Optimal Control of Water Distribution Systems by Network Flow Theory

SHIN'ICHIRO MIYAOKA AND MOTOHISA FUNABASHI

Abstract—This paper presents a modeling technique and an optimal control scheme for water distribution networks. To overcome the large scale and nonlinearity of the network, a network aggregation method and a two-level control scheme are developed. The first level of the scheme decides operating points using a nonlinear optimization method, where the pressure/flow equations are solved using a high-speed technique derived from network flow theory. The second level is a feedback control around the operating points, which absorbs estimate error and small variations in consumption. The scheme has been implemented on a minicomputer system and is presently in operation.

I. INTRODUCTION

In urban areas, scarcity of water resources has become a serious problem, as such techniques for minimizing water waste are urgently needed. Consumers want to receive water with high pressure, but excess pressure causes water leakage from pipe junctions. It is generally thought that 20 percent of all the water in distribution systems is lost through leakage. For this reason, pressure must be regulated at an adequate level.

Because of the large scale and nonlinearity of these systems, such regulation cannot be realized without advanced control technology. Even for a municipality of 300,000 people and considering pipes of diameter greater than 200 mm, the network has about 700 pipes and 500 pipe junctions. In addition, the nonlinearity between pressure/flow and control variables (valves and pumps) makes on-line control difficult.

Several computer applications for distribution systems have been studied. However, these studies [1]-[4] have not aimed at regulating pressure, but rather at scheduling the operations of reservoirs and pumping devices for economical dispatch. For the purpose, it is not necessary to deal with precise pressure, linearity can sometimes be assumed [2], and reduction of the computation time is not critical.

This paper presents a modeling technique and control scheme for pressure (as well as flow) regulation which solves the above-mentioned large-scale nonlinear problem. A system model must be realized in minimum order for control purposes. The network aggregation method makes such modeling possible. To overcome nonlinearity, a two-level control scheme is introduced, which consists of the setting of operating points and the feedback control around them. A very fast optimization technique to determine the operating points is derived from network flow theory, making on-line control possible.

The following sections describe the modeling technique and the control scheme along with some implementation experience. This scheme has been realized by a minicomputer system with telemetry devices and is presently in operation.

II. MODELING OF DISTRIBUTION SYSTEMS

A. System Equations

A water distribution system can be seen as a network, the arcs of which are pipes and the nodes of which are reservoirs, consumption points, and pipe junctions. Pressure and flow in the network are controlled by valves and pumps. The network equations can be obtained from continuity of flow at each node and balance of pressure between adjacent nodes:

\[ \sum_{j \in \delta^-(i)} x_j - \sum_{j \in \delta^+(i)} x_j = -w_i \quad (i \in N) \]  
\[ p_{\delta^+(j)} - p_{\delta^-(j)} = f_j(x_j, u_j) \quad (j \in B) \]

where

\[ x_j \text{ the flow per unit time at arc } j \text{ (m}^3/\text{s}) \]
\[ p_i \text{ the pressure at node } i \text{ (m} \text{Aq, i.e., 0.1 kg/cm}^2) \]
\[ w_i \text{ the inflow per unit time at node } i \text{ (m}^3/\text{s}) \]
\[ y_i \text{ the consumption flow per unit time at node } i \text{ (m}^3/\text{s}) \]
\[ u_j \text{ the control variable representing the state of valve or pump at arc } j \]

The pressure–flow characteristic function \( f_j \) usually depends only on \( x_j \) [Hazen–Williams relationship (2)] as in

\[ f_j(x_j) = R_j|x_j|^{1.85} \text{sgn}(x_j) \]

where \( R_j \) is a resistance coefficient determined by the roughness, the diameter, and length of the pipe associated with arc \( j \). When arc \( j \) represents not only a pipe but also a valve or a pump, \( f_j \) becomes

\[ f_j(x_j, u_j) = R_j|x_j|^{1.85} \text{sgn}(x_j) + h_j(x_j, u_j) \]

where \( h_j \) is a characteristic curve of a valve or a pump. The valve which plays the main role in the pressure regulation has the characteristic:

\[ h_j(x_j, u_j) = u_j x_j^2 \text{sgn}(x_j) \]

where \( u_j \) is constrained within limits \( u_j < u_j < u_j \).

In (1) and (2), \( p_i \) (\( i \in N \)) and \( w_i \) (\( i \in N \)) are usually observable, since measurement apparatus are attached to the reservoirs. In addition, some \( p_i \) (\( i \in N \)) are measured by pressure gauges.
located in the network. The consumption flows $y_i$ for $i \in N$ are not observable, but their estimates can be given a priori.

To illustrate the scale and features of the system more clearly, we consider the case of Takamatsu City in Japan, where we have implemented our control scheme. In the Takamatsu network (Fig. 1):

1) the number of nodes and arcs are 486 and 654, respectively, after selecting pipes more than 200 mm in diameter;
2) the network includes six reservoirs, eight control valves, two pump stations, and 36 pressure observation points.

For control purposes, the system model (1) and (2) must be reduced. The reasons are twofold. One is that the system model (1) and (2) contains redundant states in terms of controllability and observability because economic limitations do not permit satisfactory allocation of control and observation points. The other is that, even though the system model is controllable and observable, it must be simplified so as to be handled in real time. The following section describes a method for reducing the original system model.

B. Network Aggregation Method

In order to obtain a control model, we shall aggregate a number of nodes among the observation and control points. To accomplish this aggregation, a network decomposition technique is introduced. Each decomposed subarea is considered as a node of the aggregated network model (Fig. 2). Some network decomposition techniques [5] have been studied which take into consideration only network topology. However, we need a method which can deal with distances—such as pressure differences between two nodes—because pressure at different nodes in a given subarea must be nearly the same value.

We define the distance between nodes $k$ and $l$ by the pressure difference between them:

$$d(k, l) = \sum_{j \in P} \Delta p_j$$

where $\Delta p_j$ means pressure loss at arc $j$ and $P$ is the path between $k$ and $l$. We then assume a core node for each subarea and define the "extent" of a subarea as follows:

$$S(A_k) = \max_{i \in A_k} d(i, C_k)$$

where $A_k$ and $C_k$ are the subarea and corresponding core node, respectively. The decomposition $\mathcal{D}$ becomes optimal when it satisfies the following condition:

$$\max_k S(A_k) \rightarrow \min_{\mathcal{D}}$$

Namely, $\mathcal{D}$ is the decomposition which makes the "extent" of all subareas as nearly the same as possible.
In order to decompose the network satisfying condition (8), the Dijkstra method [7], which is the shortest path algorithm, can be adopted. Using the method in parallel, the minimum spanning trees, the roots of which are the core nodes, are obtained. These trees cover the whole network without overlapping, each of them representing a subarea. The decomposition algorithm is as follows.

Step 1: Initially, the potential $\pi_i$ for core nodes and other nodes is assumed as 0 and $\infty$, respectively. Labels of all nodes are set as 0.

Step 2: Find the node with the minimum potential among the nodes labeled 0, then change the label of the node to 1. If such a node does not exist, stop.

Step 3: Search the opposite nodes around the arc connected to the node whose label has just been changed to 1 and update the potential using the following:

$$\pi_{\theta(i)} = \min \{ \pi_{\theta(j)}, \pi_{\theta(j)} + \Delta p_j \}. \quad (9)$$

If the potential is changed, let node $\theta(j)$ belong to the tree which includes node $\theta(i)$, then go back to Step 2.

It is obvious that each node is included in some tree and the network is decomposed by this algorithm. $\pi_i$ means distance between the node $i$ and the corresponding core node.

The optimality in terms of the condition (8) is easily proved as follows. We assume that this decomposition is not optimal. In that case we can reduce the extent of the most extensive subarea by removing the most distant node $i$ from the subarea to some neighboring subarea. This leads to the following equation for some node $k$ in the neighboring subarea:

$$\pi_k > \pi_i + \Delta P_{ki}. \quad (10)$$

where $\Delta P_{ki}$ is the pressure loss at arc $(k,i)$. Equation (10) contradicts Step 3 in the above algorithm, hence the assumption is shown to be incorrect.

After aggregating the nodes included in each subarea, a number of arcs connecting each pair of subareas are aggregated into one arc. As a result of this aggregation, the topology of the network is determined. Resistance coefficients of aggregated arcs are determined so as to approximate the original network. The procedure to determine the aggregated resistance coefficient $R_j$ is formulated as follows:

$$\sum_i \left( \frac{P_{d(i,j)} - P_{d(j,i)}}{R_j \sqrt{x_i x_j}} \right)^2 \rightarrow \min R_j. \quad (11)$$

where for a consumption level $t$, $P_{d(i,j)}$ and $x_i$ are, respectively, pressure of the aggregated network (i.e., pressure at the core node $i$) and flow of the aggregated network (i.e., aggregated flow between subareas). These values are obtained by simulating the original network, namely solving (1) and (2).

The aggregation method is applied to the network in Fig. 1. Pressure loss $\Delta p_j$ in (9) is given by a simulation of the original network. In this simulation, it is assumed that consumption flows are at maximum level and operating points of valves and pumps are optimal for this consumption level (the optimization method is described in Section III). This is because the model error is critical to the maximum consumption flows. The core nodes are set as follows:

i) reservoirs

ii) observation points

iii) control points (both end points of each valve or pump)

iv) pipe junctions and consumption points on the main pipes (the diameter of which is more than 500 mm), where the last setting iv) is imposed so as to preserve the structure of the original network.

In order to assess accuracy of the aggregated model, a simulation study was carried out. Fig. 3 shows the aggregated network which was obtained, the number of nodes and arcs were reduced to 142 and 204, respectively. As a result both the number of state variables (pressure/flow) and the number of state equations (1) and (2) become 346. Both the number of control variables and the number of observed variables remain the same as those of the original network. The model accuracy has been confirmed to be 1.9 mAq in terms of maximum pressure error. This accuracy is considered to be satisfactory for control purposes.

III. CONTROL SCHEME

A. Control Objectives

The primary objective of the distribution control in our study is pressure regulation, namely, to provide an adequate level of pressure at each node under varying conditions of consumption. Especially very high pressure at night, which is due to a decrease in consumption, must be prevented because it causes water leakage and pipe fracture. An additional objective is to control flow from reservoirs so as to best-utilize contracted inflow and minimize usage of owned water.

These objectives are formulated as follows:

$$J = \sum_{i \in N} \gamma_i (p_i - p^*)^2 + \sum_{i \in N_i} \xi_i (w_i - w^*)^2 \rightarrow \min \quad (12)$$

where $p^*$ and $w^*$ are the desired values of $p_i$ and $w_i$, respectively, and $\gamma_i$ and $\xi_i$ are weight coefficients. If necessary, it is possible to take into account minimization of pumping energy by introducing pumping flow into the objective function (12).

B. Control Scheme

A two-level control scheme is introduced to overcome nonlinearity and uncertainty in the system (Fig. 4). The first level concerns decisions regarding operating points and the second is feedback control around them. Since consumption varies 25–100 percent during the day, the nonlinearity among pressure/flow and control variables cannot be neglected. Therefore, mere feedback control is not sufficient. Hence, the determination of operating points by a nonlinear optimization technique is required. The uncertainty is due to the consumption data used in deciding operating points. The inflow from the reservoirs is the only observable flow; the distribution of the consumption is assumed a priori. Discrepancies between computed and actual pressure must be absorbed by a feedback control.

C. Decisions of Operating Points

Finding the operating point requires solving an optimization problem formulated by constraints (1), (2), and the objective function (12). Consumption flows $y_i$ ($i \in N$) are estimated from observed inflows and previously determined distribution data. Although the network has been aggregated, on-line usage still needs a high-speed optimization technique.

Simplex Method

We shall adopt an algorithm approach for solving for the control variables because an analytical solution is impossible. The features of the problem are as follows.

i) It is rather difficult to use analytical gradient formulas because of the implicit relationship between the control variables and the objective function. It requires more than 100 dimensional matrix inverse computations, making on-line usage difficult.

ii) It is necessary to calculate pressure/flow (i.e., to solve (1) and (2) for given $U$) in order to evaluate the objective function (12). This takes up the main portion of the computation time required.

iii) The number of the control variables is rather small; it is only eight in our problem. The numerous state variables are handled only for evaluating the objective function as described in ii).
iv) An approximately optimum point is acceptable as the operating point because the required ranges of pressure/flow are rather wide.

There are several optimization methods [8], [9], [15]: the conjugate gradient method, the steepest descent method, the direct search method, etc. The conjugate gradient method is said to be the most efficient only when analytical gradient formulas can be used (see [8]). However, because of i) it is considered to be unsuitable, and so is rejected. The steepest descent method requires much repetition of objective function evaluation in numerical differential calculus and line search, and is therefore abandoned for the reason ii). The simplex method [9], [15], which is a kind of the direct search method (not a method for linear programming), does not require gradient calculus and its evaluation times of the objective function seem to be less than those of the above methods. In addition, this method is robust in ill-conditioned problems and searches an approximately optimum point efficiently. On the other hand this method has two drawbacks; one is that the number of variables solved is small (i.e., less than 10), the other is its slow convergence to a strict optimum point. However, in our problem these drawbacks are not serious because the number of the control variables is limited (see iii) and a strictly optimum point is not required (see iv). Thus this method seems to be the most suitable for the problem and is adopted.

In the simplex method \((r + 1)\) initial points (i.e., extreme points

![Fig. 3. Aggregated network.](image)

![Fig. 4. Control scheme.](image)
of a simplex) are assumed in $r$ dimensional variable space. In our problem let

$$
U^i = \begin{pmatrix}
  u^0_i \\
  \vdots \\
  u^0_i + \delta u_i \\
  \vdots \\
  u^r_i 
\end{pmatrix} \quad (i = 1, 2, \ldots, r) \quad (13)
$$

where $U^0 = (u^0_1, \ldots, u^0_r)^T$ is an initial value of the control variables. The method essentially looks at the functional values at the extreme points of a simplex. The worst extreme point is replaced by a new point along the line joining this point and the centroid of the remaining points. The new point is generated by so-called reflection, extention, or contraction operations. For example the reflection is as follows:

$$
U^R = 2U^C - U^H \quad (14)
$$

where $U^R$, $U^C$, and $U^H$ are the reflected, the centroid, and the worst points, respectively. When with these operations the functional value is not improved, the simplex size is reduced around the extreme point with the minimum functional value. The process is repeated until a suitable termination criterion is satisfied.

A more detailed description of the method is given in [9], [15].

In the method, the objective function must be evaluated repeatedly for given $U$ when each operation is carried out. Although it requires fewer evaluations than other methods, the solving speed for pressure/flow becomes critical.

**High-Speed Solving of Pressure/Flow**

Several techniques for solving pressure/flow in distribution networks are well known (see [1]–[4]). However, they are not satisfactory for on-line usage. Collins and others [6] have developed a high-speed method using optimization techniques. We have developed a new method based on network flow theory. The algebraic equations (1) and (2) for a given $U$ are transformed into a minimum cost flow problem [10], and efficient network algorithms are adopted.

Here we introduce the following function:

$$
E = \sum_{j \in B} \int_0^{x_j} f_j(x, u_j) \, dx - \sum_{i \in N} w_i p_i \quad (15)
$$

where the first term is loss energy in the network and the second term is inflow energy into the network. It is known that minimizing (15) under constraint (1) leads to (2) (cf. [10]). This fact is easily derived from the corresponding Lagrange conditions. Using this idea, (1) and (2) are transformed into the following minimum cost flow problem:

$$
\sum_{j \in B \cup B_{in}} \int_0^{x_j} c_j(x) \, dx \rightarrow \min \quad (16)
$$

s.t.

$$
\begin{align*}
\sum_{j \in \delta (i)} x_j - \sum_{j \in B (i)} x_j = & \begin{cases} 
- \sum_{j \in A_j} \alpha_j & (i = n_{in}) \\
0 & (i \in N \cup N_{in}) \\
\sum_{j \in B} \alpha_j & (i = n_{out})
\end{cases} \\
0 \leq x_j & (j \in B_{out})
\end{align*} \quad (17)$$

$$
\Delta x_j = \min \left\{ \delta_1, \alpha_j \right\} \quad (18)
$$

where $n_{in}$ and $n_{out}$ are fictional source and sink nodes, respectively. $B_{in}$ and $B_{out}$ are the sets of fictional arcs which connect the source node with the inflow nodes and the consumption nodes with the sink node. The cost function per unit flow $c_j(x_j)$ must be set, according to function (15), as follows:

i) for arcs concerned with actual pipes:

$$
c_j(x_j) = f_j(x_j, u_j) \quad (j \in B) \quad (19)
$$

ii) for arcs connecting the source node to the inflow nodes:

$$
c_j(x_j) = -p_j r_j(x_j) \quad (j \in B_{in}) \quad (20)
$$

iii) for arcs connecting consumption nodes to the sink node:

$$
c_j(x_j) = 0 \quad (j \in B_{out}). \quad (21)
$$

The capacity is set for the arc $j$ included in $B_{out}$:

$$
a_j(x_j) = y_j r_j(x_j) \quad (j \in B_{out}). \quad (22)
$$

When solving the minimum cost flow problem, node potential is calculated simultaneously. The potential is equal to pressure, except for an additional constant.

Some algorithms (e.g., the primal–dual method, the primal method, the dual method, etc.) are known which solve a minimum cost flow problem (cf. [10]–[13]). Among these algorithms the primal–dual method is the most suitable for the first stage of the solution process, since no feasible flow is given. The primal method is superior when a feasible flow is given and modification of the flow for the changed control variables is sought.

The primal–dual method algorithm is characterized by simultaneously solving the shortest path problem and the maximum flow problem along the path. The former takes a large portion of the computation time required. However, it is solved efficiently using the Dijkstra method (cf. [12]). The primal method algorithm is characterized by its searching for negative cycles (i.e., sum of the arc costs along the cycle is negative) and modifying the flow along them. The procedure is carried out using the Bennington's tree algorithm [13]. These methods have been improved to be more efficient by recent developments [16]–[18]. However, their program codes seem too large and complicated for implementation on a minicomputer system and we exploit the above-mentioned rather conventional techniques. It is noted that if the recent techniques could be adopted in the proposed scheme, the overall computation time to obtain the operating points would be reduced further.

When applying either method, the nonlinearity of the cost functions must be overcome. With the primal–dual method, piecewise linear approximation of $\int_0^{x_j} c_j(x) \, dx$ [i.e., stepwise approximation of $c_j(x_j)$] is adopted. The segment length for the linearization is determined by the following rule in order to get a good approximation (Fig. 5). When the flow value is small, the flow axis is divided uniformly by the interval $\delta_k$. If the flow value becomes larger, the cost axis is divided by $\delta_k$ in the same way. Let $d_k^x$ be the $k$th dividing point on the flow axis on the arc $j$, then the $(k + 1)$th dividing point is determined as follows:

$$
d_k^{x+1} = d_k^x + \min \left\{ \delta_1, \Delta x_j \right\} \quad (23)
$$

where $\Delta x_j$ is the solution of

$$
c_j \left( d_k^x + \Delta x_j \right) - c_j \left( d_k^{x+1} \right) = \delta_2. \quad (24)
$$

The cost function in the interval $[d_k^x, d_k^{x+1})$ is approximated by the average of $c_j(d_k^x)$ and $c_j(d_k^{x+1})$. Both $\delta_1$ and $\delta_2$ are given constants depending on the required accuracy. This approximation is carried out during the primal–dual iteration procedure. Only five data values are memorized for an arc, i.e., three dividing points and two approximated cost values between them,
these are updated according to the flow modification by the primal–dual procedure. The advantages of this approximation scheme are twofold: one is to reduce calculation amount for the approximation, the other is to minimize memory area.

The primal method deals with the nonlinearity more directly. A negative cycle is identified using a spanning tree; a cycle is obtained by adding a cutree arc to the tree and its negativity is judged. If it is negative the flow along it is modified so as to eliminate the negativity. The Newton–Raphson method is adopted to solve the negative cycle equation, that is,

$$\sum_{j \in L^+} c_j(x_j + \delta x) - \sum_{j \in L^-} c_j(x_j - \delta x) = 0$$

where $L^+$, $L^-$ are the sets of arcs which are included in the negative cycle in the same and opposite direction, respectively. $\delta x$ is the modifying value of the flow along the cycle. This procedure is iterated until all the negative cycles disappear. Differing from the Bennington’s algorithm [13], the above procedure does not need to modify the tree structure because the arcs included in $B \cup B_{in}$ do not have finite capacities. This fact makes the algorithm simple and saves computation time.

The whole optimization procedure is shown by the problem analysis diagram (PAD) [19] in Fig. 6. At the first stage of generating the initial simplex, the primal–dual method is used for finding feasible pressure/flow. After this, the primal method is adopted which only modifies the previous pressure/flow. This combined usage of both the primal–dual and the primal methods yields enormous saving of computation time. The termination condition in searching the optimal point is as follows:

$$I(U^0, \ldots, U^r) = \sum_{i=0}^{r} (J(U^i) - J(U^0))^2 < \epsilon$$

where $U^i$ ($i=0, \ldots, r$) are extreme points of the improved simplex and $U^0$ is the best point among them. In the lower part of Fig. 6 the sensitivity matrix around the optimal point is calculated and will be used in the feedback control described in Section IV.

**Numerical Experiments**

Numerical experiments were carried out for both the original network (Fig. 1) and the reduced one (Fig. 3). The code is written in Fortran and was run on HITAC M-180 (3 MIPS). Ten cases of water consumption, from the minimum to the maximum during the day, are assumed and simulated. In the objective function (12), $\gamma_i$ and $\delta_{ji}$ are fixed to 1 and 0, respectively. The desired pressure $p^*_i$ is set to the sum of 30 mAq (adequate service pressure) and the ground level at node $i$.

The average of the optimized pressure above the ground level is shown according to the ten consumption levels in Fig. 7. Another pressure curve shown illustrates the case where the control variables are not optimized; they are assumed fixed to conventional values. The average computation time is 3 min for the original network and 30 s for the aggregated one. A conventional method using steepest descent and Marlow’s method [14] has taken more than 1 h for the original network. Clearly, great progress was made in reducing computation time.

**D. Feedback Control**

Control variables given by the nonlinear optimization are modified based on the measured variables by the feedback control.

**Linearized Model:** System equations (1) and (2) are linearized for small perturbations around the operating point. Particular consideration is given to the control variables and the observed pressure/flow:

$$\Delta Z = Y \Delta U + N$$

where $\Delta Z$, $\Delta U$ are small perturbations of the observed pressure/flow and the control variables, respectively. $Y$ is a sensitivity matrix at the operating point, which has been calculated when the operating point is determined (see Fig. 6). $N$ is the influence term derived from the estimate error of the consumption and is obviously unknown. We assume that the consumption may not vary in the short term (e.g., a few minutes). The dynamics of $N$ are written as:

$$N_{t-1} = N_t$$

where $t$ means time step.

**Control Algorithm:** A performance index is introduced to regulate pressure/flow to their optimal points:

$$J^* = \Delta Z^T P \Delta Z + (\Delta U_i - \Delta U_{i-1})^T Q (\Delta U_i - \Delta U_{i-1})$$

where $P$ and $Q$ are diagonal weighting matrices. A simple control algorithm is derived from the minimization conditions of $J^*$ under constraints (27) and (28):

$$\Delta U_i = G \Delta Z_i + \Delta U_{i-1}$$

where

$$G = - (Y^T P Y + Q)^{-1} Y^T P$$

This computation includes the matrix inverse. However, we have no difficulty for on-line usage because the dimensions are limited to the number of control variables.

In implementation, values of the weighting matrices $P$ and $Q$ must be determined. In our study, we adopt the following procedure: First each element of $P$ is determined so as to reflect the consumption volume over the area represented by the corresponding observation point. Then $Q$ is found based on numerical experiments so as to extend the stability range. It has been verified that the feedback control with adequately selected $P$ and $Q$ stabilizes the system when the error of estimated consumption is less than 20 percent. This implies that a new operating point
Generation of the initial simplex

Set $u^0$: initial value of $u$

Obtain $(P(u^0), X(u^0))$ by Primal-Dual Method and calculate $J(u^0)$

Generate $(u^1, \ldots, u^r)$ by Eq. (13)

Obtain $(P(u^i), X(u^i))$ by Primal Method and calculate $J(u^i)$

Search of the optimal $u^*$ by Simplex Method

Until $I(u^i, \ldots, u^r) < \varepsilon$

Generate a new $u$ by the operations of Simplex Method

Obtain $(P(u), X(u))$ by Primal Method and calculate $J(u)$

if $I(u, \ldots, u^r) < \varepsilon$

else

Replace a extreme point of the simplex by $u$

Reduce the simplex size

Calculate $I(u^i, \ldots, u^r)$

Calculation of the sensitivity matrix for the Feedback Control

Perturbate the optimal $u^*(=P)$

Obtain $(P(u^*+\delta u^*), X(u^*+\delta u^*))$ by Primal Method

Calculate the sensitivity matrix $Y$

$P(u)$: pressure by control $u$

$X(u)$: flow by control $u$

$J(u)$: objective func. defined by Eq. (12)

$I(u)$: termination criterion defined by Eq. (26)

Fig. 6. PAD of optimization procedure.

![Graph showing average pressure in optimized and conventional cases.](image)

Fig. 7. Average pressure in optimized and conventional cases.
must be computed before consumption varies more than 20 percent.

IV. IMPLEMENTATION

The whole control scheme was implemented at Takamatsu, Japan. The computation for the optimal control is carried out by a process control computer HIDIC 80-E (192 KW) and takes about 5 min. The computed results are checked by human operators through the CRT before actually being used for control in the introduction phase. In the future, unmanned control will be realized. The system has been working since December 1981. Although not enough data have yet been gathered to evaluate the system, some results have been reported. In particular, pressure at night has been brought down to 40 mAq from 60 mAq above ground level (Fig. 9). As a result, the number of pipe fractures has been reduced. Water leakage may have been reduced.

We estimate the rate of reduction of the leakage based on the simulation results (Fig. 7), which are confirmed to be nearly the same as the actual control results. Water leakage can be expressed by a function of pressure above ground level:

$$\Lambda = kh^{1.15}$$

where $\Lambda$ is the leakage volume, $h$ is the pressure above ground level, and $k$ is an unknown coefficient. We assume that the leakage is $\Lambda_0$ when $h$ is 30 mAq. $\Lambda$ is written as:

$$\Lambda = \left( \frac{h}{30} \right)^{1.15} \Lambda_0.$$  \hspace{1cm} (33)

By applying (33) to the pressure curves in Fig. 7 and the consumption trajectory during the day, we get the leakage trajectory shown in Fig. 10. The average rate of reduction during the day is about 22 percent. As a result, about 4 percent of the whole distribution amount may be saved by the water distribution control.

V. CONCLUSION

An optimal control system for water distribution networks has been developed and implemented. To overcome the large scale and nonlinearity of the network, a network aggregation method and a two-level control scheme were introduced. The first level decides operating points using the simplex method, where the pressure/flow equations are solved using a high-speed technique derived from network flow theory. The second level is the feedback control around the operating points. It absorbs estimate error and small variations in consumption.

The control scheme was implemented on a minicomputer system at Takamatsu, Japan. Not enough evaluation of the system has been done yet, but some benefits of this control have already been confirmed. Water leakage and pipe fractures caused by high pressure have been reduced. The reduction rate of water leakage is estimated at 22 percent.

ACKNOWLEDGMENT

The authors would like to express their gratitude to the many participants of the Takamatsu City Water Works who supported this project. Many thanks are also due to F. Shinomiya and T. Takagi, Senior Engineers of the Ohmika Works, and S. Shimauchi, Engineer of the Systems Engineering Division, Hitachi, Ltd. for
Fig. 10. Reduction of water leakage.

their useful technical advice. They would also like to thank Dr. T. Miura, General Manager, H. Ihara, Division Manager, Dr. M. Ohnari, Chief Researcher, and T. Sekozawa, Researcher of the Systems Development Laboratory, Hitachi, Ltd. for their continuing encouragement and contributions made during the course of carrying out this research.

REFERENCES


Shin'ichiro Miyaoka was born in Tokyo, Japan, in 1950. He received the B.S. and M.S. degrees from Kyoto University, Kyoto, Japan, in 1973 and 1975, respectively, both in applied mathematics and physics.

Since 1975 he has been a Research Engineer at the Systems Development Laboratory, Hitachi, Ltd., Kawasaki, Japan. His research interests are in large-scale systems, in particular, network systems. His recent work is an application of parallel processing to large-scale system control.

Motohisa Funabashi was born in Nagoya, Japan, in 1944. He received the M.S. degree in applied mathematics and physics from Kyoto University, Kyoto, Japan, in 1969.

Since 1969, he has been with Hitachi, Ltd., Kawasaki, Japan, working on the development of control computer systems. From 1975 to 1976 he was with the Massachusetts Institute of Technology, Cambridge, and the Department of Engineering Economics Systems, Stanford University, Stanford, CA, where he studied systems planning methodology. He is presently Senior Researcher of the Systems Development Laboratory, Hitachi, Ltd. His research interest ranges from control computer applications to industrial and public systems.