JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 43, No. 3, JULY 1984

Differential Dynamic Programming and Newton's Method for Discrete Optimal Control Problems[1]

D. M. MURRAY[2] AND S. J. YAKOWITZ[3]

Communicated by D. Q. Mayne

Abstract. The purpose of this paper is to draw a detailed comparison between Newton's method, as applied to discrete-time, unconstrained optimal control problems, and the second-order method known as differential dynamic programming (DDP). The main outcomes of the comparison are: (i) DDP does not coincide with Newton's method, but (ii) the methods are close enough that they have the same convergence rate, namely, quadratic.

The comparison also reveals some other facts of theoretical and computational interest. For example, the methods differ only in that Newton's method operates on a linear approximation of the state at a certain point at which DDP operates on the exact value. This would suggest that DDP ought to be more accurate, an anticipation borne out in our computational example. Also, the positive definiteness of the Hessian of the objective function is easy to check within the framework of DDP. This enables one to propose a modification of DDP, so that a descent direction is produced at each iteration, regardless of the Hessian.

Key Words. Nonlinear programming, optimal control, optimal control algorithms, nonlinear dynamics, quadratic convergence.

1. Introduction

We consider nonlinear programming problems which can be formulated as discrete optimal control problems. For such problems, one has to

[1] Efforts of the first author were partially supported by the South African Council for Scientific and Industrial Research, and those of the second author by NSF Grants Nos. CME-79-05010 and CEE-81-10778.

[2] Research Mathematician, Institute for Maritime Technology, Simonstown, South Africa.

[3] Professor, Systems and Industrial Engineering Department, University of Arizona, Tucson, Arizona.

0022-3239/84/0700-0395$03.50/0

determine the optimal control policy

$$u^* = \{u_i^*\}_{i=1}^n,$$

n being the number of stages or decision times, so as to minimize the performance index

$$V(u) = \sum_{i=1}^{n} L_i(x_i, u_i). \tag{1}$$

The states $\{x_i\}_{i=1}^n$ and controls $\{u_i\}_{i=1}^n$ are presumed related by the dynamical relationship

$$x_{i+1} = f_i(x_i, u_i), \qquad i = 1, \ldots, n-1, \qquad x_1 = \bar{x}_1. \tag{2}$$

The assumption that we hereby adopt of scalar states x_i and controls u_i yields simpler expressions; but each expression has an obvious vector counterpart, and the reader will see that the results remain valid for the general case. In this study, the subscripts identify stages and the unsubscripted variables x and u (respectively the state trajectory and the control policy) denote the n-vectors consisting of the components x_i and u_i, respectively.

The functions L_i and f_i are assumed to be three times continuously differentiable. The existence of a local minimum u^*, associated with the vanishing of the gradient $\nabla V(u)$, with nonsingular Hessian $\mathcal{V}(u^*)$ is assumed. These assumptions are sufficient to assure the quadratic convergence rate of Newton's method (Ref. 1) in a neighborhood of the optimum.

In Section 2, we state the differential dynamic programming (DDP) procedure, while Section 3 develops the terminology for discussing Newton's method. Section 4 presents our main results, such as the proof that DDP is quadratically convergent, illuminating expressions providing Newton counterparts to key DDP constructs, and substantiation of the fact that Newton's method and DDP need not coincide, except in the case that the dynamical Eq. (2) is linear. Section 5 provides an illustrative computational result comparing the performances of the DDP and Newton's method. The concluding section (Section 6) proposes refinements for solving optimal control problems in which the objective functions do not possess globally positive-definite Hessian matrices. It is inexpensive to assure that the DDP method always computes a descent direction, in contrast to the computational effort needed to make such an alteration for Newton's method.

Differential dynamic programming, in a discrete-time context, was introduced by Mayne (Ref. 2). It was described and further refined in Chapter 4 of Jacobson and Mayne (Ref. 3), where a proof of global convergence is sketched. The authors believe that the present work is the first to derive a convergence rate (namely, quadratic) for discrete-time DDP.

Dyer and McReynolds (Ref. 4, Section 3.7) describe a method which is a generalization of DDP, the generalization being that they allow that optimization also be with respect to a control parameter. These authors claim (Section 3.7) that their algorithm is a version of Newton's method. It will be transparent from our study that DDP does not coincide with Newton's method applied in standard form to the objective function $V(u)$ as in (1). Our central intention is to present a detailed analysis of the relationship between the DDP and Newton's methods. Through this comparison, we are able to prove that, like Newton's method, DDP is quadratically convergent; thus, the claims of Dyer and McReynolds regarding the successive sweep method are indeed true in the unconstrained case.

The study closest in result and spirit to the present work is that of Ohno (Ref. 5), who proves the quadratic convergence of a variation of DDP. Ohno's algorithm is fundamentally different from DDP and has the drawback that stepsize adjustment is considerably more expensive to incorporate, because quadraticizations need to be recomputed. This algorithm does not seem to enjoy the popularity of DDP. Ohno's proof is substantially different from ours, in that it does not rely on connections with Newton's method. It is not obvious that his proof is extensible to conventional DDP.

The present study is part of a long-range effort by the authors to derive the computational properties (such as convergence rates, conditions for convergence, etc.) of the major dynamic programming algorithms. By major, we refer to those algorithms which overcome the curse of dimensionality, i.e., the exponential growth of the computational effort with the increase in state dimension, which characterizes discrete dynamic programming. Our realm of major methods is further restricted to optimal control algorithms which avail themselves of functional equation-type decompositions, in the sense of Section 10 of Bellman and Dreyfus (Ref. 6). The properties of the gradient, conjugate gradient, quasi-Newton, and related methods, applied directly to the objective function $V(u)$, are available from the nonlinear programming literature.

There are only three essentially different major methods which have come to our attention: the state-increment dynamic programming technique (SIDP, Larson, Ref. 7, Chapter 12); methods based on use of the discrete maximum principle (DMP); and DDP. The parallel of the discrete maximum principle with the continuous maximum principle and its implementation by standard two-point boundary-value techniques (such as quasilinearization or the shooting method with Newton interation) suggest that, in principle, quadratic convergence ought to be achievable. However (e.g., Halkin, Ref. 1), the conditions under which the discrete maximum principle holds are surprisingly strenuous. It is known (Ref. 8), on the other hand, that SIDP is a generalized nonlinear Gauss–Seidel algorithm and, as such, achieves only linear convergence. An advantage, however, is the SIDP does

not require explicit computation of derivatives. However, the computer implementation of SIDP has so far turned out to be relatively delicate.

By contrast, the authors have found the implementation of DDP to be reliable, robust, and stable. Murray and Yakowitz (Ref. 9) transcribed traditional nonlinear programming (NLP) test problems to optimal control problems (OCP) and solved them by DDP. For many such problems, DDP required fewer functional evaluations than conventional NLP methods, and the advantage of DDP seems to increase with the size of the NLP problem. Yakowitz and Rutherford (Ref. 10) have reported solution to OCP's with highly nonlinear dynamics, nonquadratic loss functions, and as many as 40 state variables.

Murray and Yakowitz (Ref. 11) describe an application of a version of DDP for constrained optimal control to a multireservoir control problem as discussed in the water resource literature. Results of that study suggest that DDP is the most effective of the available techniques for a certain class of large-scale control problems. In this regard, as documented in Ref. 12, water resource theoreticians have been enthusiastic and inventive advocates of the dynamic programming approach over the years.

For reasons such as those that we have just outlined above and discussed in greater length in Refs. 8–13, the authors have concluded that, at present, DDP is surprisingly powerful, robust and reliable when second derivatives can be conveniently calculated. Thus, for most optimal control problems, it is the method of choice. For that reason, we believe that our somewhat laborious demonstration here that DDP is quadratically convergent is a worthwhile contribution to the control literature. Furthermore, it may be of theoretic interest, beyond the subject of optimal control, to establish whether or not there exists a dynamic programming (i.e., stagewise) implementation of Newton's method. For, if there were, and our analysis suggests otherwise, one could solve any nth order linear equation whose coefficient matrix can be viewed as a Hessian of an optimal control problem, with order n arithmetic operations, instead of order n^3, as required by Gaussian elimination.

2. Description of the DDP Algorithm

The purpose of this brief section is to give a self-contained exposition of the DDP algorithm. For information concerning the motivation and computational advantages of DDP, the reader is invited to consult Refs. 3, 4, 9, 11, 12.

DDP is a successive approximation procedure. The initial or nominal policy is denoted by

$$\bar{u} = \{\bar{u}_i\}_{i=1}^{n},$$

and the trajectory determined by $\bar{u}$ through (2) is denoted

$$\bar{x} = \{\bar{x}_i\}_{i=1}^{n},$$

n being the number of decision stages. The objective now is to set forth the DDP construction of the successor policy to $\bar{u}$.

The backward-run phase of the DDP algorithm entails the recursive computation of quadratics $Q_i(x_i, u_i)$. The construction is most easily related, if we adopt certain notational conventions. Symbols like $(f_i)_{ux}$, $(Q_i)_u$, etc., will denote partial derivatives with respect to the state and/or contro. variables, as indicated by subscripts to the functions enclosed in the parentheses. These derivatives will be presumed evaluated at the appropriate coordinates of the nominal policies and trajectories.

The notation $\mathrm{QP}(L_i)$ will denote the operator which takes the quadratic part of the Taylor series expansion of $L_i(x_i, u_i)$ about $(\bar{u}_i, \bar{x}_i)$. Since the constant part $L_i(\bar{x}_i, \bar{u}_i)$ of the expansion does not play a role in the determination of the successor control policy, it is neglected (i.e., set to zero). The operator $\mathrm{QP}(\cdot)$, with respect to other functions of states and controls, is defined in exactly the same manner. Thus, for example, letting

$$\delta x_{n-1} = x_{n-1} - \bar{x}_{n-1}, \qquad \delta u_{n-1} = u_{n-1} - \bar{u}_{n-1},$$

we have

$$\begin{aligned}\mathrm{QP}(L_n(f_{n-1}(x_{n-1}, u_{n-1}), \bar{u}_n)) &= (L_n)_x[(f_{n-1})_x \delta x_{n-1} + (f_{n-1})_u \delta u_{n-1} \\ &+ \tfrac{1}{2}[(f_{n-1})_{xx} \delta x_{n-1}^2 + 2(f_{n-1})_{xu} \delta x_{n-1} \delta u_{n-1} + (f_{n-1})_{uu} \delta u_{n-1}^2]] \\ &+ \tfrac{1}{2}(L_n)_{xx}((f_{n-1})_x \delta x_{n-1} + (f_{n-1})_u \delta u_{n-1})^2. \end{aligned} \tag{3}$$

The DDP backward run begins by setting

$$\begin{aligned} &Q_n(x_n, u_n) = \mathrm{QP}(L_n(x_n, u_n)), \\ &\alpha_n = -(Q_n)_u/(Q_n)_{uu}, \qquad \beta_n = -(Q_n)_{xu}/(Q_n)_{uu}; \end{aligned} \tag{4}$$

and, for $i = n-1, \ldots, 1$, it proceeds recursively, according to the rule

$$\begin{aligned} Q_i(x_i, u_i) = \mathrm{QP}(&L_i(x_i, u_i) \\ &+ Q_{i+1}(f_i(x_i, u_i), \bar{u}_{i+1} + \alpha_{i+1} + \beta_{i+1}(f_i(x_i, u_i) - \bar{x}_{i+1}))), \end{aligned} \tag{5}$$

$$\alpha_i = -(Q_i)_u (Q_i)_{uu}^{-1}, \qquad \beta_i = -(Q_i)_{xu}(Q_i)_{uu}^{-1}. \tag{6}$$

Recall that the partial derivatives are to be evaluated at appropriate coordinates of the nominal policy and trajectory. Thus, for example,

$$(Q_i)_u = [(\partial/\partial u_i) Q_i(x_i, u_i)]_{x_i = \bar{x}_i, u_i = \bar{u}_i}.$$

For example, Ref. 10 gives recursive formulas for determining the coefficients of the quadratic Q_i in terms of those of Q_{i+1}. The basic idea of

(5) and (6) is that $Q_i(x_i, u_i)$ is a quadratic approximation of the value function from the state x_i onward, under the control u_i and the strategy

$$u_j(x_j) = \alpha_j + \beta_j(x_j - \bar{x}_j) + \bar{u}_j, \qquad j > i.$$

It is a simple matter to check that this strategy assures that the optimality condition

$$[(\partial/\partial u_j) Q_j(\bar{x}_j, u_j)]_{u_j = \bar{u}_j} = (Q_j)_{xu} \delta x_j + (Q_j)_{uu} \delta u_j + (Q_j)_u = 0 \tag{7}$$

is satisfied for every j and x_j. That is, $u_j(x_j)$ is the optimizer for the quadratic approximation $Q_j(x_j, u_j)$.

The linear feedback coefficients α_i, β_i, $i = 1, 2, \ldots, n$, having been determined according to the backward run construction above, the successor policy $u^D = \{u_i^D\}_{i=1}^n$ is determined in the DDP forward run, which consists of the recursive rule below:

$$x_1 = \bar{x}_1, \qquad u_1^D = \bar{u}_1 + \alpha_1; \tag{8}$$

and, for $i = 2, 3, \ldots, n$ and $\delta x_i = x_i - \bar{x}_i$,

$$(x_i = f_{i-1}(x_{i-1}, u_{i-1}^D), \qquad u_i^D = \bar{u}_i + \alpha_i + \beta_i \delta x_i. \tag{9}$$

This is, in essence, the prototypical DDP algorithm. The algorithm depends on the quantities $(Q_i)_{uu}$ being positive (or positive definite in the vector case). In the general case, it is possible to modify the algorithm to account for the failure of the positivity condition, and this matter will be discussed in Section 6. Also, as is pointed out in Ref. 9, by augmenting a damping factor to the rule (9), one can assure that DDP is globally convergent to a stationary point. The basic advantage of DDP, as opposed to Newton's method, is that the latter requires inversion of a matrix of order n, with operations count proportional to n^3, while DDP is a stagewise process with operations count linear to n.

3. Notation for Newton's Method

Newton's method determines

$$u^N = \{u_i^N\}_{i=1}^n$$

from $\bar{u}$ by solving the system of linear equations

$$\mathcal{V}(\bar{u})(u^N - \bar{u}) = -\nabla V(\bar{u}), \tag{10}$$

which represents simultaneous equations of the form

$$\sum_{j=1}^{n} v_{ij} \delta u_j^N + v_i = 0, \qquad 1 \le i \le n. \tag{11}$$

In (11),

$$\delta u_j^N = u_j^N - \bar{u}_j, \qquad v_i = (\partial/\partial u_i)\, V(u), \qquad v_{ij} = (\partial^2/\partial u_i \partial u_j)\, V(u).$$

This notation will serve us in the subsequent discussion. Our assumption that the partial derivatives are evaluated at appropriate coordinates of the nominal policy and trajectory is presumed in force here and in the remainder of this study.

The components of $\nabla V(\bar{u})$ are given explicitly in terms of the stagewise loss function and dynamics as follows:

$$v_i = (L_i)_u + \sum_{j=i+1}^{n} (L_j)_x (\partial x_j/\partial u_i), \qquad 1 \le i \le n, \tag{12}$$

where

$$\frac{\partial x_j}{\partial u_i} = \begin{cases} (f_{j-1})_x \cdots (f_{i+1})_x (f_i)_u, & j > i, \\ 0, & j \le i. \end{cases} \tag{13}$$

The components of the Hessian $\mathcal{V}(\bar{u})$ in (10) are similarly given by

$$v_{ij} = (L_i)_{uu}\delta_{ij} + (L_i)_{ux}(\partial x_i/\partial u_j) + (L_j)_{ux}(\partial x_j/\partial u_i) + \sum_{k=m+1}^{n} [(L_k)_{xx}(\partial x_k/\partial u_i)(\partial x_k/\partial u_j) + (L_k)_x(\partial^2 x_k/\partial u_i \partial u_j)], \tag{14}$$

where

$$m = \max\{i, j\}$$

and δ_{ij} is the Kronecker delta. One may observe from (14) that

$$\mathcal{V} = \{v_{ij}\}$$

can be a full matrix. We are unaware of any effective shortcuts to its inversion.

4. DDP, Newton's Method, and Quadratic Convergence

The purpose of this section is to carefully compare the structure of the DDP iteration with that of a multivariable Newton iteration applied to the optimal control problem (OCP) objective function $V(u)$. The investigations reported in this section will lead us to the following conclusions:

(i) DDP does not generally coincide with Newton's method.

(ii) If the Hessian $\mathcal{V}$ [defined in (14)] of the optimal control problem is positive definite, then the convergence rate of DDP is quadratic.

(iii) When DDP takes the quadratic part of the value function accord-

ing to Eqs. (3) and (5), it uses all the second derivatives of the dynamical relationship. From (14), however, one may verify that Newton's method does not use, for example, $\partial^2 x_k/\partial^2 x_i$. Because of this additional Newton's method approximation, one might anticipate that DDP ought usually to be slightly more accurate.

The proof in this section is on the lengthy side, and some constructions, when introduced, may appear poorly motivated. For that reason, we supply below an overview of the quadratic convergence proof.

Overview of the Quadratic Convergence Proof. The proof proceeds by constructing a linear strategy [that is, one resembling the DDP strategy $u_j(x_j)$ introduced in Section 2], which determines a policy, to be denoted by $\hat{u}$, which is intermediate between the DDP policy u^D and the Newton policy u^N. This strategy is obtained by examination of a Gaussian elimination construction for Newton's method. By use of a matrix perturbation formula, we establish that $\|\hat{u}-u^N\|$ is $O(\delta^2)$, where δ is a measure of gradients of approximants of the value function. The developments are the contents of Lemmas 4.1 and 4.2. By comparison of quadratic expansions of the value function induced by the intermediate strategy with those induced by DDP, we then show (Lemma 4.3) that $\|\hat{u}-u^D\|$ is also $O(\delta^2)$. The rest of the proof consists in showing that the norm of the Newton increment $u^N-\bar{u}$ is proportional to δ; as a consequence, the quadratic convergence of Newton's method implies quadratic convergence of DDP.

Our derivation begins by viewing the Gaussian elimination of a Newton iteration within the framework of quadratic expansions such as used in DDP. Triangularize the Newton equation (10), which we repeat below

$$\mathcal{V}(u^N-\bar{u})=-\nabla V,$$

by starting at the bottom row and working up. We will define

$$\mathcal{V}^{(n)}=\{V_{ij}^{(n)}\}_{ij}$$

to be $\mathcal{V}$, and $\mathcal{V}^{(n-1)}$, $\mathcal{V}^{(n-2)}$, ..., $\mathcal{V}^{(1)}$ will denote the successive coefficient matrices obtained in the Gaussian elimination triangularization. A later formula [Eq. (23)] will make the construction of $\mathcal{V}^{(k)}$ from $\mathcal{V}^{(k+1)}$ explicit. Similarly,

$$v^{(n)}=(v_j^{(n)})$$

will denote the gradient vector ∇V above, and $-v^{(n-2)}, \ldots, -v^{(1)}$ will be the successive coefficient vectors obtained as the elimination proceeds.

In the discussion to follow, δx_t, δu_t, etc., will denote increments such as $x_t-\bar{x}_t$, $u_t-\bar{u}_t$, etc. We will assume throughout that the Hessian $\mathcal{V}=\mathcal{V}(\bar{u})$ is nonsingular. Other notation, such as $(L_t)_u$, $(Q_t)_u$, will be as in the

preceding sections. As before, such partial derivatives are presumed evaluated at nominal states and controls.

Let $\hat{Q}_k(x_k, u_k)$, $k = n, n-1, \ldots, 1$, be recursively defined according to

$$\hat{Q}_n(x_n, u_n) = L_n(x_n, u_n), \tag{15}$$

$$\hat{Q}_k(x_k, u_k) = L_k(x_k, u_k) + \hat{Q}_{k+1}(f_k(x_k, u_k), \bar{u}_{k+1} + \hat{\alpha}_{k+1} + \hat{\beta}_{k+1}(f_k(x_k, u_k) - \bar{x}_{k+1})), \tag{16}$$

with

$$\hat{\alpha}_{k+1} \triangleq -(\hat{Q}_{k+1})_u/(\hat{Q}_{k+1})_{uu}, \qquad \hat{\beta}_{k+1} \triangleq -(\hat{Q}_{k+1})_{xu}/(\hat{Q}_{k+1})_{uu}. \tag{17}$$

That is,

$$\hat{Q}_k(x_k, u_k) = L_k(x_k, u_k) + \sum_{t=k+1}^{n} L_t(\hat{x}_t, \phi_t(\hat{x}_t)), \tag{18}$$

where $\{\phi_t\}$ is the strategy

$$\phi_t(x) = \bar{u}_t + \hat{\alpha}_t + \hat{\beta}_t(x - \bar{x}_t),$$

and $\{\hat{x}_t\}_{t=k+1}^{n}$ is the trajectory determined by $\{\phi_t\}_{t=k+1}^{n}$, $\hat{x}_{k+1} = f(x_k, u_k)$, and (2). With $\hat{Q}_k(x_k, u_k)$ so defined, we will find extensive use for the function

$$N_k(u_1, \ldots, u_k) = \sum_{t=1}^{k-1} L_t(x_t, u_t) + \hat{Q}_k(x_k, u_k). \tag{19}$$

From comparing the constructed $\hat{\alpha}_k$, $\hat{\beta}_k$, $\hat{Q}_k$ with similarly constructed α_k, β_k, Q_k [Eqs. (5) and (6)] for the DDP approach, one will see that $\hat{Q}_k(x_k, u_k)$ bears great similarity to the optimal value function $Q_k(x_k, u_k)$ of DDP. But, as is seen in the lemma to follow, the derivatives of N_k, which is related to $\hat{Q}_k$, through (19), are close to the Newton coefficient vectors and matrices and their partial triangularizations.

Define ∇N_k and $\mathcal{N}^{(k)}$ to be, respectively, the gradient and Hessian matrix for $N_k(u_1, \ldots, u_k)$, evaluated at the nominal policy $\bar{u}$; then, represent the elements of ∇N_k by $\{n_j^{(k)}\}$ and those of $\mathcal{N}^{(k)}$ by $\{n_{ij}^{(k)}\}$.

Lemma 4.1. The coordinates of ∇N_k and $\mathcal{N}^{(k)}$ are related to those of the partially triangularized matrix $\mathcal{V}^{(k)}$ by

$$n_{ij}^{(k)} = v_{ij}^{(k)} + O(\delta), \qquad 1 \le i, j \le k, \tag{20}$$

$$n_j^{(k)} = v_j^{(k)} + O(\delta^2), \qquad 1 \le j \le k, \tag{21}$$

where

$$\delta = \max_{1 \le t \le n} \{|(\hat{Q}_t)_u|\}. \tag{22}$$

Remark 4.1. By way of motivation for this definition of δ, it will transpire in the development to follow that $\bar{u}$ is a stationary policy if and only if $\delta = 0$. Thus, examination of order-of-δ-type relations will be useful. For instance, if δu^N denotes the policy increment determined by the Newton iteration, then, as we will see,

$$\delta u^N = O(\delta).$$

Of course, δ is the uniform norm of the first derivatives, with respect to the control, of the value function approximations.

Proof of Lemma 4.1. Of course,

$$\mathcal{N}^{(n)} = \mathcal{V}.$$

From the definition of Gaussian elimination, but with the modification that we seek to end up with a lower (rather than upper) triangular matrix, $\mathcal{V}^{(k)}$ is determined by

$$v_{ij}^{(k)} = v_{ij}^{(k+1)} - (v_{i,k+1}^{(k+1)} v_{k+1,j}^{(k+1)} / v_{k+1,k+1}^{(k+1)}). \tag{23}$$

Assertion (20) follows if only we can show that

$$n_{ij}^{(k)} - n_{ij}^{(k+1)} = -n_{i,k+1}^{(k+1)} n_{k+1,j}^{(k+1)} / n_{k+1,k+1}^{(k+1)} + O(\delta). \tag{24}$$

By definition and then power series expansion in δx_{k+1}, we have

$$\begin{aligned} n_{ij}^{(k)} - n_{ij}^{(k+1)} &= (\partial^2 / \partial u_i \, \partial u_j)[\hat{Q}_{k+1}(x_{k+1}, \bar{u}_{k+1} + \hat{\alpha}_{k+1} + \hat{\beta}_{k+1} \delta x_{k+1}) \\ &\quad - \hat{Q}_{k+1}(x_{k+1}, \bar{u}_{k+1})]_{\bar{u}} \\ &= (\partial^2 / \partial u_i \, \partial u_j)[(\hat{Q}_{k+1})_u (\hat{\alpha}_{k+1} + \hat{\beta}_{k+1} \delta x_{k+1}) \\ &\quad + (\hat{Q}_{k+1})_{xu} \delta x_{k+1} (\hat{\alpha}_{k+1} + \hat{\beta}_{k+1} \delta x_{k+1}) \\ &\quad + \tfrac{1}{2} (\hat{Q}_{k+1})_{uu} (\hat{\alpha}_{k+1} + \hat{\beta}_{k+1} \delta x_{k+1})^2 + O(\delta x_{k+1}^3)]_{\bar{u}}. \end{aligned}$$

From (22), $(\hat{Q}_{k+1})_u$ is $O(\delta)$; and so, in view of (17),

$$\begin{aligned} n_{ij}^{(k)} - n_{ij}^{(k+1)} &= [2(\hat{Q}_{k+1})_{xu} \hat{\beta}_{k+1} + (\hat{Q}_{k+1})_{uu} \hat{\beta}_{k+1}^2] \\ &\quad \times (\partial x_{k+1} / \partial u_i)(\partial x_{k+1} / \partial u_j) + O(\delta) \\ &= \frac{-[(\hat{Q}_{k+1})_{xu} (\partial x_{k+1} / \partial u_i)][(\hat{Q}_{k+1})_{uu} (\partial x_{k+1} / \partial u_j)]}{(\hat{Q}_{k+1})_{uu}} + O(\delta), \end{aligned}$$

which is tantamount to (24). We leave it to the reader to similarly verify (21). □

Let

$$u^{(k)} = \{u_j^{(k)}\}_{j=1}^k$$

denote the successor policy determined by a Newton iteration for $N_k(u_1, \ldots, u_k)$. That is,

$$\mathcal{N}^{(k)} \delta u^{(k)} = \nabla N_k. \tag{25}$$

Now, the Newton increment

$$\delta u^N = u^N - \bar{u}$$

not only satisfies (10), but also all the partially Gaussian triangularized systems

$$\mathcal{V}^{(k)} \delta u^N = -v^{(k)}, \qquad 1 \le k \le n. \tag{26}$$

Our next result shows that the solutions of (25) and (26) are close.

Lemma 4.2. For $k = 1, \ldots, n$ and $1 \le i \le k$,

$$u_i^N - u_i^{(k)} = O(\delta^2).$$

Proof. Let

$$\mathcal{A}x = b \quad \text{and} \quad \hat{\mathcal{A}}\hat{x} = \hat{b}$$

denote two linear systems of like orders. A standard perturbation formula (e.g., Ref. 14, p. 214) bounds the distance between the solutions in terms of the norms of the differences of the coefficient matrices and vectors and in terms of γ, the condition number of $\mathcal{A}$. Specifically, for

$$\|\mathcal{A} - \hat{\mathcal{A}}\| < \|\mathcal{A}\| / \gamma,$$

we have

$$\|x - \hat{x}\| \le \frac{\gamma[\|b - \hat{b}\| \, \|x\| + \|\mathcal{A} - \hat{\mathcal{A}}\|]}{\|\mathcal{A}\| - \gamma \|\mathcal{A} - \hat{\mathcal{A}}\|}. \tag{27}$$

Apply this to the first k rows of the partially triangularized system (26), which of course determine the first k coordinates of u^N. Then, for δ sufficiently small, we have

$$\|\{u_i^{(k)} - u_i^N\}\| \le \frac{\gamma[\|\{n_{ij}^{(k)} - v_{ij}^{(k)}\}\| \, \|\{\delta u_i^N\}\| + \|\{n_j^{(k)} - v_j^{(k)}\}\|]}{\|\mathcal{N}^{(k)}\| - \gamma \|\{n_{ij}^{(k)} - v_{ij}^{(k)}\}\|}. \tag{28}$$

In (28), γ denotes the condition number of $\mathcal{N}^{(k)}$, which must be finite if $\mathcal{V}$ is nonsingular and δ is sufficiently small, and i and j range from 1 to k. The last addend in the numerator of (28) is $O(\delta^2)$ from Lemma 4.1, and the other addend is $O(\delta^2)$ from that lemma if only it can be shown that

$$\delta u^N = O(\delta). \tag{29}$$

Toward that end, note that, in view of (22),

$$n_k^{(k)} = (\hat{Q}_k)_u = O(\delta).$$

Also, from Lemma 4.1, we have that, for $1 \le k \le n$,

$$v_k^{(k)} = n_k^{(k)} + O(\delta^2), \tag{30}$$

and $v_j^{(j)}$ is, by definition of the Gaussian elimination process, a linear combination of the terms $v_m^{(k)}$, $m \ge j$. Thus, the final Gauss triangularized system can be seen to have the form

$$\{v_{ji}^{(1)}\}\delta u^N = -\{v_j^{(j)}\}, \tag{31}$$

where i and j range from 1 to n. But (31) implies that δu^N must be $O(\delta)$. □

Define $\hat{u} = \{\hat{u}_i\}$ by the condition that

$$\hat{u}_k = u_k^{(k)}, \qquad 1 \le k \le n, \tag{32}$$

with $u^{(k)}$ as in the preceding lemma.

Lemma 4.3. If u^D denotes the DDP successor policy,

$$u^D - \hat{u} = O(\delta^2). \tag{33}$$

Proof. Consider the last row of (25), namely,

$$\sum_{j=1}^{k-1} n_{kj}^{(k)} \delta u_j + n_{kk}^{(k)} \delta u_k = -(\hat{Q}_k)_u. \tag{34}$$

From the defining conditions of $\mathcal{N}^{(k)}$ and the representation (19) of Hessian coefficients, (34) represents, in fact,

$$\sum_{j=1}^{k-1} (\hat{Q}_k)_{ux} (\partial x_k / \partial u_j) \delta u_j + (\hat{Q}_k)_{uu} \delta u_k = -(\hat{Q}_k)_u. \tag{35}$$

Using the definition

$$\delta x_k^l \equiv \sum_{j=1}^{k-1} (\partial x_k / \partial u_j) \delta u_j, \tag{36}$$

(35) may be rewritten as

$$(\hat{Q}_k)_{ux} \delta x_k^l + (\hat{Q}_k)_{uu} \delta u_k = -(\hat{Q}_k)_u, \tag{37}$$

or, in terms of (17),

$$\delta u_k^{(k)} = \delta \hat{u}_k = \hat{\alpha}_k + \hat{\beta}_k \delta x_k^l. \tag{38}$$

The lemma will be demonstrated by showing that, with respect to the corresponding DDP quantities α_k and β_k [see (6)], for $k = 1, \ldots, n$,

$$\hat{\alpha}_k = \alpha_k + O(\delta^2), \qquad \hat{\beta}_k = \beta_k + O(\delta); \tag{39}$$

and for

$$\delta x_k^l = \sum_{j=1}^{k-1} (\partial x_k / \partial u_j) \delta u_j^{(k)},$$

with $u^{(k)}$ as in (32),

$$\delta x_k^l - \delta x_k^D = O(\delta^2), \qquad \delta x_k^l = O(\delta), \tag{40}$$

x_k^D denoting the kth state determined by the DDP successor policy u^D. First, we verify (39). Observe that

$$\hat{Q}_n = L_n, \qquad Q_n = \mathrm{QP}(L_n),$$

and so, from (6) and (17), (39) holds. In fact, for $k = n$,

$$\hat{\alpha}_k = \alpha_k, \qquad \hat{\beta}_k = \beta_k.$$

Notice that Q_k and $\hat{Q}_k$ are determined recursively from the relations

$$\hat{Q}_k(x, u) = L_k(x, u) + \hat{Q}_{k+1}(f_k(x, u), \bar{u}_{k+1} + \hat{\alpha}_{k+1} + \hat{\beta}_{k+1}(f_k(x, u) - \bar{x}_{k+1})), \tag{41}$$

$$Q_k(x, u) = \mathrm{QP}[L_k(x, u) + Q_{k+1}(f_k(x, u), \bar{u}_{k+1} + \alpha_{k+1} + \beta_{k+1}(f_k(x, u) - \bar{x}_{k+1}))]. \tag{42}$$

From these relations, it is a straightforward but tedious matter to verify that, if (39) and

$$(\hat{Q}_k)_u = (Q_k)_u + O(\delta^2), \tag{43a}$$

$$(\hat{Q}_k)_x = (Q_k)_x + O(\delta^2), \tag{43b}$$

$$(\hat{Q}_k)_{uu} = (Q_k)_{uu} + O(\delta), \tag{43c}$$

$$(\hat{Q}_k)_{xu} = (Q_k)_{xu} + O(\delta), \tag{43d}$$

$$(\hat{Q}_k)_{xx} = (Q_k)_{xx} + O(\delta) \tag{43e}$$

hold for $k = M + 1$, then (39) and therefore (43) hold for $k = M$. Thus, (39) holds for all $k = 1, \ldots, n$. We note that the reader who verifies (43) will be rewarded for his trouble by seeing that the DDP strategy accounts for terms neglected in the $\hat{\alpha}_k$, $\hat{\beta}_k$ strategy.

Now, only verification of (40) remains. For $k=1$, (40), and thus (33) hold, since x_1 is the given initial state and

$$x_1^D = \hat{x}_1 = x_1.$$

Suppose, inductively, that (40) and hence (33) hold for $k=M$. Let $u^{(M)}$ and $u^{(M+1)}$ denote the solution vectors of (25) for $k=M$ and $M+1$, respectively. Let $x_j^l(u)$ be the linearized state at stage j determined by the policy u through (36), and let $x_j(u)$ be the state determined by the dynamical equations (2). It is clear that

$$x_{M+1}(u^D) = (f_M)_x \delta x_M^D + (f_M)_u \delta u_M^D + O(\delta^2),$$

$$\begin{aligned}\hat{x}_{M+1}^l &= x_{M+1}^l(u^{(M+1)}) \\ &= (f_M)_x[\delta x_M^l(u^{(M)}) + (\delta x_M^l(u^{(M+1)}) - \delta x_M^{(M)}(u^{(M)})] \\ &\quad + (f_M)_u \delta u_M^{(M+1)},\end{aligned}$$

so (40) holds for $k=M+1$ if only it can be established that

$$\bar{u}_M^{(M+1)} - \bar{u}_M^{(M)} = O(\delta^2).$$

But this is so because, in view of Lemma 4.2, they both differ from the corresponding Newton control u_M^N by $O(\delta^2)$. Note that, from (37), δu^N and thus δx_k^l is $O(\delta)$. □

On the basis of the preceding lemmas, we are able to state and prove the central result of this study.

Theorem 4.1. If u^* is a stationary policy for the optimal control problem, and if the Hessian matrix $\mathcal{V}(u^*)$ is nonsingular, then the convergence rate of DDP to u^* is quadratic.

Proof. Let u^N and u^D denote the Newton and DDP successors of a nominal policy $\bar{u}$. From standard results in Newton–Kantorovich theory (e.g., Ref. 15, Section 12.6), one may conclude that, under the conditions of the theorem and our smoothness assumptions concerning L_t and f_t introduced in Section 1, there exists a neighborhood U_1 of u^* and a constant C_1 such that

$$\|u^N - u^*\| \leq C_1 \|\bar{u} - u^*\|^2, \tag{44}$$

for all $u \in U_1$. A consequence of (44) is that, for some number C_2 and U_2 a neighborhood of u^*, for any $\bar{u} \in U_2$,

$$\|\bar{u} - u^N\| \leq C_2 \|\bar{u} - u^*\|. \tag{45}$$

Finally, one may check that, under the smoothness assumptions of this

study, the constants for the order-of-δ approximations appearing in the preceding lemmas will be uniformly bounded on a small enough neighborhood U_3 of u^*. From this and Lemma 4.3, we may conclude that, for $\bar{u} \in U_3$,

$$\|u^D - u^N\| \leq C_3 \|u^N - \bar{u}\|^2 \tag{46}$$

and that, for

$$\bar{u} \in \bigcap_{j=1}^{3} U_j,$$

(44), (45), (46) hold simultaneously. But this implies that

$$\begin{aligned} \|u^D - u^*\| &\leq \|u^D - u^N\| + \|u^N - u^*\| \\ &\leq C_3 \|u^N - \bar{u}\|^2 + C_1 \|\bar{u} - u^*\|^2 \\ &\leq (C_3 C_2^2 + C_1) \|\bar{u} - u^*\|^2. \end{aligned} \tag{47}$$

□

There are two useful results that spring from the preceding analysis.

Corollary 4.1. If the dynamical equation (2) of the optimal control problem is linear, then the DDP and Newton's method coincide.

Proof. One may check that, in the linear dynamics case, the $O(\delta)$ term in (25) disappears and

$$\mathcal{V}^{(k)} = \mathcal{N}^{(k)}, \quad N_k^{(k)} = n_k^{(k)}, \qquad 1 \leq k \leq n. \tag{48}$$

Now, one checks that, up to quadratic coefficients, Q_k and $\hat{Q}_k$ coincide and that $x_k = x_k^l$, so that, for every k,

$$\alpha_k = \hat{\alpha}_k, \qquad \beta_k = \hat{\beta}_k.$$

Therefore, $\hat{u}_k$ of (38) equals u_k^D, determined by (9). But now we may conclude from (48) that

$$\hat{u}_k = u_k^N.$$

□

The next statement is the basis for an easy DDP scheme to check for local positive definiteness of the Hessian $\mathcal{V}(u^*)$.

Corollary 4.2. In a sufficiently small neighborhood of a stationary policy u^*, the Hessian $\mathcal{V}(u^*)$ is positive definite if and only if the terms $(Q_t)_{uu}$ are positive for all t, $1 \leq t \leq n$.

Proof. It is known (e.g., Ref. 16, p. 163) that symmetric matrices are positive definite if and only if the diagonal of the triangularized matrix

contains only positive elements. Thus, $\mathcal{V}(u^*)$ is positive definite if and only if

$$v_{kk}^{(k)} > 0, \qquad 1 \leq k \leq n.$$

But, from the preceding lemmas, one may conclude that

$$v_{kk}^{(k)} = (Q_k)_{uu} + O(\delta)$$

and, of course, δ can be made arbitrarily small by taking small enough neighborhoods of u^*. □

As a final comment in this section, we note that one cannot obtain quadratic convergence by simply quadraticizing the single-stage loss function and linearizing the dynamics about the current nominal policy, and then solving the resulting linear dynamics-quadratic performance problem to obtain the successor policy. It is the linearization of the dynamics that dooms this plan; for one sees from (14) that, after linearization, the term v_{ij} of the Hessian matrix $\mathcal{V}(\bar{u})$ no longer contains the terms $(L_k)_x \partial^2 x_k / \partial u_i \partial u_j$, and these neglected terms do not vanish in a neighborhood of the stationary policy. Thus, the Hessian matrix of the original optimal control problem will not typically become close to that of the optimal control problem with the linearized dynamics; but, from examination of Taylor's series expansions, one sees that such approximation of the Hessian $\mathcal{V}(u^*)$ is necessary for quadratic convergence.

5. Examples

For the purpose of illustrating the points raised in this paper, we pose the simple nonlinear programming problem, studied in Ref. 9, of minimizing

$$V(u) = \sum_{i=1}^{n} \left[\tfrac{1}{2} u_i^2 + \tfrac{1}{2} \left(\sum_{j=1}^{i} \exp(u_j) \right)^2 \right]. \tag{49}$$

When we formulate this as a discrete optimal control problem, we are required to minimize

$$V(u) = \sum_{i=1}^{n} \tfrac{1}{2} [u_i^2 + (u_i + \exp(u_i))^2], \tag{50}$$

where

$$x_{i+1} = x_i + \exp(u_i), \qquad i = 1, \ldots, n-1, \qquad x_1 = 0. \tag{51}$$

Because of the nonlinearity in (51), we can demonstrate the difference between the DDP and Newton's method. Furthermore, the Hessian of V is nonsingular, and thus the quadratic convergence of both methods will

Table 1. Progression of DDP parameters for Example 5.1.

Iteration	α_1	β_2	δx_2	δu_2
1	−0.4762	−0.2500	−0.7121	−0.4053
2	−0.2242	−0.2895	−0.2238	−0.1608
3	−0.0309	−0.2947	−0.0238	−0.0153
4	−0.0005	−0.2952	−0.0003	−0.0001

manifest itself. By choosing different values of n, we also obtain problems of different sizes.

We now consider the two cases $n=2$ and $n=5$ and always choose the nominal control for the intial iteration as

$$\bar{u}_i = 0, \qquad i = 1, \ldots, n.$$

Example 5.1. The minimization problem is solved by using DDP and the Newton method which solves the system of linear equations (10). Some of the DDP parameters are listed in Table 1. In Table 2, the Newton method iterates are compared with those obtained by DDP.

Example 5.2. Our second example, with $n=5$ in (49), was also solved by the DDP and Newton algorithms. The DDP and Newton successor controls after the first iteration are compared in Table 3. In Table 4, the ultimate quadratic convergence of these methods is again observed.

6. Modified Newton Method and DDP Method

In this section, the more general case of a control vector of dimension m is considered, while n still denotes the number of stages.

In many important instances, it is known that the optimal control problem objective function $V(u)$ is locally convex in neighborhoods of

Table 2. Iterates for DDP and Newton's method for Example 5.1.

	DDP			Newton's method		
Iteration	u_1	u_2	$V(u)$	u_1	u_2	$V(u)$
0	0.0	0.0	2.5	0.0	0.0	2.5
1	−0.4762	−0.4053	1.2178	−0.4348	−0.3913	1.2566
2	−0.7004	−0.5661	1.0949	−0.6813	−0.5596	1.0971
3	−0.7313	−0.5814	1.0934	−0.7304	−0.5812	1.0934
4	−0.7318	−0.5815	1.0934	−0.7318	−0.5815	1.0934

Table 3. First iteration for Example 5.2.

Control	DDP	Newton's method
u_1	−0.6679	−0.4873
u_2	−0.6056	−0.4856
u_3	−0.5580	−0.4813
u_4	−0.5154	−0.4711
u_5	−0.4612	−0.4393

Table 4. Values of the performance index (49) for Example 5.2.

Iteration	DDP	Newton's method
0	27.5	27.5
1	9.39112	11.12318
2	6.09578	6.60392
3	5.88887	5.91434
4	5.88762	5.88767
5	5.88762	5.88762

stationary policies, but not necessarily elsewhere. For the solution of such problems, use of the modified Newton's method (Ref. 15) is popular and sensible. The basic procedure consists of choosing the successor policy to satisfy

$$[\mathcal{V}(u)+\lambda\mathcal{I}]\delta u = -\nabla V,$$

where $\mathcal{I}$ is the identity matrix and the number λ is chosen to be large enough to guarantee that the matrix $\mathcal{V}(u)+\lambda\mathcal{I}$ has positive eigenvalues. The idea is that, with this modification, $-(\mathcal{V}(u)+\lambda\mathcal{I})^{-1}\nabla V$ is guaranteed to be a descent direction and that, possibly with the help of a damping factor, the initial iterations will descend to policies in the vicinity of the stationary policy. At this stage, pure Newton iterations can be employed to obtain rapid convergence. The expensive part of this procedure, for large systems, is the determination of whether the eigenvalues of $\mathcal{V}(u)$ are all positive.

On the other hand, differential dynamic programming provides a very simple and inexpensive way for carrying out the modified Newton method idea. It is easy to check that a DDP successor policy δu^D is a descent direction whenever the matrices $\{(Q_k)_{uu}\}_{k=1}^n$ are all positive definite. The

modified DDP rule is to replace (6) by

$$\alpha_k = -[(Q_k)_{uu} + \lambda_k \mathcal{I}]^{-1}(Q_k)_u, \tag{52a}$$

$$\beta_k = -[(Q_k)_{uu} + \lambda_k \mathcal{I}]^{-1}(Q_k)_{xu}, \tag{52b}$$

where λ_k is a positive number large enough to assure that the matrix $(Q_k)_{uu} + \lambda_k \mathcal{I}$ is positive definite. Thus, the modified Newton's approach requires checking a single matrix of order $m \times m$, m being the dimension of the control vector, for positive eigenvalues, whereas the modified DDP technique requires checking n matrices, each of order m.

7. Conclusions

We have examined the relationship between Newton's method and differential dynamic programming and have proved the quadratic convergence of DDP. This quadratic convergence was also borne out by the computational examples. One conclusion that we arrived at is that any discrete optimal control method which does not incorporate second derivatives of the dynamical equations cannot achieve quadratic convergence.

In this paper, our emphasis was on DDP being a second-order method and also a stagewise method. Therefore, it achieves a rapid convergence rate without the need to solve large linear equations. For the practical implementation of DDP, one requires a procedure to guarantee that descent steps are generated, and one requires a mechanism to choose a steplength. Global convergence issues and the performance of DDP on other test problems were addressed in another study (Ref. 9).

References

1. HALKIN, H., *A Maximum Principle of Pontryagin Type for Systems Described by Nonlinear Difference Equations*, SIAM Journal on Control, Vol. 4, pp. 90–111, 1966.
2. MAYNE, D., *A Second-Order Gradient Method for Determining Optimal Trajectories of Nonlinear Discrete-Time Systems*, International Journal on Control, Vol. 3, pp. 85–95, 1966.
3. JACOBSON, D. H., and MAYNE, D. Q., *Differential Dynamic Programming*, American Elsevier, New York, New York, 1970.
4. DYER, P., and MCREYNOLDS, S., *The Computational Theory of Optimal Control*, Academic Press, New York, New York, 1979.
5. OHNO, K., *A New Approach of Differential Dynamic Programming for Discrete-Time Systems*, IEEE Transactions on Automatic Control, Vol. AC-23, pp. 37–47, 1978.

6. BELLMAN, R., and DREYFUS, S., *Applied Dynamic Programming*, Princeton University Press, Princeton, New Jersey, 1962.
7. LARSON, R., *State Increment Dynamic Programming*, Elsevier, New York, New York, 1968.
8. YAKOWITZ, S., *Convergence Bounds for the State Increment Dynamic Programming Method*, Automatica, Vol. 19, pp. 53–60, 1983.
9. MURRAY, M., and YAKOWITZ, S., *The Application of Optimal Control Methodology to Nonlinear Programming Problems*, Mathematical Programming, Vol. 21, pp. 331–347, 1981.
10. YAKOWITZ, S., and RUTHERFORD, B., *Computational Aspects of Differential Dynamic Programming*, Applied Mathematics and Computation (to appear).
11. MURRAY, D. M., and YAKOWITZ, S. J., *Constrained Differential Dynamic Programming, with Application to Multi-Reservoir Control*, Water Resource Research, Vol. 15, pp. 1017–1027, 1979.
12. YAKOWITZ, S., *Dynamic Programming Applications in Water Resources*, Water Resource Research, Vol. 18, pp. 673–698, 1982.
13. MURRAY, D. M., *Differential Dynamic Programming for the Efficient Solution of Optimal Control Problems*, University of Arizona, PhD Thesis, 1978.
14. SZIDAROVSZKY, F., and YAKOWITZ, S., *Principles and Procedures of Numerical Analysis*, Plenum Press, New York, New York, 1978.
15. ORTEGA, J., and RHEINBOLDT, W., *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, New York, 1970.
16. DAHLQUIST, G., and BJÖRCK, A., *Numerical Methods*, Prentice-Hall, Englewood Cliffs, New Jersey, 1973.