JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 63, No. 1, OCTOBER 1989

Efficient Dynamic Programming Implementations of Newton's Method for Unconstrained Optimal Control Problems[1]

J. C. DUNN[2] AND D. P. BERTSEKAS[3]

Communicated by E. Polak

Abstract. Naive implementations of Newton's method for unconstrained N-stage discrete-time optimal control problems with Bolza objective functions tend to increase in cost like N^3 as N increases. However, if the inherent recursive structure of the Bolza problem is properly exploited, the cost of computing a Newton step will increase only linearly with N. The efficient Newton implementation scheme proposed here is similar to Mayne's DDP (differential dynamic programming) method but produces the Newton step exactly, even when the dynamical equations are nonlinear. The proposed scheme is also related to a Riccati treatment of the linear, two-point boundary-value problems that characterize optimal solutions. For discrete-time problems, the dynamic programming approach and the Riccati substitution differ in an interesting way; however, these differences essentially vanish in the continuous-time limit.

Key Words. Unconstrained optimal control, Newton's method, dynamic programming.

1. Introduction

Assume that J is a twice continuously differentiable real-valued function defined on an open set in a real Hilbert space Ω with inner product $\langle \cdot . \cdot \rangle$, and let $\nabla J(u)$ and $\nabla^2 J(u)$ denote the corresponding gradient vector

[1]This work was supported by the National Science Foundation, Grant No. DMS-85-03746.

[2]Professor, Department of Mathematics, North Carolina State University, Raleigh, North Carolina.

[3]Professor, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts.

and Hessian operator at $u \in \Omega$ (with respect to $\langle \cdot, \cdot \rangle$). By definition, ξ is an extremal of J iff

$$\nabla J(\xi) = 0, \tag{1}$$

and an extremal ξ is nonsingular iff $\nabla^2 J(\xi)$ is one-to-one and onto. Near a nonsingular extremal, Newton's map is well defined by

$$u \to u + \delta u, \tag{2a}$$

with

$$\nabla^2 J(u)\delta u = -\nabla J(u). \tag{2b}$$

Moreover, the iterates of this map converge rapidly (in fact, superlinearly) to ξ for all nearby starting points (Ref. 1).

In general, the rapid convergence rate of Newton's method is at least partially offset by the cost of computing δu at each iteration, particularly when Ω is finite-dimensional and $d \triangleq \dim \Omega$ is large, e.g., it may require as many as $O(d^3)$ multiplications to obtain δu from (2b) by Gaussian reduction, over and above those calculations entailed in the construction of matrix representors for $\nabla J(u)$ and $\nabla^2 J(u)$. On the other hand, for certain specially structured objective functions J, the $O(d^3)$ estimate is much too conservative and (2) may actually rival or surpass the standard quasi-Newton methods (Ref. 2) in computational efficiency. In the present paper, this point is developed for unconstrained discrete-time and continuous-time optimal control problems with Bolza objective functions.

The N-stage discrete-time optimal control problems treated in Section 2 have objective functions J defined by

$$J(u) = P(x_{N+1}) + \sum_{i=1}^{N} l_i(x_i, u_i), \tag{3a}$$

$$x_1 = a, \tag{3b}$$

$$x_{i+1} = f_i(x_i, u_i), \qquad i \le i \le N, \tag{3c}$$

where x_i and u_i lie in real Hilbert spaces $\mathscr{X}$ and $\mathscr{U}$ with $m = \dim \mathscr{U} < \infty$, and where the domain of J lies in the N-fold direct sum

$$\Omega = \mathscr{U} \oplus \cdots \oplus \mathscr{U} \qquad (N\text{-times}), \tag{4a}$$

with elements $u = (u_1, \ldots, u_N)$, $v = (v_1, \ldots, v_N)$, etc., and $\dim \Omega = d = Nm$. The inner product in $\mathscr{U}$ is denoted by $(\cdot, \cdot)$, and the corresponding inner product in Ω is given by

$$\langle u, v \rangle = \sum_{i=1}^{N} (u_i, v_i). \tag{4b}$$

For simplicity, the inner product in X is also denoted by $(\cdot,\cdot)$, and all norms are represented by the common symbol $\|\cdot\|$. If the initial vector a is given, if P and l_i are real-valued C^2-maps defined on all of $\mathscr{X}$ and $\mathscr{X}\times\mathscr{U}$, respectively, and if the f_i's are C^2-maps from $\mathscr{X}\times\mathscr{U}$ to $\mathscr{X}$, then formula (3) defines a real-valued C^2-map J on all of Ω. Under somewhat stronger conditions on Ω and J, Murray and Yakowitz (Ref. 3) have recently shown that Mayne's second-order differential dynamic programming (DDP) method developed in Refs. 4 and 6 produces iterates that converge quadratically to nonsingular local minimizers $\xi=(\xi_1,\ldots,\xi_N)$ and are asymptotically like Newton iterates. In the same reference, it is claimed that the DDP algorithm is far less expensive to implement than Newton's method because the former scheme exploits the recursive structure inherent in (3), while the latter scheme cannot (see the remarks on pp. 398 and 400 in Ref. 3). However, in Section 2, it is shown that Newton's method actually does have its own comparably efficient dynamic programming implementation scheme for (3).

The discrete-time Bolza objective function (3) has a continuous-time counterpart,

$$J(u)=P(x(1))+\int_0^1 l(t,x(t),u(t))\,dt, \tag{5a}$$

$$x(0)=a, \tag{5b}$$

$$dx/dt=f(t,x,u(t)), \qquad 0\le t\le 1, \tag{5c}$$

where $u\colon[0,1]\to\mathscr{U}$ lies in some appropriate Hilbert space Ω, e.g.,

$$\Omega=\mathscr{L}^2([0,1],\mathscr{U}), \tag{6a}$$

with inner product

$$\langle u,v\rangle=\int_0^1 (u(t),v(t))\,dt. \tag{6b}$$

The initial vector $a\in\mathscr{X}$ is supposed to be given, the functions P and l are real-valued C^2-maps defined on $\mathscr{X}$ and $\mathbb{R}^1\times\mathscr{X}\times\mathscr{U}$, and f is a C^2-map from $\mathbb{R}^1\times\mathscr{X}\times\mathscr{U}$ to $\mathscr{X}$. However, these conditions alone are no longer sufficient for J to be C^2, or even well-defined near $u\in\Omega$. Other restrictions are needed to ensure that the initial-value problem (5b)–(5c) has a unique solution on $[0,1]$ for every u in some open subset of Ω. Furthermore, for all such u, for every $v\in\Omega$, and for all sufficiently small real s, the unique solution $x(\cdot\,;u+sv)$ of

$$x(0)=a, \tag{7a}$$

$$dx/dt=f(t,x,u(t)+sv(t)), \qquad 0\le t\le 1, \tag{7b}$$

must have first and second s-derivatives that satisfy standard linear equations of variation associated with (5b)–(5c). These conditions produce tractable integral representations for the first and second differentials of J and lead to a simple procedure for approximating the Newton increment δu for J. Section 3 is concerned mainly with the formal aspects of this procedure, and the (admittedly important) technical issues just raised are largely set aside. Formally, the continuous-time dynamic programming implementation amounts to an unrelaxed version of Merriam's second-variation method (Ref. 14), and is also closely related to Mitter's Riccati implementation of quasilinearization (Ref. 15).

In passing, let us note that, while $\mathscr{X}$ and $\mathscr{U}$ are typically $\mathbb{R}^n$ and $\mathbb{R}^m$ with the standard unweighted inner products, there is little to be gained by imposing this restriction at the outset. On the contrary, a number of interesting control problems are either complicated or excluded in this way, and the mathematical presentation is severely burdened by unwieldy subscript notation and irrelevant formulas for inner products, norms, differentials, and the like. Let it also be noted that no generality is lost by confining the following development to minimization problems.

2. Discrete-Time Optimal Control Problems

Near a nonsingular local minimizer, the Hessian operator $\nabla^2 J(u)$ is positive-definite and it follows that Newton's increment δu is the unique minimizer of the quadratic function

$$\phi(v) = \langle \nabla J(u), v\rangle + (1/2)\langle v, \nabla^2 J(u) v\rangle, \tag{8a}$$

i.e.,

$$\{\delta u\} = \arg\min_{v\in\Omega} \phi(v). \tag{8b}$$

It will now be shown that, if J is defined by (3), then $\phi(v)$ is itself an objective function for a related discrete-time linear-quadratic optimal control problem and δu can be found with the standard dynamic programming embedding technique for such problems. The second-order DDP method (Ref. 6) for (3) is properly viewed as an approximate version of the recursive procedure developed here, and the two schemes actually coincide when the difference equations (3c) are affine in (x_i, u_i).

If J is defined by (3), the chain rule then yields

$$\begin{aligned}\langle \nabla J(u), v\rangle &= (d/ds) J(u+sv)|_{s=0}\\ &= \langle \nabla P, y_{N+1}\rangle + \sum_{i=1}^{N} [(\nabla_x l_i, y_i) + (\nabla_u l_i, v_i)],\end{aligned} \tag{9a}$$

where it is understood that the gradient ∇P and the partial gradients $\nabla_x l_i$ and $\nabla_u l_i$ are evaluated at x_{N+1} and (x_i, u_i), respectively, and where

$$y_i \triangleq (d/ds)x_i(u+sv)|_{s=0}, \tag{9b}$$

with

$$x_1(u+sv)=a, \tag{9c}$$

$$x_{i+1}(u+sv)=f_i(x_i(u+sv),\ u_i+sv_i), \qquad 1\le i\le N, \tag{9d}$$

for all real s. It follows easily from (9b)-(9d) that

$$y_1=0, \tag{10a}$$

$$y_{i+1}=A_i y_i+B_i v_i, \qquad 1\le i\le N, \tag{10b}$$

with

$$A_i=\partial f_i/\partial x, \qquad B_i=\partial f_i/\partial u, \tag{10c}$$

where it is once again understood that the partial Fréchet differentials in (10c) are evaluated at (x_i, u_i).

Expressions (9)-(10) for $\langle \nabla J(u), v\rangle$ can be simplified by introducing new adjoint variables $\psi_i \in \mathscr{X}$ satisfying

$$\psi_{N+1}=\nabla P, \tag{11a}$$

$$\psi_i=A_i^*\psi_{i+1}+\nabla_x l_i, \qquad N\ge i\ge 1, \tag{11b}$$

where $A_i^*:\mathscr{X}\to\mathscr{X}$ is the adjoint of the bounded linear operator $A_i:\mathscr{X}\to\mathscr{X}$. By construction,

$$\begin{aligned}(\psi_{i+1}, y_{i+1}) &= (\psi_{i+1}, A_i y_i+B_i v_i)\\ &= (A_i^*\psi_{i+1}, y_i)+(\psi_{i+1}, B_i v_i),\end{aligned}$$

and therefore

$$(\psi_{i+1}, y_{i+1})-(\psi_i, y_i)+(\nabla_x l_i, y_i)=(\psi_{i+1}, B_i v_i). \tag{12}$$

Now, put

$$H_i(\psi, x, u)=(\psi, f_i(x, u))+l_i, \tag{13a}$$

and observe that

$$\nabla_u H_i(\psi_{i+1}, x_i, u_i)=(B_i^*\psi_{i+1})+\nabla_u l_i. \tag{13b}$$

By summing expressions (12) for $1\le i\le N$, and applying (9a), (10a), (11a), and (13), one gets

$$\langle \nabla J(u), v\rangle=\sum_{i=1}^{N}(r_i, v_i), \tag{14a}$$

with

$$r_i = \nabla_u H_i(\psi_{i+1}, x_i, u_i). \tag{14b}$$

The second-order term on the right in (8a) also can be expressed succinctly in terms of the Hamiltonians H_i and the adjoint variables ψ_i when J is defined by (3). Once again, the chain rule yields

$$\begin{aligned}\langle v, \nabla^2 J(u)v\rangle &= (d^2/ds^2)J(u+sv)|_{s=0}\\ &= (\nabla P, z_{N+1}) + (y_{N+1}, \nabla^2 P y_{N+1}) + \sum_{i=1}^{N} [(\nabla_x l_i, z_i)\\ &\quad + (y_i, \nabla^2_{xx} l_i y_i) + 2(y_i, \nabla^2_{ux} l_i v_i) + (v_i, \nabla^2_{uu} l_i v_i)],\end{aligned} \tag{15a}$$

where y_i is defined as before, and

$$z_i = (d^2/ds^2)x_i(u+sv)|_{s=0}. \tag{15b}$$

In view of (9c)-(9d), the vectors $z_i \in X$ are recursively generated by

$$z_1 = 0, \tag{16a}$$

$$z_{i+1} = A_i z_i + (C_i y_i)y_i + 2(D_i y_i)v_i + (E_i v_i)v_i, \tag{16b}$$

for $1 \leq i \leq N$, with

$$C_i = \partial^2 f_i/\partial x^2, \qquad D_i = \partial^2 f_i/\partial u \partial x, \qquad E_i = \partial^2 f_i/\partial u^2, \tag{16c}$$

where the second Fréchet differentials are evaluated at (x_i, u_i). Notice that $C_i \in BL(\mathscr{X}, \mathscr{X}))$, $C_i y_i \in BL(\mathscr{X}, \mathscr{X})$, etc., where $BL(\mathscr{X}, \mathscr{Y})$ denotes the space of bounded linear operators from $\mathscr{X}$ to $\mathscr{Y}$. Equations (11b) and (16b) easily produce

$$\begin{aligned}&(\psi_{i+1}, z_{i+1}) - (\psi_i, z_i) + (\nabla_x l_i, z_i)\\ &= (\psi_{i+1}, (C_i y_i)y_i) + 2(\psi_{i+1}, (D_i y_i)v_i) + (\psi_{i+1}, (E_i v_i)v_i).\end{aligned} \tag{17}$$

By summing these expressions, and using (11a), (13a), (15a), and (16a), one now obtains

$$\begin{aligned}\langle v, \nabla^2 J(u)v\rangle &= (y_{N+1}, Q_{N+1} y_{N+1})\\ &\quad + \sum_{i=1}^{N} [(y_i, Q_i, y_i) + 2(y_i, R_i v_i) + (v_i, S_i v_i)],\end{aligned} \tag{18a}$$

with

$$Q_{N+1} = \nabla^2 P(x_{N+1}) \tag{18b}$$

and

$$Q_i = \nabla^2_{xx} H_i(\psi_{i+1}, x_i, u_i), \tag{18c}$$

$$R_i = \nabla^2_{ux} H_i(\psi_{i+1}, x_i, u_i), \tag{18d}$$

$$S_i = \nabla^2_{uu} H_i(\psi_{i+1}, x_i, u_i). \tag{18e}$$

Notice that the auxiliary variables z_i do not appear in (18).

The results obtained thus far are collected in the following theorem.

Theorem 2.1. Let Ω be the N-fold direct sum (4a), let J be the discrete-time Bolza objective function defined by (3), and suppose that the functions P, l_i, and f_i are twice continuously differentiable. Furthermore, let ξ be a nonsingular local minimizer of J in Ω. Then, for each u sufficiently near ξ in Ω, the Hessian $\nabla^2 J(u)$ is positive definite and the corresponding Newton increment $\delta u = -\nabla^2 J(u)^{-1} \nabla J(u)$ is the unique minimizing solution of the discrete-time, linear-quadratic optimal control problem with Bolza objective function

$$\phi(v) = \tfrac{1}{2}(y_{N+1}, Q_{N+1} y_{N+1}) + \sum_{i=1}^{N} [(r_i, v_i) + 1/2(y_i, Q_i y_i) + (y_i, R_i v) + 1/2(v_i, S_i v_i)], \tag{19a}$$

where

$$y_1 = 0, \tag{19b}$$

$$y_{i+1} = A_i y_i + B_i v_i, \qquad 1 \le i \le N, \tag{19c}$$

and where the vectors r_i and the operators A_i, B_i, Q_i, R_i, and S_i are obtained from (13a), (14b), and (18b)-(18e) by first finding $x_1, \ldots, x_{N+1}$ with (3b)-(3c), and then finding $\psi_{N+1}, \ldots, \psi_1$ with (11).

The linear-quadratic control problem in Theorem 2.1 can be solved by dynamic programming. Since the procedure is well known (Ref. 5), a brief summary should suffice here. To start with, one embeds the N-stage objective function $\phi(v)$ in a family of similar objectives,

$$\phi_k(y; v_k, \ldots, v_N) = 1/2(y_{N+1}, Q_{N+1} y_{N+1}) + \sum_{i=k}^{N} q_i(y_i, v_i), \tag{20a}$$

with

$$q_i(y_i, v_i) = (r_i, v_i) + 1/2(y_i, Q_i y_i) + (y_i, R_i v_i) + 1/2(v_i, S_i v_i), \tag{20b}$$

$$y_k = y, \tag{20c}$$

$$y_{i+1} = A_i y_i + B_i v_i, \qquad k \le i \le N, \tag{20d}$$

where y is an arbitrary vector in $\mathscr{X}$ and $k = N, N-1, \ldots.$

It is known that

$$\phi_i^0(y) = \inf_{\substack{v_j \in \mathcal{U} \\ i \le j \le N}} \phi_i(y; v_i, \ldots, v_N), \qquad i \le N,\ y \in \mathcal{X}, \tag{21}$$

if and only if

$$\phi_i^0(y) = \inf_{v \in \mathcal{U}} [q_i(y, v) + \phi_{i+1}^0(A_i y + B_i v)], \qquad i \le N,\ y \in \mathcal{X}, \tag{22a}$$

with

$$\phi_{N+1}^0(y) = (1/2)(y, Q_{N+1} y). \tag{22b}$$

Furthermore, suppose that $\phi_i^0(y)$ is generated by (22) for $i \le N$, and consider the set of control vectors

$$v_i(y) = \arg\min_{v \in \mathcal{U}} [q_i(y, v) + \phi_{i+1}^0(A_i y + B_i v)]. \tag{23}$$

Then,

$$\phi_k(y; \eta_k, \ldots, \eta_N) = \phi_k^0(y), \tag{24}$$

if and only if

$$\eta_i \in v_i^0(y_i), \qquad k \le i \le N, \tag{25a}$$

where

$$y_k = y, \tag{25b}$$

$$y_{i+1} = A_i y_i + B_i \eta_i, \qquad k \le i \le N. \tag{25c}$$

Observe now that

$$\phi(v) = \phi_i(0; v_1, \ldots, v_N), \tag{26}$$

by construction. Hence, the minimizers $\eta = (\eta_1, \ldots, \eta_N)$ of ϕ can be obtained by solving (22) backward for $\phi_{N+1}^0(y), \ldots, \phi_2^0(y)$, constructing the sets $v_N^0(y), \ldots, v_1^0(y)$ in the process, and then generating all possible solutions of (25), with $k = 1$ and $y = 0$. In particular, it can be seen that the quadratic function $\phi(v)$ has a unique minimizer $\eta = (\eta_1, \ldots, \eta_N)$ in Ω if and only if there are scalars α_i, vectors $\beta_i \in \mathcal{X}$ and $\gamma_i \in \mathcal{U}$, and bounded linear operators $\theta_i : \mathcal{X} \to \mathcal{X}$ and $\Gamma_i : \mathcal{X} \to \mathcal{U}$, such that, for $1 \le i \le N$,

$$\phi_i^0(y) = \alpha_i + (\beta_i, y) + \tfrac{1}{2}(y, \theta_i y), \tag{27a}$$

$$S_i + B_i^* \theta_{i+1} B_i \text{ is positive definite,} \tag{27b}$$

$$v_i^0(y) = \{\gamma_i + \Gamma_i y\}, \tag{27c}$$

$$\gamma_i = -(S_i + B_i^* \theta_{i+1} B_i)^{-1}(r_i + B_i^* \beta_{i+1}), \tag{27d}$$

$$\Gamma_i = -(S_i + B_i^* \theta_{i+1} B_i)^{-1}(R_i^* + B_i^* \theta_{i+1} A_i), \tag{27e}$$

$$\eta_i = \gamma_i + \Gamma_i y_i, \tag{27f}$$

with

$$\beta_{N+1}=0, \tag{27g}$$

$$\beta_i=\Gamma_i^* r_i+(A_i+B_i\Gamma_i)^*\beta_{i+1}, \qquad N\geq i\geq 1, \tag{27h}$$

$$\theta_{N+1}=Q_{N+1}, \tag{27i}$$

$$\theta_i=Q_i+A_i^*\theta_{i+1}A_i+(R_i^*+B_i^*\theta_{i+1}A_i)^*\Gamma_i, \qquad N\geq i\geq 1, \tag{27j}$$

$$y_1=0, \tag{27k}$$

$$y_{i+1}=(A_i+B_i\Gamma_i)y_i+B_i\gamma_i, \qquad 1\leq i\leq N. \tag{27l}$$

Notice that, since dim $\mathcal{U}<\infty$, the negation of (27b) for some i would imply that, for all y, the sets $v_i^0(y)$ are either empty or contain infinitely many elements; in this case, it is not difficult to see that ϕ would have no minimizer or infinitely many minimizers. Furthermore, since dim $\Omega<\infty$, the quadratic function ϕ has a unique (global) minimizer η if and only if η is a nonsingular (local) minimizer of ϕ. Consequently, ϕ has a positive-definite Hessian if and only if conditions (27) hold.

Our recursive procedure for computing the Newton increment δu may now be summarized as follows.[4]

Algorithm A1

Step 1. Given $u\in\Omega$, construct $x_1,\ldots,x_{N+1}$ with (3b)-(3c) and $\psi_{N+1},\ldots,\psi_1$ with (11).

Step 2. Compute the vectors r_i and the operators A_i, B_i, Q_i, R_i, and S_i with (13a), (14b), and (18b)-(18e).

Step 3. Solve (27g)-(27j) backward for θ_i and β_i, and compute γ_i and Γ_i with (27d)-(27e).

Step 4. Solve (27k)-(27l) forward for y_i, and compute η_i with (27f).

Step 5. Set $\delta u=\eta$.

Note 2.1. When u is sufficiently near a nonsingular minimizer ξ of J in (3), the function $\phi(v)$ has a unique minimizer, condition (27b) is automatically satisfied, and Algorithm A1 is well posed; moreover, the resulting Newton increment δu provides a descent direction for J at u. Elsewhere, if (27b) fails at certain stages k in the implementation of (27g)-(27j), a simple alteration of the algorithm will at least yield a descent direction for J; e.g., one merely replaces S_k by $S_k+\lambda_k I$, where λ_k is a

[4]Note added in proof. Pantoja has published an independent and substantially different derivation of Algorithm A1 in Ref. 16.

positive scalar large enough to make $S_k + \lambda_k I + B_k^* \theta_{k+1} B_k$ positive definite. The resulting vector $\eta = (\eta_1, \ldots, \eta_N)$ then minimizes

$$\langle \nabla J(u), v\rangle + 1/2\langle v, (\nabla^2 J(u) + \Lambda(u))v\rangle, \tag{28a}$$

where

$$\Lambda(u)v = (\lambda_1 v_1, \ldots, \lambda_N v_N). \tag{28b}$$

By construction, $\nabla^2 J(u) + \Lambda(u)$ is positive definite, and

$$\eta = -(\nabla^2 J(u) + \Lambda(u))^{-1} \nabla J(u) \tag{29}$$

is a descent direction for J at u; furthermore, when dim $\mathscr{U}$ is small, the most negative eigenvalue of $S_k + B_k^* \theta_{k+1} B_k$ is easily estimated and a suitable value for λ_k is readily found. A similar procedure is described in Ref. 6 for the second-order DDP method.

Note 2.2. When $\mathscr{U} = \mathbb{R}^m$ and $\mathscr{X} = \mathbb{R}^n$ with the standard unweighted inner products, Algorithm A1 is implemented by replacing vectors with column matrices, operators with matrix representors, operator composition with matrix multiplication, and operator adjoints with matrix transposes.

Note 2.3. If the computational cost associated with the evaluation of the functions f_i, l_i and their differentials is roughly the same at each stage i, then the overall cost of implementing Algorithm A1 is directly proportional to the number of stages N. On the other hand, if $m = \dim \mathscr{U}$, the cost of the linear equation solver in a standard quasi-Newton method for (3) will vary like $(Nm)^3$ in the first iteration, and like $(Nm)^2$ in subsequent iterations (Ref. 2).

Note 2.4. The recursions in (27a) are similar but not equivalent to those obtained in Ref. 7 by treating the quasilinearization boundary-value problem for (3) with a Riccati substitution. More specifically, the latter approach requires inverses for certain operators (e.g., S_i) that actually need not be invertible, even when u is near a nonsingular minimizer. In contrast, condition (27b) always holds near such a minimizer.

3. Continuous-Time Optimal Control Problems

The continuous-time Bolza objective function (5) may be viewed as a "limit" of N-stage discrete-time counterparts, in which integrals are replaced by Riemann sums and differential equations are replaced by their Euler difference approximations. More precisely, let $\Delta t = 1/N$, and construct the mesh

$$0 = t_1 < t_2 < \cdots < t_{N+1} = 1, \tag{30a}$$

with

$$t_{i+1} - t_i = \Delta t. \tag{30b}$$

In place of (5), consider the associated discrete-time objective

$$J(u) = P(x_{N+1}) + \sum_{i=1}^{N} l_i(x_i, u_i), \tag{30c}$$

with

$$x_i = a, \tag{30d}$$

$$x_{i+1} = f_i(x_i, u_i), \tag{30e}$$

where

$$l_i(x, u) = l(t_i, x, u)\Delta t, \tag{30f}$$

$$f_i(x, u) = x + f(t_i, x, u)\Delta t. \tag{30g}$$

A straightforward application of the results in Section 2 establishes that the quadratic function (8) associated with (30) has the Bolza representation

$$\phi(v) = \tfrac{1}{2}(y_{N+1}, Q_{N+1} y_{N+1}) + \sum_{i=1}^{N} q_i(y_i, v_i), \tag{31a}$$

with

$$q_i(y_i, v_i) = [(r_i, v_i) + (1/2)(y_i, Q_i y_i) + (y_i, R_i v_i) + (1/2)(v_i, S_i v_i)]\Delta t, \tag{31b}$$

$$y_1 = 0, \tag{31c}$$

$$y_{i+1} = (I + A_i \Delta t) y_i + (B_i \Delta t) v_i, \tag{31d}$$

$$A_i = \partial f/\partial x(t_i, x_i, u_i), \qquad B_i = \partial f/\partial u(t_i, x_i, u_i), \tag{31e}$$

$$r_i = \nabla_u H_i(\psi_{i+1}, x_i, u_i), \tag{31f}$$

$$Q_{N+1} = \nabla^2 P(x_{N+1}), \tag{31g}$$

$$Q_i = \nabla^2_{xx} H_i(\psi_{i+1}, x_i, u_i), \tag{31h}$$

$$R_i = \nabla^2_{ux} H_i(\psi_{i+1}, x_i, u_i), \tag{31i}$$

$$S_i = \nabla^2_{uu} H_i(\psi_{i+1}, x_i, u_i), \tag{31j}$$

$$H_i(\psi, x; u) = (\psi, f(t_i, x, u)) + l(t_i, x, u), \tag{31k}$$

$$\psi_{N+1} = \nabla P(x_{N+1}), \tag{31ℓ}$$

$$\psi_i = (I + A_i^* \Delta t)\psi_{i+1} + \nabla_x l(t_i, x_i, u_i)\Delta t. \tag{31m}$$

By passing to the limit as $\Delta t \to 0$, one now guesses that the quadratic function (8) associated with the underlying continuous-time objective (5) has the Bolza representation

$$\phi(v) = (1/2)(y(1), \nabla^2 P(x(1))y(1)) + \int_0^1 q(t, y(t), v(t))\, dt, \tag{32a}$$

with

$$q(t, y, v) = (r(t), v) + \tfrac{1}{2}(y, Q(t)y)$$
$$+ (y, R(t)v) + (1/2)(v, S(t)v), \tag{32b}$$

$$y(0) = 0, \tag{32c}$$

$$dy/dt = A(t)y + B(t)v(t), \tag{32d}$$

$$A(t) = (\partial f/\partial x)(t, x(t), u(t)),\ B(t) = (\partial f/\partial u)(t, x(t), u(t)), \tag{32e}$$

$$r(t) = \nabla_u H(t, \psi(t), x(t), u(t)), \tag{32f}$$

$$Q(t) = \nabla^2_{xx} H(t, \psi(t), x(t), u(t)), \tag{32g}$$

$$R(t) = \nabla^2_{ux} H(t, \psi(t), x(t), u(t)), \tag{32h}$$

$$S(t) = \nabla^2_{uu} H(t, \psi(t), x(t), u(t)), \tag{32i}$$

$$H(t, \psi, x, u) = (\psi, f(t, x, u)) + \ell(t, x, u), \tag{32j}$$

$$\psi(1) = \nabla P(x(1)), \tag{32k}$$

$$d\psi/dt = -A(t)^*\psi - \nabla_x l(t, x(t), u(t)). \tag{32ℓ}$$

While this guess is fundamentally sound, it is by no means always correct; in fact, the initial-value problem (5b)-(5c) may have no solution on [0, 1]; when this happens, the objective (5) and the corresponding quadratic function ϕ are undefined at $u(\cdot)$. A rigorous derivation of (32) requires existence, uniqueness, and dependence-on-parameters theorems for the initial-value problem (5b)-(5c). For a further discussion of formula (32) and related analysis, see Refs. 7 and 8.

Assuming that (32) does give the correct Bolza representation for the quadratic part of (5), a plausible formula for the corresponding Newton increment δu may be obtained in a similar fashion [i.e., by adapting Eqs. (27) of Section 2 to the discrete-time objective function (31) and then passing to the limit as $\Delta t \to 0$]. In this way, one guesses that the quadratic function ϕ in (32) has a unique minimizer $\eta(\cdot)$ if and only if

$$S(t) \text{ is positive-definite a.e. in } [0, 1], \tag{33a}$$

and there are vector-valued functions $\beta(\cdot):[0,1]\to\mathscr{X}$ and $\gamma(\cdot):[0,1]\to\mathscr{U}$ and operator-valued functions $\theta(\cdot):[0,1]\to \mathrm{BL}(\mathscr{X},\mathscr{X})$ and $\Gamma(\cdot):[0,1]\to \mathrm{BL}(\mathscr{X},\mathscr{U})$, such that

$$\eta(t)=\gamma(t)+\Gamma(t)y(t), \tag{33b}$$

$$\gamma(t)=-S^{-1}(t)[r(t)+B^*(t)\beta(t)], \tag{33c}$$

$$\Gamma(t)=-S^{-1}(t)[R^*(t)+B^*(t)\theta(t)], \tag{33d}$$

$$\beta(1)=0, \tag{33e}$$

$$d\beta(t)/dt=-A^*(t)\beta(t)-[R(t)+\theta(t)B(t)]\gamma(t), \tag{33f}$$

$$\theta(1)=\nabla^2 P(x(1)), \tag{33g}$$

$$d\theta(t)/dt=-Q(t)-[A^*(t)\theta(t)+\theta(t)A(t)] -[R(t)+\theta(t)B(t)]\Gamma(t), \tag{33h}$$

$$y(0)=0, \tag{33i}$$

$$dy(t)/dt=[A(t)+B(t)\Gamma(t)]y(t)+B(t)\gamma(t), \tag{33j}$$

a.e. in $[0, 1]$. Furthermore, when $u(\cdot)$ is sufficiently near a nonsingular local minimizer of (5), the corresponding Newton increment δu is the unique minimizer of ϕ. Hence, it is plausible that the following algorithm is well posed near a nonsingular minimizer of J and will actually provide the Newton increment δu.

Algorithm A2

Step 1. Given $u(\cdot)\in\Omega$, solve the forward and backward initial-value problems (5b)-(5c) and (32k)-(32l) for $x(\cdot)$ and $\psi(\cdot)$ on $[0, 1]$.

Step 2. Compute the functions $r(\cdot)$, $A(\cdot)$, $B(\cdot)$, $Q(\cdot)$, $R(\cdot)$, and $S(\cdot)$ with (32e)-(32j).

Step 3. Solve the backward initial-value problems (33e)-(33f) and (33g)-(33h) for $\beta(\cdot)$ and $\theta(\cdot)$ on $[0, 1]$, and compute $\gamma(\cdot)$ and $\Gamma(\cdot)$ with (33c)-(33d).

Step 4. Solve the forward initial-value problem (33i)-(33j) for $y(\cdot)$, and compute $\eta(\cdot)$ with (33b).

Step 5. Set $\delta u(\cdot)=\eta(\cdot)$.

Once again, a rigorous justification for this algorithm is nontrivial and requires the construction of a smooth field of extremals with an associated Hamilton-Jacobi equation [i.e., a continuous-time analog of the dynamic programming equation (22); see Refs. 8 and 9]. In fact, Merriam treats the accessory minimum problem for (32) directly with the Hamilton-Jacobi equation and obtains formal solutions with Equations (33) (Ref. 14).

Note 3.1. When the so-called strengthened Legendre-Clebsch condition (33a) does not hold, a simple modification of Algorithm A2 will still produce useful descent directions for the objective J (see Note 2.1).

Note 3.2. In most cases, the nonlinear initial-value problems in Algorithm A2 will have to be solved numerically (e.g., by finite-difference methods). The Euler difference scheme that helped us to guess the continuous-time equations (33) can also be used to compute their solutions approximately on a mesh (30a); however, it is generally more efficient to use a higher-order finite difference method (e.g., the 4th order Runge-Kutta method). The overall cost of such a scheme will typically increase in direct proportion with the number N of subintervals in the net (30a). On the other hand, the cost of implementing a standard quasi-Newton method for (5) will increase like $(Nm)^3$ in the first iteration and $(Nm)^2$ in subsequent iterations (see Note 2.3). Furthermore, the convergence rates of quasi-Newton methods may deteriorate substantially with increasing N, even if the iterations start quite close to a nonsingular minimizer of (5); see Refs. 10 and 11.

If (32) correctly represents the quadratic part of (5) at $u(\cdot)$, and if the operators $S(t)$ are invertible a.e. in $[0, 1]$, then the corresponding Newton increment can be also obtained by putting $\delta u(\cdot) = \eta(\cdot)$, where $\eta(\cdot)$ is derived from the solution of the following linear two-point boundary-value problem:

$$dy/dt = A(t)y + B(t)\eta(t), \tag{34a}$$

$$dp/dt = A^*(t)p - Q(t)y - R(t)\eta(t), \tag{34b}$$

$$y(0) = 0, \tag{34c}$$

$$p(1) = \nabla^2 P(x(1))y(1), \tag{34d}$$

where

$$S(t)\eta(t) = -[R^*(t)y(t) + B^*(t)p(t) + r(t)]. \tag{34e}$$

Regardless of whether $S(t)$ is invertible a.e., it can be shown that the functions $\eta(\cdot)$ obtained in this way are precisely the extremals of the quadratic function defined by (32). This procedure for constructing $\delta u(\cdot)$ differs from Steps 2-5 in Algorithm A2 and is more closely related to a shooting method or a quasilinearization method for the nonlinear two-point boundary value problem that characterizes the extremals of the original objective (5); see Refs. 12 and 13. The costs of solving (34) "numerically" with a finite difference scheme are again (essentially) proportional to N in (30a) when the dimension of $\mathscr{X}$ is finite and small compared to N; however, the constant of proportionality may differ from the corresponding constant for Algorithm A2. Moreover, finite precision calculations with (34) and with Algorithm A2 may produce substantially different approximations to $\delta u(\cdot)$, depending on the procedure used to solve this linear two-point boundary-value problem. One of the more effective procedures is actually equivalent to the dynamic programming scheme (33); i.e., the Riccati substitution $p = \beta + \theta\eta$ leads directly to the final-value problems (33e)-(33h) (Ref. 15); recall that this does not happen in the discrete-time case (see Note 2.4).

References

1. LUENBERGER, D. G., *Optimization by Vector Space Methods*, Wiley, New York, New York, 1969.
2. DENNIS, J. E., JR., and SCHNABEL, R. B., *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, New Jersey, 1983.
3. MURRAY, D. M., and YAKOWITZ, S., *Differential Dynamic Programming and Newton's Method for Discrete Optimal Control Problems*, Journal of Optimization Theory and Applications, Vol. 43, pp. 395-414, 1984.
4. MAYNE, D. Q., *A Second-Order Gradient Method for Determining Optimal Trajectories of Nonlinear Discrete-Time Systems*, International Journal on Control, Vol. 3, pp. 85-95, 1966.
5. BELLMAN, R. E., *Dynamic Programming*, Princeton University Press, Princeton, New Jersey, 1957.
6. JACOBSON, D. H., and MAYNE, D. Q., *Differential Dynamic Programming*, American Elsevier, New York, New York, 1970.
7. POLAK, E., *Computational Methods in Optimization*, Academic Press, New York, New York, 1971.
8. BLISS, G. A., *Lectures on the Calculus of Variations*, University of Chicago Press, Chicago, Illinois, 1946.
9. HESTENES, M., *Calculus of Variations and Optimal Control*, Robert E. Krieger Publishing Company, Huntington, New York, 1980.
10. GRIEWANK, A., *A Superlinear Convergence of Secant Methods on Mildly Nonlinear Problems in Hilbert Space* (to appear).
11. KELLEY, C. T., and SACHS, E., *A Pointwise Quasi-Newton Method for Unconstrained Optimal Control Problems*, Numerische Mathematik, Vol. 55, pp. 159-176, 1989.
12. DREYFUS, S. E., *Dynamic Programming and the Calculus of Variations*, Academic Press, New York, New York, 1965.
13. KELLER, H., *Two-Point Boundary-Value Problems*, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1976.

14. MERRIAM, C. W., III, *An Algorithm for the Iterative Solution of a Class of Two-Point Boundary-Value Problems*, SIAM Journal on Control and Optimization, Series A, Vol. 2, pp. 1-10, 1964.
15. MITTER, S. K., *Successive Approximation Methods for the Solution of Optimal Control Problems*, Automatica, Vol. 3, pp. 135-149, 1966.
16. PANTOJA, J., *Differential Dynamic Programming and Newton's Method*, International Journal on Control, Vol. 47, pp. 1539-1553, 1988.