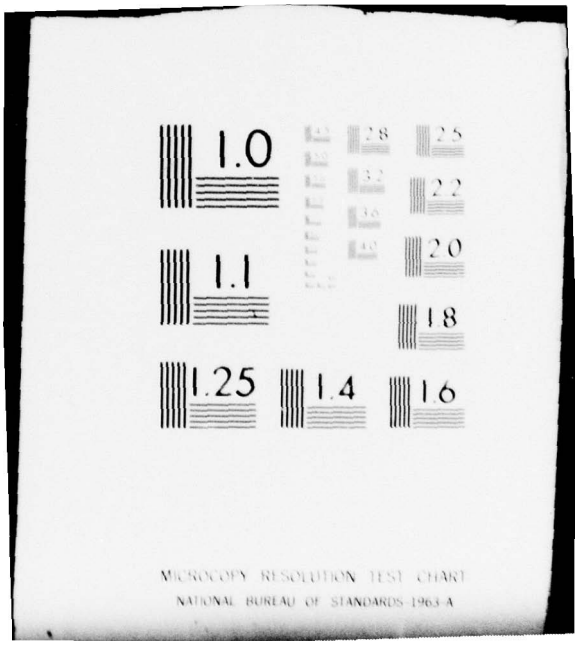
AD-A070 311 AERONAUTICAL RESEARCH LABS MELBOURNE (AUSTRALIA) F/G 12/1
GRADIENT OPTIMIZATION ALGORITHMS FOR SYSTEMS WITH TERMINAL CONS--ETC(U)
AUG 78 R B ZMOOD

UNCLASSIFIED ARL-SYS-NOTE-58 NL

| OF |
AD
A070311

END
DATE
FILMED
8-79
DDC


1.0
1.1
1.25
1.4
1.6
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.8
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

ARL-SYS-NOTE-58

AR-001-297

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

AERONAUTICAL RESEARCH LABORATORIES

MELBOURNE, VICTORIA

SYSTEMS NOTE 58

GRADIENT OPTIMIZATION ALGORITHMS FOR SYSTEMS WITH TERMINAL CONSTRAINTS

by

RONALD B. ZMOOD

Approved for Public Release.

COPY No 11

AUGUST 1978

AR-001-297

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
AERONAUTICAL RESEARCH LABORATORIES

(14) ARL — SYSTEMS NOTE-58

(6) # GRADIENT OPTIMIZATION ALGORITHMS FOR SYSTEMS WITH TERMINAL CONSTRAINTS

by

(10) RONALD B./ZMOOD

(11) Aug 78 (12) 23 p.

SUMMARY

The application of optimization techniques to the derivation of predictive information for flight displays is being investigated in connection with operational situations involving large disturbances and manoeuvres. This note surveys current gradient methods for computing extremal solutions of optimal control problems for systems having boundary conditions but without state or control constraints. It is concluded that a two stage procedure is required, with the first stage using a first order gradient, while the second stage would use a higher order variable metric method. Proposals are advanced for further research on optimization algorithms.

POSTAL ADDRESS: Chief Superintendent, Aeronautical Research Laboratories,
Box 4331, P.O., Melbourne, Victoria, 3001, Australia.

CONTENTS

1. INTRODUCTION

The extension of flight director capabilities to a wide range of operational situations including major manoeuvres and disturbances is becoming feasible through advances in digital avionics and displays. This will permit the timely presentation of alternative strategies to the aircrew, for their use in decision and control. These types of predictive aids, which for example would be useful for STOL aircraft landing at forward tactical airfields, and for combat aircraft involved in tactical engagements, are receiving increased attention in the United States and NATO countries.[1-5] Research work is being initiated at ARL to gain an improved understanding of the potentialities and limitations of these systems.

The determination of the optimal manoeuvres required to achieve specific objectives, whilst taking into account physical and operational constraints, would enhance the value of a director under normal operating conditions, and would be of especial value in emergency situations. This note surveys current methods used for computing extremal solutions of optimal control problems for systems having boundary constraints, but without state or control constraints. The extension to the general case with constraints through penalty function and other techniques will be the subject of further work.

Particular attention is focused on the so-called "direct methods" and as many of these methods cannot be immediately applied to problems having terminal constraints, a discussion of techniques for adapting them to this need is presented. Finally a number of approaches not studied in the literature, which are worthy of further examination will be briefly considered.

2. ALGORITHMS FOR UNCONSTRAINED PROBLEMS

Computational algorithms for seeking the extremal solution of general optimal control problems fall into two main groups, which are known as

(*a*) Indirect Methods, and

(*b*) Direct Methods.

In addition to the methods mentioned above, special algorithms for finding the optimal control of systems described by linear differential equations have also been extensively developed. As the equations of motion for the flight mechanics problems of interest are generally non-linear these methods are inapplicable and so will not be considered further.

The flight mechanics optimization problems being considered can be formulated within the framework of the general optimization problem of Bolza: for a fixed final time, t_f, find an unbounded control function $u(t)$, $t \in [t_o, t_f]$ which minimizes the cost functional

$$J(u) = \phi[x(t_f), t_f] + \int_{t_o}^{t_f} L(x, u, t)dt \tag{2.1}$$

subject to the differential equation

$$\frac{dx}{dt}(t) = f(x, u, t), \qquad x(t_o) = x_o \tag{2.2}$$

and terminal conditions

$$\left.\begin{aligned} \psi_1[x(t_f), t_f] &= 0 \\ &\vdots \\ \psi_q[x(t_f), t_f] &= 0, \end{aligned}\right\} \tag{2.3}$$

where $q \leqslant n$, and n is the dimension of the state vector $x(t)$. It will be convenient to write the terminal conditions (2.3) in the vector function form

$$\psi[x(t_f), t_f] = 0, \tag{2.4}$$

where we define the vector function $\psi = [\psi_1, \ldots, \psi_q]^T$.† In the present work it will be assumed that the functions $\phi : R^n \times R^1 \to R^1$, $L : R^n \times R^m \times R^1 \to R^1$, $f : R^n \times R^m \times R^1 \to R^n$ and $\psi : R^n \times R^1 \to R^q$, which define the optimization problem, are continuously differentiable in all arguments.

2.1 Indirect Methods

These methods use an iterative scheme to solve some of the necessary conditions for optimality, while satisfying the remaining conditions exactly. Two methods which have met with a degree of success in applications are

(*a*) the neighbouring extremal or shooting method, and

(*b*) the quasi-linearization method.

The necessary conditions[6] for optimality of the problem defined by (2.1)–(2.3) are most easily stated by use of the Hamiltonian $H(x, u, \lambda, t)$ which is defined by

$$H(x, u, \lambda, t) = L(x, u, t) + \lambda^T(t) f(x, u, t), \tag{2.5}$$

where $\lambda(t)$ is the Lagrangian multiplier function. These conditions give rise to the two-point boundary value problem of finding functions $x(t)$, $u(t)$ and $\lambda(t)$, $t \in [t_o, t_f]$ which simultaneously satisfy

(i) the n state equations, $\dfrac{dx}{dt}(t) = \dfrac{\partial H}{\partial \lambda}(x, u, \lambda, t)$;

(ii) the n adjoint equations, $\dfrac{d\lambda}{dt}(t) = -\dfrac{\partial H}{\partial x}(x, u, \lambda, t)$;

(iii) the m optimality conditions, $\dfrac{\partial H}{\partial u} = 0$;

(iv) the n initial conditions $x(t_o) = x_o$;

(v) the q terminal conditions $\psi[x(t_f), t_f] = 0$; and

(vi) the n–q adjoint terminal conditions $\lambda^T(t_f) = \left[\dfrac{\partial \phi}{\partial x} + \nu^T \dfrac{\partial \psi}{\partial x}\right] t = t_f$, where ν is an arbitrary q-vector.

2.1.1 Neighbouring Extremal Algorithms

These methods which are also known as shooting or root finding methods in R^n, attempt to solve conditions (i)–(iii) exactly by direct integration of the differential equations, while using an iterative procedure to converge to the satisfaction of conditions (iv)–(vi), as discussed in References 6 and 7.

In the case where condition (iii), above, can be explicitly solved for $u(t)$ in terms of $x(t)$ and $\lambda(t)$, this variable can be eliminated from the differential equations given in (i) and (ii) above. Adjoining vectors $x(t)$ and $\lambda(t)$ to form the $2n$-vector

$$y(t) = \begin{bmatrix} x(t) \\ \lambda(t) \end{bmatrix}, \tag{2.6}$$

it can easily be seen that the equations given in (i) and (ii) above, with $u(t)$ eliminated, take the form of the two-point boundary value problem

† The transpose of a matrix A will be denoted by A^T.

$$\frac{dy}{dt}(t) = F(y, t), \tag{2.7}$$

where

$$F(y, t) = \begin{bmatrix} \dfrac{\partial H}{\partial \lambda}(x, u(x, \lambda)(t), t) \\ \dfrac{-\partial H}{\partial x}(x, u(x, \lambda)(t), \lambda, t) \end{bmatrix},$$

and the boundary conditions are

$$y(t_o) = \begin{bmatrix} x_o \\ \lambda(t_o) \end{bmatrix}, \text{ and } \begin{bmatrix} \psi(x(t_f), t_f) \\ \lambda(t_f) - \dfrac{\partial \phi}{\partial x} - \nu^T \dfrac{\partial \psi}{\partial x} \end{bmatrix} = 0. \tag{2.8}$$

For simplicity, in the remainder of this discussion it will be assumed that $x(t_f) = x_f$, where x_f is a given vector. In this case the terminal condition given in (2.8) simplifies to

$$y(t_f) = \begin{bmatrix} x_f \\ \lambda(t_f) \end{bmatrix}. \tag{2.9}$$

One approach to solving this problem is to consider that (2.7) implicitly defines a function, $\bar{F}$, mapping $\lambda(t_o) \in R^n$ into $(x(t_f) - x_f) \in R^n$, so that it now becomes a problem of finding a $\lambda(t_o)$ such that $\bar{F}(\lambda(t_o)) = 0$. This can be solved by using the classical Newton-Raphson method where the interates are constructed as follows

$$\lambda_{i+1}(t_o) = \lambda_i(t_o) - \left[\frac{\partial \bar{F}}{\partial \lambda(t_o)}\right]^{-1} \bar{F}(\lambda_i(t_o)), \qquad i = 0, 1, 2, 3, \ldots; \tag{2.10}$$

assuming that $[\partial \bar{F}/\partial \lambda(t_o)]^{-1}$ exists. In applying this method, Bryson[6] suggests the following three approaches for determining the multiplier matrix $[\partial \bar{F}/\partial \lambda(t_o)]^{-1}$:

(i) direct numerical differentiation by perturbing each component of $\lambda(t_o)$ in turn, followed by integration of (2.7) and matrix inversion of $\partial \bar{F}/\partial \lambda(t_o)$;

(ii) unit solutions using second variation equations, followed by matrix inversion of $\partial \bar{F}/\partial \lambda(t_o)$;

(iii) using the backward sweep method, which yields $[\partial \bar{F}/\partial \lambda(t_o)]^{-1}$ directly.

Methods (i) and (ii) often suffer from severe numerical sensitivity because the matrix $\partial \bar{F}/\partial \lambda(t_o)$ is often ill-conditioned. However, the third approach tends to be much less sensitive to this conditioning problem.

The main disadvantage of this approach is that it often suffers from severe conditioning problems, where small changes in $\lambda(t_o)$ lead to very large changes in $x(t_f)$. This can result in the algorithm failing to converge, unless the initial trajectory is close to an extremal trajectory. In addition it is quite sensitive to the effects of numerical rounding. However, because of the quadratic convergence properties of Newton's method it is quite useful for conducting parametric perturbation studies once an extremal solution has been found by some other means, such as the gradient method to be discussed subsequently.

2.1.2 Quasi-Linearization Algorithm

An alternative approach to that discussed was proposed by McGill and Kenneth,[8] and Bellman and Kalaba.[9,10] In their method, instead of iterating on the boundary conditions the procedure now iterates on the entire solution trajectory $y(t)$, $t \in [t_o, t_f]$.

Considering the same problem as defined in (2.7)–(2.9), define a function space operator, $\mathcal{F}$, as

$$\mathcal{F}(y) = \frac{dy}{dt} - F(y, t). \tag{2.11}$$

It can be seen that if $y(t)$ is to satisfy (2.7) it is necessary for $\mathcal{F}(y) = 0$, that is y has to be a root

of the operator $\mathcal{F}$. Applying Newton's method in function space to (2.11), where we let $y_{k+1}(t) = y_k(t) + \delta y_k(t)$, and denote the partial derivative $\partial F(y, t)/\partial y$ by $F_y(y, t)$, yields

$$0 = \mathcal{F}(y_k + \delta y_k) = \frac{dy_k}{dt} - F(y_k, t) + \left(\frac{d}{dt} - F_y(y_k, t)\right)\delta y_k. \tag{2.12}$$

Consequently the linear two-point boundary value problem

$$\frac{d}{dt}\delta y_k - Fy(y_k, t)\delta y_k = F(y_k, t) - \frac{dy_k}{dt}, \tag{2.13}$$

with boundary conditions $\delta x(t_o) = \delta x(t_f) = 0$, must be solved for the iterate $\delta y_k(t)$, for $k = 0, 1, 2, \ldots$. This problem can be easily solved by standard methods.

This approach, while exhibiting quadratic convergence near an extremal solution again suffers with the problem of ill-conditioning,[11] which may lead to the algorithm failing to converge unless the initial trajectory is near an extremal trajectory.

2.2 Direct Methods on Control Function Space

The direct methods of computing the extremal controls for optimal control problems largely overcome the convergence difficulties of the indirect methods. In these methods the cost functional, J, in (2.1), is minimized directly without recourse to the necessary conditions, by iteratively adjusting the control function $u(t)$, $t \in [t_o, t_f]$. These methods, which are well suited to handling problems without terminal constraints, cannot be immediately applied to problems of the type defined by Equations (2.1)–(2.3). Methods of applying the algorithms under consideration to this latter class of problems will be discussed in the next section.

2.2.1 First Order Gradient Methods

Applications of first order gradient methods have been extensively studied,[12–19] because of their relative simplicity on the one hand, and their reliability in solving a broad class of problems with an acceptable speed of convergence on the other.

Consider the case where the terminal constraints (2.4) are not present. The cost functional, J, defined by (2.1) can be considered as mapping the control $u \in L^2[t_o, t_f]$ into R^1. In this case the gradient, $\nabla J(u)$, of (2.1), subject to the differential equation constraints (2.2) can be determined by expanding (2.1) using a truncated Taylor series. Thus if δu denotes a variation in the control u, we obtain

$$J(u + \delta u) = \phi[x(t_f), t_f] + \int_{t_o}^{t_f} L(x, u, t)dt + \phi_x[x(t_f), t_f]\delta x + \int_{t_o}^{t_f} [L_x(x, u, t)\,\delta x(t) + L_u(x, u, t)\delta u(t)]dt \tag{2.14}$$

where $\delta x(t)$ is the variation in the solution of (2.2) caused by $\delta u(t)$. The variation $\delta x(t)$, $t \in [t_o, t_f]$ is the solution of the variational differential equation

$$\frac{d}{dt}\delta x(t) = f_x(x(t), u(t)\,\delta x(t) + f_u(x(t), u(t), t)\delta u(t), \qquad \delta x(t_o) = 0. \tag{2.15}$$

Introducing the adjoint differential equation

$$\frac{d\lambda}{dt}(t) = -f_x{}^T(x(t), u(t), t)\lambda(t) - L_x{}^T(x(t), u(t), t), \qquad \lambda(t_f) = \phi_x{}^T[x(t_f), t_f], \tag{2.16}$$

it can be shown that

$$\phi_x[x(t_f), t_f]\delta x(t) + \int_{t_o}^{t_f} L_x(x, u, t)\delta x(t)dt = \int_{t_o}^{t_f} \lambda^T(t) f_u(x, u, t)\delta u(t)dt. \tag{2.17}$$

Substituting (2.17) into (2.14) it follows that

$$J(u + \delta u) = J(u) + \int_{t_o}^{t_f} [\lambda^T(t) f_u(x, u, t) + L_u(x, u, t)]\delta u(t) dt,$$

and as a consequence the gradient $\nabla J(u)$ can be identified as

$$\nabla J(u)(t) = \lambda^T(t) f_u(x(t), u(t), t) + L_u(x(t), u(t), t), \qquad t \in [t_o, t_f]. \tag{2.18}$$

Assuming that the initial estimate of the control, $u_o(\cdot)$, is given, and using the usual gradient procedure, the $(k + 1)$th iterate of the control, $u_{k+1}(\cdot)$, is given by

$$u_{k+1} = u_k - \alpha_k \nabla J(u_k), \tag{2.19}$$

where $\nabla J(u_k)$ is given by (2.18). Provided that an appropriate procedure is used for selecting the parameters $\{\alpha_k\}_{k=0}^{\infty}$ it can be shown that such an algorithm converges to an extremal solution of the above problem. One such procedure is the method of steepest descent where the parameter α_k is chosen by conducting a one-dimensional search so that

$$J(u_k) - \alpha_k \nabla J(u_k))$$

is minimized. This procedure is very robust because at each iteration step the value of the cost functional, J, must decrease until a limit is reached.

The main advantages of methods using a first order gradient are:

(1) it is simple to program;

(2) it requires first order derivative evaluations only;

(3) global convergence to local minima can be proved so that the algorithm is reliable;

(4) the algorithm coverges rapidly during the initial iterations.

However, these methods exhibit the significant disadvantage of slow convergence once the iterates are in the neighbourhood of an extremal solution. In fact it can be shown that they only have a linear rate of convergence near a solution so that the convergence error only decreases with a geometric progression on the convergence factor.

2.2.2 Second Order Gradient Methods

Because of the poor speed of convergence of first order gradient methods many researchers have studied second order gradient methods.[20,21,22,23,24] Essentially these methods involve expansion of the cost functional (2.1) to include quadratic terms, thus leading to a linear quadratic optimization problem, which has to be solved for the new control iterate.

Again consider the case where the terminal constraints (2.4) are not present. Following Miele[18] a multiplier function $\lambda(t)$, $t \in [t_o, t_f]$ is introduced which satisfies (2.16) and is used to augment the functional (2.1) with the term $\lambda^T(t)(f(x(t), u(t), t) - \frac{dx}{dt}(t))$, thus yielding the new functional

$$\begin{aligned} J(u) &= \phi[x(t_f), t_f] + \int_{t_o}^{t_f} L(x, u, t) + \lambda^T(t)(f(x, u, t) - \frac{dx}{dt}(t))dt, \\ &= \phi[x(t_f), t_f] + \int_{t_o}^{t_f} H(x, u, \lambda, t) - \lambda^T(t)\frac{dx}{dt}(t)dt, \end{aligned} \tag{2.20}$$

where $H(x, u, \lambda, t)$ is defined in (2.5). Expanding (2.20) using a truncated Taylor series including quadratic terms, we obtain

$$\begin{aligned} J(u + \delta u) = J(u) &+ \phi_x(x(t_f), t_f)\delta x(t_f) + \int_{t_o}^{t_f} [H_u(x, u, \lambda, t)\delta u(t) + H_x(x, u, \lambda, t)\delta x(t) \\ &- \lambda^T(t)\frac{d}{dt}\delta x(t)]dt + \tfrac{1}{2}\delta x(t_f)^T \phi_{xx}(x(t_f), t_f)\delta x(t_f) \\ &+ \tfrac{1}{2}\int_{t_o}^{t_f} [\delta x^T(t) \delta u^T(t)] \begin{bmatrix} H_{xx}(x, u, \lambda, t) & H_{xu}(x, u, \lambda, t) \\ H_{ux}(x, u, \lambda, t) & H_{uu}(x, u, \lambda, t) \end{bmatrix} \begin{bmatrix} \delta x(t) \\ \delta u(t) \end{bmatrix} dt, \end{aligned} \tag{2.21}$$

where $\delta x(t)$, $t \in [t_o, t_f]$ is the solution of (2.15). It follows from (2.17) that (2.21) becomes

$$J(u + \delta u) = J(u) + \int_{t_o}^{t_f} H_u \delta u dt + \tfrac{1}{2} \int_{t_o}^{t_f} [\delta x^T \delta u^T] \begin{bmatrix} H_{xx} & H_{xu} \\ H_{ux} & H_{uu} \end{bmatrix} \begin{bmatrix} \delta x \\ \delta u \end{bmatrix} dt$$
$$+ \tfrac{1}{2}[\delta x^T \phi_{xx} \delta x]_{t=t_f}. \tag{2.22}$$

Applying the Newton-Raphson method to (2.22) the problem becomes one of finding a $\delta u_i(t)$, $t \in [t_o, t_f]$ which minimizes the quadratic cost functional (2.22) subject to the linear variational differential equation (2.15). Once δu_i is computed, the next control iterate u_{i+1} is determined from $u_{i+1} = u_i + \delta u_i$, and the process then repeated in a manner similar to the first order gradient procedure.

The necessary conditions for optimality when applied to (2.22) and (2.15) yield the following two point boundary value problem

$$\begin{bmatrix} \frac{d}{dt}\delta x \\ \frac{d}{dt}\delta \lambda \end{bmatrix} = \begin{bmatrix} A(t) & B(t) \\ C(t) & A^T(t) \end{bmatrix} \begin{bmatrix} \delta x \\ \delta \lambda \end{bmatrix} + \begin{bmatrix} r(t) \\ s(t) \end{bmatrix} \quad \begin{matrix} \delta x(t_o) = 0 \\ , \\ \delta \lambda(t_f) = \phi_{xx}(x(t_f), t(f)\delta x(t_f)) \end{matrix} \tag{2.23}$$

where

$$A(t) = f_x - f_u H_{uu}^{-1} H_{ux}, \tag{2.24}$$
$$B(t) = f_u H_{uu}^{-1} f_u^T, \tag{2.25}$$
$$C(t) = H_{xx} - H_{xx} H_{uu}^{-1} H_{ux}, \tag{2.26}$$
$$r(t) = f_u H_{uu}^{-1} f_u^T, \tag{2.27}$$

and

$$s(t) = H_{xu} H_{uu}^{1} H_u^T. \tag{2.28}$$

The functions in equations (2.24)–(2.28) are assumed to be evaluate along $x(t)$, and $u(t)$, for $t \in [t_o, t_f]$.

Two main approaches have been proposed for solving the above two-point boundary problem. Breakwell[20] and Kelley[21,22], by determining a set of n linearly-independent solutions to the $2n$ differential equations (2.23) such that each solution satisfies the final boundary conditions, have used the principle of super-position to find a solution which also satisfies the initial boundary conditions. Unfortunately this method can suffer from severe ill-conditioning problems; a fact which led to the development of the sweep methods proposed by Jacobson,[22,25] McReynolds[26,27] and Mitter[24] which tends to overcome this problem.

The main advantages of this method are—

(1) The step size of δu is automatically determined, so that a one-dimensional search is not required at each algorithm iteration.

(2) The algorithm exhibits quadratic convergence properties near an extremal trajectory, and as a consequence will converge much more rapidly to a final solution than the first order gradient method.

However, there are two disadvantages which limit its usefulness. These are—

(1) The algorithm may not always converge, particularly if the initial trajectory is far from the extremal trajectory. In fact for this algorithm to be applicable it is necessary for H_{uu} to be positive definite for $t \in [t_o, t_f]$, which is difficult to ensure unless the initial trajectory is near a minimal trajectory. As a consequence it is often necessary to initially use an alternative algorithm, such as the first order gradient method, as a preliminary to its use.

(2) Programming and problem preparation requires much greater effort compared with first order gradient methods, because of the large number of second derivatives which need to be evaluated.

2.2.3 Conjugate Gradient Methods

The application of the conjugate gradient algorithm to optimal control problems was

proposed by Sinnot and Luenberger[28], Lasdon, Mitter and Waren,[29] and Pagurek and Woodside[30] about 1967. More recently Hestenes[31] has discussed its use with his "Method of Multipliers" for solving constrained problems, while Miele[32] has considered its use in conjunction with his method of constraint restoration.

The method is essentially a generalization of the conjugate gradient algorithm of Fletcher and Reeves[33] to function space problems. Supposing the gradient $\nabla J(u)$ of the control u, denoted by $g(u)$ is computed as in Section 2.2.1 above, and that the initial estimate of the control, u_o, is chosen arbitrarily. The algorithm proceeds as follows:

(1) let $g_o = g(u_o)$ and set $d_o = -g_o$;

(2) find $\alpha = \alpha_i$ such that $J(u_i + \alpha d_i)$ is minimized;

(3) set $u_{i+1} = u_i + \alpha_i d_i$;

(4) find $g_{i+1} = g(u_{i+1})$, and $\beta_i = (g_{i+1}, g_{i+1})/(g_i, g_i)$, where (g_i, g_j) is defined by

$$(g_i, g_j) = \int_{t_o}^{t_f} g_i^T(t) g_j(t) dt;$$

(5) set $d_{i+1} = -g_{i+1} + \beta_i d_i$;

(6) Return to step (2) until algorithm has converged to problem solution.

It has been shown in Reference 29 that not only does this algorithm exhibit many of the theoretical properties of the finite dimensional conjugate gradient algorithm, but numerical experience shows that it converges more rapidly than first order gradient algorithms.

The principal advantages of this method are:

(*a*) The algorithm only requires the evaluation of first order derivatives;

(*b*) It appears to exhibit quadratic convergence near an extremal trajectory, although some doubt about this is expressed in Reference 11.

(*c*) Programming effort required is only a moderate increase over the first order methods.

The main disadvantages are—

(*a*) A significant amount of computation time is required to evaluate the conjugate directions in function space.

(*b*) The algorithm is quite sensitive to numerical rounding, which means that some computations need to be performed in double precision.

(*c*) A one-dimensional search is required at each iteration, which can consume a significant amount of computation time.

(*d*) The algorithm may fail to converge if the initial estimate of the trajectory is too far from an extremal trajectory. As a consequence a first order gradient method may be needed to obtain a starting solution for the algorithm.

2.2.4 Variable Metric Methods

Variable metric methods were initially developed for minimization of functions of a finite number of variables. The best known of these being the Davidon algorithm, was subsequently refined by Fletcher and Powell[34] and has become known as the Davidon-Fletcher-Powell (DFP) method. The application of this method to function space minimization problems is discussed in References 35, 36, 31, 37. Although only a limited amount of computational experience with the DFP method has been reported,[37,38] this work shows that its speed of convergence is superior to the conjugate gradient method.

The Davidon-Fletcher-Powell Method

Suppose the gradient $\nabla J(u)$, denoted by $g(u)$, is computed as in Section 2.2.1, and the initial estimate of the control, u_o, is chosen arbitrarily. The algorithm proceeds as follows:

(1) let $g_o = g(u_o)$ and set $d_o = -g_o$;

(2) find $\alpha = \alpha_i$ such that $J(u_i + \alpha d_i)$ is minimized;

(3) set $u_{i+1} = u_i + \alpha_i d_i$ and find $g_{i+1} = g(u_{i+1})$;

(4) find $s_i = u_{i+1} - u_i$ and $y_i = g_{i+1} - g_i$;

(5) find

$$H_i y_i = \begin{cases} y_o, & i = 0, \\ y_i + \sum_{j=0}^{i-1} \left[\dfrac{(s_j, y_i)}{(s_j, y_i)} s_j - \dfrac{(H_j y_i, y_i)}{(H_j y_j, y_i)} H_j y_j, \right], & i > 0; \end{cases}$$

(6) find

$$d_{i+1} = -H_{i+1} g_{i+1} = -\left\{ g_{i+1} + \sum_{j=0}^{i} \left[\frac{(s_j, g_{i+1})}{(s_j, y_j)} s_j + \frac{(H_j y_j, g_{i+1})}{(H_j y_j, y_i)} H_j y_j \right] \right\}$$

(7) return to step (2) k times;

(8) return to step (1) until algorithm has converged to problem solution.

In the above algorithm the inner product (x, y) is defined as

$$(x, y) = \int_{t_o}^{t_f} x^T(t) y(t) dt.$$

Steps (7) and (8) in the algorithm, where it is re-started after every k iterations, have been introduced to overcome the difficulties of excessive storage requirements and the large increase in computation time per iteration as k increases. In fact to compute d_{i+1} it is necessary to store $2(i + 2)$ functions. The experience of Tripathi[37] indicates that k should lie in the range 5–12, while Pierson[38] has found evidence that selecting k to be 3 or 4 actually enhances the convergence rate.

Apart from the DFP method discussed above, other variable metric algorithms whose application to the computation of optimal controls has been studied are the Davidon "Rank-One" method by Garg[39] and the Broyden quasi-Newton algorithm by Edge and Powers.[40] Both of these methods have similar memory storage requirements to the DFP method.

The "rank-one" method, discussed by Garg, has a unique feature of not requiring a one dimensional search at each iteration. Since a considerable amount of computation time is often consumed in these searches, there is a possibility of speedier convergence when this method is applied to optimal control problems. His experience, albeit on a simple problem, indicates that this may be the case.

Edge and Power's study on the Space Shuttle ascent trajectory optimization is the only example, of which the author is aware, where one of these methods has been applied to a realistic aerospace problem. Their experience shows that the Broyden algorithm (which is closely related to the DFP method) can be successfully used for studying these types of problems although they make no attempt to compare its efficiency with other types of algorithms.

Since only limited experience of the use of these methods has been reported our conclusions are to some extent tentative. However, reported experience with the finite dimensional algorithm plus analytical results leads to the following observations—

(1) Only first order derivatives are required.

(2) Algorithm exhibits quadratic convergence near an extremal trajectory. Experience with finite dimensional algorithms indicates that this method converges more reliably than the conjugate gradient methods.

(3) Analytical studies show that the method starts like a first order gradient method and gradually becomes like a Newton method.

(4) Experience with finite dimensional problems shows that it is less sensitive to numerical rounding than the conjugate gradient method.

(5) In References 37, 38, 39 it is reported that these methods converge more rapidly than other comparable methods.

Their most significant disadvantages are:

(1) They require considerable memory storage when compared with first order gradient and conjugate gradient methods.

(2) A considerable amount of computation time must be used in computing the variable metric operator.

2.2.5 Balakrishnan Epsilon Method

This method, proposed by Balakrishnan,[41] differs from the direct methods discussed above, in respect of the state variable $x(t)$ which is here treated as an independent variable rather than as being causally dependent on the control $u(t)$ through the differential equation (2.2). To ensure Equation (2.2) is satisfied he proposed that it be adjoined to the cost functional (2.1) by use of a penalty function. Thus (2.1) becomes

$$J_\epsilon(x, u) = \phi[x(t_f), t_f] + \int_{t_o}^{t_f} L(x, u, t)dt + \frac{1}{\epsilon}\int_{t_o}^{t_f} \| \frac{dx}{dt}(t) - f(x, u, t)\|^2 dt, \tag{2.29}$$

where $\|\cdot\|$ denotes the Euclidean norm.

By taking a sequence of ϵ_i, $i = 1, 2, \ldots$, where $\epsilon_i \to 0$ as $i \to \infty$, and minimizing $J_{\epsilon i}$ for each $i = 1, 2, \ldots$ it can be shown that the trajectory and control approach a solution of the problem defined by (2.1)–(2.3). It has been found[42,43] that the convergence of this method is not particularly sensitive to the value of ϵ, and as a consequence only a small number of values of ϵ need to be chosen for practical application.

The technique used for minimizing (2.29) in References 42 and 43 is to transform this functional using the Rayleigh-Ritz method, and then solve the resulting finite dimensional minimization problem using the Newton-Raphson algorithm. Their experience seems to indicate that it is a method worthy of further investigation when multiple state space and control constraints are present. Taylor and Constantinides[44] found the indirect relationship between the error in the satisfaction of (2.2) and the error in the satisfaction of the terminal constraints (2.3) made it difficult to gain an insight into the convergence behaviour of the algorithm. This proves to be a significant disadvantage.

2.3 Discussion

It can be observed from the preceding sections that no single method exhibits all the characteristics of an ideal algorithm. As a consequence it is useful to examine them from the viewpoint of how the strengths of one algorithm may be used to complement the weaknesses of others.

Since the first order gradient methods reliably converge to extremal trajectories, even though they have poor terminal convergence behaviour, it appears that they are well suited for computing the starting trajectories for methods which are not globally convergent, but have rapid terminal convergence. In this case no attempt would be made to obtain complete convergence using a first order gradient method; instead it would be used to execute a small number of iterations before transferring to a more rapidly convergent algorithm.

Of the techniques discussed, the variable metric methods seem to offer the greatest potential for general purpose use. Computational experience, particularly with finite dimensional problems, has shown them to be globally convergent for a broad class of problems, even though this fact has not been universally proved by analytical means. In addition these methods exhibit excellent terminal convergence behaviour. Experience in use has also shown them to have a superior convergence rate, and to be less sensitive to numerical rounding errors, than the conjugate gradient methods. One significant disadvantage of the variable metric methods, and something which is not apparent for the finite dimensional case, is the large memory storage requirements when they are applied to function space problems. In cases where memory storage is at a premium it may be necessary to use second order gradient or conjugate gradient methods as their memory requirements are quite modest.

3. ALGORITHMS FOR PROBLEMS WITH TERMINAL CONSTRAINTS

In Sections 2.2.1–2.2.4 methods for solving unconstrained optimization problems of the type defined by equations (2.1) and (2.2), but without the presence of the terminal constraints (2.3), were examined. In this section a number of techniques are discussed, which can be used in conjunction with the above methods for solving problems with terminal constraints.

3.1 Penalty Function Methods

In the penalty function method discussed by Kelley,[14] an optimization problem with terminal constraints is transformed into one without constraints. In applying this method to the problem defined by (2.1)–(2.3) the terminal constraints (2.3) are ignored. However, to ensure that the constraints (2.3) are satisfied a new cost functional, $J_K(u)$, is defined by adjoining a term proportional to the magnitude of the constraint error, $\psi[x(t_f), t_f]$, to the cost functional (2.1), thus giving

$$J_K(u) = \phi[x(t_f), t_f] + \int_{t_o}^{t_f} L(x, u, t)dt + \tfrac{1}{2}\psi^T[x(t_f), t_f]K\psi[x(t_f), t_f], \tag{3.1}$$

where K is a positive definite diagonal penalty function weighting matrix.

By taking a sequence of matrices K_i, $i = 1, 2, \ldots$, where $\|K_i\| \to \infty$ as $i \to \infty$, and minimizing (3.1) for each K_i, subject to (2.2) using one of the methods given in Section 2.2, it can be shown that the corresponding sequence of trajectories and controls converge to a solution of the problem defined by (2.1)–(2.3). Schemes for adjusting the elements of the weighting matrix K after each cycle of the algorithm are discussed by Kelley,[14] and Moyer and Pinkham.[16]

Experience with this method[16,45,47] has shown that it can lead to failure of convergence when using an optimization algorithm which is otherwise reliable. The difficulty appears to partially arise from rounding errors, as the magnitude of the elements of matrix K approach infinity, causing the penalty function to dominate the original cost function. Also for finite dimensional problems, where penalty functions are used, it is well known that when the elements of K are large the augmented cost functional often has long narrow ravines, which lead to slow convergence to the final solution. Similar difficulties appear to manifest themselves for function space problems.

In spite of these difficulties the ease of application of the penalty function method has meant that it has been widely used[16,38,29,21,14,17,46,40] with many of the computational algorithms discussed above.

3.2 Shifting Boundary Method

Moyer and Pinkham[16] proposed a slight variation of the penalty function method using the idea of successive approximation which they found gave more reliable results. In this method the penalty function weighting matrix K is held constant while the cost functional

$$J_K(u) = \phi[x(t_f), t_f] + \int_{t_o}^{t_f} L(x, u, t)dt + \tfrac{1}{2}(\psi[x(t_f), t_f] + c)^T K(\psi[x(t_f), t_f] + c) \tag{3.2}$$

is minimized subject to (2.2), where the q-vector c is introduced to shift the terminal constraints. The ith iteration of this vector, denoted c_i, is defined recursively by the relation

$$c_i = c_{i-1} - \psi[x_{i-1}(t_f), t_f]. \tag{3.3}$$

Noting that $x_{i-1}(t_f)$ is a function of c_{i-1} it follows that the algorithm will converge providing that the term $\psi[x_{i-1}(t_f), t_f]$ defines a contraction mapping. This method does not appear to have been extensively studied in the literature, since Moyer's early work, although it has some similarities to the gradient projection method to be discussed below.

3.3 The Augmented Penalty Function Method

An alternative approach for improving the penalty function method, called variously the Method of Multipliers and the Method of Augmented Penalty Functions, has been presented by Hestenes.[47] A number of variants of this method have also been proposed by Tripathi and Narendra,[48] O'Doherty and Pierson,[49] Connor and Vlach,[50] and Connor and Saltavareas.[51]

In the method proposed by Hestenes the cost functional (2.1) is augmented with a quadratic and a linear penalty function of the terminal constraint functions (2.3), thus yielding

$$W(u, \lambda, K) = \phi[x(t_f), t_f] + \int_{t_o}^{t_f} L(x, u, t)dt + \lambda^T\psi[x(t_f),t_f] + \tfrac{1}{2}\psi^T[x(t_f), t_f]K\psi[x(t_f), t_f], \quad (3.4)$$

where K is a diagonal positive definite penalty function weighting matrix and λ is a q-vector multiplier which is to be determined. The matrix K is chosen and held fixed throughout the computation. The method proceeds by selecting a λ_o and then minimizing $W(u, \lambda_o, K)$ with respect to u, thus giving u_o. In general, given the multiplier λ_n the next estimate is determined from

$$\lambda_{n+1} = \lambda_n + K\psi[x_n(t_f), t_f], \quad (3.5)$$

after which u_{n+1} is determined by minimizing $W(u, \lambda_{n+1}, K)$.

Since the problem defined by the augmented cost functional (3.4) and the differential equation (2.2) has no terminal constraints the optimization algorithms discussed in Section 2.2 can be used for minimizing $W(u, \lambda, K)$. From the limited amount of experience reported it appears to be a reliable method. Connor and Saltavareas[51] claim that their variant of the basic algorithm appears to give superior computational performance to all the others of this class which they have tested.

3.4 Gradient Projection Technique

The application of the gradient projection technique to optimal control problems with terminal constraints was proposed by Kelley,[14] and Bryson and Denham.[15] More recently Leese[52] has presented a generalization of this method which is also applicable to optimization problems of the type defined by (2.1)–(2.3). Experience with this method has shown that it functions well for problems with linear constraints but is often unreliable when the constraints are non-linear. Recent work on improving its performance when used with non-linear constraints has been carried out by Kelley and Speyer,[53] Kelley, Lefton and Johnson,[54] and Rosen and Kreuser.[55]

The essential idea of the gradient projection method is to minimize the variation $\delta J = J(u + \delta u) - J(u)$ of the cost functional (2.1), due to a variation δu of the control function. This minimization is carried out subject to satisfying (2.2) and (2.3) to first order, and in addition satisfying a quadratic integral constraint on the control variation δu. The latter condition is introduced to ensure that the problem has a meaningful solution. If it is assumed that the state trajectory $x(t)$, $t \in [t_o, t_f]$ satisfies the boundary conditions a control iterate $\delta u(t)$, $t \in [t_o, t_f]$ is sought so as to minimize

$$\delta J = \int_{t_o}^{t_f} [\lambda^T(t)fu(x, u, t) + L_u(x, u, t)]\delta u(t)dt, \quad (3.6)$$

subject to

$$\frac{d}{dt}\delta x(t) = f_x(x(t), u(t), t)\delta x(t) + f_u(x(t), u(t), t)\delta u(t), \qquad \delta x(t_o) = 0, \quad (3.7)$$

$$\psi_x[x(t_f), t_f]\delta x(t_f) = 0, \quad (3.8)$$

and

$$\tfrac{1}{2}\int_{t_o}^{t_f} \delta_u{}^T(t)\delta u(t)dt = 1, \quad (3.9)$$

where $\lambda(t)$ is the solution of (2.16). This is a standard linear quadratic optimization problem whose solution is presented in Reference 6.

After each iteration of the control function it is necessary, if the terminal constraint function $\psi[x(t_f), t_f]$ is non-linear, to apply a constraint restoration procedure. Bryson and Ho[6] proposed the simple approach of combining the restoration procedure with the basic gradient algorithm using the method of successive approximation. In this case (3.8) is replaced by

$$\psi_x[x(t_f), t_f]\delta x(t_f) = -\epsilon\psi[x(t_f), t_f], \quad (3.10)$$

where the parameter $\epsilon \in (0, 1]$. This method of constraint restoration should be compared with the shifting boundary method discussed above, and also the combined gradient-restoration algorithm of Miele.[18]

More recently Kelley and Speyer[53] have presented a gradient projection version of the Davidon-Fletcher-Powell algorithm, which has been found to give good performance for finite dimensional optimization problems, when the constraints are linear. In order to improve the performance for non-linearly constrained optimization problems Kelley *et al.*[54] have introduced the curvilinear projection version of the Davidon method which appears to give a further improvement in performance over the previous methods.

3.5 Constraint Restoration Methods

The need for a constraint restoration procedure has been previously mentioned in relation to the use of the gradient projection method with non-linear constraints. The work of Miele[18,32] and his collaborators has led to a family of optimization-restoration algorithms which ensure constraint satisfaction. Moyer[45] has also proposed an algorithm which combines an optimization phase with a constraint restoration phase. In his approach, instead of the cost functional being minimized it is treated as an additional terminal constraint.

The Miele Algorithm

Each cycle of the sequential gradient-restoration algorithm of Miele[18] consists of two phases. Supposing that the terminal constraints are satisfied at the beginning of a cycle, then in the first phase one step of an unconstrained optimization procedure, such as described in Section 2.2, is used to decrease the cost functional (2.1). In Miele's work he has concentrated on the use of the first order gradient methods for this purpose. Since in this first phase no account is taken of the terminal constraint requirements it is likely that condition (2.3) is violated. The second phase consists of adjusting the control determined in phase one so that the terminal constraints are satisfied by minimizing the cost functional

$$C(u) = \tfrac{1}{2}\psi^T[x(t_f), t_f]\psi[x(t_f), t_f], \tag{3.11}$$

subject to the differential equation (2.2). After this phase is completed a new algorithm cycle begins. As the cost functional (2.1) does not appear in (3.11) explicitly, care needs to be taken, by adjusting the iteration step-size in phase one, to ensure that the algorithm will converge to a solution of the problem defined by (2.1)–(2.3). Miele's experience has shown that this can easily be done.

A large number of variations of this basic algorithm are described by Miele in Reference 32, where an extensive bibliography to this work is given.

The Moyer Algorithm

The algorithm described by Moyer[45] can be best illustrated by examining the problem defined by (2.1)–(2.3), where it is assumed that $L(x, u, t) \equiv 0$, so that the cost functional, denoted by $J^A(u)$, becomes

$$J^A(u) = \phi[x(t_f), t_f]. \tag{3.12}$$

In addition it will be assumed that the terminal constraints have the simple form

$$x_i(t_f) = x_{if}, \qquad i = 1, \ldots, q \leqslant n - 1, \tag{3.13}$$

which will be denoted in vector form by $\bar{x}(t_f) = \bar{x}_f$.

In the first phase the problem defined by (3.12), (2.2) and (2.3) is solved using a penalty function approach, where the penalty function weighting matrix K, in (2.30), is held fixed. This yields an estimate of the optimal value of the cost functional, which will be denoted by J_m^A. In the second phase of the algorithm, where the terminal constraints are restored, the cost functional (3.12) is treated as an additional terminal constraint by setting

$$\phi[x(t_f), t_f] = J_m^A, \tag{3.14}$$

and then seeking to minimize the functional

$$J^C(u) = \tfrac{1}{2}\{\phi[x(t_f), t_f] - J_m^A)^2 + (\bar{x}(t_f) - \bar{x}_f)^T K(\bar{x}(t_f) - \bar{x}_f)\}, \tag{3.15}$$

subject to (2.2). To improve the estimate of J_m^A an iterative scheme is used, where the $(i+1)$th iteration, $J_{m,i+1}^A$, is defined recursively by the relation

$$J_{m,i+1}^A = J^A(u_c) + \frac{1}{J^A(u_c) - J_{m,i}^A} (\bar{x}_c(t_f) - \bar{x}_f)^T K(\bar{x}_c(t_f) - \bar{x}_f). \tag{3.16}$$

The subscripted variables u_c and $\bar{x}_c$ indicate that they are obtained from the minimization of (3.15).

A disturbing feature of the Moyer algorithm is that it will only converge to an extremal solution of the problem defined by (3.12), (2.2) and (2.3) if the estimated optimal cost J_m^A is less than the true value of the cost for this problem. If this is not so then the algorithm will converge to a non-extremal solution.

4. CONCLUDING REMARKS

An examination of methods for computing the extremal solutions of optimal control problems, which are suitable for handling typical flight mechanics problems, has led to the conclusion that a two stage computational procedure should be used. This is necessary because at the present time no single algorithm exhibits the desirable features of rapid convergence in the neighbourhood of an optimum solution on the one hand, and reliable convergence from a starting solution which is not necessarily close to the optimum on the other. Experience has shown the Bryson first order gradient projection method to converge reliably from an arbitrary starting solution, and to have a high initial rate of convergence. Thus it is well suited for computing the initial solution estimate in a two stage procedure. This algorithm has been implemented and will be the subject of a subsequent report.

For the second stage an algorithm exhibiting a high rate of terminal convergence is desired. Of the higher order algorithms variable metric methods are preferred over the Newton and conjugate gradient procedures because of their greater reliability and speed of convergence.

Arising from this work the author has become aware of two approaches to algorithms for optimal control problems which are worthy of further consideration. The first relates to the Moyer algorithm. Possibilities exist to improve the performance of this algorithm by using variable metric methods for cost functional minimization, and improved techniques for adjusting the cost functional estimate. In addition the applicability of this class of algorithms to optimal control problems involving state space and terminal inequality constraints needs to be investigated.

The second approach is of a more fundamental nature relating to methods of generating optimal algorithms for classes of problems. These ideas arose from a consideration of the form taken by the iteration relations (2.18) for gradient methods. The steepest descent method can be considered as a one-step optimization where the parameter α_k is chosen so that $J[u_k - \alpha_k \nabla J(u_k)]$ is minimized. A development of this notion would be to choose simultaneously the parameters $\alpha_k, \ldots \alpha_{k+t}$ so as to minimize $J[u_k - \alpha_k \nabla J(u_k) - \ldots - \alpha_{k+t} \nabla J(u_{k+l})]$ using a multi-stage optimization procedure. It is conjectured that this will yield an improved speed of convergence over the steepest descent method. Whether this will be achieved, or achieved at excessive "computational cost" remains to be determined. The concept is easily generalized to variable metric methods, and so is capable of generating whole families of new algorithms.

REFERENCES

1. Dayton, A. D., "The Digital Airplane and Optimal Aircraft Guidance", *Advances in Control Systems*, AGARD Conf. Proc. No. 137 (Sept. 1973), pp. 3-1, 3-14.

2. Palmer, E. A., Cronn, F. W., "Head-Up Displays for STOL Visual Approaches", STOL Technology, *NASA Conf.* at A.R.C. (Oct. 1972), pp. 227-43.

3. Cloud, E. L., *et al.*, "Combat Energy Maneuverability Display—CEMD", *Second Adv. Aircrew Display Symp.*, Dept. Nav., Nav. Air Test. Center (April 1975), pp. 314-32.

4. Walter, J. F., Hoover, G. W., "The Maneuvering Flight Path Display", *Third Adv. Aircrew Display Symp.*, Dept. of Navy, Nav. Air Test Center (May 1976), pp. 215-29.

5. Klopstein, C., "Rational Study of Aircraft Piloting", *Third Adv. Aircrew Display Symp.*, Dept. of Navy, Nav. Air Test Center (May 1976), pp. 232-48.

6. Bryson, A. E., Ho, Y. C., *Applied Optimal Control*, Ginn and Co., Mass., 1969.

7. Paiewonsky, B. H., "A Study of Time Optimal Control", *Proc. of Int. Symp. on Non-linear Diff. Equns. and Non-linear Mech.*, (ed.) J. P. Lasalle and S. Lefschetz, Academic Press, New York (1963), pp. 333-65.

8. McGill, R., Kenneth, P., "A Convergence Theorem on the Iterative Solution of Nonlinear Two-Point Boundary Value Systems", *Proc. XIVth Int. Astronautical Congress* (Paris, France, Sept. 1963), vol. 4, Gauthier-Villars, Paris, 1965, pp. 173-88.

9. Bellman, R., Kalaba, R., "Dynamic Programming, Invariant Imbedding and Quasilinearization: Comparisons and Interconnections", in *Computing Methods in Optimization Problems*, (ed.) A. V. Balakrishnan and L. W. Neustadt, Academic Press, New York, 1964.

10. Bellman, R., Kalaba, R., *Quasilinearization and Boundary Value Problems*, American Elsevier. New York, 1965.

11. Polak, E. ,"An Historical Survey of Computational Methods in Optimal Control", *SIAM Review.*, 15, 2 (April 1973), pp. 553-84.

12. Breakwell, G. V., "The Optimization of Trajectories", *J. Soc. Indust. Appl. Math.*, 7 (1959), pp. 215-47.

13. Kelley, H. J., "Gradient Theory of Optimal Flight Paths", *Amer. Rocket Soc. J.*, 30 (1960), pp. 948-54.

14. Kelley, H. J., "Method of Gradients", in *Optimization Techniques*, (ed.) G. Leitman, Academic Press, N.Y., 1962, Chap. 6, pp. 204-54.

15. Bryson, A. E., Denham, W. F., "A Steepest Ascent Method for Solving Optimum Programming Problems", *Trans. ASME, Ser. E*, 29 (1962), pp. 247-57.

16. Moyer, H. G., Pinkham, G., "Several Trajectory Optimization Techniques Part II: Application", in *Computing Methods in Optimization Problems*, (eds.) A. V. Balakrishnan, L. W. Neustadt, Academic Press, N.Y., 1964.

17. Kopp, R. E., McGill, R., "Several Trajectory Optimization Techniques Part I: Discussion", in *Computing Methods in Optimization Problems*, (eds.) A. V. Balakrishnan, L. W. Neustadt, Academic Press, N.Y., 1964, pp. 65-89

18. Miele, A., "Gradient Methods in Optimal Control Theory", in *Optimization and Design*, (ed.) M. Avriel *et al.*, Prentice Hall, N.J., 1973.

19. Mayne, D. G., Polak, E., "First Order Strong Variation Algorithms for Optimal Control", *J. Opt. Th. and Appl.*, 16, 3/4 (1975).

20. Breakwell, J. V., Speyer, J. L., Bryson, A. E., "Optimization and Control of Non-linear Systems Using the Second Variation", *SIAM J. on Cootrol*, 1, 2 (1963).

21. Kelley, H. G., Kopp, R., Moyer, H. G., "A Trajectory Optimization Technique Based Upon the Theory of the Second Variation", *Prog. in Astronautics and Aeronautics*, Vol. 14, Academic Press N.Y., 1964, pp. 559–82.

22. Jacobson, D. H., "Second Order and Second Variation Methods for Determining Optimal Control: A Comparative Analysis Using Differential Dynamic Programming", *Int. J. Control*, 7, 2 (1968), pp. 175–96.

23. Kelley, H. G., Uzzell, B. R., McKay, S. S., "Rocket Trajectory Optimization by a Second Order Numerical Technique", *AIAA J.*, 7 (May 1969), pp. 879–84.

24. Mitter, S. K., "Successive Approximation Methods for the Solution of Optimal Control Problems", *Automatica*, 3 (1966), pp. 135–49.

25. Jacobson, D. H., "New Second Order and First Order Algorithms for Determining Optimal Control: A Differential Dynamic Programming Approach", *J. Opt. Theory and Appl.*, 2, 6 (1968), pp. 411–40.

26. McReynolds, S. R., "The Successive Sweep Method and Dynamic Programming", *J. Math. and Appl.* 19 (1967), pp. 565–98.

27. McReynolds, S. R., "A Successive Sweep Method for Solving Optimal Programming Problems", *Proc. J.A.C.C.*, Troy, N.Y., 1965, p. 551.

28. Sinnot, G. F., Luenberger, D. G., "Solution of Optimal Control Problems by the Method of Conjugate Gradients", *Proc. J.A.C.C.*, Philadelphia, Penn., 1967, pp. 566–573.

29. Lasdon, R. S., Mitter, S. K., Waren, A. D., "The Conjugate Gradient Method for Optimal Control Problems", *IEEE Trans. Aut. Control*, AC-12, 2 (Feb. 1967), pp. 132–38.

30. Pagurek, B., Woodside, C. M., "The Conjugate Gradient Method for Optimal Control Problems with Bounded Control Variables", *Automatica*, 4, 5–6 (1968).

31. Hestenes, M. R., "Multiplier and Gradient Methods", from *Computing Methods in Optimization Problems*, (eds.) L. A. Zadeh, L. W. Neustadt, A. V. Balakrishnan, Academic Press, N.Y., 1969, pp. 143–63.

32. Miele, A., "Recent Advances in Gradient Algorithms for Optimal Control Problems", *J. Opt. Th. and Appl.*, 17, 5/6 (1975), pp. 361–430.

33. Fletcher, R., Reeves, C. M., "Functional Minimization by Conjugate Gradients, *Comput. J.*, 7 (1964), pp. 10–26.

34. Fletcher, R., Powell, M. J. D., "A Rapidly Convergent Descent Method for Minimization", *Comput. J.*, 6 (1963), pp. 149–54.

35. Horwitz, L. B., Sarachik, P. E., "Davidon's Method in Hilbert Space", *SIAM J. on App. Math.*, 16, 4 (1968), pp. 676–695.

36. Lasdon, L. S., "Conjugate Direction Methods for Optimal Control", *IEEE Trans. Aut. Cont.*, AC-15, 2 (1970), pp. 267–68.

37. Tripathi, S. S., Narendra, K. S., "Optimization Using Conjugate Gradient Methods", *IEEE Trans. Aut. Cont.*, AC-15, 2 (1970) pp. 268–70.

38. Pierson, B. L., Rajtora, S. G., "Computational Experience with the Davidon Method Applied to Optimal Control Problems", *IEEE Trans. Syst. Sci. & Cyb.*, SSC-6 (July 1970), pp. 240–42.

39. Garg, S., "A Rank-One" Method for Optimal Control Problems", *IEEE Trans. Aut. Cont.*, AC-19 (Feb. 1974), pp. 69–71.

40. Edge, E. R., Powers, W. F., "Shuttle Ascent Trajectory Optimization with Function Space Quasi-Newton Techniques", *AIAA J.*, 14 (Oct. 1976), pp. 1369–76.

41. Balakrishnan, A. V., "On a New Computing Technique in Optimal Control and its Application to Minimal Time Flight Profile Optimization", *J. Opt. Th. and Appl.*, 4, 1 (1969).

42. Hewett, M. D., Kirk, D. E., "Trajectory Optimization Using a Second-Order Epsilon Method", *J. Aircraft*, 13 3 (Mar. 1976), pp. 174–79.

43. Taylor, L. W., Smith, H. J., Iliff, K. W., "Experience Using Balakrishnan's Epsilon Technique to Compute Optimum Flight Profiles", *J. Aircraft*, 7, 2 (Mar. 1970), pp. 182–87.

44. Taylor, J. M., Constantinides, C. T., "Optimization: Application of Epsilon Method", *IEEE Trans. Aut. Cont.*, AC-17 (Feb. 1972), pp. 128–31.

45. Moyer, H. G., "Improved Gradient Algorithm for Two-Point Boundary Value Problems", *AIAA J.*, 13, 1 (Jan. 1975), pp. 17–19.

46. Powers, W. F., Shieh, C. J., "Convergence of Gradient-Type Methods for Free Final Time Problems", *AIAA J.*, 14, 11 (Nov. 1976), pp. 1598–603.

47. Hestenes, M. R., "Multiplier and Gradient Methods", from *Computing Methods in Optimization Problems—2*, (ed.) L. A. Zadeh, L. W. Neustadt, A. V. Balakrishnan, Academic Press, N.Y., 1969, pp. 143–63.

48. Tripathi, S. S., Narendra, K. S., "Constrained Optimization Problems Using Multiplier Methods", *J. Opt. Th. and Appl.*, 9, 1 (1972), pp. 59–70.

49. O'Doherty, R. J., Pierson, B. L., "A Numerical Study of Augmented Penalty Function Algorithms for Terminally Constrained Optimal Control Problems", *J. Opt. Th. and Appl.*, 14, 4 (1974), pp. 393–403.

50. Connor, M. A., Vlach, M., "A New Augmented Penalty Function Technique for Optimal Control Functions", *Math. Res. Report 89*, Loughborough Univ. of Technol., Jan. 1976.

51. Connor, M. A., Saltavareas, P., "The Application of Augmented Penalty Functions to Optimal Control Problems", *Math. Res. Report 99*, Loughborough Univ. of Technol., June, 1976.

52. Leese, S. J., "A New Algorithm for Constrained Optimization with Application to Optimal Control", *Math. Res. Report 94*, Loughborough Univ. of Technol., March 1976.

53. Kelley, H. J., Speyer, J. L., "Accelerated Gradient Projection", *Lecture Notes in Mathematics 132*, Springer-Verlag, Berlin, 1970.

54. Kelley, H. J., Lefton, L., Johnson, J. L., "Curvilinear Projection Developments", *J. Guidance and Control*, 1, 1 (Feb. 1978), pp. 80–82.

55. Rosen, J. B., Kreuser, J., "Gradient-Projection Algorithm for Non Linear Constraints", *Conf. on Num. Methods for Nonlin. Optimization*, Univ. of Dundee, Scotland, June 1971.

DOCUMENT CONTROL DATA SHEET

Security classification of this page: Unclassified

1. Document Numbers
   (a) AR Number: AR-001-297
   (b) Document Series and Number: Systems Note 58
   (c) Report Number: ARL-Sys-Note-58

2. Security Classification
   (a) Complete document: Unclassified
   (b) Title in isolation: Unclassified
   (c) Summary in isolation: Unclassified

3. Title: GRADIENT OPTIMIZATION ALGORITHMS FOR SYSTEMS WITH TERMINAL CONSTRAINTS

4. Personal Author(s): Ronald B. Zmood

5. Document Date: August, 1978

6. Type of Report and Period Covered

7. Corporate Author(s): Aeronautical Research Laboratories

8. Reference Numbers
   (a) Task:
   (b) Sponsoring Agency:

9. Cost Code: 71 7900

10. Imprint: Aeronautical Research Laboratories, Melbourne

11. Computer Program(s) (Title(s) and language(s)): UNCOPT.FOR (fortran 10) NAGF (library subroutines)

12. Release Limitations (of the document): Approved for public release

12-0. Overseas:	No.		P.R.	1	A		B		C		D		E	

13. Announcement Limitations (of the information on this page): No limitation

14. Descriptors:
Optimization
Automatic control
Steepest descent method
Constraints
Flight control
Algorithms
Functionals
Numerical analysis
Newton-Raphson method
Differential equations
Calculus of variations

15. Cosati Codes: 1201, 1402

16. *ABSTRACT*

The application of optimization techniques to the derivation of predictive information for flight displays is being investigated in connection with operational situations involving large disturbances and manoeuvres. This note surveys current gradient methods for computing extremal solutions of optimal control problems for systems having boundary conditions but without state or control constraints. It is concluded that a two stage procedure is required, with the first stage using a first order gradient, while the second stage would use a higher order variable metric method. Proposals are advanced for further research on optimisation algorithms.

DISTRIBUTION

Copy No.

AUSTRALIA

DEPARTMENT OF DEFENCE

Central Office

Chief Defence Scientist	1
Executive Controller, ADSS	2
Superintendent, Defence Science Administration	3
Australian Defence Scientific and Technical Representative (UK)	4
Counsellor, Defence Science (USA)	5
Defence Library	6
JIO	7
Assistant Secretary, DISB	8–23

Aeronautical Research Laboratories

Chief Superintendent	24
Superintendent, Systems Division	25
Divisional File, Systems Division	26
Author: Ronald B. Zmood	27
Library	28

Materials Research Laboratories

Library	29

Defence Research Centre

Library	30

Central Studies Establishment Information Centre

Library	31

RAN Research Laboratory

Library	32

Engineering Development Establishment

Library	33

Army Office

Royal Military College	34

Air Office

Aircraft Research and Development Unit	35

Department of Transport

Director General/Library	36

STATUTORY, STATE AUTHORITIES AND INDUSTRY

CSIRO Central Library	37
Qantas, Library	38
Trans Australia Airlines, Library	39
Ansett Airlines of Australia, Library	40

UNIVERSITIES AND COLLEGES

Australian National	Library	41
La Trobe	Library	42