Computational Algorithm of Optimal Control

Yasuhiro KAWAZOE*

Abstract

The problem of solving the optimal control problem is discussed. It is well known that the optimal control problem can be formulated as a nonlinear two-point boundary-value problem. This paper discusses the algorithm for solving the two-point boundary-value problem. The approach is converting the two-point boundary-value problem into the initial single-point problem and applying the Newton method to improving assumed initial variables. The feature of the algorithm is that the storage is little. A computational procedure is shown in the flow chart.

Introduction

The modern formulation of optimal control is by means of calculus of variations[1], dynamic programming[2] or maximum principle[3].

Dynamic programming provides an elegant solution in the case of linear systems with quadratic performance criteria of these approaches. In the general case of nonlinear systems with nonquadratic performance criteria, some numerical computational techniques are required. The determination of optimal controls is in general a most difficult task. An analitical solution can be applied only in the most restricted cases.

Therefore, determination of optimal control must often be obtained with an iterative procedure by the aid of a computer.

Optimal control problems can be formulated as nonlinear two-point boundary-value problems. These problems are generally difficult to handle both analytically and computationally.

The approaches of numerical solutions by gradient method are discussed by Dyer, P. and McReynalds, S. R.[4] and successive sweep algorithm, the Newton-Raphson algorithm is derived.

Bryson, A. E. and Ho, Y. C.[5] and Kelly, H. J.[6] employed the steepest descent methods using control variables as the independent variables in the search procedure. Furthermore, the algorithm based on the generalized reduced gradient algorithm of Abadie for nonlinear programming is proposed to solve the optimal control problem numerically by Mehra, R. K. and Davis, R. E.[7].

The purpose of this paper is to present a successive procedure for seeking optimal control, which is converted into the nonlinear two-point boundary-value problem. The approach is solving the initial single value problem which is converted from the two-point boundary-value problem applying the Newton method. The organization of this paper is as follows:

* Department of Electrical Engineering, Technical Junior College of Yamaguchi University

The first section contains a statement of the problem and the second develops the solution. The computational algorithm is described in the third section. Furthermore, the expansion in the case where the initial state is unspecified is considered.

Statement of the problem

The discrete-time optimal control problem is considered.

Let the state transition equation be

$$\mathbf{x}(k+1)=\mathbf{f}[\mathbf{x}(k)]+G\mathbf{u}(k),\ k=0,\ 1,\ldots,\ N-1,\ \mathbf{x}(0)\ \text{given} \qquad \cdots\cdots(1)$$

and the cost function be

$$J=\frac{1}{2}\|\mathbf{x}(N)\|^2_{\Phi_{xx}}+\frac{1}{2}\sum_{k=0}^{N-1}\{\|\mathbf{x}(k)\|^2_{L_{xx}}+\|\mathbf{u}(k)\|^2_{L_{uu}}\} \qquad \cdots\cdots\cdots(2)$$

where k is the discrete-time index: $\mathbf{x}$ is an n-vector, the state: $\mathbf{u}$ is an m-vector, the control: G is an $n\times m$ matrix: Φ_{xx} and L_{xx} are $n\times n$ positive definite matrices: L_{uu} is an $m\times m$ positive definite matrix. Furthermore, $\mathbf{f}$ is n-dimensional vector-valued function of the indicated variable, N is a positive integer and $\|\mathbf{Z}\|^2_B$ is the quadratic form $\mathbf{z}'\mathbf{B}\mathbf{z}$. The problem is to find the sequence $\mathbf{u}(k)$, where $k=0,1,\ldots,N-1$, that minimize J.

Solution

The first step in deriving the algorithm is to form an augmented performance index J^* with a set of Lagrange multiplier vector λ.

$$J^*=\frac{1}{2}\|\mathbf{x}(N)\|^2_{\Phi_{xx}}+\frac{1}{2}\sum_{k=0}^{N-1}\{\|\mathbf{x}(k)\|^2_{L_{xx}}+\|\mathbf{u}(k)\|^2_{L_{uu}}\}$$

$$+\sum_{k=0}^{N-1}\lambda'(k+1)\{\mathbf{f}[\mathbf{x}(k)]+G\mathbf{u}(k)-\mathbf{x}(k+1)\} \qquad \cdots\cdots\cdots(3)$$

The problem becomes one of minimizing J^* subject to no constraints. It is shown that the first variation of Eq. (3) is given by

$$\Delta J^*=\Phi_{xx}\mathbf{x}(N)\Delta\mathbf{x}(N)+\sum_{k=0}^{N-1}\{L_{xx}\mathbf{x}(k)\Delta\mathbf{x}(k)+L_{uu}\mathbf{u}(k)\Delta\mathbf{u}(k)$$

$$+\nabla_x'\mathbf{f}[\mathbf{x}(k)]\lambda(k+1)\Delta\mathbf{x}(k)+G'\lambda(k+1)\Delta\mathbf{u}(k)-\lambda(k+1)\Delta\mathbf{x}(k+1)\}$$

$$=[\Phi_{xx}\mathbf{x}(N)-\lambda(N)]\Delta\mathbf{x}(N)+\sum_{k=0}^{N-1}\{[L_{xx}\mathbf{x}(k)+\nabla_x'\mathbf{f}[\mathbf{x}(k)]\lambda(k+1)-\lambda(k)]\Delta\mathbf{x}(k)$$

$$+[L_{uu}\mathbf{u}(k)+G'\lambda(k+1)]\Delta\mathbf{u}(k)\} \qquad \cdots\cdots\cdots(4)$$

One of the conditions of optimality is that the terms in Eq. (4) which depend on $\Delta\mathbf{x}$ and $\Delta\mathbf{u}$ must vanish identically.

Hence, the coefficient of $\Delta\mathbf{x}$ and $\Delta\mathbf{u}$ must be equated to zero. This leads to the following conditions.

$$\Phi_{xx}\mathbf{x}(N)-\lambda(N)=0 \qquad \cdots\cdots\cdots(5)$$

$$L_{xx}\mathbf{x}(k)+\nabla'_x\mathbf{f}[\mathbf{x}(k)]\lambda(k+1)-\lambda(k)=0,\ k=0,1,\ldots,N-1 \qquad \cdots\cdots(6)$$

$$L_{uu}\mathbf{u}(k)+G'\lambda(k+1)=0,\ k=0,1,\ldots,N-1 \qquad \cdots\cdots\cdots(7)$$

The control input is shown to be

$$\mathbf{u}(k)=-L_{uu}^{-1}G'\lambda(k+1),\ k=0,1,\ldots,N-1 \qquad \cdots\cdots\cdots(8)$$

from Eq. (7). Substituting Eq. (8) into Eq. (1), the state-transition equation is obtained as the function of λ.

$$\mathbf{x}(k+1)=\mathbf{f}[\mathbf{x}(k)]-GL_{uu}^{-1}G'\lambda(k+1),\ k=0,1,\ldots,N-1 \qquad \cdots\cdots\cdots(9)$$

Eq. (9) is expressed in the form

$$\mathbf{x}(k+1)=\boldsymbol{\alpha}[\mathbf{x}(k),\lambda(k+1)],\ k=0,1,\ldots,N-1. \qquad \cdots\cdots\cdots(10)$$

On the other hand, if $\nabla_x\mathbf{f}[\mathbf{x}(k)]$ is nonsingular for all $k=0,1,\ldots,N-1$, from Eq. (6),

$$\lambda(k+1)=\{\nabla'_x\mathbf{f}[\mathbf{x}(k)]\}^{-1}[\lambda(k)+L_{xx}\mathbf{x}(k)],\ k=0,1,\ldots,N-1 \qquad \cdots(11)$$

Eq. (11) is expressed in the form,

$$\lambda(k+1)=\boldsymbol{\beta}[\mathbf{x}(k),\lambda(k)],\ k=0,1,\ldots,N-1 \qquad \cdots\cdots\cdots(12)$$

Noting the requirement in Eq. (5) that

$$\lambda(N)=\Phi_{xx}\mathbf{x}(N) \qquad \cdots\cdots\cdots(13)$$

it follows that nonlinear two-point boundary-value problem is specified by Eq. (9), (11), (13) and the state initial condition $\mathbf{x}(0)$ given.

Computational procedure

The nonlinear two-point boundary-value problem under consideration is rewritten as follows:

$$\mathbf{x}(k+1)=\boldsymbol{\alpha}[\mathbf{x}(k),\lambda(k+1)] \qquad \cdots\cdots\cdots(10)$$

$$\lambda(k+1)=\boldsymbol{\beta}[\mathbf{x}(k),\lambda(k)] \qquad \cdots\cdots\cdots(12)$$

$$\mathbf{x}(0)=\mathbf{x}_0,\ \text{given} \qquad \cdots\cdots\cdots(14)$$

$$\lambda(N)=\Phi_{xx}\mathbf{x}(N) \qquad \cdots\cdots\cdots(13)$$

where $k=0,1,\ldots,N-1$ and $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the n-dimensional vector-valued functions. In order to overcome the analytical difficulties associated with solving this two-point boundary-value problem, the Newton method is examined. Given $\mathbf{x}(0)=\mathbf{x}_0$ and assumed $\lambda(0)=\lambda_0$, for each iteration $i=1, 2, 3,\ldots$, let $\mathbf{x}^i(k)$ and $\lambda^i(k)$ be the solutions of

$$\mathbf{x}^i(k+1)=\boldsymbol{\alpha}[\mathbf{x}^i(k), \lambda^i(k+1)] \quad \cdots\cdots\cdots(15)$$

$$\lambda^i(k+1)=\boldsymbol{\beta}[\mathbf{x}^i(k), \lambda^i(k)]. \quad \cdots\cdots\cdots(16)$$

If incremental changes of $\mathbf{x}^i(k)$ and $\lambda^i(k)$ are $\Delta\mathbf{x}^i(k)$ and $\Delta\lambda^i(k)$ respectively, the new $\mathbf{x}(k)$ and $\lambda(k)$ for the next iteration may be represented by

$$\mathbf{x}^{i+1}(k+1)=\mathbf{x}^i(k)+\Delta\mathbf{x}^i(k) \quad \cdots\cdots\cdots(17)$$

$$\lambda^{i+1}(k+1)=\lambda^i(k)+\Delta\lambda^i(k) \quad \cdots\cdots\cdots(18)$$

respectively. Then the variational equations for Eq. (15) and Eq. (16) are obtained as follows:

$$\Delta\mathbf{x}^i(k+1)=\nabla_x\boldsymbol{\alpha}[\mathbf{x}^i(k), \lambda^i(k+1)]\Delta\mathbf{x}^i(k)+\nabla_\lambda\boldsymbol{\alpha}[\mathbf{x}^i(k), \lambda^i(k+1)]\Delta\lambda^i(k+1) \quad \cdots\cdots\cdots(19)$$

$$\Delta\lambda^i(k+1)=\nabla_x\boldsymbol{\beta}[\mathbf{x}^i(k), \lambda^i(k)]\Delta\mathbf{x}^i(k)+\nabla_\lambda\boldsymbol{\beta}[\mathbf{x}^i(k), \lambda^i(k)]\Delta\lambda^i(k) \cdots\cdots\cdots(20)$$

where, $\nabla_x\boldsymbol{\alpha}$, $\nabla_\lambda\boldsymbol{\alpha}$, $\nabla_x\boldsymbol{\beta}$ and $\nabla_\lambda\boldsymbol{\beta}$ are $n\times n$ Jacobian matrices which are evaluated at the indicated arguments, e.g.,

$$\{\nabla_x\boldsymbol{\alpha}[\mathbf{x}^i(k), \lambda^i(k+1)]\}^{jl}=\left[\frac{\partial\boldsymbol{\alpha}^j(\mathbf{x}, \lambda)}{\partial\mathbf{x}^l}\right]_{\substack{x=x^i(k)\\ \lambda=\lambda^i(k+1)}} \quad \cdots\cdots\cdots(21)$$

where, j, $l=1, 2,\ldots, n$ and the superscripts denote the element.

The values of the variables which are needed to evaluate these Jacobian matrices are obtained from the solution of Eq. (15) and Eq. (16) for a given $\mathbf{x}_0$ and an assumed λ_0. It is quite obvious that in general, the solution of Eq. (15) and Eq. (16) for an arbitrary assumed λ_0 will not satisfy the boundary condition Eq. (13). However, the variational equation can be utilized to compute the correction of $\lambda^i(0)$, so that, to within the first-order term, the solution of Eq. (15) and Eq. (16) subject to the initial condition,

$$\lambda^{i+1}(0)=\lambda^i(0)+\Delta\lambda^i(0) \quad \cdots\cdots\cdots(22)$$

will satisfy Eq. (13).

For notational convenience, let the variational equations Eq. (19) and Eq. (20) be written

$$\Delta\mathbf{x}^i(k+1)=\alpha^i_x(k)\Delta\mathbf{x}^i(k)+\alpha^i_\lambda(k)\Delta\lambda^i(k+1) \quad \cdots\cdots\cdots(23)$$

$$\Delta\lambda^i(k+1)=\beta^i_x(k)\Delta\mathbf{x}^i(k)+\beta^i_\lambda(k)\Delta\lambda^i(k) \quad \cdots\cdots\cdots(24)$$

where,

$$\alpha_x^i(k) = \nabla_x \boldsymbol{\alpha}[\mathbf{x}^i(k),\ \lambda^i(k+1)] \qquad \cdots\cdots\cdots(25)$$

$$\alpha_x^i(k) = \nabla_\lambda \boldsymbol{\alpha}[\mathbf{x}^i(k),\ \lambda^i(k+1)] \qquad \cdots\cdots\cdots(26)$$

$$\beta_x^i(k) = \nabla_x \boldsymbol{\beta}[\mathbf{x}^i(k),\ \lambda^i(k)] \qquad \cdots\cdots\cdots(27)$$

$$\beta_x^i(k) = \nabla_\lambda \boldsymbol{\beta}[\mathbf{x}^i(k),\ \lambda^i(k)] \qquad \cdots\cdots\cdots(28)$$

Eq. (23) and Eq. (24) are written with the matrix form as follows:

$$\begin{bmatrix} \Delta\mathbf{x}^i(k+1) \\ \Delta\lambda^i(k+1) \end{bmatrix} = \begin{bmatrix} \alpha_x^i(k) & \alpha_\lambda^i(k) \\ \beta_x^i(k) & \beta_\lambda^i(k) \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}^i(k) \\ \Delta\lambda^i(k) \end{bmatrix} \qquad \cdots\cdots\cdots(29)$$

Setting,

$$\boldsymbol{\Phi}^i(k) = \begin{bmatrix} \alpha_x^i(k) & \alpha_\lambda^i(k) \\ \beta_x^i(k) & \beta_\lambda^i(k) \end{bmatrix} \qquad \cdots\cdots\cdots(30)$$

then, Eq. (29) is written,

$$\begin{bmatrix} \Delta\mathbf{x}^i(k+1) \\ \Delta\lambda^i(k+1) \end{bmatrix} = \boldsymbol{\Phi}^i(k) \begin{bmatrix} \Delta\mathbf{x}^i(k) \\ \Delta\lambda^i(k) \end{bmatrix}. \qquad \cdots\cdots\cdots(31)$$

From Eq. (31),

$$\begin{bmatrix} \Delta\mathbf{x}^i(N) \\ \Delta\lambda^i(N) \end{bmatrix} = \Theta^i(N) \begin{bmatrix} \Delta\mathbf{x}^i(0) \\ \Delta\lambda^i(0) \end{bmatrix} \qquad \cdots\cdots\cdots(32)$$

where, $\Theta^i(N)$ is obtained by the following recursive computation,

$$\Theta^i(k+1) = \boldsymbol{\Phi}^i(k)\Theta^i(k) \qquad \cdots\cdots\cdots(33)$$

where, Θ^i is the $2n \times 2n$ matrix and $\Theta^i(0)$ is setted unit matrix.

Partitioning matrix $\Theta^i(N)$ into submatrices by $n \times n$ matrices as follows,

$$\Theta^i(N) = \left[\begin{array}{c:c} \Theta_{11}^i(N) & \Theta_{12}^i(N) \\ \hdashline \Theta_{21}^i(N) & \Theta_{22}^i(N) \end{array}\right] \qquad \cdots\cdots\cdots(34)$$

then, from Eq. (32), $\Delta\lambda^i(N)$ is obtained.

$$\begin{aligned} \Delta\lambda^i(N) &= \Theta_{21}^i(N)\Delta\mathbf{x}^i(0) + \Theta_{22}^i(N)\Delta\lambda^i(0) \\ &= \Theta_{22}^i(N)\Delta\lambda^i(0) \end{aligned} \qquad \cdots\cdots\cdots(35)$$

where $\Delta\mathbf{x}^i(0)$ is zero because $\mathbf{x}^i(0)$ is given. Under the assumption that $\Theta_{22}^i(N)$ is nonsingular, from Eq. (35), $\Delta\lambda^i(0)$ is obtained.

$$\Delta\lambda^i(0) = [\Theta_{22}^i(N)]^{-1}\Delta\lambda^i(N) \qquad \cdots\cdots\cdots(36)$$

From the new iteration of $\lambda(N)$,

$$\lambda^{i+1}(N)=\lambda^i(N)+\Delta\lambda^i(N) \qquad \cdots\cdots\cdots(37)$$

which must satisfy Eq. (13),

$$\begin{aligned}\Delta\lambda^i(N)&=\lambda^{i+1}(N)-\lambda^i(N)\\&=\boldsymbol{\Phi}_{xx}\mathbf{x}^i(N)-\lambda^i(N)\end{aligned} \qquad \cdots\cdots\cdots(38)$$

Substituting Eq. (38) into Eq. (36),

$$\Delta\lambda^i(0)=[\Theta^i_{22}(N)]^{-1}[\boldsymbol{\Phi}_{xx}\mathbf{x}^i(N)-\lambda^i(N)] \qquad \cdots\cdots\cdots(39)$$

In Eq. (39), $\mathbf{x}^i(N)$ and $\lambda^i(N)$ is calculated by Eq. (15) and Eq. (16) for $k=0, 1,\ldots, N-1$ respectively. At the initial stage, $\lambda^{i+1}(0)$ for each succeeding iteration can be determined from the relation

$$\lambda^{i+1}(0)=\lambda^i(0)+\Delta\lambda^i(0). \qquad \cdots\cdots\cdots(40)$$

The computational procedure can now be summarized as follows:

Step 1. Guess initial value of $\lambda(0)$.

Step 2. With given $\mathbf{x}(0)$ and $\lambda(0)$ assumed above, solve Eq. (15) and Eq. (16).

Step 3. Concurrently with Step 2 evaluate the matrix $\boldsymbol{\Phi}^i(k)$ and use Eq. (33), to determine $\Theta^i(N)$.

Step 4. Retain $\Theta^i_{22}(N)$ at the completion of Step 2 and Step 3, and substitute this quantity into Eq. (39) to obtain $\Delta\lambda^i(0)$.

Step 5. Set,

$$\lambda^{i+1}(0)=\lambda^i(0)+\Delta\lambda^i(0) \qquad \cdots\cdots\cdots(40)$$

and return to the Step 2.

Step 6. Repeat until convergence is obtained.

Above procedure is shown in Fig. 1 as the flow chart.

Extension to the case of unknown initial state

In the above section the case of given initial state is treated and in this section we consider the case of unknown initial state. The problem is stated as follows:

State transition equation: $\mathbf{x}(k+1)=\mathbf{f}[\mathbf{x}(k)]+G\mathbf{u}(k)$, $\mathbf{x}(0)$ unspecified (41)

Cost function: Eq. (2).

Let $\hat{\mathbf{x}}(0)$ be representative of the assumed value of $\mathbf{x}(0)$ unspecified at the initial stage. Given $\hat{\mathbf{x}}(0)$ and $\lambda(0)$, for each iteration $i=1, 2,\ldots,$ let $\hat{\mathbf{x}}^i(k)$ and $\lambda^i(k)$ satisfy the following relations respectively.

$$\hat{\mathbf{x}}^i(k+1)=\boldsymbol{\alpha}[\hat{\mathbf{x}}^i(k), \lambda^i(k+1)] \qquad \cdots\cdots\cdots(42)$$

$$\lambda^i(k+1)=\boldsymbol{\beta}[\hat{\mathbf{x}}^i(k), \lambda^i(k)] \qquad \cdots\cdots\cdots(43)$$

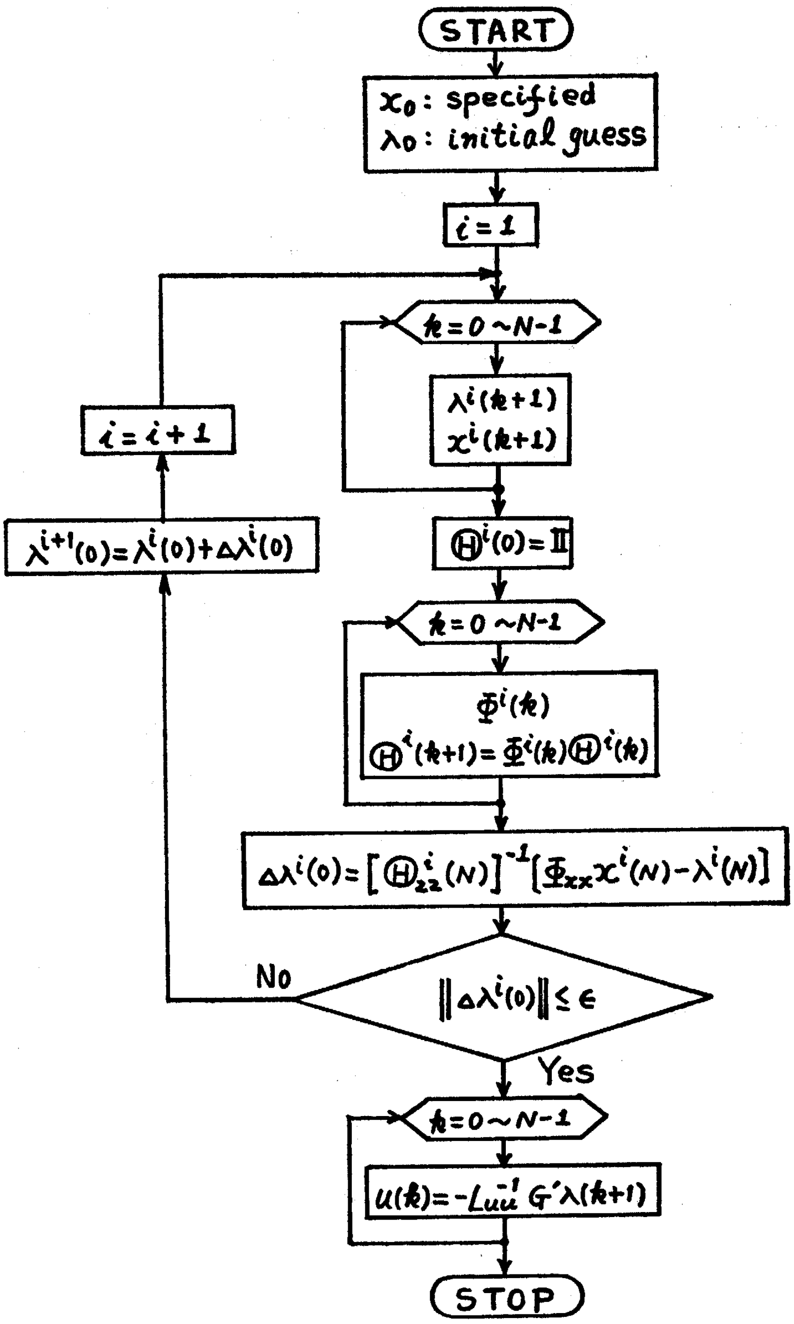

Fig. 1 Flow chart in the case of given state at initial stage.

Applied the expression $\Delta \mathbf{x}^i(k)$ and $\Delta\lambda^i(k)$ as the incremental changes of $\hat{\mathbf{x}}^i(k)$ and $\lambda^i(k)$ respectively, the itrative relation of $\hat{\mathbf{x}}(k)$ and $\lambda(k)$ is written,

$$\hat{\mathbf{x}}^{i+1}(k+1) = \hat{\mathbf{x}}^i(k) + \Delta\hat{\mathbf{x}}^i(k) \qquad \cdots\cdots\cdots(44)$$

$$\lambda^{i+1}(k+1) = \lambda^i(k) + \Delta\lambda^i(k) \qquad \cdots\cdots\cdots(45)$$

respectively. Then the variational equation is obtained in the matrix form.

$$\begin{bmatrix} \Delta\hat{\mathbf{x}}^i(k+1) \\ \Delta\lambda^i(k+1) \end{bmatrix} = \begin{bmatrix} \alpha^i_{\hat{x}}(k) & \alpha^i_{\lambda}(k) \\ \beta^i_{\hat{x}}(k) & \beta^i_{\lambda}(k) \end{bmatrix} \begin{bmatrix} \Delta\hat{\mathbf{x}}^i(k) \\ \Delta\lambda^i(k) \end{bmatrix} \qquad \cdots\cdots\cdots(46)$$

In the similar fashion in the above section, the following relation is obtained.

$$\begin{bmatrix} \Delta\hat{\mathbf{x}}^i(N) \\ \Delta\lambda^i(N) \end{bmatrix} = \hat{\Theta}^i(N) \begin{bmatrix} \Delta\hat{\mathbf{x}}^i(0) \\ \Delta\lambda^i(0) \end{bmatrix} \qquad \cdots\cdots\cdots(47)$$

where, $\hat{\Theta}^i(N)$ is obtained by the relation,

$$\hat{\Theta}^i(k+1) = \hat{\Phi}^i(k)\hat{\Theta}^i(k) \qquad \cdots\cdots\cdots(48)$$

$$\hat{\Phi}^i(k) = \begin{bmatrix} \alpha^i_{\hat{x}}(k) & \alpha^i_{\lambda}(k) \\ \beta^i_{\hat{x}}(k) & \beta^i_{\lambda}(k) \end{bmatrix}. \qquad \cdots\cdots\cdots(49)$$

Then, assuming that the inverse of $\hat{\Theta}^i(N)$ exists,

$$\begin{bmatrix} \Delta\hat{\mathbf{x}}^i(0) \\ \Delta\lambda^i(0) \end{bmatrix} = [\hat{\Theta}^i(N)]^{-1} \begin{bmatrix} \Delta\hat{\mathbf{x}}^i(N) \\ \Delta\lambda^i(N) \end{bmatrix} \qquad \cdots\cdots\cdots(50)$$

From the new iteration of $\hat{\mathbf{x}}^{i+1}(N)$ and $\lambda^{i+1}(N)$, incremental changes of $\hat{\mathbf{x}}(N)$ and $\lambda(N)$ are shown by the aid of the requirement of Eq. (13).

$$\begin{bmatrix} \Delta\hat{\mathbf{x}}^i(N) \\ \Delta\lambda^i(N) \end{bmatrix} = \begin{bmatrix} \Phi^{-1}_{xx}\lambda^i(N) \\ \Phi_{xx}\hat{\mathbf{x}}^i(N) \end{bmatrix} - \begin{bmatrix} \hat{\mathbf{x}}^i(N) \\ \lambda_i(N) \end{bmatrix} \qquad \cdots\cdots\cdots(51)$$

Then substituting Eq. (51) into Eq. (50), incremental changes of $\hat{\mathbf{x}}$ and λ at the initial stage are shown to be,

$$\begin{bmatrix} \Delta\hat{\mathbf{x}}^i(0) \\ \Delta\lambda^i(0) \end{bmatrix} = [\hat{\Theta}^i(N)]^{-1} \begin{bmatrix} \Phi^{-1}_{xx}\lambda^i(N) - \hat{\mathbf{x}}^i(N) \\ \Phi_{xx}\hat{\mathbf{x}}^i(N) - \lambda^i(N) \end{bmatrix}. \qquad \cdots\cdots\cdots(52)$$

The new iterations of $\hat{\mathbf{x}}$ and λ at the initial stage are repeated by the relations,

$$\begin{bmatrix} \hat{\mathbf{x}}^{i+1}(0) \\ \lambda^{i+1}(0) \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}^i(0) \\ \lambda^i(0) \end{bmatrix} + \begin{bmatrix} \Delta\hat{\mathbf{x}}^i(0) \\ \Delta\lambda^i(0) \end{bmatrix}. \qquad \cdots\cdots\cdots(53)$$

The computational procedure can be summarized as follows:

Step 1. Guess initial values of $\lambda(0)$ and $\mathbf{x}(0)$.

Step 2. With above values, solve Eq. (42) and Eq. (43).

Step 3. Concurrently with Step 2 evaluate the matrix $\hat{\Phi}^i(k)$ and use Eq. (48), to determine $\hat{\Theta}^i(N)$.

Step 4. Retain $\hat{\Theta}^i(N)$ at the completion of Step 2 and Step 3, and substitute this quantity into Eq. (52) to obtain Eq. (52).

Step 5. Set Eq. (53) and return to Step 2.
Step 6. Repeat until convergence is obtained.
Above procedure is shown in Fig. 2.

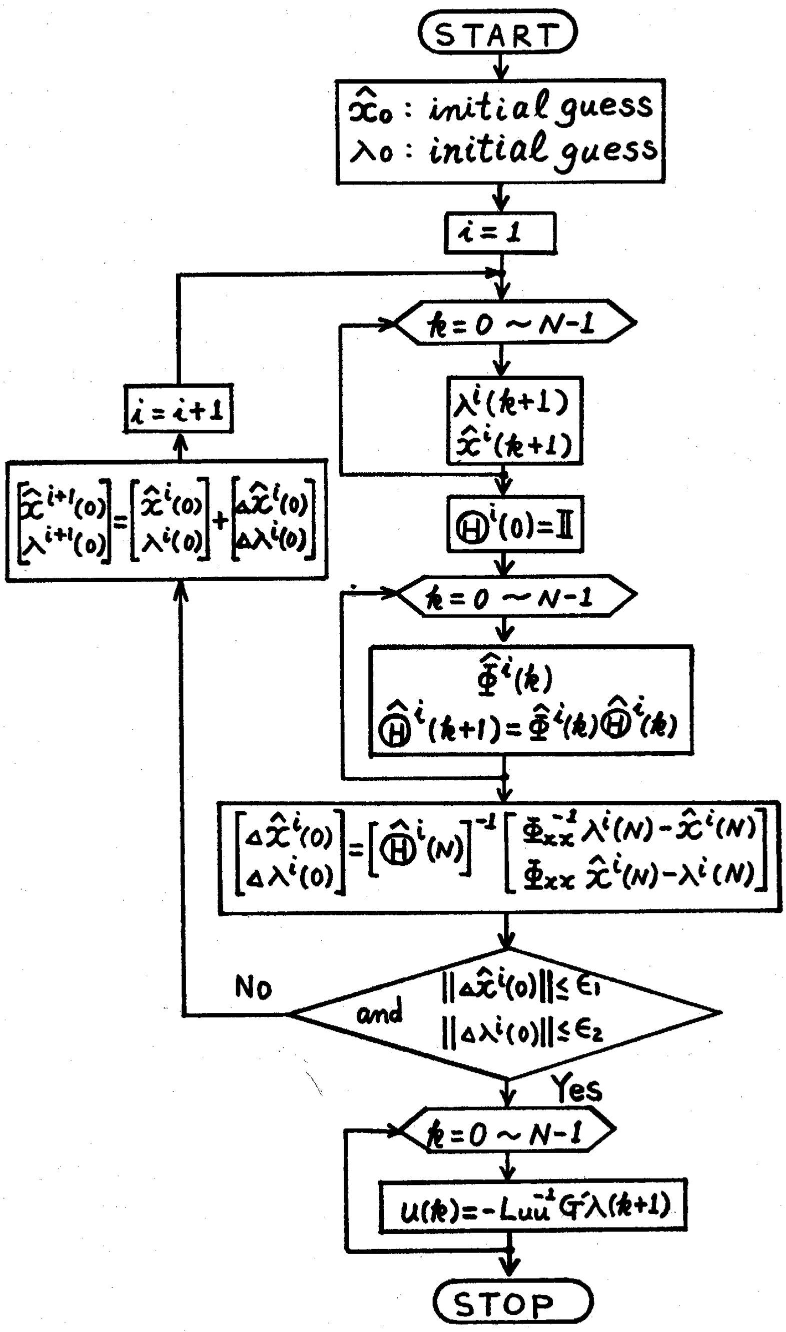

Fig. 2 Flow chart in the case of unspecified initial state.

Discussion

An important feature of the present method is that it deals with the optimal control problem with the certain cost function without the requiring solution of dynamic programming and/or maximum principle. Lagrange multiplier method is employed to solve the minimization of the cost function subject to the state constraints. The optimal control is obtained by the solution of the nonlinear two-point boundary-value problem in Eq. (10), Eq. (12), Eq. (13) and Eq. (14). The above problem is converted into an initial value problem. The another approaches converting the two-point boundary-value problem into an initial value problem are invariant imbedding method[9] or sweep method, which lead to a sequential computational algorithm for determining the optimal control. The gradient method is the approach that is developed by considering the gradient of the cost function straightforward with respect to the nominal control variable sequence. The independent variable λ is calculated recursively by solving Eq. (15) and Eq. (16) for satisfying the terminal condition Eq. (13). The directions of search are determined by the Newton method algorithm which is first-order approximation. At the conclusion of this procedure, the initial value of the adjoint variable λ is corrected by the final condition of the state variable and the adjoint variables at the final stage.

The method presented in this paper is relatively simple to apply in practice because this method requires low storage.

In particular, the variables which are required for storing to initiate the next iteration are only $\lambda^{i+1}(0)$ in the former section, and $\lambda^{i+1}(0)$ and $\hat{\mathbf{x}}^{i+1}(0)$ in the latter section, which are computed at the end of one iteration. Quasilinearization techniques[8),9)], require in general storage of entire time histories to proceed from one iteration to the next. The Newton-type methods as presented in this paper have the problem of a good initial estimate. In the case where an estimate is not available it may be necessary to utilize a gradient method to obtain a good initial estimate. The reasonable criterion to terminate the iterations is designed $||\Delta\lambda^{i}(0)||\leq\varepsilon$ where ε is a small positive number. In the latter case where the initial state variable is unspecified, the choice of $\hat{\mathbf{x}}(0)$ to initialize the procedure is very important as well as the choice of $\lambda(0)$ and in general dictates whether or not the procedure converges. The iterative procedure has the danger of converging on a local minimum of the cost function. This is related with a good initial estimate. For example, the desired control in a given region of state space is computed in the case of the air craft landing problem which gives the approximate form of the known desired landing trajectory.[10]

The author have demonstrated an iterative procedure for computing deterministic optimal control. The two-point boundary-value problem which is generally difficult to handle both analytically and computationally is effectively reduced to an initial value problem. The initial value problem is solved as the iterative alhorithm. The two cases of the specified and the unspecified initial state are considered and the computational procedure is shown.

References

1) Balakrishnan, A. V. Ed. "Control Theory and the Calculus of Variations" (Academic Press, N. Y.) (1969)
2) Bellman, R. E. "Dynamic Programming" (Prenceton Univ. Press, Princeton, N. J.) (1957)
3) Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidge, R. V. and Mischenko, E. F. "The Mathematical Theory of Optimal Processes" (Interscience Publishers, N. Y.) (1962)
4) Dyer, P. and McReynalds, S. R. "The Computation and Theory of Optimal Control" (Academic Press, N. Y.) p. 56 (1970)
5) Bryson, A. E. and Ho, Y. C. "Applied Optimal Control" (Blaisdell Publishing, Waltham, Mass.) p. 212 (1969)
6) Kelly, H. J. "Method of Gradients, in Optimization Techniques, Leightman, G. Ed." (Academic Press, N. Y.) p. 206 (1962)
7) Mehra, R. K. and Davis, R. E. "A Generalized Gradient Method for Optimal Control Problems with Inequality Constraints and Singular Arcs", JACC, p. 141 (1971)
8) Bellman R. E. and Kalaba, R. E. "Quasilinearization & Nonlinear Boundary-Value Problems" (Elsevier, N. Y.) p. 27 (1965)
9) Lee, E. S. "Quasilinearization and Invariant Imbeding" (Academic Press, N. Y.) p. 9 (1968)
10) Noton, A. R. M., Dyer, P. and Markland, C. A. "Numerical Computation of Optimal Control", IEEE Trans. AC-12, p. 59 (1967)