Control
and Cybernetics
VOL. 17 (1988) No. 2–3

Theoretical and computational advances in differential dynamic programming

by

S. YAKOWITZ

The University of Arizona
Systems and Industrial Engineering
Civil Engineering Bldg., Rm. 303
Tucson, Arizona 85721, USA

This survey begins with an overview of the differential dynamic programming (DDP) idea. The review concentrates on efforts by Ohno, the author, and his colleagues, toward understanding the convergence properties of the classical DDP algorithm and its close variants. Computational evidence of the power of this approach is cited. In my view, the essence of the DDP concept is stagewise application of nonlinear programming algorithms. I cite recent accomplishments in bringing quasi-Newton techniques into this framework, and outline some ongoing investigations into constrained DDP.

1. Introduction

The central aim of this study is to give a self-contained survey of developments over the past decade regarding discrete-time differential dynamic programming. A motivation for this survey is that the source papers on which the theory is founded are scattered through several different journals; it is not easy to sort through these works to gain a rounded view of the fairly comprehensive picture that is emerging. This survey additionally provides the author with an excuse to conjecture on the strengths and limitations of the differential dynamic programming idea, and to offer views concerning where the limitations are fundamental, or alternatively exist because apparently the issues have not been examined carefully.

The setting for differential dynamic programming (DDP) methodology, up to this point, has been the initial-state, finite-horizon problem. Let us present some notation. The state and input spaces willl be presumed to be respectively the spaces of real n and m-dimensional vectors. A stagewise loss function $g(x, u, t)$ and a state transition function $T(x, u, t)$ is presumed

specified. The specified initial state is presumed designated by $x(1)$, and the terminal decision time by N.

Given any control vector $\mathbf{u} = (u(1), \ldots, u(N))$ of inputs, a state trajectory $\mathbf{x} = (x(1), x(2), \ldots, x(N+1))$ is thereby determined by the recursive rule,

$$x(t+1) = T(x(t), u(t), t), \quad 1 \leqslant t \leqslant N, \tag{1.1}$$

(recalling that $x(1)$ is specified). This, in turn, engenders an overall process loss

$$J(\mathbf{u}) = \sum_{t=1}^{N} g(x(t), u(t), t). \tag{1.2}$$

Late I will add the possibility that the available inputs are constrained, but for the time being, I discuss DDP ideas in as simple a setting as possible. DDP depends on smoothness properties of the control process functions. Thus for now, we presume $g(\ )$ and $T(\ )$ are twice-continuously differentiable with respect to states and inputs.

Toward motivating differential dynamic programming, I assert what I believe to be the prototypical dynamic programming (DP) algorithms for finite-horizon, unconstrained optimal control problems, and then pinpoint why this algorithm is not directly computer-realizable.

The Fundamental Dynamic Programming Algorithm

One initializes by defining

$$V(x; N+1) = 0, \text{ all } x.$$

Then the backward DP recursion proceeds by defining, for $t = N, N-1, \ldots, 1$, the *Backward DP Recursion Step* for every state x:

$$V(x; t) = \min_{u} [g(x, u, t) + V(T(x, u, t), t+1)], \tag{1.3}$$

and

$$S(x; t) = u^*,$$

where u^* is minimizer of the right side of (1.3). When $S(x; t)$ has been defined for $t = N-1, N-2, \ldots, 1$, the backward recursion is complete.

Set $x^*(1) = x(1)$, the prescribed initial state. For $t = 1, 2, \ldots, N$, perform the *Forward DP Recursion Step*:

$$u^*(t) = S(x^*(t), t), \; x^*(t+1) = T(x^*(t), u^*(t), t). \tag{1.4}$$

Then it is a simple matter to show that if this plan can be implemented, $\mathbf{u}^* = (u^*(1), \ldots, u^*(N))$ is an optimal control.

The drawback to executing this plan is that for most optimal control functions $g(\ )$ and $T(\ )$, it is not an algorithm in the sense of Turing machines or recursive functions. One can anticipate that (1.3) cannot be

implemented, and even if it could be, the reeded optional return and strategy functions $V(\cdot; t)$ and $S(\cdot; t)$ could not be stored for further use.

As will be revealed next, differential dynamic programming is a means of numerical approximation to the DP algorithm. It bears a close resemblance to Newton's method, and for $N = 1$, would, in fact, coincide with Newton's method.

2. The fundamental DDP algorithm

The algorithm we now give is due to Mayne (1966) (also Chapter 4 of Jacobson and Mayne (1970), where the name "differential dynamic programming" is attached to the method). The presentation to flow is based on an algorithmic exposition in Yakowitz and Rutherford (1984).

Differential dynamic programming proceeds much like the traditional formal dynamic programming algorithm, except that at each stage, the optimal return function from the next stage onward as well as the loss for that stage are replaced by their quadratic approximations about the current control and trajectory. Toward making this idea precise, let $\mathbf{u}^C = (u^C(1), u^C(2), \ldots, u^C(N))$ denote the "current" control sequence, and $\mathbf{x} = (x^C(1), x^C(2), \ldots, x^C(N))$ the state trajectory induced by $\mathbf{u}^C$ and the initial state $x(1)$. For any function $Q(\mathbf{x}, \mathbf{u})$ defined on controls and trajectories we will let $QP(Q(\mathbf{x}, \mathbf{u}))$ denote the linear and quadratic (but not the constant) parts of the Taylor's series expansion of $Q(\ )$ about $(\mathbf{u}^C, \mathbf{x}^C)$.

The DDP backward recursion commences at decision time N by construction of the quadratic (in $\delta x = (x - x_N^C)$, $\delta u = (u - u_N^C)$), function

$$L(x, u, N) = QP(g(x, u, N)) = (1/2)\,\delta x^T (g_{xx})\,\delta x + \\ + \delta x^T (g_{xu})\,\delta u + (1/2)\,\delta u^T (g_{uu})\,\delta u + (g_u)\,\delta u + (g_x)\,\delta x. \qquad (2.1)$$

The gradients and Hessians of $g(x, u, N)$ are evaluated at x_N^C and u_N^C presumed I find it useful to represent the above equation in a more compact fashion as

$$L(x, u, N) = \delta x^T A_N\,\delta x + \delta u^T B_N\,\delta x + \\ + \delta u^T C_N\,\delta u + D_N^T\,\delta u + E_N^T\,\delta x, \qquad (2.2)$$

where δx and δu are state and input perturbations $(x - x_N^C)$ and $(u - u_N^C)$, respectively, and the terms $A_N, B_N, \ldots$, can be read off (2.1) by comparing coefficients of the perturbations, That is,

$$A_N = (1/2)\,g_{xx},\ B_N = g_{xu},\ \text{and so on.}$$

The DDP idea is to minimize quadratic approximations such as $L(x, u, N)$, instead of the actual control problem value functions, thereby obtaining

computer amenable functions (at the expense of invoking truncation error). A necessary condition that an input u^* be a minimizer of $L(x, u, N)$ is that

$$\nabla_u L(x, u, N) = 2C_N\,\delta u + B_N\,\delta x + D_N = 0. \tag{2.3}$$

One point in making the quadratic approximation is that the optimal input u^* can be easily found by solving (2.3). Assuming C_N is nonsingular, this gives

$$\delta u\,(x, N) = (u^* - u_N^C) = -(1/2)\,C_N^{-1}\,(D_N + B_N\,\delta x) = \alpha_N + \beta_N\,\delta x \tag{2.4}$$

where obviously we have set

$$\alpha_N = (-1/2)\,C_N^{-1}\,D_N, \quad \text{and} \quad \beta_N = (-1/2)\,C_N^{-1}\,B_N. \tag{2.5}$$

The optimal value function is defined by

$$f(x; N) = \min_u g(x, u, N). \tag{2.6}$$

We approximate the optimal value function by the quadratic:

$$V(x; N) = L\big(x, u(x, N), N\big) = L\big(x, \bar{u}_N + (\alpha_N + \beta_N\,\delta x), N\big). \tag{2.7}$$

One readily checks that $V(x; N)$ is a quadratic,

$$V(x; N) = \delta x^T P_N\,\delta x + Q_N\,\delta x, \tag{2.8}$$

where the coefficients are

$$\begin{aligned} P_N &= A_N - (1/4)\,(B_N)^T\,C_N^{-1}\,B_N, \\ Q_N^T &= -(1/2)\,(\mathrm{D}_N)^T\,C_N^{-1}\,B_N + E_N. \end{aligned} \tag{2.9}$$

under the circumstance that C_N is nonsingular.

The general DDP backward recursion proceeds for $t = N, N-1, \ldots, 1$ as follows. Assume inductively that the quadratic approximate optimal return function

$$V(x, t+1) = (\delta x)^T P_{t+1}\,\delta x + Q_{t+1}\,\delta x. \tag{2.10}$$

has already been constructed. Define the quadratic

$$L(x, u, t) = QP\big[g(x, u, t) + V\big(T(x, u, t); t+1\big)\big]. \tag{2.11}$$

Analogously to (2.2), display its coefficients as

$$L(x, u, t) = \delta x^T A_t\,\delta x + \delta u^T B_t\,\delta x + \delta u^T C_t\,\delta u + D_t^T E_t^T\,\delta x. \tag{2.12}$$

By calculus, one confirms that these coefficients may be written explicitly in terms of first and second order derivatives of the state transition function and the current stagewise loss function, as well as the coefficients P_{t+1} and Q_{t+1} by

$$A_t = (1/2)\left[g_{xx} + 2\left(\frac{\partial T}{\partial x}\right)^T P_{t+1}\left(\frac{\partial T}{\partial x}\right) + \sum_{i=1}^{n} (Q_{t+1})_i\,(T_i)_{xx}\right],$$

$$B_t^T = g_{xu} + 2\left(\frac{\partial T}{\partial x}\right)^T P_{t+1}\left(\frac{\partial T}{\partial x}\right) + \sum_{i=1}^{n} (Q_{t+1})_i (T_i)_{xu}\Bigg],$$

$$C_t = (1/2)\Bigg[g_{uu} + 2\left(\frac{\partial T}{\partial u}\right)^T P_{t+1}\left(\frac{\partial T}{\partial u}\right) + \sum_{i=1}^{n} (Q_{t+1})_i (T_i)_{uu}\Bigg], \tag{2.13}$$

$$D_t^T = \nabla_u g + Q_{t+1}\left(\frac{\partial T}{\partial x}\right),$$

$$E_t^T = \nabla_x g + Q_{t+1}\left(\frac{\partial T}{\partial x}\right),$$

where $\nabla_x g$, $\nabla_u g$ are the components of the gradient $g(x, u, t)$; g_{xx}, g_{xu}, g_{uu} are the components of the Hessian of $g(x, u, t)$; $\partial T/\partial x$, $\partial T/\partial u$ are the Jacobians of $T(x, u, t)$; $(T_i)_{xx}$, $(T_i)_{xu}$, $(T_i)_{uu}$, $1 \leqslant i \leqslant n$, are the blocks of the Hessian matrices of the coordinates of $T(x, u, t)$. Of course, all derivatives are taken about current states and inputs (x_t^C, u_t^C).

As in the argument of the case $t = N$, the first order necessary condition is that

$$\nabla_u L(x, u, t) = 0, \tag{2.14}$$

whence the minimizing strategy for the quadratic $L(x, u, t)$ is

$$u(x; t) = \alpha_t + \beta_t (x - x_t^C), \tag{2.15}$$

where

$$\begin{aligned} \alpha_t &= (-1/2)\, C_t^{-1} D_t^T, \\ \beta_t &= (-1/2)\, C_t^{-1} B_t, \end{aligned} \tag{2.16}$$

The approximating polynomial for the optimal return function

$$V(x; t) = L\big(x, u(x, t), t\big) = (\delta x)^T P_t (x - x_t^C) + Q_t^T \delta x, \tag{2.17}$$

has coefficients given by

$$\begin{aligned} P_t &= A_t - (1/4)\, B_t^T C_t^{-1} B_t, \\ Q_t &= -(1/2)\, D_t^T C_t^{-1} B_t + E_t^T. \end{aligned} \tag{2.18}$$

This completes the inductive step of the DDP backward recursion. The vectors and matrices α_t, β_t, $1 \leqslant t \leqslant N$, must be stored for use in the forward run.

The forward run, to determine the successor DDP policy, simply amounts to successively choosing inputs according to the rule $u(x^*, t)$ and then calculating the successor state, at each decision time. Thus $u^*(1) = u\big(x(1); 1\big)$ and $x^*(2) = T\big(x^*(1), u^*(1), 1\big)$. For $t = 2, \ldots, N$,

$$u^*(t) = u\big(x^*(t); t\big) + u^C(t), \tag{2.19a}$$

and

$$x^*(t+1) = T\left(x^*(t), u^*(t), t\right). \tag{2.19b}$$

The control $\mathbf{u}^* = \left(u^*(1), \ldots, u^*(N)\right)$ thereby obtained is the DDP successor control, and for the next DDP iteration, it will play the role of the current control sequence, $\mathbf{u}^C$.

3. A survey of computational results in differential dynamic programming

Before proceeding onward with a synopsis of theoretical results about and technical refinements of DDP, it is well to offer some computational evidence that the method is worth the effort of analyzing and uncerstanding.

The Unconstrained Case

Mine and Ohno (1970) had studied circumstances in which mathematical programming problems can be rephrased as optimal control problems. The key property is called "separability". Murray and Yakowitz (1981) undertook a substantial computational study of separable mathematical programming problems, such problems being susceptible to DDP solution. They tested this approach against results reported in the optimization literature for certain standard "benchmark" programs (sum of exponentials, Polak bowl, Oren power function, Rosenbrock functions I and II, etc.). The number of DDP recursions and function calls required was competitive or less than that reported by optimization experts using conventional mathematical programming algorithms. The improvement increased with the number of problem variables, which in our studies ranged as high as 200. The mechanism for this improvement comes from the fact that DDP only requires solving a great number of small dimensional problems (one for each stage), whereas mathematical programming schemes require simultaneous solution of a problem of many variables, at each iteration. By pure Newton iteration, for instance, the cost of an iteration grows as N^3, whereas by DDP, it is as N; both algorithms are quadratically convergent.

Yakowitz and Rutherford (1984) sought to apply the DDP method of the preceding section to a generic optimal control problem. They were unaware of any high-dimensional problems already in the literature. Therefore they devised the problem class with state transition function and stagewise loss function.

Transition function:

$$x' = T(x, u, t) = x + F\omega(u), \tag{3.1}$$

where for $u = (u^1, u^2, \ldots, u^m)$

$$\omega(u) = (\sin u^1, \sin u^2, \ldots, \sin u^m).$$

Loss function:

$$g(x, u, t) = \exp(\|x\|^2)\,[\sin^2(\|u\|^2/m)+1], \tag{3.2}$$

the norm being Euclidean.

In retrospect, this might not have been the finest problem imaginable, but it did have useful ingredients: the state and input dimensions n and m, as well as the stage horizon N, were readily modified. Its dynamics were nonlinear, and the loss was not quadratic. This was the one and only class we looked at. The components of F in (3.1) were chosen at random, and the initial control was chosen to be the sequence of zero vectors. The computation times on a CYBER 175 ranged from 0.724 sec. for a problem with $m = n = N = 5$ to 81.2 sec. for a problem with $m = n = 40$, and $N = 5$. To our knowledge, this latter problem is astronomical compared to the size of any other control problem of general structure reported solved in the literature, even to this day.

Recently, Sen and Yakowitz (1987) have developed a "quasi-Newton" version of DDP which requires only first derivative information. More will be said about the algorithm and its properties in the section to follow. Some computational studies were reported for the problem (3.1) and (3.2). Specifically, for a $n = m = N = 5$ case, the QDDP rule required 13 iterations to DDP's 7, for nine-digit accuracy. One should bear in mind that DDP requires second-derivative information and therefore a single iteration is typically much more expensive than a QDDP iteration, since QDDP is first-order. The calculations do make it clear that QDDP has a superlinear convergence rate, in conformance with theoretical results to be reviewed in Section 4.

The Contrained Case

We have already mentioned that the origins of DDP lie in attempts to implement variational-calculus solutions to continuous-time optimal control problems. McReynolds (1967) (also Dyer and McReynolds (1970)) derived the DDP algorithm as a finite-difference approximation to his "successive-sweep" method for continuous-time optimal control problems. He applied the algorithm to the brachistochrone and orbit-transfer problems. In the latter case, the state dimension is 3, the input is real, and the numbers of decision times N was in the order of 30. Gershwin and Jacobson (1970) applied their version of constrained DDP to McReynolds (1967) orbit transfer problem.

Ohno (1978a, b) proposed using a Lagrange-multiplier scheme for extending DDP to constrained optimization problems. In essence, by Ohno's plan, one simultaneously solves for the control and the Lagrange multiplier sequence which gives a Kuhn-Tucker point. The study (1978a) developed the theory in context of a separable mathematical programming problem

and applied it to a smallish (one dimensional, three stage) example. The second study applied the technique to an optimal-control problem with state constraints, previously analyzed by Jacobsen and Lele. Ohno's discretized version had 100 decision times and two state variables, the control being scalar.

Murray and Yakowitz (1979) presented a method of constrained DDP which was based on stagewise solution of a quadratic programming problem. Thus, in constrast to the Ohno-Dyer-McReynolds plan, it does not carry the Lagrange multipliers as a global variable. We applied the technique to a multireservoir problem previously studied using other techniques by Larson (1968) and Heidari et al. (1971). State constraints are imposed by reservoir capacities. The multireservoir problem has some distinction from the problems we have already mentioned: it does not arise from a discrete--time approximation to a continuous optimal control problem, and by making fictitious reservoir networks, the complexity in terms of state and control dimensions, is readily modified. The constrained DDP algorithm easily solved the four dimensional problems analyzed by the other authors, and additionally solved a 10 reservoir (state and control dimension problem which we thought might be inaccessible to the earlier techniques).

Jones et al. (1987) have applied the constrained differential dynamic programming method of Murray and Yakowitz (1979) to a substantial problem in groundwater management. This development in encouraging because it is a life-size practically motivated study. Furthermore, the deterministic dynamics assumption here is more reasonable and hydrologically traditional than the multireservoir model.

To summarize, my enthusiasm for the DDP idea was kindled by its success on what I consider to be substantial problems. It fares well against mathematical programming techniques on mathematical programming's home ground. My collaborators and I have solved control problems of both unconstrained and constrained variety which are larger (in terms of number of state and control variables) than problems I have encountered in the literature solved by others, regardless of method.

In the section to follow, I will point out properties of the DDP approach that give it advantageous features in comparison to alternatives.

4. The theory

In this section, I examine convergence properties of the basic DDP method presented in Section 2 and its extensions. The extensions are i) "damping" to assure convergence, ii) secant-type approximation of Hessians to avoid explicit computing of second derivatives, and iii) alterations to encompass constrained optimal control problems.

Quadratic Convergence Rate

The discrete-time DDP idea has its origins in approximating methods for continuous-time problems. These methods were second-order and were inspired by the Kantorovich theory of Newton methods for function spaces. At least one early author thought that DDP in the discrete setting inherited the property of being a Newton method, and another, while not making the claim, did claim flalty that the convergence rate was quadratic. In his thesis, Murray (1978) examined this issue carefully and proved that DDP is quadratically convergent, but nevertheless it does not coincide with Newton's method. Murray and Yakowitz (1984) contains a careful comparison of the two methods, and contains a quadratic convergence proof which refines that of Murray (1978).

At about the same time as Murray's study, Ohno (1978b) published a variation of DDP and gave a very inclusive analysis showing that the overall convergence rate of his method is the same as that of the mathematical programming technique applied at each stage. He explicitly gave a rule which obtains quadratic convergence. Ohno's method has not been followed up in the literature; it involves more derivative evaluations for each control recursion step than does the prototype DDP scheme, and conditions assuring convergence have not been postulated yet. Nevertheless, I regard his study as being foundational, and admire the comprehensiveness of his result on convergence rates.

Conditions for Convergence

Demonstrations of quadratic convergence are predicated on the assumption that convergence does occur. Like pure Newton steps, in order to assure convergence of a sequence of DDP controls to an optimal control, one must demand that the initial "guess" be sufficiently close to an optimal control. But one can "borrow" the idea of the damped Newton steps (e.g., Ortega and Rhineboldt (1970)) to assure global convergence under lenient circumstances. The following developments are from Yakowitz and Rutherford (1984). The step search method was given by Mayne (1966).

Modification of DDP for Global Convergence

At the terminal stage N, compute the eigenvalues of C_N, and let λ_{MIN} denote the minimum of these. If $\lambda_{\text{MIN}} < 0$, set

$$\tilde{C}_N = C_N - 2\lambda_{\text{MIN}} I. \tag{4.1}$$

I being the nth order identity, to make $\tilde{C}_N$ positive definite. Otherwise, set $\tilde{C}_N = C_N$, and in any case, set

$$\Theta_N = -D_N^T \tilde{C}_N^{-1} D_N, \tag{4.2}$$

and during the backward recursions $j = N-1, \ldots, 1$, also compute

$$\Theta_j = -D_j^T C_j^{-1} D_j + \Theta_{j+1}. \tag{4.3}$$

where $\tilde{C}_j$ is computed analogously to $\tilde{C}_N$. During the forward-run, initially set $\varepsilon = 1$. But generally, replace the forward-run recursion (2.19) by

$$\begin{aligned} u^*(t) &= u^\iota(t) + (1/2)\left[-\varepsilon \tilde{C}_t^{-1} D_t - \tilde{C}_t^{-1} B_t \left(x^*(t) - x^\iota(t)\right)\right], \\ x^*(t+1) &= T\left(x^*(t), u^*(t), t\right). \end{aligned} \tag{4.4}$$

At the end of the forward run, let $\mathbf{u}(\varepsilon) = (u^*(1), \ldots, u^*(N))$, and compute

$$J\left(\mathbf{u}(\varepsilon)\right) = \sum_{t=1}^{N} L\left(x_t^*, u_t(\varepsilon), t\right). \tag{4.5}$$

If

$$J\left(\mathbf{u}(\varepsilon)\right) < J(\mathbf{u}^\iota) + \Theta(1) * \varepsilon, \tag{4.6}$$

then accept $\mathbf{u}(\varepsilon)$ as the successor control. Otherwise, set

$$\varepsilon = \varepsilon/2, \tag{4.7}$$

and repeat the recursion (4.1–4.7).

In his original paper credited with the conception of discrete DDP, Mayne (1966) showed that the algorithm above assures improvement at each iteration, unless the nominal policy is already stationary. Yakowitz and Rutherford (1984) showed that if g and T have continuous second partials, then if $\mathbf{u}^\iota$ is not stationary eventually a successor is accepted, and that any accumulation point $\mathbf{u}^*$ of DDP iterates is a stationary control sequences (i.e., the gradient of $J(\mathbf{u})$ is 0) at that control $\mathbf{u}^*$. (This is a stronger statement than Mayne's, since improvement at each iteration does not in itself assure convergence to a stationary point). (The idea of forcing C_t to be positive definite was employed by Murray (1978)).

First Order Methods

A broad view of DDP, and one to which I subscribe, is that DDP is any dynamic programming method that is suited to take advantage of differentiability of the loss and transition functions. Under this expanded conception, first-order gradient algorithms suggested by Dyer and McReynolds (1970), Bellman and Dreyfus (1962, The Appendix) and others merit consideration. The background of these methods is that they were proposed in the early days of mathematical programming when the steepest descent method was in vogue. In subsequent years, pure gradient methods have been displaced by other first-order schemes which sidestep some of the gradient method's "zig-zag" problems, and which, moreover, have the super-

linear convergence property. The book by Gill et al. (1981) gives perspective to these assertions and also views quasi-Newton methods as being an attractive replacement, perhaps sharing this position with conjugate gradient algorithms.

Sen and Yakowitz (1987) have proposed a quasi-Newton differential dynamic programming algorithm. It replaces A_t, B_t, and C_t, in the prototypical DDP rule (2.13) by secant-type updates which require that at each QDDP iteration, the current estimates $\tilde{A}_t^C$, $\tilde{B}_t^C$ and $\tilde{C}_t^C$ of the unknown second-order derivative matrices be consistent with the most recent gradient values. That is

$$\begin{bmatrix} A_t^C & (B_t^C)^T \\ B_t^C & C_t^C \end{bmatrix} \begin{bmatrix} \Delta x_t^C \\ \Delta u_t^C \end{bmatrix} = \begin{bmatrix} \Delta E_t^C \\ \Delta D_t^C \end{bmatrix}. \tag{4.8}$$

In this expression $\Delta x^C = x_t^C - x_t^-$, and $\Delta u^C = u_t^C - u_t^-$. Here, as before, x_t^C is the state at time t of the current trajectory, and we define x_t^- to be the corresponding state at the preceding QDDP iteration. The term Δu_t^C is defined similarly, and $\Delta E_t^C = E_t^C - E_t^-$, where E_t^C is as in (2.3), and E_t^- is the E_t term of the preceding QDDP iteration; of course, $\Delta D_t^C = = D_t^C - D_t^-$.

Rewrite (4.8) as

$$M^C \delta = \gamma.$$

Then we compute M^C from the matrix M^- at the preceding QDDP recursion by Broyden updates (e.g., Fletcher (1980)),

$$M^C = M^- + (\gamma - M^- \delta)(\gamma - M^- \delta)^T / [(\gamma - M^- \delta)^T \delta]. \tag{4.9}$$

In Sen and Yakowitz (1987), we demonstrated that this procedure does inherit the quasi-Newton property of super-linear convergence. We did enough computational testing to see that it is quite effective on a benchmark control problem; these results were mentioned in the preceding section.

The Constrained Case

McReynolds (1967) presented a DDP algorithm in a form that allowed constrained problems. Suppose at each stage, we have a constraint

$$h(x, u, t) = 0,$$

with k being the dimension of the range of $h(\ )$. The idea is to view the Lagrange parameter (vector) λ_t in the Hamiltonian

$$g(x, u, t) + V_t(T(x, u, t)) + \lambda_t h(x, u, t),$$

as another control variable. This effectively reduces the problem to an unconstrained problem with $m+k$ variables at each stage. I have not checked the details, but I think it highly likely that the method is quadratically convergent, as McReynolds claimed (but did not prove).

Ohno (1978b) presented his DDP plan in a setting essentially the same as McReynolds' (1968), so that it is suited to constrained problems. He did provide convergence proofs, but for an algorithm that is somewhat different from the fundamental DDP construct of Section 2.

Murray and Yakowitz (1979) offered a constrained DDP algorithm which does not treat the Lagrange multiplier as a separate control variable, but rather replaces the Newton step at each stage by the quadratic programming problem,

$$\underset{u}{\text{Minimize}}\ QP\left[g(x,u,t)+V\left(T(x,u,t);t+1\right)\right],$$

subject to

$$LP\left[h(x,u,t)\right]=0. \tag{4.10}$$

Here, as in Section 2, "QP" means the quadratic and linear part of the Taylor's series expansion about $\mathbf{x}^C$ and $\mathbf{u}^C$. "LP" indicates the constant and linear part of the expansion. By conventional methods (Fletcher, 1980), this reduces to a linear equation

$$\begin{bmatrix} C_t & U_t^T \\ U_t & \mathbf{0} \end{bmatrix}\begin{bmatrix} \delta u \\ \lambda \end{bmatrix}=\begin{bmatrix} -D_t-B_t\,\delta x \\ -W_t-X_t\,\delta x \end{bmatrix}. \tag{4.11}$$

Here B_t, C_t, and D_t are as in (2.13), $\mathbf{0}_t$ is a matrix of zeros, and U_t, W_t, and X_t are the coefficients of the linear part of h:

$$LP\left[h(x,u,t)\right]=W_t+X_t\,\delta x+U_t\,\delta u. \tag{4.12}$$

By multiplying by the inverse of the system matrix in (4.11), one can get the QDDP control law counterpart to (2.4), namely

$$\delta u=\alpha_t+\beta_t\,\delta x,$$

where

$$\alpha_t=-\pi^{-1}\begin{bmatrix} D_t \\ -W_t \end{bmatrix},$$

$$\beta_t=-\pi^{-1}\begin{bmatrix} B_t \\ X_t \end{bmatrix},$$

where

$$\pi=\begin{bmatrix} C_t & U_t^T \\ U_t & \mathbf{0} \end{bmatrix}$$

Yakowitz (1986) has derived a theory for this approach and proved convergence under the assumption that $J(\mathbf{u})$ is positive definite at all policies $\mathbf{u}$, the constraint functions actually are linear, and a stepsize selection rule such as (4.4–4.7) is followed.

All of these methods have obvious counterparts for the inequality constraint case.

The Alternatives to DDP

One groundrule that I impose in seeking DDP alternatives is that the method must not depend on discretization of state or policy space, as does conventional discrete dynamic programming (e.g., Bellman and Dreyfus (1962)), for such methods are subject to "the curse of dimensionality": The storage and effort grows as a power of state and control dimension. Also, I discard those methods which do not make a stage-wise decomposition of the control problem. This includes techniques which minimize $J(\mathbf{u})$ in (1.2) directly as a function of $\mathbf{u}$, without regard to the control process structure. The only techniques in the permissible class that I know of, aside from DDP, which are computer-realizable, are

i. The Successive Approximations (SA) method of Larson and Korsak (1970).
ii. Methods based on the discrete maximum principle.

I am unenthusiastic about SA. It amounts to optimizing on one variable at a time, and I have shown (Yakowitz (1983)) that in so doing, it is essentially a block-Seidel method and thus has only linear convergence. Such methods, known as "coordinate search" techniques, are not popular in the mathematical programming literature.

The discrete maximum principle has more theoretical attraction. Analogously to the continuous-time case, the idea is to convert the problem into a two-point boundary-value difference equation problem. Such problems are within the domains of shooting and quasi-linearization methods, which are quadratically convergent. I have not undertaken computer experimentation on such problems, but my experience with continuous-time problems has convinced me that the maximum principle, in conjunction with quasi-linearization, is a very effective approach. There is some difficulty in establishing conditions under which the discrete maximum principle applies (Halkin (1966)).

5. Research Directions

Some Activities and Refinements

My impression is that for deterministic, finite-horizon optimal control problems having smooth stagewise loss and transition functions, the status of DDP theory and methods is reasonably complete and powerful. At this point, the main need I see is for "customers" from the world of applications. For a while, optimal control was oversold in the sense that the computational methodology limited applications to small-scale problems, whereas the conceptual effort needed to understand the literature was relatively sizeable. During this period, optimal control theory lost its luster for a great many people, especially those outside academia. At this writing,

and in view of advances such as I have been describing in this paper, and in view of hardware and software progress in computer technology, for certain classes of problems, I believe the situation is much more hopeful, and I emphatically urge people close to "real-world" control problems to give modern developments a try. Briefly, the most needed ingredient in optimal control theory is a little encouragement by way of impact on engineering problems, along the lines of Jones et. al. (1987).

On the theoretical side, there are a few odds and ends that should be straightened up. For example, Powell (1986) has recently published some analytic and computational investigations which suggest that certain conjugate gradient methods have advantage over the quasi-Newton idea for mathematical programming. It would seem worthwhile to derive and examine a conjugate gradient version of DDP. Lasdon et al. (1967) and others have devised a conjugate gradient rule for continuous-time optimal control problem. My student T. Jayawardena is investigating convergence properties for a discrete-time version.

The constrained DDP method has some loose ends. I am sure the stagewise quadratic programming algorithm I presented in the preceding section is not quadratically convergent. But in proving global convergence, I developed (Yakowitz (1986), Section IV) some expressions which I think could lead to a modified version of the rule that would be quadratically convergent. Such an algorithm would be the first to be shown globally and quadratically convergent to a Kuhn-Tucker point.

The Stochastic Bottleneck

A stochastic control problem arises if the deterministic transition law (1.1) is replaced by a stochastic rule

$$x(n+1) = T(x(n), u(n), t) + W(n),$$

where $\{W(n)\}$ is a random noise sequence. In this case, one is to minimize the expected value of J in (1.2). One typically applies the strategy function $S(x(t), t)$ at each stage, instead of seeking a control sequence $\mathbf{u}^*$, to allow the flexibility of letting the input depend on state as well as time. In the stochastic case, such feedback leads to improved performance since the states cannot be determined in advance, under a given decision plan.

There is no question that stochastic problems are of paramount importance to practical activities; their importance exceeds that of deterministic problems. About every six months I am sent a paper from the practitoner's community for review which gives an algorithm for stochastic optimal control problems and hints that this algorithm yields an optimal solution. I am convinced that the DDP idea does not readily generalize to stochastic control because the mechanism that makes DDP work (namely, that the dynamics and return

functions are accurately described by quadratics in a close neighborhood of the current control) has no bearing on the stochastic case where randomness makes states flit around erratically. Quadratics are not accurate if the noise is not negligible. Except in a few cases (notably the Gauss--Markov model) computational methodology is still subject to the "curse of dimensionality". Further discussion on this matter, and information about certain stochastic problems that have solutions is offered in Section 5 of my survey paper Yakowitz (1982). I see no hope for a general approach to stochastic control, and think that the field has not advanced dramatically over discrete dynamic programming techniques given by Bellman and Dreyfus (1962).

Artificial Intelligence

My own investigations for stochastic control have, at the current time, departed from attempts at strickt optimization and turned toward heuristic methods. I think that notions of artificial intelligence and heuristic [graph] search are promising. They were developed in the context of deterministic games and decision problems (e.g., Nilsson (1980)). I have been paying particular attention to "machine learning" ideas, about which I cannot find much adequate scientific literature. My initial efforts in this direction are outlined in Yakowitz and Lugosi (1987), and were motivated by attempts to solve the 8 and 15 puzzles with randomly chosen initial configurations.

Acknowledgement. My former student and collaborator D. M. Murray has been inspirational over the years. Also, much of my studies described herein was supported by NSF grants, especially CEE 81-10778.

References

[1] Bellman R., Dreyfus S. Applied Dynamic Programming. Princeton, NJ. Princeton University Press, 1962.

[2] Dyer P., McReynolds S. The Computational Theory of Optimal Control. New York, Academic, 1970.

[3] Fletcher R. Practical Methods of Optimization, Vol. 1, Unconstrained Optimization. NY., John Wiley and Sons, 1980.

[4] Gershwin S. H., Jacobson D. H. A Discrete-Time Differential Dynamic Programming Algorithm with Application to Optimal Orbit Transfer. *AIAA Journal*, **8** (1970) 9, 1616–1626.

[5] Gill P. E., Murray W., Wright M. H. Practical Optimization, London, Academic Press, 1981.

[6] Halkin H. A Maximum Principle of Pontryagin Type for Systems Described by Nonlinear Difference Equations. *SIAM Journal on Control*, **4** (1966), 90–111.

[7] HEIDARI M., CHOW V. T., KOKOTOVIC P. V., MEREDITH D. Discrete Differential Dynamic Programming Approach to Water Resources Systems Organization. *Water Resources Research*, **7** (1971), 273–282.
[8] JACOBSON D., MAYNE D. Differential Dynamic Programming, New York, Elsevier, 1970.
[9] JONES L. WILLIS R. YEH W. W-G. Optimal Control of Groundwater Hydraulics Using Differential Dynamic Programming, *Water Resour. Res.*, (1987), to appear.
[10] LARSON R. State Increment Dynamic Programming. New York, Elsevier, 1968.
[11] LARSON R., KORSAK A. A Dynamic Programming Successive Approximations Technique with Convergence Proofs. *Automatica*, **6**, (1970), 245–252.
[12] LASDON L. S., MITTER S. K., WAREN A. D. The Conjugate Gradient Method for Optimal Control Problems. *IEEE Trans. on Automatic Control*, **AC-12** (1967) 2, 132–138.
[13] MAYNE D. A Second Order Gradient Method for Determining Optimal Trajectories for Nonlinear Discrete-Time Systems. *International Journal on Control*, **3** (1966), 85–95.
[14] MCREYNOLDS S. R. The Successive Sweep Method and Dynamic Programming, *Jour. Mathematical Analysis and Applications*, **19** (1967), 565–598.
[15] MINE H., OHNO K. Decomposition of Mathematical Programming Problems by Dynamic Programming and its Application to Block-Diagonal Geometric Programs, *J. Math. Analysis and its Applications*, 32 (1970), 370–385.
[16] MURRAY D. M. Differential Dynamic Programming for the Efficient Solution of Optimal Control Problems. Ph. D. Dissertation, Dept. of Mathematics, Univ. of Arizona, Tucson, Arizona, University Microfilm Inc., Ann Arbor, Mich., 1978.
[17] MURRAY D. M., YAKOWITZ S. Constrained Differential Dynamic Programming with Application to Multi-Reservoir Control. *Water Resources Res.*, **15** (1979) 2, 1017–1027.
[18] MURRAY D. M., YAKOWITZ S. The Application of Optimal Control Methodology to Nonlinear Programming Problems, *Mathematical Programming*, **21** (1981), 331–347.
[19] MURRAY D. M., YAKOWITZ S. J. Differential Dynamic Programming and Newton's Method for Discrete Optimal Control Problems. *Journal of Optimization Theory and Applications*, **43** (1984), 395–414.
[20] NILSSON N. J. Principles of Artificial Intelligence. Palo Alto, Tioga Publishing Co., 1980.
[21] OHNO K. Differential Dynamic Programming and Separable Programs. *Journ. of Optimization Theory and Applications*, **24** (1978a) 4, 617–637.
[22] OHNO K. A New Approach to Differential Dynamic Programming. *IEEE Transactions on Automatic Control*, **23** (1978b), 37–47.
[23] ORTEGA J. M., RHEINBOLDT W. C. Iterative Solution of Nonlinear Equations in Several Variables. New York, Academic Press, 1970.
[24] POWELL M. J. Convergence Properties of Algorithms for Nonlinear Optimization. *SIAM Review*, **28** (1986) 487–500.
[25] SEN S., YAKOWITZ S. A Quasi-Newton Differential Dynamic Programming Algorithm for Discrete-Time Optimal Control. *Automatica*, (1987), to appear.
[26] YAKOWITZ S. Dynamic Programming Applications in Water Resources. *Water Resources Res.*, **18** (1982), 673–698.
[27] YAKOWITZ S. J. Convergence Rate Analysis of the State Increment Dynamic Programming Method. *Automatica*, **19** (1983), 53–60.
[28] YAKOWITZ S. J. (1986) A Stagewise Kuhn-Tucker Condition and Differential Dynamic Programming. *IEEE Trans. on Auto. Control*, **AC-31** (1986), 25–30.
[29] YAKOWITZ S. J., RUTHERFORD B. Computational Aspects of Discrete-Time Optimal Control. *Applied Mathematics and Computation*, **15** (1984), 29–45.
[30] YAKOWITZ S., LUGOSI E. Random Search in the Presence of Noise, with Application to Machine Learning. Submitted for publication, 1987.

Received, January 1987.

Postępy teoretyczne i obliczeniowe w różniczkowalnym programowaniu dynamicznym

Poniższa praca przeglądowa zaczyna się od przedstawienia idei różniczkowego programowania dynamicznego (DDP). Następnie skoncentrowano się na opisaniu prac Ohno, autora i jego kolegów związanych ze zbieżnością klasycznych algorytmów DDP i ich wariantów. Przedstawiono obliczenia potwierdzające duże możliwości tego podejścia. Istotą DDP jest poetapowe zastosowanie algorytmów programowania nieliniowego. Opisane ostatnie osiągnięcia dotyczące wprowadzenia do algorytmów DDP metody Newtona i naszkicowano prowadzone obecnie badania związane z zadaniami DDP z ograniczeniami.

Теоретический и вычислительный прогресс в дифференцируемом динамическом программировании

Данная обзорная работа начинается от представления идеи дифференцируемого динамического программирования (ДДП). Затем обращено внимание на описание работ автора данной статьи и его коллег, а также Оно, связанных со сходимостью классических алгоритмов ДДП и их вариантов. Представлены вычисления, подтверждающие большие возможности этого подхода. Суть ДДП состоит в поэтапном применении алгоритмов нелинейного программирования. Описаны последние достижения, касающиеся введения в алгоритмы ДДП метода Ньютона и оговорены проводимые в настоящие время исследования, связанные с задачами ДДП с ограничениями.