n = number of point sources used to approximate the line source, dimensionless
r = cylindrical radial coordinate, length, in.
$S_{p,\lambda}$ = point source strength, energy/time
$S_{L,\lambda}$ = linear source strength, energy/length-time
t' = inner reactor wall thickness, length
z = reactor depth coordinate, length, in.
z' = linear source length coordinate, length

Greek Letters

λ = wavelength of radiation, length
ρ = spherical radial coordinate coordinate, length
θ = angle, radians
$\mu_{\lambda,c}$ = absorption coefficient for solution of concentration c, length^{-1}
$\mu_{\lambda,g}$ = absorption coefficient for Pyrex glass wall, length^{-1}

Subscript

λ = light of wavelength λ

LITERATURE CITED

1. Harris, P. R., and J. S. Dranoff, *AIChE J.*, **11**, 497 (1965).
2. Jacob, S. M., and J. S. Dranoff, *Chem. Eng. Progr. Symposium No. 68*, **62**, 47 (1966).
3. *Ibid., No. 89*, **64**, 54 (1968).
4. Dolan, W. J., C. A. Dimon and J. S. Dranoff, *AIChE. J.*, **11**, 1000 (1965).
5. Schechter, R. S., and E. H. Wissler, *Appl. Sci. Res.*, **A9**, 334 (1960).
6. Gaertner, R. F., and J. A. Kent, *Ind. Eng. Chem.*, **50**, 1223 (1958).
7. Huff, J. E., and C. A. Walker, *AIChE J.*, **8**, 193, (1962).
8. Jacob, S. M., Ph.D. dissertation, Northwestern Univ., Evanston, Ill. (1967).
9. Rockwell, T., "Reactor Shielding Manual," p. 392, Van Nostrand, Princeton, N. J. (1956).
10. "Solar Cell and Photocell Handbook," International Rectifier Corp., El Segundo, Calif. (1964).

Manuscript received January 26, 1968; revision received July 15, 1968; paper accepted September 11, 1968.

An Optimal Control Algorithm Using the Davidon-Fletcher-Powell Method with the Fibonacci Search

LOUIS G. BIRTA and PETER J. TRUSHEL

National Research Council, Ottawa, Ontario, Canada

A numerical method for solving a class of nonlinear optimal control problems is presented. The approach reformulates the associated two-point boundary value problem as a multidimensional minimization problem. This problem is, in turn, solved by using the method of Davidon-Fletcher-Powell. The one-dimensional minimization problem implicit in the implementation of the Davidon-Fletcher-Powell algorithm is handled with the Fibonacci search technique. Several examples are presented to demonstrate the effectiveness of the method for problems with and without magnitude constraints on the control variable(s).

A broad class of optimal control problems can, via the necessary conditions of the Pontriagin maximum (minimum) principle be recast as two-point boundary value problems. A variety of approaches to the solution of the boundary value problem have been suggested. One such approach (*1*) reformulates the boundary value problem as a multidimensional minimization problem. This is achieved by introducing an auxiliary error function whose minimization is equivalent to the solution of the boundary value problem. The intent of this paper is to demonstrate that the potential of this approach has not been fully exploited, especially in the light of recent advances in minimization techniques.

Among the first-order multidimensional minimization methods (those requiring only gradient information), a technique originally due to Davidon (*2*) and subsequently refined by Fletcher and Powell (*3*) has become widely accepted as one of the most powerful. It is demonstrated in this paper that the Davidon-Fletcher-Powell (henceforth DFP) algorithm can be effectively utilized in solving a wide class of optimal control problems by applying it to the auxiliary n dimensional minimization problem. A noteworthy property of the DFP algorithm is its capability of locating the minimum of a quadratic function of n variables, in at most n steps. In the context of the control algorithm suggested herein, this feature ensures that the linear system/quadratic cost problem can be solved in at most n iterations, where n is the dimension of the system state vector.*

The implementation of the DFP technique requires the solution of a one-dimensional minimization problem at each stage of the iterative process. In the proposed optimal control algorithm, the Fibonacci search technique is used to accomplish this task.

In many practical situations, the control variable(s) associated with the problem is (are) subject to magnitude constraints. It is shown via several examples that the solu-

* A formal proof of this appears in reference 4.

tion algorithm is capable of handling problems with such constraints.

The mechanics of the solution algorithm are based on a search in the initial costate space. As with all iterative procedures, a priming guess is required to initiate the process. The major difficulty with initial costate space searching algorithms is the problem of uncovering a stable initial guess, that is, one for which the canonical equations have a bounded solution over the interval of interest. By using a suggestion made in (5), it is shown how the proposed algorithm can itself be used to generate such a stable priming point.

PROBLEM STATEMENT AND THE SOLUTION APPROACH

We consider a dynamic system described by a system of n first-order nonlinear equations of the form

$$\dot{x}(t) = f[x(t), u(t), t] \tag{1}$$

The n vector x is referred to as the state vector of the system and the r vector u represents the control vector for the system. The problem is that of finding a control in some admissible set A_u which minimizes a performance (or cost) functional J of the form

$$J = M[x(t_1)] + \int_{t_0}^{t_1} L[x(t), u(t), t]\, dt$$

subject to the differential constraint of Equation (1). In addition, it is assumed that m components $0 \leq m \leq n$ of the state vector x are prescribed at $t = t_1$. With no loss in generality we assume that when $m > 0$, it is desired that $x_i(t_1) = \sigma_i$ for $i = 1, 2, \ldots m$. To complete the problem specification, t_0, t_1, and $x(t_0) = x_0$ are taken to be prescribed. The solution $u = u^*$ of the above problem is called the *optimal control;* it is, furthermore, assumed to exist and to be unique.

Necessary conditions which the optimal control must satisfy can be obtained from the Pontriagin maximum principle. For the above problem formulation, these can be stated as follows.

Choose μ to be any nonzero scalar, and define the Hamiltonian function $H = H[x(t), p(t), u(t), t]$ as $H[x(t), p(t), u(t), t] = p^T(t)f[x(t), u(t), t] + \mu L[x(t), u(t), t]$ (The n vector p is called the *system costate* or *adjoint variable*). Let u^* be an element of A_u, and let x^* be the corresponding trajectory of (1), originating at x_0; that is, $x^*(t_0) = x_0$. In order that u^* be the optimal control, it is necessary that there exist a vector function $p^*(t)$, such that

$$\dot{x}^*(t) = H_p[x^*(t), p^*(t), u^*(t), t] \tag{2a}$$

$$\dot{p}^*(t) = -H_x[x^*(t), p^*(t), u^*(t), t] \tag{2b}$$

with

$$x_i^*(t_1) = \sigma_i \qquad i = 1, 2, \ldots m \tag{3a}$$

$$p_i^*(t_1) = \mu M_{x_i}[x^*(t_1)] \qquad i = m+1, m+2, \ldots n \tag{3b}$$

and that

$$\mu H[x^*(t), p^*(t), u^*(t), t] \leq \mu H[x^*(t), p^*(t), u(t), t] \tag{4}$$

for $t \in [t_0, t_1]$ and all $u \in A_u$. When the components of the control vector are not subject to magnitude constraints, the extremizing condition of (4) implies that

$$H_u[x^*(t), p^*(t), u^*(t), t] = 0 \tag{5}$$

The majority of the algorithms suggested in recent years for solving optimal control problems of the type posed herein are based on the three necessary conditions of Equations (2), (3), and (4) or, alternately, (5). Generally speaking, these methods force two of the necessary conditions and then attempt to satisfy the third via some iterative process. The control function resulting at the termination of the iterative procedure is then taken as the optimal control for the problem under consideration. The approach taken in this presentation follows this same pattern.

We begin by assuming that the equation $H_u[x(t), p(t), u(t), t] = 0$ can be explicitly solved for $u(t)$; that is, $u(t) = U[x(t), p(t), t]$.† Consider, then, the following equations:

$$u(t) = U[x(t), p(t), t] \tag{6}$$

$$\dot{x}(t) = H_p[x(t), p(t), u(t), t]; \quad x(t_0) = x_0 \tag{7a}$$

$$\dot{p}(t) = -H_x[x(t), p(t), u(t), t]; \quad p(t_0) = \omega \tag{7b}$$

and

$$E = \sum_{i=1}^{m} [x_i(t_1) - \sigma_i]^2 + \sum_{i=m+1}^{n} \{p_i(t_1) - \mu M_{x_i}[x(t_1)]\}^2 \tag{8}$$

Let Ω represent that set of n vectors ω for which the differential system of (7), with u given by (6), has a unique solution in the interval $[t_0, t_1]$. Then, for each $\omega \in \Omega$, there corresponds a unique nonnegative value for the scalar function E; that is, the function E (the terminal error function) can be viewed as a function of the initial costate $\omega = p(t_0)$.

Notice that if one could find $\omega^* \in \Omega$ such that $E(\omega^*) = 0$, then the resultant solution of (7) subject to (6) would satisfy the necessary conditions of Equations (2), (3), and (5). The associated control function u, as specified by Equation (6), would, therefore, be taken as the optimal control for the original problem. Furthermore, it should be noted that such an ω^* renders E its minimum value. Consequently, the problem of finding ω^* (and hence of solving the optimal control problem) is equivalent to the problem of minimizing the terminal error function $E = E(\omega)$.

SOLUTION OF THE n DIMENSIONAL MINIMIZATION PROBLEM

As indicated in the preceding discussion, the optimal control problem can be reduced to the problem of minimizing a particular scalar function of n variables, namely, $E = E(\omega)$ of Equation (8). This minimization problem is, however, quite complex inasmuch as the evaluation of the function for any argument requires the solution of a system of $2n$ differential equations, namely Equation (7) together with Equation (6).

The DFP method is proposed as an effective tool in solving this multidimensional minimization problem. A statement of this algorithm can be found in (3). Two points in particular should be stressed. First, the algorithm requires the gradient of the function at each stage of the iteration, and, secondly, a one-dimensional minimization problem must be solved at each stage.

An analytic method for obtaining the gradient of $E(\omega)$ for $\omega \in \Omega$ can be formulated (*1, 4*). This approach requires the reverse-time solution of an auxiliary system of $2n$ differential equations derived from a small perturbation analysis of Equations (6) and (7). However, experience has shown that this approach, although rather elegant from

† $H_u = 0$ might also be solved numerically for $u(t)$.

an analysis point of view, has several practical drawbacks. The derivation of the relevant system of equations for any particular problem can be tedious. Their incorporation into the computer program increases the likelihood of clerical error. And, finally, these equations have been found to be especially ill conditioned, and great care must be taken in obtaining their numerical solution.

It has been found that a far more satisfactory method for obtaining the gradient of $E(\omega)$ is by finite-difference methods. Generally speaking, an absolute perturbation size of 10^{-7} or 10^{-8} has been found to yield good results. The solutions for all examples presented in this paper have been obtained by using finite-difference gradients in the DFP algorithm.

The convergence of the DFP algorithm requires the precise location of a local minimum along each of the algorithm's search directions, that is, the solution of a one-dimensional minimization problem. When derivative information is time consuming to obtain, as it is in the present context, the Fibonacci search method (*6, 7*) is widely recognized as one of the most effective methods for locating a minimum of a function of a scalar variable. The applicability of the method is, however, dependent on knowledge of an interval (the initial interval of uncertainty, or simply IIOU) on which the function is unimodal (single valleyed). A method for handling this requirement in the context of the DFP minimization of the terminal error function $E(\omega)$ is now presented.

Let* $\hat{E}(\alpha) = E(\omega^k + \alpha s^k)$, and observe that for small positive α, $\hat{E}(\alpha)$ is decreasing, since s^k is a descent direction relative to ω^k. Hence, there does exist an interval $I = (0, \alpha')$, on which the function $\hat{E}(\alpha)$ is unimodal and therefore has a local minimum.

A candidate for an IIOU can be obtained through the use of a monotone increasing sequence $(\alpha_0, \alpha_1, \ldots)$ with the following two properties: $\alpha_0 = 0$ and $\hat{E}(\alpha_1) < \hat{E}(\alpha_0)$. The function $\hat{E}(\alpha_j)$, $j = 2, 3, \ldots$, is then evaluated until $\hat{E}(\alpha_j) > \hat{E}(\alpha_{j-1})$, for some j. The interval $A = (\alpha_{j-2}, \alpha_j)$ becomes the candidate for the IIOU.

It should be stressed that while A must contain a local minimum of $\hat{E}(\alpha)$, it does not necessarily follow that $\hat{E}(\alpha)$ is unimodal on A. [Only by making the elements of the series $(\alpha_0, \alpha_1 \ldots)$ arbitrarily close could the latter be guaranteed.] In practice however, it has been found that normally $\hat{E}(\alpha)$ is unimodal on A.

After a prescribed number of iterations, the Fibonacci search reduces the IIOU to a subinterval which contains a local minimum of $\hat{E}(\alpha)$. It has been found that a quadratic fit through three points in this subinterval is effective in locating this local minimum.

THE BOUNDED CONTROL PROBLEM

Thus far the discussion has been confined to those situations where the control vector u is not subject to constraints of any kind. The proposed solution technique can, however, be easily extended to accommodate situations where control constraints of the form

$$|u_j(t)| \leq \beta_j; \quad t \epsilon [t_0, t_1], \quad j = 1, 2, \ldots r$$

are imposed.

Let (x^*, p^*, u^*) denote the optimal trajectory. The method for handling constraints of the above type is based on the fact that for each $t \epsilon [t_0, t_1]$, $u_j^*(t)$ $(1 \leq j \leq r)$ is either on its boundary, or $H_{u_j}[x^*(t), p^*(t), u^*(t), t] = 0$.† Consequently, in the evaluation of the error function $E = E(\omega)$, Equation (6) is replaced by

$$u_j(t) = U_j[x(t), p(t), t]$$

if

$$|U_j[x(t), p(t), t]| \leq \beta_j$$

and

$$u_j(t) = \beta_j \; sgn \; \{U_j[x(t), p(t), t]\}$$

otherwise.

There are circumstances, however, in which the above approach to the bounded control problem can lead to difficulty. This can occur, in particular, when the initial guess ω^0 at the minimizing argument of $E(\omega)$ (that is, the initial guess at the optimal initial costate) is such that $|U_j[x(t), p(t), t]| > \beta_j$ for each $j = 1, 2, \ldots r$ and for all $t \epsilon [t_0, t_1]$. Under these conditions, the evaluation of the gradient $E_\omega(\omega^0)$ (normally) yields the zero vector. This, in turn, causes a collapse of the algorithm.

It has been found that the most practical way to treat bounded control problems is via a sequential procedure [an approach also used in (*5*)]. Namely, the unconstrained problem is first solved, yielding an optimal initial costate of $\bar{\omega}^*$. Bounds which are only slightly restrictive are then imposed, and this new problem is solved by using $\bar{\omega}^*$ as the initial guess. The resultant optimal initial costate $\bar{\bar{\omega}}^*$ is then used as the starting guess for the problem, with somewhat harder constraints imposed. This procedure is continued with progressively harder constraints until the specified bounds are imposed. The sequential nature of this approach is very suggestive of the penalty function methods which have been proposed (*8*) for treating constrained optimization problems.

INITIATION OF THE ALGORITHM

The initiation of the solution procedure requires an initial guess at the minimizing argument of the terminal error function $E = E(\omega)$, that is, an initial guess at the optimal initial costate. In general, however, there are constraints on this initial guess which arise from the fact that Equation (7) [together with (6)] may not have a solution for arbitrary ω. That is, for certain ω, the solution of (7) may become unbounded for $t < t_1$. Furthermore, it is generally impossible to predict which values of ω are unsuitable. The matter of choosing an initial guess for priming the procedure is often quoted (*9, 10*) as the major drawback of optimal control algorithms based on a search in the initial costate space.

In the context of the proposed method, this difficulty can be easily overcome in the following way. Choose $\omega = \omega^0$ to be any convenient n vector. If $\omega^0 \notin \Omega$ [the set of initial costates for which (6) and (7) have a solution on the interval (t_0, t_1)], then there will, in general, exist some $t' \epsilon (t_0, t_1)$ to the left of which the solutions remain tolerably large. Consider, then, the dynamic optimization problem which is identical to the original problem, except that t_1 is replaced by t'. The solution algorithm can then be brought to bear on this modified problem by using ω^0 as the priming guess. Once this problem is solved (normally a crude solution, for example, $E < 1$, is adequate), the associated optimal initial costate $\hat{\omega}^*$ is taken as a candidate for membership in Ω. If, however, $\hat{\omega}^* \notin \Omega$, then there will,

* ω^k and s^k denote, respectively, the k^{th} estimate of the optimal initial costate and the k^{th} search direction as generated by the DFP algorithm.

† The validity of this statement relies on the assumption made prior to Equation (6).

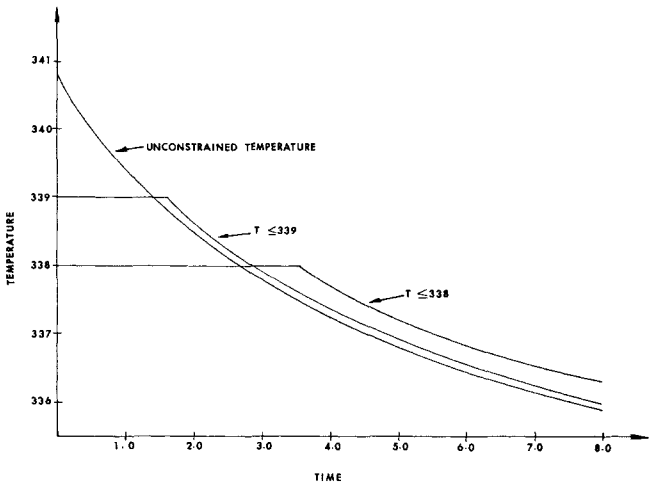

Fig. 1. Optimal controls for the tubular reactor problem.

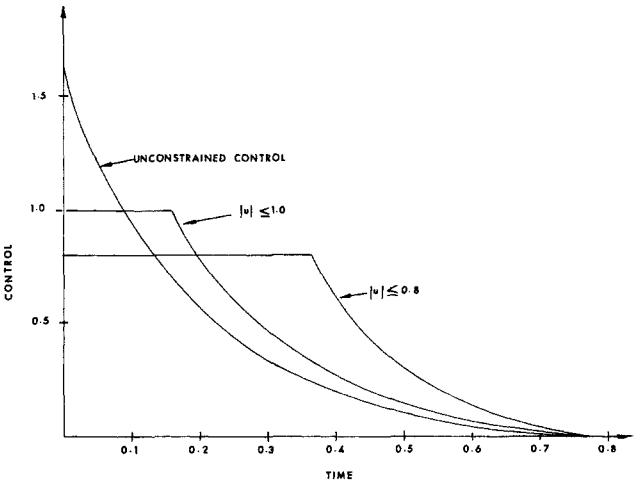

Fig. 2. Optimal controls for the continuous stirred tank reactor problem.

in general, exist $t'' > t'$ in the interval (t_0, t_1) to the left of which the solutions remain tolerably large. A new dummy optimization problem is then posed in terms of t'', and the process is repeated.

The above procedure can be readily automated and has, in fact, been successfully used in treating several problems. The convergence to an element in Ω is quite rapid. Of the specific instances where it was needed, the worst case required five cycles.

It should be emphasized that once the basic algorithm is initiated with a suitable initial estimate $\omega^0 \epsilon \Omega$, all subsequent estimates generated by the minimizing procedure will also be elements of Ω. This follows from the fact that each subsequent estimate of the minimizing argument of the error function necessarily renders it progressively smaller. The possibility that such estimates fall out of Ω is, therefore, precluded.

EXAMPLE PROBLEMS

To provide insight into the effectiveness of the proposed solution algorithm, three example problems are solved. In these examples, and in others which have been treated, it has been observed that increasing the number of function evaluations permitted in the one-dimensional Fibonacci search tends to reduce the number of iterations required for solution. However, it is extremely difficult to decide, a priori, how many function evaluations are warranted at each iteration. On the basis of many experimental tests, a value of 11 has been found to provide a satisfactory compromise between solution efficiency and computation time required. This value was used in all the examples presented.

The first example requires finding the optimal temperature gradient for the tubular chemical reactor considered in (*11* to *13*). For this problem, the error function of Equation (8) is

$$E = p_1^2(t_1) + [p_2(t_1) + \mu]^2$$

Since the parameter μ is at our disposal, it is chosen as $\mu = -p_2(t_1)$. Notice that this value of μ minimizes E for any choice of $p_1(t_1)$ and $p_2(t_1)$.*

This example was solved from a priming initial costate of $(-1, -2)$ with no temperature constraints imposed. The resultant optimal initial costate was used to initiate the solutions for temperature constraints of 339, 338, and 337. The integration step size used in the solution was 0.025.

The second example considers the control of a continuous stirred tank reactor (CSTR), a problem which was originally formulated by Aris and Amundson (*14*) and has been more recently treated by Luus and Lapidus (*15*). By following Luus and Lapidus, the integration step size was taken to be 0.0025. This problem was solved for a priming initial costate of (1, 1) with no control constraints. It was then solved from the resultant optimal initial costate with a control constraint of 1.0. From this priming guess, however, the algorithm was unable to find a solution when control constraints of 0.9 and 0.8 were imposed. The former case was, however, solved from an initial guess of (1.611, 0.3003) and the latter from a guess of (2.230, 0.4337). These are the optimal initial costates when the control is constrained by 1.0 and 0.9, respectively.

As a third example, the van der Pol system considered by Isaacs et al. (*5*) was treated. In this case the integration step size was taken to be 0.0075. This example was solved from a priming initial costate of (9.0, 4.0) with no control constraints. The resultant optimal initial costate

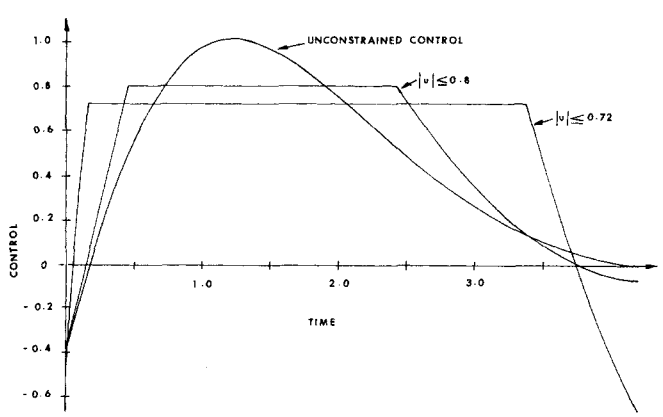

Fig. 3. Optimal controls for the van der Pol problem.

* This approach to simplifying the terminal error function can be directly extended to any Mayer problem.

TABLE 1. SUMMARY OF COMPUTATIONAL RESULTS

	Control constraints		$\omega^0 = (\omega_1^0, \omega_2^0)$	$E(\omega^0)$	No. of iter'ns	$\omega^* = (\omega_1^*, \omega_2^*)$	$E(\omega^*)$	J^*
Continuous stirred tank reactor		None	(1.0, 1.0)	1×10^2	8	(1.07817, 0.191786)	5×10^{-16}	0.02660336
	$\lvert u \rvert$	≤ 1.0	(1.07817, 0.191786)	4×10^2	10	(1.61143, 0.300281)	3×10^{-15}	0.02894491
	$\lvert u \rvert$	≤ 0.9	(1.61143, 0.300281)	3×10^{-1}	9	(2.23005, 0.433737)	5×10^{-16}	0.03181703
	$\lvert u \rvert$	≤ 0.8	(2.23005, 0.433737)	2×10	10	(4.36654, 0.920120)	3×10^{-15}	0.03979814
Tubular reactor		None	$(-1.0, -2.0)$	3×10^2	4	$(-1.33975, -1.81913)$	2×10^{-14}	-0.6794369
	T	≤ 339	$(-1.33975, -1.81913)$	7×10^{-4}	3	$(-1.33795, -1.82046)$	1×10^{-17}	-0.6785257
	T	≤ 338	$(-1.33975, -1.81913)$	5×10^{-3}	4	$(-1.33379, -1.82350)$	2×10^{-17}	-0.676448
	T	≤ 337	$(-1.33975, -1.81913)$	2×10^{-2}	2	$(-1.31798, -1.83459)$	2×10^{-16}	-0.6688820
van der Pol		None	(9.0, 4.0)	1×10^2	10	(4.87288, 0.823135)	1×10^{-15}	2.861494
	$\lvert u \rvert$	≤ 0.8	(4.87288, 0.823135)	7×10^{-1}	9	(6.26695, .819342)	6×10^{-16}	2.980250
	$\lvert u \rvert$	≤ 0.75	(4.87288, 0.823135)	1	10	(8.76832, 0.832110)	3×10^{-17}	3.174784
	$\lvert u \rvert$	≤ 0.72	(8.76832, 0.832110)	7×10^{-2}	8	(18.5289, 1.06019)	6×10^{-16}	3.550721

was used to initiate the solution for control constraints of 0.8 and 0.75. It was also solved for a control constraint of 0.72 with a priming guess of (8.768, 0.8321), that is, the optimal initial costate when the control is constrained by 0.75.

The computational results for these example problems are summarized in Table 1. This table gives the initial guess ω^0, $E(\omega^0)$, the number of iterations required for solution, the optimal initial costate ω^*, $E(\omega^*)$, and the minimum cost J^* for each of the problems. Optimal controls are plotted for each of the examples in Figures 1, 2, and 3.

SUMMARY AND CONCLUSIONS

By recasting the associated two-point boundary value problem as a multidimensional minimization problem, it becomes possible to apply the Davidon-Fletcher-Powell method to a broad class of optimal control problems. This approach to the numerical solution of such dynamic optimization problems is both conceptually straightforward and conveniently implemented, particularly when finite-difference methods are used to obtain gradient information.

In the experiments reported herein, each iteration of the algorithm required in the order of seventeen function evaluations. The computation time required for each such function evaluation is directly proportional to the step size used in the numerical integration process.

The examples treated have demonstrated that the proposed solution method can readily accommodate problems with a bounded control variable. Since the preparation of the original manuscript, the method has also been used to solve problems with free terminal time (that is, free t_1), as well as problems with bang-bang controls and surface constraints on the final state variables (*4*).

By using penalty function methods, it is felt that the algorithm can be extended to effectively handle bounded state problems.

NOTATION

A_u = set of admissible controls
E = terminal error function
H = Hamiltonian function
H_x = n vector whose j^{th} component is $\partial H/\partial x_j$
J = cost functional
L = integrand of the Lagrange portion of the cost functional
M = Mayer portion of cost functional
m = the number of fixed final states
n = the dimension of the state vector
p = costate vector
p_i = i^{th} component of the costate vector
r = dimension of the control vector
s^k = k^{th} search direction
t = independent variable (time)
t_1 = final time
U = function of x, p, and t
u = control vector
u_i = i^{th} component of the control vector
x = state vector
x_i = i^{th} component of the state vector

Greek Letters

α = distance along search direction
β_i = magnitude constraint on u_i
μ = nonzero scalar
σ_i = desired value of x_i at time t_1
ω = initial costate vector
Ω = set of initial conditions for which Equation (7) has a unique solution

Subscripts

0 = initial value
x = partial derivative with respect to x

Superscripts

* = optimal variable
k = nominal variable
T = transpose of matrix

LITERATURE CITED

1. Knapp, C. H., and P. A. Frost, *IEEE Trans. Automatic Control*, **AC-10**, 189 (1965).
2. Davidon, W. C., *AEC Res. Develop. Rept. ANL-5990 (Rev.)* (1959).
3. Fletcher, R., and M. J. D. Powell, *Computer J.*, **6**, 163 (1963).
4. Birta, L. G., and P. J. Trushel, *Natl. Res. Council Rept. MK-25* (1969).
5. Isaccs, D., C. T. Leondes, and R. A. Niemann, "Proceeding Joint Automatic Control Conference," p. 158, Seattle, Wash. (1966).
6. Kiefer, J., *Proc. Am. Math. Soc.*, **4**, 502 (1953).
7. Trushel, Peter J., *Natl. Res. Council Rept. MK-24* (1968).
8. Lasdon, L. S., A. D. Waren, and R. K. Rice, *IEEE Trans. Automatic Control*, **AC-12**, 388 (1967).
9. Rothenberger, B. F., and Leon Lapidus, *AIChE J.*, **13**, 973 (1967).
10. Mitter, S. K., *Automatica*, **3**, 135 (1966).
11. Levine, M. D., *ibid.*, 203.
12. Sutherland, J. W., and E. V. Bohn, "Proceeding Joint Automatic Control Conference," p. 177, Seattle, Wash. (1966).
13. Rothenberger, B. F., and Leon Lapidus, *AIChE J.*, **13**, 114 (1967).
14. Aris, Rutherford, and N. R. Amundson, *Chem. Eng. Sci.*, **7**, 121, 132, 148 (1958).
15. Luus, Rein, and Leon Lapidus, *AIChE J.*, **13**, 108 (1967).

Manuscript received April 18, 1968; revision received August 26, 1968; paper accepted September 23, 1968.

APPENDIX

Example 1: Optimization of Tubular Reactor

It is required to find the optimal temperature gradient in a tubular reactor whose reaction scheme is $A \xrightarrow{K_1} B \xrightarrow{K_2} C$, where B is the desired end product and K_1 and K_2 are the rate constants of the reaction. Let $x_1(t)$ and $x_2(t)$ denote the concentrations of A and B, respectively, where t is the holding time up to a given point. Let $t_1 = 8$ be the total holding time.

The kinetics of the above reaction are

$$\dot{x}_1(t) = -K_1(t)x_1(t) \quad ; \quad x_1(0) = 0.53 \text{ mole/liter}$$

$$\dot{x}_2(t) = K_1(t)x_1(t) - K_2(t)x_2(t); \quad x_2(0) = 0.43 \text{ mole/liter}$$

where $K_i(t) = G_i \text{ EXP}[-E_i/RT(t)]$; $i = 1, 2$, and $T(t)$ is the temperature gradient. The problem is that of maximizing the yield of product B by choosing the best possible temperature gradient. The cost functional associated with the problem can, therefore, be taken as $J = -x_2(t_1)$.

The following system parameters were used: $E_1 = 18{,}000$ cal./mole, $E_2 = 30{,}000$, cal./mole, $G_1 = 0.535 \times 10^{11}$ min.$^{-1}$, $G_2 = 0.461 \times 10^{18}$ min.$^{-1}$, $R = 2$ cal./mole-°K.

For convenience, the control variable can be taken as $u = K_1$. Under these circumstances, Equations (6) and (7) become

$$u(t) = \left[\frac{v(t)}{bc}\right]^{\frac{1}{b-1}}$$

$$\dot{x}_1(t) = -u(t)x_1(t)$$

$$\dot{x}_2(t) = u(t)x_1(t) - cu^b(t)x_2(t)$$

$$\dot{p}_1(t) = u(t)\,[p_1(t) - p_2(t)]$$

$$\dot{p}_2(t) = cp_2(t)u^b(t)$$

where

$$b = E_2/E_1 = \frac{5}{3} \qquad c = \frac{G_2}{G_1{}^b},$$

and

$$v(t) = \frac{x_1(t)\,[p_2(t) - p_1(t)]}{x_2(t)\;p_2(t)}.$$

Example 2: The Continuous Stirred Tank Reactor Problem

This example considers the control of a first-order irreversible exothermic reaction carried out in a CSTR. The control of this system is achieved by injecting coolant through a valve into a cooling coil inserted in the reactor.

Let $x_1(t)$ and $x_2(t)$, respectively, be the deviations from steady state temperature and concentration at time $t \epsilon [0, 0.78]$. Then $x_1(t)$ and $x_2(t)$ are the solutions to the differential system given by

$$\dot{x}_1(t) = -[1 + 2x_1(t)] + R - S, \qquad x_1(0) = 0.05$$

$$\dot{x}_2(t) = 1 - x_2(t) - R \quad , \qquad x_2(0) = 0.0$$

where

$$R = 0.5 + [x_2(t) + 0.5] \text{ EXP}\left[\frac{25x_1(t)}{x_1(t) + 2}\right],$$

$$S = u(t)\,[x_1(t) + 0.25]$$

The performance functional associated with the problem is

$$J = \int_0^{0.78} [x_1{}^2(t) + x_2{}^2(t) + 0.1\;u^2(t)]\;dt$$

For this problem, Equations (6), (7), and (8) are

$$u(t) = 5p_1(t)\;[x_1(t) + 0.25]$$

$$\dot{x}_1(t) = -[1 + 2x_1(t)] + R - S$$

$$\dot{x}_2(t) = 1 - x_2(t) - R$$

$$\dot{p}_1(t) = [p_2(t) - p_1(t)]\,R_{x1} + [2 + u(t)]\,p_1(t) - 2x_1(t)$$

$$\dot{p}_2(t) = [p_2(t) - p_1(t)]\,R_{x2} + p_2(t) - 2x_2(t)$$

$$E = p_1{}^2(t_1) + p_2{}^2(t_1)$$

where

$$R_{x1} = \frac{50\,[x_2(t) + 0.5]}{[x_1(t) + 2]^2}\;\text{EXP}\left[\frac{25x_1(t)}{x_1(t) + 2}\right]$$

$$R_{x2} = \text{EXP}\left[\frac{25x_1(t)}{x_1(t) + 2}\right]$$

Example 3: The van der Pol System

The third problem treated is the forced van der Pol system defined as follows:

$$\dot{x}_1(t) = x_2(t) \qquad ; \qquad x_1(0) = 1.0$$

$$\dot{x}_2(t) = [1 - x_1{}^2(t)]\,x_2(t) - x_1(t) + u(t); \qquad x_2(0) = 0.0$$

It is required to find the control $u = u^*$ which transfers the initial state to the desired final state $(-0.08, 0.0)$ while minimizing the functional

$$J = \int_0^{4.2} [x_1{}^2(t) + x_2{}^2(t) + u^2(t)]\;dt$$

For this problem, Equations (6), (7), and (8) become

$$u(t) = -0.5\;p_2(t)$$

$$\dot{x}_1(t) = x_2(t)$$

$$\dot{x}_2(t) = [1 - x_1{}^2(t)]\,x_2(t) - x_1(t) + u(t)$$

$$\dot{p}_1(t) = p_2(t)\,[2x_1(t)\,x_2(t) + 1] - 2x_1(t)$$

$$\dot{p}_2(t) = -p_1(t) - p_2(t)\,[1 - x_1{}^2(t)] - 2x_2(t)$$

$$E = [x_1(t_1) + 0.08]^2 + x_2{}^2(t_1)$$