CHAPTER III

ESTIMATION AS A TWO POINT BOUNDARY VALUE PROBLEM AND ITS SOLUTION BY STEEPEST DESCENT METHOD

This chapter covers the formulation of estimation problem as a TPBV problem and its numerical solution by using steepest method.

3.1 Problem Statement

The problem to be considered is the estimation of initial states and parameters of a physical system(in general, nonlinear) from the sampled record of its input and output. The formulation of the problem will cover in its general framework, the problem of estimation of states and parameters of turbo-alternator transfer functions from the normal input-output record. It is assumed that the form of the transfer function and hence the differential equation description of the system is known. In partcular, let the system be described by a vector differential equation of the following form

$$\dot{x}(t) = g(x(t) , t) \qquad t_o \leq t \leq t_N \qquad (3.1)$$

The discrete-time formulation is more appropriate if the input-output record is in sampled form and if a digital computer is to be used to solve the problem. Let the discrete-time equivalent of equation (3.1) be

$$x(i + 1) = f(x(i) , i) \qquad i = 0, 1, \ . \ . \ ,N \qquad (3.2)$$

All inputs to the system are assumed to be known(i.e. measured) and are accounted for by the explicit dependance of $f(x(i) , i)$ on the time parameter i . It is further assumed that the output is a linear combination of the unobservable states and is corrupted by additive noise which may be due to random disturbances

inherent in the system and imperfect measurements. The observed output signal is

$$y(i) = H\,x(i) + n(i) \qquad i = 0, 1, \ldots, N \qquad (3.3)$$

where

x is an n-dimensional state vector augmented to include unknown constant parameters that must be estimated,

g and f are n-dimentional vector-valued functions,

y is an m-dimensional observation vector,

H is an m x n constant matrix, and

n is an m-dimensional noise vector.

The vectors x, y, g and f are given by

$$x(i) = \mathrm{col}\left[x_1(i), x_2(i), \ldots, x_n(i)\right] \qquad (3.4)$$

$$y(i) = \mathrm{col}\left[y_1(i), y_2(i), \ldots, y_m(i)\right] \qquad (3.5)$$

$$g = \mathrm{col}\left[g_1, g_2, \ldots, g_n\right] \qquad (3.6)$$

$$f = \mathrm{col}\left[f_1, f_2, \ldots, f_n\right] \qquad (3.7)$$

where "col" denotes the column vector.

Having observed a finite sequence $\{y(0), y(1), \ldots, y(N)\}$, the problem is to find the best estimate of the initial state x(0). This will be obtained in the least squares sense.

3.2 Estimation as a TPBV Problem

Let the physical system be simulated on the digital computer by a dynamic mathematical model similar to that in equation (3.2) and is given by

$$\bar{x}(i+1) = f(\bar{x}(i), i) \qquad i = 0, 1, \ldots, N \qquad (3.8)$$

where $\bar{x}(i)$ is an n-vector and represents a nominal trajectory for the same input as that measured on the physical system and for some initial condition $\bar{x}(0)$. The computed output for the simulated system is

$$\bar{y}(i) = H\,\bar{x}(i) \tag{3.9}$$

The deviation of the observed output $y(i)$ and the simulated model output $\bar{y}(i)$ is given by the residual error

$$\begin{aligned} e(i) &= y(i) - \bar{y}(i) \\ &= y(i) - H\,\bar{x}(i) \qquad i = 0, 1, \ldots, N \end{aligned} \tag{3.10}$$

If the choice of $\bar{x}(0)$ were correct, the corresponding trajectory $\bar{x}(i)$ obtained from equation (3.8) would be the true one and the residual error $e(i)$ would be either zero for no additive noise at the output or as small as possible when noise is present. In other words, when the residual error is minimum, the initial state $\bar{x}(0)$ would be as close as possible to the true one. The best possible estimate of $\bar{x}(0)$ will be obtained in the least squares sense by minimizing the following functional of the errors.

$$\begin{aligned} I &= \sum_{i=0}^{N} [e(i)]' Q [e(i)] \\ &= \sum_{i=0}^{N} [y(i) - H\,\bar{x}(i)]' Q [y(c) - H\,\bar{x}(i)] \end{aligned} \tag{3.11}$$

where the "prime" on the vector or matrix demotes its transpose and Q is a symmetrical positive $m \times m$ matrix representing the weighting factor.

The nominal trajector $\bar{x}(i)$ in equation (3.11) follows from equation (3.8), given $\bar{x}(0)$. Thus the minimization of functional I

in equation (3.11) constrained by equation (3.8) is equivalent to minimizing the following functional[2]

$$I^* = \sum_{i=0}^{N} \left[y(i) - H\,\bar{x}(i)\right]' Q \left[y(i) - H\,\bar{x}(i)\right] + \lambda'(i)\left[\bar{x}(i+1) - f(\bar{x}(i), i)\right] \tag{3.12}$$

where $\lambda(i)$ is an n-dimensional vector whose elements are the Lagrange multipliers and is given by

$$\lambda(i) = \text{col}\left[\lambda_1(i), \lambda_2(i), \ldots, \lambda_n(i)\right] \tag{3.13}$$

The necessary conditions for the minimization of equation (3.12) is obtained by determining the differential of I^* and setting the coefficients of all independent differentials equal to zero. The differential of I^* is given by

$$\Delta I^* = \sum_{i=0}^{N} \left\{-2\left[y(i) - H\,\bar{x}(i)\right] Q\, H\, \Delta\bar{x}(i) + \Delta\lambda'(i)\left[\bar{x}(i+1) - f(\bar{x}(i), i)\right] + \lambda'(i)\left[\Delta\bar{x}(i+1) - f_{\bar{x}(i)}\,\Delta\bar{x}(i)\right]\right\} \tag{3.14}$$

where $f_{\bar{x}(i)}$ is an n x n Jacobian matrix whose $(j, k)^{th}$ element $f^{jk}_{\bar{x}(i)}$ is given by

$$f^{jk}_{\bar{x}(i)} = \frac{\Delta f_j(\bar{x}(i), i)}{\Delta \bar{x}_k(i)} = \frac{\Delta \bar{x}_j(i+1)}{\Delta \bar{x}_k(i)} \qquad \text{(for } \Delta \text{ sufficiently small)} \tag{3.15}$$

Rearranging the tersm on the right hand side of equation (3.13)

$$\Delta I^* = \sum_{i=0}^{N} \left\{-2\left[y(i) - H\,\bar{x}(i)\right] Q\, H - \lambda'(i)\, f_{\bar{x}(i)} + \lambda(i-1)\right\} \Delta\bar{x}(i)$$

$$+ \sum_{i=0}^{N} \Delta\lambda'(i)\{\bar{x}(i+1) - f(\bar{x}(i), i)\}$$

$$- \{\lambda'(i-1)\}\Delta\bar{x}(0) + \{\lambda'(N)\}\Delta\bar{x}(N+1) \qquad (3.16)$$

In equation (3.16), the terms within the curlie brackets are the coefficients of independent differentials. Setting these equal to zero, the Euler-Lagrange difference equations are obtained. They are

$$\bar{x}(i+1) = f(\bar{x}(i), i) \qquad i = 0, 1, \ldots, N \qquad (3.17)$$

and

$$\lambda(i-1) = f'_{\bar{x}(i)} \lambda(i) + 2 H'Q[y(i) - H\bar{x}(i)] \qquad i = 0, 1, \ldots, N \qquad (3.18)$$

with end conditions

$$\lambda(-1) = 0 \qquad (3.19)$$

$$\lambda(N) = 0 \qquad (3.20)$$

The end conditions of equations (3.19) and (3.20) are the natural boundary conditions that must be satisfied since the both ends $\bar{x}(0)$ and $\bar{x}(N+1)$ are free. Also the Euler-Lagrange equations (3.17) and (3.18) must be satisfied for minimization of the functional I^*. Thus the problem of estimating the initial state $\bar{x}(0)$ which also includes the constant parameters is reduced to solving for $\bar{x}(i)$ satisfying equations (3.17), (3.18), (3.19) and (3.20). This is a usual Two Point Boundary Value problem. Since all the subsequent states $\bar{x}(i)$; $i = 1, 2, \ldots, N$ depend on the initial state $\bar{x}(0)$ by virtue of equation (3.17), the solution of the TPBV problem requires to choose suitable $\bar{x}(0)$. The $\bar{x}(i)$

trajectory obtained from equation (3.17) for this $\bar{x}(0)$, when substituted in equation (3.18), satisfies the end conditions for the Lagrange multipliers λ's, given by equations (3.19) and (3.20).

The exact analytical solution to such problems is not known and one has therefore to solve the problem using iterative techniques. Cox[2] solved the TPBV problem by using dynamic programming considering the gaussian noise. Sridhar and Detchmandy[3] obtained solution of the problem in continuous-time employing invariant-imbedding technique without assuming any statistical description of noise. Both the techniques are of sequential nature. Pearson[4] found a discrete-time analog for the same. The merits and demerits of the sequential scheme[3,4] are discussed in section 2.5C as evident from the results shown in Appendix A. In view of this, the technique developed here is of nonsequential type to suit the problem of estimating turbo-alternator transfer functions from the observations heavily corrupted with noise. In the sequential scheme, one begins with some arbitrary guess on $\bar{x}(0)$ and the modifies successively the values of $\bar{x}(0)$ on each iteration using the steepest descent rule until the performance index I given by equation (3.11) is minimized. The minimum of I means that the best fit of the simulated model output $\bar{y}$ on the observed output y is obtained in the least squares sense. The principle of steepest descent method[97] is explained in the following section.

3.3 Solution of the TPBV Problem by Steepest Descent Method

For some initial arbitrary choice on

$$\bar{x}(0) = \text{col}[\bar{x}_1(0), \bar{x}_2(0), \ . \ . \ , \bar{x}_n(0)] \qquad (3.21)$$

and for the given input, suppose that the nominal trajectory $\bar{x}(i)$ is computed using equation (3.17). The $\lambda(i)$ trajectory obtained by substitution of these $\bar{x}(i)$ values in equation (3.18) may not satisfy simultaneously the boundary conditions given by equations (3.19) and (3.20). In other words, if one obtains the $\lambda(i)$ trajectory by solving the equation (3.18) backwards starting with $\{\lambda(N) = 0\}$, one may not end up with the other necessary condition $\{\lambda(-1) = 0\}$ simultaneously. This indicates that the assumed values of the elements of $\bar{x}(0)$ are not the true solutions of the TPBV problem implying that the functional I^* is equation (3.13) is not minimized. If the choice of $\bar{x}(0)$ were true, both the boundary conditions are satisfied simultaneously. This is illustrated by the plots of $\bar{x}_1(i)$ and $\lambda_1(i)$ in Fig. 3.1. If these computed values for $\bar{x}(i)$ and $\lambda(i)$ are substituted in equation (3.16), it gives

$$\Delta I^* = -\lambda'(-1)\,\Delta\bar{x}(0) \tag{3.22}$$

The differential ΔI^* has a nonzero value so long as the choice of $\bar{x}(0)$ is not true and approaches zero as $\bar{x}(0)$ approaches the true value. This fact can be used to modify the initial guess $\bar{x}(0)$ towards its true value.

Since the trajectories $\bar{x}(i)$ and $\lambda(i)$ depend on the initial state $\bar{x}(0)$, it follows from equation (3.12) that I^* is a function of $\bar{x}(0)$, i.e.

$$I^* = I^*[\bar{x}(0)] \tag{3.23}$$

The differential of I^* is given by

$$\Delta I^* = I^*[\bar{x}(0) + \Delta\bar{x}(0)] - I^*[\bar{x}(0)]$$

$$= \left[\nabla_{\bar{x}(0)} I^*\right]' \Delta\bar{x}(0) \qquad (3.24)$$

where

$$\nabla_{\bar{x}(0)} I^* = \text{col}\left[\frac{\Delta I}{\Delta\bar{x}_1(0)}, \frac{\Delta I}{\Delta\bar{x}_2(0)}, \ . \ . \ , \frac{\Delta I}{\Delta\bar{x}_n(0)}\right] \qquad (3.25)$$
$$(3.2$$

Comparison of equations (3.22) and (3.24) yields

$$\nabla_{\bar{x}(0)} I^* = -\lambda(-1) \qquad (3.26)$$

Here, the vector $\nabla_{\bar{x}(0)} I^*$ and hence $-\lambda(-1)$ represents a gradient vector at a point $\bar{x}(0)$ on the surface I^* in the (n+1) dimensional space formed by plotting I^* versus $\bar{x}_1(0)$, $\bar{x}_2(0)$, . . , $\bar{x}_n(0)$. It is difficult to visualize a space of dimension greater than three. The steepest descent[97] method will be explained here with $\bar{x}(0)$ as a two-dimensional vector, i.e.

$$\bar{x}(0) = \text{col}\left[\bar{x}_1(0), \bar{x}_2(0)\right] \qquad (3.27)$$

Fig. 3.2 depicts a plot of functional I^* versus $\bar{x}_1(0)$ and $\bar{x}_2(0)$. In fact, it is a projection, on $\{\bar{x}_1(0), \bar{x}_2(0)\}$ plane, of a three-dimensional surface obtained by plotting I^* against $\bar{x}_1(0)$ and $\bar{x}_2(0)$. The functional I^* is shown to be minimum at Q for one particular set of values $\{\bar{x}_1(0), \bar{x}_2(0)\}$. For any other set of values, the functional I^* is always higher than the minimum. The closed curves are contours of constant I^* possible for different sets of values $\{\bar{x}_1(0), \bar{x}_2(0)\}$. One can approach the minima taking small steps, $\Delta\bar{x}(0)$, along the negative gradient starting from some arbitrary point such as P. Let this small step $\Delta\bar{x}(0)$ be proportional to negative of the gradient difined in equation (3.26).

Thus

$$\Delta\bar{x}(0) = k\{-[-\lambda(-1)]\} \quad ; k>0$$
$$= k\,\lambda(-1) \tag{3.28}$$

Substituting for $\Delta\bar{x}(0)$ from equation (3.28) in equation (3.22), one obtains

$$\Delta I^* = -k[\lambda^{-1}(-1)\,\lambda(-1)] \tag{3.29}$$

where k is to be chosen suitably. The steepest descent method also involves choosing the best length of the step along the gradient. The length ΔA of the chosen step $\Delta\bar{x}(0)$ is given by

$$\Delta A = \sqrt{[\Delta\bar{x}(0)]'[\Delta\bar{x}(0)]} \tag{3.30}$$

which is equivalent to for a n-dimensional case

$$\Delta A = \sqrt{\Delta\bar{x}_1(0)^2 + \Delta\bar{x}_2(0)^2 + \ldots + \Delta\bar{x}_n(0)^2} \tag{3.31}$$

The corresponding variation in performance index is given by equation (3.22). It is desired to obtain the minima of I^* with the minimum number of iterations to save computer time. The length ΔA of the step should therefore be so chosen that the variation ΔI^* in the performance index I^* is maximum negative, i.e. minimum. Minimization of ΔI^* in equation (3.22) with the constraint given by equation (3.30) is equivalent to the minimization of the function Ψ given by

$$\Psi = -\lambda'(-1)\,\Delta\bar{x}(0) + \beta\{\Delta A^2 - [\Delta\bar{x}(0)]'[\Delta\bar{x}(0)]\} \tag{3.32}$$

where β is the Lagrange multiplier. Differentiating this with respect to $\Delta\bar{x}(0)$, one obtains

$$\frac{\partial \bar{\Psi}}{\delta [\Delta \bar{x}(0)]} = -\lambda(-1) - 2\beta \Delta \bar{x}(0) \qquad (3.33)$$

where

$$\frac{\partial \Psi}{\delta [\Delta \bar{x}(0)]} = \text{col}\left[\frac{\partial \Psi}{\delta [\Delta \bar{x}_1(0)]}, \frac{\delta \Psi}{\delta [\Delta \bar{x}_2(0)]}, \ldots, \frac{\delta \Psi}{\delta [\Delta \bar{x}_n(0)]}\right], \qquad (3.34)$$

Equating this to null vector and rearranging, one gets

$$\Delta \bar{x}(0) = -\lambda(-1)/2\beta \qquad (3.35)$$

Substitution for $\Delta \bar{x}(0)$ from equation (3.35) in equation (3.30) yields

$$\Delta A = \pm\sqrt{[\lambda'(-1)\,\lambda(-1)]}/2\beta \qquad (3.36)$$

Therefore, for minimizing ΔI^*, it is necessary that

$$\beta = \pm\sqrt{\lambda'(-1)\lambda(-1)}/2\Delta A \qquad (3.37)$$

Substitution for β from equation (3.37) in equation (3.35) gives

$$\Delta \bar{x}(0) = \pm \Delta A \frac{\lambda(-1)}{\sqrt{\lambda'(-1)\,\lambda(-1)}} \qquad (3.38)$$

Comparison of this with the equation (3.29) suggests positive sign in equation (3.38)

$$\Delta \bar{x}(0) = +\frac{\Delta A}{\sqrt{\lambda'(-1)\,\lambda(-1)}}\,\lambda(-1) \qquad (3.39)$$

and

$$k = \frac{\Delta A}{\sqrt{\lambda'(-1)\,\lambda(-1)}} \qquad (3.40)$$

Thus the optimum step $\Delta \bar{x}(0)$ is obtained.

3.4 Computational Procedure

The theory developed in the forgoing sections is to be used to estimate the transfer functions of the turbo-alternator. The form of the transfer function is assumed to be known. The transfer function with its unknown parameters can easily be transformed into the vector differential equation of the form given by equation (3.1). In order that the estimation scheme developed here could be applied directly, the differential equation must be converted to the difference equation of the form given by equation (3.17) to obtain the numerical values of the state vector $\bar{x}(i)$ and the Jacobian matrix $f_{\bar{x}(i)}$. It is not always easy to obtain the discrete equivalent of the differential equation. Alternatively, the state vector $\bar{x}(i)$ at discrete instants $i = 0, 1, \ldots, N$ may be obtained by numerical integration of the differential equation, similar to the equation (3.1), given by

$$\dot{\bar{x}}(t) = g(\bar{x}(t), t) \qquad t_o \leq t \leq t_N \tag{3.41}$$

where $\bar{x}(t)$ is the nominal trajectory in continuous-time. And the Jacobian matrix $f_{\bar{x}(i)}$ required in equation (3.18) may be obtained in the following manner. Let the state vector $\bar{x}$ (i.e. $\bar{x}(t)$) be varied by the amount $\Delta\bar{x}$ in the equation (3.41). Then, writing $\frac{d\bar{x}}{dt}$ for $\dot{\bar{x}}$, one obtains

$$\frac{d}{dt}(\bar{x} + \Delta\bar{x}) = g(\bar{x} + \Delta\bar{x}, t) \tag{3.42}$$

which simplifies to

$$\dot{\bar{x}} + \Delta\dot{\bar{x}} \simeq g(\bar{x}, t) + g_{\bar{x}}\Delta\bar{x} \qquad \text{(neglecting other terms)} \tag{3.43}$$

where $g_{\bar{x}}$ is the Jacobian matrix in continuous-time. Use of equation (3.41) in equation (3.42) yields

$$\Delta\dot{\bar{x}} = g_{\bar{x}} \Delta\bar{x} \tag{3.44}$$

The solution of such a equation[1] for a discrete-time case is

$$\Delta\bar{x}(i+1) = \Phi(i+1 , i)\, \Delta\bar{x}(i) \tag{3.45}$$

where $\Phi(i+1, i)$ is a state transition matrix for transition of the state $\bar{x}$ at the i^{th} instant to $(i+1)^{st}$. By virtue of the property of state transition matrix, $\Phi(i+1 , i)$ can be obtained from the following equation

$$\dot{\Phi}(i+1 , i) = g_{\bar{x}}\, \Phi(i+1 , i) \quad ; \quad \Phi(i , i) = I \tag{3.46}$$

where "I" is the identity matrix. Comparing equations (3.15) and (3.45), there results

$$f_{\bar{x}(i)} = \Phi(i+1 , i) \tag{3.47}$$

Thus the Jacobian matrix is obtained. The entire computational procedure is briefly summarized as follows:

1. Make an initial guess on the initial condition $\bar{x}(0)$.
2. Obtain the values of the nominal trajectory $\bar{x}(i)$, $i = 0, 1, \ldots, N$ from either the difference equation (3.17), if known or from the differential equation (3.41) by numerical integration between the sample points. Also obtain the Jacobian matrix $f_{\bar{x}(i)}$ at sample points $i = 0, 1, \ldots, N$ either from equation (3.15) or from equation (3.47). Store these values.
3. Compute the performance index I using equation (3.11).

$Q \cong I$.

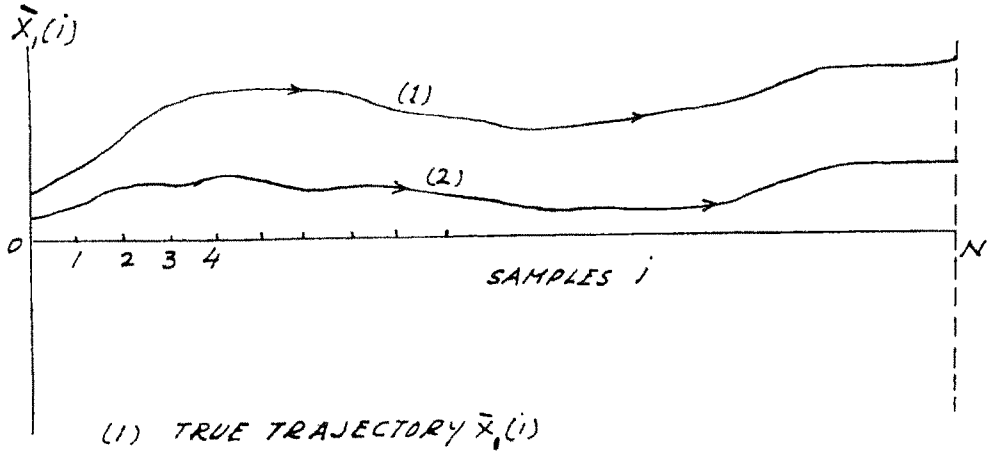
4. Use the values computed in (2) above to solve equation (3.18) backwards starting with $\lambda(N) = 0$ and compute $\lambda(-1)$.

5. Modify the values of the initial condition $\bar{x}(0)$ using equation (3.39) in the following manner

$$\text{new } \bar{x}(0) = \text{old } \bar{x}(0) + \frac{\Delta A}{\sqrt{\lambda(-1)\ \lambda(-1)}}\ \lambda(-1) \quad (3.48)$$

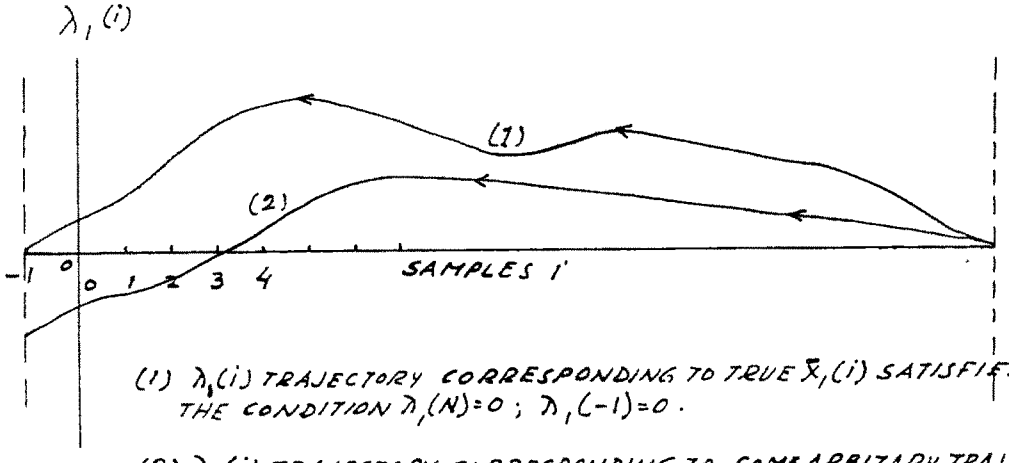
where ΔA, representing the step-size along the negative gradient, may be chosen suitably as will be discussed lateron.

6. Repeat this procedure until I becomes minimum in which case the initial state $\bar{x}(0)$ is expected to have converged as close as possible to their true values.



(1) TRUE TRAJECTORY $\bar{X}_1(i)$

(2) NOMINAL TRAJECTORY $\bar{X}_1(i)$ CORRESPONDING TO ARBITARY INITIAL CONDITION $\bar{X}_1(0)$.



(1) $\lambda_1(i)$ TRAJECTORY CORRESPONDING TO TRUE $\bar{X}_1(i)$ SATISFIES THE CONDITION $\lambda_1(N)=0$; $\lambda_1(-1)=0$.

(2) $\lambda_1(i)$ TRAJECTORY, CORRESPONDING TO SOME ARBITARY TRAJECTORY $\bar{X}_1(i)$, DOES NOT SATISFY BOTH BOUNDARY CONDITIONS SIMULTANEOUSLY.

FIG. 3·1 ILLUSTRATION OF $\bar{X}$ AND λ TRAJECTORIES AND THE BOUNDARY CONDITIONS

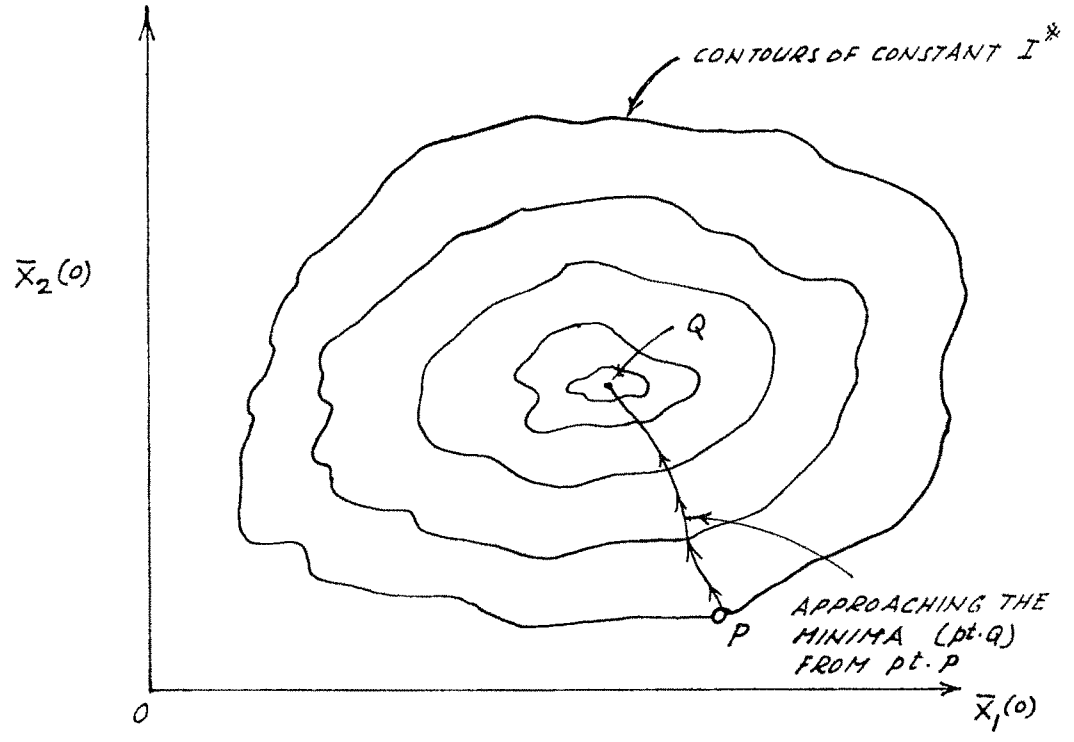


FIG. 3·2 ILLUSTRATION OF THE GRADIENT METHOD TO SEEK MINIMA ON A SURFACE REPRESENTING PERFORMANCE INDEX I AS A FUNCTIONAL OF TWO VARIABLES $X_1(0)$ AND $X_2(0)$