Mathematical Programming and Control Theory:
Trends of Interplay*

David G. Luenberger†

I. Introduction

There is a common phenomenon in science for two initially distinct fields to progress independently for a number of years and then later, while still maintaining their individual identity, find that they share a common core of basic principles. Once the commonality is recognized it is natural to expect a variety of attitudes and activities directed toward interaction between the fields. There will be the skeptic who is certain that the other field is merely a special case of his own, and there will be the restless who will immediately turn to this other surely more exciting field. Ideally, there will be cross-fertilization, which amounts to the application of techniques from one field to specific problems in the other. There will be development of the common core of principles, derived from comparison and refinement of the principles already established by each field separately, that ultimately forms the foundation of a more general discipline. Finally, there will be numerous survey articles and lectures on the connections between the two fields.

The fields of mathematical programming and control theory have developed in a fashion similar to that described above. We restrict attention in this paper, however, to the areas of greatest overlap, i.e. to certain subareas within each field. Thus, by mathematical programming we refer here to continuous variable programming, including linear and nonlinear programming. We exclude dynamic programming, not because it is irrelevant but, contrarily, because its connection with control is so direct that it has equal claim by both fields. By control theory we refer exclusively to optimal control, leaving aside the general theory of dynamic systems (stability, controllability, observability, etc.).

---

*This research was supported in part by the National Science Foundation under Grant NSF-GK-16125. Paper presented at the 7th International Symposium on Mathematical Programming, The Hague, The Netherlands, Sept. 1970.

†Stanford University.

General recognition, by researchers in both fields, of the commonality between the fields occurred roughly in about 1962, but naturally it was recognized by many individuals before then. Since that time there has been vigorous interplay between the two fields and numerous conferences whose themes enhanced this interplay. Both fields have, as a result, been substantially influenced by one another. On balance, however, to date the contribution of mathematical programming to control theory is generally better recognized than the reverse contribution. This is due to the fact that mathematical programming, having less inherent structure, can from one viewpoint be regarded as the more general--with control problems being a special case--and to the fact that the modern development of mathematical programming predated modern control theory by about seven years. On the other side, however, the particular structure of control problems introduces some difficulties and some simplifications that were not conceived under the general programming setting. Resolution and exploitation of these has led to both theoretical and algorithmic development in mathematical programming.

Both mathematical programming and control can be naturally dichotomized into the study of properties of solution points, i.e. necessary and sufficient conditions, and the development and analysis of effective computational methods for obtaining a solution. We adhere to this natural dichotomy in the discussion presented in this paper.

Section II presents a brief account of the interplay between the fields that was inspired by the initial gap between the primary sets of necessary conditions for optimality: the Kuhn-Tucker Theorem for mathematical programming, and the Pontryagin maximum principle for control. Here we point out how the initial gap was closed by working from both directions. Sections III and IV discuss computational methods. In this area the contribution has been almost entirely from mathematical programming to control, but there have been differences in emphasis and misinterpretations among the two groups that has tended to retard full development in this area. We point out some of these differences in viewpoint and offer some suggestions for closing these gaps.

The paper is not intended to be a comprehensive survey of work in either field nor of the interaction between them (a discussion of algorithms alone would constitute a large volume). Instead, the paper attempts to point out how certain initial differences were satisfactorily resolved, and attempts to bring into focus some issues that have not yet received full attention and which require further interactive investigation. The selection of topics that are highlighted in this paper are, of course, somewhat influenced by the author's interest, areas of experience, and personal assessments as to where further interaction will be most fruitful.

II. Optimality Conditions

2.1 Continuous-Time Problems

The Pontryagin Maximum Principle is the fundamental set of necessary conditions for continuous-time optimal control problems. We state a version of it here partly to display the result itself and partly for the purpose of introducing the continuous-time optimal control problem [1].

<u>Continuous-Time Optimal Control Problem (Fixed Terminal Time)</u>

Consider a dynamical system described by the system of ordinary differential equations

$$\frac{d}{dt} x(t) = h\big(x(t), u(t)\big) \qquad t \in [0,T] \tag{2.1}$$

where $x(t) \in E^n$ is the <u>state</u> of the system at time t, $u(t) \in E^p$ is the <u>control input</u> at time t, and h takes values in E^n. Given an initial state $x_0 \in E^n$, a <u>cost function</u> f from $E^n \times E^p$ into the reals, a set $U \subset E^p$, and a <u>terminal constraint function</u> g from E^n into E^m, $m \leq n$, find the piecewise continuous control $u(\cdot)$ on $[0,T]$ and corresponding state trajectory $x(\cdot)$ determined from (2.1) satisfying

$$u(t) \in U \quad \text{for all} \quad t \in [0,T] \tag{2.2a}$$

$$x(0) = x_0 \tag{2.2b}$$

$$(2.2c) \qquad g\big(x(T)\big) = 0$$

that minimizes the objective

$$(2.3) \qquad J = \int_0^T f\big(x(t),u(t)\big)dt \ .$$

There are alternative formulations of the continuous-time optimal control problem in which the objective is expressed as a function of the terminal state and some auxiliary constraints are expressed as constraints on the value of an integral involving the state and control. These alternative formulations are, under mild smoothness assumptions, all equivalent. One formulation can be transformed to another by introduction of appropriate additional state variables and associated augmentation of the system equations. These transformations are now quite familiar to control theorists and we shall not belabor them here, but make free implicit use of them throughout the paper.

To formulate the maximum principle we introduce the additional assumptions that the functions f and h are continuous differentiable with respect to x and continuous with respect to u and that the function g is continuously differentiable, and its Jacobian $m \times n$ matrix has rank m for all x satisfying $g(x) = 0$.

Pontryagin Maximum Principle

If $\hat{u}(\cdot)$ together with its associated state trajectory $\hat{x}(\cdot)$ is a solution to the fixed terminal time optimal control problem, then there exist an adjoint trajectory $\lambda(\cdot)$ from $[0,T]$ into E^n, a $\lambda^o \leq 0$, and a $\mu \in E^m$ such that $(\lambda^o,\mu) \in E^{m+1}$ is not zero and

$$(2.4) \quad -\frac{d}{dt}\lambda(t) = \left[\frac{\partial h}{\partial x}\big(\hat{x}(t),\hat{u}(t)\big)\right]'\lambda(t) + \lambda^o\left[\frac{\partial f}{\partial x}\big(\hat{x}(t),\hat{u}(t)\big)\right]'$$

for $t \in [0,T]$,

$$(2.5) \qquad \lambda(T) = \left[\frac{\partial g}{\partial x}\big(\hat{x}(T)\big)\right]'\mu$$

and such that for almost all $t \in [0,T]$, the Hamiltonian

$$(2.6) \qquad H\big(\hat{x}(t),\lambda(t),v\big) = \lambda^o f\big(\hat{x}(t),v\big) + \lambda'(t)h\big(\hat{x}(t),v\big)$$

is maximized by $\hat{u}(t)$, i.e. for every $v \in U$

$$(2.7) \qquad \lambda^o f\big(\hat{x}(t),\hat{u}(t)\big) + \lambda'(t)h\big(\hat{x}(t),\hat{u}(t)\big) \geq \lambda^o f\big(\hat{x}(t),v\big) + \lambda'(t)h\big(\hat{x}(t),v\big).$$

An unusual feature of this principle, over that of more classical necessary conditions, is that they are partly expressed in differential form, as in (2.4), and partly in global form, as in (2.7). This mixture of differential and global Lagrange multiplier conditions is, as discussed later in this section, a challenging point even in a finite-dimensional setting.

The multiplier $\lambda^o \leq 0$ is of special concern for if it is equal to zero then the objective does not enter the conditions. A non-degenerate problem will have $\lambda^o < 0$, in which case it is set to $\lambda^o = -1$, but we shall not pursue the conditions that guarantee this.

The original proof of the maximum principal given in [1] is quite involved, closely following classical arguments in the calculus of variations and physics, where the term Hamiltonian was borrowed, and bearing little relation to work in mathematical programming. Indeed, for a few years it was the relations between the calculus of variations and continuous-time control that highlighted work on necessary conditions--mathematical programming being excluded as being restricted to finite-dimensional problems.

Attention quickly was directed, both in mathematical programming and control circles, to an attempt to generate a parallel result for discrete time optimal control problems--the former circle because the resulting problem is finite-dimensional, the latter because sampled systems were becoming increasingly important as computers found their way into control applications. This was the first strong point of interaction between the fields.

2.2 Discrete-Time Problems

Discrete Time Optimal Control Problem (Fixed Terminal Time)

Consider a dynamical system described by the system of difference equations

$$(2.8) \qquad x(k+1) - x(k) = h\big(x(k),u(k)\big) \quad , \quad 0 \le k \le N-1$$

where $x(k) \in E^n$, $u(k) \in E^p$, and h maps into E^n. Given an initial state $x_0 \in E^n$, f mapping from $E^n \times E^p$ into the reals, a set $U \subset E^p$, and g mapping from E^n into E^m, find the sequence of controls $u(k)$, $k = 0,1,\ldots,N-1$ and corresponding state trajectory determined from (2.8) satisfying

$$(2.9a) \qquad u(k) \in U \quad , \quad 0 \le k \le N-1$$

$$(2.9b) \qquad x(0) = x_0$$

$$(2.9c) \qquad g\big(x(N)\big) = 0$$

that minimizes the objective

$$(2.10) \qquad J = \sum_{k=0}^{N-1} f\big(x(k),u(k)\big) \ .$$

Again it is assumed that f and h are continuously differentiable with respect to x but only continuous with respect to u; and that the Jacobian matrix of g has rank m for all x satisfying $g(x) = 0$ [2].

The problem of obtaining a satisfactory result for the discrete-time problem was soon found to be extremely challenging and was taken up by both mathematical programming and control theorists. Available results for finite-dimensional problems, such as the Kuhn-Tucker Theorem, were able to yield results of the mixed differential and global form only for the most simple problem structures (e.g. linear dynamics, convex costs, and convex constraints); hence this problem motivated work directed toward extending these classical results. A notable contribution in this respect is a generalized Fritz John result [3], [4] which yields a maximum principle result for the discrete-time problem.

There is no unique way in which to characterize inherent difference between discrete-time problems and continuous-time problems which leads to the analytic difficulty of obtaining a maximum principle. From the viewpoint of more or less classical variational theory the difference can be traced to the fact that in continuous-time a control history perturbation of large magnitude but arbitrarily short duration can be considered which introduces only small changes in the resulting trajectory; while in discrete-time a large magnitude perturbation of control at any instant introduces a large change in the trajectory. Thus, only in continuous-time is it possible to introduce control perturbations that are simultaneously large in magnitude (at a given instant) but small in their effect on the objective and constraint functionals. Alternatively, from a viewpoint more closely connected with mathematical programming, the difference between continuous-time and discrete-time problems is that a certain set of trajectory and points defined by allowing a family of perturbations is convex in the continuous-time case but not necessarily convex in the discrete-time case.

From a control theoretic viewpoint (i.e. by explicit consideration of trajectories) Halkin [5] obtained a mixed differential-global maximum principle for the discrete-time problem that exploits a convexity assumption. His result was slightly strengthened by Holtzman [6] who pointed out that convexity could be replaced by directional convexity, a concept closely related, as we point out later in this section, to that of a convex epigraph popular in mathematical programming.

While work progressed on the discrete-time problem two other related activities were also, at least partially, inspired by the maximum principle. The first was the further development of a general theory of optimization valid in infinite-dimensional spaces. Here the aim was to obtain results having the simple structural characteristics of the classical programming results but wider applicability. This theory is characterized by the work of Liusternik [7], Hurwicz [8], Rockafellar [9], Luenberger [10], and many others [11-15]. The second activity, initiated by Gamkrelidze [16] and developed by Neustadt [17-19], Halkin [20], and Canon, Cullum, and Polak [2], [21], was aimed more directly toward the maximum principle.

This took the form of a generalized theory of optimization for problems having a finite number of equality constraints from which the Pontryagin maximum principle (discrete or continuous-time) as well as the classical mathematical programming results are easily deduced.

It is impossible to obtain a maximum principle for the discrete-time optimal control problem that is as strong as that for the corresponding continuous-time problem. One must settle for a weaker condition than global maximization of the Hamiltonian over U, or some sort of convexity assumption must be introduced to yield the maximization condition. We state here one version that closely parallels the original one for continuous-time problems[2].

Discrete-Time Maximum Principle

Assume that for every $x \in X^n$ the set in E^{n+1} defined by

$$\{(r,z): \ r = f(x,u), \ z = h(x,u), \ u \in U\}$$

is $(-1,0,0,\ldots,0)$-directionally convex. (See below).

If $\hat{u}(\cdot)$ together with its associated state trajectory $\hat{x}(\cdot)$ is a solution to the discrete-time optimal control problem, then there exist an adjoint trajectory $\lambda(k) \in E^n$, $k = 0,1,2,\ldots,N$, a $\lambda^o \leq 0$, and a $\mu \in E^m$, such that $(\lambda^o,\mu) \in E^{m+1}$ is not zero and

$$(2.11) \quad \lambda(k) - \lambda(k+1) = \left[\frac{\partial h}{\partial x}\left(\hat{x}(k),\hat{u}(k)\right)\right]' \lambda_{k+1} + \lambda^o\left[\frac{\partial f}{\partial x}\left(\hat{x}(k),\hat{u}(k)\right)\right]'$$

for $k = 0,1,2,\ldots,N-1$,

$$(2.12) \qquad \lambda(N) = \left[\frac{\partial g}{\partial x}\left(\hat{x}(N)\right)\right]' \mu$$

and such that for every $v \in U$ and $k = 0,\ldots,N-1$,

$$(2.13) \quad \lambda^o f\left(\hat{x}(k),\hat{u}(k)\right) + \lambda'(k)h\left(\hat{x}(k),\hat{u}(k)\right) \geq \lambda^o f\left(\hat{x}(k),v\right) + \lambda'(k)h\left(\hat{x}(k),v\right) .$$

2.3 Local-Global Results

In the remainder of this discussion of optimality conditions we turn to the definition of directional convexity, its relation to epigraphs, and to an elaboration upon the theme most striking about the original maximum principle--the categorization of the problem variables into local and global parts. We show that this theme can easily be incorporated into the general theory of optimization.

A set S in a vector space V is said to be e-directionally convex, where $e \in V$, if for every vector v in the convex hull of S there is a vector $s \in S$ and $\beta \geq 0$ such that $v = s + \beta e$.

An equivalent and perhaps simpler definition is that S is e-directionally convex if the set $\Omega = \{v: \ v = s - \beta e, \ \beta \geq 0, \ s \in S\}$ is convex. In other words, if the sum of S and the ray $-\beta e$ is a convex set. If f is a functional on a convex subset Ω, then f is convex if its graph in $R \times V$ defined by $\{(r,v): \ r = f(v), v \in \Omega\}$ is $(-1,0)$-directionally convex, which is equivalent to f having a convex epigraph.

We conclude this section with a theorem that is not difficult to prove but which captures the spirit of the original maximum principle and transfers it to mathematical programming. The proof itself employs ideas used frequently in mathematical programming and thus also captures the spirit of that field. In order not to introduce new concepts the result is stated here in finite-dimensional form, but is easily extended to situations where both the unknowns and the constraints are infinite dimensional. The result is, in a sense, the analog for inequalities of the general results obtained by Neustadt and others [16-21] for problems with a finite number of equalities. The generalized inequality result, however, is easier to prove, is not restricted to a finite number of constraints, and carries with it a regularity assumption that guarantees a unity coefficient on the objective in the Lagrangian.

Local-Global Programming Problem

Find x and u to

$$\text{(2.13a)} \qquad \text{minimize} \quad f(x,u)$$

(2.13b) $\text{subject to} \quad g(x,u) \leq 0$

(2.13c) $u \in \Omega$

where $x \in E^n$, $\Omega \subset E^p$ is convex, and the functions f and g take values in the reals and E^p respectively, and both are continuously differentiable with respect to x. For each fixed x the set in E^{p+1}

$$\{(r,z): \quad r = f(x,u), \ z \geq g(x,u), \quad u \in \Omega\}$$

is $(-1,0,0,\ldots,0)$-directionally convex.

This problem possesses standard differentiability properties with respect to x and convexity properties with respect to u. Indeed, if f and g are both convex with respect to u for each fixed x, the directional convexity assumption will be satisfied, and for fixed x the problem is a convex program in u. We introduce a regularity condition that is a natural combination of standard conditions for local and global theories [10]--that if x_0, u_0 is a solution, then there is a $h_0 \in E^n$, $v_0 \in \Omega$ such that

(2.14) $$g(x_0,v_0) + \nabla_x g(x_0,u_0)h_0 < 0 .$$

Lagrange-Multiplier Result

Let x_0, u_0 be an optimal solution to the local-global program (2.13) and suppose the solution satisfies the regularity condition (2.14). Then there is a $\lambda \in E^p$, $\lambda \geq 0$ such that

(2.15) $$\nabla_x f(x_0,u_0) + \lambda' \nabla_x g(x_0,u_0) = 0 ,$$

(2.16) $$f(x_0,u_0) + \lambda' g(x_0,u_0) \leq f(x_0,v) + \lambda' g(x_0,v) ,$$

for all $v \in \Omega$, and

(2.17) $$\lambda' g(x_0,u_0) = 0 .$$

Proof.

Define the mapping G taking values in E^{p+1} by

$$G(x,u) = \left(f(x,u) - f(x_0,u_0),\ \ g(x,u)\right) .$$

In E^{p+1} defines the sets

$$A = \{y: \ y \geq G(x_0,v) + \nabla_x G(x_0,u_0)h, \ \ v \in \Omega\}$$

$$B = \{y: \ y \leq 0\} .$$

Both of these sets are convex--A because of the directional convexity assumption--and B has nonempty interior.

We now show that A contains no interior points of B. Suppose there is a point in A which is interior to B. This means there is $h \in E^n$, $v \in \Omega$ such that

$$G(x_0,v) + \nabla_x G(x_0,u_0)h < 0 .$$

The point $G(x_0,v) + \nabla_x G(x_0,u_0)h$ is the center of some open sphere contained in the negative orthant N of E^{p+1}. Suppose this sphere has radius $\rho > 0$. Then $\alpha[G(x_0,v) + \nabla_x G(x_0,u_0)h]$ is the center of such a sphere having radius $\alpha\rho$; hence so is the point

$$(1-\alpha)G(x_0,u_0) + \alpha[G(x_0,v) + \nabla_x G(x_0,u_0)h] .$$

This point can be written

$$\begin{aligned}
&G(x_0,u_0) + \alpha\nabla_x G(x_0,u_0)h + \alpha[G(x_0,v) - G(x_0,u_0)] \\
&= G(x_0 + \alpha h,u_0) + \alpha[G(x_0,v) - G(x_0,u_0] + o(\alpha) \\
&= (1-\alpha)G(x_0 + \alpha h,u_0) + \alpha G(x_0 + \alpha h,v) \\
&\quad + \alpha[G(x_0 + \alpha h,u_0) - G(x_0,u_0)] \\
&\quad - \alpha[G(x_0 + \alpha h,v) - G(x_0,v)] + o(\alpha) \\
&= (1-\alpha)G(x_0 + \alpha h,u_0) + \alpha G(x_0 + \alpha h,v) + o(\alpha) .
\end{aligned}$$

By the assumption of directional convexity there is a point $v_\alpha \in \Omega$ such that

$$G(x_0 + \alpha h, v_\alpha) \leq (1-\alpha)G(x_0 + \alpha h, u_0) + \alpha G(x_0 + \alpha h, v) .$$

Thus the point $G(x_0 + \alpha h, v_\alpha) + o(\alpha)$ is contained in a sphere of radius $\alpha\rho$ in the negative orthant. Hence, for sufficiently small α $G(x_0 + \alpha h, v_\alpha) < 0$. This contradicts the optimality of x_0, u_0 and hence A contains no interior points of B.

In view of the above conclusions there is a hyperplane separating the sets A and B. Thus, there is a vector in E^{p+1} of the form $\psi = (r, \lambda)$ with r real, $\lambda \in E^p$, and a δ such that

$$\psi' y \geq \delta \quad \text{for all} \quad y \in A$$

$$\psi' y \leq \delta \quad \text{for all} \quad y \in B .$$

We take $\delta = 0$ since the point $(0,0)$ belongs to both A and B. Since B is the negative orthant it follows immediately that $\psi \geq 0$ or equivalently that $r \geq 0$, $\lambda \geq 0$. Furthermore, $r \neq 0$ because otherwise the hyperplane would not separate the point $G(x_0, v_0) + \nabla_x G(x_0, u_0) h_0$ from B. Hence, without loss of generality we may take $r = 1$.

From the separation property we must have

$$\psi' G(x_0, u_0) \geq 0$$

and thus

$$\lambda' g(x_0, u_0) \geq 0$$

but since

$$\lambda \geq 0, \quad g(x_0, u_0) \leq 0$$

it follows that

$$\lambda' g(x_0, u_0) = 0 .$$

From the separation property for A we have

$$f(x_0,v) - f(x_0,u_0) + \lambda' g(x_0,v) + f_x(x_0,u_0)\, h + \lambda' g_x(x_0,u_0) h \geq 0 \,.$$

Setting $h = 0$ and using (2.17) yields (2.15). Noting that h is arbitrary yields (2.16).

III. Computation (Continuous-Time Problems)

3.1 Evaluation of Gradient

Consider, initially, the following simple structure.

<u>Continuous-Time Optimal Control</u>

Given a dynamical system

$$\frac{d}{dt}\, x(t) = h\big(x(t),u(t)\big) \tag{3.1}$$

together with initial condition $x(0) = x_0$ where $x(t) \in E^n$, $u(t) \in E^p$, $t \in [0,T]$ and given the objective functional

$$J = \int_0^T f\big(x(t),u(t)\big)\, dt \tag{3.2}$$

where f and h are both continuously differentiable with respect to x and u, find $u(\cdot)$ to minimize J.

This structure is the simplest nontrivial one for control since it is unconstrained except for the dynamical relation between $x(\cdot)$ and $u(\cdot)$. Indeed, the most profitable manner in which to view this problem is as an unconstrained problem with respect to $u(\cdot)$. Note that if $u(\cdot)$ is specified on $[0,T]$ then $x(\cdot)$ is determined uniquely by the dynamic system (3.1) and the initial conditions. This means that the value of the objective functional is also uniquely determined. Thus, we often write $J(u)$, rather than simply J, to explicitly point out this implicit dependence.

To obtain necessary conditions for a problem with this structure or to formulate an algorithmic scheme for its solution, it is most natural to seek an expression for the gradient of J (with respect to u). Thus given $u(\cdot)$ and its associated $x(\cdot)$ we have

$$\nabla_u J(u) = \lambda(t)' \nabla_u h\big(x(t),u(t)\big) + \nabla_u f\big(x(t),u(t)\big) \tag{3.3}$$

where

$$(3.4a) \qquad -\frac{d}{dt}\lambda(t) = \left[\nabla_x h\left(x(t),u(t)\right)\right]'\lambda(t) + \nabla_x f\left(x(t),u(t)\right)'$$

$$(3.4b) \qquad \lambda(T) = 0 \; .$$

Necessary conditions are obtained by setting this gradient to zero.* For computation we note that given $u(\cdot)$, forward integration of the system equations (3.1) yields $x(\cdot)$. Then backward integration of the adjoint equations (3.4) yields $\lambda(\cdot)$, and the gradient can be directly evaluated from (3.3). Additional constraints are incorporated in the necessary conditions through introduction of Lagrange multipliers and in computational techniques by gradient projection, etc.

This procedure for evaluation of the gradient forms the backbone of most algorithms in use designed to solve control problems. It can be regarded as the basic technique for exploiting the dynamic structure of the problem.

3.2 Direct Approach for Continuous-Time Problem

The usual approach taken by control theorists when solving a continuous-time problem is to carry out analyses and develop algorithms in continuous time and then execute the required computations as accurately as necessary on a digital computer. This viewpoint more or less excludes consideration of simplex-like algorithms since they are not applicable to infinite-dimensional problems but does not exclude the numerous programming techniques that can be generalized to infinite dimensions.

There have been a number of distinct techniques suggested for the continuous-time problem, essentially all of which parallel techniques developed for general nonlinear programming problems. One class of techniques are those that attempt to solve the equations

*The adjoint variable $\lambda(\cdot)$ resulting from (3.4) is the negative of that in the Pontryagin formulation. The form (3.4) seems, from many respects, the most natural, and it has been argued that things would have been simpler if Pontryagin had used (3.4) and formulated a "minimum principle." As it is, both formulations of the adjoint equation occur with about equal frequency in the literature.

expressing the necessary conditions [22]. In this approach $u(t)$ is expressed in terms of $\lambda(t)$ and $x(t)$ from (2.7). The remaining unknowns $x(\cdot)$, $\lambda(\cdot)$ then satisfy a system of $2n$ ordinary differential equations with boundary conditions at both the initial and final times. This reduces the problem to that of numerically solving a two-point boundary value problem. Both first-order (simple successive approximation) and second-order (Newton-type) methods have been developed to solve the resulting two-point boundary value problem.

Another technique, called the epsilon method [23], is to consider the dynamic system equation as a differential constraint and treat it by a penalty function technique. Generally, this yields an ill-conditioned problem. If, however, the problem has special analytical structure and if a second order minimization routine can be used, the technique is workable.

By far the most widely used methods are descent methods such as the gradient method [24], [25], conjugate gradients [26], [27], and Newton's method [28-31]. All of these have solved a variety of optimal control problems with good success.

The success of gradient method lies, as pointed out above, in the simple procedure available for evaluation of the gradient in a control problem. There is a parallel to this for Newton's method that makes second-order methods computationally feasible. The second-order correction is found by integrating a matrix Ricatti equation, of dimension $n \times n$, backward from T to the initial time. Thus, the effort to obtain the second-order correction is only about n times that of obtaining the gradient. Details can be found in [30]. The discrete-time analog of the Ricatti equation is discussed in Section 4.2 below.

The precise details associated with these methods, step size selection, incorporation of constraints, etc., are largely identical to those developed for general nonlinear programming. Thus, although this is an area of major cross fertilization between mathematical programming and control (mainly from the former to the latter), it is senseless to attempt to detail these procedures here.

3.3 Free Terminal Time Problems

In the free terminal time problem the final time T is not specified *a priori* but is determined as part of the optimization. In problems of this type there is usually a terminal constraint on the state vector that implicitly determines the final time for a given trajectory. This is exemplified by the problem of guiding a rocket to the moon with minimum fuel expenditure; or the aircraft climb problem where it is desired to control the angle of attack of an aircraft so as to reach a given altitude in minimum time. The terminal time associated with a trajectory is determined by the terminal constraint.

Problems of this type cannot directly be cast into the framework of a vector space defined over $[0,T]$ as is done, at least implicitly, in fixed-time problems. Several techniques have been employed to handle the problem indirectly. One, often used in conjunction with good physical insight, is to reparameterize the equations of motion with respect to the terminal constraint variable. In the aircraft climb problem, for example, the dynamic equations might be reparameterized with altitude as the independent running variable, instead of time, so that the problem is transformed into one on a fixed interval. This scheme is most effective only if the terminal constraint can be expressed in terms of a variable that itself can be expected to increase along a trajectory.

Another approach to variable time problems, a favorite among mathematical programmers, is to solve, for various values of terminal time T, the associated problem having fixed terminal time T. Successive trial values of T can be generated by a curve fitting procedure applied to the points of cost versus final time. Logically, this technique amounts to having an inner loop that minimizes with respect to $u(\cdot)$ and an outer loop that minimizes with respect to T.

We suggest here that problems with variable terminal time can often be most effectively solved through the introduction of a penalty function, which in these situations not only eliminates the terminal constraint by modifying the objective but also allows one to consider the problem as though it were of fixed duration.

We write the problem as

$$
\begin{aligned}
(3.5) \qquad & \min_{u(\cdot),T} \int_0^T f\big(x(t),u(t)\big)\,dt \\
& \text{subject to} \quad \dot{x}(t) = h\big(x(t),u(t)\big) \;, \quad x(0) = x_0 \\
& g\big(x(T)\big) \doteq 0
\end{aligned}
$$

and cast the problem into vector space format by considering that $u(\cdot)$ is defined on an interval $[0,T_1]$ where T_1 is sufficiently large to contain any interval of interest. We then solve the approximating penalty problem

$$
(3.6) \qquad \min_{u(\cdot),x(\cdot)} \; \min_{T} \int_0^T f\big(x(t),u(t)\big)\,dt + K\big\|g\big(x(T)\big)\big\|^2
$$

where

$$
\dot{x}(t) = h\big(x(t),u(t)\big) \quad , \quad x(0) = x_0
$$

and K is some (large) positive constant.

The minimization problem is unconstrained and can be regarded as being one of finding $u(\cdot)$ on $[0,T_1]$ with T being determined from $u(\cdot)$ by minimization along the resulting trajectory. The problem is thus a standard fixed-time constraint-free problem, except that the objective function does not have the usual integral form but must be evaluated by minimization over T--a computationally minor modification since the integral is automatically generated as a function of T. Furthermore, the gradient of this objective can be calculated by backward integration from the current T just as in fixed time problems [32].

3.4 Discretization of Continuous-Time Problems

When employing a digital computer to solve a continuous-time optimal control problem, operations such as the solution of differential equations cannot be executed exactly. This means that either the original problem itself or its solution (or both) must be approximated in some appropriate fashion. Aside from some fairly specialized function approximation schemes, there are two primary approaches to this approximation problem--both of which are based on discretization of time. Crudely, the approaches can be labeled the mathematical programming and the control theory approaches--but these labels are not meant to have any real significance; they are based on the observation that workers from each field have tended to favor one particular approach to the discretization problem.

The first approach, that favored by mathematical programmers, is to discretize the original problem at the outset, converting it to a discrete-time optimal control problem, and then later direct attention toward the solution of the resulting finite-dimensional problem. The simplest scheme for doing this is to define $\Delta = \frac{T}{N}$ for some positive integer N and introduce the notation $x(k) \equiv x(k\Delta)$, $u(k) \equiv u(k\Delta)$. Next, define the difference equations

$$\frac{x(k+1) - x(k)}{\Delta} = h\Big(x(k),u(k)\Big) \quad , \quad 0 \leq k < N \tag{3.7}$$

and the objective functional

$$J = \sum_{k=0}^{N-1} f\Big(x(k),u(k)\Big) \ , \tag{3.8}$$

which approximate the system differential equations and objective respectively. If the original problem has additional constraints, they are discretized in a similar manner.

There are a number of important questions regarding the appropriateness of such approximations in terms of their convergence to the true solution as Δ goes to zero. For information on this topic see Cullum [33] and Daniel [34]. Roughly, however, the accuracy can be expected to be on the order of Δ, i.e. $O(\Delta)$. In some circumstances, of course, it may be advantageous to use nonuniform discretization intervals.

This first approach to discretization is attractive philosophically because the approximation is made at the outset and then an accurate answer to the approximate problem can be sought. It is attractive to a mathematical programmer because the whole array of programming techniques becomes applicable to the approximate problem.

Unfortunately, it can easily get out of hand yielding a problem of enormously large dimension with poor accuracy properties.

The second approach, the one largely employed by control theorists, is based on analysis of the continuous-time problem. The problem is constantly regarded as being one in continuous time and one attempts to carry out, at least approximately, the calculations dictated by the continuous time analysis. The primary components of these calculations is integration of the system differential equations forward from $t = 0$ to $t = T$ given a control history over this interval and integration of the adjoint equations backward to obtain the gradient. Again, this is done by consideration of N equally spaced points in the interval $[0,T]$, but the first-order Euler approximation is generally found to be too crude for practical use--or more precisely, Δ must be taken extremely small to yield a satisfactory approximation. It is more efficient to use higher order integration schemes such as a predictor-corrector, or Runge-Kutta procedure [35], [36].

For illustration consider the system

$$\frac{d}{dt} x(t) = h\Big(x(t),u(t)\Big) \quad (3.9)$$

We lay out intervals of equal width Δ and use the correspondence $x(k) \equiv x(k\Delta)$ as before. A standard corrector formula, such as the Adams-Moulton method, has the form

$$x(k+1)-x(k) = \Delta \sum_{i=k-q_k}^{k+1} \beta_{ik} h\Big(x(i),u(i)\Big) , \quad k = 0,1,2,\ldots,N-1 . \quad (3.10)$$

Usually, q_k is a constant, say q, for all values of k except near $k = 0$. To initialize the process q values $x(0), x(1),\ldots, x(q-1)$ must be generated by some other method, or we can develop a scheme with $q_k = k$ for $k \le q$. The value $q = 4$ is often used for the main part of the procedure.

The right-hand side of the corrector formula (3.10) depends on $x(k+1)$ and hence (3.10) cannot be used to determine $x(k+1)$ without iteration. For this reason a predictor formula, similar to

(3.10) but with different coefficients and without $x(k+1)$ on the right-hand side, is used to obtain an initial estimate of $x(k+1)$. Then starting from this estimate the corrector is solved by successive approximation.

Typically, when solving optimal control problems, the same integration procedure is used both for forward integration of the system equations and for backward integration of the adjoint system used for calculation of the gradient. The procedure is also used, in the case where a second-order procedure is being incorporated, for the backward integration of the matrix Riccati equation. This control theoretic approach to discretization has found wide acceptance because of its convenience and high accuracy, and because it fully exploits the simple method for generating the gradient.

It is impossible to muster a definitive argument that favors either of these two approaches over the other. Which is better is partially dependent on the overall problem structure beyond that of the dynamics alone. For the majority of continuous-time problems, however, the control theoretic viewpoint has a great advantage because of the relatively short time required for accurate integration. It is only problems having a plethora of constraints on the state vector and the control, or which have a particularly simple structure, such as being linear or quadratic programs (and for which consequently, gradient-based schemes must be replaced or supplemented by simplex-type algorithms) that the mathematical programming approach is superior.

We suggest here that it is possible to incorporate the best features of both of the above viewpoints into a single philosophically satisfying and computationally effective discretization viewpoint. Like the mathematical programmer, we decide that it is best to discretize the original problem into an approximating finite-dimensional one that can be attacked directly. Like the control theorist, however, we recognize that simple Euler discretization is too crude and that a more sophisticated approach must be taken. In order to make our discussion somewhat specific let us agree that a standard corrector formula with fixed step size Δ is an appropriate approximation to both the dynamic system and to the evaluation of the cost function. This, in turn, will produce a discretization of the problem of the form

$$X(k+1) = \Phi\left(X(k),U(k)\right) \tag{3.11}$$

$$J = c^T x(N) \tag{3.12}$$

where the dimensions of $X(k)$ and $U(k)$ are greater than n and p by factors that depend on the particular corrector formula employed. For convenience in what follows, we have assumed that the problem has been put in the form of a terminal cost problem.

Philosophically, introduction of the corrector formula can be viewed as a transformation of the original problem to a discrete-time control problem. Practically, however, this does not allow us to bring into play the algorithm applicable to most discrete problems, since the Φ function is made up of implicit functions that can only effectively be evaluated by forward recursion in conjunction with an associated predictor.

Holding to the viewpoint that the discrete problem obtained through the somewhat indirect process of using a predictor-corrector scheme should be solved exactly, we next consider the question of evaluating the associated gradient. In the pure control theoretic approach the gradient is approximated by applying the predictor-corrector in reverse time to evaluate λ. We instead consider the possibility of evaluating this gradient exactly for the discrete formulation obtained. Evaluation of the gradient then forms the basis for a family of optimization techniques.

Suppose that the optimal control problem is expressed as

$$\text{minimize} \quad J = c^T x(N) \tag{3.13}$$

where

$$x(k+1)-x(k) = \Delta \sum_{i=k-q_k}^{k+1} \beta_{ik} h\left(x(i),u(i)\right) , \quad k = 0,1,2,\ldots,N-1 , \tag{3.14}$$

$$x(0) = x_0.$$

To find the gradient of J with respect to $u(\cdot)$, multiply (3.14) by $\lambda(k)$, sum over k, and evaluate the total differential of the result, obtaining

$$\sum_{k=0}^{N-1} \lambda(k)dx(k+1) - \lambda(k)dx(k) = \sum_{k=0}^{N-1} \Delta \sum_{i=k-q_k}^{k+1} \beta_{ik}\lambda(k)\Big[h_x\big(x(i),u(i)\big)dx(i) + h_u\big(x(i),u(i)\big)du(i)\Big] .$$

Or upon rearrangement,

$$\lambda(N-1)dx(N) + \sum_{i=1}^{N-1} [\lambda(i-1)-\lambda(i)]dx(i) = \Delta \sum_{i=1}^{N-1} \left\{\sum_{k} \beta_{ik}\lambda(k)\right\}\Big[h_x\big(x(i),u(i)\big)dx(i)\Big] + h_u\big(x(i),u(i)\big)du(i) .$$

Thus defining the λ sequence by

$$\text{(3.15)} \quad \lambda(i-1)-\lambda(i) = \Delta \sum_{k} \beta_{ik}\lambda(k)h_x\big(x(i),u(i)\big) , \quad i = 1,2,\ldots,N-1$$

$$\lambda(N-1) = c$$

$$dJ = \lambda(N-1)dx(N) = \Delta \sum_{i} \left[\sum_{k} \beta_{ik}\lambda(k)\right]h_u\big(x(i),u(i)\big)du(i)$$

and hence

$$\text{(3.16)} \qquad (\nabla J)_i = \Delta \left[\sum_{k} \beta_{ik}\lambda(k)\right]h_u\big(x(i),u(i)\big) .$$

This formula is similar to the adjoint equation used before, but it is not calculated by direct application of the corrector formula to the adjoint differential equation, as is done in the pure control theoretic approach. In the form presented here the λ sequence is evaluated by the simple backward formula (3.15) and then $\sum_k \beta_{ik}\lambda(k)$ is evaluated. In practice, it may be desirable to develop the appropriate recursion for $w(i) = \sum_k \beta_{ik}\lambda(k)$ and solve for it directly in a single backward sweep.

IV. Computation (Discrete-Time Problems)

4.1 General Remarks

If a control problem is inherently one in discrete time or if there are compelling reasons for using the Euler discretization of a continuous-time problem, then it takes the form

$$x(k+1) = \Phi\Big(x(k),u(k)\Big) \ , \quad k = 0,1,2,\ldots,N-1$$

$$x(0) = x_0$$

$$J = \sum_{k=0}^{N-1} f\Big(x(k),u(k)\Big) + \Gamma\Big(x(N)\Big)$$

together with possibly several constraints restricting $x(k)$ and $u(k)$, $k = 0,\ldots,N$. Since the problem arose directly in this form, presumably the functions f and Φ and the constraint functions are known explicitly. In this case the problem can be regarded as a finite-dimensional mathematical programming problem and the full range of algorithms becomes available.

There has been a good deal of computational experience accumulated on problems of this type which, on the whole, has essentially verified that general nonlinear programming algorithms can effectively solve control problems. Indeed, control problems often serve as convenient sources of large dimensional test problems for new algorithms. Those algorithms which use gradients can, of course, exploit the simple formula for the gradient of a discrete-time control problem. Generally, however, these straightforward applications of known algorithms to control problems carry little significance toward the fundamental development of either mathematical programming or optimal control. It is only when the special structure of large control problems is exploited in the development of an algorithm that there is a meaningful contribution.

A special case that is obviously potentially rich in structure is the linear problem, where the system equations, objective functional, and constraints are all linear. Some linear programs arising in this way have a natural structure that can be exploited to yield algorithms that are more efficient than the standard simplex method. A notable contribution in this respect is the work initiated by Dantzig [37] on dynamic Leontief models.

Another potentially rich special structure is where the system equations and constraints are linear and the objective function is quadratic (in both x and u). This leads to a quadratic programming problem that can be solved by standard algorithms. The dominant structure of these problems is often due to the dynamic equations rather than additional constraints and an efficient algorithm would fully exploit this. Indeed, the solution to the "linear system, quadratic cost" problem is one of the fundamental results from control theory.

4.2 Quadratic Problems

In the absence of additional constraints a simple form of the quadratic cost control problem is

(4.1a) $$x(k+1) = \Phi x(k) + Bu(k) \quad , \quad k = 0,1,\ldots,N-1$$

(4.1b) $$x(0) = x_0$$

(4.2) $$J = \frac{1}{2}\sum_{k=1}^{N} x(k)'Qx(k) + u(k-1)'Ru(k-1)$$

where Q and R are respectively $n \times n$ and $p \times p$ positive semi-definite matrices with either Q or R positive definite. Recalling that J may be regarded as a function of the sequence $u(\cdot)$, with the $x(\cdot)$ sequence determined from (4.1), it is clear that in this case J is a quadratic function of $u(\cdot)$. From a mathematical programming viewpoint the problem is equivalent to one of the form

$$\min USU - 2AU$$

where U is the control sequence vector of dimension $p \cdot N$ and S is the induced $p \cdot N \times p \cdot N$ matrix. The solution is $U = S^{-1}A$ which is trivial but unusable because S is not given explicitly and has very high dimension. The special solution available for this problem can thus be regarded as a method for exploiting the special structure of S. Indeed, if one pursues the details of constructing S, the special recursive solution available can be viewed as a special formula for solving a system of equations whose coefficient matrix is block tridiagonal. We do not state the recursive solution for this problem here since a more general problem is treated below.

Constrained Quadratic Cost Control Problem

Given the discrete dynamic equations

$$x(k+1) = \Phi x(k) + Bu(k) \quad , \quad k = 0,1,2,\dots,N-1 \; , \tag{4.3a}$$

the initial condition

$$x(0) = \bar{x}_0 \; , \tag{4.3b}$$

the objective

$$J = \frac{1}{2} \sum_{k=1}^{N} x(k)'Qx(k) + u(k-1)'Ru(k-1) \; , \tag{4.4}$$

and the constraints

$$Cx(k) \leq c \qquad k = 1,2,\dots,N \tag{4.5}$$

$$Du(k) \leq d \qquad k = 0,1,2,\dots,N-1 \tag{4.6}$$

where C and D are $m_1 \times n$ and $m_2 \times p$ respectively, find the sequences $u(\cdot)$, $x(\cdot)$ that minimize the objective while satisfying the constraints.

Again $x(\cdot)$ can be explicitly eliminated from the formulation resulting in a quadratic programming problem of dimension $p \cdot N$ and requiring a great deal of storage. To exploit the dynamic structure of the problem we may employ duality theory [10] to write the problem in the equivalent dual form [38]

(4.7) $$\text{maximize} \quad \psi(\mu, \eta)$$

subject to $$\mu \geq 0 \ , \quad \eta \geq 0$$

where the dual function ψ is defined by

$$\psi(\mu, \eta) = \min_{u,x} \sum_{k=1}^{N} \frac{1}{2} x(k)'Q(k)x(k) + \frac{1}{2} u(k-1)'R(k)u(k-1)$$

$$+ \mu(k)'[Cx(k)-c] + \eta(k)'[Du(k-1)-d]$$

with

$$x(k+1) = \Phi x(k) + Bu(k) \quad , \quad k = 0,1,2,\ldots,N-1 \ ,$$

$$x(0) = x_0 \ .$$

Assuming for the moment that the dual function can be evaluated efficiently, we note that it is itself a quadratic function of μ, η. Thus the dual problem is a quadratic programming problem but the inequality constraints are simple positivity constraints.

Quadratic programming problems with only positivity constraints can be solved by simple modification of the gradient or conjugate gradient method [38], [39, [40]. Fortunately, the gradient of the dual function is readily computable once the dual function is evaluated. We have

(4.9a) $$\nabla_{\mu(k)} \psi = Cx(k) - c \quad k = 0,1,2,\ldots,N-1 \ ,$$

(4.9b) $$\nabla_{\eta(k)} \psi = Du(k-1) - d \quad k = 1,2,\ldots,N \ ,$$

where $x(\cdot)$ and $u(\cdot)$ are the minimizing trajectory and control history determined by evaluation of the dual function. Thus, once an effective procedure for evaluating the dual function is developed, efficient quadratic programming algorithms can be brought to bear on the problem. The result is a nontrivial synthesis of mathematical programming and control theory techniques for the solution of an otherwise almost impossible problem.

Evaluation of the dual function is (for fixed μ and η) a standard quadratic loss control problem with some additional linear terms. It can be shown that its solution can be found recursively from the following relations [38]

(4.10a) $P(k-1) = Q(k) + \Phi' P(k) S(k) \quad , \quad k = 1,2,\ldots,N \; ,$

(4.10b) $P(N) = Q \; ,$

(4.11) $M(k) = -[R + BP(k)B]^{-1} BP(k)\Phi \; ,$

(4.12) $S(k) = \Phi + BM(k) \; ,$

(4.13) $f(k-1) = S'(k)f(k) + M'(k)D'\eta(k) + C'\mu(k-1), \; f(N) = C'\mu(N) \; ,$

(4.14) $u(k) = M(k+1)x(k) + y(k) \; ,$

(4.15) $x(k+1) = \Phi x(k) + Bu(k) \quad , \quad x(0) = x_0 \; ,$

(4.16) $y(k) = -[R(k) + B'P(k)B]^{-1}[Bf(k) + B'\eta(k)] \; .$

Although at first these relations may seem formable we note that (4.10), (4.11), and (4.12) are independent of μ and η and hence $P(k)$, $M(k)$, $S(k)$, $k = 0,1,2,\ldots,N$ can be computed <u>once</u> and used for all future iterations. Calculation of $f(\cdot)$ and $y(\cdot)$ is then accomplished by a single backward recursion and finally $x(\cdot)$ and $u(\cdot)$ are obtained by a forward recursion. The result yields the gradient of the dual problem. Thus, taking advantage of the structure in this manner we have a method for computing the gradient of the dual function that requires one backward and one forward recursion--about the same work as that required for evaluation of the gradient of a primal control problem (in the primal case the forward recursion precedes the backward).

4.3 Speed of Convergence

We conclude by indicating some preliminary results in an area that has to date received no formal attention but which ultimately will certainly play a dominant role in the rational development of algorithms for control: convergence analysis of algorithms. It can be argued that convergence analysis, particularly speed of convergence, has unfortunately lagged behind the formal development of mathematical programming algorithms of all types. The only widely known result is the analysis of steepest descent. In control theory the situation is even worse since there has been virtually no analysis even of steepest descent. In what follows we offer an analysis of a most elementary optimal control problem solved by steepest descent in order to indicate the kind of qualitative conclusions that can be drawn.

We consider the quadratic problem

$$x(k+1) = \phi x(k) + bu(k) \quad , \quad x(0) = x_0 \;, \tag{4.17}$$

$$J = qx(N)^2 + \sum_{k=0}^{N-1} u(k)^2 \tag{4.18}$$

where for simplicity we assume that both $x(k)$ and $u(k)$ are scalars. This problem can be easily solved analytically or numerically by finding a Riccati-type sequence, but our concern is not with the question of solving the problem explicitly but with using it as a model to analyze steepest descent. In expanded form the problem can be written as a quadratic minimization problem in terms of an N-dimensional vector U. The quadratic form associated with the expanded problem is

$$\Gamma = I + b^2 q \Phi' \Phi \tag{4.19}$$

where Φ is the row vector

$$\Phi = (\phi^{N-1}, \phi^{N-2}, \ldots, 1) \; .$$

Thus, the matrix Γ is equal to the identity plus a diad.

It is well known [10] that for a quadratic problem the method of steepest descent converges according to

$$J(u_{k+1}) \le \left(\frac{A-a}{A+a}\right)^2 J(u_k) \tag{4.20}$$

where a and A are respectively the smallest and largest eigenvalues of the quadratic form matrix Γ. Since Γ has a particularly simple structure for this problem we can see that there is one eigenvalue of magnitude $1 + b^2 q \|\Phi\|^2$ and $N-1$ eigenvalues of value unity. Thus, the rate of convergence can be written explicitly as

$$\left[\frac{b^2 q \|\Phi\|^2}{2 + b^2 q \|\Phi\|^2}\right]^2 .$$

Several qualitative conclusions can be drawn from this result. We see that for unstable systems $(|\phi| > 1)$ $\|\Phi\|^2$ grows geometrically with N and hence as N increases it becomes increasingly difficult to solve the problem by steepest descent. For stable systems $(|\phi| < 1)$ the difficult increases slowly with N, approaching a finite limit. In general, the more stable the system the faster will be the convergence of steepest descent.

Similar convergence analyses can be made for more complex quadratic problems, for continuous as well as discrete systems, and for algorithms other than steepest descent. These provide a concrete basis of comparison among problems and solution techniques.

V. Conclusions

Although the similarity between mathematical programming and optimal control has been widely recognized for at least eight years, interplay between the two fields has been less pronounced than might be expected, and surely less than potentially possible. This is largely due, it seems, to the fact that a major part of the interplay has been the activity of control theorists rather than mathematical programmers. The development of the unified theory of optimization, for instance, which includes both the continuous-time and discrete-time optimal control problems as well as classical mathematical programming results, was largely the result of efforts on the part of control theorists striving to close the gap that existed between necessary conditions in the two fields. The majority of existing algorithms used to solve control problems on a routine basis were developed from a continuous-time viewpoint by control theorists who borrowed algorithms from mathematical programming. Mathematical programmers, not motivated by the original practical problems themselves, have, for the most part, taken a fairly passive role in the development of solutions to control problems. There is still much room for interaction, particularly in the areas of convergence analysis and exploitation of structure--areas where mathematical programming has a great tradition.

References

1. Pontryagin, L. S., V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko, The Mathematical Theory of Optimal Processes, Interscience, New York, 1962.

2. Canon, Michael D., Clifton D. Cullum, Jr., and Elijah Polak, Theory of Optimal Control and Mathematical Programming, Chapters 1,2,4, McGraw-Hill Book Company, New York, 1970.

3. Mangasarian, Olvi L., Nonlinear Programming, McGraw-Hill Book Company, New York, 1969.

4. Mangasarian, Olvi L. and S. Fromovitz, "Fritz John Necessary Optimality Conditions in the Presence of Equality and Inequality Constraints," J. Math. Analysis and Applications, Vol. 17, pp. 37-47; 1967.

5. Halkin, Hubert, "A Maximum Principle of the Pontryagin Type for Systems Described by Nonlinear Difference Equations," J. SIAM Control, Vol. 4, No. 1, pp. 90-111; Feb. 1966.

6. Holtzman, J. M., "On the Maximum Principle for Nonlinear Discrete Time Systems," IEEE Trans. for Automatic Control, Vol. 11, pp. 273-274; 1966.

7. Liusternik, L. and V. Sobelev, Elements of Functional Analysis, Frederick Ungar, New York, 1961.

8. Hurwicz, L., "Programming in Linear Spaces," Studies in Linear and Nonlinear Programming (K. J. Arrow, L. Hurwicz, and H. Uzawa, eds.), Stanford University Press, Stanford, Calif., pp. 38-102; 1958.

9. Rockafellar, R. T., Convex Analysis, Princeton University Press, Princeton, N.J., 1970.

10. Luenberger, David G., Optimization by Vector Space Methods, John Wiley and Sons, Inc., New York, 1969.

11. Goldstine, H. H., "A Multiplier Rule in Abstract Spaces," Bull. Amer. Math. Soc., 44, pp. 388-394; 1938.

12. Goldstein, A. A., "Convex Programming and Optimal Control," J. SIAM Control, Vol. 3, No. 1, pp. 142-146; 1965.

13. Halkin, H., "Optimal Control as Programming in Infinite Dimensional Space," in C.I.M.E.: Calculus of Variations, Classical and Modern (E. R. Cremonese, Ed.), Roma, pp. 179-192; 1966.

14. Russell, D. L., "The Kuhn Tucker Conditions in Banach Space with an Application to Control Theory," J. Math. Anal. and Appl., Vol. 15, pp. 200-212; 1966.

15. Varaiya, P. P., "Nonlinear Programming in Banach Spaces," J. SIAM Appl. Math., Vol. 15, No. 2, pp. 284-293; 1967.

16. Gamkrelidze, R. B., "On Some Extremal Problems in the Theory of Differential Equations with Applications to the Theory of Optimal Control," J. SIAM Control, Vol. 3, pp. 106-128; 1965.

17. Neustadt, Lucien W., "An Abstract Variational Theory with Applications to a Broad Class of Optimization Problems. I. General Theory," J. SIAM Control, Vol. 4, No. 3, pp. 505-527; Aug. 1966.

18. -------, "An Abstract Variational Theory with Applications to a Broad Class of Optimization Problems. II. Applications," J. SIAM Control, Vol. 5, No. 1, pp. 90-137; Feb. 1967.

19. -------, "A General Theory of Extremals," J. Computer and System Sciences, Vol. 3, pp. 57-92; 1969.

20. Halkin, H. and L. W. Neustadt, "General Necessary Conditions for Optimization Problems," Proceedings of National Academy of Sciences, Vol. 56, (4), pp. 1066-1071; 1966.

21. Canon, Michael D., Clifton D. Cullum, Jr., and Elijah Polak, "Constrained Minimization Problems in Finite-Dimensional Spaces," J. SIAM Control, Vol. 4, No. 3, pp. 528-547; Aug. 1966.

22. Breakwell, John V., "The Optimization of Trajectories," J. Soc. Indust. Appl. Math., Vol. 7, pp. 215-247; June 1959.

23. Balakrishnan, A. V., "On a New Computing Technique in Optimal Control," J. SIAM Control, Vol. 5; May 1968.

24. Bryson, A. E., Jr. and W. F. Denham, "A Steepest Ascent Method for Optimal Programming Problems," Journal of Applied Mechanics, Vol. 29, No. 2, pp. 247-257; 1962.

25. Kelley, H. J., "Method of Gradients," Optimization Techniques (G. Leitmann, ed.), Academic Press, New York, 1962.

26. Lasdon, L. S., S. K. Mitter, and A. D. Warrer, "The Conjugate Gradient Method for Optimal Control Problems," Trans. IEEE, AC-12, pp. 132-138; April 1967.

27. Sinnott, J. F. and D. G. Luenberger, "Solution of Optimal Control Problems by the Method of Conjugate Gradients," reprint of paper presented at JACC, July, 1967.

28. McReynolds, S. R. and A. E. Bryson, Jr., "A Successive Sweep Method for Solving Optimal Control Problems," JACC, pp. 551-555; 1965.

29. Bullock, T. E., "Computation of Optimal Controls by a Method Based on Second Variations," SUDAAR No. 297, Stanford University; Dec. 1966.

30. Bryson, Arthur E., Jr., and Yu-Chi Ho, Applied Optimal Control, Blaisdell Publishing Company, Waltham, Mass., 1969.

31. Jacobson, D. H., "Second-Order and Second Variation Methods for Determining Optimal Control: A Comparative Analysis Using Differential Dynamic Programming," Intl. J. of Control, Vol. 7, No. 2, pp. 175-196; 1968.

32. Luenberger, D. G., "A Primal-Dual Method for the Computation of Optimal Control," 2nd International Conference on Computing Methods in Optimization Problems, San Remo, Italy, Sept. 9-13, 1968.

33. Cullum, Jane, "Perturbations of Optimal Control Problems," J. SIAM Control, Vol. 4, No. 3, pp. 473-487; Aug. 1966.

34. Daniel, James W., "The Approximate Minimization of Functionals by Discretization," to appear.

35. Henrici, Peter, Discrete Variable Methods in Ordinary Differential Equations, John Wiley and Sons, Inc., New York, 1968.

36. Hamming, R. W., Numerical Methods for Scientists and Engineers, McGraw-Hill Book Company, New York, 1962.

37. Dantzig, George B., "Optimal Solution of a Dynamic Leontief Model with Substitution," Econometrica, 23, pp. 295-302; 1955.

38. Maxfield, Robert Roy, "Techniques for Computing Optimal Controls for Linear Systems with Inequality Constraints," Ph.D. Dissertation, Stanford University, Stanford, Calif.; Feb. 1969.

39. Hildreth, C., "A Quadratic Programming Procedure," Naval Research Logistics Quarterly, Vol. 4, pp. 79-85; 1957.

40. Houthakker, H. S., "The Capacity Method of Quadratic Programming," Econometrica, Vol. 28, pp. 62-87; 1960.