A SURVEY OF METHODS OF FEASIBLE DIRECTIONS
FOR THE SOLUTION OF OPTIMAL CONTROL PROBLEMS

E. Polak

Department of Electrical Engineering and Computer Sciences
and the Electronics Research Laboratory
University of California, Berkeley, California 94720

1. INTRODUCTION

The class of nonlinear programming algorithms known as methods of feasible directions, or as modified methods of centers, is quite large. All the algorithms in this class apply to discrete optimal control problems (see [9]), but only three or four of these can be extended for the solution of continuous optimal control problems. In this paper we shall review three of the most promising methods of feasible directions for optimal control: an extension of the Frank-Wolfe method [5], which is a composite of algorithms proposed by Demyanov [4], Levitin and Polyak [7], Barnes [2] and Armijo [1], a dual method of feasible directions devised by Pironneau and Polak [8], and a Zoutendijk method [13].

From the point of view of feasible directions algorithms, continuous optimal control problems must be divided into four categories: (i) fixed time problems with fixed initial state, free terminal state, and simple constraints on the control; (ii) fixed time problems with inequality constraints on both the initial and the terminal state and no control constraints, (iii) free time problems with inequality constraints on the initial and terminal states and simple constraints on the control, and finally, (iv) fixed time problems with inequality state space constraints and constraints on the control.

We shall show that the above mentioned extension of the Frank-Wolfe method can be used for solving problems in category (i), that the Pironneau-Polak method can be used for solving problems in category (ii), and that the Zoutendijk method can be used for solving discretized problems in category (iv). The Pironneau-Polak method can also be used for solving problems in category (iii). However, this requires a messy modification of the method,

based on a Valentine type transformation. The interested reader will find the details of this in [10].

2. THE NONLINEAR PROGRAMMING ALGORITHMS

The three nonlinear programming algorithms, which are going to adapt for the solution of optimal control problems, were originally intended to solve problems of the form

$$\min\{f^0(z) \mid f^i(z) \le 0, \; j = 1, \; 2, \; \ldots, \; m\}; \tag{2.1}$$

where the f^j: $\mathbb{R}^n \to \mathbb{R}^1$, $j = 0, 1, \ldots, m$, are continuously differentiable.

We begin with a modification of the Frank-Wolfe method [5],[1] which can be used only when the set

$$\Omega \triangleq \{z \mid f^j(z) \le 0, \; j = 1, \; 2, \; \ldots, \; m\} \tag{2.2}$$

is convex. The modification of the Frank-Wolfe algorithm below combines a direction finding subroutine proposed by Levitin and Polyak [5] and by Barnes [2], with an efficient step length subroutine due to Armijo [1]. Such "hybrids" are quite common in nonlinear programming.

Algorithm 2.3 (Modification of Frank-Wolfe Method)

Step 0: Select a continuous, symmetric, positive semi-definite nxn matrix D(z), an $\alpha \in (0,1)$ and a $\beta \in (0,1)$.[2] (Try $\alpha = 0.5$, $\beta = 0.7$).

Step 1: Compute a starting point $z_0 \in \Omega$, as explained in (2.7), below, and set $i = 0$.

Step 2: Compute a point $\bar{z}_i$ as a solution of the problem

$$d^0(z_i) \triangleq \min\{\langle \nabla f^0(z_i), z-z_i \rangle + \langle z-z_i, D(z_i)(z-z_i) \rangle \mid$$

$$f^j(z) \le 0, \; j = 1, \; 2, \; \ldots, \; m\} \tag{2.4}$$

and set $d(z_i) = \bar{z}_i - z_i$.[3]

<u>Step 3</u>: If $d^0(z_i) = 0$, stop; else, compute the smallest integer $k(z_i) \geq 0$ such that

$$f^0(z_i + \beta^{k(z_i)} d(z_i)) - f^0(z_i) - \beta^{k(z_i)} \alpha \langle \nabla f^0(z_i), d(z_i) \rangle \leq 0 \qquad (2.5)$$

<u>Step 4</u>: Set $z_{i+1} = z_i + \beta^{k(z_i)} d(z_i)$.

<u>Step 5</u>: Set $i = i+1$ and go to Step 2. ¤

The function $d^0(\cdot)$ used in algorithm (2.3) has the following proper-ties: (i) $d^0(z) \leq 0$ for all $z \in \Omega$. (ii) Suppose that Ω is convex and that $z_i \in \Omega$ is optimal for (2.1), then $d^0(z_i) = 0$ (i.e. $d^0(z_i) = 0$ is a necessary condition of optimality). This result can be established by reasoning similar to that in Section 4.4 in [9]. (iii) When the set Ω satisfies the Kuhn-Tucker constraint qualification, $d^0(z_i) = 0$, for a $z_i \in \Omega$, if and only if there exist multipliers $\mu^j \geq 0$, $j = 1, 2, \ldots, m$, such that $\nabla f^0(z_i) + \sum_{j=1}^{m} \mu^j \nabla f^j(z_i) = 0$ and $\mu^j f^j(z_i) = 0$, $j = 1, 2, \ldots, m$.

When Ω is convex, algorithm (2.3) does not jam up. Its convergence properties can be summed up as follows (see Sec. 4.3 of [9]).

2.6 <u>Theorem</u>: Suppose that Ω is convex, and compact, and that the sequence $\{z_i\}$ is constructed by algorithm (2.3). If $\{z_i\}$ is finite, then its last element, z_s, satisfies $d^0(z_s) = 0$. If $\{z_i\}$ is infinite, then every accumulation point $\hat{z}$ of $\{z_i\}$ satisfies $d^0(\hat{z}) = 0$. ¤

2.7 <u>Remark</u>: Algorithm (2.3) requires a starting point $z_0 \in \Omega$. Such a point can be computed by applying algorithm (2.3) to the problem, in $\mathbb{R}^{n+1}$,

$$\min\{y^0 | f^j(y) - y^0 \leq 0, j = 1, 2, \ldots, m\}. \qquad (2.7)$$

A starting point (y_0^0, y_0) for solving (2.7) is obtained by taking y_0 to

be a good guess and then setting $y_0^0 = \max_j f^j(y_0)$. When the set $\{z \mid f^j(z) < 0,\ j = 1,\ 2,\ \ldots,\ m\}$ is not empty, after a finite number of iterations, the algorithm will construct a (y_i^0, y_i) such that $f^j(y_i) \le 0$, $j = 1,\ 2,\ \ldots,\ m$, at which point we set $z_0 = y_i$. This is so since the optimal $\hat{y}^0$ is strictly negative.   ¤

For the sake of saving space and so as to exhibit their common features, we state the following two algorithms as one, with a parameter p. When p = 1, the algorithm becomes a composite using the Zoutendijk Procedure 1 [13] direction finding subroutine and the Armijo step size subroutine [1]. When p = 2, the algorithm becomes the Pironneau-Polak modified method of centers [8]. These two algorithms differ both in their direction finding and step length subroutines. Both of these algorithms require that the set $\Omega = \{z \mid f^j(z) < 0,\ j = 1,\ 2,\ \ldots,\ m\}$ be non empty, otherwise they jam up. Convexity of Ω is not required.

Algorithm 2.8 (Zoutendijk Method of Feasible Directions and Pironneau-Polak Modified Method of Centers).

Step 0: Select parameters $\lambda > 0$, $\varepsilon' \in (0, \varepsilon_0]$, $\alpha \in (0,1)$, $\beta \in (0,1)$, $\gamma > 0$. Set p = 1 to obtain Zoutendijk Procedure 1 type method of feasible directions; set p = 2 to obtain Pironneau-Polak modified method of centers. (It is difficult to recommend values for ε_0 and γ, but try $\varepsilon_0 = 0.1$, $\gamma = 0.1$; ε' is a precision parameter, try $\varepsilon' = 10^{-6}$; try $\alpha = 0.25$, $\beta = 0.7$. Try $\lambda = 2$ or $\lambda = 1$.)

Step 1: Compute a $z_0 \in \Omega$ by applying (2.8) to (2.7), and set i = 0, $\varepsilon = \varepsilon_0$.

Step 2: Set

$$I(z_i, \varepsilon) = \{j \in \{1,\ 2,\ \ldots,\ m\} \mid f^j(z_i) \ge -\varepsilon\} \qquad (2.9)$$

$$J(z_i,\epsilon) = I(z_i,\epsilon) \cup \{0\} \tag{2.10}$$

and go to Step 3p (p = 1 or 2).

Step 31: Compute $(d^0(z_i,\epsilon), d(z_i,\epsilon))$, where $d^0(z_i,\epsilon) \in \mathbb{R}^1$, $d(z_i,\epsilon) \in \mathbb{R}^n$, as a solution of the linear program

$$\phi(z_i,\epsilon) \triangleq \min d^0 | \langle \nabla f^j(z_i),d \rangle - d^0 \leq 0, \; j \in J(z_i,\epsilon) \tag{2.11}$$

$$|d^\ell| \leq 1, \; \ell = 1, 2, \ldots, n\},$$

and go to Step 4.

Step 32: Compute $(d^0(z_i,\epsilon), d(z_i,\epsilon))$, where $d^0(z_i,\epsilon) \in \mathbb{R}^1$, $d(z_i,\epsilon) \in \mathbb{R}^n$, as a solution of the quadratic program

$$\phi(z_i,\epsilon) = \min\{d^0 + \frac{1}{2}\|d\|^2 | \langle \nabla f^0(z_i),d \rangle - d^0 \leq 0;$$

$$f^j(z_i) + \langle \nabla f^j(z_i),d \rangle - d^0 \leq 0, \; j \in I(z_i,\epsilon)\}, \tag{2.12}$$

and go to Step 4.

Step 4: If $\phi^0(z_i,\epsilon) \leq -\gamma\epsilon^p$, go to Step 6p; else go to Step 5.

Step 5: If $\epsilon \leq \epsilon'$, stop; else, set $\epsilon = \beta\epsilon$ and go to Step 2.

Step 61: Compute the smallest integer $k(z_i,\epsilon) \geq 0$ such that

$$f^j(z_i + \lambda\beta^{k(z_i,\epsilon)} d(z_i,\epsilon)) \leq 0 \text{ for } j = 1, 2, \ldots, m \text{ and}$$

$$f^0(z_i + \lambda\beta^{k(z_i,\epsilon)} d(z_i,\epsilon_i)) - f^0(z_i)$$

$$\tag{2.13}$$

$$- \lambda\beta^{k(z_i,\epsilon)} \alpha \langle \nabla f^0(z_i), d(z_i,\epsilon) \rangle \leq 0,$$

and go to Step 7.

Step 62: Compute the smallest integer $k(z_i,\epsilon) \geq 0$ such that[4]

$$\max\{f^0(z_i + \lambda\beta^{k(z_i,\epsilon)} d(z_i,\epsilon)) - f^0(z_i); \; f^j(z_i + \lambda\beta^{k(z_i,\epsilon)} d(z_i,\epsilon)),$$

$$\tag{2.14}$$

$$j = 1, 2, \ldots, m\} - \lambda\beta^{k(z_i,\epsilon)} \alpha\phi(z_i,\epsilon) \leq 0,$$

and go to Step 7.

Step 7: Set $z_{i+1} = z_i + \lambda \beta^{k(z_i, \varepsilon)} d(z_i, \varepsilon)$.

Step 8: Set $i = i+1$, and go to Step 2.   ¤

Remark: Suppose that $z_i \in \Omega$ and that $\phi(z_i, 0)$ is defined by (2.11) or by (2.12). Then $\phi(z_i, 0) \leq 0$, and $\phi(z_i, 0) = 0$ if and only if there exist multipliers $\mu^0 \geq 0$, $\mu^1 \geq 0$, ..., $\mu^m \geq 0$, such that $\sum_{j=0}^{m} \mu^j \nabla f^j(z_i) = 0$, $\mu^j f^j(z_i) = 0$, $j = 1, 2, \ldots, m$, $\sum_{j=0}^{m} \mu^j = 1$, i.e., $\phi(z_i, 0) = 0$ if and only if z_i satisfies the F. John optimality condition [4] (see Sec. 4.3 of [9]).   ¤

The convergence properties of the two algorithms defined by (2.12), (which can be used even when the set $\{z | f^j(z) \leq 0, j = 1, 2, \ldots, m\}$ is not convex, provided the set $\{z | f^j(z) < 0, j = 1, 2, \ldots, m\}$ is non-empty) are stated below.   ¤

2.15 Theorem: Suppose that the set $\{z | f^j(z) < 0, j = 1, 2, \ldots, m\}$ is non-empty. Then, for $p = 1$ or $p = 2$, and $\varepsilon' = 0$, algorithm (2.8) either jams up at a point z_s, in which case $\phi(z_s, 0) = 0$, or it constructs an infinite sequence $\{z_i\}_{i=0}^{\infty}$ such that every accumulation point $\hat{z}$ of $\{z_i\}_{i=0}^{\infty}$ satisfies $\phi(\hat{z}, 0) = 0$. (see [13] for $p = 1$ and [8] for $p = 2$).   ¤

Note that a sequence $\{z_i\}_{i=0}^{\infty}$ constructed by (2.8) will always have accumulation points when the set $\Omega'(z_0) \triangleq \{z | f^0(z) - f^0(z_0) \leq 0; f^j(z) \leq 0, j = 1, 2, \ldots, m\}$ is bounded. Note also that in using an algorithm such as (2.8) or (2.3), there is no need to extract a convergent subsequence of $\{z_i\}$, since usually the sequence $\{z_i\}_{i=0}^{\infty}$ converges to the set $\{z \in \Omega'(z_0) | \phi(z,0) = 0\}$, i.e. $\inf_{i \in K}\{\|z_i - z\| \mid z \in \Omega'(z_0), \phi(z,0) = 0\} = 0$, where K is the set

-6-

of all positive integers.

3. THE OPTIMAL CONTROL PROBLEMS

For the purpose of applying the algorithms in Section 2, we must state our optimal control problems in a form similar to (2.1). Thus, suppose that $t_0 < t_f$ are given and that $L_\infty^q[t_0, t_f]$ is the Banach space of equivalence classes of essentially bounded, measurable functions from $[t_0, t_f]$ into $\mathbb{R}^q$, with norm $\|u\|_\infty = \text{ess} \sup_{t \in [t_0, t_f]} \|u(t)\|$, where $\|\cdot\|$ denotes the euclidean norm. Suppose that $h^0: \mathbb{R}^s \times \mathbb{R}^q \times \mathbb{R}^1 \to \mathbb{R}^1$, $\psi: \mathbb{R}^s \to \mathbb{R}^1$ and $h: \mathbb{R}^s \times \mathbb{R}^q \times \mathbb{R}^1 \to \mathbb{R}^s$ are continuously differentiable functions. Then we define $f^0: \|^s \times L_\infty^q[t_0, t_f] \to \|^1$ by

$$f^0(\xi, u) = \int_{t_0}^{t_f} h^0(x(t,\xi,u), u(t), t) \, dt + \psi(x(t_f, \xi, u)) \qquad (3.1)$$

where $x(t, \xi, u)$, $t \in [t_0, t_f]$, is the solution of the d.e.

$$\frac{d}{dt} x = h(x, u, t), \quad t \in [t_0, t_f], \qquad (3.2)$$

with $x(t_0) = \xi$.

Next, let $g_0^j: \mathbb{R}^s \to \mathbb{R}^1$, $j = 1, 2, \ldots, m_0$, and $g_f^j: \mathbb{R}^s \to \mathbb{R}^1$, $j = 1, 2, \ldots, m_f$, be continuously differentiable functions. For $j = 1, 2, \ldots, m$, $m = m_0 + m_f$, we define $f^j: \mathbb{R}^s \times L^q[t_0, t_f] \to \mathbb{R}^1$ as follows:

$$f^j(\xi, u) = g_0^j(\xi), \quad j = 1, 2, \ldots, m_0 \qquad (3.3)$$

$$f^{j+m_0}(\xi, u) = g_f^j(x(t_f, \xi, u)), \quad j = 1, 2, \ldots, m_f. \qquad (3.4)$$

With these definitions we shall be able to solve the problems P1, P2,

P3, below.

P1 $\quad\min\{f^0(\xi,u)\,|\,\xi = \xi_0;\ u(t) \in U \subset \mathbb{R}^q,\ t \in [t_0,t_s]\}$, $\qquad$ (3.5)

with

$$U = \{v \in \mathbb{R}^q\,|\,a^i \leq v^i \leq b^i,\ i = 1,\ 2,\ \ldots,\ q\}, \qquad (3.6)$$

or with

$$U = \{v \in \mathbb{R}^q\,|\,\|v\| \leq 1\}. \qquad \unlhd \qquad (3.7)$$

P2 $\quad\min\{f^0(\xi,u)\,|\,f^j(\xi,u) \leq 0,\ j = 1,\ 2,\ \ldots,\ m\}$ $\qquad$ (3.8)

For P2 (see (3.3), (3.4)) we assume that either the set $\{(\xi,u)\ g_0(\xi) \leq 0,\ g_f(x(t_f,$ $\xi,u)) < 0\}$ is not empty $(g_0 = (g_0^1,g_0^2,\ \ldots,\ g_0^{m_0}),\ g_f = (g_f^1,g_f^2,\ \ldots,\ g_f^{m_f}))$, or that $\{\xi_0\} = \{\xi\,|\,g_0(\xi) \leq 0\}$ and the set $\{u\,|\,g_f(x(t_f,\xi_0,u)) < 0\}$ is not empty. $\quad\unlhd$

Finally, let $N > 0$ be an integer and let $\Delta = (t_f - t_0)/N$. For $i = 0,\ 1,$ $2,\ \ldots,\ N$, let $g_i^j:\ \mathbb{R}^s \to \mathbb{R}^1,\ j = 1,\ 2,\ \ldots,\ m_i$, be continuously differentiable. Then the discrete optimal control problem that we can solve is

P3 $\quad\min\{f^0(\xi,u)\,|\,g_i^j(x(i\Delta,\xi,u)) \leq 0,\ i = 0,\ 1,\ 2,\ \ldots,\ N$,

$$j = 1,2,\ldots,m_i;\ u(t) = \sum_{i=0}^{N-1} v_i \pi(t-i\Delta),\ v_i \in U \subset \mathbb{R}^q\}, \qquad (3.9)$$

with U as in (3.6) or (3.7), f^0 as in (3.1) and with

$$\pi(t) = 1 \quad \text{for } t \in [0,\Delta) \qquad (3.10)$$

$$= 0 \quad \text{otherwise}.$$

For P3, with U as in (3.6), we must assume that either the set $\{(\xi,v_0,v_1,$ $\ldots,\ v_{N-1})\,|\,g_i^j(x(i\Delta,\xi,u)) < 0,\ j = 1,\ 2,\ \ldots,\ m_i,\ i = 0,\ 1,\ 2,\ \ldots,\ N$; $u(t) = \sum_{i=0}^{N-1} v_i\pi(t-i\Delta),\ v_i \in U\}$ is non-empty; or that the set $\{\xi\,|\,g_0(\xi) \leq 0\} =$ $\{\xi_0\}$ and that the set $\{(v_0,\ v_1,\ \ldots,\ v_{N-1})\ g_i^j(x(i\Delta,\xi_0,u)) < 0,\ j = 1,\ 2,\ldots,m_i,$

$i = 1, 2, \ldots, N, \; u(t) = \sum_{i=0}^{N-1} v_i \pi(t-i\Delta), \; v_i \in U.\}$ is not empty. When U

is as in (3.7), replace U by int U in the preceding conditions. $\quad$ ◻

Note that the discretization in P3 is only of the control and not

of the differential equation (3.2). However, while Pl and P2 are pro-

blems on the infinite dimensional space $\mathbb{R}^s \times L_\infty^q[t_0, t_f]$, problem P3 is

defined on the finite dimensional space $\mathbb{R}^s \times \mathbb{R}^q \times \ldots \times \mathbb{R}^q \; (= \mathbb{R}^{s+Nq})$,

and is obviously of the form of the problem (2.1), with $n = s+Nq$ and m

determined by the number of the g_j^i and the description of U.

The algorithms in Section 2 make use of gradients and scalar prod-

ucts. For the problems stated in this section, we use the scalar

product $\langle \cdot, \cdot \rangle_2$ on $\mathbb{R}^s \times L_\infty^q[t_0, t_f]$, defined by

$$\langle (\xi_1, u_1), (\xi_2, u_2) \rangle_2 = \langle \xi_1, \xi_2 \rangle + \int_{t_0}^{t_f} \langle u_,(t), u_2(t) \rangle \, dt, \quad (3.11)$$

where, as before, $\langle \cdot, \cdot \rangle$ denotes the euclidean scalar product. The deri-

vation of the gradients below can be found in [9] Section 2.5; here we

shall merely state the formulas for their computation. These gradients

have the same properties with respect to linear expansions as gradients

in $\mathbb{R}^n$. Note that, as defined below, $\nabla f^j(\xi, u)$, $j = 0, 1, \ldots, m$, is an

element (a pair) in $\mathbb{R}^s \times L_\infty^q[t_0, t_f]$. The first part of the pair is

the gradient with respect to the initial state, while the second part

is the gradient with respect to the control. Thus, for problems Pl and

P2,

$$\nabla f^0(\xi, u) = (\nabla_\xi f^0(\xi, u), \nabla_u f^0(\xi, u)(\cdot)) = (- p_0(t_0, \xi, u),$$

$$\hspace{11cm} (3.12)$$

$$- [\frac{\partial h}{\partial u}(x(\cdot, \xi, u), u(\cdot), \cdot)]^T p_0(\cdot, \xi, u) + [\frac{\partial h^0}{\partial u}(x(\cdot, \xi, u), u(\cdot), \cdot)]^T),$$

(i.e. it is a pair consisting of a vector in $\mathbb{R}^s$ and of a vector valued function in $L_\infty^q [t_0, t_f]$) where $p_0(\cdot, \xi, u)$ is defined by

$$\frac{d}{dt} p_0(t, \xi, u) = - [\frac{\partial h}{\partial x} (x(t, \xi, u), u(t), t)]^T p_0(t, \xi, u)$$

$$+ [\frac{\partial h^0}{\partial x} (x(t, \xi, u), u(t), t)]^T, t \in [t_0, t_f], \tag{3.13}$$

$$p_0(t_f, \cdot, u) = - [\frac{\partial \psi}{\partial x} (x(t_s, \xi, u))]^T. \tag{3.14}$$

Also for problems P1 and P2, and $j = 1, 2, \ldots, m$,

$$\nabla f^j(\xi, u) = (\nabla_\xi f^j(\xi, u), \nabla_u f^j(\xi, u)) = (\frac{\partial g_0^j}{\partial x}(\xi), 0), j = 1, 2, \ldots, m_0, \tag{3.15}$$

$$\nabla f^{j+m_0}(\xi, u) = (\nabla_\xi f^{j+m_0}(\xi, u), \nabla_u f^{j+m_0}(\xi, u)) = (- p_j(t_0, \xi, u),$$

$$- [\frac{\partial h}{\partial u} (x(\cdot, \xi, u), u(\cdot), \cdot)]^T p_j(\cdot, \xi, u)), j = 1, 2, \ldots, m_f, \tag{3.16}$$

where, for $j = 1, 2, \ldots, m_f$, the $p_j(\cdot, \xi, u)$ are defined by

$$\frac{d}{dt} p_j(t, \xi, u) = - [\frac{\partial h}{\partial x} (x(t, \xi, u), u(t), t))]^T p_j(t, \xi, u),$$

$$t \in [t_0, t_f], \tag{3.17}$$

$$p_j(t_f, \xi, u) = - [\frac{\partial g_f^j}{\partial x} (x(t_f, \xi, u))]^T \tag{3.17'}$$

In the case of problem P3, the discretization of the control implies that f^0 is a function of the initial state ξ and of the control sequence $V = (v_0, v_1, \ldots, v_{N-1})$. Hence, given an initial state ξ, and a control sequence V, we obtain, with $u = \sum_{i=0}^{N-1} v_i \pi(t - i\Delta)$

$$\nabla_\xi f^0(\xi,u) = [\frac{\partial f^0}{\partial \xi}(\xi,u)]^T = p_0(t_0,\xi,u), \qquad (3.18)$$

and, for $i = 0,1,2,\ldots,N-1$, (note that $\nabla_{v_i} f^0 \triangleq [\frac{\partial f^0}{\partial v_i}]^T$, etc.)

$$[\frac{\partial f^0}{\partial v_i}(\xi,u)]^T = \int_{t_0}^{t_f} \{- [\frac{\partial h}{\partial u}(x(t,\xi,u),u(t),t)]^T p_0(t,\xi,u) +$$

$$(3.19)$$

$$[\frac{\partial h^0}{\partial u}(x(t,\xi,u),u(t),t)]^T\} \pi(t-i\Delta)dt .$$

where p_0 is computed as in (3.13), (3.14).

Similarly, for $j = 1,2,\ldots,m_i$ and $i = 0,1,2,\ldots,N-1$, we obtain

$$[\frac{\partial g_i^j}{\partial \xi}(x(i\Delta,\xi,u))]^T = p_{ij}(t_0,\xi,u), \qquad (3.20)$$

$$[\frac{\partial g_i^j}{\partial v_\ell}(x(i\Delta,\xi,u))]^T = \int_{t_0}^{t_f} -\{[\frac{\partial h}{\partial u}(x(t,\xi,u),u(t),t)]^T \times$$

$$(3.21)$$

$$p_{ij}(t,\xi,u)\, \pi(t-\ell\Delta)dt, \text{ for } \ell = 0,1,\ldots,i-1,$$

$$= 0 \text{ for } \ell = i, i+1, \ldots, N-1,$$

where $p_{ij}(t,\xi,u)$ is defined by

$$\frac{d}{dt} p_{ij}(t,\xi,u) = - [\frac{\partial h}{\partial x}(x(t,\xi,u),u(t),t)]^T p_{ij}(t,\xi,u),$$

$$(3.22)$$

$$t \in [t_0,t_f],$$

$$p_{ij}(i\Delta,\xi,u) = - [\frac{\partial g_i^j}{\partial x}(x(i\Delta,\xi,u))]^T. \qquad (3.23)$$

Thus, the discretization in problem P3 does not remove the need for integrating differential equations. Its main advantage is that it results in a problem which we can solve, at least in principle, by algorithm (2.8),

with p = 1, whereas we do not know how to solve continuous time problems
with control and state space constraints by means of feasible directions
algorithms.

4. EXTENSION OF NONLINEAR PROGRAMMING ALGORITHMS.

We shall now show how to apply algorithm (2.3) to the problem P1,
algorithm (2.8) with p = 2, to problem P2, and algorithm (2.8),
p = 1, to problem P3.

Problem P1 and the modified Frank-Wolfe method (2.3).

Thus, consider problem P1 and suppose that we have a control $u_i(\cdot)$
such that $u_i(t) \in U$ for $t \in [t_0, t_f]$, where U is as in (3.6) or as in
(3.7). To compute $u_{i+1}(\cdot)$ according to algorithm (2.3) we must first
solve (2.4), where we associate z_i with $u_i(\cdot)$. Following Barnes [2],
for improved rate of convergence, we set $D(u_i)(t) = \frac{1}{2} \frac{\partial^2 h^0}{\partial u^2} (x(t, \xi_0, u_i),$
$u_i, t)$, $t \in [t_0, t_f]$, if that matrix is positive semi-definite; otherwise,
to avoid singular subproblems, we set $D(u_i)(t) \equiv I$. If we calculate
$\nabla f^0(\xi_0, u_i)$ by linearization, rather than by formula (3.12), we find that
(2.4) is equivalent to the quadratic cost optimal control problem

$$d^0(u_i) = \min\{ \int_{t_0}^{t_f} [\frac{\partial h^0}{\partial x} (x(t, \xi_0, u_i), u_i(t), t) \delta x(t) +$$

$$\frac{\partial h^0}{\partial u} (x(t, \xi_0, u_i), u_i(t), t) \delta u(t) + \langle \delta u(t), D(u_i)(t) \delta u(t) \rangle] dt +$$

$$\frac{\partial \psi}{\partial x} (x(t_f, \xi_0, u_i)) \delta x \ \Big| \ \delta \dot{x}(t) = \frac{\partial h}{\partial x} (x(t, \xi_0, u_i), u_i(t), t) \delta x$$

$$+ \frac{\partial h}{\partial u} (x(t, \xi_0, u_i), u_i(t), t) \delta u(t), \ t \in [t_0, t_f],$$

$$\delta x(t_0) = 0; \ [u_i(t) + \delta u(t)] \in U, \ t \in [t_0, t_f]\}. \quad (4.1)$$

We solve (4.1) by means of the Pontryagin Maximum Principle [11] and

denote the optimal control for (4.1) by $\delta u_i(\cdot)$. Next, we must compute the step size $\beta^{k(u_i)}$ as given by (2.5), which in this case becomes, because of (3.1) and (3.12),

$$\int_{t_0}^{t_f} [h^0(x(t,\xi_0,u_i+\beta^{k(u_i)}\delta u_i),u_i(t)+\beta^{k(u_i)}\delta u_i(t), t)$$

$$- h^0(x(t,\xi_0,u_i),u_i(t),t) - \beta^{k(u_i)}\alpha\langle \nabla_u f^0(\xi_0,u_i)(t),\delta u_i(t)\rangle]dt \leq 0. \tag{4.2}$$

Note that in solving (4.1) we have also computed $\nabla_u f^0(\xi_0,u_i)(t) =$

$- [\frac{\partial h}{\partial u}(x(t,\xi_0,u_i),u_i(t),t)]^T p_0(t,\xi_0,u_i) + [\frac{\partial h^0}{\partial u}(x(t,\xi_0,u_i),u_i(t),t)]^T,$

since the adjoint equations for (4.1) coincide with (3.13), (3.14).

The next control $u_{i+1}(\cdot)$ is then computed according to $u_{i+1}(t) = u_i(t) + \beta^{k(u_i)}\delta u_i(t), t \in [t_0,t_f].$

Note that, to compute the step size $\beta^{k(u_i)}$, we may have to integrate the system (3.2) (with $\xi = \xi_0$) several times, once for each trial value of $k \geq 0$ which we wish to test for the condition in (4.2).

Problem P2 and the Pironneau-Polak Algorithm (2.8), with p = 2.

Next, let us turn to problem P2 for which we now adapt algorithm (2.8) with p = 2. For this purpose, we must find a way for solving (2.12), with the gradients and scalar products as defined in Section 3. This task is made easy by the fact that (2.12) has a convenient dual (see [8]), so that $\phi(z_i,\varepsilon)$ and $d(z_i,\varepsilon)$ can also be computed by solving the dual quadratic program

$$\phi(z_i,\varepsilon) = \max\{ \sum_{j \in I(z_i,\varepsilon)} \mu^j f^j(z_i) - \frac{1}{2}\| \sum_{j \in J(z_i,\varepsilon)} \mu^j \nabla f^j(z_i)\|^2 \Big| \mu^j \geq 0,$$

$$j \in J(z_i,\varepsilon), \sum_{j \in J(z_i,\varepsilon)} \mu^j = 1\}, \tag{4.3}$$

and then setting

$$d(z_i, \varepsilon) = \sum_{j \in J(z_i, \varepsilon)} \mu_i^j \nabla f^j(z_i).$$ (4.4)

Where the μ_i^j, $j \in J(z_i, \varepsilon)$, solve (4.3). The importance of (4.3) is that the dimension of this problem depends only on the number of ε - active constraints and not on the dimension of z_i. Consequently, even in the case of the infinite dimensional problem P2, (4.3) remains a finite dimensional quadratic program. To be specific, given a feasible initial state ξ_i and a feasible control $u_i(\cdot)$, which we associate with z_i in (4.3), (4.4) according to $z_i = (\xi_i, u_i(\cdot))$, (4.3) becomes,

$$\phi(z_i, \varepsilon) = \max\Bigg\{ \sum_{j \in I(z_i, \varepsilon)} \mu^j f^j(\xi_i, u_i) - \frac{1}{2} \langle \mu, \Big(F_{\xi, J(z_i, \varepsilon)}^T F_{\xi, J(z_i, \varepsilon)}$$

$$+ \int_{t_0}^{t_f} F_{u, I(z_i, \varepsilon)}^T F_{u, I(z_i, \varepsilon)} dt \Big) \mu \rangle \,\Big|\, \mu^j \geq 0, \ j \in J(z_i, \varepsilon_i), \ \sum_{j \in J(z_i, \varepsilon)} \mu^j = 1 \Bigg\},$$ (4.5)

where the f^j, $j = 0,1,2,\ldots,m$, are defined as in (3.1), (3.3) and (3.4), $I(z_i, \varepsilon) = \{ j \in \{1,2,\ldots,m\} \,|\, f^j(\xi_i, u_i) \geq - \varepsilon \}, = \{\ell_1, \ell_2, \ldots, \ell_r\}, \ r \leq m$, $\mu = (\mu^0, \mu^{\ell_1}, \mu^{\ell_2}, \ldots, \mu^{\ell_r})^T$, $F_{\xi, J(z_i, \varepsilon)}$ is a matrix with colums $\nabla_\xi f^j(\xi_i, u_i)$, $j \in J(z_i, \varepsilon)$, the columns being ordered in the same way as the components of μ, and $F_{u, J(z_i, \varepsilon)}(t)$ is a matrix valued function of t, the columns of $F_{u, J(z_i, \varepsilon)}(t)$ being $\nabla_u f^j(\xi_i, u_i)(t)$, $j \in J(z_i, \varepsilon)$.

Thus, to use (2.8) with $p = 2$, at each iteration we begin with a feasible pair (ξ_i, u_i) and an $\varepsilon > 0$. Then, we carry out the following operations.

(i) We evaluate the functions $f^j(\xi_i, u_i)$, $j = 0,1,\ldots,m$.

(ii) (step 2 of (2.8)) We construct the index sets $I(z_i, \varepsilon)$ and $J(z_i, \varepsilon)$.

(iii) We calculate the gradients $\nabla f^j(\xi_i, u_i)$, $j \in J(z_i, \varepsilon)$, according to (3.12)-(3.17')

(iv) We compute the coefficients of the quadratic form in (4.5).

(v) (step 32 of (2.8)) We solve (4.5) (the dual of (2.12)) by a method
such as Wolfe's [12] to obtain a vector $\mu_i = (\mu_i^0, \mu_i^{\ell_1}, \ldots, \mu_i^{\ell_r})$ and
$\phi(z_i, \varepsilon)$.

(vi) We set

$$\delta\xi_i = -\sum_{j \in J(z_i, \varepsilon)} \mu_i^j \nabla_\xi f^j(\xi_i, u_i), \qquad (4.5')$$

and

$$\delta u_i(t) = -\sum_{j \in J(z_i, \varepsilon)} \mu^i \nabla_u f^j(\xi_i, u_i)(t); \qquad (4.5'')$$

so that we associate $d(z_i, \varepsilon)$ with the pair $(\delta\xi_i, \delta u_i(\cdot))$.

(vii) We then go through the tests in Steps 4 and Step 5 of (2.8)
until we reach Step 62, where we calculate the smallest integer
$k(z_i, \varepsilon)$ such that (see (2.14)),

$$\max\Big\{ \int_{t_0}^{t_f} [h^0(x(t, \xi_i + \lambda\beta^{k(z_i, \varepsilon)} \delta\xi_i, u_i + \lambda\beta^{k(z_i, \varepsilon)} \delta u_i),$$

$$u_i(t) + \lambda\beta^{k(z_i, \varepsilon)} \delta u_i(t), t) - h^0(x(t, \xi_i, u_i), u_i, t)]dt; \qquad (4.6)$$

$$g_0^i(\xi_i + \lambda\beta^{k(z_i, \varepsilon)} \delta\xi_i), j = 1, 2, \ldots, m_0; \ g_f^j(x(t_f, \xi_i + \lambda\beta^{k(z_i, \varepsilon)} \delta\xi_i,$$

$$u_i + \lambda\beta^{k(z_i, \varepsilon)} \delta u_i)), j = 1, 2, \ldots, m_f\} - \lambda\beta^{k(z_i, \varepsilon)} \alpha\phi(z_i, \varepsilon) \le 0.$$

(viii) (Step 7) We set $\xi_{i+1} = \xi_i + \lambda\beta^{k(z_i, \varepsilon)} \delta\xi_i$, $u_{i+1} = u_i + \lambda\beta^{k(z_i, \varepsilon)} \delta u_i$
and continue, with i+1 replacing i in all expressions.

Problem P3 and the Zoutendijk type Algorithm (2.8) with p = 1.

Apart from the cumbersome evaluation of functions and derivatives, formulas

for which were given in the preceding section, the application of algorithm (2.8), with p = 1, to problem P3 is straightforward, once the identification $z = (\xi, v_0, v_1, \ldots, v_{N-1})$ is made. We shall therefore elaborate no further.

Convergence

In the case of the optimal control problems P1 and P2, the condition $d^0(\hat{z}) = 0$ $(\hat{z} = u(\cdot))$ which the modified Frank-Wolfe method attempts to satisfy, and the condition $\phi(\hat{z}, 0) = 0$ $(\hat{z} = (\hat{\xi}, \hat{u}(\cdot)))$ which the Pironneau-Polak method tries to satisfy, can both be shown to be equivalent to the Pontryagin maximum principle in differential form, i.e. they imply the existence of an adjoint vector $p(\cdot)$, satisfying the Pontryagin transversality conditions, such that for some $p^0 \leq 0$, $\langle \frac{\partial}{\partial u}(p^0 h^0(x(t), u(t), t) + \langle p(t), h(x(t), u(t), t) \rangle), \delta u \rangle \leq 0$ for all $\delta u \in U$ $(U = \mathbb{R}^s$ for P2).

We can now summarize the convergence properties of the algorithms (2.3) and (2.8) with respect to the problems P1, P2 and P3. We find that these are slightly better than a direct extension of theorems (2.6) and (2.15) would indicate. Thus,

(i) Suppose that the sequence $\{u_i\}$ constructed by algorithm (2.3) in solving problem P1 remains bounded (i.e. there is an $M \in (0, \infty)$ such that $\|u_i(t)\| < M$ for $i = 0, 1, 2, \ldots$, and all $t \in [t_0, t_f]$). Then $d^0(\hat{z}) = 0$ for all accumulation points $\hat{z} = \hat{u}(\cdot)$ of $\{u_i(\cdot)\}$, where we may take $\hat{u}(\cdot)$ to be an accumulation point of $\{u_i\}$ in the $L_\infty \cap L_2$ sense (i.e., $\|u_i(t)\| \leq M$, for some $M < \infty$, $i = 0, 1, 2, \ldots$, $t \in [t_0 t_f]$, and

for some infinite subset $K \subset \{0, 1, 2, \ldots\}$, $\lim\limits_{\substack{i \to \infty \\ i \in K}} \int_{t_0}^{t_f} \|u_i(t) - \hat{u}(t)\|^2 dt = 0$),

which is somewhat more general than an accumulation point in $L_\infty[t_0, t_f]$.

(ii) Assuming that the sequences $\{\xi_i\}$ and $\{u_i\}$ constructed by algorithm (2.8), with p = 2, remain bounded (as in (i) above), $\phi(\hat{z},0) = 0$ for all accumulation points $\hat{z} = (\hat{\xi},\hat{u})$ of the sequence $\{(\xi_i,u_i)\}$, where we may construe $\hat{u}(\cdot)$ to be an accumulation point of $\{u_i\}$ in the $L_\infty \cap L_2$ sense.

(iii) When algorithm (2.8), with p = 1, is applied to problem P3, theorem (2.15) remains valid without qualifications.

CONCLUSION

We have shown that certain methods of feasible directions can be extended for use in optimal control. It is to be remembered that in using methods of feasible directions in optimal control, the major cost is in the many integrations required per iteration. This cost can be reduced substantially by integrating coarsely when far from a solution and by refining the precision of integration adaptively as a solution is approached. The reader will find details of procedures for doing this in Appendix A of [9], and in [9'], which deals specifically with the Pironneau-Polak method.

Footnotes

1. The Frank-Wolfe method and its extensions belong to the class of feasible directions algorithms.

2. The choice $D(z) \equiv 0$ was used by Frank and Wolfe and results in slow convergence, proportional to $\frac{1}{i}$. $D(z) > 0$ can sometimes be chosen to obtain a linear rate of convergence, see [2]. Note that the algorithms we are about to state involve various parameters which must be preselected. We shall indicate a first choice for these parameters. However, this choice may not always be the best and the reader is encouraged to experiment a little.

3. Note that the need to solve (2.4) restricts this method to problems in which the f^i, $1 = 2, 2, \ldots, m$, are affine, unless $D(z) \equiv 0$, in which case a single ($m = 1$) quadratic constraint can be accomodated.

4. Relation (2.14) defines a step size subroutine of the "centers" type. It keeps the iterates z_i in the interior of Ω, a feature which is useful in optimal control when coarse integration is used in the early iterations. Step size rule (2.12) can also be used with $p = 2$, if preferred.

REFERENCES

1. Armijo, L., "Minimization of functions having continuous partial derivatives," Pacific J. Math. $\underline{16}$, pp. 1-3 (1966).

2. Barnes, E., "A geometrically convergent algorithm for solving optimal control problems," IBM Research Report RC 3025 September 3, 1970.

3. Canon, M. D., Cullum, C. D., Polak, E., Theory of Optimal Control and Mathematical Programming, McGraw-Hill, 1970.

4. Demyanov, V. F., and A. M. Rubinov, Approximate Methods in Optimization Problems, Elsevier, New York, 1971.

5. Frank, M. and P. Wolfe, "An algorithm for quadratic programming," Naval Res. Logist. Quart., $\underline{3}$, pp. 95-110, 1956.

6. John, F., "Extremum problems with inequalities as side conditions," in Courant Anniversary Volume (K. O. Friedrichs et al., eds.) pp. 187-204, Interscience, New York, 1948.

7. Levitin, E. S. and B. T. Polyak, "Constrained minimization methods," J. of Comp. Math. and Math. Physics, $\underline{6}$, 5, pp. 787-823 (1966).

8. Pironneau, O. and E. Polak, "A dual method for optimal control problems with initial and final boundary constraints," University of California, ERL Tech. Memo. no. ERL-M299, 30 March 1971.

9. Polak, E., Computational methods in optimization: A unified approach, Academic Press, New York, 1971.

9'. Polak, E., "On a class of numerical methods with an adaptive integration subprocedure for optimal control problems," Proc. 4th IFIP Colloqium on Optimization, Santa Monica, Calif., Oct. 19-22, 1971.

10. Polak, E., H. Mukai, and O. Pironneau, "Methods of centers and methods of feasible directions for the solution of optimal control problems," Proc. 1971 IEEE Conference on Decision and Control, Dec. 15-17, 1971, Miami Beach, Florida.

11. Pontryagin, L. S., V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mischenko, The mathematical theory of optimal processes, Interscience Pub. New York, 1962.

12. Wolfe, P., "The simplex method for quadratic programming," Econometrica $\underline{28}$, 3, pp. 382-398 (1959).

13. Zoutendijk, G., Methods of feasible directions, Elsevier, Amsterdam, 1960.