National Symposium on Hydrology
Dec.16-18,1987, N.I.H., Roorkee

CONSTRAINED DIFFERENTIAL DYNAMIC PROGRAMMING APPLIED TO MULTIRESERVOIR CONTROL PROBLEMS

J.N. NANDA*

ABSTRACT

Eversince Bellman's (1957) book on Dynamic Programming (DP), new avenues have opened up for the reservoir operation or control problems. The computational effort required to solve a multireservoir control problem by the D.P. is immense when the dimensions of the problem increases, so much so that it is beyond the capacity of the present day computers. This aspect has been qualified as "the curse of dimensionality" . Many computational efforts have undergone in the literatures in the past two decades. Most of them either are not out of the problem of "curse of dimentionality" or do not assure convergence to a stationary global optimum. The Constrained Differential Dynamic Programming (Constrained DDP) due to Murray and Yakowitz (1979) and Yokavitz(1986) successfully overcomes this "curse" by successive approximations starting from an initial values of policy and state variables. The convergence to a stationary global optimum has been proved when both the loss function and the constraints are convex.

1.INTRODUCTION

The problems connected with reservoir operation have been a major application ground for Dynamic Programming (DP). But the greatest single hinderance to dynamic programming solution to a large scale optimal control problem (O.C.P.) is the " curse of dimensionality" set in by increase in the dimensions in the elements of the problem, e.g., when more than one reservoirs are considered, and also by multiplicity of state and control vector elements. All these have resulted in an exponential growth in computer space and time required for computation. To illustrate the " curse", let us suppose that each coordinate of the state variable must be discretised into 10 levels to achieve satisfactory approximation accuracy. This implies that if there are 12 stages (i.e. if $n = 12$), then the number of discretised state nodes will be 10^{12}, an impossibly large number to deal with on present or foreseable computers. Many workers have attacked this problem by way of simplification without assuring convergence to a stationary global optimum. Some other methods suggested are also not out of the problem of "curse of dimensionality" . The constrained Differential Dynamic Programming (constrained DDP) not only overcomes the " curse of dimensionality" but also

* Director, Flood Hydrology, Central Water Commission, Sewa Bhavan, R.K.Puram, New Delhi.

assures convergence to a stationary global optimum. In spite of the seamingly superiority in computational aspects of DDP over others, it appears inexplicable why this method has not been widely known to hydrologist and others involved in reservoir control problems. The reason may be two folds. One, because of lack of any clearcut computational device having been developed the research on developing a conceptual framework for constructing an OCP to suit a reservoir control problem has not made any big headway. Two, construction of an effective OCP to tackle the real time aspect of the reservoir control problem is yet to come off. The set up of this paper is as given below.

In the section 2 the Optimal Control Problem (OCP) has been defined and the D.P. solution has been introduced along with further developments in the computational aspects of D.P. problem. The comparative merits and demerits of each of the developments have been discussed. The section 4 introduces in brief the concepts and methodology behind constrained Differential Dynamic Programming and indicates the specific advantages of this procedure. In section 4, a problem of operation of four reservoir configuration has been solved using constrained Differential Dynamic Programming. The section 5 is for the conclusion.

2.Optimal Control Problem and Development in Computational Efforts

A deterministic multistage devision process, or (discrete time) 'optimal control problem (O.C.P.)' is characterised by an 'initial state' x, and functions T_t and L_t as follows:

The functions T_t determine relations between 'controls'(u_t) and states (x_t) according to

$$x_{t+1} = T_t(x_t,u_t) \quad 1 \leq t \leq N \tag{2.1}$$

where x_t's are real n-tupples and u_t's are m-tupples. The set of decision stages are N positive integers or the set of all positive integers for an infinite horizon process. The function $T_t(x_t,u_t)$ or T are referred to as 'dynamics' or 'continuity' and $L_t(x_t,u_t)$ or L_t as 'single stage loss function.

With any sequence $u = (u_t)$ of controls, which is known as a policy associates a real valued objective function defined by

$$J(u) = \sum_{t=1}^{N} L_t(x_t,u_t) \tag{2.2}$$

The goal in OCP is to construct a policy u* which minimises the objective function J(u). Typically, the 'Feasible' controls are those which satisfy a vector valued state stage dependent constraint of the form

$$f_t(x_y,u_t) \leq 0, \ 1 \leq t \leq N \tag{2.3}$$

This is derived from equation (2.1)

The D.P.procedure for the finite horizon OCP's begins by recursively solving what Bellman (1957) calls " functional equation", namely, successively determining the " optimal return functions" $V_N(x)$, $V_{N-1}(x)\ldots\ldots V_1(x)$ by the recursive equation

$$V_t(x) = \min_u (L_t\,(x,u)+V_{t+1}(f_t(x,u))) \qquad (2.4)$$

$$t=N,N-1.....,1$$

where $V_{N+1}(x)=0$.

The minimisation is with respect to controls satisfying (2.3). It is proven in (2.4) that the policy u* determined by

$$x^*_1=x_1 \text{ and } u^*_t=u_t(x^*_t),\ x^*_{t+1}=T_t(x^*_t,u^*_t) \text{ for } t=1,\ldots.,N \text{ is a}$$

global minimiser of J(u), i.e. u* is a solution of the O.C.P.

The three most highly regarded methods advanced in control theory literature to lessen the computational burden are

(1) State Incremental Dynamic Programming (SIDP) due to Larson (1968)

(2) Discrete Differential Dynamic Programming (DDDP) due to Heidari et.al. (1971).

(3) Differential Dynamic Programming (DDP)-Constrained and Unconstrained cases.

In the State Incremental Dynamic Programming (SIDP) method, it is emphasised that there must exist a function h_t such that for every pair of x_t and x_{t+1},

$$h_t(x_t,x_{t+1})=u_t \qquad (2.5)$$

where u_t is the control such that $x_t+1 = T(x_t,u_t)$.

For purpose of description of this method let x(i) denote the ith ordinate of the state vector x. With this notation, one can describe SIDP method as being an application of discrete dynamic programming to the OCP, with the added constraint that for some specified coordinate i,

$$x'_t(j)= \bar{x}_t(j) \qquad (2.6)$$

For all $i \neq j$ and all t=2,....,N+1. That is, in a given SIDP approximation, in passing from u to u', the only state coordinate that is allowed to change is the ith coordinate. The important aspect of equation (2.6) is to reduce the dimensionality of the state space of the OCP problem from n to 1. This is therefore an essentially a D.P. solution of OCP with a univariate state space.

In the Discrete Differential Dynamic Programming (DDDP) the state space is the unit n cube (i.e. each coordinates has to lie between 0 and 1 and the dimension of state vector is n). Suppose further that the state coordinates are discretised by a uniform mesh of size 0.01 for each t for some $\varepsilon > 0$

$$|\tilde{x}_t - x'_t| < \varepsilon \tag{2.8}$$

Assume $\varepsilon = 0.015$ and supposing the dynamics as a function of u, $T_t(x,u)$ is one to one, then for each devision at step t and state x, under DDDP there will be 3^n control values that must be searched over to minimise $V_t(x(i))$ in (2.5) whereas the discrete dynamic programming will require $(100)^n$ control nodal points to be searched. Although the computational effort has reduced while comparing with the discrete dynamic programming, the fact remains that DDDP effect grows exponentially with state dimension n. Hence it still suffers from the " curse of dimensionality". Besides, Turgeon (1982) has proved that such a dynamic programming may yield non-optimal solution.

The Differential Dynamic Programming (DDP) for unconstrained case was first given by Jacobson and Magne (1970). This is a successive approximation technique. Given some nonoptimal initial policy u, which is termed as a nominal policy, this procedure determines a successor policy u' which results in a lower loss than J(u). The nominal policy determines a nominal trajectory x through the recursive formula (2.1).

To begin the backward recurssion of the DDP, define the quadratic and linear Taylor series expansion of Q(x,u) as

$$\begin{aligned}\hat{Q}_t(x,u) &= \text{QP } L(x,u,t) + V_{t+1}(T(x,u,t)) \\ &= 1/2\ \delta x_t^T A_t\ \delta x_t + \delta x^T B_t\ \delta u + 1/2\ \delta u^T C_t\ \delta u + \\ &\qquad D_t^T\ \delta u + E^T\ \delta x\end{aligned} \tag{2.9}$$

where superscript ^{T} denotes ''Transpose", A_t and C_t are symmetric positive definite matrices of order n and m respectively, B_t is m x n matrix, D_t and E_t are m and n-tupples, $ox_t = x_t - x_t$ and $ou_t = u_t - u_t$. The constant terms are ignored.

For t=N,

$$(\nabla_u \hat{Q}_N(x,u))^T = 2C_N \delta u_N + B_N \delta x_N + D_N = 0 \tag{2.10}$$

from this,

$$\begin{aligned} ou(x,N) &= -\tfrac{1}{2} C_N^{-1}(D_N + B_N \partial x_N) \\ &= \alpha_N + \beta_N \delta x_N \end{aligned} \tag{2.11}$$

where

$$\alpha_N = -\frac{1}{2}C_N^{-1}D_N \text{ and } \beta_N = -\frac{1}{2}C_N^{-1}B_N$$

$$\hat{Q}_N(x,u) = \delta x^T P_N \delta x + R_N^T \delta x \tag{2.12}$$

The vector matrix pair (α_N, β_N) and the c oefficients P_N and R_N are stored in memory for use in subsequent stages.

$$\hat{Q}_t(x,u) = \text{QP } L(x,u,t) + V_{t+1}(T(x,u))$$
$$= \delta x^T A_t \delta x + \delta u^T B_t \delta x + \delta u^T C \;\; \delta u + D_t^T \;\delta u + E_t \delta x \tag{2.13}$$

Analogously to (2.13), the minimiser $u_t(x)$ is,

$$\delta u_t(x) = u_t(x) - u_t = \frac{1}{2}C_t^{-1}D_t - \frac{1}{2}C_t^{-1}B_t\delta x$$
$$= \alpha_t + \beta_t o x \tag{2.14}$$

Also, the quadratic approximation of return function is given by

$$\hat{Q}_t(x,u) = Q_t(x_t,(u_t+\alpha_t+\beta_t \;\delta x))$$
$$= g_t(\delta x) = \delta x^T P_t \;\delta x + R_t^T \;\delta x \tag{2.15}$$

The coefficient of P_t, R_t, α_t and β_t are stores for (t-1)th stage.

For the construction of the successor policy let ε denote a positive number then $u_t(\varepsilon)$ is defined as

$$u_t(\varepsilon) = u_t + \varepsilon\alpha_t + \beta_t \delta x_t \tag{2.16}$$
$$x_t+1 = T(x_t, u_t(\varepsilon), t)$$

Initially set $\varepsilon = 1$ and define

$$\theta_1 = \frac{1}{2}\sum_{t=1}^{N} - D_t^T C_t^{-1} D_t \tag{2.17}$$

If $$J(u(\varepsilon)) - J(\bar{u}) < \frac{1}{2}\theta_1 \tag{2.18}$$

then $u(\varepsilon)$ is accepted as successor policy and replaces u in the next DDP iteration. Otherwise, put $\varepsilon = 1/2\ \varepsilon$ and proceed as before.

3. CONSTRAINED DIFFERENTIAL DYNAMIC PROGRAMMING

The procedure for numerical computation is based on the work of Murray and Yakowitz (1979) and Yakowitz (1986). In this case the quadratic and linear approximation of the objective function at stage I and the constraints are as given below:

$$\hat{Q}_t(x,u) = QP[L(x,u,t)+V_t+1(T(x,u,t)]$$
$$= \frac{1}{2}\,\delta x^T A_t ox + \delta x^T B_t\,\delta u + \frac{1}{2}\,\delta u^T C_t \delta u$$
$$+ D_t^T\,\delta u + E_t^T\,\delta x \qquad (3.1)$$

$$\hat{f}(x,u,t) = LP(f(x,u,t)) \qquad (3.2)$$

The notation and dimensions are same as indicated earlier. With this the problem at stage -1 is

$$\left.\begin{array}{l} \text{Minimise } \hat{Q}_t(x_t,u) \\ \text{Subject to } \hat{f}(x_t,u,t) \end{array}\right) \qquad (3.3)$$

Let S be the index set of active constraints. Then, represent the linear system of equation

$$\hat{f}_i(x,u,t) = 0, \qquad i \in S \qquad (3.4)$$

In the matrix formit is written as

$$U_t\,\delta u - W_t - X_t\,\delta x = 0 \qquad (3.5)$$

The rank of U_t and X_t will be equal to the number of indices in S.

Then, if δu and λ are solution vector and lagrange multiplier, then solution is given by Kuha - Tucker necessary conditions as

$$\begin{bmatrix} C_t & U_t^T \\ U_t & 0 \end{bmatrix} \begin{bmatrix} du \\ \lambda \end{bmatrix} = \begin{bmatrix} -D_t \\ W_t \end{bmatrix} \qquad (3.6)$$

If $\lambda_i \leq 0$, then $i \not\in S$, U_t, W_t, X_t are stored in memory.
After the first DDP iteration, at each stage t, constraints may be dropped in active constraint index set S.

Strategy and Return function for stage t is computed.

Consider variability in x,

$$\begin{bmatrix} C_t & U_t^T \\ U_t & 0 \end{bmatrix} \begin{bmatrix} \delta u \\ \lambda \end{bmatrix} = \begin{bmatrix} -D_t & -B_t \delta x \\ W_t & +X_t \delta x \end{bmatrix} \tag{3.7}$$

from this we get

$$\delta u_t(x) = \alpha_t + \beta_t \, \delta x_t \tag{3.8}$$

α_t and β_t must be stored in memory for the forward run. The quadratic approximation of the return function is given by

$$V_t(x) = Q_t(x, \alpha_t + \beta_t \, \delta x) \tag{3.9}$$

Compute

$$\Theta_t = \Theta_{t+1} - D_t^T \, C_t^* \, D_t \tag{3.10}$$

Here C_t^* is the first m rows and columns of the inverse of the coefficient matrix

$$\begin{bmatrix} C_t & U_t^T \\ U_t & 0 \end{bmatrix} \tag{3.11}$$

All these will have to be computed in backward way from stage t = N to t = 1. After doing this the forward recursive computation is given by the following

Step 1 Compute recursively

$$u_t(\varepsilon) = \bar{u}_t + \varepsilon \alpha_t + \beta_t \, \delta x_t \tag{3.12}$$

and

$$\delta x_{t+1} = T(x_t, \, u_t(\varepsilon), t) - x_{t+1} \tag{3.13}$$

Step 2 Test $u(\varepsilon)$ for sufficient improvement as,

$$J(u(\varepsilon)) - J(u) \leq \frac{1}{2} \theta \varepsilon \tag{3.14}$$

If (3.14) obtains go to step 3 otherwise set

$$\varepsilon = \frac{1}{2} \varepsilon$$

and go to step 1

Step 3 Check to see if for all stages t

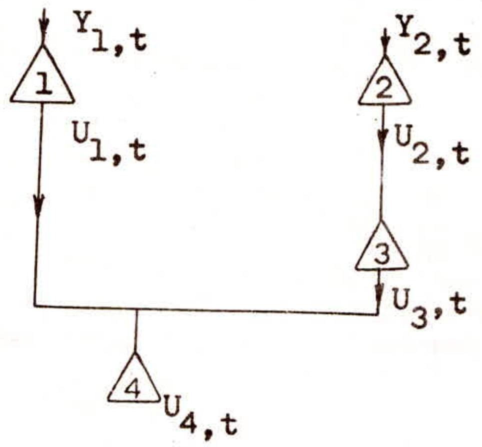
$$f(x_t(\varepsilon), \, u_t(\varepsilon), t) \leq 0 \tag{3.15}$$

If (3.14) is true $u(\varepsilon)$ is feasible and may be accepted as the new policy otherwise perform a line search by putting $\varepsilon=0.9\,\varepsilon$ and go to step 1.

The convergence criterion is satisfied by the nonsigularity of system matrix given in eqn. (3.6) and (3.7). The validity of KT condition in the forward run calculation has been proved by Yakowitz (1986). Since J(u) has a positive definite Hession matrix for all u and since the constraint function f is linear, i.e. $f(x,u,t) = LP(f(x,u,t))$ there is existence of on accumulation point of the sequence $(u^{(k)})$ of policies after repeated iteration. The general theory on convergence of D.D.P has been proved by Ohno (1979).

4. SOLUTION OF A FOUR RESERVOIR CONTROL PROBLEM

A four reservoir control problem used by Heidari et al. (1971) has been utilised. The problem specifies a 4-reservoir configuration as shown in fig.



The inflows Y_1 and Y_2 are diverministic and are constants ($Y_1 = 2$, $Y_2 = 3$). The upper and lower bounds of X_{it} and u_{it} have been specified. A linear loss function has been specified as

$$\left.\begin{aligned} J(u) &= \sum_{t=1}^{12} L(x_t, u_t, t) \\ \text{where } L(x_t, u_t, t) &= \sum_{j=1}^{4} (c_{jt}, u_{jt}) \end{aligned}\right) \quad ,(4.1)$$

Assume the nominal policy as given below in Table 4.1

TABLE 4.1

t	1	2	3	4	5	6	7	8	9	10	11	12
$u_{1,t}$	2	2	2	2	2	2	2	2	2	2	2	2
$u_{2,t}$	3	3	3	3	3	3	3	3	3	3	3	3
$u_{3,t}$	3	3	3	3	3	3	3	3	3	3	3	3
$u_{4,t}$	5	5	5	5	5	5	5	5	5	5	5	5

Cons istent with equation (4.2), the nominal values of x_t as given in Table 4.2

TABLE 4.2

t	1	2	3	4	5	6	7	8	9	10	11	12
$x_{1,t}$	5	5	5	5	5	5	5	5	5	5	5	5
$x_{2,t}$	5	5	5	5	5	5	5	5	5	5	5	5
$x_{3,t}$	5	5	5	5	5	5	5	5	5	5	5	5
$x_{4,t}$	7	7	7	7	7	7	7	7	7	7	7	7

The loss function due to nominal values of x and u is -367.5

The results after 3rd iteration are given in Table 4.3

TABLE 4.3

t	1	2	3	4	5	6	7	8	9	10	11	12
$x_{1,t}$	0	2	4	4	3	2	1	0	0	0	0	0
$x_{2,t}$	0	3	6	6	6	5	4	3	2	1	0	0
$x_{3,t}$	2	2	2	1	0	0	0	0	0	1	0	0
$x_{4,t}$	10	10	6	5	5	5	5	5	4	2	0	5

The coefficient C_{ft} matrix has been supplied in literatures cited above. The transport function or the law of continuity for the four reservoirs is given as

$$\begin{bmatrix} x_{1,t+1} \\ x_{2,t+1} \\ x_{3,t+1} \\ x_{4,t+1} \end{bmatrix} + \begin{bmatrix} x_{1,t} \\ x_{2,t} \\ x_{3,t} \\ x_{4,t} \end{bmatrix} + \begin{bmatrix} Y_{1,t} \\ Y_{2,t} \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} u_{1,t} \\ u_{2,t} \\ u_{3,t} \\ u_{4,t} \end{bmatrix} \quad (4.2)$$

It will be seen that the loss function is linear i.e. L=cu where C is a contant. For the nominal trajectory u and x, this linear function can be approximated to a quadratic function at the points (x,u) as

$$F(u) = \frac{1}{2}\left(\frac{C}{u-\gamma}\right)u^2-\left(\frac{C}{u-\gamma}\right)\gamma u$$

In the problem γ = 20(assumed)

TABLE 4.3 contd.

t	1	2	3	4	5	6	7	8	9	10	11	12
$u_{1,t}$	0	0	2	3	3	3	3	2	2	2	2	2
$u_{2,t}$	0	0	3	3	4	4	4	4	4	4	3	3
$u_{3,t}$	0	0	4	4	4	4	4	4	3	3	3	3
$u_{4,t}$	0	4	7	7	7	7	7	7	7	7	0	0

Note: The above values are given in whole numbers in order to have a clear visual perception of the reservoir operation.

The loss function after 3rd iteration is -400.2

5. CONCLUSION

The contribution of the present paper is to demonstrate the practical merits of computation of an OCP by constrained differential dynamic programming by reporting solution of a four reservoir problem.

(a) DDP overcomes the " curse of dimensionality" in a large scale OCP.

(b) The DDP is globally convergent.

(c) It is not necessary to discretise the state and policy spaces.

Considering the features as well as considering the computing experience in solving a test case in Section-4, it is clear that the DDP is by far the best method currently available for optimum control problem where the second derivatives of loss and state-transition functions exist and readily evaluated.

The procedures, as already stated earlier, assumed convexity of both loss function and constraints. Besides, it is also assumed that the first and secondderivativesof these functions exist and are continuous. The input to the problem being deterministic, this will not be able to take care of the stockastic inputs. Considering these the following areas of future researches are recommended.

(1) Development of an effective algorithm to deal with nonconvexity.

(2) Development in areas involving non-differentiable objective function.

(3) Continuation of research in real time operation to incorporate adaptive forecast in an interactive mode.

NOTATIONS

A_t	=	nth order square coefficient matrix of quadratic approximation at stage t
B_t	=	n x m coefficient matrix of quadratic approximation at stage t
C_t	=	mth order square coefficient matrix of quadratic approximation at stage t
D_t	=	m-tupple coefficient vector in the quadratic approximation at stage t
E_t	=	n-tupple coefficient vector in the quadratic approximation at stage t

$f_t(x,u)$ or f_t	=	p tupple constraint function
$h_t(x_t,x_{t+1})$	=	Function determined by the continuity equation $T_t(x,u) = x_{t+1}$
$J(u)$	=	Objective function or Loss function
$LP(.)$	=	Constant and linear parts of Taylor series expansion of (.)
N	=	Total number of stages, a positive integer
$\hat{Q}_t(x,u)$	=	$QP(Q_t)$
$QP(.)$	=	Linear and quadratic parts of Taylor series expansion of the function(.)
t	=	Discrete time variable
$T_1(x,u)$or T_1 or T	=	Transport function or continuity function
$\bar{u}$	=	Nominal policy
u'	=	Successor policy
u_t	=	Policy variable at stage t
u^*	=	Optimum policy
$V_t(x,u)$	=	Return function at stage t
$\bar{x}_t$	=	Nominal value of state variable at stage t
x'	=	Successor trajectory determined by u'
x_t	=	State variable at stage t
x^*	=	Strategy determined by u*

REFERENCES

1. Bellman, R.,Dynamic Programming, Princeton University Press, Princeton, N.J.,1957.

2. Heidari, M.,V.T.Chow, P.K.Kokotovic, and D.D.Meredith, Discrete Differential Dynamic Programming Approach to Water Resources Systems Optimisation, Water Resources Res.7(2), 273-282,1971.

3. Jacobson, D.,and D.Mayne, Differential Dynamic Programming, Academic Press, 1970.

4. Larson, R., Slate Increment Dynamic Programming, Elsevier, New York, 1968.

5. Murray, D.,S.J. Yakowitz,Constrained Differential Dynamic Programming and its Application to Multireservoir Control, Water Resources Res.,15(4), 1017-1027, 1979.

6. Yakowitz, S.J., The stagewise Kuhn-Tucker conditions and Differential Dynamic Programming, IEEE Trans. on Aut. Cont. A31(1), 25-30, 1986.

7. Ohno, K., A new Approach to Differential Dynamic Programming for Discrete Time Systems, IEEE Trans. on Automat Control, AC-23, 33-47, 1978.