

Stochastic seasonal planning in multireservoir hydroelectric power systems by differential dynamic programming†

A. GJELSVIK‡

Keywords: power system, water resources, stochastic optimization, incremental water value, differential dynamic programming, linear programming.

A first-order differential dynamic programming (DDP) algorithm is used for computing optimal control for a five-reservoir system, where the stochastic inflow process has been approximated by a few discrete disturbance values in each time step. The method is found to be faster than linear programming, previously tried on the same system model.

1. Introduction

Due to its complexity, the operation planning problem in a hydroelectric power system is usually divided into an hierarchy of subproblems, according to different time scales involved and the model complexity needed (e.g. Delebecque and Quadrat 1978). In this paper we shall deal with the medium-term planning, which is concerned with the use of the seasonal storage reservoirs, with a simplified description of the rest of the system. Typical planning periods are from about half a year up to one year, and the time steps used are usually one week or one month. The goal is to minimize the expected value of the variable operating costs for the power company.

In the long- and medium-term planning one has to take into account the stochastic nature of the inflows to the reservoirs. The releases through the power stations are restricted and the reservoir storages have upper and lower limits. The formulation of the optimal planning problem thus leads to a stochastic optimal control problem which has both state and control constraints; a very difficult problem. As the fluctuations in yearly inflow, for instance, may be of the same size as the capacity of the reservoir, the noise cannot be regarded as small. Moreover, it may be sequentially correlated and non-gaussian.

The most general way of solving such an optimal control problem, is via dynamic programming (DP) or similar procedures, e.g. the 'incremental water value method' (Lindqvist 1962). This also furnishes a feedback policy, which is highly desirable. Unfortunately one runs into the well-known dimensionality difficulties of dynamic programming when one tries to deal with systems containing more than one or two reservoirs. There are two ways of dealing with this problem. Either one can aggregate all the reservoirs into an 'equivalent' one-reservoir system, and use dynamic programming, or one can retain the reservoirs and make other approximations, both in modeling and optimization. In this paper, we shall take the second approach.

Received 24 May 1982

† This work has been sponsored in part by The Royal Norwegian Council for Scientific and Industrial Research through the project SIMULERINGSMODELL SYSTEM NORGE (SIMULATION OF THE NORWEGIAN POWER PRODUCTION SYSTEM). Sør-Trøndelag Kraftselskap has kindly permitted the use of data for their system.

‡ The Norwegian Research Institute of Electricity Supply (EFI), Trondheim.

Many solution procedures for the multi-reservoir case have been suggested in the literature. ReVelle, Joeres and Kirby (1969), Loucks and Dorfman (1975), Dillon, Martin and Sjelvgren (1979) are examples of the application of chance-constrained linear programming (CCP). In CCP, feedback may be incorporated through the popular 'Linear decision rule' introduced by ReVelle *et al.* (1969). Willén (1979) discuss these methods.

If one invokes the 'Certainty equivalence principle' (Bryson and Ho 1975) and assume that the future disturbances are known, e.g. equal to their mean values, then the optimization problem becomes deterministic and may be solved in a number of ways. For instance, 'corridor' techniques of DP may be used, as by Heidari *et al.* (1971). Stavnes (1969) use the Pontryagin maximum principle together with penalty functions and a first order gradient method. Strictly speaking, however, the 'Certainty equivalence principle' cannot be assumed to be valid in our case; one reason for this are the state constraints.

Delebecque and Quadrat (1978) and Falgarone and Lederer (1978) use decomposition techniques, whereby the problem is reduced to a problem in team theory. Another decomposition method is that described by Egeland *et al.* (1981). Takeuchi and Moreau (1974) have analysed a water supply system using a combination of linear programming and dynamic programming. Common to all the mentioned approaches are that they attack the problem via various approximations. Since different assumptions are made, it is often difficult to compare the methods.

In this paper we describe an approach to the multi-reservoir case where we approximate the stochastic inflow process by a discrete time and discrete state Markov model. This means that the inflow distribution is replaced by a few discrete values in each time step. This leads to a finite number of realizations. The cost of each realization is computed and the expected cost found, in terms of the controls for each realization. The optimal values of these controls for the different realizations may then be found by minimization of the expected cost, and we describe a differential dynamic programming (DDP) algorithm for doing this. The inflow model is a special case of models described by Jacobson and Mayne (1970) for use with differential dynamic programming. DDP has also been applied to the hydro power planning problem by Martin and Dillon (1977), but our approach differs from theirs in that we use discrete time, and in the treatment of the state constraints. A similar inflow model has earlier been used in conjunction with linear programming (Gjelsvik and Haugstad 1980), and an aim of the present work is to see if a DDP approach can reduce computer requirements.

Hence we shall not investigate the validity of the inflow model, instead we will concentrate on the optimization when the inflow model is taken for granted. It may be said that the inflow model is crude, but as we have seen, the problem is so difficult that simplifications are needed. So we use the present model because it is computationally feasible, while not treating the inflow as completely deterministic. In § 2, we describe the power system. § 3 deals with the DDP-algorithm used, and § 4 gives results of test runs for a five-reservoir system.

2. Mathematical problem formulation

2.1. Power system description

We consider a system with n hydroelectric power stations, each with one reservoir, with arbitrary hydraulic coupling between the reservoirs. We restrict ourselves to the

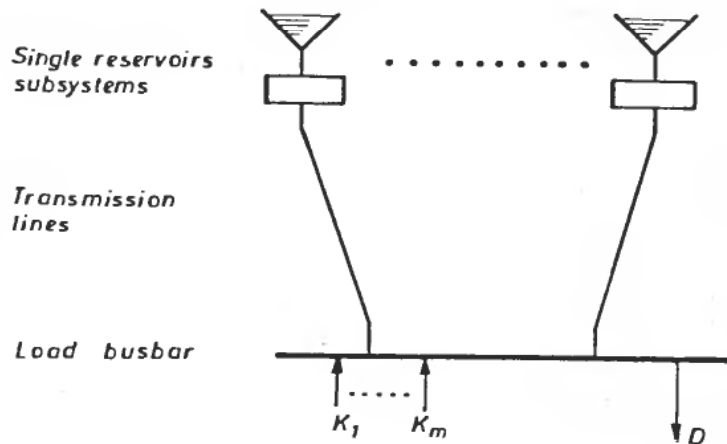


Figure 1. Example of system structure.

case when all power stations are feeding into a concentrated network represented by a single busbar, as shown the example of Fig. 1.

An electric power company has to cover a certain fixed demand for electric power, often called the firm power. To do this, the company may supply power from its own hydroelectric and thermal power stations, and it may buy power from other power companies or from the Power Pool. Surplus power may be sold. We shall consider thermal power as bought power, since its marginal cost is known quite well.

The problem of seasonal planning is to determine a strategy to tell how much power should be bought or sold next week or month and how much water should be released from each storage reservoir, so that the expected value of the operating cost is minimized.

The decision-dependent operating cost for a certain time period is the cost of buying power (including thermal generation), minus the income from selling surplus power, and minus the value of the water remaining in the reservoirs at the end of the period. Let the time period be divided in N time steps. We then seek to minimize

$$J = E \left\{ -\Phi(x(N)) + \sum_{k=0}^{N-1} L(k) \right\} \quad (1)$$

Here $E\{\cdot\}$ denotes expectation. $x(N)$ is the final state of the system, and $L(k)$ is the net cost of buying power in the k th time step. $\Phi(x(N))$ is a scalar function reflecting the final value of the water. Actually we have a problem with an infinite horizon that we approximate by a problem with a finite horizon at $k = N$. In multireservoir models the choice of Φ is one of the main difficulties; this will be commented on in § 4. No discount factor is included in (1). This is probably fair, as the time horizon is only of the order of a year away.

Let there be m 'types' of power exchange, each with constant marginal price. Then $L(k)$ is given by

$$L(k) = \sum_{i=1}^m c_i(k) K_i(k) \quad (2)$$

and the power balance equation is

$$\sum_{i=1}^n P_i(k) + \sum_{t=1}^m K_t(k) = D(k) \quad (3)$$

Here

m = number of exchange types

n = number of power stations

$c_t(k)$ = price of energy of the t th type in time step k

$K_t(k)$ = amount of energy bought of the t th type in time step k

$D(k)$ = electric energy demand in time step k

$P_i(k)$ = electric energy delivered to the busbar from the i th power station in time step k

The future c_t s will not all be accurately known, but we shall treat them as known here. The demand D is also uncertain to some extent, but will likewise be regarded as known. Thus the only stochastic element in the problem formulation is the inflow, which is believed to be by far the most important stochastic variable in the real problem.

If the firm power demand cannot be covered, we have rationing, and it is common to include this as types of exchange in (3), so that formally the power demand is always met. In (2) these terms act as a sort of penalty function. The associated cost coefficients are chosen high to indicate high social costs of shortage. Selling power means that the corresponding K_t is negative, but by suitable shifting of the zero for each such term in (3) we may regard all K_t as non-negative. D may then not be equal to the 'firm' power demand. We obtain the following constraints:

$$\left. \begin{aligned} P_i^{\min} &\leq P_i(k) \leq P_i^{\max} \\ 0 &\leq K_t(k) \leq K_t^{\max} \end{aligned} \right\} \quad (4)$$

We define

$x_i(k)$ = storage in the i th reservoir at the end of the k th time step, in volume units (e.g. Mm³)

$q(k)$ = vector of releases with components $q_i(k)$

$q_i(k)$ = release from the i th reservoir in time step k , in volume units

$v(k)$ = noise vector with components $v_i(k)$

$v_i(k)$ = controlled inflow to the i th reservoir during the k th time step, in volume units

$z_i(k)$ = overflow from the i th reservoir during the k th time step

The state of the reservoirs is then given by the state vector $x(k) = [x_1(k), \dots, x_n(k)]^T$. In general there may also be other states, e.g. for description of the inflow process.

The reservoir system will evolve according to the equation

$$x(k+1) = x(k) + Bq(k) + B'z(k) + v(k) \quad (5)$$

subject to

$$x_i^{\min} \leq x_i(k) \leq x_i^{\max} \quad i = 1, \dots, n \quad (6)$$

and

$$z_i(k) \geq 0 \quad i = 1, \dots, n \quad (7)$$

B and B' are reservoir coupling matrices with elements b_{ij} and b'_{ij} respectively.

$$b_{ij} = \begin{cases} 1 & \text{if } q_j \text{ enters the } i\text{th reservoir} \\ -1 & \text{if } q_j \text{ leaves the } i\text{th reservoir} \\ 0 & \text{else} \end{cases} \quad (8)$$

b'_{ij} has a similar definition with z instead of q . q_j and z_j do not necessarily go the same way.

The endpoint states $x(N)$ are considered free, except that (6) must be fulfilled, and it may be required that

$$x_i(N) \geq x_i(N)^{\min} \quad (9)$$

Here and in the following we shall neglect the 'uncontrolled' inflow to a power station. This is inflow that does not come through a reservoir and must be used instantaneously. Inclusion of this into the present model is considered straightforward, however.

Let q be the mean water flow through a power station during one time step and P the mean electric power output. Let η be the efficiency of the power station, so that $P = \eta q$. We neglect water head variations, and assume that

$$\eta = \text{const.} \quad (10)$$

for $P < P_0$, where P_0 is the point of maximum efficiency. Above P_0 we take the efficiency as (El-Hawary and Christensen 1979)

$$\eta = \eta_0 - a(P - P_0)^2 \quad (11)$$

where η_0 and a are constants.

2.2 Inflow modeling

We assume that the inflow vector v can take on only discrete values d_1, \dots, d_A and further that v can change only at times k_1, k_2, \dots, k_I in the period $k = 1, 2, \dots, N-1$. Here $v(k_i)$ is the value of v shortly after the transition at time k_i . If $v(k_i) = d_r$ we have $v(k) = d_r$, $k = k_i, \dots, k_{i+1} - 1$. A is the number of different d -vectors, and I is the number of disturbance transition points. We define $k_0 = 0$.

The transition of v from d_r to d_s at time k_i has probability

$$\text{prob}(v(k_i) = d_s | v(k_{i-1}) = d_r) = p_{rs}^i \quad (12)$$

where

$$\sum_{s=1}^A p_{rs}^i = 1 \quad r = 1, \dots, A, i = 1, \dots, I \quad (13)$$

$v(k)$, d_r and d_s are vectors with dimension n .

In the following we shall assume that $v(0)$ is given. Figure 2 schematically shows how the state variable trajectories decompose when there are two possible disturbances in each transition ($A = 2$). We shall refer to a line segment of Fig. 2 as a 'branch'. Thus in Fig. 2 there are seven branches. We assume that the value of the disturbance is known to the controller immediately after a transition, and the state of the system is then given by x and v .

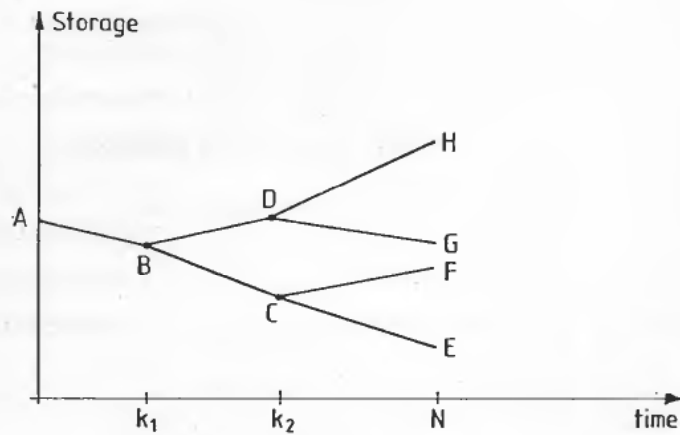


Figure 2. Example of multiple trajectories in the case of two disturbance values at each transition point.

2.3. Control policy

We shall choose as control vector $u = [P^T, K^T]^T$ and the $q_i(k)$ will be considered as functions $q_i(P_i(k))$. Feedback is introduced in that there is a separate control vector $u(\omega, k)$ for each disturbance value, that is, one control vector for each branch in Fig. 2. Here ω denotes one disturbance sequence. Let Ω denote the set of all ω . $u(\omega, k)$ is chosen so that the power balance constraint is satisfied for each realization $\omega \in \Omega$.

In practical operation one would most likely use only $u(1)$ out of the computed controls. When time step no. 1 is finished, one would translate the time scale so that the next time step becomes the first and repeat the calculations.

3. Solution by differential dynamic programming

3.1. The differential dynamic programming algorithm

To solve the optimal control problem we use a discrete time version of the method described by Jacobson and Mayne (1970). We employ a first-order method, with small variations in control. Here we outline the algorithm, while some details are given in the Appendix.

Assume that a control policy π is given. Let $J_k(x(k), v(k))$ be the expected cost of taking the system from state $[x^T(k), v^T(k)]^T$ to a feasible final state using policy π . Then

$$J_k = E \left\{ -\Phi(x(N)) + \sum_{i=k}^{N-1} L(u(i), k) \right\} \quad (14)$$

Since π need not be optimal, J_k is not necessarily the optimal cost function.

Let $\bar{\pi}$ be a nominal policy and let \bar{u} and \bar{x} be the corresponding nominal controls and states, respectively. Write the transition equation of the system as $x(k+1) = f(x(k), u(k), v(k), k)$. For a time step k with $k+1 \neq k_1, \dots, k_I$ we have:

$$J_k(x, v) = L(u, k) + J_{k+1}(f(x, u, v, k), v) \quad (15)$$

In the DDP method eqn. (15) is expanded in a Taylor series in x and u around the nominal trajectory and the expansion is used for computing an improved control policy. Consider a realization ω of the inflow process. We define

$a(\omega, k)$ = first-order change in cost function J_k due to changes in control

$$\kappa(\bar{x}, \bar{v}, k) = -\frac{\partial}{\partial x} J_k(x, v) \Big|_{x=\bar{x}(k)} \quad (16)$$

$-\kappa$ is equivalent to the quantity V_x used by Jacobson and Mayne (1970). We shall usually suppress the dependence on x and v in the notation. For a hydroelectric power system κ may be termed an incremental water value.

In terms of κ a Hamiltonian is defined:

$$\begin{aligned} H(x(k), u(k), -\kappa(k+1), v(k), k) \\ = L(u(k), k) - \kappa^T(k+1) \cdot [x(k) + Bq(k) + B'z(k) + v(k)] \end{aligned} \quad (17)$$

In each time step improvements $\delta u(k)$ in control are computed by minimizing

$$\begin{aligned} \Delta H(k) \equiv H(\bar{x}(k), \bar{u}(k) + \delta u(k), -\kappa(k+1), v(k), k) \\ - H(\bar{x}(k), \bar{u}(k), -\kappa(k+1), v(k), k) \end{aligned} \quad (18)$$

ΔH is the change in H due to changes in u with fixed x . To take care of the power balance constraint (3) and the state variable constraints (6) and (9), we define an augmented Hamiltonian function by

$$\begin{aligned} \tilde{H}(k) = H(k) + \lambda(k) \left[D(k) - \sum_{i=1}^m K_i(k) - \sum_{i=1}^n P_i(k) \right] \\ + \mu^T \cdot [x(k) + Bq(k) + B'z(k) + v(k) - x^{\max}] \\ + \psi^T \cdot [x^{\min} - x(k) - Bq(k) - B'z(k) - v(k)] - \nu^T \cdot z(k) \end{aligned} \quad (19)$$

Here μ , ψ and ν are vectors of Kuhn-Tucker multipliers (Luenberger 1972). The recursive equations for a and κ become

$$\kappa(k) = -\frac{\partial \tilde{H}}{\partial x(k)} \quad (20)$$

$$a(k) = a(k+1) + \Delta H(k) \quad (21)$$

At each branch point a and κ follows the 'addition rule' (Jacobson and Mayne, 1970)

$$\kappa(\bar{x}, d_r, k_i^-) = \sum_s p_{rs} \kappa(\bar{x}, d_s, k_i^+) \quad (22)$$

$$a(\bar{x}, d_r, k_i^-) = \sum_s p_{rs} a(\bar{x}, d_s, k_i^+) \quad (23)$$

Summary of the algorithm

(1) With given $x(0)$ solve the system equation

$$x(k+1) = f(x(k), u(k), v(k), k)$$

in this case eqn. (5) in forward time using nominal controls $\bar{u}(\omega, k)$ for all $\omega \in \Omega$ and $k=0, \dots, N-1$.

- (2) At the final time, put

$$\kappa(N) = \frac{\partial \Phi(x)}{\partial x} \bigg|_{x=\bar{x}(N)} \quad (24)$$

and solve backwards on each trajectory branch the eqns. (20) and (21). Where branches meet, use (22) and (23). If the nominal trajectory hits a state boundary, then κ is computed via the necessary conditions (A3)–(A4).

- (3) In forward time, for all ω compute changes δu to minimize ΔH and compute new states $x(\omega, k)$ with $u + \delta u$ instead of \bar{u} . When state variable limits are met, proceed as described in the Appendix.

Put $a(0)=0$ and solve equation (A 9) along each realization, giving $a(x, d, k)$ for $k=0, 1, \dots, N-1$ and all ω . Also compute the cost $J(\omega)$ for each realization. Compute the new expected cost, J , and the estimate of the reduction in expected cost (α) as

$$J = E\{J(\omega)\} = \sum_{\omega \in \Omega} p(\omega) J(\omega)$$

$$\alpha = E\{a(x, d, N)\} = \sum_{\omega \in \Omega} p(\omega) a(x, d, N)$$

Here $p(\omega)$ is the probability of the realization ω . $p(\omega)$ may be calculated from the inflow transition probabilities using Bayes' rule.

- (4) α is now the estimated cost reduction for this iteration. Calculate $\Delta J = J - \bar{J}$, the actual change in cost. If $\Delta J < 0$ and

$$\frac{|\Delta J|}{|\alpha|} > c \quad (\text{e.g. } c=0.5) \quad (25)$$

then the new trajectories and controls are accepted. Otherwise $\epsilon_1(k)$, $k=0, \dots, N-1$ are reduced and step 3 is repeated. If this occurs more than a specified number of times on the same iteration, an error exit is taken.

- (5) If $|\alpha| < \epsilon_0$ then stop. ϵ_0 is a small quantity.

- (6) Set $\bar{x}(\omega, k) = x(\omega, k)$ and $\bar{u}(\omega, k) = u(\omega, k)$ for all $k=0, \dots, N-1$ and all ω . Reset ϵ_1 and ϵ_2 and go to step 2.

A simple way of finding a starting trajectory is to assign preliminary values to κ and perform a forward run with (A 8) and (A 9) relaxed.

Remark

A key point in the algorithm is that the solution is always kept (formally) feasible. This is possible because of the rationing terms in the power balance equation (3), which act as penalty terms in (1). This might lead to oscillations in the incremental cost of power, λ , from one iteration to the next on branches where state limits are met, with corresponding 'bouncing' on the state constraints. ϵ_1 would then have to be reduced and might eventually become so small that little progress would be made. However, this has not been a problem in the tests run.

3.2. Programming of the algorithm

The algorithm described in § 3.1 has been programmed in FORTRAN for a NORD-10/S minicomputer.

The program is restricted to the case of hydraulically parallel power stations. Further, the time between two consecutive disturbance transition points has been taken as one time step. These time steps may be of different lengths. This corresponds to the linear programming approach by Gjelsvik and Haugstad (1980).

In the present program version all quantities, both trajectories and controls, are kept in-core, the space available being 64K of 16-bit words. For five reservoirs we may have approximately 350 branches (refer to Fig. 2) without exceeding the main memory.

4. Test runs

As a test model for our algorithm we have chosen a simplified representation of the system of Sør-Trøndelag Kraftselskap (S-TK) which is a regional power company that supplies power to local power companies.

We emphasize that our main goal is testing the algorithm, hence we will not be too much concerned with details in the modeling, and much of the data that we use are simplified. Many of the simplifications were initially made in order to be able to apply the 'Power Pool optimization model' described by Egeland *et al.* (1981) to the system. This system was also used for the linear programming computations of Gjelsvik and Haugstad (1980).

The simplified model consists of five parallel subsystems, each represented as a single power station. They are: Driva, Nea, Fosen, Lundesokna and an aggregated representation of various small power stations named 'Miscellaneous'. The station Lundesokna represents an aggregate of two cascaded power stations. Nea and Driva are owned together with other power companies, and our representation corresponds to the fractions owned by S-TK. 'Fosen' represents three parallel power stations in the coastal region of Trøndelag.

Two test cases were run, one with starting time at week no. 40 (autumn) and one with starting time at week no. 12 (spring). In both cases the final time is taken as the end of week no. 39 next year.

Power system data

The data used for the power stations are given in Table 1. The overall power station efficiency is represented as in the eqns. (10) and (11) with a in (11) fitted from the data in column 4 of Table 1. In addition to this the transmission losses are included in the power station efficiency function. We assume that the transmission loss for station no. i is quadratic in P_i and that the loss is 15% when the power station is operating at maximum rating.

The weekly firm power demand in GWh/week is approximated by

$$D(k) = 26.1 + 5 \cdot \cos\left(2\pi \frac{k}{52}\right)$$

The cost function L for power exchanges is assumed to be as given in Table 2. The maximum exchange on each step is represented as a percentage of the firm power.

The minimum power production values used (in GWh/week) were: Driva 7.56, Fosen 2.0, Lundesokna 1.68, and zero for the other subsystems. Minimum power production, power demand etc. for time steps more than one week long are obtained by summation of weekly data.

COLUMN	1	2	3	4
Name	Max. output P_{\max} (GWh/week)	Point of best efficiency (P_0) in per cent of P_{\max}	Efficiency below P_0 per cent	Efficiency at P_{\max} per cent
Driva	17.64	90	100	90
Nea	11.76	100	100	90
Fosen	10.95	73	100	90
Lundesokna	10.26	74.5	100	90
Miscellaneous	4.44	72	100	90

Table 1. Data for power stations. The efficiency refers to the power station.

Step no.	Text	Maximum exchange, in per cent of firm power	Marginal cost (øre/kWh)
1	Selling	25.0	2.0
2	Selling	15.0	3.0
3	Selling	10.0	5.0
4	Buying	5.0	8.25
5	Buying	5.0	12.75
6	Rationing	6.0	33.0
7	Rationing	4.0	36.0
8	Rationing	2.2	49.0
9	Rationing	3.1	76.0
10	Rationing	3.1	107.5
11	Rationing	6.6	157.0
12	Rationing	75.0	190.0

Table 2. Cost function for exchange of power. Step 3 is not present in weeks 18–39.

Final value function

We assume the final value function to be additive, as

$$\Phi(x) = \sum_{i=1}^5 \phi_i(x_i) \quad (26)$$

An aggregated single-reservoir model of the system existed, and expected incremental water values were computed for this model. The sample runs use the end of week no. 39 as final time, and the computed incremental water values at the end of this week were assigned to $\partial\phi_i/\partial x_i$ in a manner such that storage in each reservoir has the same incremental value at the same relative filling in per cent. $\partial\phi_i/\partial x_i$ is shown in Fig. 3.

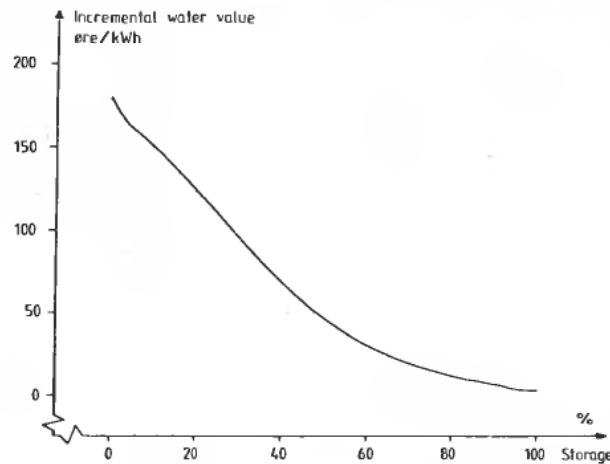


Figure 3. Final expected incremental water value for sample runs.

Inflow data

For each subsystem, historical inflow data for the years 1931–1960 were available, in units of GWh/week. For the sample runs, we have chosen to work with 3 discrete inflow states for the whole system (not 3 independent for each reservoir) in each time step except the first. This gives 121 branches. Using only 3 different inflow states implies that the inflows are treated as nearly fully correlated between reservoirs. The test system lies within one main part of the country, so the correlation will be quite high, but there may be some difference between inland and coastal subsystems, as Nea and Fosen, respectively.

The choice of the inflow states d_1 , d_2 , d_3 and the estimation of the transition probabilities were done as described in Gjelsvik (1980). Essentially, for each time interval between the chosen transition points, the 30 observations are ranked according to total inflow and then divided in 3 groups (with 9, 12 and 9 realizations respectively). d_i is taken as mean values in the i th group. After the groups for each time interval are constructed, the p_{ij} s are determined by simply counting transitions between groups according to the observed total inflow series. 30 years of data is somewhat short for this procedure, because 9 probabilities are estimated from a total of only 30 transitions.

Test case I

Five time steps were chosen, starting at weeks 40, 41, 12, 20 and 26 respectively. Here the 3rd time step enters a new year.

The initial storages were chosen as 93% of the upper limits. Figure 4 shows the extreme state trajectories obtained. All the other trajectories lie in the hatched area. Week numbers refer to end of the week in question. Figure 5 shows the domain of the mean power production of each subsystem, in GWh/week. It should be noted that the extreme control values for a power station do not necessarily correspond to the extreme storages for that power station. Figure 6 shows the variation of the incremental cost of power.

Test case II

Five time steps were used, starting at weeks no. 12, 13, 18, 24 and 31 respectively. The initial storages were chosen as: Driva 45%, Nea 20%, Fosen 36%, Lundesokna

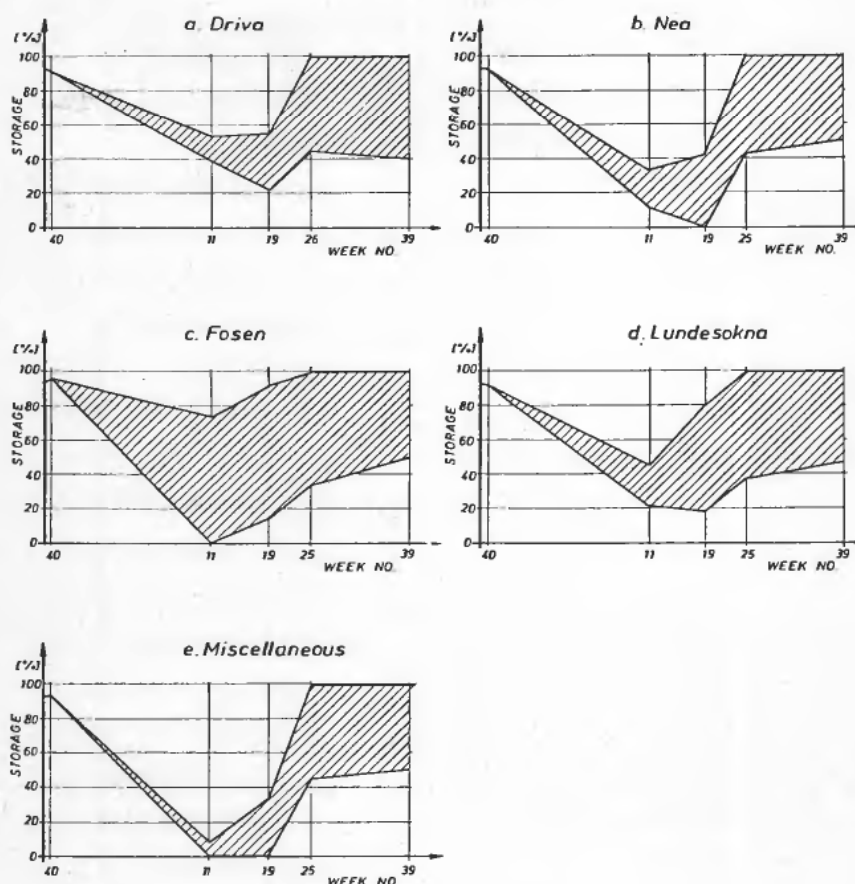


Figure 4. Domain of state trajectories in test run I.

30% and Miscellaneous 5%. The domain of the state trajectories obtained are shown in Fig. 7.

Discussion of the results and the model

In both test cases, state trajectories have quite a large variation during spring and summer. This is because the inflow in these seasons is large and often has a stochastic variation that is a large fraction of the reservoir size. For the Driva power station, large variance in trajectories may be partly due to the high minimum production required. (It is possible that in the real system this requirement exists only during part of the year.)

In test case I, the state ranges for the Fosen subsystem between the end of week no. 40 and the end of week no. 11 in the next year are unrealistically large. The reason for this is that in this coastal subsystem there is quite high inflow during winter. Shorter time steps should have been used here, but the lengths of the time steps chosen so as to have relatively short time steps in the spring and summer seasons, where the total system inflow varies rapidly. It is only the Fosen subsystem that has this large (compared to reservoir size) inflow variation during the winter season.

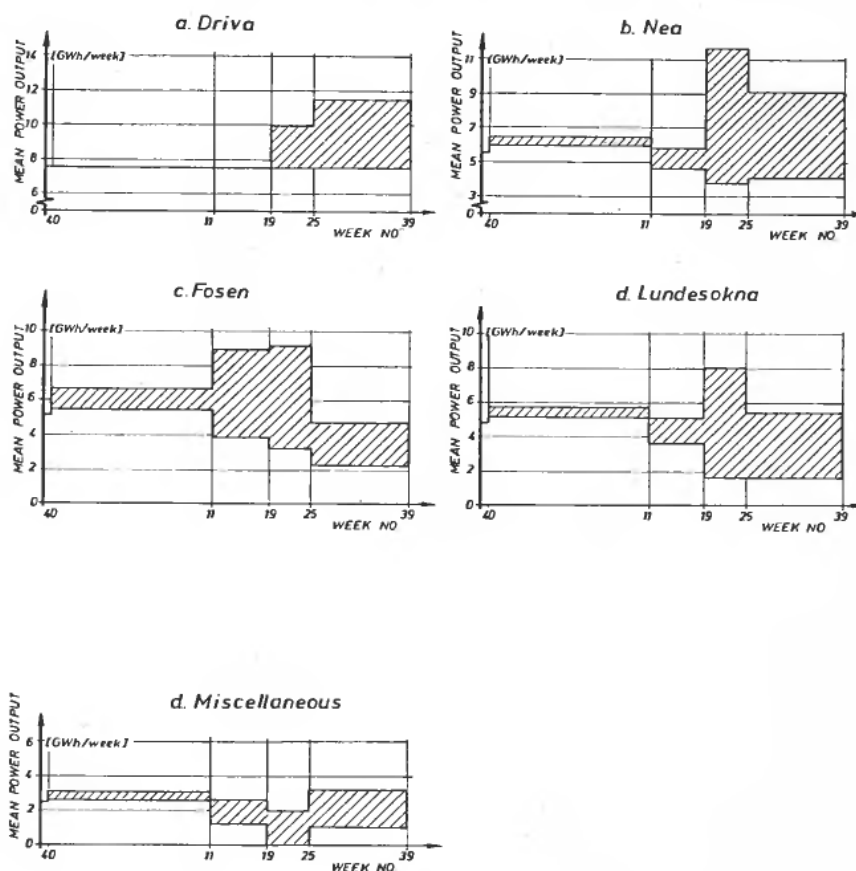


Figure 5. Domain of mean power production in test run I.

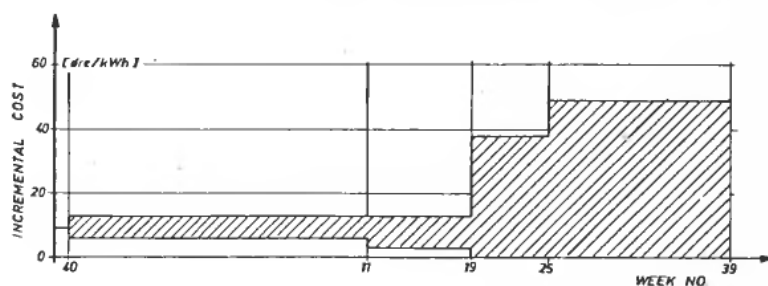


Figure 6. Domain of incremental cost of power, test run I.

However, the variation in trajectories shown in Figs. 4 and 7 is not likely to appear in practice, because in a real-world application computations would be repeated each week, with new inflow forecasts for the first week (or the first few weeks). Hence the computed controls for the first time step are of primary interest.

Test case I was run with several different starting nominal controls to see whether this would influence the final answer. Table 3 shows the controls finally obtained for the

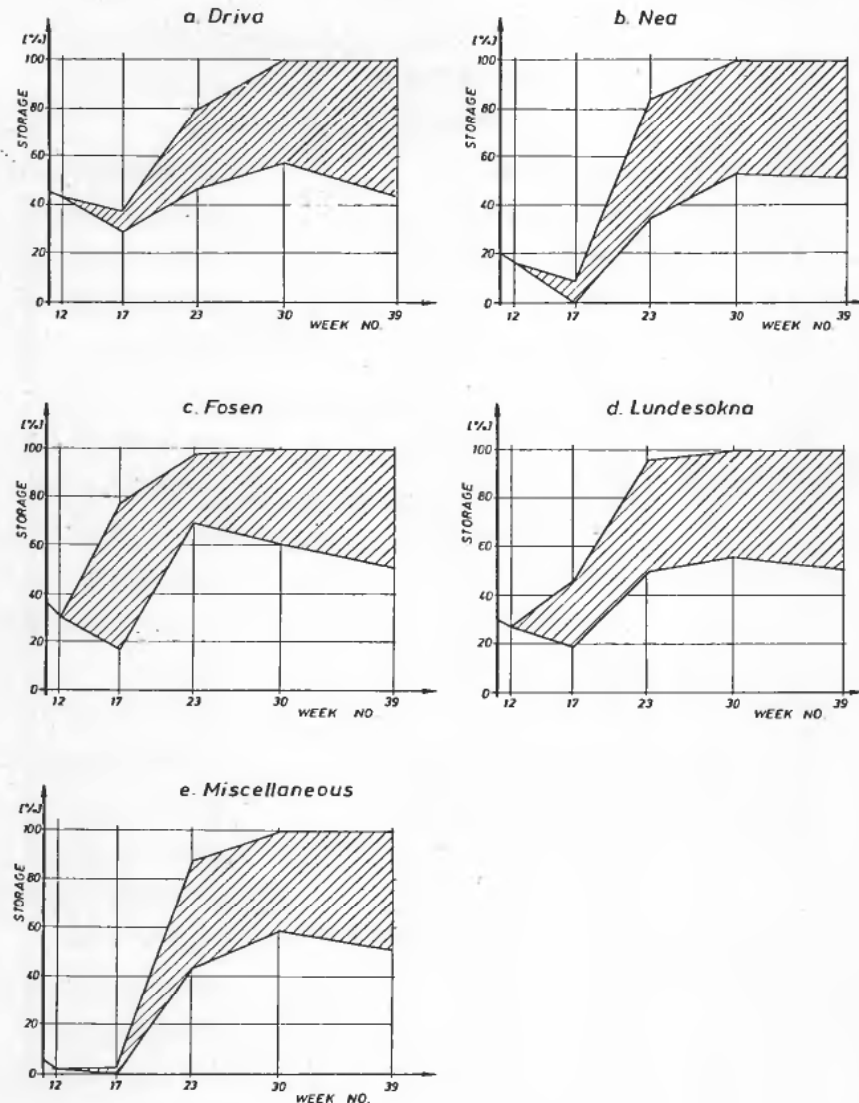


Figure 7. Domain of state trajectories in test run II.

first time step, and one sees that the results differ slightly. This is not unexpected, because there are so few constraints on the electrical connections within the system. This makes it cheap to shift hydroelectric power production from one station to another, shifting in the reverse direction in another time step, to get the same final storage. However, the variations in Table 3 are small compared with other approximations in the model.

The controls computed for the first step will naturally depend upon the inflow modeling, both the number of time steps and the number of discrete inflow values used in each time step. This sensitivity has not been investigated here. In general one can say that one should use as many time steps and discrete inflow values as computationally feasible. It is especially important to have as short time steps as possible around the time when spring melting starts.

	Starting trajectory no.			
	1	2	3	4
Driva	7.56	7.56	7.56	7.56
Nea	5.53	5.79	5.72	5.63
Fosen	5.07	4.91	5.07	5.10
Lundesokna	4.78	4.56	4.66	4.85
Miscellaneous	2.43	2.54	2.35	2.23
Exchange	14.69	14.69	14.69	14.69

Table 3. Controls for the first time step for various initial nominal trajectories. In GWh/week.

As our model uses mean controls over long time intervals, the emptying of a reservoir is not penalized if enough electrical energy can be produced at the other power stations. However, in a real system the emptying of a reservoir means that the associated power station cannot participate in the covering of the demand in the heavy load hours. Thus, while on the mean enough energy is available, the production capacity is too small in hours of high demand. Our model can be adapted to this case in several ways. One is to specify a minimum mean production, as we have done for some power stations, it being understood that this production corresponds to the need in the hours of high demand. Another way is to specify a minimum storage greater than zero, to act as a buffer against large variations in inflow. A third method would be to split each time step in two, one representing accumulation of hours of low load, the other accumulation of hours of high load.

A major problem in seasonal planning with a finite horizon is the choice of the final value function $\Phi(x)$. As in the test case, incremental water values from long-term planning by an aggregated model may be used, but the problem is how to assign values to each subsystem. In general, one cannot assume that the additive form (26) is valid. Final value functions may also be constructed from operational experience, by assigning values that favours final storages in some desired interval. We have not investigated the sensitivity of the results of the test runs to changes in the final value function.

Performance of the optimization algorithm. Computing time

Several runs of test case I were done to determine the best values of ϵ_1 in (A 8) and ϵ_2 in (A 9). It turned out that the algorithm performs well with ϵ_2 so large that the constraint (A 9) never becomes active. For ϵ_1 we put

$$\epsilon_1(k) = \epsilon'_1 T_k$$

where T_k is the length of the k th time step in weeks. Values of ϵ'_1 between 0.02 and 0.05 gave good results. Initial cost reduction may be slightly faster with a larger ϵ'_1 , but computing time will increase, since it may become necessary to repeat step 3 of the algorithm.

In the test runs we put ϵ_0 so small that the convergence criterion in step 5 was never fulfilled. Instead the algorithm stopped at the specified maximum number of iterations.

Figure 8 shows the expected cost as a function of the number of iterations for one run of test case II. It seems that about 20 iterations is enough, taking into account that we have made many approximations and that the system is stochastic. The computing time for 20 iterations in test run II was 1.5 minutes of central processor unit (CPU) time, solution output included. The total 80 iterations took about 7.5 minutes. The computer time needed for one iteration in the test cases was about 2.7 seconds. It is reasonable that the computing time should be roughly proportional to the number of branches, in our case we get 22 ms CPU-time per branch. There is some scope for improving this. The program could also be extended to cover more complex systems, including systems with cascaded reservoirs, but then it will be more complex and time-consuming.

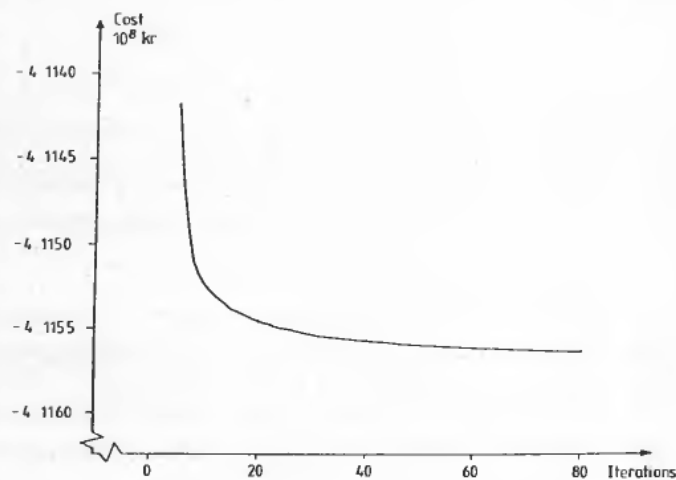


Figure 8. Expected cost vs. number of iterations, test run II.

A model of the same system with the same number of branches was treated by linear programming (LP) by Gjelsvik and Haugstad (1980). They used the commercial LP-system FMPS on a UNIVAC 1100/21 computer. The final value function was composed of three linear segments for each reservoir, and the power station characteristics were taken as piece-wise linear. CPU-time for optimization was then 8.7 minutes, with a possible reduction if an advanced starting basis had been available. The UNIVAC 1100/21 is from 3 to 10 times faster than the NORD 10/S computer (depending on problem), so referred to the same computer our method may be estimated to give an improvement over LP with a factor of order 5. In addition, non-linear final value functions are easily handled. Part of the improvement occurs because we stop before the ultimate optimum is reached.

5. Conclusion

The DDP-algorithm used for the multi-reservoir system offers considerable savings in computer time and space as compared to linear programming, while at the same time allowing a non-linear final value function. However, it may not be as robust as linear programming.

The approximate inflow model used is crude, and unless one uses many discrete inflow values, one is limited to power systems geographically so confined that their inflows may be regarded as fully correlated. A difficulty with the method, as with most other methods, except the incremental water value method, is the choice of the time horizon for optimization and the assignment of a value function for the final storages.

Appendix

Details of the DDP-algorithm

Necessary conditions for optimality

At the optimum

$$\frac{\partial \tilde{H}}{\partial u_i} = 0 \quad i = 1, \dots, n+m \quad (\text{A } 1)$$

or u_i must be at one of its bounds. When we use the Kuhn-Tucker conditions and eliminate the multipliers between (A 1) and (20), the following necessary conditions for optimality are obtained for reservoir no. i , $i = 1, 2, \dots, n$.

(a) The state boundaries are not hit. $x_i^{\min} < x_i(k+1) < x_i^{\max}$. Then

$$\kappa_i(k) = \kappa_{l(i)}(k+1) \quad (\text{A } 2)$$

(b) There is overflow, $x_i(k+1) = x_i^{\max}$ and $z_i(k) > 0$. Then

$$\kappa_i(k) = \kappa_{l(i)}(k) \quad (\text{A } 3)$$

(c) The reservoir is emptied, $x_i(k+1) = x_i^{\min}$ or it is full without overflow, $x_i(k+1) = x_i^{\max}$ and $z_i(k) = 0$. Then

$$\kappa_i(k) = \lambda \frac{\partial P_i}{\partial q_i} + \kappa_{l(i)}(k) \quad (\text{A } 4)$$

In (c) it is assumed that P_i is not at one of its bounds, which is reasonable. Here $l(i)$ is the index of the reservoir downstream the i th reservoir. $l'(i)$ is the index of the reservoir to which overflow goes. If there are no downstream reservoirs the downstream κ s are put equal to zero. The conditions above are well-known in the context of the 'incremental water value method', see e.g. Gjelsvik (1980).

Computation of improved controls

Introducing $\delta u(k) = [\delta P_1(k), \dots, \delta P_n(k), \delta K_1(k), \dots, \delta K_m(k)]^T$ we obtain from (18) and (17) to first order

$$\Delta H = \text{const} + \sum_{i=1}^m z_i(k) K_i(k) + \sum_{i=1}^n c_i^P(k) \delta P_i(k) \quad (\text{A } 5)$$

where

$$c_i^P(k) = (\kappa_i(k+1) - \kappa_{l(i)}(k+1)) \left(\frac{\partial P_i}{\partial q_i} \right)^{-1} \quad i = 1, \dots, n \quad (\text{A } 6)$$

c_i^P is the incremental cost of hydroelectric power production at the i th power station, $i = 1, \dots, n$. In a conventional DDP-algorithm, the system equation $x(k+1) = f(x(k), u(k), v(k), k)$ is solved in the forward direction, using nominal controls, and the κ -equations in backward time, computing changes to the controls.

We have modified this standard procedure somewhat. Since κ does not depend explicitly on x except at the final time, or when a storage limit is met, the optimal controls for given κ do not depend on x either. Hence they may be computed on the next forward run. If a state variable constraint is hit on the new forward run, then the controls are changed so as to keep the trajectory in the feasible domain.

δu must be kept small in order that δx be small. When δu is otherwise unrestricted, a natural choice is

$$\delta u(k) = -\epsilon(k) \frac{\partial \tilde{H}(k)}{\partial u(k)} \quad (\text{A } 7)$$

where (k) , $k=0, 1, \dots, N-1$ is 'small'. However, δu given by (A 7) may violate the constraints of type (3), so instead we require, in analogy by (A 7)

$$|\delta P_i(k)| \leq \epsilon_1(k) |c_i^P(k) - \lambda(k)| \quad \forall i, k \quad (\text{A } 8)$$

$\lambda(k)$ is not known, so we approximate it by $\bar{\lambda}(k)$, the value of $\lambda(k)$ in the preceding iteration. To be safe, we also require

$$|\delta P_i(k)| \leq \epsilon_2(i, k) \quad \forall i, k \quad (\text{A } 9)$$

Now the problem is to minimize ΔH of eqn. (A 5) subject to the constraints (3), (A 8) and (A 9) and also the K -constraint of (4).

Since ΔH is linear with linear constraints, it might be minimized by linear programming. However, since the only constraint other than single bounds is eqn (3), we choose to minimize H by loading successively more expensive resources (δP or K) until (3) is satisfied. λ is found as the marginal cost of the last resource loaded.

If a reservoir limit is met on the forward run, the expansion (A 5) for H must be modified. To ensure that the state variable constraints are not violated, we proceed as follows:

Let $x(k)$ be the new state at time k . If $f(x(k), \bar{u}(k), v(k), k)$ is in the interior of the state space, then check the limits of P to be sure that x stays within the interior.

If $f_i(x(k), \bar{u}(k), v(k), k) > x_i^{\max}$ then put $c_i^P = 0$ in (A 5) and constrain δP_i so that $f_i(x(k), u(k), v(k), k) < x_i^{\max}$ is not possible. (In the latter region $c_i^P = \kappa_i(k+1)$, so c_i^P is discontinuous. δP_i is so constrained that we stay on one side of the discontinuity.) If $f_i(x(k), \bar{u}(k), v(k), k) < x_i^{\min}$ then \bar{P}_i is decreased so as to make $f_i(x(k), \bar{u}(k), v(k), k) = x_i^{\min}$. The δP_j^{\max} for the other power stations have to be chosen so they are able to take up the deficit if that is economical.

In the forward direction, the recursion formula (21) for $a(k)$ may be written

$$a(k+1) = a(k) - \Delta L(k) - \kappa^T(k) \cdot [x(k) - \bar{x}(k)] + \kappa^T(k+1) \cdot [x(k+1) - \bar{x}(k+1)] \quad (\text{A } 10)$$

with

$$\Delta L(k) = L(\bar{u}(k) + \delta u(k)) - L(\bar{u}, k)$$

REFERENCES

- BRYSON, A. E., JR. and HO, Y. C. (1975). *Applied Optimal Control* (revised printing). John Wiley.
- DELEBECQUE, F. and QUADRAT, J. P. (1978). Contribution of stochastic control singular perturbation averaging and team theories to an example of large-scale systems: Management of hydropower production. *I.E.E.E. Trans. Automatic Control*, **23**, 209-221

- DILLON, T. S., MARTIN, R. W. and SJELVGREN, D. (1980). Stochastic optimization and modeling of large hydrothermal systems for long-term regulation. *Electrical Power & Energy Systems*, **2**, 2-20.
- EGELAND, O., HEGGE, J., KYLLING, E., and NES, J. (1982). *The Extended Power Pool Model*. CIGRÉ Paper 32.14.
- EL-HAWARY, M. E., and CHRISTENSEN, G. S. (1979). *Optimal Economic Operation of Electric Power Systems*. Academic Press.
- FALGARONE, F., and LEDERER, P. (1978). Optimal operation model of the large French hydro-power system. Sixth Power Systems Computation Conference, Darmstadt 21st-25th August, IPC Science and Technology Press, 568-575.
- GJELSVIK, A. (1980). Stochastic long-term optimization in hydroelectric power systems. Dr.ing. Thesis, University of Trondheim, The Norwegian Institute of Technology, Div. of Eng. Cybernetics, Trondheim.
- GJELSVIK, A., and HAUGSTAD, A. (1980). Long-term planning by stochastic linear programming. (In Norwegian.) The Norwegian Research Institute of Electricity Supply (EFI), techn. report no. 2590, Trondheim, Norway.
- HEIDARI, M., CHOW, W. T., KOKOTOVIC, P. V., and MEREDITH, D. D. (1971). Discrete differential dynamic programming approach to Water Resources systems optimization. *Water Resources Research*, **7**, 273-282.
- JACOBSON, D. H., and MAYNE, D. Q. (1970). *Differential Dynamic Programming*. American Elsevier, New York.
- LINDQVIST, J. (1962). Operation of a hydrothermal electric system: A multistage decision process. *A.I.E.E. Journal*, April, 1-7.
- LOUCKS, D. P., and DORFMAN, P. J. (1975). An evaluation of some linear decision rules in chance-constrained models for reservoir planning and operation. *Water Resources Research*, **11**, 777-782.
- MARTIN, R. W., and DILLON, T. S. (1977). Solutions of the problem of stochastic optimal control of hydro-thermal power systems. In *Digital Computer Applications to Process Control*, Van Nauta Lemke, ed., North-Holland Publishing Company, pp. 257-264.
- REVELLE, C., JOERES, E., and KIRBY, W. (1969). The linear decision rule in reservoir management and design. 1, Development of the stochastic model. *Water Resources Research*, **5**, 767-777.
- STAVNES, O. (1969). Optimal control of hydroelectric power systems by the discrete maximum principle (in Norwegian), *Elektroteknisk Tidsskrift*, **82**, 467-469.
- TAKEUCHI, K., and MOREAU, D. H. (1974). Optimal control of multiunit interbasin water resource systems. *Water Resources Research*, **10**, 407-414.
- WILLÉN, A. (1979). A stochastic hydrology approach to the operation of surface water reservoirs. Part 2: Optimization of hydro-electricity systems. University of Uppsala, Department of Physical Geography. UNGI report no. 50. Uppsala, Sweden.