k+1)}(n-1) + p_6 = 0 \\ &p_7 q_{k+1}(n) + p_8 u_{k+1}(n) + p_9 y_{R(k+1)}(n) \\ &\quad + p_{10} q_{k+1}(n-1) + p_{11} y_{R(k+1)}(n-1) + p_2 = 0 \end{aligned} \tag{16}$$

with $n = 1, 2, \ldots, N$. The coefficients p are known functions of the known variables with the subscript k. Equation (16) represents three simultaneous linear first-order difference equations whose solution can be expressed in closed form by the use of well-known theorems concerning linear difference or differential equations (*10*). In vector-matrix notation, Equation (16) becomes

$$\mathbf{B}_1(n)\mathbf{v}_{k+1}(n) = \mathbf{B}_2(n)v_{k+1}(n-1) + \mathbf{p}_1(n)$$

or

$$\begin{aligned}\mathbf{v}_{k+1}(n) &= \mathbf{B}_1^{-1}(n)\mathbf{B}_2(n)\mathbf{v}_{k+1}(n-1) + \mathbf{B}_1^{-1}(n)\mathbf{p}_1(n) \\ &= \mathbf{A}(n)\mathbf{v}_{k+1}(n-1) + \mathbf{p}(n)\end{aligned} \tag{17}$$

where $\mathbf{B}^{-1}$ represents the inverse of $\mathbf{B}$ and $\mathbf{v}_{k+1}(n)$ is following three dimensional vector

$$\mathbf{v}_{k+1}(n) = \begin{bmatrix} q_{k+1}(n) \\ u_{k+1}(n) \\ y_{R(k+1)}(n) \end{bmatrix} \tag{18}$$

and $\mathbf{p}(n)$ is a three-dimensional vector whose components are functions of q_k, u_k, y_{Rk} and the control variable w_{k+1}. $\mathbf{A}(n)$ is a 3x3 matrix whose elements are functions of q_k, u_k, and y_{Rk}. Note that the control variable w_{k+1} appears only in $\mathbf{p}(n)$ and the matrix $\mathbf{A}(n)$ is independent of w_{k+1}.

The solution of Equation (17) can be obtained easily and can be represented by

$$\begin{aligned}\mathbf{v}_{k+1}(n) = &\left[\prod_{i=0}^{n=1} \mathbf{A}(n-i)\right] \mathbf{v}_{k+1}(0) + \\ &\left[\prod_{l=0}^{n-1} \mathbf{A}(n-l)\right] \sum_{i=0}^{n=1} \left[\left\{\prod_{j=i}^{n=1} \mathbf{A}(n-j)\right\}^{-1} \mathbf{p}(n-i)\right]\end{aligned} \tag{19}$$

where $\mathbf{v}_{k+1}(0)$ can be considered either as the given initial conditions or as the arbitrary constants which are to be chosen to fit the specified conditions. Equation (19) consists of two terms. The first term,

$$\left[\prod_{i=0}^{n-1} \mathbf{A}(n-i)\right] \mathbf{v}_{k+1}(0)$$

is the solution of the homogeneous equation

$$\mathbf{v}_{k+1}(n) = \mathbf{A}(n)\,\mathbf{v}_{k+1}(n-1) \tag{20}$$

The coefficient of the second term represents an impulse function or the Green's function in this discrete case. Equation (19) is similar to the solution of a system of first-order linear differential equations. Since $\mathbf{A}$ is a matrix, the order of multiplication must be preserved. The product Π is defined as

$$\prod_{i=0}^{n-1} \mathbf{A}(n-i) = \mathbf{A}(n)\,\mathbf{A}(n-1)\ldots\mathbf{A}(2)\mathbf{A}(1)$$

At $n = N$, Equation (19) becomes

$$\mathbf{v}_{k+1}(N) = \left[\prod_{i=0}^{N-1} \mathbf{A}(N-i)\right] \mathbf{v}_{k+1}(0) +$$

$$\left[\prod_{l=0}^{N-1}\mathbf{A}(N-l)\right]\sum_{i=0}^{N-1}\left[\left\{\prod_{j=i}^{N-1}\mathbf{A}(N-j)\right\}^{-1}\mathbf{p}(N-i)\right] \tag{21}$$

Since the matrix **A** is independent of the control variable, the first term is completely known and fixed once N and $\mathbf{v}_{k+1}(0)$ are given. Let

$$\mathbf{c}=\left[\prod_{i=0}^{N-1}\mathbf{A}(N-i)\right]\mathbf{v}_{k+1}(0) \tag{22}$$

and

$$\mathbf{K}(N-i)=\prod_{l=0}^{N-1}\mathbf{A}(N-l)\left\{\prod_{j=i}^{N-1}\mathbf{A}(N-j)\right\}^{-1} \tag{23}$$

Equation (21) becomes

$$\mathbf{v}_{k+1}(N)=\mathbf{c}+\sum_{i=0}^{N-1}\mathbf{K}(N-i)\mathbf{p}(N-i) \tag{24}$$

where **c** is a three-dimensional constant vector and $\mathbf{K}(N-i)$ is a 3x3 matrix. Equation (24) can be rewritten as

$$\begin{bmatrix}q_{k+1}(N)\\u_{k+1}(N)\\y_{R(k+1)}(N)\end{bmatrix}=\begin{bmatrix}c_1\\c_2\\c_3\end{bmatrix}+\sum_{i=0}^{N-1}\begin{bmatrix}k_{11}(N-i) & k_{12}(N-i) & k_{13}(N-i)\\k_{21}(N-i) & k_{22}(N-i) & k_{23}(N-i)\\k_{31}(N-i) & k_{32}(N-i) & k_{33}(N-i)\end{bmatrix}\begin{bmatrix}p_1(N-i)\\p_2(N-i)\\p_3(N-i)\end{bmatrix} \tag{25}$$

It should be emphasized that **c** and $\mathbf{K}(N-i)$ are functions of the variables q_k, u_k, and y_{Rk} which are known and fixed only when the results of the previous iteration have been obtained. The vector **p** is a function of q_k, u_k, y_{Rk}, and w_{k+1}.

THE FUNCTIONAL EQUATION OF DYNAMIC PROGRAMMING

We have frequently emphasized the fact that **c** is independent of the control variable w_{k+1}. Now we wish to use this fact to reduce the dimensionality of the functional equation of dynamic programming. After k iterations, Equation (12) can be written as

$$\phi=H[q_{k+1}(N),\ y_{R(k+1)}(N)]-\lambda\sum_{n=1}^{N}w_{k+1}(n) \tag{26}$$

Using the solution of the difference equations, Equation (25), the objective function becomes

$$\phi=H\left[c_1+\sum_{i=0}^{N-1}\left[\sum_{j=1}^{3}k_{1j}(N-i)p_j(N-i)\right],\ c_3+\sum_{i=0}^{N-1}\left[\sum_{j=i}^{3}k_{3j}(N-i)p_j(N-i)\right]\right]-\lambda\sum_{n=1}^{N}w_{k+1}(n) \tag{27}$$

If we consider λ as a known parameter, the maximum value of Equation (27) depends only on c_1, c_3, and N. Note that if the explicit solution, Equation (25), were not used, the maximum value of Equation (27) would depend on c_1, c_2, c_3, and N. We have reduced the number of variables from three to two. If the original problem had m state variables and the objective function is only functions of the final values of M state variables, a reduction in dimensionality from m to M would have been obtained by the above approach.

Define the following function of two variables

$$g_N(c_1,c_3)=\max_{\{w(n)\}}\left\{H\left[c_1+\sum_{i=0}^{N-1}\left[\sum_{j=1}^{3}k_{1j}(N-i)p_j(N-i)\right],\ c_3+\sum_{i=0}^{N-1}\left[\sum_{j=1}^{3}k_{3j}(N-i)p_j(N-i)\right]\right]-\lambda\sum_{n=1}^{N}w_{k+1}(n)\right\} \tag{28}$$

where $\{w(n)\}$ denotes the sequence $w(1)$, $w(2)$, ..., $w(N)$. Applying the principle of optimality, we obtain the desired recurrence relation

$$g_N(c_1,c_3)=\max_{w(1)}\left[g_{N-1}\left[c_1+\sum_{j=1}^{3}k_{1j}(1)p_j(1),\ c_3+\sum_{j=1}^{3}k_{3j}(1)p_j(1)\right]-\lambda w_{k+1}(1)\right],\quad N=2,3,\ldots,N \tag{29}$$

If the process only had one stage, we obtain

$$g_1(c_1,c_3)=\max_{w(1)}\left[H\left[c_1+\sum_{j=1}^{3}k_{1j}(1)p_j(1),\ c_3+\sum_{j=1}^{3}k_{3j}(1)p_j(1)\right]-\lambda w_{k+1}(1)\right] \tag{30}$$

The optimal control sequence, $w(n)$, can be obtained by solving Equations (29) and (30) recursively. It should be remembered that the function $p_j(1)$ is a function of the control variable $w_{k+1}(1)$.

The computational procedure can now be summarized as follows:

1. Estimate a reasonable control sequence $w_{k=0}(n)$, $n=1,2,\ldots,N$.
2. Calculate $q_{k=0}(n)$, $u_{k=0}(n)$, and $y_{R(k=0)}(n)$ from Equations (9) to (11), using the initial conditions given by Equation (13) and the newly obtained values of $w_{k=0}(n)$.
3. Obtain $w_{k=1}(n)$ by maximizing ϕ, using the newly obtained values of $q_{k=0}(n)$, $u_{k=0}(n)$ and $y_{R(k=0)}(n)$. The maximum value of ϕ can be obtained from the recurrence relation of dynamic programming, Equations (29) and (30).
4. Calculate $q_{k=1}(n)$, $u_{k=1}(n)$, and $y_{R(k=1)}(n)$ from Equations (9) to (11), using the initial conditions given by Equation (13) and the newly obtained values of $w_{k=1}(n)$.
5. Return to step three with $k=k+1$.
6. Repeat steps three to five until the required accuracy is obtained.

SYSTEMS OF DIFFERENCE EQUATIONS

The approach discussed above can be generalized easily to more general problems. Consider the maximization of

the following nonlinear objective function

$$H[v_1(N),\ v_2(N),\ \ldots,\ v_M(N)] \tag{31}$$

over the control variables $\mathbf{w}(n)$, which are related to the state variables v by means of the nonlinear difference equations

$$v_i(n) = f_i[\mathbf{v}(n-1), \mathbf{w}(n)] \quad i = 1, 2, \ldots, m \tag{32}$$

with initial conditions

$$v_i(0) = v_i^0, \quad i = 1, 2, \ldots, m \tag{33}$$

where $M < m$ and v and w are m-dimensional vectors. In addition, the problem must satisfy the constraints

$$w_{i,\,\min} \leq w_i(n) \leq w_{i,\,\max}, \quad i = 1, 2, \ldots, m \tag{34}$$

$$\sum_{n-1}^{N} f[\mathbf{w}(n)] \leq L \tag{35}$$

where the function f_i is essentially the transformation function and L is a given constant. Introducing the Lagrange multiplier, λ, the problem becomes the maximization of

$$\phi = H[v_1(N), v_2(N), \ldots, v_M(N)] - \lambda \sum_{n=1}^{N} f[\mathbf{w}(n)] \tag{36}$$

The problem now becomes the maximization of Equation (36) subject to the constraints of Equations (32) to (34).

If the functional equation of dynamic programming are used to maximize Equation (36), this maximization problem involves the computation and storage of functions of m variables. However, when the linearization scheme discussed in the previous sections is used, this problem could be treated by dynamic programming involving sequences of functions of M variables only. Since M equals to one or two for a number of problems, this is a very significant reduction in the dimensionality of the problem.

Applying the Newton-Raphson formula, Equation (32) can be linearized into the following form

$$v_{i,k+1}(n) = f_i[\mathbf{v}_k(n-1), \mathbf{w}_k(n)] + \sum_{j=1}^{m} [v_{j,k+1}(n-1) - v_{j,k}(n-1)] \frac{\partial f_i}{\partial v_j} + \sum_{j=1}^{m} [w_{j,k+1}(n) - w_{j,k}(w)] \frac{\partial f_i}{\partial w_j}, \quad i = 1, 2, \ldots, m \tag{37}$$

In vector-matrix notation, Equation (37) becomes

$$\mathbf{v}_{k+1}(n) = \mathbf{f}[\mathbf{v}_k(n-1), \mathbf{w}_k(n)] + \mathbf{J}_v(\mathbf{v}_k, \mathbf{w}_k)[\mathbf{v}_{k+1}(n-1) - \mathbf{v}_k(n-1)] + \mathbf{J}_w(\mathbf{v}_k, \mathbf{w}_k)[\mathbf{w}_{k+1}(n) - \mathbf{w}_k(n)] \tag{38}$$

If we consider the variables with the subscript k as known and as obtained from the previous iteration, Equation (38) represents a set of linear equations. Let

$$\mathbf{A}(n) = \mathbf{J}_v(\mathbf{v}_k, \mathbf{w}_k) \tag{39}$$

$$\mathbf{p}(n) = \mathbf{f}[\mathbf{v}_k(n-1), \mathbf{w}_k(n)] - \mathbf{J}_v(\mathbf{v}_k, \mathbf{w}_k)\mathbf{v}_k(n-1) + \mathbf{J}_z(\mathbf{v}_k, \mathbf{w}_k)[\mathbf{w}_{k+1}(n) - \mathbf{w}_k(n)] \tag{40}$$

then Equation (38) becomes

$$\mathbf{v}_{k+1}(n) = \mathbf{A}(n)\mathbf{v}_{k+1}(n-1) + \mathbf{p}(n) \tag{41}$$

where $\mathbf{A}(n)$ is an mxm matrix and $\mathbf{p}(n)$ is an m-dimensional vector. The solution of Equation (41) can again be represented by Equation (19). At $n = N$, this solution becomes

$$\mathbf{v}_{k+1}(N) = \left[\prod_{s=0}^{N-1} \mathbf{A}(N-s)\right]\mathbf{v}^0 + \left[\prod_{l=0}^{N-1} \mathbf{A}(N-l)\right] \sum_{s=0}^{N-1} \left[\left\{\prod_{j=s}^{N-1} \mathbf{A}(N-j)\right\}^{-1} \mathbf{p}(N-s)\right] \tag{42}$$

where the given initial condition, Equation (33), has been used. Instead of i, s has been used as the index. By using equations similar to Equations (22) and (23), Equation (42) can be simplified to

$$\mathbf{v}_{k+1}(N) = \mathbf{c} + \sum_{s=0}^{N-1} \mathbf{K}(N-s)\mathbf{p}(N-s) \tag{43}$$

where $\mathbf{c}$ is an m-dimensional vector whose elements are independent of the control variable w and $\mathbf{K}$ is an mxm matrix. Equation (43) can be rewritten as

$$v_{i,k+1}(N) = c_i + \sum_{s=0}^{N-1} \left[\sum_{j=1}^{m} k_{ij}(N-s)p_j(N-s)\right], \quad i = 1, 2, \ldots, m \tag{44}$$

Since $\mathbf{c}$ is independent of w and only p is a function of $w_{k+1}(n)$, Equation (44) can be used to reduce the dimensionality of the functional equation for this problem. After k iterations, Equation (36) can be written as

$$\phi = H[v_{1,k+1}(N),\ v_{2,k+1}(N),\ \ldots,\ v_{M,k+1}(N)] - \lambda \sum_{n-1}^{N} f[\mathbf{w}_{k+1}(n)] \tag{45}$$

Using Equation (44), the above equation becomes

$$\phi = H\left\{c_1 + \sum_{s=0}^{N-1} \left[\sum_{j=1}^{m} k_{1j}(N-s)p_j(N-s), \ldots, c_M + \sum_{s=0}^{N-1} \left[\sum_{j=1}^{m} k_{Mj}(N-s)p_j(N-s)\right]\right\} - \lambda \sum_{n=1}^{N} f[\mathbf{w}_{k+1}(n)] \tag{46}$$

Define

$$g_N(c_1, c_2, \ldots, c_M) = \max_{\{\mathbf{w}(n)\}} \phi \tag{47}$$

Applying the principle of optimality, we have

$$g_N(c_1, c_2, \ldots, c_M) = \max_{\mathbf{w}(1)} \left\{ g_{N-1}\left[c_1 + \sum_{j=1}^{m} k_{1j}(1)p_j(1), \ldots, c_M + \sum_{j=1}^{m} k_{Mj}(1)p_j(1)\right] - \lambda f[\mathbf{w}_{k+1}(1)]\right\}, \quad N = 2, 3, \ldots, N \tag{48}$$

For a one-stage process, we have

$$g_1(c_1, c_2, \ldots, c_M) = \max_{w(1)} \left\{ H \left[c_1 + \sum_{j=1}^{m} k_{1j}(1) p_j(1), \ldots, c_M + \sum_{j=1}^{m} k_{Mj}(1) p_j(1) \right] - \lambda f[\mathbf{w}_{k+1}(1)] \right\} \quad (49)$$

We have reduced the dimensionality of the problem from m to M. Note that the functions $\mathbf{c}$, $\mathbf{K}$, and $\mathbf{p}$ are different for different iterations and are functions of the iterations. However, for simplicity, we have omitted the subscript k or $k+1$ from these variables. The computational procedure for this problem is essentially the same as that outlined in the previous section.

SEPARATE NONLINEAR CONTROL FUNCTION

Instead of Equation (32), let us consider the following simpler nonlinear difference equations

$$v_i(n) = F_i[\mathbf{v}(n-1)] + G_i[\mathbf{w}(n)], \quad i = 1, 2, \ldots, m$$

With Equations (31) and (33) to (35) unchanged. Since the control variables $\mathbf{w}(n)$ do not appear in the function, F_i, only the state variables v need to be considered for linearization. The linearized difference equations are

$$\mathbf{v}_{k+1}(n) = \mathbf{G}[\mathbf{w}_{k+1}(n)] + \mathbf{F}[\mathbf{v}_k(n-1)] + \mathbf{J}_v(\mathbf{v}_k)[\mathbf{v}_{k+1}(n-1) - \mathbf{v}_k(n-1)]$$

The coefficients $\mathbf{A}$ and $\mathbf{p}$ in Equation (41) become

$$\mathbf{A}(n) = \mathbf{J}_v(\mathbf{v}_k)$$

$$\mathbf{p}(n) = \mathbf{G}[\mathbf{w}_{k+1}(n)] + \mathbf{F}[\mathbf{v}_k(n-1)] - \mathbf{J}_v(\mathbf{v}_k)\mathbf{v}_k(n-1)$$

The other equations in the previous section remain unchanged.

FURTHER GENERALIZATIONS

The above approach can be generalized easily to problems with other forms of objective functions. For example, the maximization of the following objective function

$$\sum_{n=1}^{N} H[\mathbf{v}(n-1), \mathbf{w}(n)] \quad (50)$$

also can be treated by the present approach. Let us introduce an extra state variable, v_{m+1}, defined as

$$v_{m+1}(n) = \sum_{n=1}^{n} H[\mathbf{v}(n-1), \mathbf{w}(n)], \quad n = 1, 2, \ldots, N \quad (51)$$

with initial condition

$$v_{m+1}(0) = 0 \quad (52)$$

The transformation equation for this state variable is

$$v_{m+1}(n) = v_{m+1}(n-1) + H[\mathbf{v}(n-1), \mathbf{w}(n)], \quad n = 1, 2, \ldots, N \quad (53)$$

Our problem now becomes the maximization of $v_{m+1}(N)$ subject to the constraints of the original equations and Equations (52) and (53). The number of state variables in this problem becomes $m+1$. However, since the objective function is a function of the final value of only one variable, a functional equation with a dimensionality of one can be obtained by using the present approach.

A variety of other forms of objective functions also can be treated by the algorithms obtained in the previous sections. More discussions concerning the transformation of other forms of objective functions into the standard form, Equation (31), can be found in the literature (*11, 12*).

Instead of Equation (32), transformation equations of the more general form

$$f_i[\mathbf{v}(n), \mathbf{v}(n-1), \mathbf{w}(n)] = 0, \quad i = 1, 2, \ldots, m \quad (54)$$

can also be treated. These equations can be linearized into the following form

$$f_i[\mathbf{v}_k(n), \mathbf{v}_k(n-1), \mathbf{w}_k(n)] + \sum_{j=1}^{m} [v_{j,k+1}(n) - v_{j,k}(n)] \frac{\partial f_i}{\partial v_j(n)} + \sum_{j=1}^{m} [v_{j,k+1}(n-1) - v_{j,k}(n-1)] \frac{\partial f_i}{\partial v_j(n-1)} + \sum_{j=1}^{m} [w_{j,k+1}(n) - w_{j,k}(n)] \frac{\partial f_i}{\partial w_j(n)} = 0, \quad i = 1, 2, \ldots, m \quad (55)$$

which can be reduced to the form of Equation (41) by obtaining the inverse of the matrix which is the coefficient of the vector $\mathbf{v}_{k+1}(n)$.

For simplicity in notations, the number of control variables has been assumed to be equal to the number of state variables. Obviously, the procedure discussed in the previous sections can also be applied if this is not the case.

FURTHER REDUCTION IN DIMENSIONALITY

Owing to the limited rapid-access memory of current computers, the above algorithms cannot be used if M is larger than three. However, for a large number of problems, a further reduction of the dimensionality can be obtained. Consider the problem of maximizing the function

$$H_1[v_1(N), v_2(N), \ldots, v_m(N)] \quad (56)$$

which is a function of the final values of all the state variables. No reduction in dimensionality can be obtained if the approaches discussed in the previous sections are used. However, if we introduce an extra state variable, v_{m+1}, defined by

$$v_{m+1}(n) = H_1[\mathbf{v}(n)], \quad n = 1, 2, \ldots, N \quad (57)$$

Substituting Equation (32) into the above equation, we have

$$v_{m+1}(n) = H_1\{\mathbf{f}[v(n-1), \mathbf{w}(n)]\}, \quad n = 1, 2, \ldots, N \quad (58)$$

with initial condition

$$v_{m+1}(0) = H_1[\mathbf{v}(0)] = H_1(\mathbf{v}^0) \quad (59)$$

Thus, our enlarged system is composed of Equations (32) to (35), (58), and (59). The problem becomes the maximization of the final value of the single state variable $v_{m+1}(N)$. A functional equation with a dimensionality of one can be obtained if the approaches discussed in the previous sections are used. Note that in order to use the present approach, the function H_1 must be differentiable.

CONTINUOUS SYSTEMS

Using the general form of solution for linear differential equations, the approach can be easily extended to continu-

ous systems. In fact, most of the original work done by Bellman and his coworkers (8, 9) is for continuous systems.

Consider the problem of maximizing the function

$$H[x_1(t_f),\ x_2(t_f),\ \ldots,\ x_M(t_f)] \tag{60}$$

over the control variables $\mathbf{z}(t)$, which are related to the state variables $\mathbf{x}$ by means of the nonlinear differential equations

$$\frac{dx_i}{dt} = f_i(\mathbf{x}, \mathbf{z}) \quad i = 1, 2, \ldots, m \tag{61}$$

with initial conditions

$$x_i(0) = x_i^0 \qquad i = 1, 2, \ldots, m \tag{62}$$

where $M < m$, $0 \leq t \leq t_f$, and $\mathbf{x}$ and $\mathbf{z}$ are m-dimensional vectors. In addition, the problem must satisfy the constraints

$$z_{i,\,\min} \leq z_i(t) \leq z_{i,\,\max} \quad i = 1, 2, \ldots, m \tag{63}$$

$$\int_0^{t_f} f(z)dt \leq L \tag{64}$$

If the functional equation of dynamic programming is used, this problem involves the computation and storage of functions of m variables. However, if Equation (61) were linear, the above problem could be treated by dynamic programming involving sequences of functions of M variables. Thus, an iterative scheme can be formed by the combined use of dynamic programming and quasilinearization.

When applying the generalized Newton-Raphson formula, Equation (61) can be linearized into the following form:

$$\frac{dx_{i,k+1}}{dt} = f_i(\mathbf{x}_k,\mathbf{z}_k) + \sum_{j=1}^{m} (x_{j,k+1} - x_{j,k}) \frac{\partial f_i}{\partial x_j}(\mathbf{x}_k, \mathbf{z}_k) + \sum_{j=1}^{m} (z_{j,k+1} - z_{j,k}) \frac{\partial f_i}{\partial z_j}(\mathbf{x}_k, \mathbf{z}_k) \quad i = 1, 2, \ldots, m \tag{65}$$

In vector-matrix notation, Equation (65) becomes

$$\frac{d\mathbf{x}_{k+1}}{dt} = \mathbf{f}(\mathbf{x}_k, \mathbf{z}_k) + \mathbf{J}_x(\mathbf{x}_k, \mathbf{z}_k)(\mathbf{x}_{k+1} - \mathbf{x}_k) + \mathbf{J}_z(\mathbf{x}_k, \mathbf{z}_k)(\mathbf{z}_{n+1} - \mathbf{z}_k) \tag{66}$$

Let

$$\mathbf{A}(t) = \mathbf{J}_x(\mathbf{x}_k, \mathbf{z}_k) \tag{67}$$

$$\mathbf{p}(t) = \mathbf{f}(\mathbf{x}_k, \mathbf{z}_k) - \mathbf{J}_x(\mathbf{x}_k, \mathbf{z}_k)\mathbf{x}_k + \mathbf{J}_z(\mathbf{x}_k, \mathbf{z}_k)(\mathbf{z}_{k+1} - \mathbf{z}_k) \tag{68}$$

then Equation (66) becomes

$$\frac{d\mathbf{x}_{k+1}}{dt} = \mathbf{A}(t)\mathbf{x}_{k+1} + \mathbf{p}(t) \tag{69}$$

with initial conditions

$$\mathbf{x}_{k+1}(0) = \mathbf{x}^0 \tag{70}$$

Equation (69) is a system of first-order linear differential equations with variable coefficients. Although the complete solution of this equation cannot be obtained analytically, the general form of the solution of this linear equation is well-known and can be expressed as follows (13):

$$\mathbf{x}_{k+1}(t) = \mathbf{X}(t)\mathbf{x}^0 + \int_0^t \mathbf{X}(t)\mathbf{X}^{-1}(s)\mathbf{p}(s)ds \tag{71}$$

where $\mathbf{X}(t)$ is an mxm matrix and is the solution of the matrix equation

$$\frac{d\mathbf{X}}{dt} = \mathbf{A}(t)\mathbf{X} \tag{72}$$

with initial conditions

$$\mathbf{X}(0) = \mathbf{I}$$

where $\mathbf{I}$ is a unit matrix. At $t = t_f$, Equation (71) becomes

$$\mathbf{x}_{k+1}(t_f) = \mathbf{c} + \int_0^{t_f} \mathbf{K}(s)\mathbf{p}(s)ds \tag{73}$$

where

$$\mathbf{c} = \mathbf{X}(t_f)\mathbf{X}^0$$
$$\mathbf{K}(s) = \mathbf{X}(t_f)\mathbf{X}^{-1}(s) \tag{74}$$

and $\mathbf{c}$ is an m-dimensional constant vector and $\mathbf{K}(s)$ is an mxm matrix. Equation (73) can be rewritten as

$$x_{i,k+1}(t_f) = c_i + \int_0^{t_f} \left[\sum_{j=1}^{m} k_{ij}(s)p_j(s) \right] ds \qquad i = 1, 2, \ldots, m \tag{75}$$

Introducing the Lagrange multiplier λ, the problem becomes the maximization of

$$\phi = H[x_{1,k+1}(t_f), x_{2,k+1}(t_f), \ldots, x_{M,k+1}(t_f)] - \lambda \int_0^{t_f} f(\mathbf{z}_{k+1})dt \tag{76}$$

over all $\mathbf{z}_{k+1}(t)$. By using Equation (75), the above equation becomes

$$\phi = H\left[c_1 + \int_0^{t_f} \left[\sum_{j=1}^{m} k_{1j}(s)p_j(s) \right] ds, \ldots, c_M + \int_0^{t_f} \left[\sum_{j=1}^{m} k_{Mj}(s)p_j(s) \right] ds \right] - \lambda \int_0^{t_f} f[\mathbf{z}_{k+1}(t)]dt \tag{77}$$

where $\mathbf{z}_{k+1}(t)$ represents the unknown control variables after k iterations. Note that c depends on the homogeneous solution of Equation (69) only and is independent of the nonhomogeneous term $\mathbf{p}(t)$. Since the homogeneous solution $\mathbf{X}(t)$ is independent of the control variable $\mathbf{z}_{k+1}(t)$, the value of $\mathbf{c}$ is independent of the control variable $\mathbf{z}_{k+1}(t)$ and is fixed once $\mathbf{X}(t)$ and $\mathbf{x}^0$ are known. Thus, in establishing the functional equation of dynamic programming only $c_1, c_2, \ldots, c_M$, which appear in Equation (77), need to be considered. Define

$$g(c_1, c_2, \ldots, c_M, a) = \max_{\mathbf{z}_{k+1}[a, t_f]} \left\{ H\left[c_1 + \int_a^{t_f} \left[\sum_{j=1}^{n} k_{1j}(s)p_j(s) \right] ds, \ldots, c_M + \int_a^{t_f} \left[\sum_{j=1}^{m} k_{Mj}(s)p_j(s) \right] ds \right] - \lambda \int_a^{t_f} f[\mathbf{z}_{k+1}(t)]dt \right\} \tag{78}$$

where the maximization is executed by choosing the proper values of $\mathbf{z}$ over the interval (a, t_f).

Since the process is nonstationary, we have fixed the final time, t_f, a family of processes with different starting points will be considered. Essentially we regard a as a variable in the interval $(0, t_f)$ and $\mathbf{c}$ as the independent variables constituting the new state which replaces the original state $\mathbf{x}^0$. Applying the principle of optimality, we obtain the desired functional equation

$$g(c_1, c_2, \ldots, c_M, a) = \max_{z_{k+1}[a, a+\Delta]} \left\{ g \left[c_1 + \int_a^{a+\Delta} \left(\sum_{j=1}^{m} k_{1j}(s) p_j(s) \right) ds, \ldots, c_M \right. \right.$$
$$\left. \left. + \int_a^{a+\Delta} \left(\sum_{j=1}^{m} k_{Mj}(s) p_j(s) \right) ds,\ a + \Delta \right] - \lambda \int_a^{a+\Delta} f[z_{k+1}(s)]ds \right\} \quad (79)$$

The terms under the integral sign may be approximated by

$$\int_a^{a+\Delta} f[z_{k+1}(s)]ds = f[z_{k+1}(a)]\Delta \quad (80)$$

$$\int_a^{a+\Delta} \left(\sum_{j=1}^{m} k_{ij}(s) p_j(s) \right) ds$$
$$= \sum_{j=1}^{m} k_{ij}(a) p_j(a)\Delta, \quad i = 1, 2, \ldots, M \quad (81)$$

Equation (79) becomes

$$g(c_1, c_2, \ldots, c_M, a)$$
$$= \max_{z_{k+1}(a)} \left[g \left[c_1 + \sum_{j=1}^{m} k_{1j}(a) p_j(a)\Delta, \ldots, c_M \right. \right.$$
$$\left. \left. + \sum_{j=1}^{m} k_{Mj}(a) p_j(a)\Delta,\ a + \Delta \right] - \lambda f[z_{k+1}](a)]\Delta \right] \quad (82)$$

To obtain the final condition for Equation (82), observe that if the process had zero duration at $a = t_f$, then the maximum value of Equation (77) would be equal to zero. Thus

$$g(c_1, c_2, \ldots, C_M, t_f) = 0 \quad (83)$$

Note that we have divided the duration of the process t_f into small intervals of Δ width. Let $t_f = N\Delta$, then $a = 0, \Delta, 2\Delta, \ldots, N\Delta$. Thus, Equation (82) can be solved in a backward recursive fashion starting with the known final condition, Equation (83), at $a + \Delta = t_f$.

We have reduced the dimensionality of the problem from m to M. Again it should be noted that the functions, $\mathbf{c}$, $\mathbf{K}(s)$, and $\mathbf{p}(s)$ are different for different iterations. The computational procedure for the continuous system is essentially the same as the one outlined before. However, much more computation are required as compared to the discrete systems. In order to use the functional equation, Equation (82), the value of $\mathbf{X}(t)$ must be obtained first. Thus, we must solve the homogeneous matrix equation, Equation (72), numerically, using $\mathbf{X}(0) = I$ as its initial condition.

The above procedure can be generalized easily to problems with more general objective function. For example, the problem of maximizing the integral

$$\int_0^{t_f} f(\mathbf{x}, \mathbf{z})dt \quad (84)$$

can be treated by the above procedure if we introduce an extra state variable, $x_{m+1}(t)$, defined by

$$\frac{dx_{m+1}}{dt} = f(\mathbf{x}, \mathbf{z}) \quad (85)$$

with initial condition

$$x_{m+1}(0) = 0 \quad (86)$$

The problem now becomes the maximization of $x_{m+1}(t_f)$. A variety of other forms of objective functions also can be treated by the algorithms obtained in the previous section. More discussion can be found in the literature (*12, 14, 15*).

Following the approach used for discrete systems, a further reduction of the dimensionality to one only also can be obtained for continuous systems. Consider the problem of maximizing the function

$$H_1[x_1(t_f), x_2(t_f), \ldots, x_m(t_f)] \quad (87)$$

Let us introduce a new state variable, $x_{m+1}(t)$, defined by

$$x_{m+1}(t) = H_1[\mathbf{x}(t)] \quad 0 \leq t \leq t_f \quad (88)$$

Differentiating Equation (88) with respect to t, we have

$$\frac{dx_{m+1}}{dt} = \sum_{j=1}^{m} \frac{\partial H_1[x(t)]}{\partial x_i} \frac{dx_i}{dt}$$
$$= \sum_{i=1}^{m} \frac{\partial H_1[x(t)]}{\partial x_i} f_i[\mathbf{x}(t), \mathbf{z}(t)] \quad (89)$$

The initial condition is

$$x_{m+1}(0) = H_1[\mathbf{x}(0)] \quad (90)$$

If we consider Equations (61) and (89) as the system of differential equations, the objective function, Equation (60), becomes a function of one variable

$$H = x_{m+1}(t_f)$$

If the algorithms obtained in the previous section are used, a problem with a dimensionality of one is obtained. This is a significant reduction in terms of computational requirements.

DISCUSSION

The use of the quasilinearization technique, which is essentially an iterative approximation scheme combined with linearization, appears to be a powerful tool for overcoming the dimensionality difficulty in dynamic programming. This is especially true for discrete systems which are ideally suited for obtaining dynamic programming solutions.

The approach also appears attractive for continuous systems with inequality constraints. However, much more computing time is needed for continuous system than for discrete systems. To obtain a reasonably accurate result, the value of Δ used in Equation (82) must be reasonably small. This undoubtedly will increase the computing time. Secondly, the homogeneous equation, Equation (72), must be solved numerically during each iteration in order to obtain the values of $\mathbf{c}$. On the other hand, the values of $\mathbf{c}$ for the discrete system can be calculated fairly easily.

If there were no inequality constraints, a frequently used approach for continuous systems is to reduce Equation (82) into a first-order partial differential equation which generally can be solved by the Cauchy's method of characteristics. The dimensionality difficulty is not very severe when Cauchy's method is used. Thus, the iterative approach discussed in this work probably require more computation than the one using Cauchy's method. However, Cauchy's method cannot be used easily when the problem has inequality constraints.

The present approach can treat inequality constraints involving control variables easily. However, owing to the introduction of a new state $\mathbf{c}$, it cannot easily treat inequal-

ity constraints involving state variables. This is due to the fact that the original state $\mathbf{x}^0$ does not appear in the recursive functional equation of dynamic programming directly. The treatment of problems with state variable inequality constraints will be discussed in a later paper.

The usefulness of any iterative technique depends upon whether or not the procedure will converge to the correct solution and how fast is the convergence rate. The first question cannot be answered in general terms. However, actual experience has shown that the procedure will converge for a large number of chemical engineering problems. Furthermore, convergence can be obtained even with very approximate initial approximations for a number of problems. The advantage of the quasilinearization technique lies in the convergence rate, which has been shown to be quadratical if there is any convergence at all. Computational experience (7) has indicated that generally a five digit accuracy can be obtained in three or to the maximum of seven iterations. In other words, the recursive functional equation of dynamic programming, such as Equation (48), only needs to be solved three to seven times. It also has been found that the number of iterations required is independent of the number of the state variables m (*1, 7*).

The main disadvantage of the present approach is that the transformation or state variable equations such as Equation (61) must be differentiable. Furthermore, in order to obtain the quadratic convergence property, the partial derivatives in the Taylor series expansion must be obtained accurately. Since numerical differentiation is a very inaccurate process, the derivatives must be obtained analytically. If the function f_i in Equation (61) is fairly complex, the analytical differentiation process can be very time consuming and prone to error. Wengert (*16*) has devised a technique for computer evaluation of partial derivatives. For some problems, convergence also can be a problem. However, generally this can be overcome by the use of better initial approximations.

The present approach has the same advantages over the classical methods as those of the dynamic programming technique summarized in the introductory section, except that state variable inequality constraints can be handled directly. A distinct advantage is that the approach can handle a nonanalytic objective function without any additional effort.

It is interesting to note the similarity between the equations of the continuous and the discrete systems. The solutions of both systems of equations consist of two terms, the homogeneous solution term and a particular integral term. We have used the property of independence of the homogeneous solution term from the control variable for linear difference or linear differential equations to reduce the dimensionality of the functional equation of dynamic programming. By linear difference or linear differential equations, we mean that the state variables appear linearly in these equations. The control variables $\mathbf{w}$ or $\mathbf{z}$ do not have to appear linearly in these equations. However, the control variables must not appear in the matrix $\mathbf{A}$, which forms the coefficient of the linear state variable term.

Finally, it should be pointed out that the present approach can also be used to optimization problems whose transformation equations are governed by differential-difference equations. The interested reader can consult Bellman and Kalaba (*9*) for detail.

ACKNOWLEDGMENT

Contribution No. 26 Kansas Water Resources Research Institute, Manhattan, Kansas; partial support provided by U.S. Department of the Interior, Office of Saline Water and Office of Water Resources Research.

NOTATION

$\mathbf{A}, \mathbf{B}$ = matrices
a = starting value of the independent variable t, an imbedding variable
$\mathbf{c}$ = solution of the homogeneous difference or differential equation, the imbedding variables
g = an optimum value function, defined by Equation (29), (47), or (78)
H = criterion function
$\mathbf{J}$ = Jacobian matrix
$\mathbf{K}$ = a matrix corresponds to the Green's function
L = a given constant
N = the last stage or the total number of stages
n = the nth stage
$\mathbf{p}$ = a vector defined by Equation (17), (40), or (68)
q = flow rate of raffinate stream
t = the independent variable
t_f = final value of t
u = outlet flow rate of extract stream
v = state variable
w = feed rate of extract solvent or control variable
x = state variable
x_E, y_E = concentrations of components C and A in the extract stream, respectively
x_R, y_R = concentrations of components C and A in the raffinate stream, respectively
z = control variable
ξ = unit sale value of A, a function defined by Equation (5)
λ = Lagrange multiplier

Subscripts

k = kth iteration, assumed known
$k + 1$ = $(k + 1)$st iteration, assumed unknown
m = total number of state variables
M = number of state variables in the criterion function H

LITERATURE CITED

1. Lee, E. S., *AIChE J.*, **13**, 1043 (1967).
2. Leitmann, G., ed., "Optimization Techniques," Chapt. 6 by H. J. Kelley, Academic press, New York (1962).
3. Bryson, A. E., and W. F. Denham, *J. Appl. Mech.*, **29**, 247 (1962).
4. Lee, E. S., *Ind. Eng. Chem. Fundamentals*, **3**, 373 (1964).
5. Merriam, C. W., "Optimization Theory and the Design of Feedback Control Systems," McGraw-Hill, New York (1964).
6. Fine, F. A., and S. G. Bankoff, *Ind. Eng. Chem. Fundamentals*, **6**, 293 (1967).
7. Lee, E. S., "Quasilinearization and Invariant Imbedding," Academic Press, New York (1968).
8. Bellman, R., and S. Dreyfus, "Applied Dynamic Programming," Princeton Univ. press, Princeton, New Jersey, (1962).
9. ———, and R. Kalaba, *J. Basic Eng.*, **83**, 82 (1961).
10. Brand, L,. "Differential and Difference Equations," John Wiley, New York (1966).
11. Fan, L. T., and C. S. Wang, "The Discrete Maximum Principle," John Wiley, New York, (1964).
12. Katz, S., *Ind. Eng. Chem. Fundamentals*, **1**, 226 (1962).
13. Bellman, R., "Stability Theory of Differential Equations," McGraw-Hill, New York (1953).
14. Fan, L. T., "The Continuous Maximum Principle," John Wiley, New York (1966).
15. Rozonoer, L. I., *Automation and Remote Control*, **20**, 1288 (1960).
16. Wengert, R. E., *Comm. ACM*, **7**, 463 (1964).
17. Larson, R. E., Paper presented at IEEE International Convention, New York, (March, 1967).
18. ———, *J. Math. Anal. Appl.*, **11**, 519 (1965).

Manuscript received July 31, 1967; revision received December 5, 1967; paper accepted January 8, 1968.