ON THE COMPUTATION OF THE OPTIMAL CONTROLS FOR A LINEAR PLANT WITH TIME-VARYING WEIGHTING MATRICES IN THE QUADRATIC COST FUNCTIONAL

By ALEXIS LAZARIDIS, Ph. D.

Ministry of Coordination and Planning,

Preface

The broad objective of control theory is to provide systematic methods to affect systems in a desired way (i.e. to be more reliable or to work more accurately or more economically in spite of perturbation from outside the system).

The optimal control problem in its general context is :

For some well-defined goals, determine, within the set of all admissible policy parameters (or controls), the time sequence of the controls required to achieve these goals in an optimum manner. If the controls required are expressed as an explicit function of the system state, they are referred to as a control law or control strategy. The model and the nature of the control problem as encountered here are mostly identical in form to the control problems encountered in electronics and aerospace industries. We have limited ourselves to the linear, discrete-data, time-invariant systems, since this is the usual case when optimal control is to be applied to economic policy planning. The «system» in this case is presented by an econometric model — namely a set of difference equations. The boundary conditions are the initial values of the variables, and perhaps, desired values for the variables at some terminal time. Finally, the cost functional is a quantitative representation of the planner's goals, objectives, and utilities.

For the above class of optimal control problems, the perfect measurement of the system state is usually assumed. Hence the state transformation function (or system transition equation) is only considered, since the observation equation becomes redundant.

The main objective of this paper is to present the mathematics involved to solve the optimal control problem through using the technique of discretized minimum principle. This way the solution is derived sequentially by solving a set of Riccati type equations. Besides, we have developed a proper computer package, and part of the paper serves as a manual for the developed program to solve a linear, discrete - data, time - invariant optimal control problem, with time - varying weights which refer to the prescribed set of goals.

DERIVATION OF THE DISCRETE MINIMUM PRINCIPLE MANUAL FOR THE COMPUTER PROGRAM RICCATI

I. Purpose of the program

The computer program RICCATI — written in FORTRAN IV — computes the optimal control sequence and the corresponding state trajectory of a controllable *, discrete - time, linear system according to the specified quadratic cost functional (performance criterion). The time - varying feedback coefficients and the co-state vectors are also computed and printed.

The derivation of the solution is based upon the discrete minimum principle which allows the computation of the optimal control and state vectors by solving a set of Riccati type equations.

2. Theoretical background

2.1. Application of the Kuhn - Tucker theory in convex programming

Given the MP problem

$$\min j(\underline{y})$$

$$\text{s.t.}\ F(\underline{y}) = 0$$

where

$$F(\underline{y}) = \begin{bmatrix} f_1(\underline{y}) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ f_k(\underline{y}) \end{bmatrix}$$

* For details regarding controllability see : A. Lazaridis, «Optimal Planning for the Cattle Industry in Greece : An Application of Optimal Control», Ph. D. Thesis, University of Birmingham, 1977, pp. 185 - 187.

The $f_1(y)$, $i = 1, \ldots, k$, are scalar - valued functions of the s- vector $\underline{y}$, and $F(\underline{y})$ is a vector-valued function of dimension k.

Assumption: $j(\underline{y})$ and $F(\underline{y})$ are all differentiable in each of their arguments, and the constraint function $F(\underline{y})$ is convex in $\underline{y}$.

— Define the Lagrangean

$$L(\underline{y}, \underline{p}) = j(\underline{y}) + \underline{p}' E(\underline{y})$$

where $\underline{p} \in E^k$ is the vector of Lagrange multipliers. (Prime denotes transposition).

Kuhn-Tucker theorem

1. If $\underline{y}^*$ minimizes $j(\underline{y})$ subject to $F(\underline{y}) = 0$, it is necessary that there exist some $\underline{p}^*$ vectors of multipliers such that

$$\nabla_{\underline{y}}{}^* L = 0 \qquad 1.1$$

$$\nabla_{\underline{p}}{}^* L = 0 \qquad 1.2$$

where $\nabla_{\underline{y}}{}^* L$ denotes the gradient of the Lagrangean with respect to $\underline{y}$, evaluated at $\underline{y} = \underline{y}^*$

2. If $\underline{y}^*$ is a solution to the stated MP problem, it is sufficient that (1.1) and (1.2) hold, and that for all $\underline{y}$

$$L(\underline{y}, \underline{p}^*) \geqslant L(\underline{y}^*, (\underline{p}^*) + [\nabla_y{}^* L], (\underline{y} - \underline{y}^*) \qquad 1.3$$

Egs. (1.1) and (1.2) are the necessary conditions for a solution to the MP problem, but in general there might exist more than one pair of $\underline{y}^*$ and $\underline{p}^*$ satisfying those conditions. Theorem 2, however, states that if the optimization problem is such that the Lagrangean has a unique minimum with respect to $\underline{y}$ (note that (1.3) is just a convexity condition on L), then there is only one extremal solution and (1.1) - (1.2) consitute the set of necessary and sufficient conditions for the optimum.

2.2. Application of the K - T theorem to the Linear, Discrete - Time Optimal Control Problem

Given the system transition equation

$$\underline{x}(i+1) - \underline{x}(i) = A\underline{x}(i) + B\underline{u}(i) \qquad (1.3a)$$

$$i = 0, \ldots, N-1$$

where $\underline{x} \in E^n$, is the system state vector,

$\underline{u} \in E^m$, is the system input (or control) vector

and A, B are coefficient matrices of proper dimentions.

Assuming that $\underline{x}(o)$ and N are fixed, we want to minimize the cost functional

$$j = M[\underline{x}(N)] + \sum_{i+o}^{N-1} l(\underline{x}(i), \underline{u}(i)) \qquad (1.3b)$$

To simplify the presentation it is assumed that N = 4.

To convert the optimal control problem to a MP problem it is necessary to define : — the vector - valued function $\underline{y} \in E^{(n+m)N}$ by

$$\underline{y} = \begin{bmatrix} \underline{x}(1) \\ \cdot \\ \cdot \\ \cdot \\ \underline{x}(N) \\ \underline{u}(0) \\ \cdot \\ \cdot \\ \cdot \\ \underline{u}(N-1) \end{bmatrix}$$

and the vector - valued function $F(y) \in E^{nN}$ by

$$F(y) = \begin{bmatrix} A\underline{x}(0) + B\underline{u}(0) + \underline{x}(0) - \underline{x}(1) \\ A\underline{x}(1) + B\underline{u}(1) + \underline{x}(1) - \underline{x}(2) \\ A\underline{x}(2) + B\underline{u}(2) + \underline{x}(2) - \underline{x}(3) \\ A\underline{x}(3) + B\underline{u}(3) + \underline{x}(3) - \underline{x}(4) \end{bmatrix}$$

It can be seen then that the equivalent MP problem has the form

$$\text{mix } j(y) = M[\underline{x}(N)] + l(\underline{y})$$

$$\text{s.t.} \qquad F(y) = 0$$

The Lagrangean is defined as

$$L(\underline{y}, \underline{p}) = L(\underline{x}, \underline{u}, \underline{p}) = M[\underline{x}(N)] + l(\underline{y}) + \underline{p}'E(y)$$

where $\underline{P} \in E^{nN}$

Writing the Lagrangean in expanded form we have :

$$L(\underline{x}, \underline{u} . \underline{p}) = M[\underline{x}(N)] + l[\underline{y}]$$

$$+ [\underline{P}_1' \ \underline{P}_2' \ \underline{P}_3' \ \underline{P}_4'] \begin{bmatrix} A\underline{x}(0) + B\underline{u}(0) + \underline{x}(0) - \underline{x}(1) \\ A\underline{x}(1) + B\underline{u}(1) + \underline{x}(1) - \underline{x}(2) \\ A\underline{x}(2) + B\underline{u}(2) + \underline{x}(2) - \underline{x})3) \\ A\underline{x}(3) + B\underline{u}(3) + \underline{x}(3) - \underline{x}(4) \end{bmatrix}$$

where $P_i \in E^u$, $i = 1, \ldots, N$.

By further expansion the Lagrangean is written as

$$L(\underline{x}, \underline{u}, \underline{p}) = M[\underline{x}(N)] + l[\underline{y}]$$

$$\left.\begin{array}{l} + \underline{P}_1' A\underline{x}(0) + \underline{P}'_2 A\underline{x}(1) + \underline{P}_3' A\underline{x}(2) + \underline{P}_4' A\underline{x}(3) \\ + \underline{P}_1' B\underline{u}(0) + \underline{P}_2' B\underline{u}(1) + \underline{P}_3' B\underline{u}(2) + \underline{P}_4' B\underline{u}(3) \\ + \underline{P}_1' \ \underline{x}(0) + \underline{P}_2' \ \underline{x}(1) + \underline{P}_3' \ \underline{x}(2) + \underline{P}_4' \ \underline{x}(4) \\ - \underline{P}_1' \ \underline{x}(1) - \underline{P}_2' \ \underline{x}(2) - \underline{P}_3' \ \underline{x}(3) - \underline{P}_4' \ \underline{x}(4) \end{array}\right\} C$$

Before applying the K - T conditions we form the gradient of C with respect to $\underline{x}_1$, for $i = 1, \ldots, N - 1$.

$$\frac{\partial c}{\partial x_1} = A'\underline{p}_2 + \underline{p}_2 - \underline{p}_1$$

$$\frac{\partial \text{ɔ}}{\partial \underline{x}_2} = A'\underline{p}_3 + \underline{p}_3 - \underline{P}_2$$

$$\frac{\partial c}{\partial \underline{x}_3} = A'\underline{p}_4 + \underline{p}_4 - p_3$$

In general, for i = 1, ..., N - 1, it is

$$\nabla_{\underline{x}_i} C = A' \underline{P}(i+1) + \underline{p}(i+1) - \underline{p}(i)$$

For i = 4 = N we have

$$\frac{\partial c}{\partial \underline{x}_4} = - \underline{P}_4$$

in general, for i = N

$$\nabla_{\underline{x}_N} C = - \underline{P}(N)$$

Similarly we fndi that

$$\nabla_{\underline{u}_i} C = B' \underline{p}(i+1), \text{ for } i = 0, \ldots, N-1$$

Considering the K - T onditions, we have

$$\nabla_{\underline{y}^*} L = 0$$

i.e.

$$\begin{cases} \nabla_{\underline{x}^*} L = 0 \\ \nabla_{\underline{u}^*} L = 0 \end{cases}$$

which yield

$$\nabla_{\underline{x}_i^*} L = \frac{\partial l}{\partial \underline{x}_i} + A' \underline{p}^*(i+1) + \underline{p}^*(i+1) - \underline{p}^*(i) = 0$$

$$i = 1, \ldots, N-1$$

hence

$$\underline{p}^*(i+1) - \underline{p}^*(i) = - \frac{\partial l}{\partial \underline{x}^*_i} - A' \underline{p}^*(i+1), \ i = 1, \ldots, N-1 \qquad (1.4)$$

$$\nabla_{\underline{x}^*_N} L = \frac{\partial}{\partial x_N^*} M[\underline{x}^*(N)] - \underline{p}^*(N) = 0 \Rightarrow \underline{p}^*(N) =$$

$$= \frac{\partial}{\partial \underline{x}_N^*} M[\underline{x}^*(N)] \qquad (1.5)$$

$$\nabla \underline{u}_i{}^* L = \frac{\partial l}{\partial \underline{u}_i{}^*} + B' \underline{p}^*(i+1) = 0, \quad i = 0, \ldots, N-1 \tag{1.6}$$

$$\nabla p_i{}^* L = -\underline{x}^*(i) + \underline{x}^*(i-1) + A\underline{x}^*(i-1) + B\underline{u}^*(i-1) = 0, \; i = 1, \ldots, N$$

i.e. $$\underline{x}^*((i) - \underline{x}^*(i-1) = A\underline{x}^*(i-1) + B\underline{u}^*(i-1)$$

Shifting to $i+1$ we get

$$\nabla \underline{p}^*_{i+1} L = 0, \quad i = 0. \ldots, N-1$$

which yields

$$\underline{x}^*(i+1) - \underline{x}^*(i) = A\underline{x}(i) + B\underline{u}^*(i) \tag{1.7}$$

Eqs. (1.4), (1.5), (1.6) and (1.7) constitute a set of necessary conditions for the solution of the optimal control problem. We will restate those results in the dynamic form of a minimum principle. Define the scalar function H, which is called Hamiltonian, by

$$H(\underline{x}(i), \underline{p}(i+1), u(i)) = l(\underline{x}(i), \underline{u}(i)) + \underline{p}'(i+1)(A\underline{x}(i) + B\underline{u}(i)) \tag{1.8}$$

Note that $\underline{p}(i)$ $(i = 1, \ldots, N)$ is the n-dimensional co-state vector evaluated at time i. Again let $\underline{x}^*$, $\underline{u}^*$ and $\underline{p}^*$ denote the values of $\underline{x}$, $\underline{u}$ and $\underline{p}$ that yield an optimal solution. It can be seen then that (1.4) can be written as

$$\underline{p}^*(i+1) - \underline{p}^*(i) = -\left.\frac{\partial H}{\partial \underline{x}(i)}\right| \text{ on the optimal trajectory } i = 1, \ldots, N-1$$

Eq. (1.5) need not be restated. It holds as it stands, i.e.

$$\underline{p}^*(N) = \frac{\partial}{\partial \underline{x}_N{}^*} M[(\underline{x}^*(N)]$$

As long as H is convex in u, eq. (1.6) can be written as

$$0 = \left.\frac{\partial H}{\partial \underline{u}(i)}\right| \text{ on the optimal trajectory } i = 0, \ldots, N-1$$

Eq. (1.7) is equivalent to

$$\underline{x}^*(i+1) - \underline{x}^*(i) = \left.\frac{\partial H}{\partial p(i+1)}\right| \text{ on the optimal trajectory } i = 0, \ldots, N-1$$

2.3. The Discrete Minimum Principle

We can state now the discrete minimum principle. Let $[\underline{x}^*(i)]$, $i = 0, \ldots N$, be the state trajectory of the system described by (1.3a), corresponding to th, control sequence $\{\underline{u}^*(i)\}$, $i = 0, \ldots, N-1$, with $\underline{x}(0)$ fixed. Then if $\{\underline{u}^*(i)\}$ minimizes the cost functional (1.3b), it is necessary that there exist a sequence of vectors $\underline{p}^*(i) \in E^n$, $i = 1, \ldots, N$, called the co-states, such that

1. The Scalar Function

$$H(\underline{x}^*(i), \underline{p}^*(i+1), \underline{u}^*(i)) = l(\underline{x}^*(i), \underline{u}^*(i)) + \\ + \underline{p}^{*\prime}(i+1)\,[A\underline{x}^*(i) + B\underline{u}^*(i)]$$

called the Hamiltonian is minimized as a function of $\underline{u}(i)$ at $(\underline{u}\,i) = \underline{u}^*(i)$ for all $i = 0, \ldots, N-1$. The above can also be stated as

$$H(\underline{x}^*(i), \underline{p}^*(i+1), \underline{u}^*(i)) = \min_{\underline{u}_1 \in U_1} H(\underline{x}^*(i), \underline{p}^*(i+1), \underline{u}(i))$$

where U_i denotes the control space.

2. The dynamics of $\underline{x}^*(i)$ and $\underline{p}^*(i)$ are determined by the following difference equations

$$\underline{x}^*(i+1) - \underline{x}^*(1) = \left.\frac{\partial H}{\partial \underline{p}^*(i+1)}\right|_* = A\underline{x}^*(i) + B\underline{u}^*(i) \qquad (1.9)$$

$$\underline{p}^*(i+1) - \underline{p}^*(i) = -\left.\frac{\partial H}{\partial \underline{x}(i)}\right|_* \qquad (1.10)$$

with boundary condition

$$\underline{p}^*(N) = \frac{\partial}{\partial \underline{x}_N^*} M[\underline{x}^*(N)] \qquad (1.11)$$

The control problem which will be discussed has a linear plant equation with a quadratic cost functional, so that the convexity condition of the K-T theorem is indeed fulfilled. This, in turn, implies that an extreme solution satisfying the conditions derived above, is the unique optimal solution.

The plant equation is as in (1.3a), i.e.

$$\underline{x}(i+1) - \underline{x}(i) = A\underline{x}(i) + B\underline{u}(i) \qquad i = 0, \ldots, N-1 \tag{1.12}$$

with $\underline{x}(o)$ and N fixed (1.13)

The performance criterion, given below, has been formulated in a more general form, taking into consideration that the controller (decision maker) wants to minimize the sum of the weighted squared error between the feasible and the prescribed (nominal) output position of the system over the finite number of the time instants considered. In other words, given (1.12), (1.13) and the prescribed (nominal) state and control paths $\{\ddot{\underline{x}}(i), \ddot{\underline{u}}(i-1)\}$, the aim is to determine the optimal sequence $\{\underline{x}^*(i), \underline{u}^*(i-1)]$ for $i = 1, \ldots, N$, such that the cost functional

$$j = \frac{1}{2}[\underline{x}(N) - \overline{\underline{x}}(N)]'Q(N)[\underline{x}(N) - \overline{\underline{x}}(N)]$$

$$+\frac{1}{2}\sum_{l=o}^{N-I}\Big\{[\underline{x}(i) - \overline{\underline{x}}(i)]'Q(i)[\underline{x}(i) - \overline{\underline{x}}(i)] + [\underline{u}(i) -$$

$$- \overline{\underline{u}}(i)]'R(i)[\underline{u}(i) - \overline{\underline{u}}(i)]\Big\} \tag{1.14a}$$

is minimized.

Note that Q(i) and R(i) are the so-called weighting matrices which usually are diagonal. Q(i) is $n \times n$ positive semi-definite and R(i) is $m \times m$ positive-definite. Both are symmetric (or assumed that they have been transformed to symmetric matrices).

First we form the Hamiltonian

$$H(\underline{x}(i), \underline{p}(i+1), \underline{u}(i)) = \frac{1}{2}\Big\{[\underline{x}(i) - \overline{\underline{x}}(i)]'\mathbf{Q}(i)[\underline{x}(i) - \overline{\underline{x}}(i)]$$

$$+ [\underline{u}(i) - \overline{\underline{u}}(i)]'R(i)[\underline{u}(i) - \overline{\underline{u}}(i)]\Big\} + \underline{p}'(i+1)A\underline{x} + \ddot{\underline{p}}'(i+1)B\underline{u}(i)$$

From (1.9) we have

$$\underline{x}^*(i+1) - \underline{x}^*(i) = A\underline{x}^*(i) - B\ddot{\underline{u}}^*(i) = \frac{\partial H}{\partial \underline{P}^*(i+1)}\Bigg|_* \tag{1.14}$$

From (1.10) we get

$$\underline{p}^*(i+1) - \underline{p}^*(i) = -\left.\frac{\partial H}{\partial \underline{x}(i)}\right|_* = -Q(i)\,\underline{x}^*(i) + Q(i)\,\overline{\underline{x}}(i) +$$

$$-A'\underline{p}^*(i+1) \qquad (1.15)$$

and
$$\underline{p}^*(N) = \frac{\partial}{\partial \underline{x}^*N} M\{\underline{x}^*(N)\}$$

$$M[\underline{x}^*(N)] = \frac{1}{2}[\underline{x}^*(N) - \overline{\underline{x}}(N)]'Q(N)[\underline{x}^*(N) - \overline{\underline{x}}(N)] \qquad 1.15a)$$

Hence

$$\frac{\partial}{\partial \underline{x}_N^*} M[\underline{x}^*(N)] = Q(N)\,\underline{x}^*(N) - Q(N)\,\overline{\underline{x}}(N) = \underline{p}^*(N) \qquad (1.16)$$

From (1.16) above it is clear that $\underline{p}^*(i)$ and $\underline{x}^*(i)$ are linearly related such that

$$\underline{p}^*(i) = K(i)\,\underline{x}^*(i) + \underline{q}(i) \Rightarrow \underline{p}^*(i+1) = K(i+1)\,\underline{x}^*(i+1) + \underline{g}(i+1) \qquad (1.17)$$

where $K(i)$, $i = 1, \ldots, N$ are symmetric, positive semi-definite matrices. The minimization of Hamiltonian — which is convex in $\underline{u}$ — is written

$$\left.\frac{\partial H}{\partial \underline{u}(i)}\right|_* = 0, \Rightarrow R(i)\,\underline{u}^*(i) - R(i)\,\overline{\underline{u}}(i) + B'p^*(i+1) = 0$$

$$\Rightarrow \underline{u}^*(i) = -R^{-1}(i)\,B'\underline{p}^*(i+1) + \overline{\underline{u}}(i) \qquad (1.18)$$

Subsititution of eq. (1.18) into (1.14) for $\underline{u}^*(i)$ yields

$$\underline{x}^*(i+1) - \underline{x}^*(i) = A\underline{x}^*(i) - BR^{-1}(i)\,B'\underline{p}^*(i+1) + B\overline{\underline{u}}(i) \qquad (1.19)$$

Substituting (1.17) into (1.19) for $\underline{p}^*(i+1)$ we get

$$\underline{x}^*(i+1) - \underline{x}^*(i) = A\underline{x}^*(i) - BR^{-1}(i)\,B'K(i+1)\,\underline{x}^*(i+1) -$$

$$-BR^{-1}(i)\,B'\underline{g}(i+1) + B\overline{\underline{u}}(i)$$

$$\Rightarrow [I + BR^{-1}(i)\,B'K(i+1)]\,\underline{x}^*(i+1) = (I+A)\,\underline{x}^*(i) -$$

$$-BR^{-1}(i)\,B'\underline{g}(i+1) + B\overline{\underline{u}}(i)$$

Denote $I + BR^{-1}(i)B'K(i+1) = E(i) = 0, \ldots, N-1$. Note that $E^{-1}(i)$, exists, since for all i, E(i) is the sum of a positive definite and a positive semi-definite matrix. Hence,

$$\underline{x}^*(i+1) = E^{-1}(i)(I+A)\underline{x}^*(i) - E^{-1}(i)BR^{-1}(i)B'\underline{g}(i+1) + $$
$$+ E^{-1}(i)B\overline{\underline{u}}(i) \qquad (1.20)$$

We consider eq. (1.15), i.e.

$$\underline{p}^*(i+1) - \underline{p}^*(i) = -Q(i)\underline{x}^*(i) + Q(i)\overline{\underline{x}}(i) - A'\underline{p}^*(i+1) \qquad (1.21)$$

Subsitution of (1.17) into (1.21) for $\underline{p}^*(i)$ and $\underline{p}^*(i+1)$ yields

$$K(i+1)\underline{x}^*(i+1) + \underline{g}(i+1) - K(i)\underline{x}^*(i) - \underline{g}(i) = -Q(i)\underline{x}^*(i) + $$
$$+ Q(i)\overline{\underline{x}}(i) - A'K(i+1)\underline{x}^*(i+1) - A'\underline{g}(+1) \qquad (1.22)$$

After rearraning terms eq. (1.22) can be written as

$$(I+A)'K(i+1)x^*(i+1) + (I+A)'\underline{g}(i+1) - K(i)x^*(i) - \underline{g}(i) + $$
$$+ Q(i)\underline{x}^*(i) - Q(i)\overline{\underline{x}}(i) = 0 \qquad (1.23)$$

Substituting eq. (1.20) into (1.23) for $\underline{x}^*(i+1)$ and rearranging terms, we get

$$I+A)'K(i+1)E^{-1}(i)(I+A)\underline{x}^*(i) - (I+A)'K(i+1)E^{-1}BR^{-1}(i)B'\underline{g}(i+1)$$
$$+ (I+A)'K(i+1)E^{-1}(i)B\overline{\underline{u}}(i) + (I+A)'\underline{g}(i+1) - K(i)\underline{x}^*(i) - \underline{g}(i)$$
$$+ Q(i)\underline{x}^*(i) - Q(i)\overline{\underline{x}}(i) = 0 \qquad (1.24)$$

We can obtain one solution to (1.24) above by requiring

$$I+A)'K(i+1)E^{-1}(i)(I+A)\underline{x}^*(i) - K(i)\underline{x}^*(i) + Q(i)\underline{x}^*(i) = 0 \qquad (1.25)$$

(and

$$-(I+A)'K(i+1)E^{-1}(i)BR^{-1}(i)B'\underline{g}(i+1) + (I+A)'K(i+1)E^{-1}(i)B\overline{\underline{u}}(i)$$
$$+ (I+A)'\underline{g}(i+1) - \underline{g}(i) - Q(i)\overline{\underline{x}}(i) = 0 \qquad (1.26)$$

Eq. (1.25) yields (noting that the trivial solution $\underline{x}(i) = 0$, for all i, is rejected)

$$K(i) = (I + A)'K(i + 1)E^{-1}(i)(I + A) + Q(i) \qquad (1.27)$$

And eq. (1.26) can be rewritten as

$$\underline{g}(i) = -(I + A)'K(i + 1)E^{-1}(i)BR^{-1}(i)B'\underline{g}(i + 1) +$$

$$+ (I + A)'K(i + 1)E^{-1}(i)B\,\bar{\underline{u}}(i) + (I + A)'\underline{g}(i + 1) - Q(i)\bar{\underline{x}}(i) \qquad (1.28)$$

Eqs. (1.27) and (1.28) are Riccati type equations which can be solved backwards in time given the boundary conditions (from 1.16)

$$K(N) = Q(N), \quad \underline{g}(N) = -Q(N)\bar{\underline{x}}(N)$$

It is obvious from the above equations that the computation of the matrix sequence $\{E^{-1}(i)\}$, ... precedes the computation of $\{K(j)\}$, which in turn precedes the computation of the sequence $\{\underline{g}(j)\}$, $j = N - i$, $i = 0, \ldots, N - 1$.

The optimal state trajectory is then computed from (1.20). The optimal control sequence is in turn computed from (1.18), i.e.

$$\underline{u}^*(i) = \bar{\underline{u}}(i) - R^{-1}(i)B'\underline{p}(i+1)$$

$$\Rightarrow \underline{u}^*(i) = -R^{-1}(i)B'K(i+1)\underline{x}^*(i+1) - R^{-1}(i)B'\underline{g}(i+1) + \bar{\underline{u}}(i) \qquad (1.29)$$

The element of the matrix $-R^{-1}(i)B'K(i + 1)$ are known as the time-varying feedback coefficients, or system gains. The vector $-R^{-1}(i)B'\underline{g}(i + 1) + \bar{\underline{u}}(i)$ is usually referred to as the vector of intercepts. Substitution of eq. (1.29) into the system transition equation (1.12) finally yields eq. (1.20) which often is written as

$$\underline{x}(i + 1) - \underline{x}(i) = [E^{-1}(i)(I + A) - I]\underline{x}(i) - E^{-1}(i)BR^{-1}(i)B'\underline{g}(i + 1)$$

$$+ E^{-1}(i)B\bar{\underline{u}}(i) \qquad (1.29a)$$

The presentation by (1.29a) above is known as the closed-loop system in distinction from (1.12) which is referred to as the open-loop situation. It is varified from (1.29) above that a linear system with a quadratic cost functional yields a linear control law. In the case when the process is stochastic, the derivation of the Riccati type equations follows the same steps after adding a white noise vector to the system transition equation (1.12).

It is noted that we have considered the more general case where the weighting matrices are time-varying. In the developed computer program these matrices, as wellas matrices K, $BR^{-1}B'$, $R^{-1}B'$ and E^{-1} are treated as 3-dimentional matrices, the first dimension referring to the number of the discrete time instants considered.

Observing eq. (1.15a) we see that letting $Q_{ii}(N)$ become larger, approaching infinity * (i.e. putting an infinite cost on having $\underline{x}^*(N)$ stay away from $\overline{\underline{x}}(N)$, is equivalent to constraining $\underline{x}^*(N)$ to coincide with $\overline{\underline{x}}(N)$. This fact allows treatment of a two-point-boundary-value control problem by merely adjusting the weights in the Q(N) matrix.

So far we have shown the close relations between the K - T conditions and the discrete minimum principle, the application of which allows the sequential solution of the problem as it would be treated by dynamic programming. However, the problem of dimensionality which is the main disadvantage of the latter approach has led to the development of more efficient methods, like the minimum (maximum) principle, for obtaining numerical solutions to multivariable optimal control problems. It is noted at this point that neither dynamic programming nor the application of the minimum principle yield the closed-form solution which can be obtained through using another procedure * which is based upon the properties of the generalized inverse (pseudoinverse matrix).

2.3.1. Few Points Regarding the Co - State Vectors

Given a cost functional defined by

$$j = \sum_{i=1}^{N} S(\underline{x}(i), \underline{u}(i-1))$$

it is assumed that j* is the optimum (minimum) cost that corresponds to $\{\underline{u}^*(i-1)\}$ and $\{\underline{x}^*(i)\}$, i = 1, ..., N. Then

$$j^* = \sum_{i=1}^{N} S(\underline{x}^*(i), \underline{u}^*(i-1))$$

In the problem discussed previously S is quadratic but this is not necessarily so for any optimal control problem.

---

* We have found in practice that weights in the visinity of 0.9E6 will yield the desired result.

** For details see : A. Lazaridis, «Optimal Planning for the Cattle Industry in Greece; An Application of Optimal Control», op. cit., pp. 156 - 169.

This procedure is practically applicable for discrete-time optimal control problems of moderate size since the solution is not derived sequentially.

Now define

$$j^*(\underline{x}(t), t) = \sum_{i=t}^{N} S(\underline{x}^*(i), \underline{u}^*(i-1)) \qquad t = 1, \ldots, N$$

In other words $j^*(\underline{x}(t), t)$ denotes the optimal cost that would result if the system is to be controlled from time $i = t$ to the terminal time $i = N$. And, among other things, this is a function of $\underline{x}(t)$, the particular state that the system happens to be at time t.

One of the fundamental results of the Hamilton - Jacobi theory states that if the functional from for $j^*(\underline{x}(t), t)$ can be obtained, then the optimal co-state vector at time t is given by

$$\underline{p}^*(t) = \frac{\partial}{\partial \underline{x}(t)} j^*(\underline{x}(t), t) \tag{1.30}$$

Thus each co-state variable at time t can be interpreted as the marginal cost resulting from a small change in the value of the corresponding state variable at time t.

It is noted that at the terminal time $(i = N)$, eq. (1.30) reduces to the standard transversality condition

$$\underline{p}^*(N) = \frac{\partial}{\partial \underline{x}_N^*} j^*(\underline{x}_N, N)$$

i.e.

$$\underline{p}^*(N) = \frac{\partial}{\partial \underline{x}_N^*} M[\underline{x}^*(N)] \tag{1.31}$$

From eq. (1.31) it is clear that for the terminal time the interpretation of the co-state variables is similar to that of the Lagrange multipliers in a static optimization problem. At any other time, however, the interpretation is rather different. From eq. (1.30) it is concluded that if we change the system state at time t by a small amount $\Delta\underline{x}(t)$, then the additional (positive or negative) cost that will result from controlling the system optimally up to the terminal time, is given by

$$\Delta j^*(\underline{x}(t), (t) = \underline{p}^{*\prime} \Delta \underline{x}(t)$$

Thus the co-state variables are dynamic shadow prices that measure the marginal cost of each state variable, indicating, therefore, which of the latter ones are more or less critical — at any particular time instant — to the cost of the optimal policy.

3. Conversion of the Conventional Econometric Model to the Control System Format. (System Representation in State Space)

It is assumed that the econometric model under consideration has the following structural form

$$\underline{y}(t) = H_0\underline{y}(t) + H_1\underline{y}(t-1) + H_2\underline{y}(t-2) + ,\ldots + H_k\,\underline{y}(t-k) +$$

$$+ D_0\underline{z}(t) + D_1\underline{z}(t-1) + ,\ldots, + D_q\,\underline{z}(t-q) + \underline{b} + \underline{\varepsilon}(t) \qquad (1.32)$$

where

$\underline{y}$ is the vector of the endogenous variables

$\underline{z}$ is the vector of the exogenous variables

$\underline{b}$ is the vector of constants

$\underline{\varepsilon}$ is the vector of the uncorrelated disturbances, whose non-zero elements correspond to the stochastic relations, and H_i and D_j ($i = 0, 1, \ldots, K$; $j = 0, 1, \ldots, q$) are coefficient matrices of proper dimension. These parameters have been estimated by standard econometric techniques.

It is further assumed that the policy variables are a subset of the exogenous variables. Using the lag operator L, and denoting $I - H_0$ by ${}_0A$, (1.32) can be re-written as

$$\begin{bmatrix} {}_0A & & & & & & \\ & I & & & & & \\ & & \ddots & & & & \\ & & & I & & & \\ \hline & & & & I & & \\ \hline & & & & & I & \\ & & & & & & \ddots \\ & & & & & & & I \end{bmatrix} \begin{bmatrix} \underline{y}(t) \\ L\underline{y}(t) \\ \vdots \\ L^{k-1}\underline{y}(t) \\ \underline{z}(t) \\ L\underline{z}(t) \\ \vdots \\ L^{q-1}\underline{z}(t) \end{bmatrix} = \begin{bmatrix} H_1 \cdots H_{k-1}\, H_k & D_1 \cdots D_{q-1}\, D_p \\ I & \\ \ddots & \\ I & \\ \hline 0 \ldots 0 \quad 0 & 0 \ldots \quad 0 \quad 0 \\ \hline & I \\ & \ddots \\ & I \end{bmatrix}$$

$$\begin{bmatrix} \underline{y}(t-1)\\ \cdot \\ \cdot \\ \cdot \\ \underline{y}(t-k+1)\\ \underline{y}(t-k)\\ \underline{z}(t-1)\\ \cdot \\ \cdot \\ \cdot \\ \underline{z}(t-q+1)\\ \underline{z}(t-q)\end{bmatrix} + \left[\begin{array}{cc} \underline{b} & D_0 \\ 0 & \\ \cdot & \\ \cdot & \\ \cdot & \\ \cdot & \\ \hline 0 & I \\ \hline 0 & \\ \cdot & \\ \cdot & \\ \cdot & \\ 0 & \end{array}\right] \begin{bmatrix}1\\ \underline{z}(t)\end{bmatrix} + \begin{bmatrix}\underline{\varepsilon}(t)\\ O\\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ O\end{bmatrix} \qquad (1.33)$$

Eq. (1.33) can be written in a compact form as

$$A_1 \underline{x}(t) = A_0 \underline{x}(t-1) + B_1 \hat{\underline{u}}(t) + \hat{\underline{w}}(t)$$

By shifting to $t+1$ we get

$$A_1 \underline{x}(t+1) = A_0 \underline{x}(t) + B_1 L^{-1} \hat{\underline{u}}(t) + L^{-1} \hat{\underline{w}}(t) \qquad (1.34)$$

with the anticipation that L^{-1} denotes the linear advance operator such that $L^{-n}\, y(t) = y(t+n)$. Eqs. (1.33), (1.34) indicate the input requirements for the computer program RICCATI.

Matrices A_1, A_0 and B_1 must be input individually, row-wise and in the indicated order (see also next section).

The system in (1.34) is transformed to the conventional control system format by premultiplying (1.34) throughout by A_1^{-1}, to yield

$$\underline{x}(t+1) = A\underline{x}(t) + B\underline{u}(t) + C\underline{w}(t) \qquad (1.35)$$

where $\underline{u}(t) = L^{-1}\hat{\underline{u}}(t)$ and $\underline{w}(t) = L^{-1}\hat{\underline{w}}(t)$.

Note that the system in (1.34) and the one described by (1.35) are equivalent if A_1 is non-singular. This assumption is satisfied iff matrix ${}_0A$ (i.e. $I - H_0$) is invertible.

Note also that with the presented formulation the control vector consists of flexible control variables (i.e. the actual policy variables) and inflexible ones (i.e. all the other exogenous variables including the unit corresponding to the constant terms). Under the above consideration one has to properly adjust the weight which refer to the inflexible inpust, so that these variables track exactly their nominal paths.

4. Input Description for the RICCATI Program

Data items may be input in any format, separated by at least one space. The data required must be input in the following order :

1. Two integers.

The first one simply identifies the «run» number with a given set of data. The second (ISSTO standing for IS STOCHASTIC?) is used to identify whether the deterministic problem is only to be solved (ISSTO = 0) or both the deterministic and the stochastic case are to be considered in the same run (ISSTO ≠ 0).

2. Three integers identifying the dimension of the state vector the dimension of the control vector and the total number of the discrete time instants considered (i.e. n, m, N).

3. The following free format specifications are read in each on a new record starting from the first column

(nFO.O)

(mFO.O)

(NFO.O)

4. Eight integers IA1, IA0, IB1, IUBAR, IxBAR, IW, IQ, IR, referring to the eight matrices which consitute the main block of data for the RICCATI program. The value of each of the above integers is either O or ≠ 0, depending upon the user's option to feed all the elements of a matrix (even the repeated elements in which case the corresponding integer of the ones specified above is set equal to 0) or to call for a subrouting which reads a matrix with repeated elements in each row (in this case the corresponding integer is ≠ 0). It is noted that when this particular subroutine is called then the input format for the corresponding matrix which is to be read needs to be adjusted accordingly.

Example :

Assume that matrix A_1 (see eqs. (1.33), (1.34)) is

$$\begin{bmatrix} 2.5 & 0. & 0. & 0. \\ 1. & 1. & 0. & 0. \\ 0. & 0. & 1. & 0. \\ 0. & 0. & 0. & 1. \end{bmatrix}$$

then if the integer corresponding to matrix A_1, i.e. IA1, is set equal to zero, then matrix A1 has to be fed as it stands above (mode I). In case when one wants to avoid feeding repeated elements and IA1 is set equal to, say, 3, then matrix A_1 must be fed in the following way (mode II).

```
3                                    )
(3FO.O)                              |  1st row
2.5    0.    1000003.                |
4
(4FO.O)                              |  2nd row
1.    1000002.    0.    1000002.     |
4                                    )
(4FO.O)                              |  3rd row
0.    1000002.    1.    0.           |
3
(3FO.O)                              |  4th row
0.    1000003.    1.                 )
```

The program identifies that the element proceding the number 100000n. is repeated (n — 1) times.

This facility is worthy to be used for highly - dimensioned cases only.

The eight integers as they have been displayed above refer to the following matrices

IA1 to A_1 $(n \times n)$

IA0 to A_0 $(n \times n)$

IB1 to B_1 $(n \times m)$

IXBAR to the $(n \times N)$ matrix XBAR, whose columns are the nominal values for the state variables

IUBAR to the $(m \times N)$ matrix UBAR whose columns are the nominal values for the control variables

IW to the $(n \times N)$ matrix W whose columns refer to the simulated noise vector for each of the time instants considered.

IQ to the $(n \times n)$ weight matrices Q(i) for $i = 1, \ldots, N$

IR to the $(m \times m)$ weight matrices R(i) for $i = 0, \ldots, N-1$

It is recalled that Q and R are the weighting matrices (see eq. (1.14a), section 2.3). The more general case where Q and R are diagonal is explained in details below. In case that crossproducts are also penalized in the cost functional one must be sure that the corresponding weighting matrix has been transformed to a symmetric one *.

5. Matrix A1 (mode I or mode II)

6. Matrix A0 (» » » » »)

7. Matrix B1 (» » » » »)

8. Initial conditions (i.e. $\underset{\cdots}{x}$ (0))

9. XBAR (i.e. $\underset{\cdots}{\bar{x}}$ (j) for j = 1,.... N) in the form of an n × N matrix (mode I or mode II). The j[th] column of this matrix refers to the nominal values of the state variables, ($\underset{\cdots}{\bar{x}}$ (j)) specified for the j[th] time instant.

10. UBAR (i.e. $\underset{\cdots}{\bar{u}}$ (j), for j = 0,..., N — 1) in the form of an m × N matrix (mode I or mode II). Also the j[th] column of this matrix refers to the nominal values of the control vector at time j (i.e. $\underline{\bar{u}}$ (j)).

11. W (i.e. $\underset{\cdots}{w}$(j), j = 1, ..., N) in the form of an n × N matrix (mode I or mode II). Again the j[th] column of this matrix refers to the simulated ** noise vector, $\underset{\cdots}{w}$ (j), for the j[th] time instant. It is noted that if ISSTO = 0 matrix W must be omitted.

12. Penalty coefficients regarding the state variables.

a) Cross - products are also penalized. In this case a string of N matrices each of dimension (n × n) must be fed under the mode II (i.e. IQ ≠ 0).

b) None of the cross - products is penalized. In this case the weight matrices Q(i), i = 1, ..., N, are diagonal. Hence only the diagonal elements are fed in the form of an n × N matrix, say Q_1. The i[th] row (i = 1, ..., N) of Q_1 refers to the main diagonal of the Q(i) matrix. Note that Q_1 is read under the mode I.

13. Penalty coefficients regarding the control variables.

a) Cross - products are also penalized. In this case a string of N positive definite matrices, each of dimension (m × m) must be read in under the mode II (IR ≠ 0).

---

* Considering the quadratic form $\underset{\cdots}{x}' Q \underset{\cdots}{x}$ where Q is not symmetric, we can write $\underline{x}' Q \underline{x} = \underset{\cdots}{x}' Qx$ where Q = 1/2(Q + Q') is symmetric.

** Random variates conforming to the stochastic properties assumed for the disturbances of the econometric model can be generated from the covariance matrix of the residuals through using Cholesky's factorization. For details see : A. Lazaridis, «Optimal Planning for the Cattle Industry in Greece; An Application of Optimal Control», op. cit., pp. 85 - 85a (footnote 43).

b) None of the cross - products is penalized. As in the previous case only the diagonal elements are fed in the form of an $m \times N$ matrix, say, R_1. Again the i th row of R_1 refers to the main diagonal of the R(i) matrix. Matrix R_1 is read in under the mode I.

14. An integer (MORE) which is used to identify whether the calculations are to be repeated with the same basic data and a new set of weights (MORE $\neq 0$) or the control is to be transfered to the STOP FORTRAN statement (MORE = 0). If MORE.NE.O, the new weights are read in according to specifications 12 and 13 above. Finally MORE has to be set equal to zero.

Example :

Consider the system described by the difference equation

$$y(t) = 0.02y(t) + 0.4y(t-1) + 0.01y(t-3) - 2.5z(t-1) +$$

$$+ 0.002z(t-2) + 11.7 + \varepsilon \qquad (1.36)$$

Eq. (1.36) can be put into the state variable form by introducing artificial variables using the linear lag operator L. [i.e. $L^n \underline{x}(t) = \underline{x}(t-n)$]

$$\begin{bmatrix} 0.98 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} y(t) \\ Ly(t) \\ L^2y(t) \\ Lz(t) \end{bmatrix} = \begin{bmatrix} 0.4 & 0 & 0.01 & 0.002 \\ 1 & & & \\ & 1 & & \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y(t-1) \\ Ly(t-1) \\ L^2y(t-1) \\ Dz(t-1) \end{bmatrix} +$$

$$+ \begin{bmatrix} 11.7 & -2.5 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1. \\ z(t-1) \end{bmatrix} + \begin{bmatrix} \varepsilon \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad (1.37)$$

Shifting to t + 1, eq. (1.37) can be written as

$$A_1 \underline{x}(t+1) = A_0 \underline{x}(t) + B_1 \underline{u}(t] + \underline{w}$$

with the anticipation that $\underline{x}$ and $\underline{u}$ are identified as the state and control vector respectively.

It is assumed that

— The initial conditions are

$$\underline{x}(o) = [\,5. \quad 4.8 \quad 4.1 \quad 2.5\,]'$$

— For a total number of discrete - time instants N = 5, the nominal values for the state vector are :

XBAR =

$\bar{x}(1)$	$\bar{x}(2)$	$\bar{x}(3)$	$\bar{x}(4)$	$\bar{x}(5)$
7.	8.	9.	10.	11.
4.8	7.	8.	9.	10.
4.	4.5	5.	7.	8.
3.	3.6	4.2	4.9	5.5

and for the control vector are :

UBAR =

$\bar{u}(0)$	$\bar{u}(1)$	$\bar{u}(2)$	$\bar{u}(3)$	$\bar{u}(4)$
1.	1.	1.	1.	1.
3.	3.6	4.2	4.9	5.5

— The simulated noise vectors are :

W =

$\underline{w}(1)$	$\underline{w}(2)$	$\underline{w}(3)$	$\underline{w}(4)$	$\underline{w}(5)$
—0.03	0.1	0.0009	—0.0911	0.002
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

— The weighting matrices which assumed diagonal are as follows :

Q_1' = $(i = 1, \ldots, n)$

$Q_{ii}(1)$	$Q_{ii}(2)$	$Q_{ii}(3)$	$Q_{ii}(4)$	$Q_{ii}(5)$
0.1	0.2	0.2	5.	50.
1.	1.5	1.6	2.	6.
1.	1.5	1.6	2.	3.
4.	10.	10.	10.	20.

$$R_1' = \begin{bmatrix} R_{ii}(0) & R_{ii}(1) & R_{ii}(2) & R_{ii}(3) & R_{ii}(4) \\ \cdots & & & & \cdots \\ 9.E5 & 9.E5 & 9.E5 & 9.E5 & 9.E5 \\ 4. & 10. & 10. & 10. & 20. \\ \cdots & & & & \cdots \end{bmatrix} \quad (i = 1, \ldots, m)$$

The input data for the above example problem must be as follows :

1 1 (i.e. first run with this set of data where the deterministic and the stochastic case are to be considered)

4 2 5 (i.e. n, m, N)

(4FO.O)

(2FO.O)

(5FO.O)

0 1 0 0 0 1 0 0 (i.e. mode II for matrices A_0 and W)

0.98	0.	0.	0.	Matrix A_1
0.	1.	0.	0.	
0.	0.	1.	0.	
0.	0.	0.	1.	

Matrix A_0:

1st row:

4
(4FO.O)
0.4 0. 0.01 0.002

2nd row:

3
(3FO.O)
1. 0. 1000003.

3rd row:

4
(4FO.O)
0. 1. 0. 0.

4th row:

2
(2FO.O)
0. 1000004.

```
11.7          -2.5   ]
 0.            0.    ]
 0.            0.    ]  Matrix B1
 0.            1.    ]

5.     4.8   4.1   2.5   (i.e. initial conditions)
7.     8.    9.    10.   11.
4.8    7.    8.    9.    10.   }
4.     4.5   5.    7.    8.    }  Matrix XBAR (i.e. x(i), i = 1, ..., N)
3.     3.6   4.2   4.9   5.5   }

1.     1.    1.    1.    1.    }  Matrix UBAR (i.e. u (i), i = 0, ..,. N-1)
3.     3.6   4.2   4.9   5.5   }

5                                              }
(5FO.O)                                        }  1st row
-0.03  0.1  0.0009  -0.0911  0.002             }

2                                              }
(2FO.O)                                        }  2nd row
0.          1000005.                           }
                                                           Matrix W
2                                              }
(2FO.O)                                        }  3rd row
0.          1000005.                           }

2                                              }
(2FO.O)                                        }  4th row
0.          1000005.                           }

0.1    1.     1.     4.    }
0.2    1.5    1.5    10.   }
0.2    1.6    1.6    10.   }  Matrix Q1
5.     2.     2.     10.   }
50.    6.     3.     20.   }

900000.          4      |
900000.          10.    |
900000.          10.    |  Matri  k
900000.          10.    |
900000.          20.    |

                 0   (MORE)
```

Additional Remarks

RICCATI was created for use on the 1906A ICL computer in standard FORTRAN and only its binary code (BINRICCATI) is accessible. It can be run in two forms : either by card input or by use of a data file**. In either case the substantial command is PROG LOAD BINRICCATI with a time parameter 1 MINS or more *. Under the current (GEORGE IV) operating system, the MZ requirement is 180K **.

Current Limitations

$n \leqslant 20$

$2 \leqslant m \leqslant 20$

$N \leqslant 20$

---

* For a test problem with $\underline{x}$ 9-dimensional, $\underline{u}$ 8-dimensional, N = 20, and two sets of weights where the deterministic and the stochastic cases were considered, the required time was 41 secs.

** These specifications are subject to revision, depending upon the available MACRO regarding the University of Thessaloniki computer unit. Final details will be released by the end of October 1978.

REFERENCES

1. Canon, M. D., C. D. Culumm and Polak, E. Theory of Optimal Control and Mathematical Programming, McGraw-Hill Book Company, New York, 1970.

2. Chow, G. C., Analysis and Control of Dynamic Economic Systems, John Wiley and Sons, Inc., New York, 1975.

3. Day, S. A., Fortran Techniques, Cambridge University Press, 1974 (repr.).

4. Friedman, B. M., Methods in Optimization for Economic Stabilization Policy, North-Holland Publishing Company, Amsterdam, 1973.

5. Gelfand, I. M. and Fomin, S. V., Calculus of Variations, Prentice-Hall, Englewood Cliffs, New Jersey, 1963.

6. Halkin, H., Jordan, B.W., Polak, E. and Rosen, J.B., «Theory of Optimum Discrete-Time Systems», 3rd IFAC Conference, London, 1963.

7. Kuester, J. L. and Mize, J. H., Optimization Techniques with Fortran, McGraw-Hill Book Company, New York, 1973.

8. Kuhn, H. W. and Tucker, A. W., «Nonlinear Programming», Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability. University of California Press, Berkeley, 1951, pp. 481-492.

9. Kuo, B. C., Digital Control Systems, SRL Publishing Company, Champaign, Illinois, 1977.

10. Land, H. A. and Powell, S., Fortran Codes for Mathematical Programming, John Wiley and Sons, Inc., New York, 1973.

11. Lazaridis, A., «Optimal Planning for the Cattle Industry in Greece; An Application of Optimal Control», Ph. D. Thesis, University of Birmingham (England), 1977.

12. Mangasarian, O. L. and Fromovitz, S., «A Maximum Principle in Mathematical Programming. In A. V. Balakrishnan and L. W. Neustadt (ed.), Mathematical Theory of Control, Academic Press, New York, 1967, pp. 85-95.

13. Organick, E. I. and Meissner, L. P., FORTRAN IV, Addison-Wesley Publishing Company, Second Edition, Reading, Massachusetts, 1974.

14. Pindyck, R. S., Optimal Planning for Economic Stabilization. The Application of Control Theory to Stabilization Policy, North-Holland Publishing Company, Amsterdam, 1973.

15. Rosen, J. B., «Optimal Control and Convex Programming», MRC Technical Report No. 547 University of Wisconsin, Madison, February 1965.