DYNAMIC ECONOMIC DISPATCH OF POWER SYSTEMS BY MULTI-PASS DYNAMIC PROGRAMMING

by

Nanming Chen

Thesis submitted to the Graduate Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Electrical Engineering

APPROVED:

T. E. Bechert, Chairman

L. L. Grigsby

H. F. VanLandingham

May, 1977

Blacksburg, Virginia

ACKNOWLEDGEMENTS

I would like to thank and for their advice and interest as graduate committee members. first encouraged me to study power systems when I came to VPI and has taught me very much knowledge in this area. encouraged me to do this research and kept my life in Blacksburg interesting. guided me, gave me helpful suggestions and has spent a lot of time to teach me everything during the research and all my study here.

Gratitude is also expressed to my fiance, , for her patience and forgiveness to delay our marriage. Much appreciation is expressed to my parents for encouraging me to come to VPI from the Republic of China, the other side of the Pacific Ocean.

Finally, I have greatly appreciated the financial support provided by the Department of Electrical Engineering at VPI & SU.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES, cont.

GLOSSARY

1. a_{AB}: Conversion factor of tie line power due to different base power in area A and area B.

2. b_i: Weighting coefficient associated with the rate of change of megawatt output of generator i.

3. BTU_{ki}: Fuel (heat) consumption per hour at kth valve of generator i.

4. c_1, c_2, c_3, c_4 Weighting parameters of the cost functional.

5. C: Overall cost functional.

6. f: Terminal cost term of the cost functional.

7. Δf: Frequency deviation.

8. g: Integrand of the cost functional.

9. $h_i(x_i)$: Heat rate characteristic of generator i.

10. H: Hamiltonian of cost functional or increments of discretized time and quantized state in multi-pass dynamic programming.

11. K_p: Gain of electric power network block.

12. L: Power load.

13. LX_{oi}: Integer value of initial state.

14. LX_{Ti}: Integer value of target state.

15. $LX_{max,i}$: Integer value of $X_{max,i}$.

16. $LX_{min,i}$: Integer value of $x_{min,i}$.

17. LXV_{mi}: Integer value of P_{mi}.

18. ΔP_{ci}: Command power of generator i.

19. ΔP_D: Power demand.

20. P_i: Costate of generator i.

21. P_{mi}: The megawatt output for Mth valve point of generator i.

22. SLP_{ki}: Slope of the kth segment of the heat rate characteristic.

23. T: Time to target of the system.

24. T°: Synchronizing coefficient.

25. T_i: Time to target of generator i.

26. T_p: Time constant of electric power network.

27. T_{Ti}: Time constant of turbine steam for generating unit i.

28. U: Set of control variables or maximum rate of change of MW output.

29. u_i: Control variable of generator i.

30. $w_k(t)$: Incremental cost function for generator k.

31. $X_i=P_{Gi}$: Megawatt output of generator i.

32. X_{oi}: Initial megawatt output of generator i.

33. X_{Ti}: Target megawatt output of generator i.

34. XX_i: MW output associated with integer 0 of generator i.

35. x_{ki}: MW output associated with integer k of generator i.

36. $x_{max,i}$: Maximum MW output of generator i.

37. $x_{min,i}$: Minimum MW output of generator i.

38. y(t): MW surplus.

1. INTRODUCTION

1.1 Statement of the Problem

A typical power system of recent years consists of many control areas which are connected by tie lines to share the equipment during different peak loading and emergency conditions. Within each control area, several generators share the total load. The solution of the Economic Dispatch Problem allocates load among available generators in a way which minimizes total fuel costs. Solution of the Load Frequency Control Problem not only maintains the area frequency at the desired value but also maintains the net power interchange between control areas at the scheduled values. Since the price of fuel is getting higher and higher, economic operation of power systems becomes more and more important. The coordination of the Load Frequency Control and Economic Dispatch Problems becomes an increasingly important objective of Automatic Generation Control.[1]

Several years ago, Bechert and Kwatny formulated the area AGC problem as a dynamic optimal control problem.[2,3] Pontryagin's Maximum Principle was used to find necessary conditions for optimality. The solution not only drives the megawatt outputs to their steady state economic dispatch values in minimum time but also yields optimum schedules to reach these targets. However, this analytic method was able to yield a feedback control algorithm and optimal trajectories for a control area of only two generators. One is the time-critical generator which changes output at maximum allowable rate throughout the interval to satisfy time optimality. The other is a maneuverable generator.

For an area with more than one maneuverable generator, the analytic solution fails to work because of the great complexity of the switching surfaces. A numerical technique--Multi-Pass Dynamic Programming-- was developed[7,8] recently and extends the results to cases involving several maneuverable generators. This technique executes a new dynamic programming algorithm over the time interval many times in succession, gradually converging on the generator output schedules which minimize the dynamic costs.

A conventional dynamic programming algorithm computes the optimal cost-to-target and the optimal control for all grid points in the feasible region of the state space, and searches through them to find the optimal trajectory and the optimal control sequence. A large amount of computer time and storage must be used to solve a dynamic programming problem. The multi-pass dynamic programming algorithm, however, reduces the grid points to no more than three per generator at one time stage. Therefore, it reduces the amount of computer time and storage greatly and makes the algorithm feasible for this dynamic optimal control problem.

1.2 Scope of the Thesis

The first five passes of the multi-pass dynamic programming were previously completed in [7,8]. This thesis extends that result by searching for the convergent optimal trajectory at the desired grid fineness. After obtaining the true (locally) optimal trajectory, any zig-zag trajectories were detected and replaced by straight line trajectories in these intervals. These so-called singular trajectories are defined and discussed in Section 2.3. Computer plots are made of the optimal transient behavior of each generator. Computer examples give some applications of this method to several cases of the previously formulated problem. Necessary conditions developed before for singular solutions are checked in some examples, too.

This thesis starts with a brief description of the problem formulation and previous work. Chapter 2 introduces the state model and the cost functional. The nature of the cost functional is also discussed. Results obtained by Pontryagin's Principle are presented to yield the necessary conditions for the optimal controls. In Chapter 3 the target megawatt outputs are found first, time-to-target for each generator is then identified. Discretization of time and quantization of states are completed in preparation for multi-pass dynamic programming. The algorithm is described in Section 3.2.

In Chapter 4 the necessary conditions for singular solutions[2,9,10] are discussed. The algorithm to detect a zig-zag trajectory and to replace it by a straight line is also provided. Chapter 5 gives some examples and applications such as the effect of changing parameter values and the response to a ramp change in load. The final chapter has a comparison of the multi-pass and the conventional dynamic programming approaches. Solution times and some discussions are also given.

2. DYNAMIC OPTIMAL CONTROL FORMULATION OF POWER SYSTEM AREA AGC PROBLEM

2.1 Dynamic Optimal Control System Modeling

An optimal control problem[4] can be stated as follows: Find an admissible control $u^* \epsilon U$ which causes the system

$$\dot{x}(t) = a(x(t),\ u(t),\ t) \tag{2.1.1}$$

to follow an admissible trajectory $x^* \epsilon X$ that minimizes the cost functional

$$C = f(x(t_f),t_f) + \int_{t_o}^{t_f} g(x(t),u(t),t)dt \tag{2.1.2}$$

In order to formulate the power system AGC problem as an optimal control problem, we must find the system model.

A power system can be grouped into many control areas by the coherency of frequencies. That is to say, generators which swing together with one precise frequency are grouped as one control area and are separated from other control areas having different frequencies. The block diagram of a two-area interconnected power system LFC model is shown in Fig. 2.1. In this thesis we develop a technique for sharing the area load among several generators, in an optimal manner. In Fig. 2.1, the area A load is shared by the two generators shown. The time constants of governors and turbine steam are small compared with that of shaft inertias. Therefore, this model is simplified as shown in Fig. 2.2.

The system in Fig. 2.2 also shows two other features[6]:

(1) The system is decomposed into two subsystems: the Electric Power Control subsystem and the Area Mechanical Power Control subsystem.

(2) A supplementary area controller is added to each control area.

The network subsystem control problem was formulated and solved to find

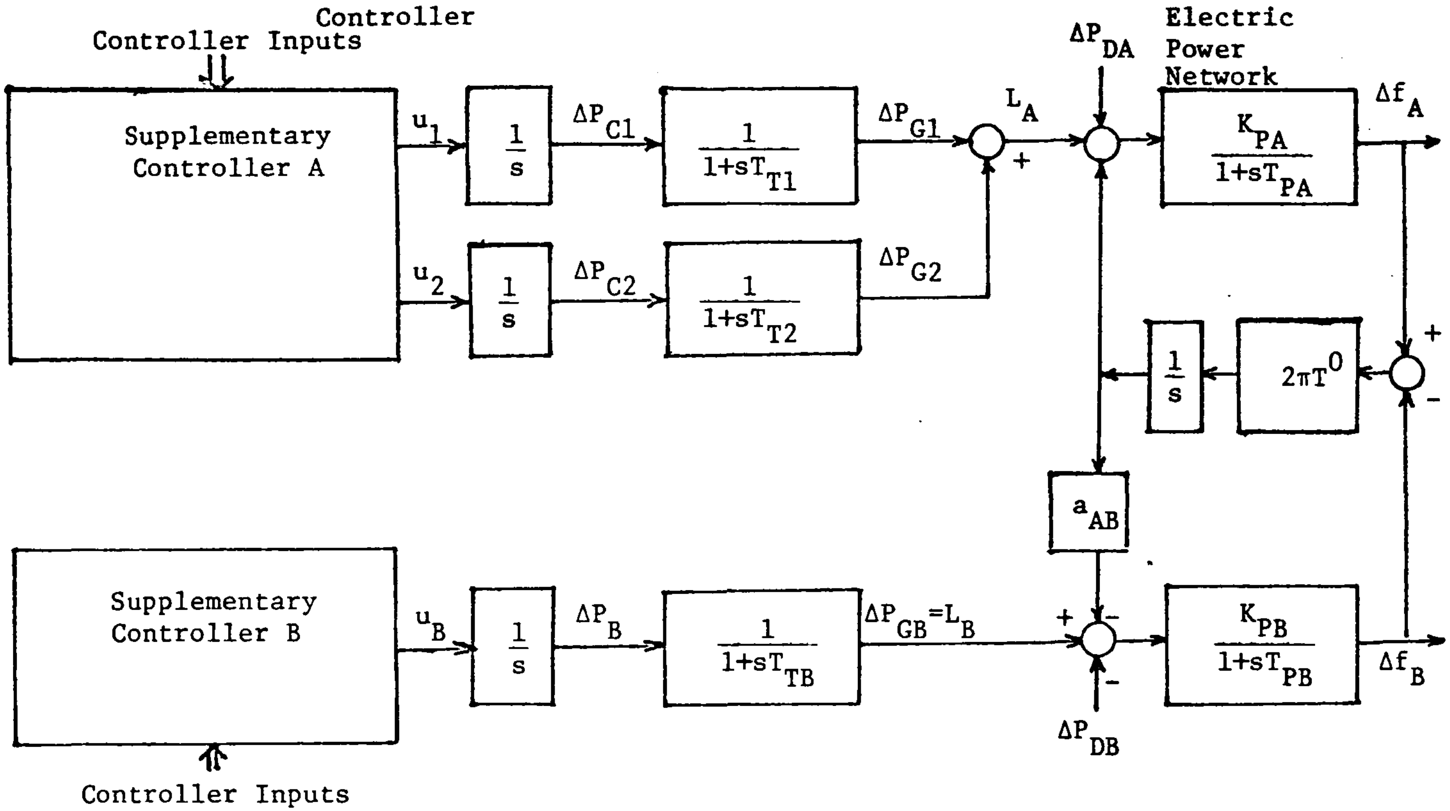


Fig. 2.1 Block Diagram of LFC for Two Area System

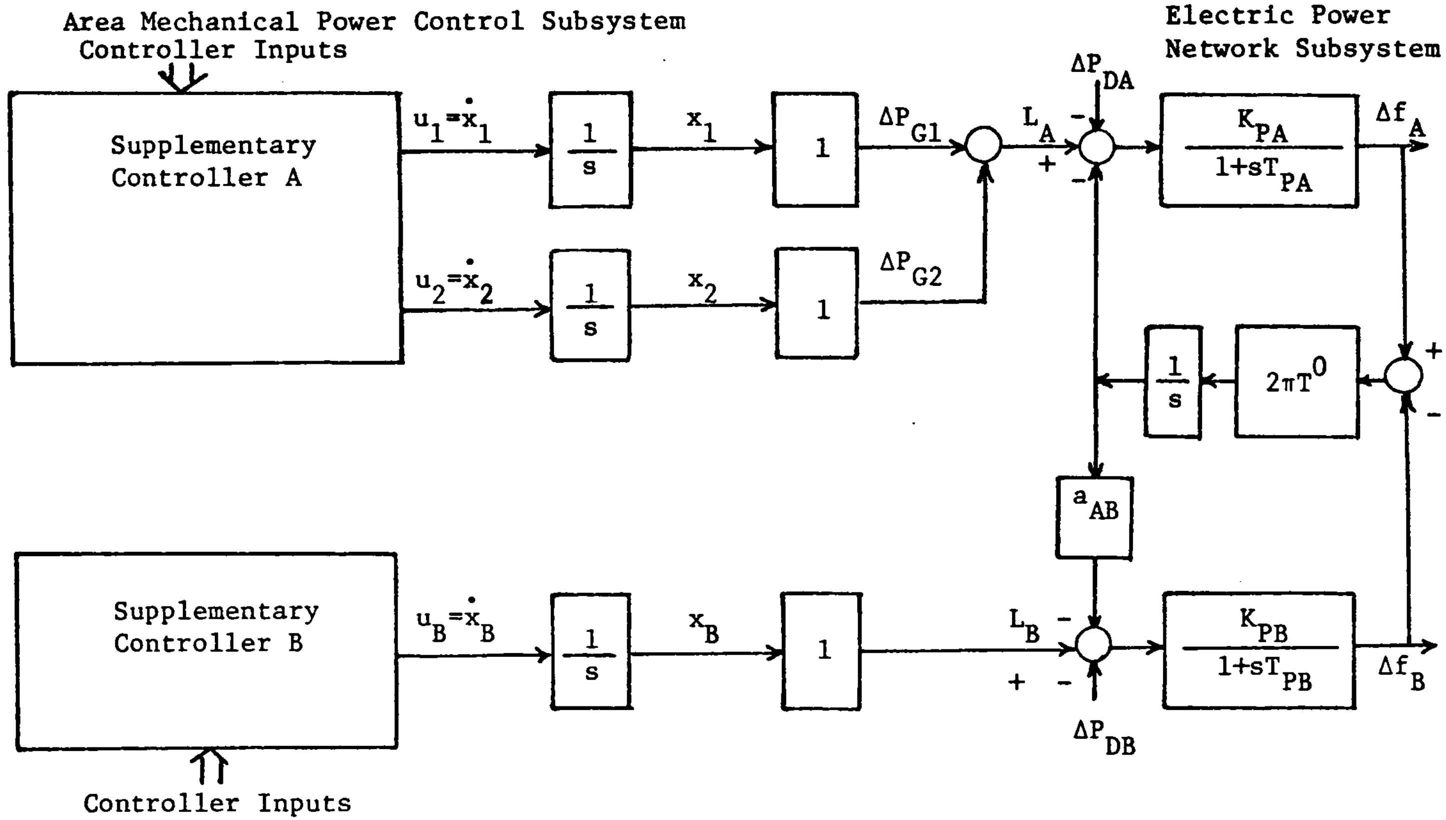


Fig. 2.2 Simplified Model Decomposed into Subsystems

the pseudocontrols L_A and L_B which minimize the frequency and tie-line deviations during the transient period[2,6]. Then the area control problem can be solved to find the supplementary area controller which drives the mechanical power output P_{GA}, P_{GR} to their target values obtained from L_A and L_B, along an optimal trajectory. This latter problem can be formulated as an optimal control problem:

Find an optimal control set

$$-U_i \le u_i^* \le U_i$$

which causes the system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} & & \\ & 0 & \\ & & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} 1 & & 0 \\ & 1 & \\ 0 & & \ddots \\ & & & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \tag{2.1.3}$$

to follow an optimal trajectory $x_{imin} \le x_i^* \le x_{imax}$ that minimizes the cost functional

$$C = \int_0^T [c_1 + c_2(\Sigma x_i - L)^2 + c_3 \Sigma b_i |u_i| + c_4 \Sigma h_i(x_i)]dt \tag{2.1.4}$$

where T = the free final time,

$x_i = P_{Gi}$ = Megawatt output of generator i,

$u_i = \dfrac{dP_{Gi}}{dt}$ = Rate of change of Mw output for generator i

L = Area Load, in megawatts,

$h_i(x_i)$ = Fuel cost characteristic of generator i as a function of x_i,

c's and b's are weighting parameters.

The cost functional will be discussed in the next section.

2.2 The Performance Measure

The cost functional has no terminal cost term. The dynamic cost terms are integrated to give the total cost. The integrand consists of four terms, each being assigned a weighting coefficient to determine the relative effect of each. The cost functional terms are interpreted as follows[6]:

1. The first term $\int_0^T c_1 dt$ penalizes the control time duration. To ensure the time optimality we can make c_1 very large compared with c_2, c_3 and c_4 so that all the generators can reach their target values in a minimum time interval. The control law for this condition, of course, is that the generator which takes the longest time to reach its target value, even if it operates at its maximum allowable rate of change of megawatt output all the way through, should increase or decrease at the maximum rate. This generator is referred to as the time-critical generator and the time duration for it to reach its target value is the control time interval of this problem. The other generators are not necessarily operated at their maximum rates of change of megawatt output but instead should be maneuvered to follow a control law to minimize the rest of the cost functional.

2. The second term $\int_0^T c_2(\Sigma x_i - L)^2 dt$ penalizes the area mechanical power mismatch. The area mechanical power demand L is considered as a constant over the time interval (0,T). When implementing the control system, the target load L will be replaced by the corresponding pseudo-control L specified by the network controller. This term will, without taking the fuel consumption into account, drive the optimal trajectory x_i^* of each generator to approach its target value as soon as possible.

However, it may cost more fuel than necessary to reach the target value in the same minimum time.

3. The third term $\int_0^T c_3 \Sigma b_i |u_i| dt$ penalizes the rate of change of megawatt output, reflecting the reduction of machine life due to increased mechanical and thermal stress. A precise formulation of costs based on the rate of change of megawatt output is not presently available; however, it is clear that the magnitudes of such rates should be penalized. This term also penalizes the unnecessary changes of megawatt generation levels. A large value of c_3 tends to "flatten" the optimal trajectory and to keep the trajectory within the megawatt range bounded by the initial and target values without excessive overshoot. This is even more important when the convergence of the multi-pass dynamic programming algorithm is concerned. A zero value of c_3 may cuase the discretized trajectories not to converge but to oscillate.

4. The fourth term $\int_0^T c_4 \Sigma h_i(x_i) dt$ is the cost based on fuel consumption. A typical heat rate characteristic $h_i(x_i)$ is shown in Fig. 2.3. The piecewise linear curve reflects the nonlinear throttling losses typical of a turbogenerator. When a valve is opened to allow more steam flow in to increase the megawatt output of turbogenerator, the heat rate and hence the fuel consumption rate increase as well.

The effect of penalizing the cost due to megawatt output itself is just opposite to that of penalizing the megawatt mismatch when generators are driven from lower initial values to higher target values. Large values of c_4 save fuel but sacrifice the megawatt accuracy. A compromise between these two effects is found to yield an optimal control law.

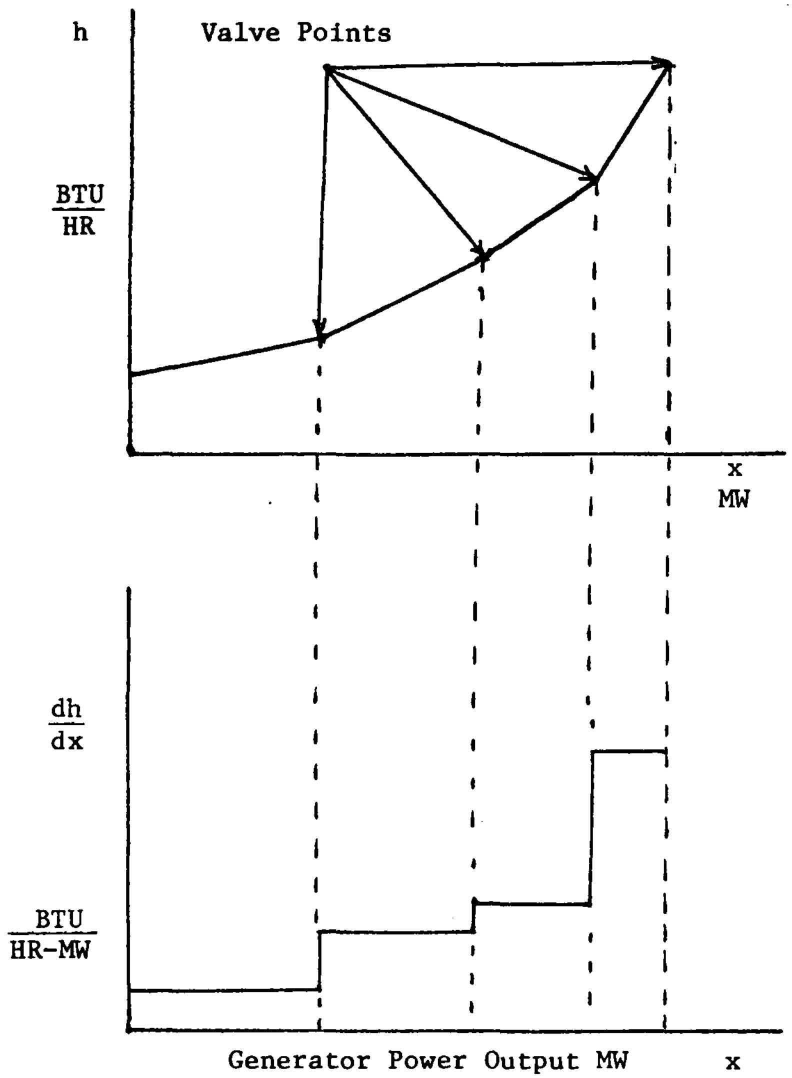


Fig. 2.3 Heat Rate and Incremental Heat Rate Characteristics for Multi-Valve Turbine

The piecewise linear heat rate characteristic yields a staircase incremental heat rate curve for the generator as shown in the lower part of Fig. 2.3. This study assumes that no two generators have precisely the same magnitude of incremental heat rate, in any valve region.

2.3 Analytical Solution by Pontryagin's Principle

The electric power network subsystem problem was solved in [2,6] and will not be discussed here. Necessary conditions for optimality of the area mechanical power control subsystem problem were obtained analytically by Pontryagin's Principle for any number of generators[2,6]. By introducing the costate vector (Lagrange Multiplier), the Hamiltonian of the area mechanical power control subsystem was formulated as:

$$H = c_1 + c_2(\Sigma x_i - L)^2 + c_3 \Sigma b_i |u_i| + c_4 \Sigma h_i(x_i) + p^T(\dot{x} - u) \quad (2.3.1)$$

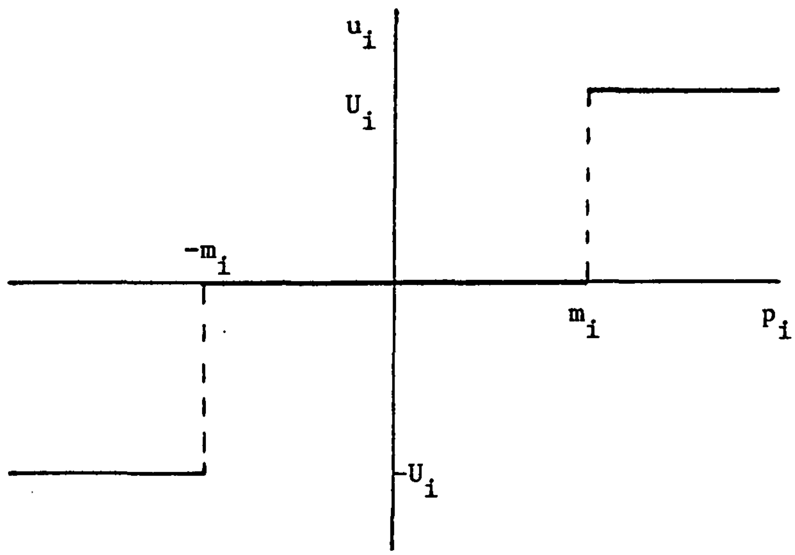
where p^T is the transpose of the costate vector.

The first four terms in H are the integrand of the cost functional, and the fifth term is the product of the costate vector and the equality constraint; that is, the state equation. Pontryagin's Principle yields from the Hamiltonian the optimal control law

$$u_i^* = \begin{cases} U_i & p_i > m_i \\ 0 & |p_i| < m_i \\ -U_i & p_i < -m_i \end{cases} \quad (2.3.2)$$

where $m_i = c_3 b_i / c_2$.

Fig. 2.4 shows the optimal control as a function of the corresponding costate. The optimal controls are well defined except for the time intervals in which the magnitude of costate p_i equals its corresponding m_i. Discontinuities occur at $p_i = m_i$ where Pontryagin's Principle fails



$$u_i^* = \begin{cases} U_i & p_i > m_i \\ 0 & |p_i| < m_i \\ -U_i & p_i < -m_i \end{cases}$$

where $m_i = c_3 b_i / c_2$

Fig. 2.4 Necessary Condition for Optimal Control to yield a unique optimal control. Other conditions are used to develop the optimal control law for these time intervals. These solutions are known as singular solutions and will be discussed in Chapter 4.

The values of costates, however, are unknown at this point and the analytical method could only yield a feedback control algorithm and optimal trajectories for a control area of two generators, the time critical generator plus one maneuverable generator only. For the case of more than one maneuverable generator, the complexity of the optimal switching logic prohibited the analytical solution. A new numerical technique called multi-pass dynamic programming was proposed[7,8] and developed[9] to solve this problem. This thesis documents the algorithm of this technique.

3. MULTI-PASS DYNAMIC PROGRAMMING

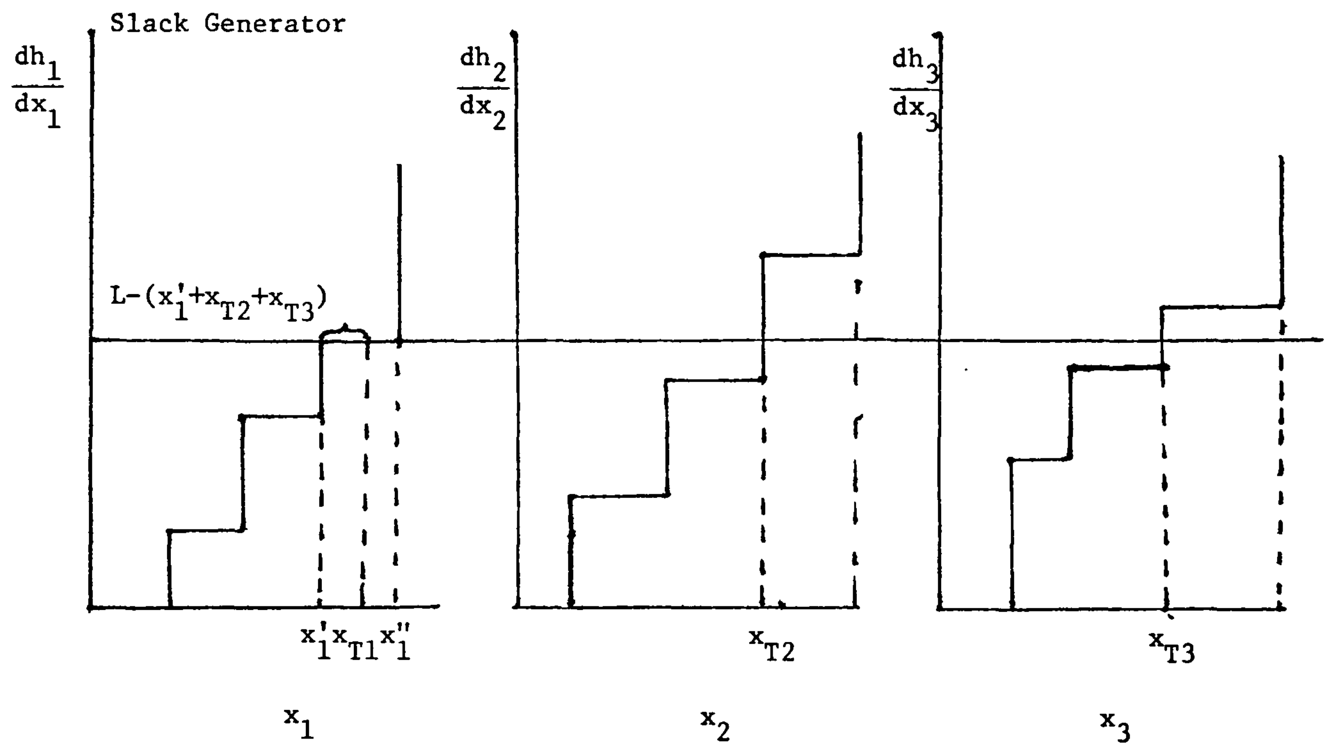
The analytical method failed to solve the area mechanical power control problem for an area with more than one maneuverable generator. The multi-pass dynamic programming algorithm is developed in this chapter to solve this problem.

The algorithm starts by finding the target value of each generator, determining the time-to-target and finding the time-critical generator. Then, the time interval is discretized into 32 stages and the megawatt output range is also quantized into 32 state units. The dynamic programming algorithm is executed for the first pass, using an increment of 16 stages and 16 state units. The optimal trajectory resulting from Pass 1, found by the principle of optimality, becomes the nominal trajectory for Pass 2. In Pass 2, the time and state grid is refined by a factor of two. That is, the increments are reduced to 8 stages and 8 state units. Then the principle of optimality is applied again. The optimal trajectory from each pass becomes the nominal trajectory for the next pass, with the grid increments refined by a factor of two, with each pass. The algorithm is carried through Pass 5, which has increments of only one time stage and one state unit and cannot be further refined. The sixth and successive passes then build allowable grid points based on the optimal trajectory of the previous pass without refining the increments, to find the final convergent optimal trajectory and optimal control law for this problem.

3.1 Targets, Time-to-Target, Time-Critical Generator, Discretization of Time and Quantization of States

The target area load L obtained from the solution of the electric power network subsystem is shared by all the generators in the area. The allocation of load to each generator is based on the economic dispatch law which says that the incremental costs of fuel consumption of all generators should be equal. Since the heat rate characteristics are assumed piecewise straight, the incremental cost characteristics are staircase curves, as shown in Fig. 2.3. Hence, all the generator megawatt outputs should be at valve points except one, which is slacked to adjust the total megawatt output to equal to the load. Fig. 3.1 shows an example of this static economic dispatch. To find the dispatch, all the stair levels for the area are arranged in ascending order. Beginning with the sum of the minimum generation of all the generators, megawatt increments are added for each level in the sequence, moving upward until the sum of the megawatt outputs of all generators finally exceeds the load. The generator whose increment caused the load to be exceeded is then denoted the slack generator. The difference between the area load and the sum of the other generator outputs becomes the target output for the slack generator. The outputs of the other generators become their target values.

Having found the megawatt target for each generator, the absolute value of the difference between the initial and target values is divided by the maximum rate of change of megawatt output for that generator, to find the minimum time for it to reach its target.



1. MW Deficiency: $x_1' + x_{T2} + x_{T3} < L$

2. MW Surplus: $x_1'' + x_{T2} + x_{T3} > L$

3. MW Balance: $x_{T1} + x_{T2} + x_{T3} = L$

Fig. 3.1 Target Load Dispatch of a 3-Generator Area

$$T_i = \frac{|x_{Ti} - x_{0i}|}{U_i} \tag{3.1.1}$$

The generator which requires the longest time to reach its target determines the time-to-target for the control problem. This generator is recognized as time-critical generator and its time-to-target is the control interval T of the problem. Its optimal control law, of course, is the maximum rate of megawatt change throughout the time interval, and its optimal trajectory is a straight line from its initial value to its target. The optimal control laws and optimal trajectories of other maneuverable generators, however, are still unknown and will be determined later by multi-pass dynamic programming. These maneuverable generators are renumbered to exclude the time-critical generator for later use.

The time interval T is divided into 32 equal increments of T/32 seconds per integer change. The control variable u_i is quantized into only three values, $-U_i$, 0, $+U_i$, in accordance with the necessary conditions for optimality. The megawatt output x_i is quantized with equal increments into 33 states which not only can be reached from the initial state but also can be steered to the target state using quantized u_i. The increments are:

$$\Delta x_i = U_i T/32 \tag{3.1.2}$$

$$i = 1, 2, \ldots n-1$$

The difference between initial and target values might not be an integer multiple of its megawatt increment.

$$\frac{|x_T - x_0|}{\Delta x} = I + \varepsilon \tag{3.1.3}$$

The target grid for the generator is the integer corresponding to the megawatt output state which is closest to the target megawatt output x_t. Any roundoff error is added to the megawatt target of the next generator. Hence, the target grid of generator 2 and the following depend on the relation:

$$\frac{|x_{Ti} + \varepsilon_{i-1} - x_{0i}|}{\Delta x_i} = I_i + \varepsilon_i \qquad (3.1.4)$$

The total system roundoff is the roundoff error of the last maneuverable generator and is smaller than the megawatt increment of this generator.

The feasible region (Fig. 3.2) is the region of states which can be reached from the initial point, and which can be driven to the target, using allowable quantized controls. Hence, the midpoint of the state interval (x_{0i}, x_{Ti}) is assigned the integer 16 so that the feasible region spreads evenly to both sides of it. The integers corresponding to x_{0i} and x_{Ti} are:

$$\begin{aligned} LX_{0i} &= 16 \pm \frac{I_i}{2} \\ LX_{Ti} &= 16 \mp \frac{I_i}{2} \end{aligned} \qquad (3.1.5)$$

depending on whether x_{Ti} is less or greater than x_{0i}. The megawatt output associated with integer 0 is

$$xx_i = x_{0i} - LX_{0i} \cdot \Delta x_i \qquad (3.1.6)$$

The megawatt output associated with integer k for generator i is

$$x_{ki} = xx_i + k\Delta x_i \qquad (3.1.7)$$

The feasible region may be restricted by the megawatt limits. Integers are also assigned to maximum and minimum limits of megawatt outputs with tolerances of 5% of Δx_i.

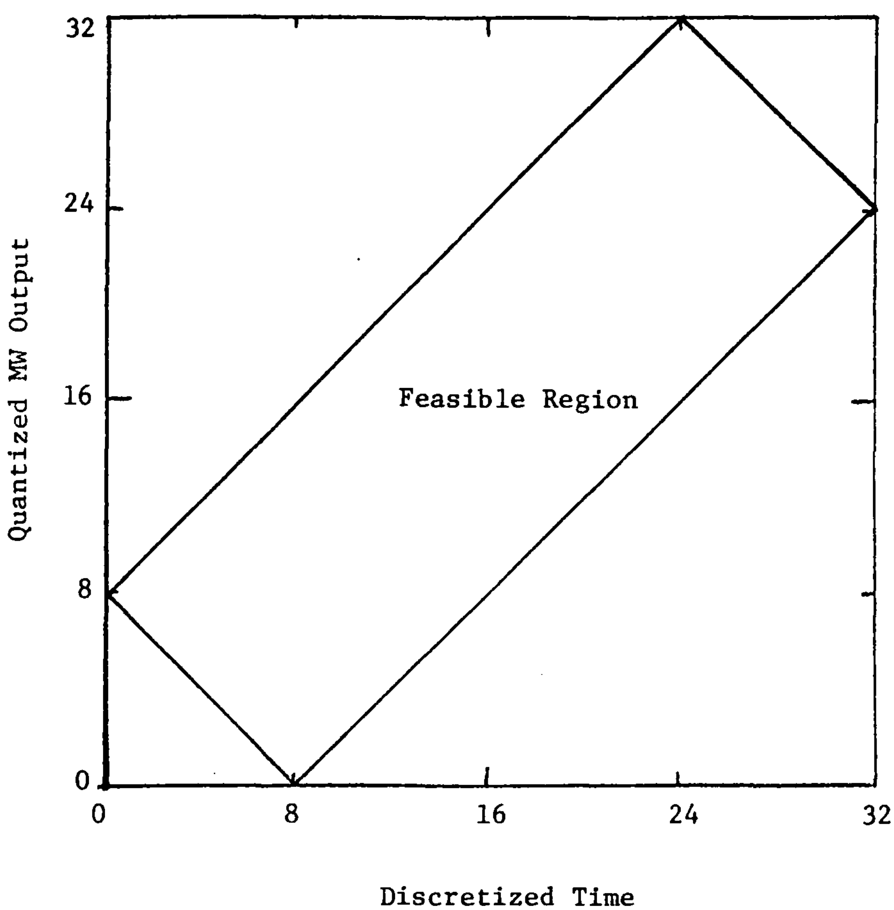


Fig. 3.2 Feasible Region

$$LX_{max,i} = \text{integer part} \left[\frac{(x_{max,i} - xx_i)}{\Delta x_i} \right] + 0.05$$
$$LX_{min,i} = \text{integer part} \left[\frac{(x_{min,i} - xx_i)}{\Delta x_i} \right] + 0.95 \qquad (3.1.8)$$

These tolerances are shown in Fig. 3.3.

The mth valve point of generator i, P_{mi}, is also expressed as an integer:

$$LXV_{mi} = \text{integer part} \left[\frac{(P_{mi} - xx_i)}{\Delta x_i} \right] + 0.5 \qquad (3.1.9)$$

The fourth term of the cost functional associated with each integer and the third term of the cost functional associated with the three values of control variable u_i of each maneuverable generator need to be calculated only once:

$$COST4_{ji} = BTU_{ki} + (x_{ji} - P_{ki})SLP_{ki} \qquad (3.1.10)$$

where k is the valve point where the next valve point k + 1 is the first

valve point to pass the megawatt output associated with integer j,

SLP_{ki} is the slope of the kth segment of the heat rate characteristic,

c_4 is the coefficient of this term.

The values are shown in Fig. 3.4. For the integration of the cost functional, we use the average value between two integer values multiplying on integer time increment.

$$COST_i = b_i U_i c_3 \qquad (3.1.11)$$

where b_i is the coefficient associated with generator i,

c_3 is the coefficient of this term.

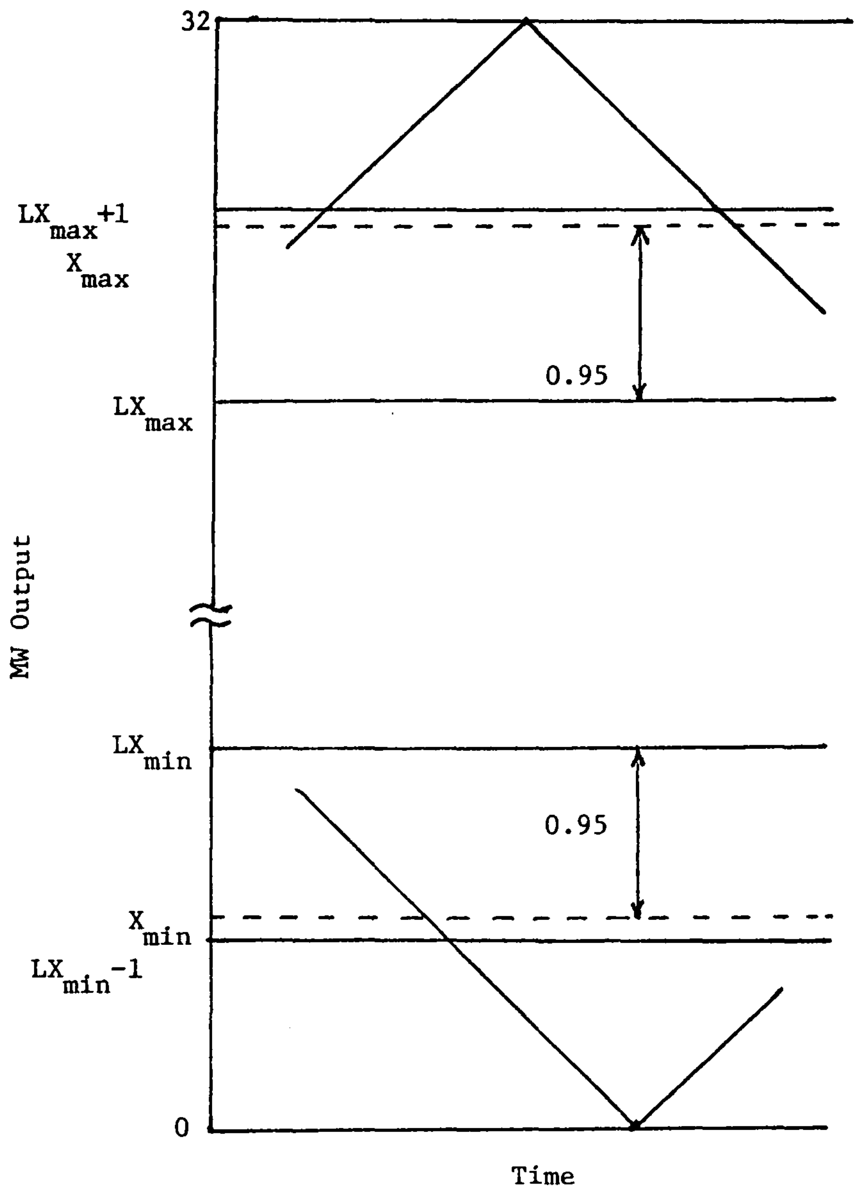


Fig. 3.3 Tolerance of LX_{max} and LX_{min}

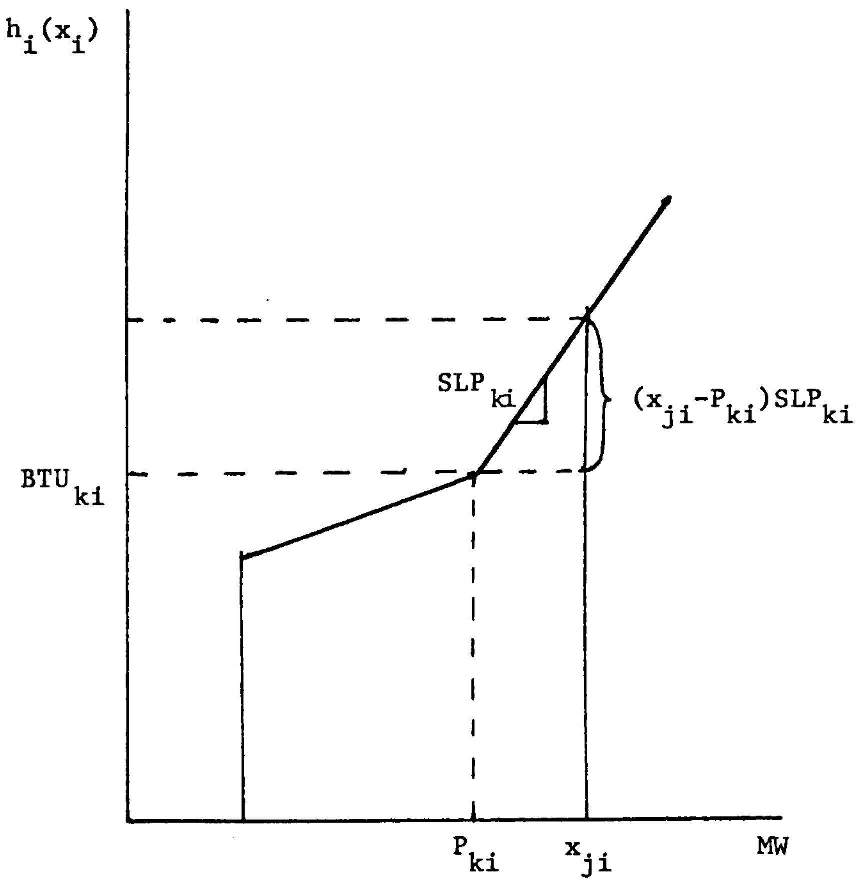


Fig. 3.4 Value of $COST4_{ji}$

The upper bound $LXHI_{ki}$ and lower bound $LXLO_{ki}$ of the feasible region are stored for later plotting.

3.2 Multi-Pass Dynamic Programming Algorithm[7,8,9]

Having assigned integers to all quantized states which can be reached by applying a sequence of the three quantized control values to the initial state, we can start the dynamic programming by using the principle of optimality at each grid point at each stage. This process is carried out stage by stage starting from the target state and going backward to find the optimal control sequence and optimal trajectory.

The first pass examines the grid points with increments of sixteen integer state units and sixteen integer time stages (H = 16). The second pass reduces the increments to eight integer state units and time stages (H = 8). The following passes keep on reducing the increments by a factor of two until Pass 5, in which the increments equal one integer state unit and one time stage (H = 1).

Using the optimal trajectory of the previous pass as the nominal trajectory, the admissible grids are chosen as H state units above and below the nominal trajectory at each stage unless the boundary of the feasible region is exceeded. Since the time increment is refined by a factor of two for each pass before the fifth, the nominal trajectory must have interpolations between two stages of the optimal trajectory of the previous pass at these interpolated stages. The interpolation state is the average of the states at the two stages adjacent to it. Since the increments are multiples of two, these interpolations will stay at integer states unless it reaches the initial state. The interpolation at the first stage depends on the state after the first stage of the optimal trajectory of the previous pass. If it is greater than the initial state, the interpolation is one new increment state below it. If it is smaller, the interpolation is one new increment state above it. Grid points outside the boundary of the feasible region are excluded.

With the grid having been set up, we can compute the minimum cost to target of each grid point from the last stage moving backward to the initial state and store the corresponding control of each grid. The c_1 term of the cost functional is the same for any trajectory from initial to target state and is not considered when comparing the minimum cost to target. Therefore, the cost calculated in the program consists of only the megawatt mismatch term, the rate of change of megawatt term, and the fuel cost term. These terms are calculated and summed up to yield the total cost to target for each grid. The minimum cost-to-target and the corresponding optimal control for each grid point at one stage are stored. This process proceeds from target state back to the initial state. The optimal control sequence is the sequence of controls that yields the minimum cost to target from the initial state. In order to save computer storage, the optimal controls are stored in a six-dimensional array (stage number and five maneuverable generators) as an encoded number which later can be decoded to yield the optimal control value. The control code of each generator will be one digit of a five-digit number.

After finding the optimal trajectory, we transform the integers back to megawatt values and print out both integer and megawatt values.

The fifth pass may have part of its optimal trajectory lying on the edge of the admissible states. This would indicate the possibility that the true optimal trajectory could exist somewhere outside that edge[9]. Therefore, additional passes are provided, without further refinement of the grid, to adjust the region of admissible states, to surround the optimal trajectory of the previous pass. Soon (Fig. 3.11) the optimal trajectory for each generator is seen to remain inside the region of admissible states.

The trajectory has been given every opportunity to change to a different path but has chosen to stay fixed. When this condition is reached by all maneuverable generators, the solution then lies in a "cost valley" and has converged to the true optimal solution within the coarseness of the grid determined by the increment of one integer state and time ($H = 1$)[9].

The procedure is shown in Figs. 3.5-3.11 and the flow chart of this program is shown in the Appendix.

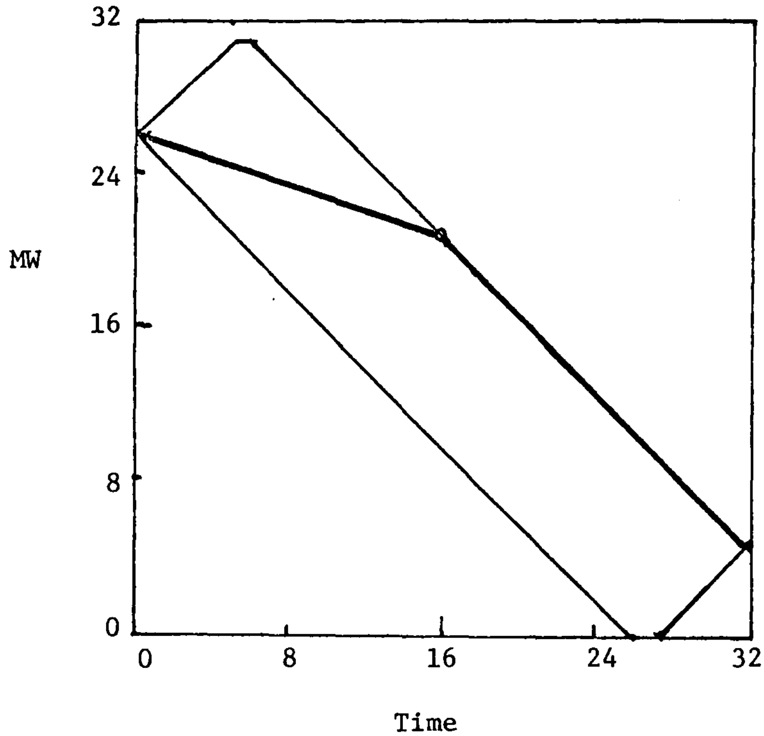


Fig. 3.5 Optimal Trajectory of a Generator: Pass 1

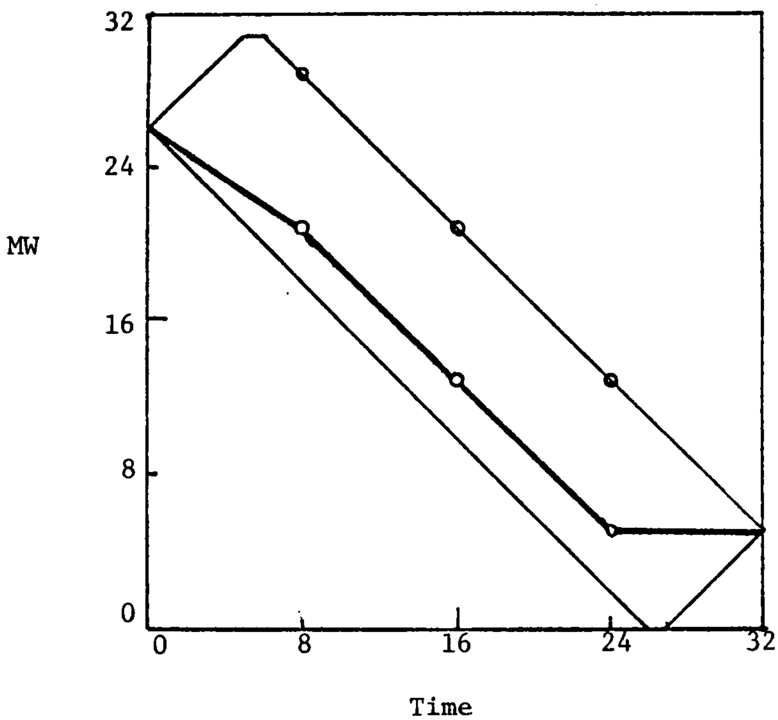


Fig. 3.6 Optimal Trajectory of a Generator: Pass 2

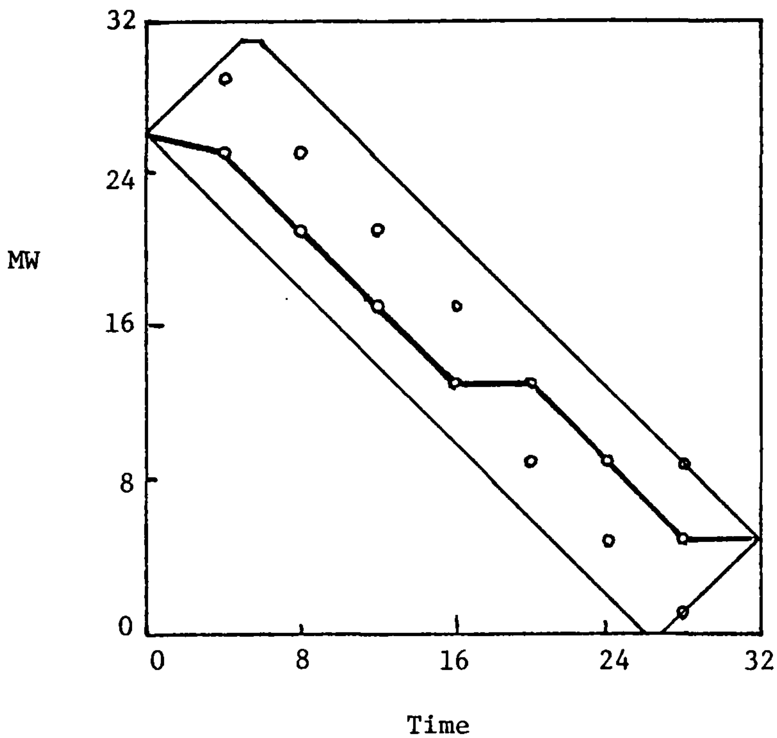


Fig. 3.7 Optimal Trajectory of a Generator: Pass 3

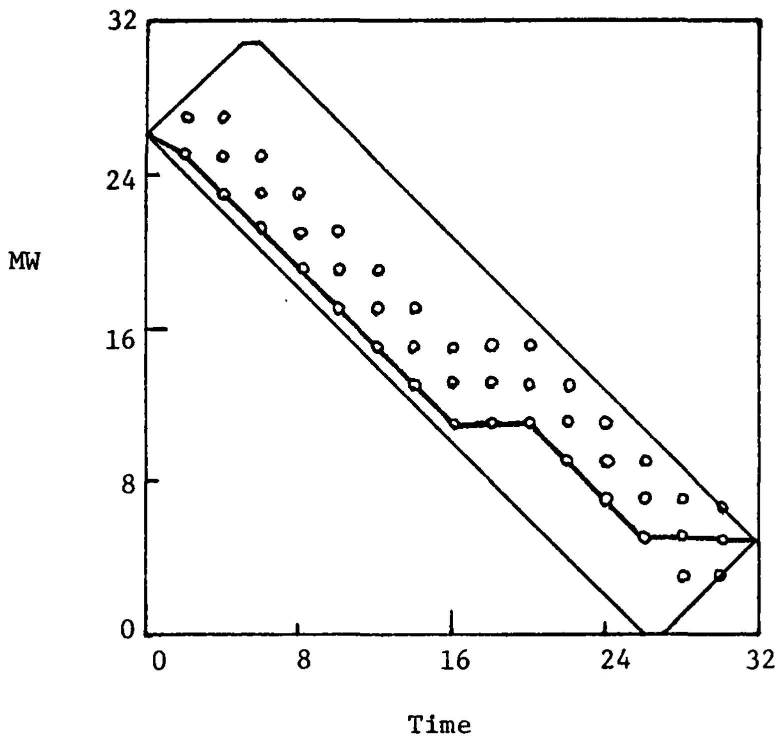


Fig. 3.8 Optimal Trajectory of a Generator: Pass 4

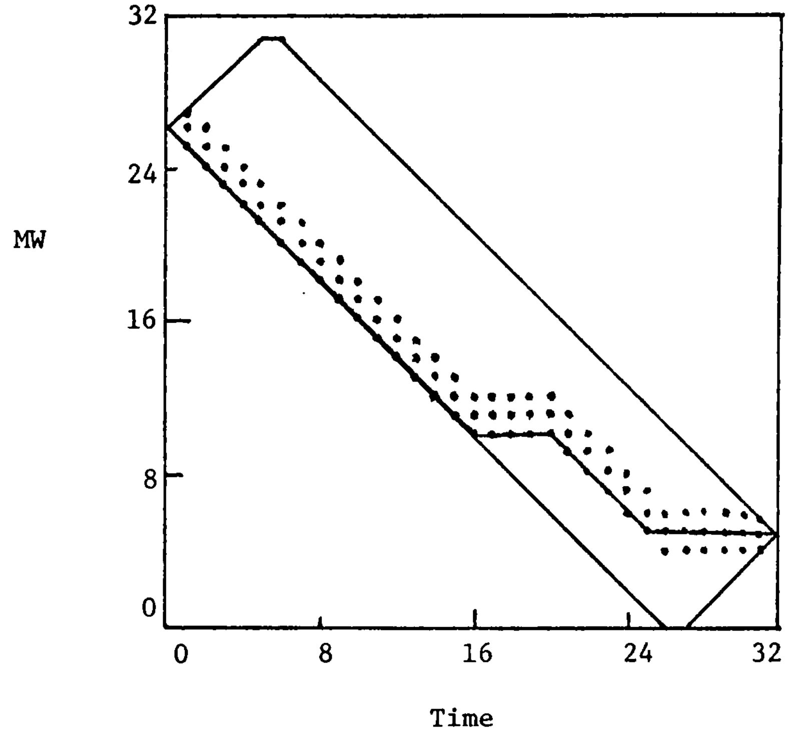


Fig. 3.9 Optimal Trajectory of a Generator: Pass 5

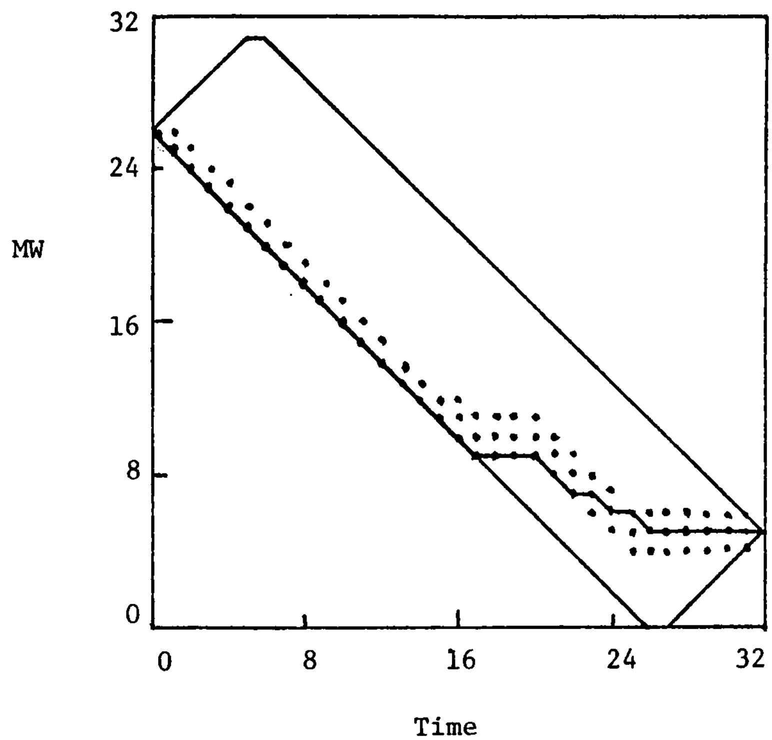


Fig. 3.10 Optimal Trajectory of a Generator: Pass 6

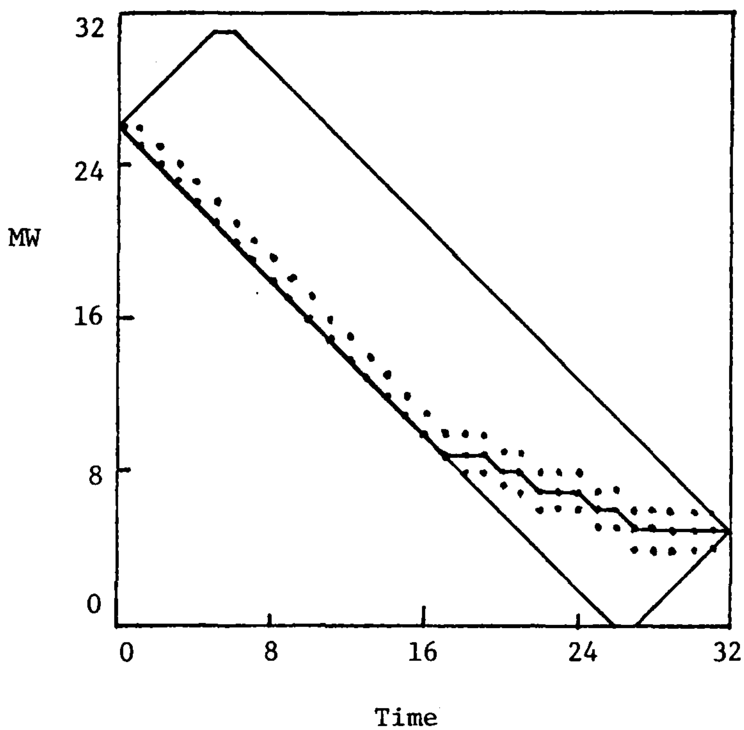


Fig. 3.11 Optimal Trajectory of a Generator: Pass 8

4. SINGULAR SOLUTIONS

The slope of the optimal trajectory obtained by multi-pass dynamic programming (Fig. 4.1)[10] represents the optimal control. It may have three values: the maximum increasing value, zero, and the maximum decreasing value as indicated in Section 2.3. Zig-zag trajectories appear in some intervals of the optimal responses in Fig. 4.1, which means that we must repeatedly open and close the valve throughout the interval to change the megawatt output level. This is not reasonable. We prefer to see the valve open to an intermediate position and stay there. Therefore, these zig-zag intervals appear to be singular solutions, which may have intermediate optimal control values between zero and its maximum value. These zig-zags should be replaced by straight lines. The necessary conditions for singular solutions[2,6,10] are discussed in detail in Section 4.1. Section 4.2 describes the computer program steps which find the singular intervals and replace the zig-zag by the singular trajectories. The flow chart of this routine is in the Appendix.

The program list of the overall program includes plot subroutines; however, they are self-explanatory and no flow chart is given for them. These time-plots are very helpful for observing the results of this program.

4.1 Necessary Conditions for Singular Solutions[2,6,10]

As mentioned in Section 2.3, if the costate $p_k = c_3 b_k / c_2$ for some non-zero time interval (t_1, t_2), then generator k is a singular generator in this interval. Since p_k is constant, $\dot{p}_k$ must equal 0. But a

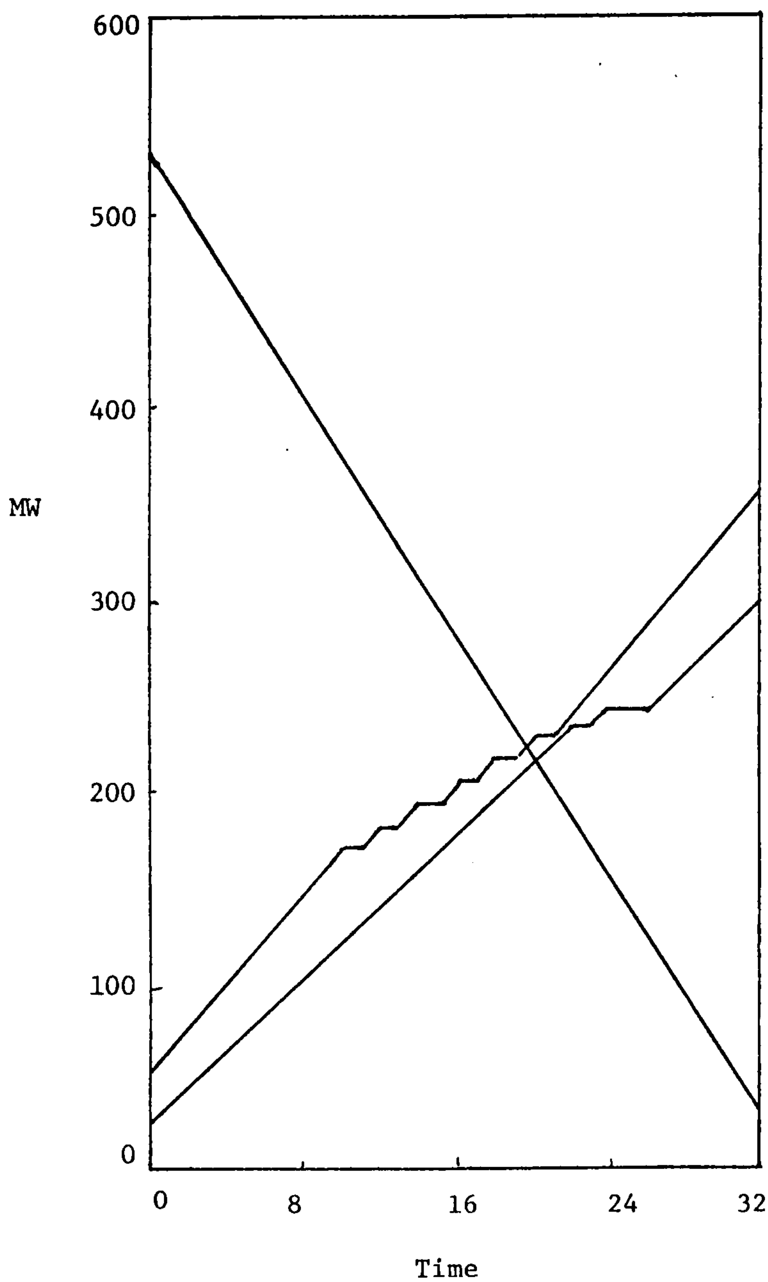


Fig. 4.1 Optimal Trajectories by Multi-Pass Dynamic Programming

necessary condition for optimality is[4]:

$$\dot{p}(t) = -\frac{\partial H}{\partial x}(x,u,p,t) \tag{4.1.1}$$

where H is the Hamiltonian,

x is the state vector,

p is the costate vector.

Therefore,

$$-\frac{\partial H}{\partial x} = 0 \tag{4.1.2}$$

Taking the derivative of H with respect to x, we have:

$$-\frac{\partial H}{\partial x} = 2y(t) + 2w_k(t) = 0 \tag{4.1.3}$$

where
$$y(t) = \Sigma x_j(t) - L = \text{MW surplus} \tag{4.1.4}$$

$$w_k(t) = \tfrac{1}{2}c_4 \frac{dh_k(x_k)}{dx_k} \tag{4.1.5}$$

which is proportional to the incremental cost function for generator k.

Therefore,

$$-y(t) = w_k(t) \tag{4.1.6}$$

Since the incremental cost $\frac{dh_k(x_k)}{dx_k}$ is a staircase function (Section 2.2, Fig. 2.3), the megawatt deficiency $L-\Sigma x_j(t)$ must be piecewise constant during the singular interval. Differentiate (4.1.6),

$$-\dot{y}(t) = \dot{w}_k(t) \tag{4.1.7}$$

From (4.1.4), we have

$$\dot{y} = \frac{d}{dt}(\Sigma x_j - L) = \Sigma \dot{x}_j(t).$$

Substitute into state equation (2.1.3), we have

$$\dot{y} = \Sigma u_j(t) \tag{4.1.8}$$

But, since $\frac{dh_k(x_k)}{dx_k}$ is a staircase function, if x_k is not at valve point during the interval,

$$\dot{w}_k(t) = \frac{d}{dt}(\tfrac{1}{2}c_4 \frac{dh_k(x_k)}{dx_k}) = 0.$$

Therefore,

$$\Sigma u_j(t) = 0.$$

Thus

$$u_k(t) = -\sum_{\substack{j=1 \\ j\neq k}}^{n} u_j(t) \tag{4.1.9}$$

$u_j(t)$ are nonsingular controls, their values are known to be either at zero or their maximum values; hence, $u_k(t)$ can be found.

Since we assume that no two generators have identical "flat" levels in their incremental heat rate curves (Section 2.2), there is only one generator which can have a singular solution $x_k(t)$ not at a valve point. Thus, only one new singular control value shows up. If $x_k(t)$ remains constant at valve point P_{mk}, the optimal control is

$$u_k(t) = \dot{x}_k(t) = 0$$

and no new control values is created.

The conditions are summarized as follows:

(1) The megawatt deficiency must be constant.

(2) Only one singular generator at a time.

(3) The singular control equals the negative sum of the nonsingular controls.

These conditions are seen to be satisfied in the computer examples (Fig. 4.2)[10].

4.2 Finding Singular Solutions

Zig-zag trajectories were found to be the singular solutions. They should be replaced by a straight line. Several questions arise. Where should this straight line start? Should it start at the first bending point or one increment before the bending point? Where should it end?

Of course, according to the necessary conditions, the straight line must have the slope which is the negative sum of the slopes of the other trajectories. However, since time is discretized into integer stages for dynamic programming calculations, the optimal control cannot be exactly the same as the theoretical value. There are four candidates that have very close values of slopes. The one yielding minimum cost will be our preference.

The four candidates are the four lines connecting two starting grid points and two ending grid points, as shown in Fig. 4.3. The starting points are the first bending point (LOEND) of the zig-zag interval and the point with one zig-zag segment before it (LOEND1). The ending grids are the last bending point (HIEND1) and the point with one zig-zag after it (HIEND).

The zig-zag segments may be irregular. Every segment of no more than three stage length will be taken into account as a possible zig-zag segment. A zig-zag solution consists of at least two successive zig-zag segments. All the zig-zag trajectories should be either increasing or decreasing throughout the singular interval. Intervals with both in-

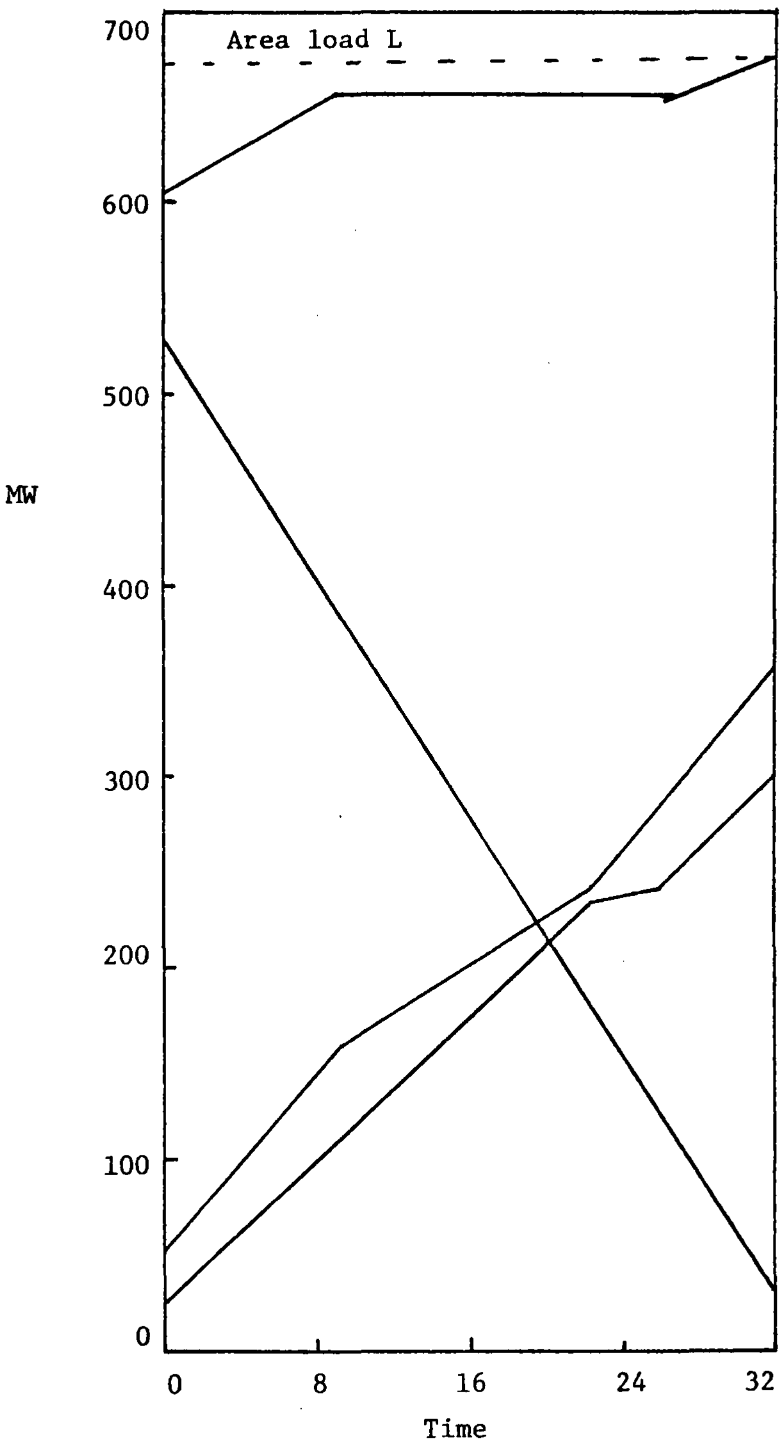


Fig. 4.2 Example of Singular Solution

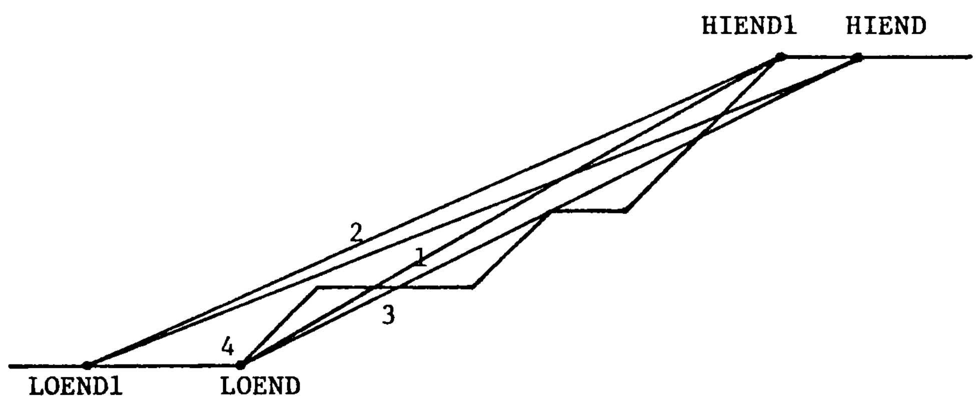


Fig. 4.3 Candidates for Singular Solution

creasing and decreasing trajectories are recognized as nonzig-zag intervals, or they may separate into two zig-zag intervals (Fig. 4.4).

Fig. 4.5 demonstrates the definitions of the zig-zag trajectory. LOEND1 is defined as a segment before LOEND which has the length of the second zig-zag segment. Similarly, HIEND is defined as a segment after HIEND1 which has the length of the second last zig-zag. These are labeled INTVL2 and INTVL1, respectively, in Fig. 4.5.

The four candidate solution trajectories are the straight lines connecting:

(1) LOEND to HIEND1;

(2) LOEND1 to HIEND1;

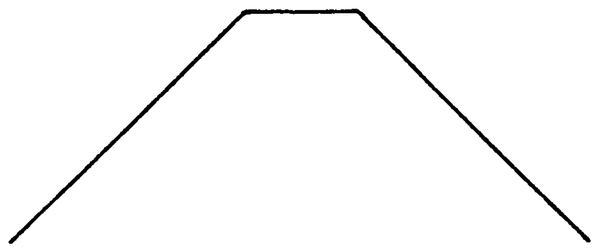
(3) LOEND to HIEND;

(4) LOEND1 to HIEND (Fig. 4.3).

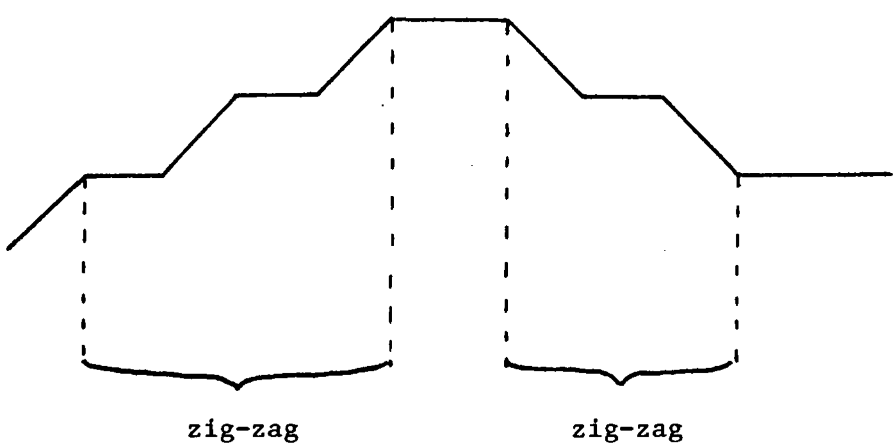
Since the slopes of the four candidates are all between zero and one, the states that they steer to may be at noninteger values at intermediate time stages. The three terms of the cost functional all take on different values. They must be recalculated based on the new megawatt outputs and new rates of changes of metawatt output associated with each candidate's solution.

The line that yields the minimum cost in the interval (LOEND1, HIEND) is chosen as the singular optimal trajectory. Its slope is the singular optimal control and the interval between its starting and ending points (may be LOEND or HIEND1) is the singular interval.

The computer is programmed to plot the time responses with zig-zag trajectories replaced by the appropriate straight line segment.



(a) Nonzig-zag



(b) With zig-zag

Fig. 4.4 Intervals with Both Increasing and Decreasing Trajectories

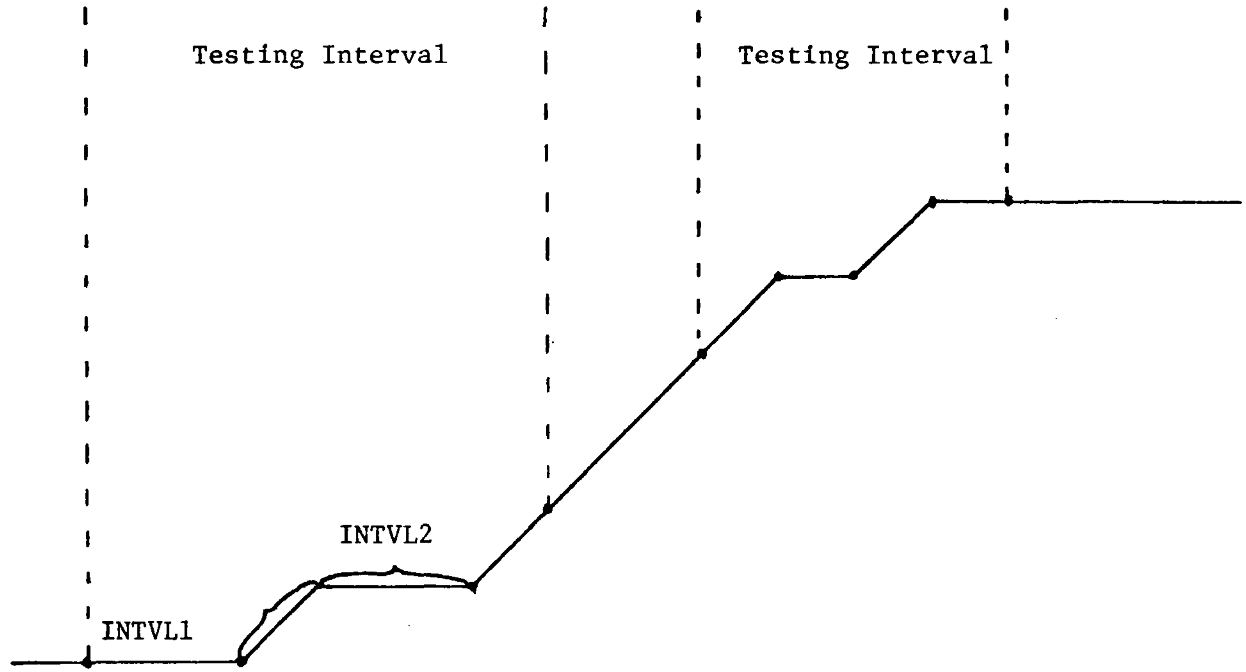


Fig. 4.5 Definition of Testing Intervals

5. EXAMPLES AND APPLICATIONS

The multi-pass dynamic programming algorithm was developed in the previous chapter. This chapter gives some examples and applications of the algorithm in the dynamic economic dispatch of power systems. Examples for parameter sensitivity study and the application to the ramp change of load case are presented.

5.1 Effect of Changing Parameter Values

Once the targets are determined by static economic dispatch, an intuitive control for the system is to move all the generators to reach their targets as rapidly as possible and then stay at those values for the rest of the control interval. However, this is not the optimal way. Fig. 5.1[7] has some generators which have optimal trajectories with values greater than both the initial and the target values. These generators should be driven to higher generation levels to help minimize the megawatt mismatch in some intervals where the cost due to megawatt mismatch dominates the overall cost. This is the effect of the c_3 term of the cost functional which penalizes the unnecessary change of generation levels. Since c_3 is small, changing the generation levels is only lightly penalized. The overall cost can be minimized following the optimal control law and the optimal trajectories shown in the figure.

Another feature of the above example is that the c_2 term, which penalizes the megawatt mismatch, is so large compared to the c_3 term that all generators help drive the overall system trajectory to approach its megawatt target as rapidly as possible.

On the other hand, the c_4 term penalizes the fuel cost. Since the

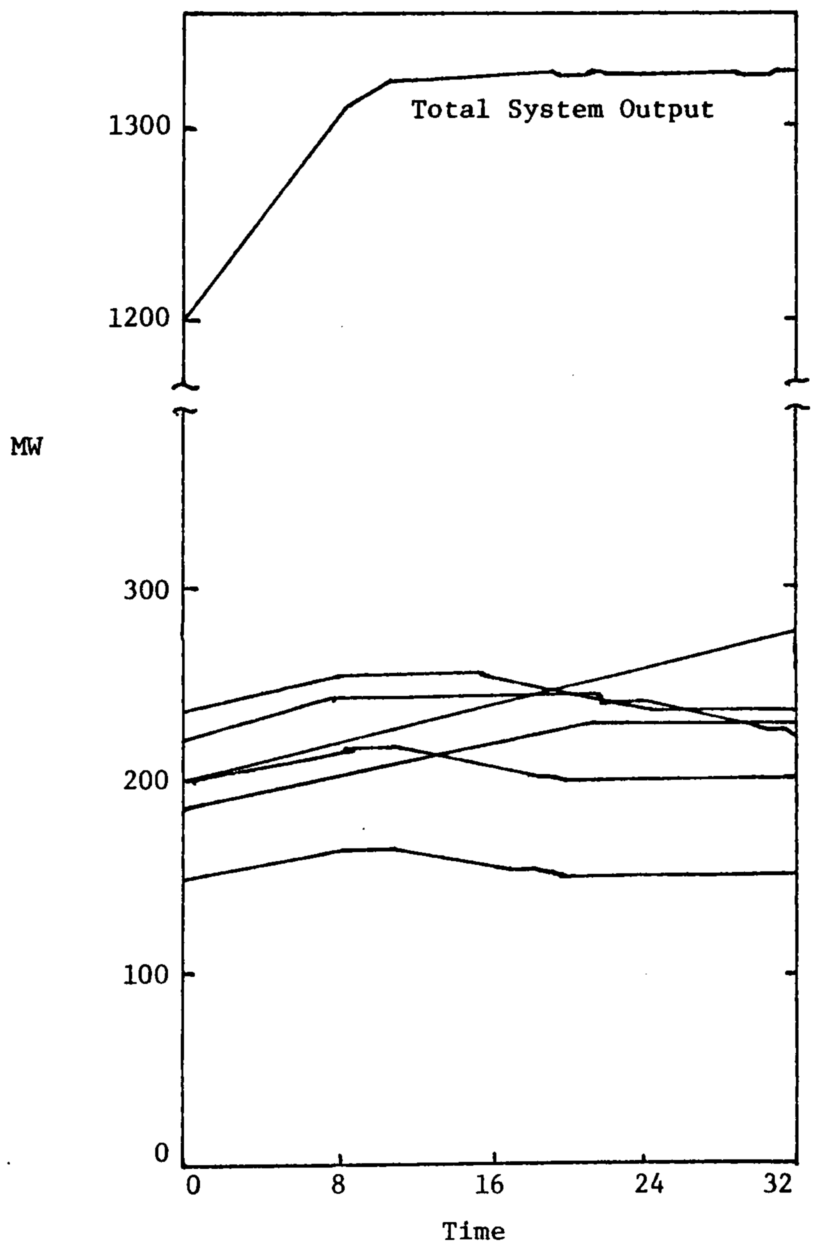


Fig. 5.1 Optimal Trajectories
(Effect of C_3)

heat rate characteristic is an increasing function, penalizing the fuel cost means penalizing megawatt outputs. Large values of c_4 tend to keep the megawatt generation as low as possible. That is, generators should stay at lower generation levels until they must increase to reach their targets. This effect is opposite to the effect of the c_2 term. A compromise must be reached and a tradeoff is shown in Fig. 5.2.

5.2 Application: Response to the Ramp Increase of Load

This program can be applied to find the optimal control and optimal trajectory in response to a ramp load such as the morning rise of the power load curve over a specified time interval. A fictitious generator must be created to solve this problem.

Let us assume that the desired control interval is (0,T), and the slope of the morning rise power load curve is U. The initial and target loads of the system are L_0 and L_T, respectively. That is,

$$\frac{L_T - L_0}{T} = 0 \tag{5.2.1}$$

and

$$L(t) = L_0 + tU(t) \tag{5.2.2}$$

Then, we can create a fictitious generator with the following characteristics:

$$x_0 - x_t = L_T - L_0$$

where x_0 is initial value,

x_t is target value,

$u = -U$ is its rate of change of MW output.

Of course, this design will make this generator work as the time-critical generator because it operates at its maximum rate throughout (0,T) to

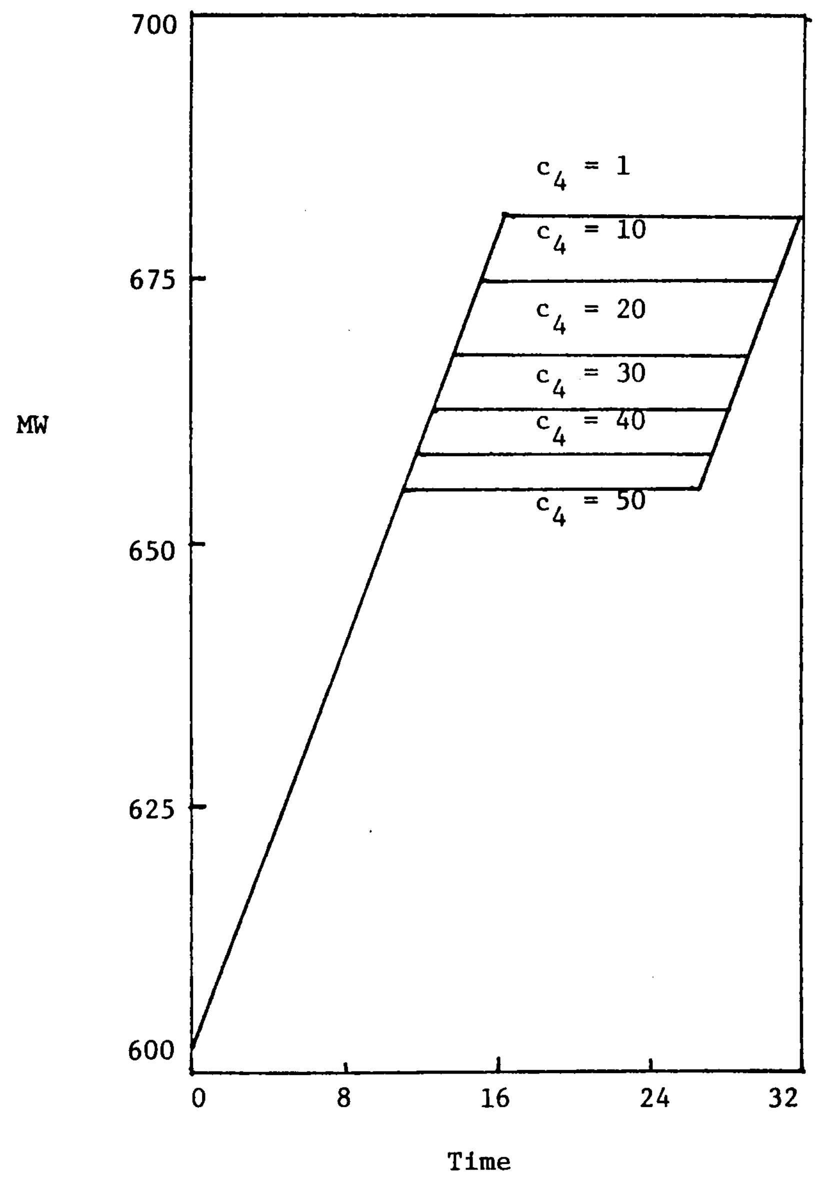


Fig. 5.2 System MW vs. Time (c_2 fixed)

reach its target, and its optimal trajectory is the straight line:

$$x_{fic} = L_T - Ut \qquad (5.2.3)$$

The sum of the megawatt output of this generator and the total system megawatt output will remain constant throughout (o,T) at the value: $L_0 + L_T$. At a given instant t, the total system generators will share the load:

$$L(t) = L_0 + L_T - x_{fic}(t) = L_0 + Ut \qquad (5.2.4)$$

An example is shown in Fig. 5.3. Generator 6 is the fictitious generator. Generators 1 through 5 share the ramp load.

The target values of all the generators, including the fictitious generator, are computed by the program. In order to keep the target of the fictitious generator at its desired value, the incremental cost of this generator must be held at a value higher than that of any other generator. Fig. 5.4 shows this condition.

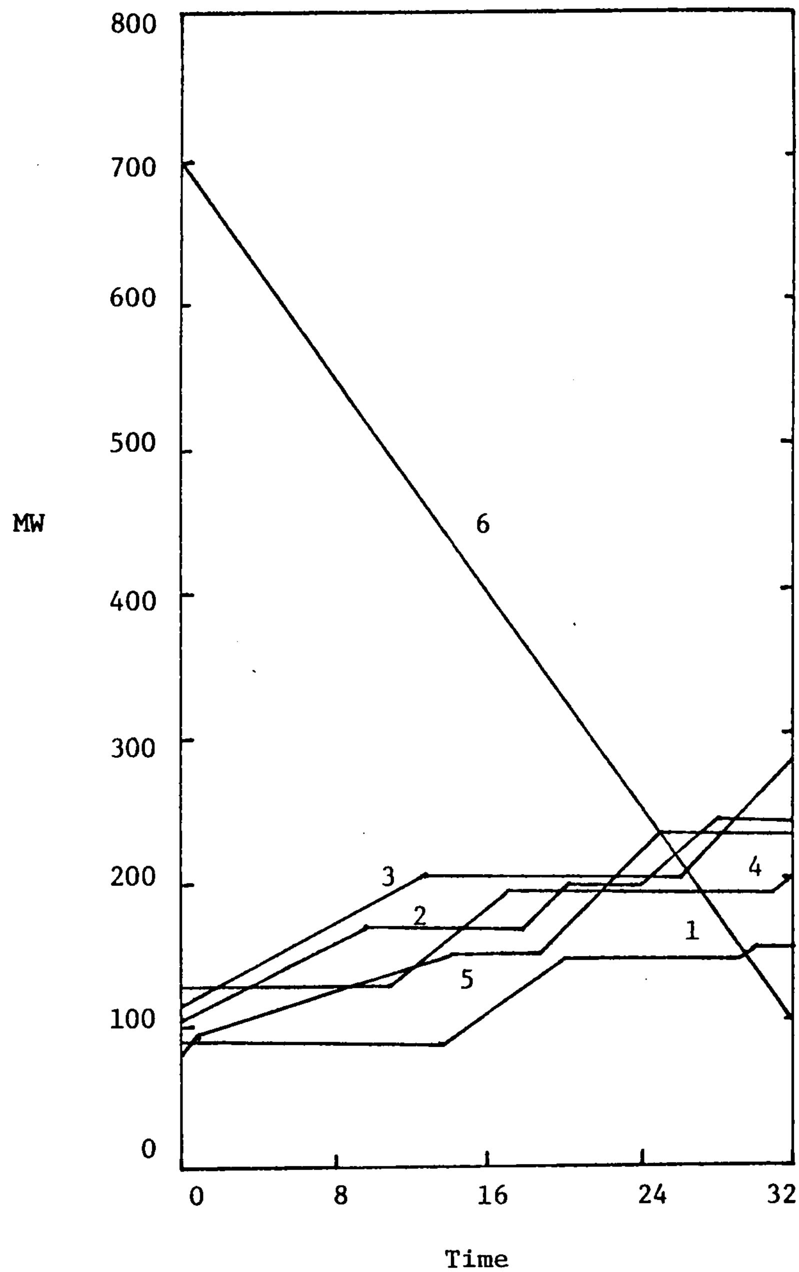


Fig. 5.3 Example for Ramp Load

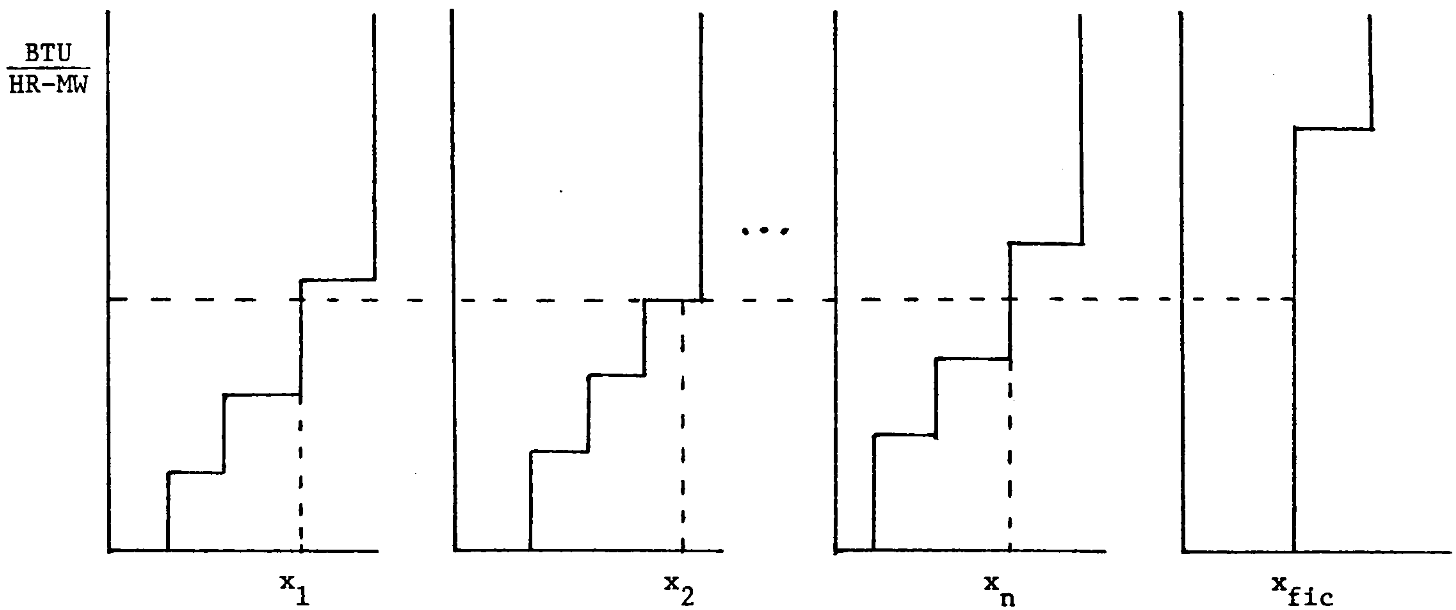


Fig. 5.4 Incremental Cost vs. MW for System with Fictitious Generator

6. CONCLUSION AND DISCUSSION

The AGC of a control area of a power system combines the load frequency control problem and economic dispatch problem together. The multi-pass dynamic programming algorithm is developed to solve this problem. Optimal controls and optimal trajectories for several generators are found so that these generators can reach their targets in an optimal way.

6.1 Comparison of Multi-Pass Dynamic Programming With Conventional Dynamic Programming

Conventional dynamic programming algorithms require large amounts of computer time and storage. For a 32-stage and 5 maneuverable generator case, the smallest possible feasible region has only $3^5 \times 31 + 2 =$ 7,535 grid points (Fig. 6.1), which is a very special case. For the typical case of Fig. 3.2 the grid contains 26,879,457 points. Conventional dynamic programming algorithms search over all these grid points.

The multi-pass dynamic programming algorithm, however, examines only the grid points one increment above the nominal and one increment below the nominal for each stage for each generator. Hence, it searches over no more than $3^5 \times 31 + 2 = 7{,}535$ points and saves much computer time and storage.

6.2 Discussion and Suggestions for Further Work

The superiority of multi-pass dynamic programming over conventional dynamic programming is apparent. It can be applied to many other problems which conventional dynamic programming fails to solve. However, although the optimal solution is found here, the resulting trajectories

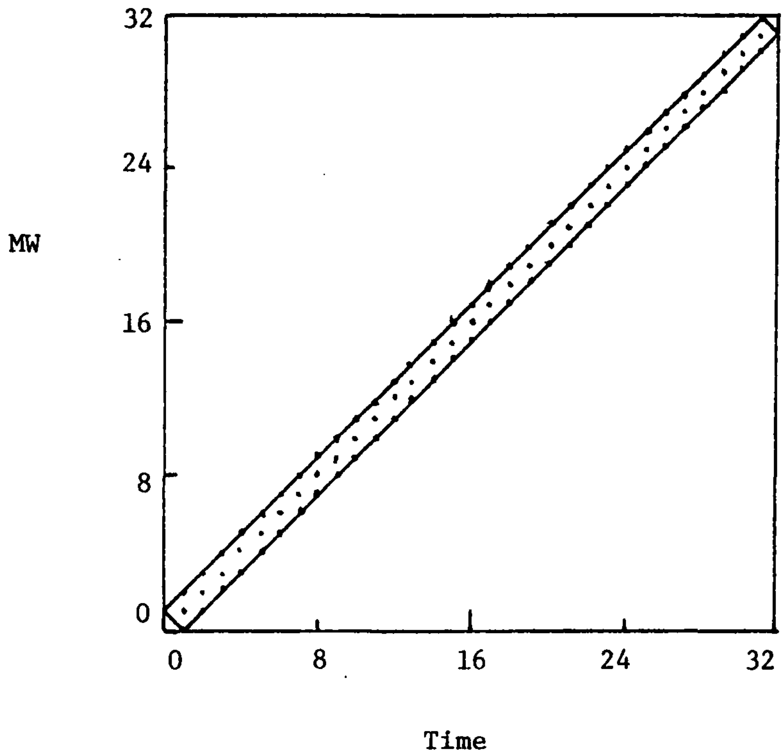


Fig. 6.1 The Smallest Feasible Region

represent a local optimum; no proof is available to verify that the solution is optimal in a global sense.

The AGC problem of a power system is an on line task. It must be solved in a very short time. Although we can program for as many generators as desired, the solution time, which increases rapidly with the number of generators, is not reasonable for on-line application. Using the present algorithm, the system would be broken into areas of approximately six generators.

The solution time depends strongly on the dimension of the system. For the examples run in this research, the longest job required 7 minutes, 28 seconds to solve a 5 maneuverable generator case. In a shorter example, only 52 seconds were needed to solve another 5 maneuverable generator case. For a very general two maneuverable generator case, only 10 seconds were required.

The weighting parameters of the cost functional are not determined. Utilities which would use this program would determine a set of parameter values to fit their own requirements. Effects of all the weighting parameter values are discussed in Chapter 5.

The AGC problem is studied by multi-pass dynamic programming. Further work may use different dynamic programming approaches to improve the solution time, to increase the number of generators, and to make it suitable for on line application.

As mentioned in Chapter 2, the network subsystem control problem was formulated and solved in [2,6]. This thesis deals with the area mechanical power control subsystem. Further work may combine the two subsystems together to solve the total system problem and to encourage electric utilities to make use of it in practical applications.

This work is based on a fixed value for area load, although an extension permits the study of responses to a ramp change of load. Further work should examine the possibility of extending these results to other, more general, load conditions.

The algorithm used in this study determines target megawatt values based on economic dispatch, neglecting transmission losses. Further work should incorporate an economic dispatch program which includes the effects of these losses.

BIBLIOGRAPHY

1. D. N. Ewart, "Automatic Generation Control--Performance Under Normal Conditions", Proc. System Engineering for Power: Status and Prospects, Henniker, H. H., August 1975. ERDA 76-66, CONF--750867, pp. 1-14.

2. T. E. Bechert and H. G. Kwatny, "On the Optimal Dynamic Dispatch of Real Power", IEEE Trans. Power App. Syst., Vol. PAS-91, May/June 1972, pp. 889-898.

3. H. G. Kwayny and T. E. Bechert, "On the Structure of Optimal Area Controls in Electric Power Networks", IEEE Trans. Automatic Control, Vol. AC-18, No. 2, April 1973, pp. 167-172.

4. D. E. Kirk, "Optimal Control Theory: An Introduction", Prentice-Hall, Inc., New Jersey, 1970.

5. O. I. Elgerd, "Electric Energy Systems Theory: An Introduction", McGraw-Hill, 1971.

6. T. E. Bechert, "Dynamic Optimal Allocation of Electric Power Generation", Ph.D. Dissertation, Drexel University, 1970.

7. T. E. Bechert, "Dynamic Economic Dispatch of Real Power by Dynamic Programming--A Feasibility Study", Proc. IEEE Region 3 Meeting, April 1976, pp. 1B-1, 1B-3.

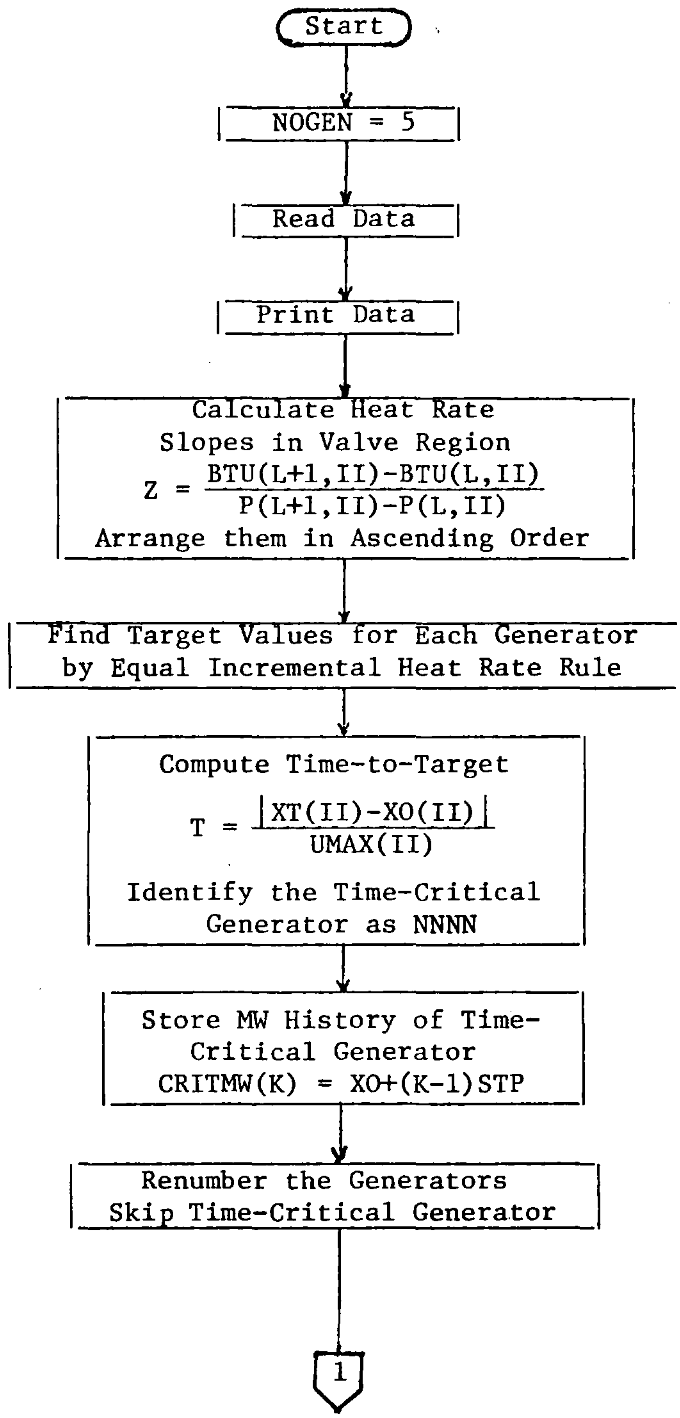
8. T. E. Bechert, "Multi-Pass Dynamic Programming in Optimal Dynamic Load Allocation", Proc. Milwaukee Symposium on Automatic Computation and Control, April 1976, pp. 113-117.

9. T. E. Bechert and N. Chen, "Area Automatic Generation Control by Multi-Pass Dynamic Programming", IEEE Trans. on Power App. Syst., 1977.

10. N. Chen and T. E. Bechert, "Parameter Sensitivity and Singular Solutions in Dynamic Optimal Real Power Dispatch", Proc. IEEE Region 3 Meeting, April 1977.

APPENDIX

FLOW CHART FOR MULTI-PASS DYNAMIC PROGRAMMING



1

Add Roundoff Error of Previous Generator to the Target Value, Calculate Integer Difference Between Initial and Target Values of each Maneuverable Generator

↓

Spread the Initial and Target Integer Equally to Both Sides of the Midpoint 17 of the Interval (1,33)

↓

Store ΔX_j and Integer Values of Valve Points

↓

Determine the Integer Values of the Minimum and Maximum Limits LXMIN(I), LXMAX(I)

↓

Determine the MW Value X_{ji} of Each Integer j in the Feasible Region of Generator i, also compute $COST4_{ji}$ of Integer j and Generator i for All Integer and Maneuverable Generators

↓

Calculate $COST3_i = b_i \; U_{maxi} \; Y_3$

↓

Determine the Edge of Feasible Region at Each Stage

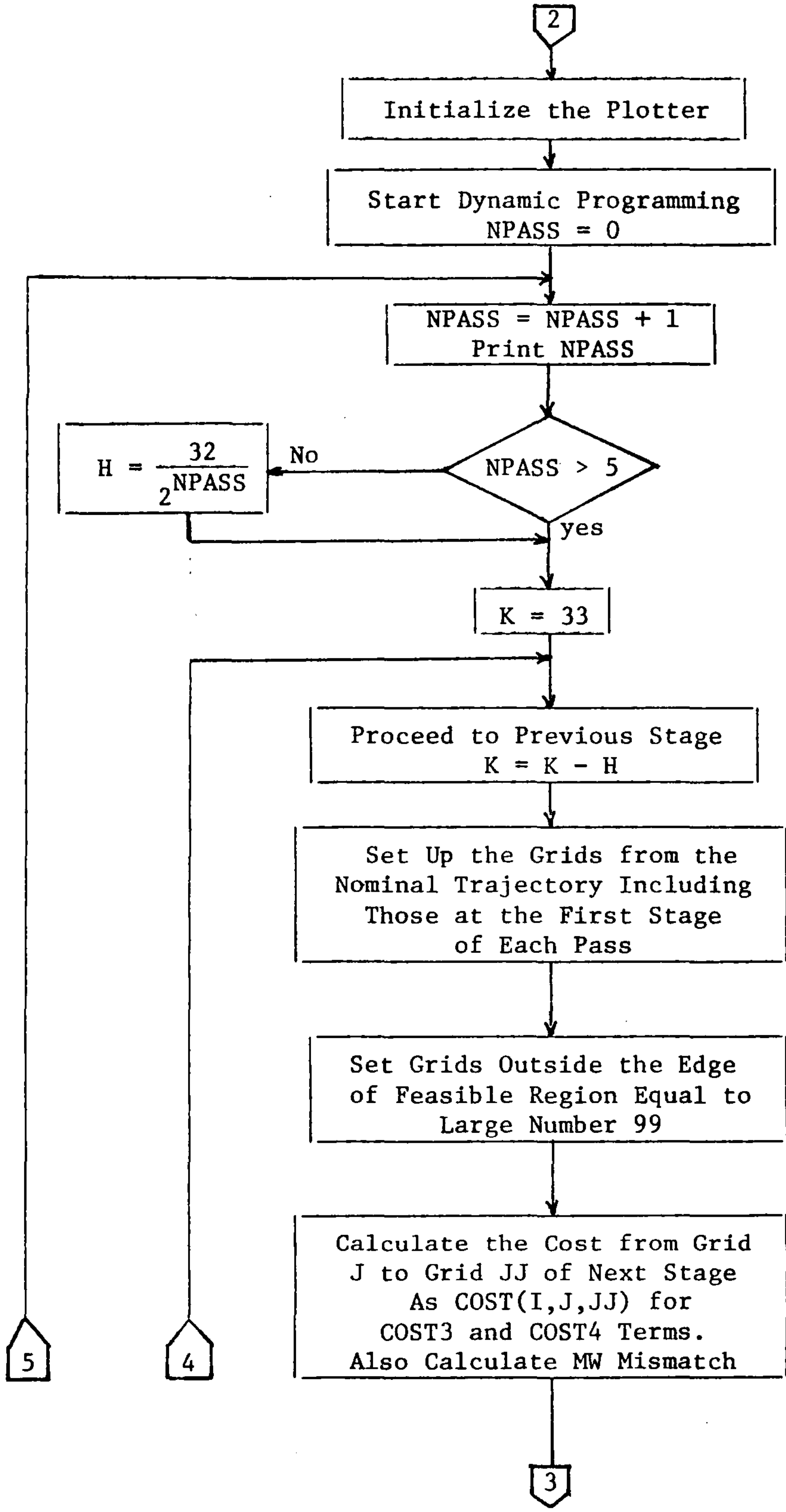
$LXHI_{ki} = MIN(LXO_i + K - 1,\; LXT_i + 33 - K,\; LXMAX_i)$

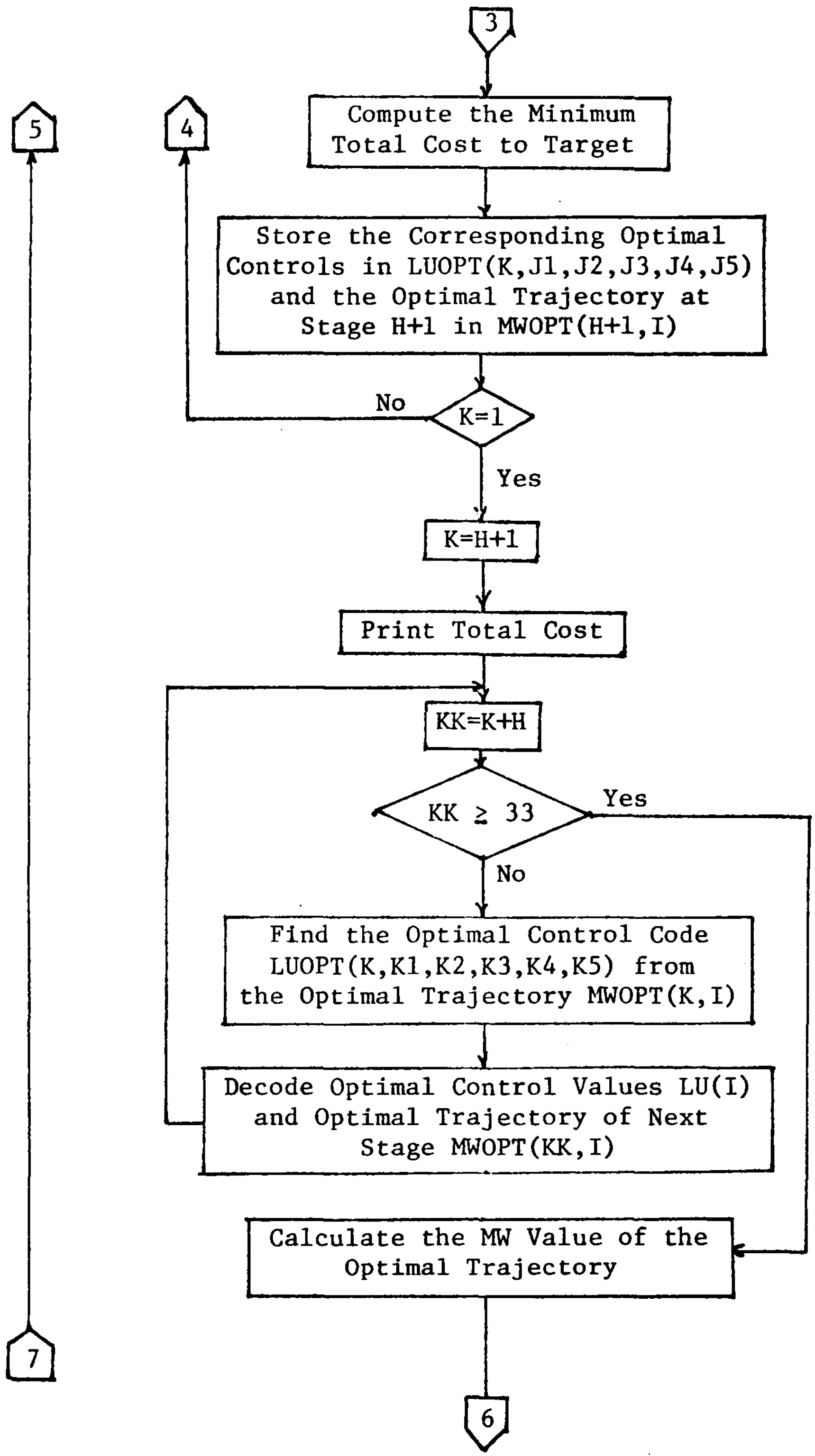
$LXLO_{ki} = MAX(LXO_i - K + 1,\; LXT_i - 33 + K,\; LXMIN_i)$

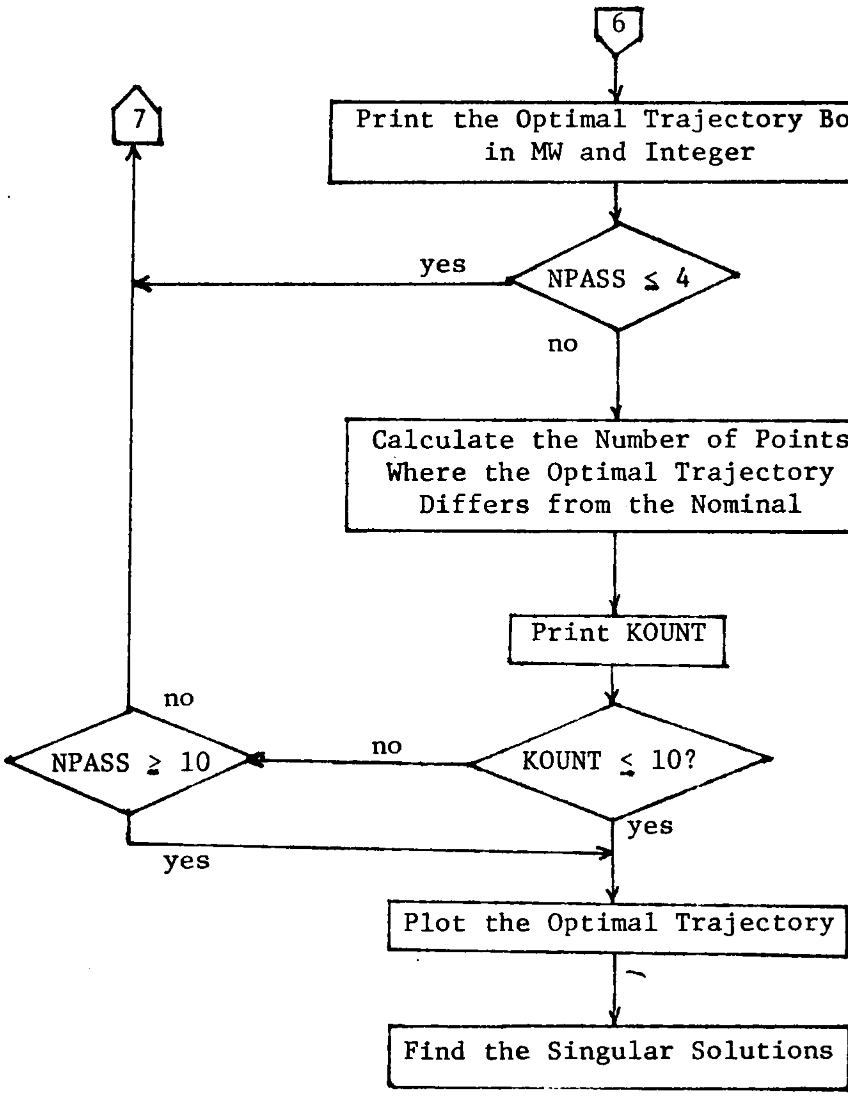
↓

Print Valve Points, Initial, and Target MW and Integers, Desired MW, Maximum and Minimum MW, MW Rate, and MW Increment of Each Generator Also Print Time Increment

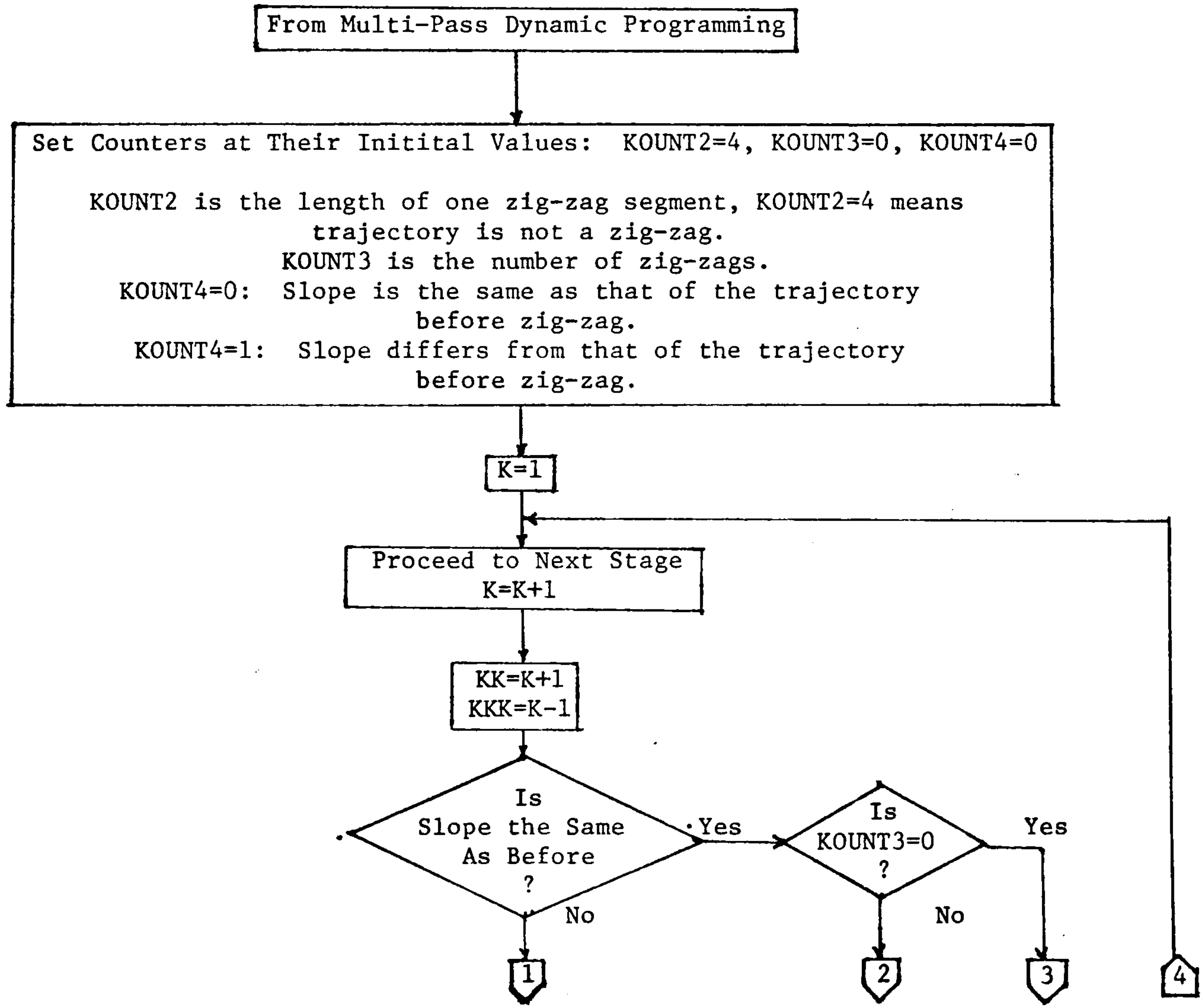
2

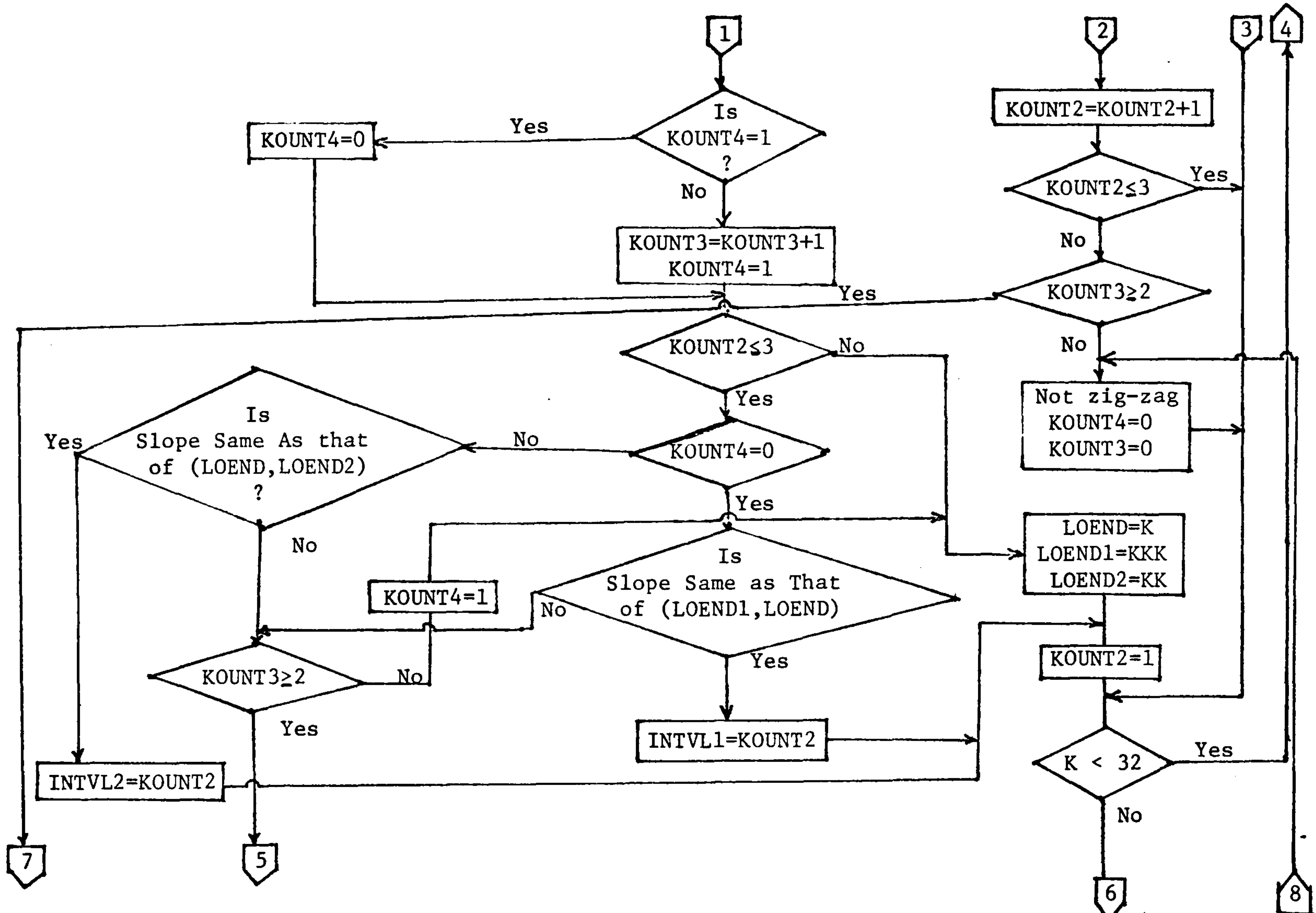

2
Initialize the Plotter
Start Dynamic Programming
NPASS = 0
NPASS = NPASS + 1
Print NPASS
H = 32 / 2^NPASS
No
NPASS > 5
yes
K = 33
Proceed to Previous Stage
K = K - H
Set Up the Grids from the
Nominal Trajectory Including
Those at the First Stage
of Each Pass
Set Grids Outside the Edge
of Feasible Region Equal to
Large Number 99
Calculate the Cost from Grid
J to Grid JJ of Next Stage
As COST(I,J,JJ) for
COST3 and COST4 Terms.
Also Calculate MW Mismatch
5
4
3


3
5
4
Compute the Minimum Total Cost to Target
Store the Corresponding Optimal Controls in LUOPT(K,J1,J2,J3,J4,J5) and the Optimal Trajectory at Stage H+1 in MWOPT(H+1,I)
No
K=1
Yes
K=H+1
Print Total Cost
KK=K+H
KK ≥ 33
Yes
No
Find the Optimal Control Code LUOPT(K,K1,K2,K3,K4,K5) from the Optimal Trajectory MWOPT(K,I)
Decode Optimal Control Values LU(I) and Optimal Trajectory of Next Stage MWOPT(KK,I)
Calculate the MW Value of the Optimal Trajectory
7
6

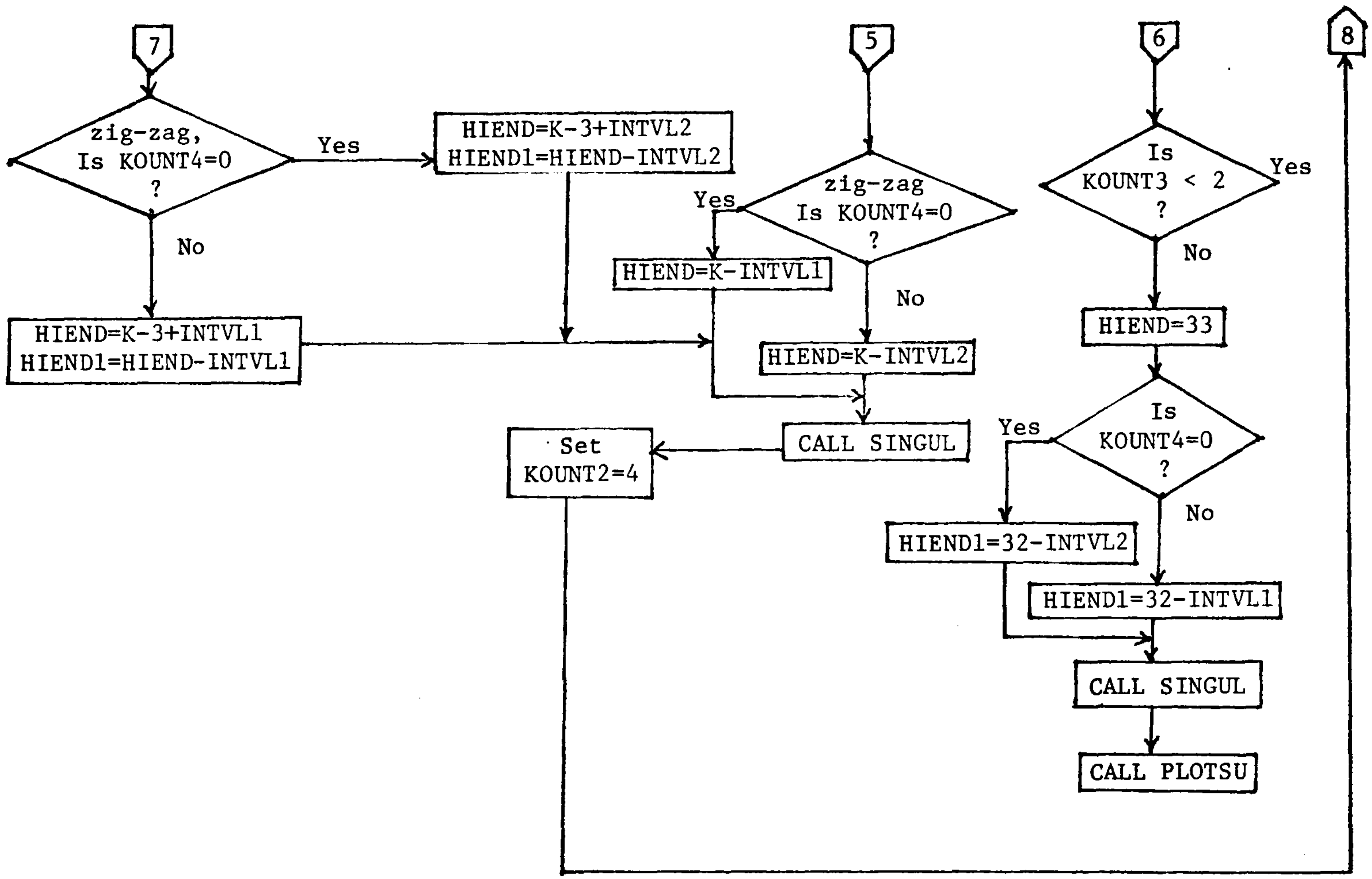

6
7
Print the Optimal Trajectory Both in MW and Integer
yes
NPASS ≤ 4
no
Calculate the Number of Points Where the Optimal Trajectory Differs from the Nominal
Print KOUNT
no
NPASS ≥ 10
no
KOUNT ≤ 10?
yes
yes
Plot the Optimal Trajectory
Find the Singular Solutions

FLOW CHART FOR FINDING SINGULAR INTERVALS



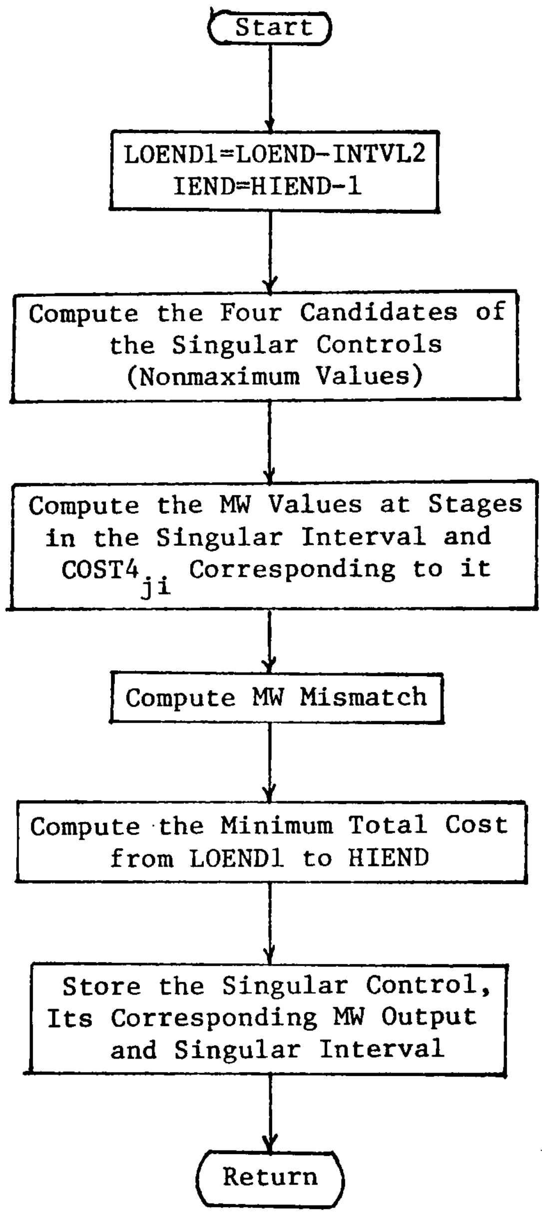


1
2
3
4
Is KOUNT4=1 ?
Yes
No
KOUNT4=0
KOUNT3=KOUNT3+1
KOUNT4=1
KOUNT2=KOUNT2+1
KOUNT2≤3
Yes
No
KOUNT3≥2
Yes
No
KOUNT2≤3
No
Yes
Not zig-zag
KOUNT4=0
KOUNT3=0
Is Slope Same As that of (LOEND,LOEND2) ?
Yes
No
KOUNT4=0
No
Yes
LOEND=K
LOEND1=KKK
LOEND2=KK
Is Slope Same as That of (LOEND1,LOEND)
No
Yes
KOUNT4=1
KOUNT3≥2
No
Yes
KOUNT2=1
INTVL1=KOUNT2
INTVL2=KOUNT2
K < 32
Yes
No
7
5
6
8



7
zig-zag, Is KOUNT4=0 ?
Yes
No
HIEND=K-3+INTVL2
HIEND1=HIEND-INTVL2
HIEND=K-3+INTVL1
HIEND1=HIEND-INTVL1
5
zig-zag Is KOUNT4=0 ?
Yes
No
HIEND=K-INTVL1
HIEND=K-INTVL2
CALL SINGUL
Set KOUNT2=4
6
Is KOUNT3 < 2 ?
Yes
No
HIEND=33
Is KOUNT4=0 ?
Yes
No
HIEND1=32-INTVL2
HIEND1=32-INTVL1
CALL SINGUL
CALL PLOTSU
8

FLOW CHART FOR DETERMINING THE SINGULAR SOLUTION



```
      IMPLICIT INTEGER*4(H-N)
      DIMENSION XXT(5),XXMAX(5),XXMIN(5)
      DIMENSION            KCRIT(33),TK(33)
      DIMENSION MJ(5),JM(5),LU(5),LUU(5),LOJJ(5,3),HIJJ(5,3),KJ(5)
C
      DIMENSION NUMV(6),XO(6),UMAX(6),XMIN(6),XMAX(6),BETA(6),BTU(7,6),
     *          P(7,6),LXV(7,6)
      DIMENSION SLP(6,6),ASLP(36),LEV(36),IGEN(36),XT(6)
      DIMENSION XY(5),YX(5),UMX(5),DELX(5),COST3(5)
      DIMENSION NUMOLD(5),LXO(5),LXT(5),LXMIN(5),LXMAX(5),JLO(5),JHI(5)
      DIMENSION CT(5),DL1(5),COS2(5),COS3(5)
      DIMENSION US(33,5),COS4(33,5),XS(33,5)                           C
      DIMENSION OPT(33),TX(33),XHI(33),XLO(33)
C
      COMMON/BLOK1/ TX,XHI,XLO,OPT
      COMMON/BLOK2/ NOGEN,OP,CRITMW,ZYX
      COMMON/BLOK3/ I,HIEND,LOEND,US,MWOPT,HIEND1,LOEND1,UMX,XS,X,
     *LXT,COS4,COST4,CT,DL2,DELX,DL1,COS2,GAMMA2,NUMOLD,COS3,DELTAT,
     *BETA,GAMMA3,P,NUMV,BTU,SLP,GAMMA4,SCT,ISG,IEND,INTVL2
C
      DIMENSION OP(33,5),ZYX(33),CRITMW(33)                            C
      DIMENSION X(33,5),COST4(33,5),LXHI(33,5),LXLO(33,5),MWOPT(33,5)  C
      DIMENSION CST(3,3,3,3,3),CCST(3,3,3,3,3),LUOPT(33,3,3,3,3,3)     C
      DIMENSION MW(33,5,3),COST(5,3,3),DEL(5,3,3)                      C
C
      NOGEN=5                                                          C
C
      NUMGEN=NOGEN+1
      DO 100 II=1,NUMGEN
      DO 100 M=1,7
      P(M,II)=0.
  100 BTU(M,II)=0.
```

```
      READ(5,4)PL
    4 FORMAT(F7.1)
      TOTMAX=0.
      TOTMIN=0.
      TOTRAT=0.
      DO 105 II=1,NUMGEN
      READ(5,5)X0(II),XMIN(II),XMAX(II),UMAX(II),BETA(II),VNUM
    5 FORMAT(16F5.1)
      NUMV(II)=VNUM
      MM=NUMV(II)+1
      TOTMAX=TOTMAX+XMAX(II)
      TOTMIN=TOTMIN+XMIN(II)
      TOTRAT=TOTRAT+UMAX(II)
  105 READ(5,5)(P(M,II),BTU(M,II),M=1,MM)
      READ(5,5)GAMMA2,GAMMA3,GAMMA4
C
      WRITE(6,10)
   10 FORMAT('1','LISTING OF SYSTEM INPUT DATA')
      WRITE(6,15)PL
   15 FORMAT('0','SYSTEM LOAD =',F7.1,' MEGAWATTS')
      WRITE(6,20)(II,II=1,NUMGEN)
   20 FORMAT('0','GENERATOR NUMBER ',6(I8,5X))
      WRITE(6,25)(X0(II),II=1,NUMGEN)
   25 FORMAT('0','INITIAL MEGAWATTS',6(F10.1,3X))
      WRITE(6,30)(XMIN(II),II=1,NUMGEN)
   30 FORMAT(' ','MINIMUM MEGAWATTS',6(F10.1,3X))
      WRITE(6,35)(XMAX(II),II=1,NUMGEN)
   35 FORMAT(' ','MAXIMUM MEGAWATTS',6(F10.1,3X))
      WRITE(6,40)(UMAX(II),II=1,NUMGEN)
   40 FORMAT(' ','MAX MEGAWATT RATE',6(F10.1,3X))
      WRITE(6,45)(NUMV(II),II=1,NUMGEN)
   45 FORMAT('0','NO. VALVE REGIONS',6(I8,5X))
```

```
      DO 110 M=1,7
  110 WRITE(6,50)(P(M,II),BTU(M,II),II=1,NUMGEN)
   50 FORMAT(' ','(MW, HEAT RATE)  ',6(F7.1,F6.1))
      WRITE(6,55)(BETA(II),          II=1,NUMGEN)
   55 FORMAT('0','BETA =',11X,6(F10.1,3X))
      WRITE(6,60)GAMMA2,GAMMA3,GAMMA4
   60 FORMAT(' ','GAMMA2 =',F5.1,',  GAMMA3 =',F5.1,',  GAMMA4 =',F5.1)
C
C     THIS ROUTINE COMPUTES THE HEAT RATE SLOPE IN EACH VALVE REGION
C     AND ARRANGES THESE SLOPES IN ASCENDING ORDER.
C
      NOLEV=0
      DO 140 II =1,NUMGEN
      NV=NUMV(II)
      DO 135 L=1,NV
      Z = (BTU(L+1,II)-BTU(L,II))/(P(L+1,II)-P(L,II))
      SLP(L,II)=Z
      IF(NOLEV.EQ.0)GO TO 125
      DO 120 M=1,NOLEV
      IF(Z.GE.ASLP(M))GO TO 120
C
C     THIS LOOP MAKES ROOM FOR THE NEW SLOPE.
C
      NN=NOLEV+1-M
      DO 115 MM=1,NN
      I=NOLEV+1-MM
      LEV(I+1)=LEV(I)
      IGEN(I+1)=IGEN(I)
      ASLP(I+1)=ASLP(I)
  115 CONTINUE
      GO TO 130
  120 CONTINUE
```

```
  125 M=NOLEV+1
  130 LEV(M)=L
      IGEN(M)=II
      ASLP(M)=Z
      NOLEV=NOLEV+1
  135 CONTINUE
  140 CONTINUE
C
C     THIS ROUTINE FINDS TARGET VALUES FOR EACH GENERATOR.
C
      GEN=0.
      DO 145 II=1,NUMGEN
      XT(II)=P(1,II)
  145 GEN=GEN+XT(II)
      DO 155 M=1,NOLEV
      L=LEV(M)
      II=IGEN(M)
      XT(II)=P(L+1,II)
      GEN=GEN-P(L,II)
      IF(GEN+XT(II).LT.PL)GO TO 150
      XT(II)=PL-GEN
      GO TO 160
  150 GEN=GEN+XT(II)
  155 CONTINUE
C
C     THIS ROUTINE COMPUTES TIME-TO-TARGET AND IDENTIFIES THE
C     TIME-CRITICAL GENERATOR.
C
  160 T=0.
      DO 170 II=1,NUMGEN
      TIMTRY=ABS(XT(II)-X0(II))/UMAX(II)
      IF(TIMTRY.LE.T)GO TO 170
```

```
      T=TIMTRY
      NNNN=II
  170 CONTINUE
      DELTAT=T/32.
      DO 165 K=1,33
  165 TK(K)=DELTAT*(K-1)
C
C     THESE STEPS PERMIT PRINTING MW HISTORY OF TIME-CRITICAL GENERATOR.
C
      STP=(XT(NNNN)-XO(NNNN))/32.
      DO 180 K=1,33
      CRITMW(K)=XO(NNNN)+STP*(K-1)
      IF(STP.LT.0.)GO TO 175
      KCRIT(K)=K
      GO TO 180
  175 KCRIT(K)=34-K
  180 CONTINUE
C
C     RE-NUMBER THE GENERATORS, SKIPPING THE TIME-CRITICAL GENERATOR.
C
      DO 190 I=1,NOGEN
      IF(I.GE.NNNN)GO TO 185
      NUMOLD(I)=I
      GO TO 190
  185 NUMOLD(I)=I+1
  190 CONTINUE
C
C     THESE STEPS SET UP THE GRID, ASSIGNING INTEGERS TO EACH MW VALUE.
C
      RERR=0.0
      DO 230 I=1,NOGEN
      II=NUMOLD(I)
```

```
      YX(I)=XT(II)+RERR
      YX(I)=AMIN1(YX(I),XMAX(II))
      YX(I)=AMAX1(YX(I),XMIN(II))
      TARGET=YX(I)-XO(II)
      TEMP=32.*ABS(TARGET)/T
      III=TEMP/UMAX(II)+0.5
      UMX(I)=UMAX(II)
      IF(TARGET)195,200,200
  195 LXO(I)=(34+III)/2
      LXT(I)=(34-III)/2
      GO TO 205
  200 LXO(I)=(34-III)/2
      LXT(I)=(34+III)/2
  205 DELX(I)=UMX(I)*T/32.
      XX=XO(II)-DELX(I)*LXO(I)
      DO 222 M=1,7
  222 LXV(M,II)=(P(M,II)-XX)/DELX(I)+0.5
      LXMIN(I)=(XMIN(II)-XX)/DELX(I)+0.95
      LXMAX(I)=(XMAX(II)-XX)/DELX(I)+0.05
      LXMIN(I)=MIN0(LXMIN(I),LXT(I))
      LXMAX(I)=MAX0(LXMAX(I),LXT(I))
      L=1
      JJ=MAX0(1,LXMIN(I))
      JJJ=MIN0(33,LXMAX(I))
      DO 220 J=JJ,JJJ
      XXX=XX+DELX(I)*J
      X(J,I)=XXX
  210 IF(XXX.LT.P(L+1,II)+0.2)GO TO 215
      IF(L.GE.NUMV(II))GO TO 215
      L=L+1
      GO TO 210
  215 COST4(J,I)=(BTU(L,II)+(XXX-P(L,II))*SLP(L,II))*GAMMA4
```

```
  220 CONTINUE
      COST3(I)=BETA(II)*UMX(I)*GAMMA3
      DO 225 K=1,33
      LXHI(K,I)=MINO(LXO(I)+K-1,LXT(I)+33-K,LXMAX(I))
  225 LXLO(K,I)=MAXO(LXO(I)-K+1,LXT(I)-33+K,LXMIN(I))
      XY(I)=XO(II)
      J=LXT(I)
      YX(I)=X(J,I)
C
C     ADJUST THE NEXT TARGET VALUE BY THE ROUNDOFF ERROR IN THIS ONE.
C
      RERR=XT(II)+RERR-YX(I)
  230 CONTINUE
      CORR=0.0
      DO 232 I=1,NOGEN
      J=LXO(I)
      JJ=LXT(I)
  232 CORR=CORR+COST4(J,I)-COST4(JJ,I)
      CORR=0.5*CORR
C
C     CORR WILL FORCE COST TO INCLUDE AVERAGE COST4 FOR EACH INTERVAL.
C
      XYO=XO(NNNN)
      YXT=XT(NNNN)
      DO 235 I=1,NOGEN
      XYO=XYO+XY(I)
  235 YXT=YXT+YX(I)
      PRINT 89
   89 FORMAT('0','INTEGERS ASSIGNED TO VALVE POINTS')
      DO 238 M=1,7
  238 PRINT 88,(P(M,NUMOLD(I)),LXV(M,NUMOLD(I)),I=1,NOGEN)
   88 FORMAT(1H ,'(MW, INTEG)',7X,6(F7.1,2X,I2,2X ))
```

```
      WRITE(6,65)
   65 FORMAT('1','LISTING OF OPTIMAL TRAJECTORIES')
      WRITE(6,70)(NUMOLD(I),I=1,NOGEN),NNNN
   70 FORMAT('0',8X,'GENERATOR NUMBER      ALL',6(I6,3X))
      WRITE(6,75)XYO,(XY(I),LXO(I),I=1,NOGEN),XO(NNNN),KCRIT(1)
   75 FORMAT(' ',8X,'INITIAL MW, INTEG',F7.1,6(F6.1,I3))
      WRITE(6,80)YXT,(YX(I),LXT(I),I=1,NOGEN),XT(NNNN),KCRIT(33)
   80 FORMAT(' ',8X,'TARGET  MW, INTEG',F7.1,6(F6.1,I3))
      DO 236 I=1,NOGEN
      II=NUMOLD(I)
      XXT(I)=XT(II)
      XXMAX(I)=XMAX(II)
      XXMIN(I)=XMIN(II)
  236 CONTINUE
      WRITE(6,81)PL,  (XXT(I),I=1,NOGEN),XT  (NNNN)
   81 FORMAT(' ',8X,'DESIRED MW',7X,F7.1,6(F6.1,3X))
      WRITE(6,82)TOTMAX,(XXMAX(I),I=1,NOGEN),XMAX(NNNN)
   82 FORMAT(' ',8X,'MAX MW',11X,F7.1,6(F6.1,3X))
      WRITE(6,83)TOTMIN,(XXMIN(I),I=1,NOGEN),XMIN(NNNN)
   83 FORMAT(' ',8X,'MIN MW',11X,F7.1,6(F6.1,3X))
      WRITE(6,84)TOTRAT,(UMX(I),I=1,NOGEN),UMAX(NNNN)
   84 FORMAT(' ',8X,'MW RATE',10X,F7.1,6(F6.1,3X))
      WRITE(6,86)(DELX(I),I=1,NOGEN),STP
   86 FORMAT(' ',8X,'MW INCREMENT',13X,6(F7.3,2X))
      WRITE(6,87)DELTAT
   87 FORMAT(' ',8X,'TIME INCREMENT = ',F6.3)
      WRITE(6,85)
   85 FORMAT('0','PASS     STAGE  TIME  COST'/)
      DO 240 I=1,NOGEN
      MWOPT(33,I)=LXT(I)
      MWOPT( 1,I)=LXO(I)
      MW(33,I,1)=99
```

```
      MW(33,I,2)=LXT(I)
  240 MW(33,I,3)=99
C
C     START DYNAMIC PROGRAMMING FROM PASS 1
C
      NPASS=0
  245 NPASS=NPASS+1
      WRITE(6,90)NPASS
   90 FORMAT(' ',I3)
      IF(NPASS.GT.5)  GO TO 246
      H=32/2**NPASS
  246 K=33
  250 K=K-H
      KK=K+H
      KKK=K-H
      DO 285 I=1,NOGEN
      IF(NPASS.LE.5) GO TO 251
      GO TO 275
  251 IF(MOD(K-1,2*H).EQ.0)GO TO 275
      IF(KKK.GT.1)GO TO 270
      IF(LXO(I)-MWOPT(KK,I))255,260,265
  255 MW(K,I,2)=MWOPT(KK,I)-H
      IF(MW(K,I,2).LT.LXMIN(I))GO TO 260
      GO TO 280
  260 MW(K,I,2)=MWOPT(KK,I)
      GO TO 280
  265 MW(K,I,2)=MWOPT(KK,I)+H
      IF(MW(K,I,2).GT.LXMAX(I))GO TO 260
      GO TO 280
  270 MW(K,I,2)=(MWOPT(KKK,I)+MWOPT(KK,I))/2
      GO TO 280
  275 MW(K,I,2)=MWOPT(K,I)
```

```
  280 MW(K,I,1)=MW(K,I,2)-H
      MW(K,I,3)=MW(K,I,2)+H
  285 CONTINUE
      DO 295 I=1,NOGEN
      JLO(I)=1
      JHI(I)=3
      IF(MW(K,I,1).GE.LXLO(K,I))GO TO 290
      MW(K,I,1)=99
      JLO(I)=2
  290 IF(MW(K,I,3).LE.LXHI(K,I))GO TO 295
      MW(K,I,3)=99
      JHI(I)=2
  295 CONTINUE
C
C     COMPUTE THE COST FROM GRID J TO GRID JJ OF NEXT STAGE.
C
      DO 325 I=1,NOGEN
      LOJ=JLO(I)
      HIJ=JHI(I)
      DO 325 J=LOJ,HIJ
      LOJJ(I,J)=0
      MMM=MW(K,I,J)
      COST1=COST4(MMM,I)
      DO 325 JJ=1,3
      MM=MW(KK,I,JJ)-MMM
      IF(IABS(MM).GT.H)GO TO 325
  300 IF(LOJJ(I,J).NE.0)GO TO 305
      LOJJ(I,J)=JJ
  305 HIJJ(I,J)=JJ
      IF(MM.EQ.0)GO TO 315
      IF(K.GT.1)GO TO 310
      COST(I,J,JJ)=COST1 + COST3(I)*ABS(FLOAT(MM)/FLOAT(H))
```

```
      GO TO 320
  310 COST(I,J,JJ)=COST1 + COST3(I)
      GO TO 320
  315 COST(I,J,JJ)=COST1
  320 LDEL=2*(MMM-LXT(I))+MM
      DEL(I,J,JJ)=LDEL*DELX(I)/2.
  325 CONTINUE
      DELNNN=(CRITMW(K)+CRITMW(KK))/2.0-CRITMW(33)
      J1=1
      J2=1
      J3=1
      J4=1
      J5=1
      JJ1=1
      JJ2=1
      JJ3=1
      JJ4=1
      JJ5=1
      JLO1=1
      JLO2=1
      JLO3=1
      JLO4=1
      JLO5=1
      JHI1=1
      JHI2=1
      JHI3=1
      JHI4=1
      JHI5=1
      LOJJ1=1
      LOJJ2=1
      LOJJ3=1
      LOJJ4=1
```

```
      LOJJ5=1
      HIJJ1=1
      HIJJ2=1
      HIJJ3=1
      HIJJ4=1
      HIJJ5=1
      DO 330 I=1,5
  330 MJ(I)=1
      IF(NOGEN-5)336,335,540
  335 JLO5=JLO(5)
      JHI5=JHI(5)
  336 DO 420 J5=JLO5,JHI5
      IF(NOGEN-4)341,340,337
  337 MJ(5)=J5
      LOJJ5=LOJJ(5,J5)
      HIJJ5=HIJJ(5,J5)
  340 JLO4=JLO(4)
      JHI4=JHI(4)
  341 DO 420 J4=JLO4,JHI4
      IF(NOGEN-3)346,345,342
  342 MJ(4)=J4
      LOJJ4=LOJJ(4,J4)
      HIJJ4=HIJJ(4,J4)
  345 JLO3=JLO(3)
      JHI3=JHI(3)
  346 DO 420 J3=JLO3,JHI3
      IF(NOGEN-2)351,350,347
  347 MJ(3)=J3
      LOJJ3=LOJJ(3,J3)
      HIJJ3=HIJJ(3,J3)
  350 JLO2=JLO(2)
      JHI2=JHI(2)
```

```
351 DO 420 J2=JLO2,JHI2
    IF(NOGEN-1)540,355,352
352 MJ(2)=J2
    LOJJ2=LOJJ(2,J2)
    HIJJ2=HIJJ(2,J2)
355 JLO1=JLO(1)
    JHI1=JHI(1)
    DO 420 J1=JLO1,JHI1
    MJ(1)=J1
    LOJJ1=LOJJ(1,J1)
    HIJJ1=HIJJ(1,J1)
    C1=0.
    C2=0.
    C3=0.
    C4=0.
    C5=0.
    DEL1=0.
    DEL2=0.
    DEL3=0.
    DEL4=0.
    DEL5=0.
    CST(J1,J2,J3,J4,J5)=1000000000.
360 DO 390 JJ5=LOJJ5,HIJJ5
    IF(NOGEN.LT.5)GO TO 365
    C5=COST(5,J5,JJ5)
    DEL5=DEL(5,J5,JJ5)
365 DO 390 JJ4=LOJJ4,HIJJ4
    IF(NOGEN.LT.4)GO TO 370
    C4=COST(4,J4,JJ4) + C5
    DEL4=DEL(4,J4,JJ4) + DEL5
370 DO 390 JJ3=LOJJ3,HIJJ3
    IF(NOGEN.LT.3)GO TO 375
```

```
      C3=COST(3,J3,JJ3) + C4
      DEL3=DEL(3,J3,JJ3) + DEL4
  375 DO 390 JJ2=LOJJ2,HIJJ2
      IF(NOGEN.LT.2)GO TO 380
      C2=COST(2,J2,JJ2) + C3
      DEL2=DEL(2,J2,JJ2) + DEL3
  380 DO 390 JJ1=LOJJ1,HIJJ1
      C1=COST(1,J1,JJ1) + C2-CORR
      DEL1=DEL(1,J1,JJ1) + DELNNN + DEL2
      CS=C1+GAMMA2*DEL1**2
      IF(KK.EQ.33)GO TO 385
      CS=CS+CCST(JJ1,JJ2,JJ3,JJ4,JJ5)
  385 IF(CS.GE.CST(J1,J2,J3,J4,J5))GO TO 390
      CST(J1,J2,J3,J4,J5)=CS
      TOTCST=CS*H*T/32.0
      JM(1)=JJ1
      JM(2)=JJ2
      JM(3)=JJ3
      JM(4)=JJ4
      JM(5)=JJ5
  390 CONTINUE
      IF(K.EQ.1)GO TO 410
C
C     ENCODE OPTIMAL CONTROLS. LUU=0,1,2 IF U=NEG, ZERO, POS.
C
      DO 395 I=1,5
  395 LUU(I)=1
      DO 400 I=1,NOGEN
      JJM=JM(I)
      MMJ=MJ(I)
  400 LUU(I)=(MW(KK,I,JJM)-MW(K,I,MMJ))/H + 1
      NUM=0
```

```
      DO 405 I=1,5
  405 NUM=NUM+LUU(I)*10**(I-1)
      LUOPT(K,J1,J2,J3,J4,J5)=NUM
      GO TO 420
  410 DO 415 I=1,NOGEN
      JMJ=JM(I)
  415 MWOPT(H+1,I)=MW(H+1,I,JMJ)
  420 CONTINUE
      IF(K.EQ.1)GO TO 455
  425 DO 450 J5=JLO5,JHI5
  430 DO 450 J4=JLO4,JHI4
  435 DO 450 J3=JLO3,JHI3
  440 DO 450 J2=JLO2,JHI2
  445 DO 450 J1=JLO1,JHI1
      CCST(J1,J2,J3,J4,J5)=CST(J1,J2,J3,J4,J5)
  450 CONTINUE
      GO TO 250
C
C     FIND THE OPTIMAL CONTROL CODE (= NUM).
C
  455 K=H+1
      WRITE(6,456)TOTCST
  456 FORMAT('+',15X,F11.2)
  460 KK=K+H
      IF(KK.GE.33)GO TO 525
      K1=1
      K2=1
      K3=1
      K4=1
      K5=1
      DO 495 I=1,NOGEN
      DO 490 J=1,3
```

```
      IF(MWOPT(K,I).NE.MW(K,I,J))GO TO 490
      GO TO(465,470,475,480,485),I
  465 K1=J
      GO TO 495
  470 K2=J
      GO TO 495
  475 K3=J
      GO TO 495
  480 K4=J
      GO TO 495
  485 K5=J
      GO TO 495
  490 CONTINUE
  495 CONTINUE
      NUM=LUOPT(K,K1,K2,K3,K4,K5)
C
C     DECODE OPTIMAL CONTROL VALUES.
C
      DO 515 I=1,5
      MUM=NUM/10
      NN=NUM-10*MUM+1
      NUM=MUM
      GO TO (500,505,510),NN
  500 LU(I)=-H
      GO TO 515
  505 LU(I)=0
      GO TO 515
  510 LU(I)= H
  515 CONTINUE
      DO 520 I=1,NOGEN
  520 MWOPT(KK,I)=MWOPT(K,I)+LU(I)
      K=KK
```

```
      GO TO 460
  525 DO 535 K=1,33,H
      XYZ=CRITMW(K)
      DO 530 I=1,NOGEN
      J=MWOPT(K,I)
      KJ(I)=J
      XY(I)=X(J,I)
  530 XYZ=XYZ+XY(I)
      ZYX(K)=XYZ/200.
  535 WRITE(6,95)K,TK(K),XYZ,(XY(I),KJ(I),I=1,NOGEN),CRITMW(K),KCRIT(K)
   95 FORMAT(' ',T11,I2,3X,F5.2,6X,F7.1,6(F6.1,I3))
      IF(NPASS.LE.4) GO TO 245
      KOUNT=0
      DO 537 K=1,33
      DO 536 I=1,NOGEN
      IF(MW(K,I,2).NE.MWOPT(K,I)) KOUNT=KOUNT+1
  536 CONTINUE
  537 CONTINUE
      WRITE(6,96) KOUNT
   96 FORMAT('0',9X,'OPTIMAL TRAJECTORY DIFFERENT FROM NOMINAL TRAJECTOR
     CY FOR ONLY ',I4,' POINTS.'/)
      IF(KOUNT.LE.10) GO TO 550
  538 IF(NPASS.GE.10)GO TO 550
      GO TO 245
  550 CALL PLOT(1.0,0.5,-3)
      WRITE(6,549)
  549 FORMAT(1H1,'LISTING OF ZIG-ZAG INTERVALS, POSSIBLE MW RATES, THE C
     COST FROM HIEND TO LOEND1 OF EACH POSSIBILITY, AND WHETHER IT IS PL
     COTTED')
      DO 560 I=1,NOGEN
      DO 561 K=1,33
      J=MWOPT(K,I)
```

```
      OP(K,I)=X(J,I)/100.0
      XLO(K)=(LXLO(K,I)-1)/10.0
      XHI(K)=(LXHI(K,I)-1)/10.0
      OPT(K)=(MWOPT(K,I)-1.0)/10.0
      TX(K)=(K-1)/10.
  561 CONTINUE
      CALL PLOTSU
  560 CONTINUE
      CALL PLOTMW
      DO 576 I=1,NOGEN
      DO 577 K=1,33
      J=MWOPT(K,I)
      OP(K,I)=X(J,I)/100.0
      XLO(K)=(LXLO(K,I)-1)/10.0
      XHI(K)=(LXHI(K,I)-1)/10.0
      OPT(K)=(MWOPT(K,I)-1.0)/10.0
  577 CONTINUE
C
C     FIND THE ZIG-ZAG INTERVAL AND REPLACE IT BY STRAIGHT LINE.
C
      KOUNT2=4
      KOUNT3=0
      KOUNT4=0
      K=1
  551 K=K+1
      KK=K+1
      KKK=K-1
      IF((MWOPT(K,I)-MWOPT(KKK,I)).EQ.(MWOPT(KK,I)-MWOPT(K,I))) GO TO
     C558
      IF(KOUNT4.EQ.1) GO TO 555
      KOUNT3=KOUNT3+1
      KOUNT4=1
```

```
552 IF(KOUNT2.LE.3) GO TO 556
553 LOEND=K
    LOEND1=KKK
    LOEND2=KK
554 KOUNT2=1
    GO TO 580
555 KOUNT4=0
    GO TO 552
556 IF(KOUNT4.EQ.0) GO TO 557
    IF((MWOPT(KK,I)-MWOPT(K,I)).NE.(MWOPT(LOEND2,I)-MWOPT(LOEND,I)))
   CGO TO 571
    INTVL2=KOUNT2
    GO TO 554
557 IF((MWOPT(KK,I)-MWOPT(K,I)).NE.(MWOPT(LOEND,I)-MWOPT(LOEND1,I)))
   CGO TO 571
    INTVL1=KOUNT2
    GO TO 554
558 IF(KOUNT3.EQ.0) GO TO 580
    KOUNT2=KOUNT2+1
    IF(KOUNT2.LE.3) GO TO 580
    IF(KOUNT3.GE.2) GO TO 574
559 KOUNT4=0
    KOUNT3=0
    GO TO 580
565 IF(KOUNT4.EQ.0) GO TO 573
    HIEND=K-INTVL2
570 CALL SINGUL
    KOUNT2=4
    GO TO 559
573 HIEND=K-INTVL1
    GO TO 570
571 IF(KOUNT3.GE.2) GO TO 565
```

```
      KOUNT4=1
      GO TO 553
  572 HIEND=K-3+INTVL2
      HIEND1=HIEND-INTVL2
      GO TO 570
  574 IF(KOUNT4.EQ.0) GO TO 572
      HIEND=K-3+INTVL1
      HIEND1=HIEND-INTVL1
      GO TO 570
  580 IF(K.LT.32) GO TO 551
      IF(KOUNT3.LT.2) GO TO 575
      HIEND=33
      IF(KOUNT4.EQ.0) GO TO 581
      HIEND1=32-INTVL1
      GO TO 582
  581 HIEND1=32-INTVL2
  582 CALL SINGUL
  575 CALL PLOTSU
  576 CONTINUE
      CALL PLOTMW
      CALL PLOT(0.0,0.0,-4)
  540 WRITE(6,545)
  545 FORMAT('1')
      STOP
      END
```

```
      SUBROUTINE PLOTSU
C
C     PLOT THE FRAME
C
      IMPLICIT INTEGER*4(H-N)
      DIMENSION OPT(33),TX(33),XHI(33),XLO(33)
      COMMON/BLOK1/ TX,XHI,XLO,OPT
      CALL AXIS(0.0,0.0,'TIME',-4,3.2,0.0,0.0,10.0)
      CALL PLOT(0.0,0.0,3)
      CALL PLOT(0.8,0.0,2)
      DO 566 K=1,3
      CALL PLOT(0.8,0.05,2)
      CALL PLOT(0.8,0.0,-3)
  566 CALL PLOT(0.8,0.0,2)
      CALL PLOT(0.8,0.8,2)
      CALL PLOT(0.75,0.8,2)
      CALL PLOT(0.8,0.8,-3)
      CALL PLOT(0.0,0.8,2)
      DO 567 K=1,2
      CALL PLOT(-0.05,0.8,2)
      CALL PLOT(0.0,0.8,-3)
  567 CALL PLOT(0.0,0.8,2)
      CALL PLOT (-0.8,0.8,2)
      CALL PLOT (-0.8,0.75,2)
      CALL PLOT(-0.8,0.8,-3)
      CALL PLOT(-0.8,0.0,2)
      DO 569 K=1,2
      CALL PLOT(-0.8,-0.05,2)
      CALL PLOT(-0.8,0.0,-3)
  569 CALL PLOT(-0.8,0.0,2)
      CALL PLOT(-0.8,-3.2,-3)
      CALL AXIS(0.0,0.0,'MW',2,3.2,90.0,0.0,10.0)
```

```
      CALL PLOT(0.0,0.0,3)
      CALL PLOT(0.0,0.8,3)
      DO 568 K=1,3
      CALL PLOT(0.05,0.8,2)
      CALL PLOT(0.0,0.8,-3)
  568 CALL PLOT(0.0,0.8,2)
      CALL PLOT(0.0,-2.4,-3)
C
C     PLOT THE FUNCTION
C
      CALL LINE(TX,XHI,33,-1)
      CALL PLOT (0.0,0.0,3)
      CALL LINE(TX,XLO,33,-1)
      CALL PLOT (0.0,0.0,3)
      CALL LINE(TX,OPT,33,-1)
      CALL PLOT(0.0,0.0,3)
      CALL PLOT(5.0,0.0,-3)
      RETURN
      END
```

```
      SUBROUTINE PLOTMW
      IMPLICIT INTEGER*4(H-N)
      DIMENSION OPT(33),TX(33),XHI(33),XLO(33)
      DIMENSION OP(33,5),ZYX(33),CRITMW(33)
      COMMON/BLOK1/ TX,XHI,XLO,OPT
      COMMON/BLOK2/ NOGEN,OP,CRITMW,ZYX
      CALL AXIS (0.0,0.0,'TIME',-4,4.0,0.0,0.0,10.0)
      CALL AXIS (0.0,0.0,'MW',2,5.0,90.0,0.0,100.0)
      CALL PLOT (3.2,0.0,3)
      CALL PLOT(3.2,5.0,2)
      CALL PLOT(0.0,0.0,3)
      DO 562 I=1,NOGEN
      DO 563 K=1,33
      OPT(K)=OP(K,I)
  563 CONTINUE
      CALL LINE(TX,OPT,33,-1)
  562 CALL PLOT(0.0,0.0,3)
      DO 564 K=1,33
  564 CRITMW(K)=CRITMW(K)/100.0
      CALL LINE(TX,CRITMW,33,-1)
      CALL PLOT(0.0,0.0,3)
      DO 565 K=1,33
  565 CRITMW(K)=CRITMW(K)*100.0
      CALL PLOT(5.0,0.0,-3)
      CALL AXIS (0.0,0.0,'TIME',-4,4.0,0.0,0.0,10.0)
      CALL AXIS (0.0,0.0,'MW',2,8.0,90.0,    0.0,200.0)
      CALL PLOT(0.0,0.0,3)
      CALL LINE(TX,ZYX,33,-1)
      CALL PLOT(3.2,0.0,3)
      CALL PLOT(3.2,8.0,2)
      CALL PLOT(5.0,0.0,-3)
      RETURN
```

```
      SUBROUTINE SINGUL
      IMPLICIT INTEGER*4(H-N)
      DIMENSION HIT(5),LOT(5)
      DIMENSION NUMV(6),XO(6),UMAX(6),XMIN(6),XMAX(6),BETA(6),BTU(7,6),
     *          P(7,6)
      DIMENSION SLP(6,6),ASLP(36),LEV(36),IGEN(36),XT(6)
      DIMENSION XY(5),YX(5),UMX(5),DELX(5),COST3(5)
      DIMENSION NUMOLD(5),LXO(5),LXT(5),LXMIN(5),LXMAX(5),JLO(5),JHI(5)
      DIMENSION CT(5),DL1(5),COS2(5),COS3(5)
      DIMENSION OPT(33),TX(33),XHI(33),XLO(33)
      DIMENSION US(33,5),COS4(33,5),XS(33,5)                                C
C
      DIMENSION X(33,5),COST4(33,5),LXHI(33,5),LXLO(33,5),MWOPT(33,5)       C
      DIMENSION OP(33,5),ZYX(33),CRITMW(33)                                 C
C
      COMMON/BLOK1/ TX,XHI,XLO,OPT
      COMMON/BLOK2/ NOGEN,OP,CRITMW,ZYX
      COMMON/BLOK3/ I,HIEND,LOEND,US,MWOPT,HIEND1,LOEND1,UMX,XS,X,
     *LXT,COS4,COST4,CT,DL2,DELX,DL1,COS2,GAMMA2,NUMOLD,COS3,DELTAT,
     *BETA,GAMMA3,P,NUMV,BTU,SLP,GAMMA4,SCT,ISG,IEND,INTVL2
C
      LOEND1=LOEND-INTVL2
      HIT(1)=HIEND1
      HIT(2)=HIEND1
      HIT(3)=HIEND
      HIT(4)=HIEND
      HIT(5)=HIEND
      LOT(1)=LOEND
      LOT(2)=LOEND1
      LOT(3)=LOEND
      LOT(4)=LOEND1
      LOT(5)=LOEND1
```

```
      IEND=HIEND-1
      DO 2 K=LOEND1,IEND
      KK=K+1
      US(K,1)=FLOAT (MWOPT(HIEND1,I)-MWOPT(LOEND,I))/FLOAT(HIEND1-LOEND)
     C  *DELX(I)
      US(K,2)=FLOAT (MWOPT(HIEND1,I)-MWOPT(LOEND1,I))/FLOAT(HIEND1-LOEND
     C1)*DELX(I)
      US(K,3)=FLOAT (MWOPT(HIEND ,I)-MWOPT(LOEND,I))/FLOAT(HIEND -LOEND)
     C  *DELX(I)
      US(K,4)=FLOAT (MWOPT(HIEND ,I)-MWOPT(LOEND1,I))/FLOAT(HIEND -LOEND
     C1)*DELX(I)
      US(K,5)=FLOAT(MWOPT(KK,I)-MWOPT(K,I))*DELX(I)
      IF(K.GE.LOEND) GO TO 1
      US(K,1)=US(K,5)
      US(K,3)=US(K,5)
    1 IF(K.LT.HIEND1) GO TO 2
      US(K,1)=US(K,5)
      US(K,2)=US(K,5)
    2 CONTINUE
      J=MWOPT(LOEND1,I)
      K=LXT(I)
      XLM=X(K,I)
      DO 3 JJ=1,5
      XS(33,JJ)=XLM
      XS(LOEND1,JJ)=X(J,I)
      COS4(LOEND1,JJ)=COST4(J,I)
      CT(JJ)=0.0
      DO 3 K=LOEND1,IEND
      KK=K+1
    3 XS(KK,JJ)=XS(K,JJ)+US(K,JJ)
      DO 8 K=LOEND1,IEND
      KK=K+1
```

```
      DL2=0.0
      DO 4 II=1,NOGEN
      IF(II.EQ.I) GO TO 4
      MMM=MWOPT(K,II)
      MM=MWOPT(KK,II)-MMM
      DL2=(2*(MMM-LXT(II))+MM)*DELX(II)/2.0+DL2
    4 CONTINUE
      II=NUMOLD(I)
      DO 7 J=1,5
      DL1(J)=DL2+(XS(K,J)+XS(KK,J))/2.0-XS(33,J)+(CRITMW(K)+CRITMW(KK))/
     C2.0-CRITMW(33)
      COS2(J)=GAMMA2*DL1(J)**2
      COS3(J)=BETA(II)*US(K,J)*GAMMA3/DELTAT
      COS3(J)=ABS(COS3(J))
      L=1
    5 IF(XS(KK,J).LT.(P(L+1,II)+0.2)) GO TO 6
      IF(L.GE.NUMV(II)) GO TO 6
      L=L+1
      GO TO 5
    6 COS4(KK,J)=(BTU(L,II)+(XS(KK,J)-P(L,II))*SLP(L,II))*GAMMA4
    7 CT(J)=(COS4(K,J)+COS4(KK,J))/2.0+COS3(J)+COS2(J)+CT(J)
    8 CONTINUE
      DO 16 J=1,5
   16 CT(J)=CT(J)*DELTAT
      ISG=1
      SCT=CT(1)
      DO 9 J=2,4
      IF(CT(J).GT.SCT) GO TO 9
      ISG=J
      SCT=CT(J)
    9 CONTINUE
      WRITE(6,10)
```

```
10 FORMAT(1H0,3X,'GEN',3X,'FROM',5X,'TO',6X,'J',3X,'MW RATE',7X,'COST
  C',5X,'PLOT')
   DO 14 J=1,5
   INDEXL=LOT(J)
   INDEXH=HIT(J)
   IF(J.EQ.ISG) GO TO 12
   WRITE(6,11)II,LOT(J),MWOPT(INDEXL,I),HIT(J),MWOPT(INDEXH,I),J,
  CUS(LOEND,J),CT(J)
11 FORMAT(4X,I2,2X,'(',I2,',',I2,')  (',I2,',',I2,')',3X,I1,3X,F6.3,
  C3X,F11.4,3X,'NO')
   GO TO 14
12 WRITE(6,13)II,LOT(J),MWOPT(INDEXL,I),HIT(J),MWOPT(INDEXH,I),J,
  CUS(LOEND,J),CT(J)
13 FORMAT(4X,I2,2X,'(',I2,',',I2,')  (',I2,',',I2,')',3X,I1,3X,F6.3,
  C3X,F11.4,3X,'YES PLOT')
14 CONTINUE
   DO 15 K=LOEND1,IEND
   KK=K+1
   OP(K,I)=XS(K,ISG)/200.0
   L=MWOPT(K,I)
   ZYX(K)=ZYX(K)+OP(K,I)-X(L,I)/200.0
   OP(K,I)=OP(K,I)*2
15 OPT(KK)=OPT(K)+US(K,ISG)/DELX(I)/10.0
   RETURN
   END
```

```
 1400.0
100.0 20.0200.0  5.0  6.0  3.0
 20.0 10.0 85.0 22.0150.0 40.0200.0 70.0
250.0 25.0300.0  6.0  6.0  4.0
 25.0 15.0105.0 25.0175.0 42.0240.0 62.0300.0 90.0
 80.0 30.0280.0  7.0  5.0  3.0
 30.0 15.0110.0 27.0200.0 46.0280.0 75.0
200.0 25.0250.0  9.0  6.0  3.0
 25.0 20.0125.0 30.0200.0 50.0250.0 80.0
170.0 35.0320.0  8.0  7.0  3.0
 35.0 20.0140.0 30.0230.0 60.0320.0100.0
250.0 50.0350.0 10.0  6.0  4.0
 50.0 15.0150.0 35.0225.0 57.0300.0 67.0350.0115.0
  2.0  5.0 10.0
```

The vita has been removed from
the scanned document

DYNAMIC ECONOMIC DISPATCH OF POWER SYSTEMS
BY MULTI-PASS DYNAMIC PROGRAMMING

by

Nanming Chen

(ABSTRACT)

The purpose of this research is to improve area Automatic Generation Control (AGC) by using the Multi-Pass Dynamic Programming developed in this research.

The AGC of power systems coordinates the Load Frequency Control problem and the Economic Dispatch problem together to form a dynamic optimal control problem. A power system was partitioned into the Electric Power Network subsystem and the Mechanical Power Control subsystem. Earlier work has solved the Electric Network subsystem control problem, used analytical methods to find optimal trajectories and controls for the Mechanical Power Control subsystem, but only for the limited case of two generators.

This research develops the multi-pass dynamic programming, checks convergence, derives the conditions for singular solutions and provides optimal control sequences and optimal trajectories for cases involving several generators. Parameter sensitivity is also studied here.

Finally, some consideration is given to the comparison of multi-pass dynamic programming and conventional dynamic programming.