DESIGN OF OPTIMAL SAMPLED DATA LINEAR REGULATOR

by

Ömer AYZAN

Submitted to the Faculty of Engineering
in Partial Fulfillment for the
Requirements of the Degree of
MASTER OF SCIENCE
in
ELECTRICAL ENGINEERING

Boğaziçi University

October 1979

This thesis has been approved by:

Dr. Yorgo Istefanopulos
(Thesis Supervisor)

Prof. Dr. Ahmet Dervişoğlu

Dr. Ali Çanar

ACKNOWLEDGEMENTS

I wish to express my gratitude to my thesis supervisor Dr. Yorgo Istefanopulos, who devoted his valuable time for discussions throughout the thesis. I am indebted to Dr. Istefanopulos for his encouraging talks and guidance without which the completion of this thesis would not have been possible.

My special thanks go to Prof. Dr. Ahmet Dervişoğlu and Dr. Ali Çınar for their helpful comments and suggestions in the improvement of this study.

I appreciate the friendship of all those friends with whom I shared my last two years, especially the help of Can Göksel and Mehmet Etişkol, who were my moral support throughout this study.

I gratefully acknowledge the patience of Gülsen Karşit in her careful typing.

ÖZETÇE

Doğrusal ve zamanla değişmeyen sistemler için en iyi örneklenmiş denetiminin eldesi ayrık en az ilkesi uygulanarak ve karesel maliyet ölçütü kullanılarak gerçekleştirildi. Ayrık en az ilkesi, sürekli sistem durum denklemleri ve integral performans indisleri ayrık duruma getirilerek uygulandı. En iyi denetimin örnekleme hızı ile ilintisinin çözümsel ifadesini bulmak üzere geniş bir çalışma yapıdı. Böylece örnekleme hızı değişiminin kapalı çevre sistem performansını nasıl etkilediği araştırıldı.

En iyi örnekleme hızını ve en iyi geri besleme kazanç dizeyini saptamak için yukarıdaki kurama bağlı kalarak sayısal yöntemler ve bir bigisayar programı geliştirildi. Çeşitli doğrusal ve zamanla değişmeyen süreçlerde örnekleme hızı ve en iyi denetimi içeren sayısal sonuçlar elde edildi ve her sistemle ilgili maliyet hesaplandı. Ek olarak, tanımlanan zaman aralığında denetimin ve sistem durumlarının zamana bağlı değişimleri çizildi.

ABSTRACT

Optimal Sampled-data controls for linear time invariant processes with quadratic cost criteria are determined through the application of the discrete minimum principle. In order to apply the discrete minimum principle both continuous system's state equations and integral performance indices are discretized. An extensive work is done to find an analytical expression for the dependance of the optimal control on the sampling period. Thus the effect of changing the sampling time upon the closed loop system's performance is investigated.

Based on the theory cited above original numerical methods and a computer program have been developed, to compute optimum sampling period and the matrix of optimum feedback gains. Then for various linear time invariant processes numerical results are obtained for the sampling period and the optimal control and, the associated costs are evaluated. Additionally the trajectories of the states, and the controls are plotted within the time interval of interest.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

One of the most frequent problems in the field of optimal control is the optimal linear regulator problem.

It is well known that the optimal control u*(t) for $t \varepsilon [t_0, t_f]$, for linear systems subject to a quadratic performance criteria is generated by a linear feedback control law of the form:

$$\underline{u}^*(t) = - \underline{G}^*(t)\ \underline{x}(t)$$

where $\underline{x}(t)$ is the current state of the system and $\underline{G}^*(t)$ is the matrix of the feedback gains. The elements of the matrix $\underline{G}^*(t)$ are computed from the solution of a nonlinear matrix differential equation, called the matrix Riccati Equation.

In spite of the apparent mathematical simplicity there are certain engineering difficulties associated with the realization and implementation of the time varying feedback gains. It is generally not possible to compute $\underline{G}^*(t)$ accurately in an on line manner.

These practical considerations impose further restrictions on the controller- The goal is to determine a linear feedback law which is relatively easy to implement, but which results in a system performance closest to the optimal. Such controllers are referred to as suboptimal control schemes in which a trade-off between mathematical optimality and practical usefulness is made by constraining the structural form of the time varying feedback gains, while leaving various free parameters to be chosen in an optimal manner. One form of the feedback gain structures that leads to a suboptimal control for a linear regulator system is the piecewise constant gains which are relatively easy to implement.

The piecewise constant gains can be obtained through a sample and hold mechanism, and the control schemes established in this manner form sampled data control systems.

In addition to the practical restrictions stated above that favor the use of sampled data systems in which the sampling operation occurs between the plant and the controller, the nature of the system itself may dictate the use of sampled data. Among such systems those that have a telemeter link in the feedback loop, or use a single instrument to monitor several variables in a sequential manner may be mentioned.

When the sampling operation is introduced to the system, one has to choose a convenient sampling strategy if it is not inherently imposed by the system itself. Due to their relative simplicity, sampled data control systems generally have fixed sampling frequencies. But several non-uniform sampling schemes have been proposed in the literature. The rationale behind these sampling schemes is to achieve a given performance using more information about the system structure. In this case, the problem is finding a system signal and the functions of that signal for controlling the variable frequency sampler, so that over a given time interval fewer samples will be needed with the variable frequency system than with a fixed frequency-system while maintaining essentially the same response characteristics. The cost of the savings produced by reducing the overall number of samples, is increased complexity of the adaptive sampling systems.

In addition to the simplicity in implementation, a fixed rate sampled systems facilitates the use of time sharing strategies which achieve economy in the use of equipment.

In this thesis, an algorithm has been developed to compute optimum sampling period and the matrix of optimum feedback gains for a linear system employed in a regulator problem. The structure of the thesis is as follows.

In Chapter 2 the optimal regulator problem is studied for the case of continuous time system and sampled data systems. In the latter case the regulator problem is transformed into an equivalent discrete-time problem and thus through the application of the Discrete minimum principle the Matrix-Difference Riccati Equation is derived. Then the optimal control law is determined as a function of the solution to the non-linear Riccati matrix difference equation. Using this control law the optimal closed-loop sampled data system design is achieved. Finally, the stability of the optimal closed loop system is examined by considering the location of its eigenvalues.

In Chapter 3, the problem of choosing a suitable sampling period and its effects on the closed loop optimal system's performance is treated. The behavior of the optimal cost as a function of the sampling period is investigated analytically. Experimental results based on the computer studies are also presented.

In Chapter 4, the overall package program which can be used in the design of optimal sampled-data or discrete regulators is presented. A simplified flow chart which describes the general features of the program flow is given. The use of the program and subroutines and their properties are explained briefly.

Finally in Chapter 5 concluding remarks and suggestions for further study and development are given.

CHAPTER 2

OPTIMAL SAMPLED DATA LINEAR REGULATORS

Consider the linear time invariant dynamical system, modeled as

$$\overset{o}{\underline{x}}(t) = \underline{A}\ \underline{x}(t) + \underline{B}\ \underline{u}(t) \qquad (2.1)$$

$$\underline{x}(t_o) = \underline{x}_o \qquad (2.1a)$$

$$\underline{y}(t) = \underline{C}'\ \underline{x}(t) + \underline{D}\ \underline{u}(t) \qquad (2.2)$$

where

$\underline{x}(t)$ - state vector $\varepsilon\ R^n$

$\underline{y}(t)$ - output vector $\varepsilon\ R^m$

$\underline{u}(t)$ - input vector $\varepsilon\ R^r$

and

$\underline{A}$ - (nxn) system matrix

$\underline{B}$ - (nxr) input matrix

$\underline{C}$ - (nxm) output matrix

In order to simplify the problem it is assumed that the system is time invariant, and therefore $\underline{A}$, $\underline{B}$ and $\underline{C}$ are constant matrices. Also for physical systems those matrices are assumed to be finite dimensional.

It is further assumed that $\underline{D} = \underline{0}$.

With the above system equations the regulator problem can qualitatively be stated as follows:

Suppose that initially the plant output as given by (2.2) or any of its derivatives, is non-zero. Provide a plant input to bring the output and its derivatives to zero.

In other words, the problem is to apply a control to take the plant from a nonzero state preferably as fast as possible to the zero state. If the constant matrix pair $\underline{A}$ and $\underline{B}$ $[\underline{A},\underline{B}]$ is completely controllable, then this objective can be achieved. The definition of complete controllability requires that there exists a control taking any nonzero state $x(t_0)$ at time t_0 to the zero state at some time t_f. In fact, since A and B are constant, t_f can be taken as close to t_0 as desired. However, the closer t_f is to t_0 the greater is the amount of control energy (and the greater is the magnitude of control) required to effect the state transfer. In any engineering system an upper bound is set on the magnitude of the various variables in the system by practical considerations.

Therefore, t_f can not be taken arbitrarily close to t_0 without exceeding these bounds. In addition the actual control can not be implemented as a linear feedback law for finite t_f unless one is prepared to tolerate at t_f.

Any other control scheme for which one or both of the above objections is valid is equally unacceptable.

In an effort to meet the first objection, it is necessary to keep some measure of control magnitude, such as

$$\int_{t_0}^{t_f} \underline{u}'(t)\ \underline{R}\ \underline{u}(t)dt\ ,$$

bounded during the course of control action, where $\underline{R}$ is a symmetric positive definite matrix.

In engineering problems, however, driving the state near enough to the desired state may be accepted as a satisfactory solution to the control problem. So the aim of achieving the - zero state will be relaxed and it is merely required that the state as measured by some norm should become small, for some fixed time t_f.

The term $\underline{x}'(t)\ \underline{S}\ \underline{x}(t)$, with $\underline{S}$ some positive definite matrix, if made small meets the requirement. Also, it is clearly helpful from the control point of view to have $||\underline{x}(t)||$ (Norm of State Vector) small for any t in the interval over which control is being exercized, and this fact can be expressed as making the

$$\int_{t_0}^{t_f} x'(t)\ Q\ x(t)dt$$

small. Where $\underline{Q}$ is a symmetric positive definite matrix.

The desirable properties of a regulator system may be summarized as follows:

Property 1: The regulator system should involve a linear control law, of the form

$$\underline{u}(t) = -\underline{G}\ \underline{x}(t)$$

Property 2: The regulator scheme should ensure the smallness of quantities such as

$$\int_{t_o}^{t_f} \underline{u}(t)\ \underline{R}\ \underline{u}(t)\ dt\ ,$$

$$\underline{x}'(t_f)\ \underline{S}\ \underline{x}(t), \text{ and } \int_{t_o}^{t_f} \underline{x}'(t)\ \underline{Q}\ \underline{x}(t)dt$$

where $\underline{R}$, $\underline{S}$, and $\underline{Q}$ have the positivity properties mentioned earlier.

Now, if the specified control input over the time interval of definition of the system $[t_o, t_f]$, where t_o is the initial time and t_f is the final time; is $\underline{u}(.)$, then the trajectory of the state generated by the control $\underline{u}$ starting at state x_o at time t_o is given by the equation

$$\underline{x}_u(t) = e^{\underline{A}(t-t_o)}\underline{x}_o + \int_{t_o}^{t_f} e^{\underline{A}(t-\tau)}\underline{B}\ \underline{u}(\tau)d\tau \qquad (2.3)$$

The above expression indicates the entire time response of the system.

If the final time t_f is assumed to be infinite then an ordered set of real numbers, called the time set, can be used to indicate the successive time points.

$$\{t_i\} = \{t_i;\ t_i = t_o + iT;\ i = 0,1,2,\ldots,\} \qquad (2.4)$$

If however the terminal time t_f is finite, an assumption is made that the time interval of interest is subdivided into a sequence of N intervals of equal length T

$$NT = t_f - t_o \qquad \text{or} \qquad t_f = t_o + NT \qquad (2.5)$$

where the constant T is called the sampling period or sampling interval.

Next, a crucial assumption is made on the structure of the time function $\underline{u}(t)$ which requires that the control vector $\underline{u}$ takes some constant value over a particular interval and that changes in the values of $\underline{u}(t)$ occur only at the sampling instant t_i. The constraint imposed above corresponds to the sample and hold operation, therefore,

$$\underline{u}(t) = u(t_o+iT) \triangleq \underline{u}_i \quad \text{for } t_o+iT \leq t \leq t_o+(i+1)T \qquad (2.6)$$

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

A choice for the cost crierion with respect to which the performance of the system is optimized is of the form of a quadratic functional as required by Property 2 mentioned earlier.

$$J[\underline{x}_o, t_o; t_f; u(.)] = \frac{1}{2} < \underline{y}(t_f), \underline{F}\ \underline{y}(t_f)>$$

$$+ \frac{1}{2} \int_{t_o}^{t_f} [<\underline{y}(t), \underline{Qy}(t)> + <\underline{u}(t), \underline{Ru}(t)>] dt \quad (2.7a)$$

The above choice for the form of performance index is appropriate to achieve Property 2. Since the main concern is maintaining the output close to the zero vector a measure is defined in terms of output in (2.7a).

The notation $<\underline{a}, \underline{M}\ \underline{a}>$ is used to denote the inner product of vectors $\underline{a}$ and $\underline{M}\ \underline{a}$. But for simplicity in writing in the foregoing paragraphs $\underline{a}'\underline{M}\ \underline{a}$ will be used, where the superscript prime denotes transposition.

In (2.7a) the matrix $\underline{F}$ is used to weight the terminal deviation, the matrix $\underline{Q}$ to weight the output trajectory deviation, and the matrix $\underline{R}$ to penalize the excessive magnitudes of the control input. $\underline{F}$ and $\underline{Q}$ are positive semidefinite, mxm symmetric matrices not both identically zero, and R is a positive definite, symmetric rxr matrix.

These requirements are necessary to ensure the linearity of the feedback law.

the output $\underline{y}$, from (2.7a) so that the cost functional can be expressed in terms of states as suggested by Property 2, then the cost functional takes the form:

$$J[\underline{x}_o,t_o;t_f;u(.)] = \frac{1}{2} x'(t_f)\underline{C}\ \underline{F}\ \underline{C}'x(t_f)$$

$$+ \frac{1}{2} \int_{t_o}^{t_f} [x'(t)\underline{C}\ \underline{Q}\ \underline{C}'x(t) + \underline{u}'(t)\underline{R}\ \underline{u}(t)]dt \qquad (2.7b)$$

The sampled data optimization problem under consideration can be formally stated as follows:

Given a dynamical system characterized by equations (2.1) through (2.6) with piecewise constant inputs, determine a control sequence

$$\{u_i,\ i = 0,1,\ldots,\ N-1\}$$

that minimizes the quadratic cost functional J of equation (2.7b) while driving the system from an arbitrary initial state to zero state.

This regulator problem can not be solved directly, because the admissible controls are constrained to be piecewise constant.

Nevertheless, it is possible to transform the problem from a constrained one to an unconstrained one by integrating the differential equation and the cost functional and thus going from a continuous time problem to a discrete time one.

THE EQUIVALENT DISCRETE TIME OPTIMIZATION PROBLEM

The optimization problem that was posed at the end of the previous section can now be transformed to an equivalent discrete time one in a form that permits the direct application of the discrete minimum principle.

The transformation is accomplished through the use of the state equation (2.3) evaluated for $t_i \le t \le t_{i+1}$ thus expressing the state at $(i+1)T$ in terms of the state at iT and the constant control $u(iT)$. The resulting discrete time system is obtained as:

$$\underline{x}_{i+1} \equiv \underline{x}(t_{i+1}) = \underline{\Phi}(T)\underline{x}_i + \underline{D}(T)\underline{u}_i ; \; \underline{x}_0 = \underline{x}(t_0) \qquad (2.8)$$

where

$$T = t_{i+1} - t_i \quad \text{for} \quad i = 0,1,\ldots, N-1$$

It is observed that the matrices $\underline{\Phi}$ and $\underline{D}$ are time invariant but depend parametrically on the size of sampling period T. This can be shown as follows:

$$\underline{\Phi}(t_{i+1},t_i) = e^{\underline{A}(t_{i+1}-t_i)} = e^{\underline{A}T} = \underline{\Phi}(T,0) \qquad (2.9)$$

and

$$\underline{D}(t_{i+1},t_i) = \int_{t_i}^{t_{i+1}} \underline{\Phi}(t_{i+1},t)\, \underline{B}\, dt = \int_0^T e^{\underline{A}t}\, \underline{B}\, dt$$

$$= \int_0^T \underline{\Phi}(t,0)\, B\, dt = \underline{D}(T,0) \qquad (2.10)$$

where $\underline{\Phi}(T)$ is the fundamental matrix (STM) and is non singular. In order to simplify the expressions in the subsequent paragraphs, with no conceptual loss in generality, the initial time t_o will be taken to coincide with the origin as was done in equations (2.8) and (2.9) above.

The cost functional of equation (2.7b) can be expressed as the sum over i of N integrals and, if the state equation (2.3) is substituted into each integral and if the fact that $\underline{u}$ is constant over the interval of each integration is taken into account, then the following expression for J is derived (See Appendix A).

$$J[x_o,t_o;t_N] = \frac{1}{2}\,[\underline{x}'_N\ \underline{S}\ \underline{x}_N] + \frac{1}{2}\sum_{i=0}^{N-1} \underline{x}'_i\ \hat{\underline{Q}}\underline{x}_i$$

$$+ 2\ \underline{x}'_i\ \underline{M}\ \underline{u}_i + \underline{u}'_i\ \hat{\underline{R}}\ \underline{u}_i \qquad (2.11)$$

where the weighting matrices $\underline{S}$ $\hat{\underline{Q}}$ $\underline{M}$ and $\hat{\underline{R}}$ respectively given as:

$$\underline{S} = \underline{C}\ \underline{F}\ \underline{C}' \qquad (2.12a)$$

$$\hat{\underline{Q}} = \hat{\underline{Q}}(t_{i+1},t_i) = \int_{t_i}^{t_i} \underline{\Phi}(t,t_i)\ \underline{C}\ \underline{Q}\ \underline{C}'\ \underline{\Phi}(t,t_i)dt$$

$$= \int_o^T e^{\underline{A}'t}\ \underline{C}\ \underline{Q}\ \underline{C}'\ e^{\underline{A}t}\ dt = \hat{\underline{Q}}(T,0) \qquad (2.12b)$$

$$\underline{M} = \underline{M}(t_{i+1},t_i) = \int_{t_i}^{t_{i+1}} \Phi'(t,t_i)\underline{C}\ \underline{Q}\ \underline{C}'\ \underline{D}(t,t_i)dt$$

$$= \int_0^T e^{\underline{A}'t} \underline{C}\ \underline{Q}\ \underline{C}'[\int_0^t e^{\underline{A}s} ds]\underline{B}dt = \underline{M}(T,0)$$

(2.12c)

and finally,

$$\hat{\underline{R}} = \hat{\underline{R}}(t_{i+1},t_i) = \int_{t_i}^{t_{i+1}} [\underline{R}+\underline{D}'(t,t_i)\underline{C}\ \underline{Q}\ \underline{C}'\underline{D}(t,t_i)]dt$$

$$= T\underline{R} + \underline{B}'\{\int_0^T [\int_0^t e^{\underline{A}'s} ds]\underline{C}\ \underline{Q}\ \underline{C}'[\int_0^t e^{\underline{A}s} ds]dt\}\underline{B}$$

$$= \hat{\underline{R}}(T,0) \qquad (2.12d)$$

In order to simplify the appearance of relations the arguments (T,0) of the above transformed matrices are suppressed.

These weighting matrices are time invariant and depend parametrically on the sampling interval. However, if the sampling period is not constant (for instance, adaptive sampling systems, or state dependent sampling systems) then the discrete time system and the weighting matrices would become time varying, that is; they woud depend on the index i even though the continuous-time system was time invariant. Furthermore, it can be shown that if $\underline{R}$ and $\underline{Q}$ are symmetric positive definite and semidefinite respectively, so are $\hat{\underline{R}}$ and $\hat{\underline{Q}}$. This property is essential for the existence of the solution of the Riccati equation which will be derived in the following section. It should also be noted that

$\hat{\underline{Q}}$, $\underline{M}$ and $\hat{\underline{R}}$ can be evaluated numerically for a given sampling period T.

With the above transformations the problem can now be formulated as a discrete optimal control problem.

The equivalent discrete time optimization problem can be restated as follows:

Given the linear discrete time system

$$\underline{x}_{i+1} = \underline{\Phi}\ \underline{x}_i + \underline{D}\ \underline{u}_i\ ;\ \underline{x}_0 = \underline{x}(t_0) \qquad (2.13)$$

determine the control sequence

$$\{u_i^*,\ i = 0,1,\ldots,\ N-1\} \qquad (2.14)$$

and corresponding trajectory $\{\underline{x}_i^*\}$, such that the cost functional

$$J[\{u_i\}] = \frac{1}{2}\ \underline{x}_N'\ \underline{S}\ \underline{x}_N + \frac{1}{2} \sum_{i=0}^{N-1} [\underline{x}_i'\ \hat{\underline{Q}} x_i + 2\ \underline{x}_i'\ \underline{M}\ \underline{u}_i + \underline{u}_i'\ \hat{\underline{R}}\ \underline{u}_i] \qquad (2.15)$$

attains its minimum value.

APPLICATION OF THE DISCRETE MINIMUM PRINCIPLE

At the end of the previous section the discrete optimization problem has been stated, and to avoid redundancy it is not repeated here. Since the objective is to minimize J given by equation (2.15) subject to the constraint equations specified by (2.13), the problem of this type may be treated as the minimization problem involving a function of several variables.

To get the Hamiltonian function the performance index, J of equation (2.15) is augmented through the use of a set of Lagrange multipliers $\{\lambda_i, i=0,1,...,N-1\}$ and the constraint equations (2.13).

Boundary conditions are as follows:

a) Initial time and state are fixed, i.e.,

$$t_o = 0 \qquad \underline{x}(t_o) = \underline{x}_o$$

b) Terminal time may be free or fixed. The former corresponds to an infinite-time regulator problem while the latter to a finite regulator problem.

c) Terminal State is fixed, that is, it is specified as the orgin for the linear regulator problem.

The form for the Hamiltonian is:

$$H = \sum_{i=0}^{N-1} \{ \frac{1}{2} \underline{x}_i' \ \hat{\underline{Q}} \ \underline{x}_i + \underline{x}_i' \ \underline{M} \ \underline{u}_i + \frac{1}{2} \underline{u}_i' \ \hat{\underline{R}} \ \underline{u}_i$$

$$+ \ \underline{\lambda}'_{i+1} \ [\underline{\Phi} \ \underline{x}_i + \underline{D} \ \underline{u}_i - \underline{x}_{i+1}]\} \qquad (2.16)$$

Since the variables are $\underline{x}_i$, $\underline{u}_i$ and $\underline{\lambda}_i$ we seek to obtain the partial derivatives of H with respect to all of the above variables for all values of i, and then equate them to zero.

$$\frac{\partial H}{\partial \underline{x}_1} = \underline{0} \qquad \forall \ i \ ; \ i=0,1,\ldots,N-1 \qquad (2.17)$$

$$\frac{\partial H}{\partial \underline{u}_i} = \underline{0} \qquad \forall \ i \ ; \ i=0,1,\ldots,N-1 \qquad (2.18)$$

$$\frac{\partial H}{\partial \underline{\lambda}_i} = \underline{0} \qquad \forall \ i \ ; \ i=0,1,\ldots,N-1 \qquad (2.19)$$

Equations (2.17) through (2.19) comprises the necessary conditions for H to have a minimum. However, as might be seen from equation (2.16) the individual relations (2.17) through (2.19) involve the differentiation of quadratic terms like $\underline{x}' \ \underline{M} \ \underline{u}$, $\underline{x}' \ \hat{\underline{Q}} \ \underline{x}$, and $\underline{u}' \ \underline{M} \ \underline{x}$.

Each of the above expressions is a scalar, but requires a differentiation with respect to the vector variables $\underline{x}$ and $\underline{u}$.

Now the differentiations as indicated by the necessary conditions will be carried out for i=0,1,...,N-1.

$$\frac{\partial H}{\partial \underline{x}_i} = \hat{\underline{Q}}\ \underline{x}_i + \underline{M}\ \underline{u}_i + \underline{\Phi}'\underline{\lambda}_{i+1} - \underline{\lambda}_i = \underline{0} \quad \forall\ i \qquad (2.20a)$$

$$\frac{\partial H}{\partial \underline{u}_i} = \underline{M}'\underline{x}_i + \hat{\underline{R}}\ \underline{u}_i + \underline{D}'\underline{\lambda}_{i+1} = \underline{0} \quad \forall\ i \qquad (2.20b)$$

$$\frac{\partial H}{\partial \underline{\lambda}_i} = \underline{\Phi}\ \underline{x}_i + \underline{D}\ \underline{u}_i - \underline{x}_{i+1} = \underline{0} \quad \forall\ i \qquad (2.20c)$$

For i = 0:

$$\frac{\partial H}{\partial \underline{x}_o} = \hat{\underline{Q}}\ \underline{x}_o + \underline{M}\ \underline{u}_o + \underline{\Phi}'\underline{\lambda}_1 = \underline{0} \qquad (2.21a)$$

$$\frac{\partial H}{\partial \underline{y}_o} = \underline{M}'\underline{x}_o + \hat{\underline{R}}\ \underline{u}_o + \underline{D}'\underline{\lambda}_1 = \underline{0} \qquad (2.21b)$$

$$\frac{\partial H}{\partial \underline{\lambda}_o} = \underline{\phi}\ \underline{x}_o + \underline{D}\ \underline{u}_o + \underline{x}_1 = \underline{0} \qquad (2.21c)$$

But, when i = N-1, a term with the index N appears in H as given by equation (2.16) so this should also be included in our conditions also,

$$\frac{\partial H}{\partial x_N} = \underline{\lambda}_N = \underline{0} \qquad (2.21d)$$

This condition specifies a boundary value for the set of Lagrange multipliers at time i=N. This boundary value will be used in the solution of equations (2.20) and (2.21).

From (2.20a) and (2.20b):

$$\underline{\lambda}_i = \underline{Q}\,\underline{x}_i + \underline{M}\,\underline{u}_i + \underline{\Phi}\,\underline{\lambda}_{i+1} \tag{2.22}$$

$$-\hat{\underline{R}}\,\underline{u}_i = \underline{M}'\underline{x}_i + \underline{D}'\underline{\lambda}_{i+1} \qquad \text{or}$$

$$\underline{u}_i^* = -\hat{\underline{R}}^{-1}[\underline{M}'\underline{x}_i + \underline{D}'\underline{\lambda}_{i+1}] \tag{2.23}$$

This latter expression for $\underline{u}_i$ comprises an open loop optimal control law for the system (2.13). If this optimal value of $\underline{u}_i$ is substituted into equation (2.20c) the following expression is obtained for the states

$$\underline{x}_{i+1} = \underline{\Phi}\,\underline{x}_i + \underline{D}\,[-\hat{\underline{R}}^{-1}(\underline{M}'\underline{x}_i + \underline{D}'\underline{\lambda}_{i+1})]$$

or

$$\underline{x}_{i+1} = (\underline{\Phi} - \underline{D}\,\hat{\underline{R}}^{-1}\underline{M}')\underline{x}_i - \underline{D}\,\hat{\underline{R}}\,\underline{D}'\underline{\lambda}_{i+1}$$

In order to simplify the results of optimization the following matrix is defined:

$$\underline{\theta} = \underline{\Phi} - \underline{D}\,\hat{\underline{R}}^{-1}\underline{M}' \tag{2.24}$$

then,

$$x_{i+1} = \underline{\theta}\,\underline{x}_i - \underline{D}\,\hat{\underline{R}}\,\underline{D}'\underline{\lambda}_{i+1} \tag{2.25}$$

This optimum open loop control law may now be turned into a feedback control law with the introduction of linear transformation known as the Riccati transformation, which relates the state vector to the Lagrange multipliers vector.

$$\underline{\lambda}_i = \underline{P}_i \; \underline{x}_i \tag{2.26}$$

If this transformation is used to eliminate $\underline{\lambda}_i$ in equations (2.22) through (2.25) the following results are obtained:

$$\underline{P}_i \; \underline{x}_i = \hat{\underline{Q}} \; \underline{x}_i + \underline{M} \; \underline{u}_i{}^* + \underline{\Phi}' \; \underline{P}_{i+1} \; \underline{x}_{i+1} \tag{2.27}$$

$$\underline{u}_i{}^* = - \hat{\underline{R}}^{-1} [\underline{M}' \underline{x}_i + \underline{D}' \; \underline{P}_{i+1} \; \underline{x}_{i+1}] \tag{2.28}$$

$$\underline{x}_{i+1} = [\underline{I} + \underline{D} \; \hat{\underline{R}}^{-1} \underline{D}' \; \underline{P}_{i+1}]^{-1} \; \underline{\theta} \; \underline{x}_i \tag{2.29}$$

Equation (2.28) is substituted into equation (2.27)

$$\underline{P}_i \underline{x}_i = \hat{\underline{Q}} \; \underline{x}_i + \underline{M} \; \{- \hat{\underline{R}}^{-1} [\underline{M}' \underline{x}_i + \underline{D}' \underline{P}_{i+1} \; \underline{x}_{i+1}]\}$$

$$+ \underline{\Phi}' \; \underline{P}_{i+1} \; \underline{x}_{i+1} = [\hat{\underline{Q}} - \underline{M} \; \hat{\underline{R}}^{-1} \underline{M}'] \; \underline{x}_i$$

$$+ [\underline{\Phi}' - \underline{M} \; \hat{\underline{R}}^{-1} \underline{D}'] \; \underline{P}_{i+1} \; \underline{x}_{i+1}$$

Recall that $\underline{\theta}$ was defined as $\underline{\Phi} - \underline{D} \; \hat{\underline{R}}^{-1} \underline{M}'$, then $\underline{\Phi} - \underline{M} \hat{\underline{R}}^{-1} \underline{D}'$ is recognized as $\underline{\theta}'$, since $\underline{R}$ is assumed to be a symmetric matrix. Again for simplicity a matrix $\underline{\Gamma}$ is defined

$$\underline{\Gamma} = \hat{\underline{Q}} - \underline{M}\,\hat{\underline{R}}^{-1}\underline{M}' \qquad (2.30)$$

Finally, the equation can be put in the form of

$$\underline{P}_i\underline{x}_i = \underline{\Gamma}\,\underline{x}_i + \underline{\theta}'\underline{P}_{i+1}\underline{x}_{i+1} \qquad (2.31)$$

Solving for $\underline{u}_i^*$ in terms of $\underline{P}_i$ after eliminating $\underline{x}_{i+1}$ terms by substituting equation (2.29) into (2.28) yields,

$$\underline{u}_i^* = -\,\hat{\underline{R}}^{-1}\{\underline{M}'x_i + \underline{D}'\underline{P}_{i+1}[\underline{I} + \underline{D}\,\hat{\underline{R}}^{-1}\underline{D}'\underline{P}_{i+1}]^{-1}\underline{\theta}x_1\}$$

$$= -\,\{\hat{\underline{R}}^{-1}\underline{M}' + \hat{\underline{R}}^{-1}\underline{D}'\underline{P}_{i+1}[\underline{I} + \underline{D}\,\hat{\underline{R}}^{-}\,\underline{D}'\underline{P}_{i+1}]^{-1}\underline{\theta}\}\underline{x}_i$$

Application of the matrix operations on the above equation, considerably simplifies the expression (see Appendix B I)

$$\underline{u}_i^* = -\,\{\hat{\underline{R}}^{-1}\underline{M}' + [\hat{\underline{R}} + \underline{D}'\underline{P}_{i+1}\underline{D}]^{-1}\underline{D}'\underline{P}_{i+1}\underline{\theta}\}\underline{x}_i$$

$$\underline{u}_i^* = -\,\underline{G}_i\,\underline{x}_i \qquad (2.32)$$

where the feedback gain matrix $\underline{G}_i$ is given by

$$\underline{G}_i = \hat{\underline{R}}^{-1}\underline{M}' + [\hat{\underline{R}} + \underline{D}'\underline{P}_{i+1}\underline{D}]^{-1}\underline{D}'\underline{P}_{i+1}\,\theta \qquad (2.33)$$

This is the sampled data optimal feedback control law. It states that in order to evaluate the constant control for the interval [iT,(i+1)T], only the state at time iT has to be measured.

Alternatively, the elements of the matrix $\underline{G}$ specify the optimal weighting of the states.

In the case of finite time regulator G is a time varying matrix, that is the elements of G change as the index i changes and approaches a constant as the final time t_f approaches infinity, so the regulator becomes an infinite time regulator.

The structure of the optimal sampled-data feedback control system is shown in Figure 1 below:

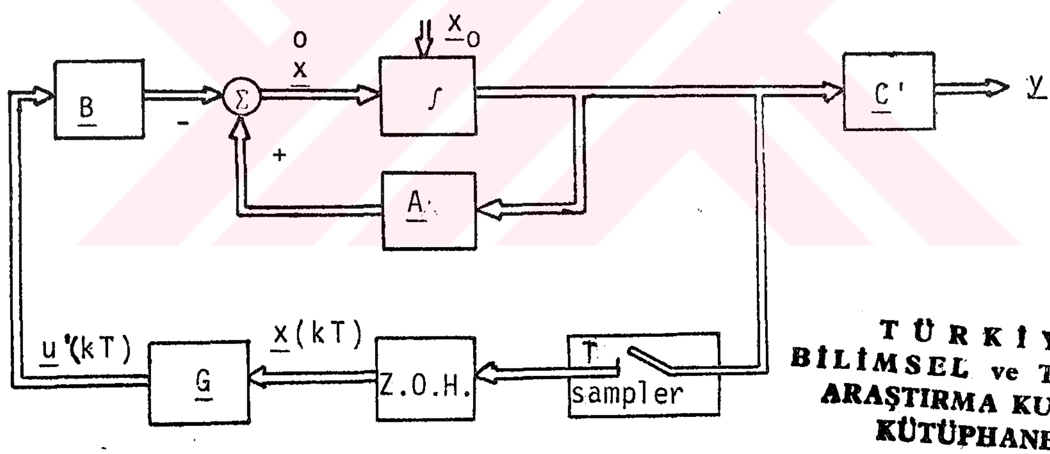


FIGURE 2.1. The Structure of the Optimal Sampled-Data Feedback Control System.

In order to obtain the final form of the Riccati transformed expression equation (2.31) is considered.

Substitution of equation (2.29) into (2.31) for $\underline{x}_{i+1}$ yields:

$$\underline{P}_i\underline{x}_i = \underline{\Gamma}\ \underline{x}_i + \theta' P_{i+1} | I + D\hat{R}^{-1} D' P_{i+1} |^{-1}\ \underline{\theta}\ \underline{x}_i$$

finally,

$$\underline{P}_i = \underline{\theta}' \underline{P}_{i+1} [\underline{I} + \underline{D}\hat{\underline{R}}^{-1} \underline{D}' \underline{P}_{i+1}]^{-1}\ \underline{\theta} + \underline{\Gamma} \qquad (2.34a)$$

After applying matrix inversion lemma (See Appendix B II) on the above equation, the required matrix multiplications are considerably reduced

$$\underline{P}_i = \underline{\theta}' \{ \underline{P}_{i+1} - \underline{P}_{i+1} \underline{D} [\hat{\underline{R}} + \underline{D}' \underline{P}_{i+1} \underline{D}]^{-1} \underline{D}' \underline{P}_{i+1} \} \underline{\theta} + \underline{\Gamma} \qquad (2.34b)$$

This form of the Riccati equation is more useful than equation (2.33a) because it simplifies the necessary calculations during the numerical solution of the problem.

The nxn time varying matrix $\underline{P}_i$ is the solution to the matrix Riccati difference equation (2.33b) with the boundary condition (See Appendix D).

$$\underline{P}_N = \underline{S} \qquad (2.35)$$

The minimum cost J*, associated with the optimal trajectory from state x_j at time t_j to the final time $t_N = t_f$ has been shown (1) to be

$$J^*(\underline{x}_j, t_j; t_N; \underline{u}^*) = \frac{1}{2} < \underline{x}_j, \underline{P}_j\ x_j > = \frac{1}{2}(\underline{x}_j' \underline{P}_j \underline{x}_j) \qquad (2.36)$$

As in the continuous case, the solution to the Riccati equation determines the feedback gain matrix, the optimal closed-loop system and the minimum cost. Consequently, investigation of its properties yields information about the optimal system's performance. The first result concerns the matrix $\underline{\Gamma}$ defined by equation (2.30).

Lemma 1: The symmetric matrix $\underline{\Gamma}$ is positive semidefinite, if $\underline{C}\ \underline{Q}\ \underline{C}'$ is positive semidefinite.

Proof: Let $\underline{C}\ \underline{Q}\ \underline{C}'$ be positive semidefinite. The cost J of equation (2.7a) can be expressed as:

$$J = \frac{1}{2}\,\underline{x}_N'\underline{S}\,\underline{x}_N + \frac{1}{2}\sum_{i=0}^{N-1}\int_{t_i}^{t_{i+1}} |\underline{y}'\underline{Q}\,\underline{y} + \underline{u}'\underline{R}\,\underline{u}|\, dt$$

which is non negative. Each integral of the sum is also non-negative since the integrand cannot be negative, i.e.,

$$L_i = \int_{t_i}^{t_{i+1}} [\underline{y}'\underline{Q}\,\underline{y} + \underline{u}'\underline{R}\,\underline{u}]\, dt \geq 0$$

If a piecewise constant control function is assumed then the integral above becomes:

$$L_i = \underline{x}_i'\hat{\underline{Q}}\,\underline{x}_i + 2\underline{x}_i'\hat{\underline{M}}\,\underline{u}_i + \underline{u}_i'\hat{\underline{R}}\,\underline{u}_i \geq 0 \qquad \forall\ u_i$$

Now let $\underline{u}_i$ be given by:

$$\underline{u}_i = - \hat{\underline{R}}^{-1} \underline{M}' \underline{x}_i$$

then

$$L_i = \underline{x}_i' \, |\hat{\underline{Q}} - 2 \underline{M} \hat{\underline{R}}^{-1} \underline{M}' + M \hat{R}^{-1} M'| \, \underline{x}_i$$

$$= \underline{x}_i' \, [\hat{\underline{Q}} - \underline{M} \hat{\underline{R}}^{-1} \underline{M}'] \, \underline{x}_i$$

$$= \underline{x}_i' \, \underline{\Gamma} \, \underline{x}_i \geq 0$$

Since this holds true for all $\underline{x}_i$, it follows that $\underline{\Gamma}$ is positive definite. Similarly if $\underline{CQC}'$ is positive definite, then so is $\underline{\Gamma}$. This result is essential for the existence of the solution to the Riccati equation for all $\underline{\Gamma}$.

The sequence of solutions $\{P_i\}$ consist of symmetric positive semidefinite matrices. Semidefiniteness can be justified directly from the quatratic form of equation (2.36) since the optimal cost is non negative.

If time invariance is assumed, then the sequence $\{\underline{P}_i\}$, i=N,N-1,...,0 is monotonically nondecreasing. Furthermore, if the discrete system equation (2.13) is controllable, then the sequence is bounded from above for any value of N (6).

INFINITE TIME REGULATOR PROBLEM

The results described in the previous paragraphs are based on the assumption that the interval of definition of the system is finite, that is, the optimization problem is considered to be a finite time regulator. These results are now extended to cover the infinite time regulator.

As the terminal time t_f approaches infinity, the discrete cost-functional of equation (2.15) becomes:

$$J[\underline{x}_0, t_0, \infty, y] =$$

$$= \frac{1}{2} \sum_{i=0}^{\infty} [\underline{x}_i' \hat{\underline{Q}}_i + 2\underline{x}_i' \underline{M}\, \underline{u}_i + \underline{u}_i' \hat{\underline{R}}\, \underline{u}_i] \qquad (2.37)$$

where the terminal cost matrix $\underline{F}$ has been set equal to zero. The optimal control law is given by:

$$\underline{u}_i^* = - \underline{G}\, \underline{x}_i^*, \; i = 0,1,2,... \qquad (2.38)$$

where

$$\underline{G} = \underline{R}^{-1}\underline{M}' + [\hat{\underline{R}} + \underline{D}'\underline{P}\, \underline{D}]^{-1}\, \underline{D}'\underline{P}\, \underline{\theta}$$

The matrix P is the steady-state solution

$$P = \lim_{i\to\infty} \underline{P}_i$$

of the matrix Riccati equation (2.32) with the boundary condition

$$\underline{P}_\infty = \underline{P}_N = \underline{0} \qquad (2.39)$$

Since the feedback gain matrix is constant, the closed loop system is time-invariant. Substitution of equation (2.36) into systems equation (2.13) yields

$$\underline{x}_{i+1} = [\underline{\Phi} - \underline{D}\ \underline{G}]\ \underline{x}_i$$

$$= [\underline{I} - \underline{D}(\hat{\underline{R}} + \underline{D}'\ \underline{P}\ \underline{D})^{-1}\underline{D}'\underline{P}]\ \underline{\theta}\ \underline{x}_i$$

$$\equiv \underline{\Phi}^*\ \underline{x}_i \qquad (2.40)$$

The steady state solution $\underline{P}$ exists and unique, provided the discrete system is controllable. In general, $\underline{P}$ is positive semidefinite, but if the system is also observable, then it is definite.

STABILITY OF THE TIME INVARIANT REGULATOR

Consider the closed loop system equation (2.38) repeated here for convenience.

$$\underline{\Phi}^*\ \underline{x}_i = [\underline{\Phi} - \underline{D}\ \underline{G}]\ \underline{x}_i \qquad (2.41)$$

The optimal closed loop system is stable if absolute values of all the eigenvalues of $\underline{\Phi}^*$ matrix are less than one, that is

$$[\lambda_j(\underline{\Phi}^*)| < 1 \qquad \forall_J;\ J = 1,2,\ldots,n$$

If the above inequality holds true for all J, then it is said that the closed loop system is stable. If, however, a single root is equal to one in absolute value, then the optimal closed loop system is marginally stable. If the absolute value of single complex eigenvalue is exactly one, then this produces sustained "oscillations" of angular frequency $2\pi/\psi T$ where T is the sampling interval and ψ is the angle of the root, i.e.,

$$\psi = \tan^{-1}\frac{I_m(\lambda_J)}{R_e(\lambda_J)} \qquad (2.42)$$

This is also form of marginal stability. In the case that more than one eigenvalues are found to be equal to one in absolute value then the stability of the system is determined by the minimal polynomial. For stability the minimal polynomial should not have multiple eigenvalues on the unit circle in the complex plane.

Instability may arise due to two factors:

1. The original open-loop system is unstable.
2. The unstable trajectories do not contribute in any way to the performance index - in a sense, the unstable states are not observed by the performance index.

If (1) and (2) is true, then there would be grounds for supposing that the closed loop system would be unstable.

To ensure asymptotic stability of the closed-loop system it is necessary to prevent the occurrence of (1) and (2). In this thesis unstable open loop systems are not considered. Thus the occurrence of (1) is avoided.

The performance index of equation (2.7a) was defined in terms of output and then using output equation (2.2) output y is expressed in terms of state $\underline{x}$ and the equation (2.7b) was arrived at which is repeated here for convenience.

$$J[\underline{x}_o, t_o; t_f; u(o)] = \frac{1}{2}\, \underline{x}'(t_f)\, \underline{Q}_N\, \underline{x}(t_f)$$

$$+ \frac{1}{2} \int_{t_o}^{t_f} [\underline{x}'(t)\underline{Q}_x\underline{x}(t) \quad \underline{u}'(t)\underline{R}\ \underline{u}(t)]\, dt \qquad (2.43)$$

where $\underline{Q}$ is chosen such that the pair $[\underline{A}, \underline{\Gamma}]$ is observable where $\underline{\Gamma}$ is the matrix such that

$$\underline{Q}_x = \underline{\Gamma}\underline{\Gamma}'$$

Then, all trajectories will show up in the $\underline{x}'\underline{Q}_x\underline{x}$ part of performance index of equation (2.43). It has been shown '(7,8,9) that the above choice of $\underline{Q}_x$ matrix ensures the stability of the closed loop system. Thus, any potentially unstable trajectories will be stabilized by the application of feed-back control.

CHAPTER 3

EFFECTS OF THE SAMPLING PERIOD ON THE CLOSED LOOP SYSTEM'S PERFORMANCE

The design of optimal sampled data linear tors described in the previous chapter depends parametrically on the size of sampling interval T. If the sampling interval T is specified beforehand, then optimal design associated with this value of T can be determined. In practice, however, it might be desirable to work with a range of acceptable values, or alternatively, with qualitative criteria for T and some desired performance characteristics for the overall closed-loop system. In that case, knowledge for the parametric dependence of the optimal solution on the sampling time T is required. But the nonlinear nature of the Riccati equation forces one to obtain solutions numerically for different values of T, on a digital computer. Then, based on the results obtained in this manner, the sampling period which gives the best performance can be chosen. But, this however is a time consuming, and costly operation. Therefore, it is of interest to obtain information on the effect of the choice of T on the optimal-closed-loop system performance without actually solving numerically the problem.

A rational way to study the performance of the system with respect to the choice of sampling time is to display the effect of T on certain characteristic quantities, such as the optimal cost, the eigenvalues of the closed loop system, etc. Since for each choice of T there is an optimal cost J* associated with it then, the optimal cost is a very good indicator of the systems performance. It is then natural to choose it as a performance criterion even though it depends on the initial conditions. Ideally, it would be desirable to obtain an analytical expression for the optimal cost as an explicit function of T and then investigate the change of J* with T, i.e., evaluate $\partial J^*/\partial T$ with respect to T. Since the optimal cost is given by

$$J^* = \frac{1}{2}\,\underline{x}_0{}'\underline{P}(T)\underline{x}_0 \qquad (3.1)$$

which is a highly nonlinear function of T. The above approach does not yield a useful expression, because, the resulting expression for the derivative requires more computational effort than the actual solution of the problem. Alternatively, if the optimal cost (3.1) is normalized with respect to initial conditions, it is then possible to obtain upper and lower bounds that are independent of the initial state.

It has been shown (2) that if $\lambda_{min}\{\underline{P}\}$ and $\lambda_{max}\{\underline{P}\}$ denotes the minimum and maximum eigenvalues of the symmetric positive semidefinite matrix $\underline{P}$, then the following inequality holds for all values of $\underline{x}$ other than zero.

$$\lambda_{min}\{\underline{P}\} \leq \frac{1/2\ \underline{x}_0{}'\underline{P}\ \underline{x}_0}{1/2\ \underline{x}_0{}'\underline{x}_0} \leq \lambda_{max}\{\underline{P}\} \qquad (3.2)$$

For simplicity, the normalized ratio is denoted by

$$\xi^*(T) \triangleq \frac{J^*(T)}{1/2\langle \underline{x}_0, \underline{x}_0 \rangle} \qquad (3.3)$$

so that equation (3.2) becomes

$$\lambda_{min}\{\underline{P}\} \leq \xi^*(T) \leq \lambda_{max}\{\underline{P}\}$$

Numerical solutions showed that the cost increases very slowly for small T, and for large T the curves become straight lines. This fact is illustrated in Figure 3.1 where curves A and B correspond to two sets of data. This implies that there are two basic modes of behavior of the system. The first mode, for small T, is essentially similar to that of the optimal continuous system and it exhibits fast oscillatory response. In the second mode the effect of the feedback is to make all the closed-loop eigenvalues real and negative; that is the optimal system is overdamped. In the transition region between the two modes each successive pair of eigenvalues becomes a pair of negative real ones. It can be concluded, therefore, that a satisfactory upper bound for design values

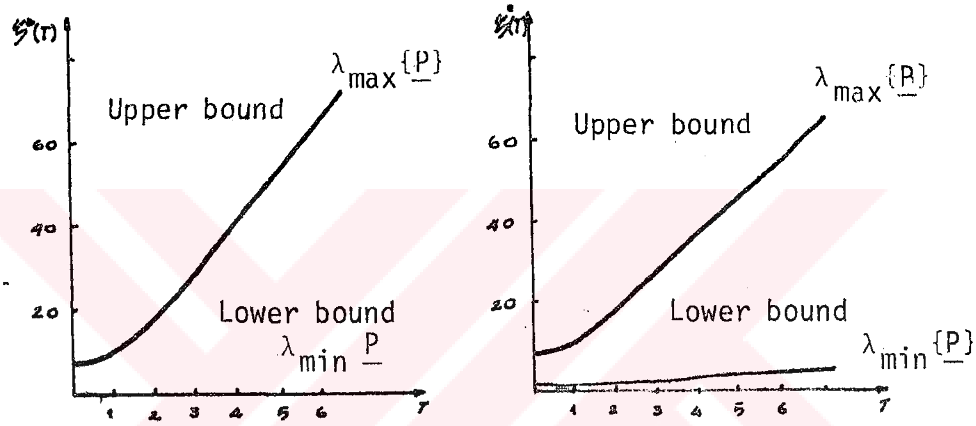


FIGURE 3.1. The normalized optimal cost as a function of the sampling period.

of the sampling interval, such that the cost increase remains small, is the smallest T for which all eigenvalues of the optimal closed-loop system become real.

On the other hand, T cannot be increased arbitrarily. Because Nyquist criteria determines the highest possible value T can take as,

$$T_{max} = \frac{1}{2\,\lambda_{max}} \tag{3.4}$$

where λ_{max} is the largest eigenvalue of the continuous system matrix. It is possible now to derive results for the eignevalues of P or equivalently eigenvalues of $\xi^*(T)$, for small and large values of T, since finding an analytic expression for them is almost impossible.

THE BEHAVIOR OF THE OPTIMAL TIME-INVARIANT SYSTEM FOR SMALL T

If T is sufficiently small, i.e., terms in T^2 or higher can be neglected then equation (2.32) takes the following form:

$$\underline{P}_i = \underline{T}(\underline{C}\ \underline{Q}\ \underline{C}') + (I + \underline{A}T)'\{\underline{P}_{i+1} - \underline{P}_{i+1}\ \underline{B}\ T(\underline{R}T + T\underline{B}'\underline{P}_{i+1}\underline{B}T)^{-1}T\underline{B}'\underline{P}_{i+1}\}(I + AT) \qquad (3.5)$$

This equation had been derived by Kleinman (8) to approximate the Riccati differential equation. Indeed, as T goes to zero, the solution to the above equation approaches the solution to the differential one and, at the limit, the difference equation becomes the differential one. This establishes the continuous feedback control law as the limiting case of the optimal sampled-data control for linear processes. This implies the following requirement

$$J^*(o) \leq J^*(T) \quad \forall\ T \in |0,\infty| \qquad (3.6)$$

where J*(o) is the optimal cost associated with the continuous time regulator problem. Since continuous state feedback is the optimal control, it is reasonable to decide that piecewise constant controls produce costs that are larger than the optimal (minimum) cost of the continuous-time system. It can also be established that J*(T) is monotonically nondecreasing with respect to T. But it has been shown by Kalman, Ho and Narendra (4) that such a statement, however is not always true. Because, if the periodicity inherent in sampling at constant rate is allowed to interact with the natural frequencies of the open loop plant, then the resulting discrete system may not be controllable. Loss in controllability may result in unbounded P and hence unbounded optimal cost for a countable set of values of T. When the plant has only real poles, then controllability is preserved for all values of T unless it exceeds the value determined by Nyguist criterion. In this case the optimal cost is monotonically nondecreasing with respect to T.

THE ASYMPTOTIC BEHAVIOR OF THE OPTIMAL SYSTEM FOR LARGE T

The asymptotic behavior for large T is determined as follows:

First complete results are presented for the first order system then the results are generalized and the form of the asymptote is related to the location and multiplicity of the plant's eigenvalues.

Proofs are presented for two cases.

1. Single input, single output n'th order system.
2. Multiple input, multiple output n'th order system.

THE FIRST ORDER SYSTEM WITH REAL EIGENVALUE

Consider the continuous plant defined by the state equations:

$$\mathring{x}(t) = \lambda x(t) + u(t) \qquad t \in |0,\infty) \tag{3.7a}$$

$$x(o) = x_o \tag{3.7b}$$

$$y(t) = x(t) \tag{3.8}$$

With the quadratic cost functional

$$J(x_o,0;\infty;u) =$$

$$\frac{1}{2} \int_o^\infty [qx^2(t) + u^2(t)]dt \tag{3.9}$$

where the coefficients of u and u^2 have been suppressed in equations (3.7a) and (3.9) without any loss of generality.

If the control is piecewise constant, then application of the discretization procedure of Chapter 2 gives the first order discrete regulator problem

$$x_{i+1} = \phi x_i + du_i \; ; \quad i = 0,1,2,... \tag{3.10}$$

$$J|x_0,0;\infty;u| = \frac{1}{2} \sum_{i=0}^{\infty} (\hat{q} x_i^2 + 2m x_i u_i + \hat{r} u_i^2) \tag{3.11}$$

where

$$\phi = e^{\lambda T} \tag{3.12a}$$

$$d = \frac{1}{\lambda} (e^{\lambda T} - 1) \tag{3.12b}$$

Obviously, the scalar regulator problem is the simplest one, and the Riccati-difference equation associated with it can be solved analytically.

$$\hat{q} = \frac{1}{2} \frac{q}{\lambda} (e^{2\lambda T} - 1) \tag{3.12c}$$

$$m = \frac{1}{2} \frac{q}{\lambda^2} (e^{\lambda T} - 1)^2 \tag{3.12d}$$

$$\hat{r} = (1 + \frac{q}{\lambda^2}) T + \frac{1}{2} \frac{q}{\lambda^3} (e^{2\lambda T} - 4e^{\lambda T} + 3) \tag{3.12e}$$

It is clear that ϕ, d, $\hat{q}$, m and $\hat{r}$ are all positive numbers. The optimal time varying feedback gain is given as:

$$g_i = \frac{m}{\hat{r}} + \frac{d \, P_{i+1}}{\hat{r} + d^2 P_{i+1}} \, \theta \tag{3.13}$$

tion

$$P_i = \gamma + \theta \, |P_{i+1} - P_{i+1} d(\hat{r} + dP_{i+1} d)^{-1} dP_{i+1}| \, \theta \qquad (3.14)$$

with γ and θ is defined as

$$\gamma = \hat{q} - \frac{m^2}{\hat{r}} \qquad (3.15a)$$

$$\theta = \phi - \frac{md}{\hat{r}} \qquad (3.15b)$$

Solution of the Riccati equation (see Appendix C) is then

$$P_i = \frac{r}{d^2} \frac{V_{i-1} - V_i}{V_i} \qquad (3.16)$$

where

$$V_i = \frac{1}{|\theta|^i} \, |(1-|\theta| \, e^{-\alpha}) e^{(N-i)\alpha} - (1-|\theta| \, e^{\alpha}) e^{-(N-i)\alpha}|$$

and

$$\alpha = \cosh^{-1} \frac{1+\theta^2+\frac{x_d{}^2}{\hat{r}}}{2|\theta|} \qquad (3.17)$$

If the limit is taken as the index i goes to $-\infty$ the steady state solution is obtained as

$$P = \lim_{i\to-\infty} P_i = \frac{\hat{r}}{d^2}(|\theta| e^{\alpha} - 1) \qquad (3.18)$$

This final result valid for all values of λ, contains in it all the information on the behavior of the steady state solution to the Riccati equation as a function of the sampling period T. Although the result is expressed simply, actual evaluation is impractical, as it involves the nontrivial determination of α. For this reason only the asymptotic behavior of P as t goes to infinity is derived for the three distinct cases.

Case i	$\lambda < 0$
Case ii	$\lambda = 0$
Case iii	$\lambda > 0$

Case (i) $\lambda < 0$ Stable Plant.

Taking limit as $T \to \infty$ in equation (3.12), it follows that for λ less than zero

$$\phi_\infty = 0; \quad d_\infty = -\frac{1}{\lambda}; \quad \hat{q}_\infty = -\frac{1}{2}\frac{q}{\lambda}$$

$$m_\infty = \frac{1}{2}\frac{q}{\lambda^2} \quad ; \quad \hat{r}_\infty = \left(1 + \frac{q}{\lambda^2}\right) T$$

$$\gamma_\infty = -\frac{1}{2}\frac{q}{\lambda}, \quad \theta_\infty = 0$$

Since $\theta_\infty = 0$, then it is observed from equation (3.17) as α goes to infinity, the product $|\theta| e^{\alpha}$ is indeterminate. This difficulty may be avoided by observing that

$$\cosh\alpha = \frac{e^{\alpha}+e^{-\alpha}}{2} = \frac{1+\theta+\frac{\gamma d^2}{\hat{r}}}{2|\theta|} \qquad (3.19)$$

and hence

$$|\theta|e^{\alpha} - 1 = \theta^2 + \frac{\gamma d^2}{\hat{r}} - |\theta|e^{-\alpha} \qquad (3.20)$$

Now for $\alpha \to \infty$ and $\theta_\infty = 0$, the right hand side becomes $\gamma d^2/\hat{r}$ and so

$$\lim_{T\to\infty} P(T) = \lim_{T\to\infty} \frac{\hat{r}}{d^2} (|\theta|e^{\alpha}-1) = \gamma_\infty = -\frac{1}{2}\frac{q}{\lambda} \qquad (3.21)$$

Case ii: $\lambda = 0$ Simple Integrator

It can be shown that

$$\phi = 1 \; ; \; d = T \; ; \; \hat{q} = qT \; ; \; m = \frac{1}{2} qT^2 \; ;$$

$$\hat{r} = T + \frac{1}{3} qT^3$$

and taking the limit as T goes to infinity

$$\theta_\infty = -\frac{1}{2} \; ; \quad \gamma_\infty = \frac{1}{4} qT \; ; \quad \lim_{T\to\infty} \frac{\gamma d^2}{2} = \frac{3}{4}$$

Therefore, form equation (3.17), $\cosh\alpha$ is equal to 2, and $e^{\alpha} = 2 + \sqrt{3}$, and as a consequence of the above results,

$$P(T) = \frac{1}{2\sqrt{3}} qT \quad \text{for } \lambda = 0 \text{ and large } T \qquad (3.22)$$

Case iii: $\lambda > 0$ Unstable Plant

Since the fundamental solution to this system is the growing exponential, only the dominant terms in equation (3.12) are retained in determining the asymptotic behavior for large T.

$$0_\infty = -1, \quad \gamma_\infty = (q + \lambda^2)T, \quad d_\infty = \frac{e^{\lambda T}}{\lambda},$$

$$r_\infty = \frac{q}{2\lambda^2} e^{2\lambda T}$$

so that

$$\cosh\alpha \simeq \lambda(1 + \frac{\lambda^2}{q})T, \quad e^\alpha \simeq 2\cosh\alpha \text{ for large } T$$

and

$$P(T) = (q + \lambda^2)T \tag{3.22}$$

The above results show that the optimal cost approaches a constant value for stable plants, and it is proportional to sampling period T for plants with a non-negative eigenvalue; the slope of the asymptote depends on q and λ. In Figures 3.2 and 3.3 below numerical results that are obtained through a general digital computer algorithm are illustrated to demonstrate the properties of the first order-system.

For T = 0, the optimum cost of the system may be computed from the equation (5) below:

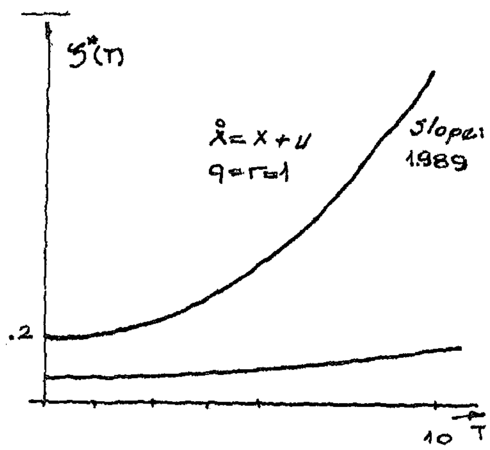


FIGURE 2.3.

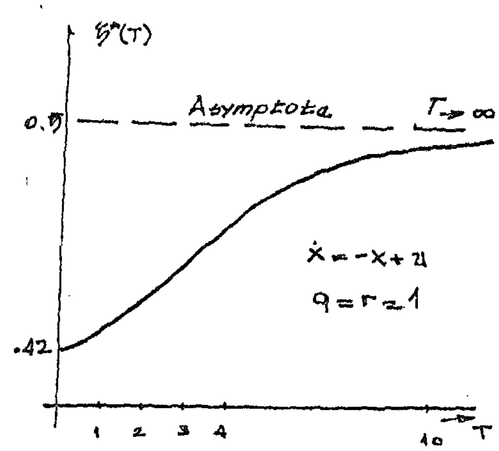


FIGURE 3.3

Optimal Normalized Cost Curves for the First Order Systems.

$$\underline{A}'\underline{P} + \underline{P}\,\underline{A} - \underline{P}\,\underline{B}\,\underline{R}^{-1}\underline{B}'\underline{P} + \underline{Q} = 0 \qquad (3.23)$$

This is known as the algebraic Riccati equation (ARE), and for the first order plant all of the above matrix quantities become scalars. For the problem under consideration the equation (3.23) reduces to

$$P^2 - 2\lambda P - 1 = 0 \qquad (3.24)$$

The solution to this equation is given by

$$P = \lambda + (\lambda^2 + 1)^{\frac{1}{2}} \qquad (3.25)$$

where the larger root is selected since P must be positive.

THE SINGLE-INPUT, SINGLE OUTPUT n-th ORDER SYSTEM

Consider the following single input, single output n'th order system whose eigenvalues are all distinct and negative real.

$$\mathring{\underline{x}}(t) = \underline{A}\ \underline{x}(t) + \underline{b}\ \underline{u}(t) \tag{3.26a}$$

$$\underline{x}(o) = \underline{x}(t_0) \tag{3.26b}$$

$$y = \underline{c}'\ \underline{x}(t) \tag{3.27}$$

with a cost functional

$$J = \frac{1}{2}\int_0^{\infty} |qy^2(t) + u^2(t)|dt \tag{3.28}$$

To simplify the results, the above system is assumed to be in control canonical form, and will be put into diagonal form through use of the similarity transformation.

$$\underline{x}(t) = \underline{T}\ \underline{z}(t) \rightarrow \mathring{x}(t) = \underline{T}\ \mathring{\underline{z}}(t)$$

$$\underline{T}\ \mathring{\underline{z}}(t) = \underline{A}\ \underline{T}\ \underline{z}(t) + \underline{b}\ \underline{u}(t)$$

$$\mathring{\underline{z}}(t) = \underline{T}^{-1}\underline{A}\ \underline{T}\ \underline{z}(t) + \underline{T}^{-1}\underline{b}\ \underline{u}(t)$$

$$\mathring{\underline{z}}(t) = \underline{\Lambda}\ \underline{z}(t) + \underline{h}\ \underline{u}(t) \tag{3.29a}$$

$$\underline{z}(o) = \underline{T}^{-1}\underline{x}(o) \qquad (3.29b)$$

$$y = \underline{c}'\underline{T}\ \underline{z}(t) \qquad (3.30)$$

where $\underline{\Lambda} = \underline{T}^{-1}\underline{A}\ \underline{T}$, $\underline{h} = \underline{T}^{-1}\underline{b}$ and $\underline{T}$ is the Vandermonde matrix associated with the companion matrix $\underline{A}$. With the change of variables the cost functional takes the following form

$$J = \frac{1}{2}\ {}_{o}\!\int^{\infty} [q\underline{z}'(t)\underline{T}'\underline{c}\ \underline{c}'\underline{T}\ \underline{z}(t) + y^2(t)]dt \qquad (3.31)$$

Since $\underline{c}' = [1,0,\ldots,0]$, the weighting matrix $\underline{T}'\underline{c}\ \underline{c}'\underline{T}$ reduces to an nxn matrix with elements all equal unity,

$$\underline{E} = \underline{T}'\underline{c}\ \underline{c}'\underline{T}, \quad e_{ij} = 1 \quad \text{for all } i,j$$

The discrete system matrices are obtained using the equations (2.9), (2.10) and (2.12) of Chapter 2 as

$$\underline{\Phi} = e^{\underline{\Lambda}T}, \quad \underline{D} = \underline{\psi}\ \underline{h} \text{ where } \underline{\psi} = \underline{\Lambda}^{-1}(e^{\underline{\Lambda}T} - \underline{I})$$

$$\underline{Q} = q\ \underline{S}, \quad s_{ij} = \frac{e^{(\lambda_i+\lambda_j)T} - 1}{\lambda_i + \lambda_j} \quad \forall i,j$$

$$M = q(\underline{s} - \underline{\psi}\ \underline{E})\ \underline{\Lambda}^{-1}h \qquad (3.32)$$

$$\hat{r} = T + \underline{q}h'\underline{\Lambda}^{-1}(\underline{S}-\underline{\psi}\ \underline{E}-\underline{E}\ \underline{\psi}+T\underline{E})\underline{\Lambda}^{-1}\underline{h} \qquad (3.33)$$

Taking limits as T goes to infinity, and since eigenvalues of the system are all negative real and distinct from each other it is observed that $e^{\underline{\Lambda} T} = \underline{0}$.

Thus:

$$\underline{\Phi}_\infty = \underline{0} \;; \qquad \underline{D}_\infty = - \underline{\Lambda}\ \underline{h}$$

$$\underline{\hat{Q}}_\infty = q\ \underline{S}_\infty \quad \text{where} \quad \underline{S}_\infty = - \frac{1}{\lambda_i + \lambda_j}$$

$$m_{ij}|_\infty = q \frac{1}{\lambda_i} \frac{1}{\lambda_i + \lambda_j} \quad \text{for } j = 1;\ i = 1,2,\ldots,n$$

where $m_{ij}|_\infty$ represents the i,j'th element of $\underline{M}$ matrix evaluated for T large.

$$\hat{r}_\infty = (1 + q\ \underline{h}'\underline{L}\ \underline{h})T + \text{Constant}$$

where

$$l_{ij} = \frac{1}{\lambda_i\ \lambda_j} \qquad \forall\ i,j$$

As a consequence of the presence of T in the expression for $\hat{r}_\infty$, the terms that modify $\underline{\Phi}$ and $\underline{\hat{Q}}$ in equations (2.24) and (2.30) of chapter 2 tend to zero as T goes to infinity, and therefore,

$$\underline{\theta}_\infty = \underline{0} \;; \qquad \underline{\Gamma}_\infty = \underline{\hat{Q}}$$

Finally, from the Riccati equation (2.34b)

$$\lim_{T \to \infty} \underline{P}(T) = \underline{\Gamma}_\infty = \underline{\hat{Q}}_\infty \qquad (3.34a)$$

$$\xi^*(T) \leq \lambda_{max}\{\underline{\hat{Q}}_\infty\} \qquad (3.34b)$$

Equation (3.34b) shows that all possible normalized optimal cost curves lie between zero and $\lambda_{max}\{\hat{Q}_\infty\}$ where λ_{max} denotes the maximum eigenvalue of $\underline{\hat{Q}}$ matrix when T is large.

The same type of argument maybe used to extend the last result to include the case of multiple negative real eignevalues. In this case the system is transformed to its Jordan canonical form and the matrix exponential has terms of the form $t^\nu e^{\lambda_i t}$ where ν is integer related to the multiplicity of the i'th root. But, since the system under consideration has only negative real eigenvalues, that is

$$\lambda_i < 0 \quad \text{for all } i;\ i = 1,2,\ldots,n-\nu$$

and, since the decaying exponential is the dominant factor for large t, it follows that $\underline{\Phi}_\infty = \underline{0}$. The final result reduces again to equation (3.16), where the elements of $\underline{\hat{Q}}_\infty$ are rational functions of the eigenvalues of the system matrix $\underline{\Lambda}$.

GENERAL BEHAVIOR OF THE ASYMPTOTE OF THE OPTIMAL COST

In order to generalize the result of the previous section, consider a single state system, i.e., a system where the index i takes only the zero value. Then the cost functional of equation (2.11) reduces to

$$J_0 = \frac{1}{2}\left(\underline{x}_0'\underline{\hat{Q}}\ \underline{x}_0 + 2\underline{x}_0'\underline{M}\ \underline{u}_0 + \underline{u}_0'\underline{R}\ \underline{u}_0\right) \tag{3.35}$$

where the terminal cost matrix has been set equal to zero. The weighting matrices are computed directly from their defining integrals (2.12b) through (2.12d) evaluated between t_0 and t_f. Since, $t_f - t_0$ is the interval of definition of the system and is equal to a single sampling period T, and since at the same time the terminal cost - in this case P_1 - is zero, then from equation (2.33)

$$\underline{u}_0 = -\underline{\hat{R}}^{-1}\underline{M}'\underline{x}_0 \tag{3.36}$$

and if this value of optimal control is substituted into equation (2.8) the optimal discrete-time system is obtained as

$$\begin{aligned}\underline{x}_{i+1} &= \underline{\Phi}\ \underline{x}_0 - \underline{D}\ \underline{\hat{R}}^{-1}\underline{M}_0'\underline{x}_0 \\ &= (\underline{\Phi} - \underline{D}\ \underline{\hat{R}}^{-1}\underline{M}')\underline{x}_0\end{aligned}$$

In equation (2.34) $\underline{\Phi} - \underline{D}\ \underline{\hat{R}}^{-1}\underline{M}'$ was defined as θ, thus

$$\underline{x}_{i+1} = \underline{\theta}\ \underline{x}_o \tag{3.37}$$

Equation (3.18) is substituted into the cost functional of equation (3.17) to yield

$$J_o = \frac{1}{2}(\underline{x}_o'\hat{\underline{Q}}\ \underline{x}_o - \underline{x}_o'\underline{M}\ \hat{\underline{R}}^{-1}\underline{M}'\underline{x}_o)$$

$$= \frac{1}{2}\underline{x}_o'\ (\hat{\underline{Q}} - \underline{M}\ \hat{\underline{R}}^{-1}\underline{M}')\underline{x}_o$$

Since $\hat{\underline{Q}} - \underline{M}\ \hat{\underline{R}}^{-1}\underline{M}'$ is defined as $\underline{\Gamma}$ by equation (2.30) then,

$$J_o = \frac{1}{2}\underline{x}_o'\ \underline{\Gamma}\ \underline{x}_o \tag{3.38}$$

Now if the terminal time t_f is made infinite, then

$$\lim_{t_f \to \infty} J_o = \frac{1}{2}\underline{x}_o'\ \underline{\Gamma}_\infty\ \underline{x}_o \tag{3.39}$$

Equation (3.39) describes the general form of the asymptotic behavior of the optimal cost for large T. In the special case in which the eigenvalues of the system are negative real, $\underline{\Gamma}_\infty$ is approximately equal to $\hat{\underline{Q}}_\infty$, since $\hat{\underline{R}}^{-1}$ is inversely proportional to T and $\underline{M}$ attains a constant value. This implies that optimal control is a step function whose magnitude is also inversely proportional to T. This can be observed from equation (3.18) where the feedback gain is,

$$\underline{G} = - \hat{\underline{R}}^{-1} \underline{M}' \tag{3.40}$$

Hence at the limit the gain of controller becomės zero, and the optimal cost, evaluated directly from equation (2.7b) ofr $\underline{u} = \underline{0}$, is given by

$$J_o = \frac{1}{2} \underline{x}_o' \int_0^\infty e^{\underline{A}t} \underline{C} \underline{Q} \underline{C}' e^{\underline{A}t} dt \underline{x}_o$$

$$= \frac{1}{2} \underline{x}_o' \hat{\underline{Q}} \underline{x}_o \tag{3.41}$$

It is obvious from the above derivations that if the plant matrix has any zero eigenvalue, i.e., $\underline{A}$ is singular, then the integrals involving matrix exponentials can not be expressed in terms of A^{-1}.

Furthermore, as can be deduced from the results of the derivation for the asymptotic behavior of the first order system, the matrix $\underline{\Phi}$ apparently does not go to zero for large T, and the optimal cost does not approach a constant value, but is approaches infinity as a function of T. For this reason it is necessary to establish a second approach for the asymptotic behavior of plants with zero eigenvalues.

ASYMPTOTIC BEHAVIOR OF THE MULTIPLE INPUT MULTIPLE OUTPUT SYSTEMS FOR LARGE T

a) A system with zero eigenvalues of multiplicity n:

Consider the n'th order system in its Jordan canonical form:

$$\overset{o}{\underline{z}}(t) = \underline{T}^{-1}\underline{A}\ \underline{T}\ \underline{z}(t) + \underline{T}^{-1}\underline{B}\ \underline{u}(t)$$
$$= \underline{\Lambda}\ \underline{x}(t) + \underline{H}\ \underline{u}(t) \qquad (3.42a)$$

$$\underline{z}(o) = \underline{T}^{-1}\underline{x}(o) \qquad (3.42b)$$

$$J = \frac{1}{2}\ {}_{o}\int^{\infty} [\underline{z}'(t)\underline{L}\ \underline{z}(t) + \underline{u}'(t)\underline{R}\ \underline{u}(t)]dt \qquad (3.43)$$

where

$$\underline{\Lambda} = \begin{bmatrix} 0 & 1 & . & . & . & 0 \\ 0 & 0 & 1 & . & . & . & 0 \\ & & & & & 1 \\ & \underline{0} & & & & \\ & & & & & 0 \end{bmatrix} \qquad \underline{L} = \underline{T}'\underline{C}\ \underline{Q}\ \underline{C}' \text{ with elements } l_{ij}$$

As stated in the previous section the optimal cost as T goes to infinity is given by

$$\lim_{T\to\infty} J = \frac{1}{2}\ \underline{x}_o{}'\ \underline{\Gamma}_{\infty}\ \underline{x}_o$$

It was established that $\underline{\Gamma}$ is at least positive semidefinite; this implies that

$$\hat{\underline{Q}} \geq \underline{M}\ \ \hat{\underline{R}}^{-1}\underline{M}' \qquad \forall T$$

Let the matrix $\underline{Q}_o$ be given by

$$\underline{Q}_o = {}_o\int^t e^{\underline{\Lambda}'t} \, \underline{L} \, e^{\underline{\Lambda}t} \, dt \qquad (3.44)$$

where the elements of the matrix exponential of $\underline{\Lambda}t$ are

$$\psi_{ij}(t) = \begin{cases} 0 & \text{when } i > j \\ \dfrac{t^{j-i}}{(j-i)!} & \text{when } i \leq j \end{cases} \qquad (3.45a)$$

and the elements of its transpose are

$$\psi'_{ij}(t) = \begin{cases} 0 & \text{when } i < j \\ \dfrac{t^{i-j}}{(i-j)!} & \text{when } i \geq j \end{cases}$$

$$= \psi_{ji}(t) \qquad (3.45b)$$

The inegrand of equation for Q_o (3.21) is a positive semi-definite matrix with its i,j'th element given by

$$\text{Integrand } Q(i,j) = \sum_{\nu=1}^{i} \sum_{\mu=1}^{j} \ell_{\nu\mu} \frac{t^{(i+j)-(\nu+\mu)}}{(i-\nu)!(j-\mu)!} \qquad (3.46)$$

This expression for the (i,j)th element of $\underline{Q}_o$ matrix is derived as follows:

(i,h)th element of the matrix that result from the multiplication of three matrices $\underline{A}$, $\underline{B}$ and $\underline{C}$ is given by

$$\text{Result}(i,h) = (i,h)\text{th element of } A\,B\,C$$

$$= \sum_{J=1}^{m} \sum_{k=1}^{r} a_{ij}\, b_{jk}\, c_{kh}$$

where A, B and C are (nxm), (mxr) and (rxp) dimensional matrices respectively.

If this formula is applied to the problem under consideration, the result will be of the form

$$\text{Result} = \sum_{\nu=1} \sum_{\mu=1} \psi'_{i\nu}\, \ell_{\nu\mu}\, \psi_{\mu j}$$

$$= \sum_{\nu=1} \sum_{\mu=1} \psi_{\nu i}\, \ell_{\nu\mu}\, \psi_{\mu j} \, . \tag{3.47}$$

But, on the other hand,

$$\psi_{\nu i} \triangleq \frac{t^{i-\nu}}{(i-\nu)!} \qquad \text{for } i \geq \nu$$

so ν can take at most the value equal to i whenever the ψ value is nonzero as indicated by equation (3.45a). In the same manner

$$\psi_{\mu j} \triangleq \frac{t^{i-\mu}}{(j-\mu)!} \qquad \text{for } j \geq \mu$$

that is; μ can take at most the value of j as indicated by equation (3.45b) whenever the ψ value is nonzero.

Then,

$$\int_0^T Q_{o_{i,j}}(t)dt = \int_0^T \sum_{\nu=1}^{i} \sum_{\mu=1}^{j} \ell_{\nu\mu} \frac{t^{i+j-\nu-\mu}}{(i-\nu)!(j-\mu)!}$$

Interchanging the operations of summation and integration, the above equation becomes

$$\int_0^T Q_{i,j}(t)dt = \sum_{\nu=1}^{i} \sum_{\mu=1}^{j} \ell_{\nu\mu} \int_0^T \frac{t^{i+j-\nu-\mu}}{(i-\nu)!(j-\mu)!} dt$$

If the required integration is performed, then

$$Q_{o_{i,j}}(T) = \sum_{\nu=1}^{i} \sum_{\mu=1}^{j} \ell_{\nu\mu} \frac{T^{(i+j-\nu-\mu+1)}}{(i-\nu)!(j-\mu)!(i+j-\nu-\mu+1)}$$

and $\qquad i,j = 1,2,\ldots, n \qquad$ (3.48)

In a manner exactly analogous to the procedure above the (i,j)th elements of $\underline{M}$ and $\hat{\underline{R}}$ matrices can be found. These terms are required for the evaluation of the asymptotic behavior of $\underline{\Gamma}$ matrix. In this case however, in each of the matrices $\underline{M}$ and $\hat{\underline{R}}$ the integral of $e^{\underline{\Lambda}t}$ is involved. Hence,

$$\int_0^T \psi_{ij}(t)dt = \begin{cases} 0 & \text{for } i>j \\ \dfrac{t^{j-i+1}}{(j-i)!(j-i+1)} & \text{for } i \leq j \end{cases} \qquad (3.49a)$$

and

$$\int_0^T \psi'_{ij}(t)\, dt = \begin{cases} 0 & \text{for } i < j \\ \dfrac{t^{i-j+1}}{(i-j)!(i-j+1)!} & \text{for } i \geq j \end{cases} \qquad (3.49b)$$

Recalling that $\underline{M}$ matrix is of the form $\underline{M} = \underline{M}_1 \cdot \underline{H}$ where matrix $\underline{M}_1$ matrix is given by

$$\underline{M}_1 = \int_0^T \underline{\psi}_{ji}(t)\ \underline{L}\ [\int_0^T \underline{\psi}_{ij}(s)ds]dt \qquad (3.50)$$

or

$$\underline{M}_1(i,j) = \int_0^T \sum_{\nu=1}^{i} \sum_{\mu=1}^{j} \ell_{\nu\mu} \frac{t^{(i+j-\nu-\mu+1)}}{(i-\nu)!(j-\mu)!(j-\nu-\mu+1)} dt$$

Interchanging the operations of summation and integration, and, performing the integration yields

$$\underline{M}_{1_{i,j}}(T) = \sum_{\nu=1}^{i} \sum_{\mu=1}^{j} \ell_{\nu\mu} \frac{T^{(i+j-\nu-\mu+2)}}{(i-\nu)!(j-\mu)!(j-\mu+1)(i+j-\nu-\mu+2)}$$

$$\begin{cases} i = 1,2,\ldots,n \\ j = 1,2,\ldots,n \end{cases} \qquad (3.51)$$

Following the same procedure for the (i,j)th element of $\underline{R}_1$ matrix, the following expression is obtained:

$$\underline{R}_{1_{i,j}}(T) = \sum_{\nu=1}^{i} \sum_{\mu=1}^{jj} \frac{T^{(i+j-\nu-\mu+3)}}{(i-\nu)!(j-\mu)!(i-\nu+1)(j-\mu+1)(i+j-\nu-\mu+3)}$$

and

$$i,j = 1,2,\ldots,n \qquad (3.52)$$

$\underline{R}_1$ matrix will be used in the evaluation of $\hat{\underline{R}}$ matrix defined by

$$\underline{\hat{R}} = \underline{H}' \int_0^T \{[\int_0^t e^{\underline{\Lambda}' s} ds] \; \underline{L} \; [\int_0^t e^{\underline{\Lambda} s} \, ds] \, dt \quad H + T\underline{R}$$

$$\underline{\hat{R}} = \underline{H}'\underline{R}_1\underline{H} + T\underline{R} \tag{3.58}$$

In each of the above matrices $\underline{\hat{Q}}_0$, $\underline{M}_1$ and $\underline{R}_1$, the highest power is 2n-1 and occurs when $i = j = n$ and $\nu = \mu = 1$. This largest power appears in the lowest diagonal elements of those matrices since all other terms are polynomials of lower degree in T, the trace Q_0 is also a polynomial of degree (2n-1) while the trace of $\underline{M}\ \underline{\hat{R}}^{-1}\underline{M}'$ is a polynomial of degree at most 2n-1 or less. Since $\underline{\Gamma}$ is positive semidefinite matrix, it follows that the trace of the difference

$$f(T) = \text{trace } \underline{Q}_0 - \text{trace}(\underline{M}\ \underline{\hat{R}}^{-1}\underline{M}')$$

is nonnegative for all T. The resulting polynomial in T must be at most of degree 2n-1, and, furthermore, the coefficient of the highest power present must be positive. If this were not the case, then, for very large T it would be possible for the polynomial to attain negative values, thus leading to a contradiction to the assumption of semidefiniteness. Since the highest eigenvalue must be given by a polynomial of the same degree with the trace, it follows that

$$\lambda_{max}\{Q_o\} = q(q_{2n-1}T^{2n-1}+\ldots+a_o) \geq \lambda_{max}\{\underline{\Gamma}\} \qquad (3.54)$$

Evaluation of the $\underline{Q}_o$ matrix for $\nu = \mu = 1$ and $i = j = n$ yields

$$Q_{o_{n,n}}(T) = \ell_{11} \frac{T^{2n-1}}{[(n-1)!]^2(2n-1)} \qquad (3.55)$$

as the (n,n)th element of this matrix.

It is further required to post multiply $\underline{M}_1$ by $\underline{H}$ in order to obtain the $\underline{M}$ matrix and premultiply $\underline{R}_1$ by $\underline{H}'$ and post multiply by $\underline{H}$ to obtain the $\hat{\underline{R}}$ matrix

$$\underline{M} = \underline{M}_1 \cdot \underline{H}$$

where the (i,j)th element of $\underline{M}$ matrix is,

$$\underline{M}_{i,j}(T) = \sum_{k=1}^{n} m_{1_{ik}} h_{kj} \quad \text{and} \begin{cases} i=1,2,\ldots,n \\ j=1,2,\ldots,r \end{cases} \qquad (3.56)$$

If the above multiplication is carried out each element of the $\underline{M}$ matrix can still be expressed as a polynomial in T. Recalling that the elements of $\underline{M}_1$ matrix were also a polynomial in T and H is a matrix of constants and the highest power of T appears when $\nu = \mu = 1$, then

$$\underline{M}_{1_{i,j}}(T) = \ell_{11} \frac{T^{i+j}}{(i-1)!(j-1)!(i+j)j} \qquad (3.57)$$

and elements of $\underline{M}$ matrix can be obtained from:

$$M_{i,j}(T) = \sum_{k=1}^{n} \ell_{11} \frac{T^{i+k}}{(i-1)!(k-1)!(i+k)_k} h_{kj}$$

where $\begin{cases} i = 1,2,\ldots,n \\ j = 1,2,\ldots,n \end{cases}$ (3.58)

Power of T is maximum when k = n, hence

$$\underline{M}_{i,j}(T) = \ell_{11}\, h_{nj} \frac{T^{i+n}}{(i-1)!(n-1)!n(i+n)} \qquad (3.59)$$

In this approximation it is assumed that the terms having smaller powers of T can be neglected, since only the asymptotic behavior of the $\underline{M}$ matrix is required when T is large. Coefficients of T terms become smaller as ν and μ are increased so approximation holds always true.

Similarly, the (i,j)th element of $\underline{R}_1$ matrix can be approximated, using the approximated form of equation (3.52)

$$\underline{R}_{1\,i,j}(T) = \ell_{11} \frac{T^{i+j+1}}{(i-1)!(j-1)!i.j(i+j+1)}$$

and $\begin{cases} i = 1, \ldots, n \\ j = 1, \ldots, n \end{cases}$ (3.60)

and $\hat{\underline{R}}$ is obtained as follows if the required multiplications are carried out

$$\hat{\underline{R}}_{i,j}(T) = \sum_{k=1}^{n} \sum_{\ell=1}^{n} h'_{ik}\, \eta_{k\ell}\, h_{\ell j}$$

$$= \sum_{k=1}^{n} \sum_{\ell=1}^{n} h_{ki} h_{\ell j} \ell_{11} \frac{T^{k+\ell+1}}{(k-1)!(\ell-1)!k.\ell(k+\ell+1)}$$

when $k = \ell = n$, then

$$\hat{\underline{R}}_{i,j}(T) = \ell_{11} h_{ni} h_{nj} \frac{T^{(2n+1)}}{[(n-1)!]^2 n^2 (2n+1)} \qquad (3.61)$$

It is obvious from the above expression that the inverse of the $\hat{\underline{R}}$ matrix is of the form

$$\hat{\underline{R}}^{-1} = \underline{\beta}^{-1} \; T^{-(2n+1)} \qquad (3.62)$$

where the elements of β matrix are given by

$$\beta_{ij} = \frac{\ell_{11} \; h_{ni} \; h_{nj}}{n^2 [(n-1)!]^2 (2n+1)} \qquad i,j = 1,2,\ldots,r \qquad (3.63)$$

All of the above approximations are made to obtain a simple approximate form for matrix $\underline{M}\hat{\underline{R}}^{-1}\underline{M}'$ when T is large.

Using equations (3.55), (3.59) and (3.61) the (i,j)th element of $\underline{M}\hat{\underline{R}}^{-1}\underline{M}'$ is obtained as

$$\underline{M}\hat{\underline{R}}^{-1}\underline{M}'_{i,j} = \sum_{k=1}^{r} \sum_{\ell=1}^{r} m_{ik} \; \hat{r}^{-1}_{k\ell} \; m'_{\ell j}$$

$$= \sum_{k=1}^{r} \sum_{\ell=1}^{r} m_{ik} m_{j\ell} \hat{r}^{-1}_{k\ell} \quad \text{where} \quad \begin{cases} i=1,2,\ldots,n \\ j=1,2,\ldots,n \end{cases}$$

$$= \sum_{k=1}^{r} \sum_{\ell=1}^{r} \ell_{11} h_{nk} \frac{T^{i+n}}{(i-1)!(n-1)!n(i+n)}$$

$$\ell_{11} h_{n\ell} \frac{T^{(j+n)}}{(j-1)!(n-1)!n(j+n)} \hat{r}_{k\ell}^{-1} \qquad (3.64)$$

Since only the lowest diagonal element of the $\underline{M}\ \hat{\underline{R}}^{-1} \underline{M}'$ matrix is required; that is i = j = n, then

$$\underline{M}\ \hat{\underline{R}}^{-}\ \underline{M}'_{n,n}(T) = \sum_{k=1}^{r} \sum_{=1}^{r} \ell_{11}^{2} h_{nk} h_{n} \frac{T^{4n}}{[(n-1)!]^{4} 4n^{4}} \beta_{k\ell}^{-1} T^{-(2n+1)}$$

$$\sum_{k=1}^{r} \sum_{=1}^{r} \ell_{11}^{2} h_{nk} h_{n} \frac{T^{(2n-1)}}{[(n-1)!]^{4} 4n^{4}} \beta_{k\ell}^{-1} \qquad (3.65)$$

If these last results are substitued into equation (2.30), then

$$\xi^*(T) \leq \mathrm{Trace}\{\Gamma_\infty\} =$$

$$= \left[\frac{\ell_{11}}{[(n-1)!]^{2}(2n-1)} - \right.$$

$$\left. - \sum_{k=1}^{r} \sum_{j=1}^{r} \frac{\ell_{11}^{2}\ h_{nk}\ h_{nj}}{[(n-1)!]^{4} 4n^{4}} \beta_{kj}^{-1} \right] T^{2n-1} \qquad (3.66)$$

which clearly shows that the asymptotic behavior of the normalized optimal cost $\xi^*(T)$ of equation (3.3) is determined by the (2n-1)st power of sampling period T for systems with n'th order eigenvalue at the origin.

b) A system with both negative and zero eigenvalues

Up to now investigation was made when the systems eigenvalues are all zero.

In general, a stable system contains both negative real and zero eigenvalues. In such cases the $\underline{\Lambda}$ matrix will contain two blocks one associated with zero eigenvalues of multiplicity ν and the other associated with the negative real eigenvalues if the states are arranged suitably. That is,

$$\underline{\Lambda} = \left[\begin{array}{c|c} \underline{\Lambda}_{11} & \underline{0} \\ \hline \underline{0} & \underline{\Lambda}_{22} \end{array}\right]$$

where $\underline{\Lambda}_{11}$ is a $(\nu x \nu)$ matrix and $\underline{\Lambda}_{22}$ is a $(n-\nu)$ by $(n-\nu)$ matrix

In the previous sections it was found that $\underline{\Gamma}_{\infty}$ matrix determines the asymptotic behavior of the optimal cost. On the other hand $\underline{\Gamma}_{\infty}$ matrix requires the evaluation of $\hat{Q}$, M, $\hat{R}$ matrices for large T. But each of these matrices involves exp $\underline{\Lambda}t$ in their defining integrals.

Elements of exponential $\underline{\Lambda}t$ can be found as given below

$$\exp \underline{\Lambda}t = \exp \left[\begin{array}{c|c} \underline{\Lambda}_{11} & \underline{0} \\ \hline \underline{0} & \underline{\Lambda}_{22} \end{array}\right] t$$

i,j'th element of exp $\underline{\Lambda}_{11}$

$$\exp \underline{\Lambda}_{11} = \psi^{11}_{i,j}(T) = \begin{cases} 0 & i > j \\ \dfrac{t^{j-i}}{(j-i)!} & \text{when} \quad i \leq j \end{cases}$$

and $i,j = 1,2,\ldots,\nu$.

$$\psi^{22}_{ij}(t) = \text{i,j'th element of } \exp \underline{\Lambda}_{22}t = \begin{cases} e^{\lambda_i t} & \text{when } i=j \\ 0 & \text{otherwise} \end{cases}$$

As obvious from the expression for the i,j'th element of exp $\underline{\Lambda}_{22}t$ matrix, for large T exp $\underline{\Lambda}_{22}t$ becomes a null matrix. Then it is only the $\underline{\Lambda}_{11}$ part that determines the asymptote. If $\hat{\underline{Q}}$, $\underline{M}$ and $\hat{\underline{R}}$ are partitioned as $\underline{\Lambda}$ were, then

$$\underline{\Gamma} = \left[\begin{array}{c|c} \underline{\Gamma}_{11} & \underline{\Gamma}_{12} \\ \hline \underline{\Gamma}_{21} & \underline{\Gamma}_{22} \end{array}\right] = \left[\begin{array}{c|c} \underline{Q}_{11} & \underline{Q}_{12} \\ \hline \underline{Q}_{21} & \underline{Q}_{22} \end{array}\right] - \left[\begin{array}{c|c} (\underline{M}\hat{\underline{R}}^{-1}\underline{M}')_{11} & (\underline{M}\hat{\underline{R}}^{-1}\underline{M}')_{12} \\ \hline (\underline{M}\hat{\underline{R}}^{-1}\underline{M}')_{21} & (\underline{M}\hat{\underline{R}}^{-1}\underline{M}')_{22} \end{array}\right]$$

It is then $\underline{\Gamma}_{11}$ that determines the asymptotic behavior of $\xi^*(T)$

$$\underline{\Gamma}_{11} = \underline{Q}_{11} - (\underline{M}\hat{\underline{R}}^{-1}\underline{M}')_{11}$$

Since this part corresponds to the zero eigenvalue of multiplicity ν the result will be the same as the result of the previous section with the appropriate change of dimensions as given below

$$\xi^*(T) \leq \{\frac{\ell_{11}}{[(\nu-1)!]^2(2\nu-1)} - \sum_{k=1}^{r}\sum_{j=1}^{r} \ell_{11}^2 h_{\nu k} h_{\nu i} \frac{\beta_{kj}^{-1}}{[(\nu-1)!]^4 4\nu^4}\} T^{2\nu-1} \tag{3.67}$$

and

$$\beta_{ij} = \frac{\ell_{11}\ h_{\nu i}\ h_{\nu j}}{\nu^2 |(\nu-1)!|^2 (2\nu+1)} \qquad i,j = 1,2,\ldots,r$$

PLANTS WITH COMPLEX EIGENVALUES

In the previous sections, all the n'th order plants that were considered could be decomposed into n-first order systems in cascade using only real coefficients. For this reason, their asymptotic behavior was essentially a generalization of that of the scalar one. The introduction of complex conjugate roots, however changes the behavior. Since loss of controllability is possible, it is the sampling periods for which the system is not controllable that are of interest rather than the asymptotic behavior.

The following theorem on the preservation of controllability in the presence of sampling is due to Kalman, Ho and Narendra (4).

THEOREM:

Let the continuous time-invariant system be completely controllable. Then the time invariant discrete time system is completely controllable if:

$$\mathrm{Im}\ \{\lambda_i(A) - \lambda_j(A)\} \neq n\,\frac{2n}{T} . \qquad (3.68)$$

whenever

$$\mathrm{Re} = \{\lambda_i(A) - \lambda_j(A)\} = 0$$

and $n = \pm 1, \pm 2, \ldots$

Therefore sampling period T should not be equal to T_k, where

$$T_k = n\,\frac{n}{\omega_k} , \quad n = \pm 1, \pm 2, \ldots \qquad (3.69)$$

and

$$\omega_k = I_m\ [\lambda_k] \qquad (3.70)$$

CHAPTER 4

OVERALL PACKAGE PROGRAM

In the subsequent paragraphs some salient features of the digital computer simulation for the application of the theory developed in the previous chapters are discussed.

The existing algorithm can handle up to tenth order systems and by a simple modification of the dimensioning statements it can be made to handle higher order systems, the only limitation being the size of the storage of the digital computer.

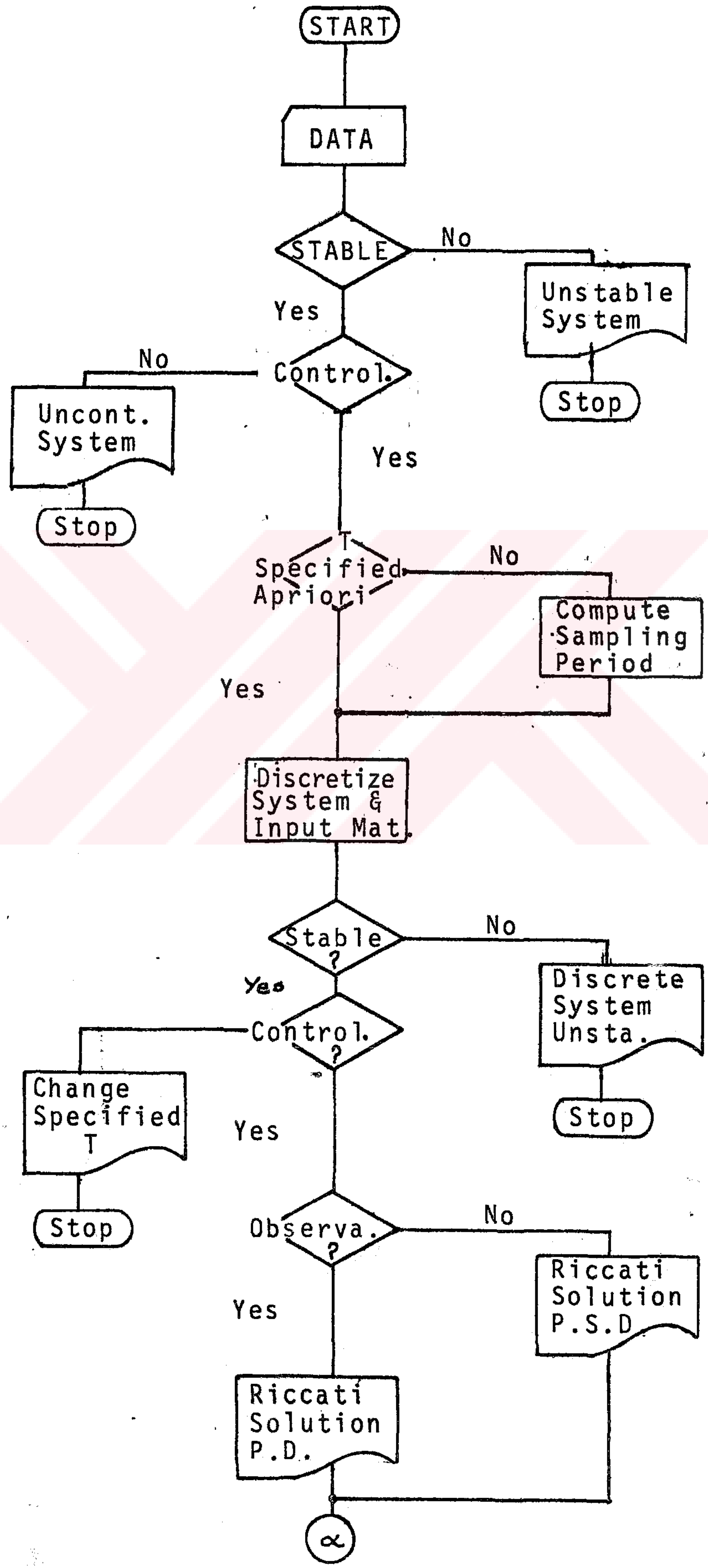
The computer programs are written in FORTRAN IV language and has been tested on the UNIVAC-1106.

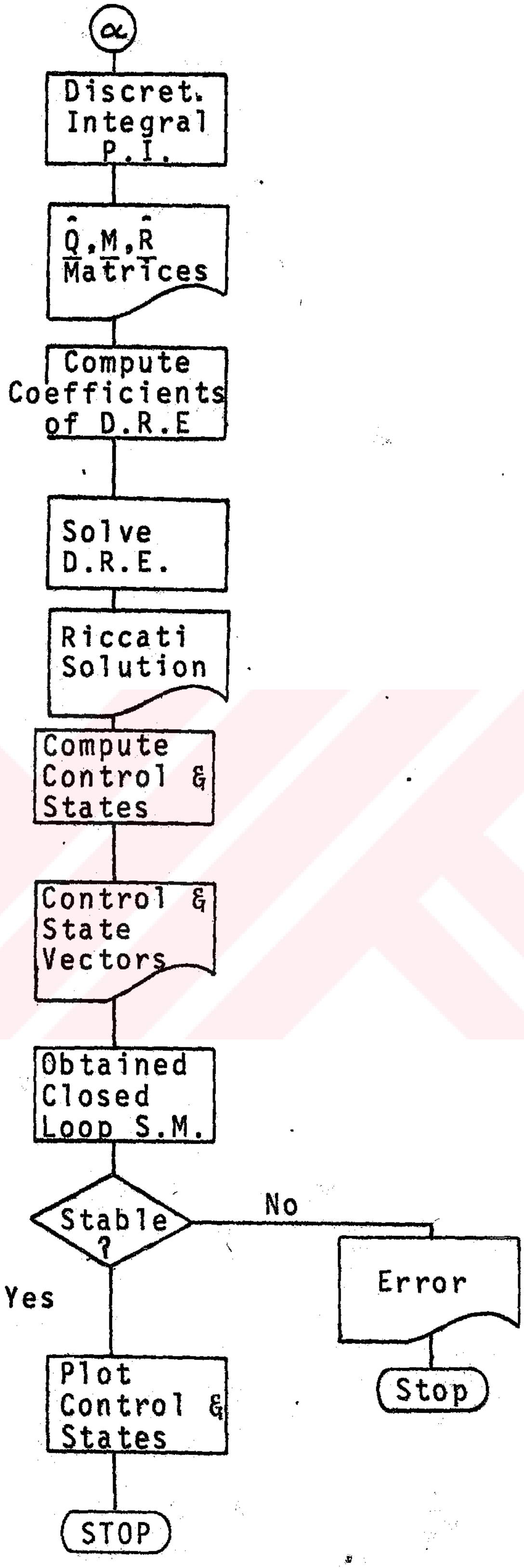
For a given set of input data, i.e., matrices $\underline{A}$, $\underline{B}$, $\underline{C}$, $\underline{Q}$, $\underline{R}$ and $\underline{F}$, the corresponding matrices $\underline{\Phi}$, $\underline{D}$, $\underline{\hat{Q}}$, $\underline{M}$ and $\underline{\hat{R}}$ of the discrete time optimization problem are computed.

There are various error tolerances within the program, which determine the accuracy of the final results. The only one need be specified as an input data is for the computation of the roots of an n'th order polynomial. Others are specified with the DATA declaration and if necessary they can be changed easily. Computation of the matrices $\underline{\Phi}$, $\underline{D}$, $\hat{\underline{Q}}$, $\underline{M}$ and $\hat{\underline{R}}$ requires the sampling period T be known, beforehand. If the sampling period is specified apriori as an input data, computation of sampling interval is skipped. Otherwise, results of Chapter 3 are used to compute the sampling period. For the given or computed value of T it is possible to obtain the above matrices. Availability of these matrices permits the iterative solution of the Riccati Equation. Although the Riccati equation is non-linear in nature a direct iterative algorithm can be used successfully, because the coefficient matrices are constant. The program can handle both infinite-time and finite time regulator problems. In the former case Riccati Equation is solved until a steady-state is reached. On the other hand, for the latter case the sequence of $\{P_i\}$; $i = N, N-1, \ldots, 1, 0$ are obtained as a solution of the Riccati Difference Equation (R.D.E.). These solutions are then used to evaluate the matrix of feedback gains and thus control vectors and the results are printed out at each iteration.

Finally, the closed loop system matrix and then states are computed within the time interval of interest and are plotted to illustrate the behavior of states as a function of time.

The simplified flow chart of the algorithm is given below:


START
DATA
STABLE
No
Unstable System
Stop
Yes
Control.
No
Uncont. System
Stop
Yes
T Specified Apriori
No
Compute Sampling Period
Yes
Discretize System & Input Mat.
Stable ?
No
Discrete System Unsta.
Stop
Yes
Control. ?
Change Specified T
Stop
Yes
Observa. ?
No
Riccati Solution P.S.D
Yes
Riccati Solution P.D.
α



α
Discret. Integral P.I.
Q̂,M,R̂ Matrices
Compute Coefficients of D.R.E
Solve D.R.E.
Riccati Solution
Compute Control & States
Control & State Vectors
Obtained Closed Loop S.M.
Stable ?
No
Yes
Error
Stop
Plot Control & States
STOP

END

The program comprises mainly two parts. The first one is the MAIN program itself and exercises overall control of the program. The DATA read in and then by means of various CALL statements the necessary computations shown in the flow-chart are performed.

The second part consists of the following SUBROUTINES:

1. SUBROUTINE STABIL
2. SUBROUTINE EIGEN
3. SUBROUTINE INTBSR
4. SUBROUTINE DISCO
5. SUBROUTINE EVOINT
6. SUBROUTINE SIMFOR
7. SUBROUTINE PICATI
8. SUBROUTINE INVERS
9. SUBROUTINE CONTR
10. SUBROUTINE OBSERV
11. SUBROUTINE RANKT
12. SUBROUTINE SAMPLE
13. SUBROUTINE DEFNIT
14. SUBROUTINE MULTIQ
15. SUBROUTINE SIFIR
16. SUBROUTINE TRANS
17. SUBROUTINE LOADM
18. SUBROUTINE SUBTRT
19. SUBROUTINE DEVRET
20. SUBROUTINE CIZERO
21. SUBROUTINE TOPLA

Among these SUBROUTINEs, MULTIQ, SIFIR, TRANS, LOADM, SUBTRT, DEVRET, CIZERO, TOPLA are self explanatory, but others need be explained.

1. SUBROUTINE STABIL

Stability of both continuous and discrete systems are tested by this subroutine.

Investigation of stability for continuous (discrete) systems is through the calculation of the eigenvalues of the matrix $\underline{A}(\underline{\Phi})$. A condition for stability requires that the absolute values of all the eigenvalues of the system matrix $\underline{A}(\underline{\phi})$ be less than zero (less than one) i.e.,

For continuous systems:

$$|\lambda_i\{A\}| < 0 \quad \text{for all } i: \quad i = 1,2,\ldots,n$$

For discrete systems:

$$|\lambda_i\{\Phi\}| < 1 \quad \text{for all } i: \quad i = 1,2,\ldots,n$$

Subroutine STABIL calls the subroutine EIGEN for the computation of the eigenvalues of the system matrix.

Parameters of SUBROUTINE STABIL are:

A1, N. and JDORC

A1 is the nxn matrix which determines the stability of the system.

N is the order of the system.

JDORC specifies whether the stability is investigated for discrete or continuous systems. If JDORC is set to one stability is checked for discrete systems, if two for continuous systems.

2. SUBROUTINE EIGEN

This subroutine first evaluates the coefficients of the characteristic polynomial of the given matrix A1. That is

$$f(\lambda) = \lambda^n - P_1\lambda^{n-1} - P_2\lambda^{n-2} - \ldots - P$$

P_i's are computed for all i, i = 1,2,...,n.

The algorithm used is due to Frame (11). Using these coefficients then, root finding routine ROOTCP is used to obtain the roots of the characteristic polynomial. Parametets of SUBROUTINE EIGEN are as follows.

A1,N,A,XR,KMAX,EPS,JJJ

where A1 is nxn matrix whose eigenvalues will be found.

N is the dimesion of the A1 matrix (Equivalently the order of the system)

A is a complex array which contains the coefficients of the characteristic polynomial.

XR is the array of the roots of the characteristic polynomial. It is also complex.

KMAX is the maximum number of iterations during the solution of the i'th root.

EPS specified "error tolerance" for the i'th root, i.e., the absolute value of the difference between the exact value of the root and the calculated one should be less than or equal to EPS using the maximum number of iterations KMAX.

Finally, JJJ is used to skip the WRITE statements if set equal to one.

With subroutine EIGEN it is possible to obtain real eigenvalues, complex conjugate eigenvales, and the zero eigenvalues of multiplicity ν(NU).

3. SUBROUTINE INTBSR (Integration by Simpson Rule)

Subroutine INTBSR is used to calculate the matrices $\hat{\underline{Q}}$, $\underline{M}$ and $\hat{\underline{R}}$ given by equations (2.12b), (2.12c) and (2.12d) respectively. Each of these matrices involves matrix exponentials in their integrands. Integrals of these matrices can be expressed in terms of the inverse of the system matrix $\underline{A}$. But in the case that $\underline{A}$ is singular, i.e., if one or more eigenvalues of A are zero, than inverse of $\underline{A}$ does not exist. Hence, the proposed method does not have practical importance. Instead, somewhat indirect but more satisfactory approach is chosen for the evaluation of the integrals at hand. Since integrands of the three integrals are functions of time, it is possible to find on a digital computer then, points satisfying these functions for a given point on the time axis. Then these points are used to extrapolate the function of the integrand with a simple function so that the area under this new curve be easily obtained. For the problem at hand the parabola is used as an extrapolating function. This is the so called Simpson Rule (15) and expressed mathematically as

$$I(f) \approx S = \frac{b-a}{6}\left[f(a) + 4f(\frac{a+b}{2}) + f(b)\right] \qquad (4.1)$$

where I(f) is the integral of the function f and (b-a) is the interval of integration.

The rule however for the estimation of the integral $I = \int_a^b f(t)dt$ will usually not produce sufficiently accurate estimates especially when the interval is reasonably large. Therefore, the given interval is subdivided into N smaller intervals and then the above rule is applied to these intervals. That is

$$I(f) = \sum_{i=1}^{N} \int_{t_{i+1}}^{t_i} f(t)dt = I(g_k) = \Sigma \int_{t_{i+1}}^{t_i} P_{i\ k}(t)dt \tag{4.2}$$

Interval of integration is the sampling period for equations (2.12b) through (2.12d). Subdividing this period into N intervals requires the selection of appropriate step size. It is suggested (16) that it will be satisfactory to choose a step size as one tenth of the minimum time constant available in the system. Because of the forms of the integrands powers of exponential terms could be at most $2\lambda_{max}$. Thus, as a step size one twenty-fifth of a minimum time constant is chosen. For this case the Simpson formula changes as

$$\underline{S}_N = \frac{h}{6} \left| \underline{f}_o + \underline{f}_N + \sum_{i=1}^{N-1} \underline{f}_i + 4 \sum_{i=1}^{N} \underline{f}_{i-\frac{1}{2}} \right| \tag{4.3}$$

If the equation (4.3) is applied to perform the integrations given by equations (2.12b), (2.12c) and

(2.12d), the functions of f_i's become matrix functions of time. For each time point the entire matrix is evaluated by means of subroutine DISCO and finally the above formula has been applied to get the required matrices. Subroutine INTBSR calls the following subroutines during the course of computation.

They are respectively,

a. SUBROUTINE EVOINT (Evaluates integrand)

b. SUBROUTINE TOPLA

c. SUBROUTINE LOADM

d. SUBROUTINE SIFIR

e. SUBROUTINE MULTIQ

f. SUBROUTINE SIMFOR

Parameters of SUBROUTINE INTBSR are

T - sampling period

$\underline{R}$ - R matrix (Dimension rxr)

$\underline{IR}$ - r dimension of $\underline{R}$ matrix

$\underline{Q}$ - Q matrix

$\underline{A1}$ - system matrix

KONTRL: When set to one $\hat{Q}$ matrix, when 2 M matrix and finally when 3 $\hat{R}$ matrix is found.

SUBROUTINE DISCO:

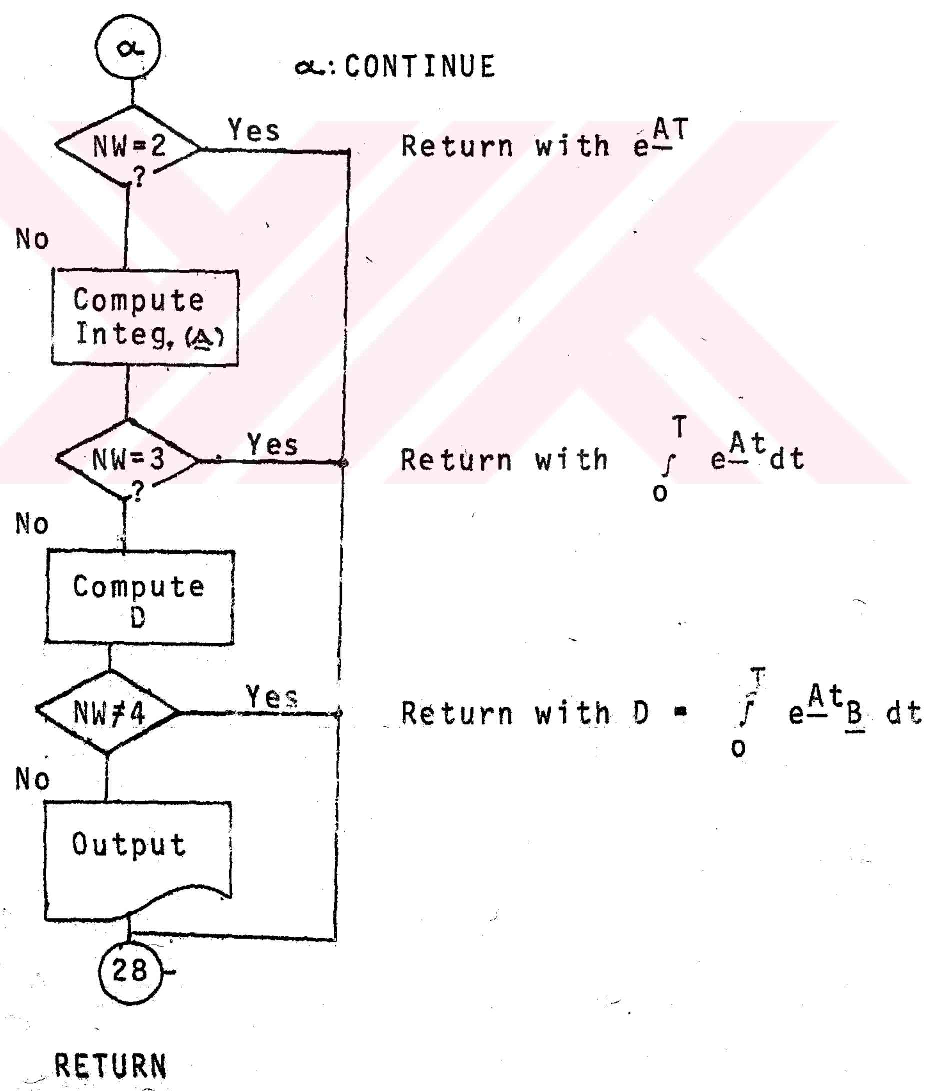
This subroutine is used to evaluate the matrices

$$\underline{\Phi} = e^{\underline{A}T} \quad ; \quad D = \int_0^T e^{\underline{A}t} \cdot B \, dt$$

There are many algorithms for the evaluation of these matrices but in subroutine DISCO the method of Liou (10) in which the exp $\underline{A}T$ is approximated by the truncated

Taylor series is used. Number of terms in the series is increased starting from 10 until the elements of Ø (thus D) matrix becomes sufficiently accurate.

Except for the discretization of continuous system the subroutine DISCO is referenced by subroutine INTBSR via subroutine EVOINT. Therefore, following control is essential to avoid unnecessary computations



Parameters of subroutine DISCO.

A1 (nxn) system matrix

B (nxm) input matrix

N : n

M : m

T : sampling period

NW: function of NW shown in the flowchart above.

SUBROUTINE EVOINT

Subroutine EVOINT is referenced only by SUBROUTINE INTBSE and is used to evaluate the integrand for a given point. Required information is supplied to subroutine EVOINT by subroutine DISCO.

SUBROUTINE SIMFOR

Applies the Simpson formula to the points found by subroutine EVOINT as adapted to matrices. Referenced only by subroutine INTBSR.

SUBROUTINE INVERS:

Inverts the given square matrix and finds also the determinant using Gauss-Jordan complete elimination method with the maximum pivot strategy.

SUBROUTINE RICATI

Solution to the Riccati difference equation is found by subroutine RICATI.

Calculation of the coefficients of Riccati equation also takes place in this subroutine. Subroutine Riccati checks first whether the final cost matrix is zero. If so, the Riccati equation is solved until a steady state solution is obtained. Iteration stops when the least squre error between the results of two consecutive iteration is less than the specified error tolerance. Otherwise, continue until this limit is reached.

If final cost matrix is not zero subroutine Riccati solves iteratively backward in time the nonlinear Riccati difference equation, and sequence of solution matrices is obtained associated with the finite time regulator problem.

Then, matrix of feedback gains, control vectors, and state vectors are calculated and printed out at each iteration. In the case of finite time regulator problem, feedback gain matrices are time varying.

$\hat{\underline{R}}$, $\underline{M}$, $\hat{\underline{Q}}$, $\underline{\phi}$ and $\underline{D}$ matrices along with the final time IFT and final cost matrix $\underline{S}$ should be supplied to subroutine Riccati.

SUBROUTINE CONTR

Is used to check whether the system under consideration is controllable. Condition for complete controllability requires that A pair of constant matrices $|\underline{A},\underline{B}|$ with $\underline{F}$ (nxn) and $\underline{G}$(nxr) is of rank n, that is

$$\text{Rank } [\underline{G} \vdots \underline{A}\ \underline{B}\ \ldots \vdots \underline{A}^{n-1}\underline{B}\,] = n$$

The above augmented matrix is of dimension (nxnr).

SUBROUTINE CONTR

References 3 more subroutines, subroutine INVERS$ subroutine RANKT and subroutine DEVRET.

SUBROUTINE OBSERV

Is used to test if the system is observable through searching the rank of the matrix pair

$$[\underline{C}\ \underline{A}]$$

where $\underline{M}$ (mxn) and $\underline{F}$ (nxn) dimensional, that is

$$\text{Rank } [\underline{C}\ \ \underline{A}\ \underline{C}'\ \ \ldots\ \underline{A}^{(n-1)}\underline{C}'] = n$$

The augmented matrix is obviously (nxnm) dimensional.

SUBROUTINE RANKT

Tests the rank of an (nxm) matrix applying Gauss elimination method. After elimination, number of non-zero rows (columns) gives the rank.

SUBROUTINE DEFNIT

With the application of Sylvester's expansion theorem, subroutine DEFNIT checks whether the (nxn) square matrix positive definite, positive semidefinite or neither.

SUBROUTINE MULTIQ - Multiplies two matrices.

SUBROUTINE SIFIR - Sets initially all the elements of the given matrix to zero.

SUBROUTINE TRANS - Transposes the given matrix.

SUBROUTINE LOADM - Transfers the constants of one matrix into another.

SUBROUTINE SUBTRT - Subtracts one matrix from the other.

SUBROUTINE DEVRET - Constants of one matrix is transferred to the other with subscript, i.e., $\underline{P}_i$.

SUBROUTINE CIZERO - Checks if the given matrix is a null matrix.

SUBROUTINE TOPLA - Adds two matrices.

CHAPTER 5

CONCLUSIONS

This study covers the design of the optimum sampled-data regulators for linear, time invariant, completely observable and controllable plants through the use of the discrete minimum principle. The term sampled-data describes the sampling operation between the plant and the controller, where the states have been sampled with a fixed rate and then transformed into a piece-wise constant form by means of a data hold process. The use of a pieve-wise constant inputs and the minimization of quadratic cost functional which has been considered as a performance measure, result in a sub-optimal control scheme.

Since, the performance of the system is affected by sampling, more emphasis has been given to the effects of sampling on the system's behavior and as consequence different costs have been observed for different sampling intervals. In addition analytical expressions have been derived for the behavior of normalized cost as a function of sampling-period T.

This study covers also the development of a computer algorithm for the selection of sampling-interval T which gives the best system performance and for the complete design of the sampled data regulator using this sampling-interval.

The design of the regulator starts with the evaluation of the eigenvalues of the continuous system matrix A. If the eigenvalues are all real, then it is multiplicity of non-negative roots that determines the behavior of the system's performance as a function of sampling period T. In the case of the existence of complex conjugate eigenvalues the location of the imaginary part of the complex roots determines the sequence of critical values of T for which the discrete system may not be controllable. These values of sampling-i interval are given as

$$T_k = n\frac{\Pi}{w_k} = n\,\frac{\Pi}{I_m(\lambda_k)} \quad \text{and } n = \pm 1, \pm 2, \ldots$$

If the statement of problem specifies the sampling period or a narrow range for it, then the optimal system associated with the given sampling rate can be designed if and only if the given sampling-interval is less than the smallest period for which the system may not be controllable, i.e.,

$$T \in (0, T_{kmin})$$

If the specified value of T is in between the two successive T_k's which are widely spaced from each other, then an optimal design is feasible though seldom practical. If T is very near any of the values of T_k, then further investigation is made to establish controllability. If for T_k the system is not controllable, then an optimal design is not possible.

If the sampling period is not specified or if its range is large, then maximum value of T is determined, i.e.,

$$T_{max} = \frac{\Pi}{w_{max}} = \frac{\Pi}{I_m|\lambda_{max}|}$$

The best range for T is then

$$T \varepsilon (0, \alpha T_{max}) \; ; \quad 0 < \alpha < 1$$

To establish the increase of the optimal cost with increasing T, three values of α has been chosen (α = .5, .75, .9) in the computer algorithm and the normalized cost associated with each T, has been determined. Then on the basis of this information and on the basis of trade-off between increasing the sampling period and decreasing the cost, a good choice of T has been made. It should also be noted that the stability of the discrete system has been taken into account for the choice of sampling period T.

If the $\underline{A}$ matrix has only real roots, then the optimal cost curve behaves asymptotically as $T^{2\nu-1}$, where ν is the highest multiplicity of zero roots. Then, based on the information got from the normalized optimal cost a good choice of T has been made.

Once, the value of sampling period is selected, the computer algorithm is then determines numerically the complete optimal design, so that the whole design procedure has been computerized.. The only data should be given are, the $\underline{A}$, $\underline{B}$, $\underline{C}$, $\underline{Q}$, $\underline{F}$ and $\underline{R}$ matrices to get the optimal solution, which is optimal feed-back gain matrix.

As the foregoing discussion implies the sampled-data regulator with a fixed rate sampling constitutes a sub-optimal control law. A formulation with sampling times which are constrained or dependent on states, constitutes a better approximation to the continuous time control problem, and is open to further study. Furthermore, in this study, the effects of changing sampling period has been justified by considering the behavior of normalized cost. Since, the locations of the eigenvalues of discrete system changes with sampling period, it is also possible to use the locations of the eigenvalues as a performance criteria to study the effects of sampling period.

In addtion, utilization of a digital computer in a control system requires the sampled signal be quantized, because the limited bit capacity of such an equipment does not allow whole signal levels to be represented. It is plausible then to study the effects of quantization on the system's performance in addition to the effects of sampling.

Also, throughout the development of computer programs it has been assumed that the states of the controlled plant are available for measurement. Frquently in practical situations, this will not be the case, and some artifice to get around this problem is required. In this case, the addition of a state estimator will improve the flexibility of the already developed programs, but the effects of the errors that are introduced during state estimation on the performance of sampled-data regulator is another area of research.

REFERENCES

1. Dorato and A.H. Levis, "Optimal Linear Regulators, The Discrete Time Case", IEEE Transactions on Automatic Control, 1971.

2. R. Bellman, Introduction to Matrix Analysis, McGraw Hill.

3. Kleinman, D.L., "On the Linear Regulator Problem and the matrix Riccati Equation", Report Electronic Systems Laboratory, 1967.

4. Kalman, R.E., Ho, Y.C.$ and Narendra$ K.S., "Controllability of Linear Dynamical Systems", Contributions to Differential Equations", Interscience, 1963.

5. Shultz and Melsa, State Functions and Linear Control Systems, McGraw Hill.

6. Alexander H. Levis$ Michael Athons, Robert A. Schlueter, "On the Behavior of Optimal Linear Sampled-Data Regulator", Session Paper, 1968.

7. Kalman, R.E. "When is a Linear Control System Optimal", J. Basic Eng., Ser. D, Vol. 86, March 1964.

8. Harold J. Payne and Leonard M. Silverman, "On the Discrete Time Algebraic Riccati Equation", IEEE Transactions on Automatic Control, June, 1973.

9. M. Athons and P.F. Falb, Optimal Control, McGraw-Hill.

10. M.L. Liou, "A Novel Method of Evaluating Transient Responses", Proceedings of IEEE, January 1969.

11. Frame, J.S., "Matrix Functions and Applications", IEEE Spectrum, June 1964.

12. Robert A. Schlueter and Alexander H. Levis, "The Optimal Linear Regulator with State-Dependent Sampling", IEEE Transactions on Automatic Control, 1973.

13. David L. Kleinman, Thomas Fortmann, and Michael Athons, "On the Design of Linear Systems with Piece-wise Constant Feed-back Gains", IEEE Transactions on Automatic Control.

14. Alexander H. Levis and Machael Athons, "On the Optimal Sampled Data Control of Strings of Vehicles", Transportation Science, November, 1968.

15. Conte-de Boor, Elementary Numerical Analysis, McGraw Hill.

16. A.H. Levis, "Some Computational Aspects of the Matrix Exponential", IEEE Transactions on Automatic Control.

17. J.B. Plant, "Some Iterative Solutions in Optimal Control", Research Monograph, The MIT Press, Cambridge, Massachusetts.

18. Albert B. Bishop, "Introduction to Discrete Linear Controls, Theory and Application", Academic Press.

19. F.R. GantMacher, "The Theory of Matrices", Volume I, II, Chelsea Publishing Company.

20. Julious T. Tou, "Digital and Sampled Data Control Systems", McGraw-Hill.

21. Julious T. Tou, "Optimum Design of Digital Systems", McGraw Hill.

22. James A. Cadzow, Hinrich R. Martens "Discrete Time and Computer Control Systems", Prentice-Hall, Electrical Engineering Series.

23. Brian D.O. Anderson, John B. Moore, "Linear Optimal Control", Prentice Hall, Electrical Engineering Series.

24. A.C. Aitken, "Determinant and Matrices", Oliver and Boyd.

25. L.A. Zadeh, E. Polak, "System Theory", McGraw-Hill.

26. Rosenbrock, "State Space andMultivariable Theory.

27. Numerical Solution of Systems of Nonlinear Algebraic Equations, Edited by George D. Byrne and Charles A. Hall, Academic Press.

APPENDIX A

Consider the homogenous system equation

$$\overset{o}{\underline{x}}(t) = \underline{A}\ \underline{x}(t) \qquad (A.1)$$

where A is the (nxn) system matrix, and $\underline{x}$ is the (nx1) state vector. If $\underline{\Phi}(t,t_i)$ is the transistion matrix of equation (B-1), then for arbitrary initial state $\underline{x}_i = \underline{x}(t_i)$ at some time t, we may write:

$$\underline{x}_i(t) = e^{\underline{A}(t-t_i)}\ \underline{x}_i = \underline{\Phi}(t,t_i)\underline{x}_i \qquad (A.2)$$

This is the solution of the equation of free motion. If we consider also that the control exists, i.e.,

$$\overset{o}{\underline{x}}(t) = \underline{A}\ \underline{x}(t) + \underline{B}\ \underline{u}(t) \qquad (A.3)$$

where $\underline{B}$ is (nxr) gain matrix, and $\underline{u}$ is (rx1) control vector, the solution to the above system is given as

$$\underline{x}_i(t) = e^{\underline{A}(t-t_i)}\underline{x}_i + \int_{t_i}^{t} e^{\underline{A}(t-\tau)}\underline{B}\ \underline{u}(t)d\tau \qquad (A.4)$$

If a piecewise constant control input is assumed, that is:

$$\underline{u}_i(\tau) = u_i \quad \text{for } t_i \leq \tau < t \qquad (A.5)$$

Then with use of the STM

$$\underline{x}_i(t) = \underline{\Phi}(t,t_i)\underline{x}_i + \int_{t_i}^{t} \underline{\Phi}(t,t_i)\underline{B}u_i dt$$

finally,

$$\underline{x}_i(t) = \underline{\Phi}(t,t_i)\underline{x}_i + \underline{D}(t,t_i)\underline{u}_i \qquad (A.6)$$

where

$$D(t,t_i) = \int_{t_i}^{t} \underline{\Phi}(t,t_i)B\ dt$$

Transpose of $\underline{x}_i(t)$ is given as

$$\underline{x}_i'(t) = \underline{x}_i\ \underline{\Phi}'(t,t_i) + \underline{u}_i'\underline{D}'(t,t_i) \qquad (A.7)$$

If equations (A.6) and (A.7) are substituted into the expression for performance index J (Equation (2.7b)) and since it is assumed that the control interval is divided N equal parts

$$J = \sum_{i=0}^{N-1} \int_{t_i}^{t_{i+1}} \frac{1}{2}\{[\underline{x}_i'\underline{\Phi}'(t,t_i)+\underline{u}_i'\underline{D}'(t,t_i)]\underline{C}\underline{Q}\underline{C}'[\underline{\Phi}(t,t_i)\underline{x}_i$$

$$+ \underline{u}_i\underline{D}(t,t_i)]+\underline{u}_i'\underline{R}\ \underline{u}_i\}\ dt = \sum_{i=0}^{N-1} \frac{1}{2} \int_{t_i}^{t_{i+1}} \{\ \underline{x}_i'\underline{\Phi}'(t,t_i)\underline{CQC}'\underline{\Phi}(t,t_i)\underline{x}_i$$

$$+ \underline{u}_i'\underline{D}'(t,t_i)\underline{C}\ \underline{Q}\ \underline{C}'\underline{\Phi}\ (t,t_i)\underline{x}_i + \underline{x}_i'\underline{\Phi}'(t,t_i)\underline{C}\ \underline{QC}'\underline{D}(t,t_i)\underline{u}_i$$

$$+ \underline{u}_i'\underline{D}'(t,t_i)\underline{C}\ \underline{Q}\ \underline{C}'\underline{D}(t,t_i)\underline{u}_i + \underline{u}_i'\underline{R}\ \underline{u}_i\}\ \ dt$$

Each of the above quantities is scalar, additionally Q is a symmetric matrix, then

$$\underline{u}_i'\underline{D}'(t,t_i)\underline{C}\ \underline{Q}\ \underline{C}'(t,t_i)\underline{x}_i = \underline{x}_i'\underline{\Phi}(t,t_i)\underline{C}\ \underline{QC}'D(t,t_i)\underline{u}_i$$

finally

$$J = \sum_{i=0}^{N-1} \underline{x}_i' \int_{t_i}^{t_i+1} \underline{\Phi}'(t,t_i)\underline{C}\ \underline{Q}\ \underline{C}'\_(t,t_i)dt\ \underline{x}_i$$

$$+ 2\underline{x}_i' \int_{t_i}^{t_i+1} \underline{\Phi}'(t,t_i)\underline{C}\ \underline{Q}\ \underline{C}'\underline{D}(t,t_i)dt\ \underline{u}_i$$

$$+ \underline{u}_i' \int_{t_i}^{t_i+1} [\underline{R} + \underline{D}'(t,t_i)\underline{C}\ \underline{Q}\ \underline{C}'\underline{D}(t,t_i)]\ dt\ \underline{u}_i \quad (A.8)$$

For simplicity in the appearance of the above equation the following matrices are defined.

$$\hat{\underline{Q}} = \int_{t_i}^{t_i+1} \underline{\Phi}(t,t_i)\underline{C}\ \underline{Q}\ \underline{C}'\underline{\Phi}(t,t_i)dt \quad (A.9a)$$

$$\underline{M} = \int_{t_i}^{t_i+1} \Phi'(t,t_i)CQC'D(t,t_i)dt \quad (A.9b)$$

$$\hat{\underline{R}} = \int_{t_i}^{t_i+1} [\underline{R} + \underline{D}'(t,t_i)\underline{C}\ \underline{Q}\ \underline{C}'\underline{D}(t,t_i)]dt \quad (A.9c)$$

Consequently, equation (B-8) reduces to:

$$J = \frac{1}{2} \sum_{i=0}^{N-1} \underline{x}_i'\hat{\underline{Q}}\underline{x}_i + 2\underline{x}_i'\underline{M}\ \underline{u}_i + \underline{u}_i'\hat{\underline{R}}\ \underline{u}_i \quad (A.10)$$

APPENDIX B.I

Consider the following optimal control law equation

$$\underline{u}_i^* = -\{\hat{\underline{R}}^{-1}\underline{M}' + \hat{\underline{R}}^{-1}\underline{D}'\underline{P}_{i+1}[\underline{I} + \underline{D}\,\hat{\underline{R}}^{-1}\underline{D}'\underline{P}_{i+1}]^{-1}\underline{\theta}\ \}\underline{x}_i \qquad (B,I,1)$$

Let us take the second term inside the parantehsis, and successively apply the following operations.

1. $\hat{\underline{R}}^{-1}\underline{D}'\underline{P}_{i+1}[\underline{I} + \underline{D}\,\hat{\underline{R}}^{-1}\underline{D}'\underline{P}_{i+1}]^{-1}$

2. $[(\underline{I} + \underline{D}\hat{\underline{R}}^{-1}\underline{D}'\underline{P}_{i+1})\ (\hat{\underline{R}}^{-1}\underline{D}'\underline{P}_{i+1})^{-1}]^{-1}$

3. $[(\hat{\underline{R}}^{-1}\underline{D}'\underline{P}_{i+1})^{-1} + \underline{D}]^{-1}$

4. $[(\underline{D}'\underline{P}_{i+1})^{-1}\hat{\underline{R}} + \underline{D}]^{-1}(\underline{D}'\underline{P}_{i+1})^{-1}(\underline{D}'\underline{P}_{i+1})$

5. $\{(\underline{D}'\underline{P}_{i+1})[(\underline{D}'\underline{P}_{i+1})^{-1}\hat{\underline{R}} + \underline{D}]\}^{-1}\underline{D}'\underline{P}_{i+1}$

6. $[\hat{\underline{R}} + \underline{D}'\underline{P}_{i+1}\underline{D}]^{-1}D'P_{i+1}$

This final form of the step 1 has some superority over its initial form. Since the former requires five matrix multiplications and three matrix inversions, while the latter four multiplications and one inversion.

If now the second term inside the paranthesis is replaced by its equivalent:

$$\underline{u}_i^* = -\{\hat{R}^{-1}M'++[\hat{R} + D'P_{i+1}D]^{-1}D'P_{i+1}\theta\}\ x_i \qquad (B.I.2)$$

is obtained.

B.II. MATRIX INVERSION LEMMA:

Consider the (nxn) square matrix partitioned as follows

$$\underline{A} = \left[\begin{array}{c|c} \underline{A}_{11} & \underline{A}_{12} \\ \hline \underline{A}_{21} & \underline{A}_{22} \end{array}\right]$$

Now, assuming that the inverse of $\underline{A}$ exists, i.e., $\underline{A}$ is non-singular; we define the following equality

$$\underline{B} = \underline{A}^{-1}$$

where $\underline{B}$ is partitioned as $\underline{A}$, i.e.,

$$\underline{B} = \left[\begin{array}{c|c} \underline{B}_{11} & \underline{B}_{12} \\ \hline \underline{B}_{21} & \underline{B}_{22} \end{array}\right]$$

If Gaus-Jordan elimination procedure is applied to $\underline{A}$

matrix to find its inverse, then:

$$\left[\begin{array}{cc|cc} \underline{A}_{11} & \underline{A}_{12} & \underline{I} & \underline{0} \\ \hline \underline{A}_{21} & \underline{A}_{22} & \underline{0} & \underline{I} \end{array}\right] \qquad \left[\begin{array}{c|c|c|c} \underline{I} & \underline{A}_{11}^{-1} \; \underline{A}_{12} & \underline{A}_{11}^{-1} & \underline{0} \\ \hline \underline{A}_{21} & \underline{A}_{22} & \underline{0} & \underline{I} \end{array}\right]$$

$$\left[\begin{array}{c|c|c|c} \underline{I} & -\underline{A}_{21} \; \underline{A}_{11} \; \underline{A}_{12} & -\underline{A}_{21} \; \underline{A}_{11}^{-1} & \underline{0} \\ \hline \underline{0} & \underline{A}_{22} \; -\underline{A}_{21} \; \underline{A}_{11}^{-1} & -\underline{A}_{21} \; \underline{A}_{11}^{-1} & \underline{I} \end{array}\right]$$

$$\left[\begin{array}{c|c|c|c} \underline{I} & -\underline{A}_{21} \; \underline{A}_{11}^{-1} \; \underline{A}_{12} & -\underline{A}_{21} \; \underline{A}_{11}^{-1} & \underline{0} \\ \hline \underline{0} & \underline{I} & -\underline{F}^{-1} \; \underline{A}_{21} \; \underline{A}_{11}^{-1} & \underline{F}^{-1} \end{array}\right]$$

where

$$\underline{F}^{-1} = \left[\underline{A}_{22} - \underline{A}_{21} \; \underline{A}_{11}^{-1} \; \underline{A}_{12} \right]^{-1}$$

finally,

$$\left[\begin{array}{c|c} \underline{B}_{11} & \underline{B}_{12} \\ \hline \underline{B}_{21} & \underline{B}_{22} \end{array}\right] = \left[\begin{array}{cc|cc} \underline{I} & \underline{0} & -\underline{A}_{21} \; \underline{A}_{11}^{-1} - \underline{A}_{21} \; \underline{A}_{11}^{-1} \; \underline{A}_{12} \; \underline{F}^{-1} \; \underline{A}_{21} \; \underline{A}_{11}^{-1} & \underline{A}_{21} \; \underline{A}_{11}^{-1} \; \underline{A}_{12} \; \underline{F}^{-1} \\ \hline \underline{0} & \underline{I} & -\underline{F}^{-1} \; \underline{A}_{21} \; \underline{A}_{11}^{-1} & \underline{F}^{-1} \end{array}\right]$$

since $\underline{B}_{22} = \underline{F}^{-1} \rightarrow \underline{F} = \underline{B}_{22}^{-1} = \underline{A}_{22} - \underline{A}_{21} \; \underline{A}_{11}^{-1} \; \underline{A}_{12}$

The $\underline{A}^{-1} = \underline{B}$ can obviously be rewritten in a different form by simply reordering the subscripts

$$\underline{B}_{11}^{-1} = \underline{A}_{11} - \underline{A}_{12} \; \underline{A}_{22}^{-1} \; \underline{A}_{21}$$

$$\underline{B}_{11} = \left[\underline{A}_{11} - \underline{A}_{12} \; \underline{A}_{22}^{-1} \; \underline{A}_{22} \right]^{-1}$$

If now these two values of $\underline{B}_{11}$ is equated, the following useful form for matrix inversion is obtained.

$$\left[\underline{A}_{11} - \underline{A}_{12}\,\underline{A}_{22}^{-1}\,\underline{A}_{21}\right]^{-1} = \underline{A}_{11}^{-1} + A_{11}^{-1}\,\underline{A}_{12}\left[\underline{A}_{22} - \underline{A}_{21}\,\underline{A}_{11}^{-1}\,\underline{A}_{12}\right]^{-1}\underline{A}_{21}\,\underline{A}_{11}^{-1}$$

Instead if we had $-A_{12}$ for $\underline{A}_{12}$, then all the terms multiplied by $\underline{A}_{12}$ will change its sign.

$$\left[\underline{A}_{11} + \underline{A}_{12}\,\underline{A}_{22}^{-1}\,\underline{A}_{21}\right]^{-1} = \underline{A}_{11}^{-1} - \underline{A}_{11}^{-1}\,\underline{A}_{12}\left[\underline{A}_{22} + \underline{A}_{21}\,\underline{A}_{11}^{-1}\,\underline{A}_{12}\right]^{-1}\underline{A}_{21}\,\underline{A}_{11}^{-1}$$

This result may be applied now to equation (2.33a) of Chapter 2. (2.33a) is repeated here for convenience.

$$\underline{P}_i = \theta'\underline{P}_{i+1}\left[\underline{I} + \underline{D}\underline{\hat{R}}^{-1}\underline{D}'\underline{P}_{i+1}\right]^{-1}\underline{\theta} + \underline{\Gamma} \qquad \text{(B.II.1)}$$

The result of the above derivation is applied to the matrix $[I + D\hat{R}^{-1}D'P_{i+1}]$ using the following definitions

$$\underline{A}_{11} = \underline{I}$$
$$\underline{A}_{12} = \underline{D}$$
$$\underline{A}_{22} = \underline{\hat{R}}$$
$$\underline{A}_{21} = \underline{D}'\underline{P}_{i+1}$$

Consequently,

$$[\underline{I} + \underline{D}\underline{\hat{R}}^{-1}\underline{D}'\underline{P}_{i+1}]^{-1} = \underline{I}^{-1} - \underline{I}^{-1}\underline{D}[\underline{\hat{R}} + \underline{D}'\underline{P}_{i+1}\underline{I}^{-1}\underline{D}]^{-1}\underline{D}'\underline{P}_{i+1}\underline{I}^{-1}$$

$$= \underline{I} - \underline{D}\,[\underline{\hat{R}} + \underline{D}'P_{i+1}\underline{D}]^{-1}\underline{D}'\underline{P}_{i+1}$$

If this equivalent form is replaced into equation (B.II.1) then,

$$\underline{P}_i = \underline{\theta}'\underline{P}_{i+1}\{\underline{I} - \underline{D}[\hat{\underline{R}} + \underline{D}'\underline{P}_{i+1}\underline{D}]^{-1}\underline{D}'\underline{P}_{i+1}\}\ \underline{\theta} + \underline{\Gamma}$$

$$\underline{P}_i = \underline{\theta}'\ \{\underline{P}_{i+1} - \underline{P}_{i+1}\underline{D}[\hat{\underline{R}} + \underline{D}'\underline{P}_{i+1}\underline{D}]^{-1}\underline{D}'\underline{P}_{i+1}\}\ \underline{\theta} + \underline{\Gamma}$$

(B.II.2)

is obtained.

APPENDIX C

Consider equation (2.33) of Chapter 2 which states that the boundary condition of $R_N E.$ is equal to S. This boundary condition is very simply derived. Reference to the performance index (2.11) shows that

$$J[\underline{x}_N, t_N; \underline{u}(.)] = \frac{1}{2}\, \underline{x}_N{}' \underline{S}\underline{x}_N \text{ for all } \underline{u}(.),$$

and accordingly, the minimum value of this performance index with respect to u(.) is also $\frac{1}{2}\, \underline{x}_N{}' \underline{S}\, \underline{x}_N$, that is,

$$J^*[\underline{x}_N, t_N] = \frac{1}{2}\, \underline{x}_N{}' \underline{S}\, \underline{x}_N$$

On the other hand J* is given by equation (2.34) as

$$J^*[\underline{x}_j, t_j; t_N; u^*] = \frac{1}{2}\, \underline{x}_j{}' \underline{P}_j \underline{x}_j$$

for j = N.

$$J^*[\underline{x}_N, t_N; \underline{u}^*] = \frac{1}{2}\, \underline{x}_N{}' \underline{P}_N \underline{x}_N$$

If these two values of J* is equated,

$$\underline{x}_N{}' \underline{P}_N \underline{x}_N = \underline{x}_N{}' \underline{S}\, \underline{x}_N$$

Since both $\underline{P}_N$ and $\underline{S}$ are symmetric, and $\underline{x}_N$ is arbitrary, then,

$$\underline{P}_N = \underline{S}$$

APPENDIX D

The scalar Riccati difference equation, equation ( ) can be rewritten as,

$$P_i + \frac{d^2}{r} P_i P_{i+1} - (\theta^2 + \nu \frac{d^2}{r}) P_{i+1} - \gamma = 0 \qquad (D.1)$$

$$P_N = 0$$

Since all products commute.

Equation (D.1) is a nonlinear first order difference equation that can be transformed to a second order linear one by the following change of variables:

$$P_i = \frac{r}{d^2} \frac{V_{i-1} - V_i}{V_i} \qquad (D.2)$$

The resulting equation is,

$$V_i[V_{i-1} - (1 + \theta^2 + \gamma \frac{d^2}{r})V_i + \theta^2 V_{i+1}] = 0 \qquad (D.3)$$

To obtain the nontrivial solution, let $V_i - s^i$, and obtain the characteristic equation

$$\theta^2 s^2 - (1 + \theta^2 + \gamma \frac{d^2}{r})s + 1 = 0 \qquad (D.4)$$

The two roots are

$$s_1, s_2 = \frac{1}{|\theta|} e^{\pm|\alpha|}$$

where

$$\alpha = \cosh^{-1} \frac{1 + \theta^2 + \frac{\gamma d^2}{2}}{2|\theta|}$$

The general solution can be written as

$$V_i = \frac{1}{|\theta|^i} (C_1 e^{i\alpha} + C_2 e^{-i\alpha})$$

Application of the boundary condition $R_N = 0$, yields finally that

$$V_i = \frac{1}{|\theta|^i} [(1 - |\theta| e^{-\alpha}) e^{(N-i)\alpha} - (1 - |\theta| e^{\alpha}) e^{-(N-i)\alpha}] \quad (D.5)$$

```
SIS*AYZAN1(1).MAIN
      IMPLICIT REAL*8(A-H,O-Z)
      COMPLEX COP,EIGE
      DIMENSION STATE(50),CONT(50),TIME(50)
      DIMENSION A1(10,10),B(10,10),C(10,10),Q(10,10),R(10,
     110),CTR(10,10),DUMY(10,10),QHAT(10,10),RHAT(10,10),
     1EM(10,10),CQCTR(10,10),PHI(10,10),D(10,10),DI(10,10)
     1,S(10,10),X(10),F(10,10)
      COMMON TIP1,R,QHAT,EM,RHAT
      COMMON/DIC/T,IR,JW,B,PHI,D,DI
      COMMON/EIDC/A1,N
      COMMON/EIC/EPS,KMAX,COP(11),EIGE(10),JJJ
      COMMON/STA/JDC
      COMMON/CWO/CQCTR,KN
      COMMON/ICFR/S,X,IFT
      COMMON/OMAR/XK(10,10),UOPT(10,10)
      READ(5,100)KMAX,EPS
  100 FORMAT(I4,E10.1)
      READ(5,1)N,IR,M,IFT
    1 FORMAT(4I5)
      WRITE(6,101)N,IR,M,IFT
  101 FORMAT('1',21X,'N =',I3/22X,'IR=',I3/22X,'M =',I3/22X,'FT=',I3)
      DO 15 I=1,N
   15 READ(5,13)(A1(I,J),J=1,N)
      DO 18 K=1,N
   18 READ(5,13)(B(K,L),L=1,IR)
      DO 19 I=1,N
   19 READ(5,13)(C(I,J),J=1,M)
      DO 21 K=1,M
   21 READ(5,13)(Q(K,L),L=1,M)
      DO 22 I=1,IR
   22 READ(5,13)(R(I,J),J=1,IR)
      DO 51 I=1,M
   51 READ(5,13) (F(I,J),J=1,M)
      READ(5,13)(X(I),I=1,N)
      WRITE(6,103)
  103 FORMAT(22X,'CONTINUOUS SYSTEM MATRIX A,'//)
      DO 104 I=1,N
  104 WRITE(6,20)(A1(I,J),J=1,N)
      WRITE(6,105)
  105 FORMAT(///,22X,'CONTINUOUS INPUT MATRIX B,'//)
      DO 106 K=1,N
  106 WRITE(6,20)(B(K,L),L=1,IR)
      WRITE(6,107)
  107 FORMAT(///,22X,'CONTINUOUS OUTPUT MATRIX C,'//)
      DO 108 I=1,N
  108 WRITE(6,20)(C(I,J),J=1,M)
      WRITE(6,109)
  109 FORMAT(///,22X,'STATE WEIGHTING MATRIX Q,'//)
      DO 210 K=1,M
  210 WRITE(6,20)(Q(K,L),L=1,M)
      WRITE(6,110)
  110 FORMAT(///,22X,'CONTROL WEIGHTING MATRIX R,'//)
      DO 220 I=1,IR
  220 WRITE(6,20)(R(I,J),J=1,IR)
      WRITE(6,111)
  111 FORMAT(///,22X,'FINAL COST MATRIX F,'//)
      DO 510 I=1,M
  510 WRITE(6,20)(F(I,J),J=1,M)
      WRITE(6,102)
  102 FORMAT(///,22X,'GIVEN INITIAL CONDITIONS,'//)
```

```
              WRITE(6,20)(X(I),I=1,N)
       13     FORMAT(10F8.0)
              JJJ=2
              JDC=2
              CALL STABIL ($88)
              JW=4
              CALL SAMPLE($88)
              CALL DISCO
              CALL CONTR($88,PHI,D,N,IR)
              CALL OBSERV($88,PHI,C,N,M)
              CALL LOADM(A1,DUMY,N,N)
              CALL LOADM(PHI,A1,N,N)
              JJJ=2
              JDC=1
              CALL STABIL($88)
              CALL LOADM(DUMY,A1,N,N)
              JJJ=1
              CALL EIGEN
              CALL MULTIQ(C,Q,DUMY,N,M,M)
              CALL TRANS(C,CTR,N,M)
              CALL MULTIQ(DUMY,CTR,CQCTR,N,M,N)
              DO 5 I=1,3
              KN=I
              CALL INTBSR
              GO TO(4,6,8)I
       4      WRITE(6,30)
              DO 25 K=1,N
       25     WRITE(6,20)(QHAT(K,L),L=1,N)
              GO TO 5
       6      WRITE(6,40)
              DO 35 K=1,N
       35     WRITE(6,20)(EM(K,L),L=1,IR)
              GO TO 5
       8      WRITE(6,50)
              DO 45 K=1,IR
       45     WRITE(6,20)(RHAT(K,L),L=1,IR)
       5      CONTINUE
         20   FORMAT(22X,1P3E17.6)
         30   FORMAT('1'///22X,'DISCRETE QHAT MATRIX IS,'//)
         40   FORMAT(///22X,'DISCRETE M MATRIX IS,'//)
         50   FORMAT(///22X,'DISCRETE RHAT MATRIX IS,'//)
              CALL MULTIQ(C,F,DUMY,N,M,M)
              CALL MULTIQ(DUMY,CTR,S,N,M,N)
              CALL RICATI($55,$65,$75)
              DO 27J=1,N
              DO 17I=1,IFT
              STATE(I)=XK(J,I)
              CONT(I)=UOPT(J,I)
        17    TIME(I)=I*T
              CALL GRAPH4(5.,8.,IFT,TIME,STATE)
              IF(J.GT.IR) GO TO 27
              CALL GRAPH4(5.,8.,IFT,TIME,CONT)
         27   CONTINUE
              GO TO 88
       55     WRITE(6,2)
       2      FORMAT(//10X,'INVERS OF RHAT MATRIX DOES NOT EXIST.')
              GO TO 88
       65     WRITE(6,3)
       3      FORMAT(//10X,'INVERS OF TP MATRIX DOES NOT EXIST.')
              GO TO 88
       75     WRITE(6,7)
```

```
      7   FORMAT(//10X,'INVERS OF G2 MATRIX DOES NOT EXIST.')
     88   STOP
          END
```

```
SIS*AYZAN1(1).SUB1
      SUBROUTINE STABIL($)
      IMPLICIT REAL*8(A-H,O-Z)
      COMPLEX COP,EIGE
      COMMON/STA/JDORC
      COMMON/EIDC/A1,N
      COMMON/EIC/EPS,KMAX,COP(11),EIGE(10),JJJ
      DIMENSION A1(10,10),EIGV(10)
      CALL EIGEN
      GO TO (4,6) JDORC
    4 DO 1 J=1,N
      EIGV(J)=CABS(EIGE(J))
      IF(EIGV(J).GT.1.) GO TO 2
    1 CONTINUE
      WRITE(6,3)
    3 FORMAT(///21X,'DISCRETE SYSTEM IS STABLE,ABSOLUTE VALUES'/
     122X,'OF ALL THE ROOTS ARE LESS THAN ONE.'//)
      RETURN
    2 WRITE(6,5)
    5 FORMAT(///21X,'DISCRETE SYSTEM IS UNSTABLE...A ROOT GREATER'/
     122X,'THAN ONE IN ABSOLUTE VALUE IS DETECTED.'//)
      RETURN 1
    6 DO 7 I=1,N
      IF(REAL(EIGE(I)).GT.0.) GOTO 9
    7 CONTINUE
      WRITE(6,8)
    8 FORMAT(///22X,'GIVEN CONTINIOUS SYSTEM IS STABLE.'//)
      RETURN
    9 WRITE(6,10)
   10 FORMAT(///22X,'GIVEN CONTINIOUS SYSTEM IS UNSTABLE.'//)
      RETURN 1
      END
```

```
5IS*AYZAN1(1).SUB2
      SUBROUTINE EIGEN
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/EIDC/A1,N
      COMMON/EIC/EPS,KMAX,A(11),XR(10),JJJ
      DIMENSION A1(10,10),B(10,10),C(10,10),P(10),COEFF(11)
      COMPLEX A,XR
      DO 1 I=1,N
      DO 1 J=1,N
    1 B(I,J)=A1(I,J)
      DO 20M=1,N
      P(M)=0.0
      FM=M
      DO 21 I=1,N
   21 P(M)=P(M)+B(I,I)
      P(M)=P(M)/FM
      DO 3 I=1,N
    3 B(I,I)=B(I,I)-P(M)
      CALL MULTIQ(A1,B,C,N,N,N)
      DO 6 K=1,N
      DO 6 L=1,N
    6 B(K,L)=C(K,L)
   20 CONTINUE
      KK=N
      DO 4 I=1,N
      IM1=I-1
      IF(DABS(P(N-IM1)).LT.EPS)GO TO 25
      GO TO 24
   25 KK=KK-1
    4 CONTINUE
   24 DO 40I=1,KK
      J=KK-I+1
   40 A(J)=CMPLX(-P(I),0.0)
      KP1=KK+1
      A(KP1)=CMPLX(1.0,0.0)
      DO 31 I=1,KP1
   31 COEFF(I)=REAL(A(I))
      CALL ROOTCP(A,KK,EPS,KMAX,XR,IQ,$2)
      NU=N-KK
      IF(NU)43,43,42
   42 DO 41 J=1,NU
      JJ=KK+J
   41 XR(JJ)=(0.0,0.0)
   43 IF(JJJ.EQ.1) GO TO 32
      WRITE(6,100)
  100 FORMAT(///22X,'THE INPUT AND OUTPUT FOR SUBROUTINE EIGEN'//)
      WRITE(6,101)N,EPS,KMAX
  101 FORMAT(22X,'N    =',I3/22X,'EPS =',1PE15.6/22X,'KMAX=',I3)
      WRITE(6,107)((A1(I,J),J=1,N),I=1,N)
  107 FORMAT(///22X,'INPUT MATRIX'//(22X,1P3E15.6))
      WRITE(6,103)(I,COEFF(I),I=1,KP1)
  103 FORMAT(///22X,'COEFFICIENTS OF CHARACTERISTIC POLYNOMIAL
     1,'//22X,'I',10X,'COEFF(I)'/(19X,I4,5X,1PE17.6))
      GO TO 30
    2 WRITE(6,105)IQ
  105 FORMAT(///22X,'ERROR RETURN...MAXIMUM NUMBER OF ITERATIONS'/
     122X,'EXCEEDED DURING THE SOLUTION FOR',I4,' TH ROOT.')
      GO TO 32
   30 WRITE(6,106)(I,XR(I),I=1,IQ)
      IF(NU.GT.0)WRITE(6,110)(I,XR(I),I=KK+1,N)
  106 FORMAT(///22X,'ROOTS OF CHARACTERISTIC POLYNOMIAL.'//22X,
```

```
     1'I',16X,'XR(I)'/(19X,I4,1P2E17.6))
      WRITE(6,109)NU
  109 FORMAT(//22X,'MULTIPLICITY OF ZERO ROOTS IS,NU=',I3)
  110 FORMAT(///22X,'ZERO ROOTS.'//22X,'I',16X,'XR(I)'/(19X,I4,
     11P2E17.6))
   32 RETURN
      END
```

```
SIS*AYZAN1(1).SUB3
                SUBROUTINE MULTIQ(A1,B,C,N1,N2,N3)
                IMPLICIT REAL*8(A-H,O-Z)
                DIMENSION A1(10,10),B(10,10),C(10,10)
                DO11 I=1,N1
                DO11 K=1,N3
                C(I,K)=0.
                DO11 J=1,N2
          11    C(I,K)=C(I,K)+A1(I,J)*B(J,K)
                RETURN
                END
```

S*AYZAN1(1).SUB4

```
      SUBROUTINE INTBSR
      IMPLICIT REAL*8(A-H,O-Z)
      COMPLEX COP,EIGE
      DIMENSION EIGV(10),EXPO1(10,10),VINT(10,10),DUMY(10,10),
     1VINTAM(10,10),TR(10,10),FTI(10,10),FTIN(10,10),FU(10,10),
     1QBAR(10,10),EM(10,10),RBAR(10,10),R(10,10),A1(10,10),B(10,10),
     1FONK(10,10)
      COMMON TIP1,R,QBAR,EM,RBAR
      COMMON/EIDC/A1,N
      COMMON/EIC/EPS,KMAX,COP(11),EIGE(10),JJJ
      COMMON/DIC/T,IR,JW,B,EXPO,DB,DI
      COMMON/CWI/FONK,MH
      COMMON/CWO/FU,KONTRL
      DO 1 J=1,N
    1 EIGV(J)=CABS(EIGE(J))
      DEIGEN=EIGV(1)
      DO 2 K=2,N
    2 DEIGEN=AMAX1(DEIGEN,EIGV(K))
      H=1./(DEIGEN*20.)
      NIT=IFIX(TIP1/H)
      H=TIP1/FLOAT(NIT)
      HOV2=H/2.
      IF(KONTRL-2)4,6,8
    4 CALL SIFIR(QBAR,N,N)
      GO TO 10
    6 CALL SIFIR(EM,N,IR)
      GO TO 10
    8 CALL SIFIR(RBAR,N,N)
   10 T=HOV2
      CALL EVOINT
      CALL LOADM(FONK,VINTAM,N,MH)
      NITM1=NIT-1
      DO 5 I=1,NITM1
      T=H*I
      CALL EVOINT
      CALL LOADM(FONK,EXPO1,N,MH)
      IF(KONTRL-2)40,41,42
   40 CALL TOPLA(QBAR,EXPO1,N,N)
      GO TO 24
   41 CALL TOPLA(EM,EXPO1,N,IR)
      GO TO 24
   42 CALL TOPLA(RBAR,EXPO1,N,N)
   24 T=T+HOV2
      CALL EVOINT
      CALL LOADM(FONK,VINT,N,MH)
      IF(KONTRL-2)50,51,50
   50 CALL TOPLA(VINTAM,VINT,N,N)
      GO TO 5
   51 CALL TOPLA(VINTAM,VINT,N,IR)
    5 CONTINUE
      T=0.
      CALL EVOINT
      CALL LOADM(FONK,FTI,N,MH)
      T=TIP1
      CALL EVOINT
      CALL LOADM(FONK,FTIN,N,MH)
      GO TO(72,73,74)KONTRL
   72 CALL SIMFOR(H,FTI,FTIN,VINTAM,QBAR,N,N)
      GO TO 68
   73 CALL SIMFOR(H,FTI,FTIN,VINTAM,EM,N,IR)
```

```
      GO TO 68
74    CALL SIMFOR(H,FTI,FTIN,VINTAM,RBAR,N,N)
      CALL TRANS(B,BTR,N,IR)
      CALL MULTIQ(BTR,RBAR,DUMY,IR,N,N)
      CALL MULTIQ(DUMY,B,RBAR,IR,N,IR)
      DO 3 I=1,IR
      DO 3 J=1,IR
3     TR(I,J)=TIPI*R(I,J)
      CALL TOPLA(RBAR,TR,IR,IR)
68    RETURN
      END
```

SIS*AYZAN1(1).SUB5

```
      SUBROUTINE EVOINT
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/CWO/FU,KONTRL
      COMMON/EIDC/A1,N
      COMMON/DIC/T,IR,JW,B,EXPO,DB,DI
      COMMON/CWI/EVFU,MH
      DIMENSION A1(10,10),B(10,10),EXPO(10,10),DB(10,10),DI(10,10)
      DIMENSION EVFU(10,10),EXPOTR(10,10),DUMY(10,10),FU(10,10)
      IF(KONTRL-2)12,13,14
   12 JW=2
      GO TO 25
   13 JW=1
   25 CALL DISCO
      IF(KONTRL.EQ.3)CALL LOADM(DI.EXPO,N,N)
      CALL TRANS(EXPO,EXPOTR,N,N)
      CALL MULTIQ(EXPOTR,FU,DUMY,N,N,N)
      IF(KONTRL.EQ.2) GO TO 15
      CALL MULTIQ(DUMY,EXPO,EVFU,N,N,N)
      GO TO 9
   15 CALL MULTIQ(DUMY,DB,EVFU,N,N,IR)
      GO TO 9
   14 JW=3
      GO TO 25
    9 MH=N
      IF(KONTRL.EQ.2)MH=IR
      RETURN
      END
```

```
IS*AYZAN1(1).SUB6
              SUBROUTINE DISCO
              IMPLICIT REAL*8(A-H,O-Z)
              INTEGER POWER
              DIMENSION  D(10,10),ST(10,10),A1(10,10),B(10,10),PSI(10,10)
              REALINTEGA(10,10),NORMA(10)
              COMMON/EIDC/A1,N
              COMMON/DIC/T,M,NW,B,PSI,D,INTEGA
              DATA EP1/1.D-20/
              DO 1 I= 1,N
              NORMA(I) = 0.0
              DO 1 J= 1,N
              ST(I,J)=A1(I,J)*T
              NORMA(I) =NORMA(I)+ABS(ST(I,J))
        1     PSI(I,J)=ST(I,J)
              ANORM=NORMA(1)
              DO 2 K= 2,N
        2     ANORM=AMAX1(ANORM,NORMA(K))
              POWER=10
              IF(POWER.LE.IFIX(ANORM))POWER=POWER+ANORM
       14     DO 7 I= 2,POWER
              APOWR=POWER-I+2
              DO 5 J= 1,N
              DO 3 K= 1,N
        3     INTEGA(J,K) = PSI(J,K)/APOWR
        5     INTEGA(J,J)=INTEGA(J,J)+1.0
        7     CALL MULTIQ(ST,INTEGA,PSI,N,N,N)
              DO 12 J= 1,N
       12     PSI(J,J)=PSI(J,J)+1.0
              EPS=ANORM/(POWER+2)
              IX=POWER+1
              UPP=ANORM**IX
              DO 4 J= 1,POWER
              APOW=POWER-J+2
        4     UPP=UPP/APOW
              UPP=UPP/(1-EPS)
              DO 8 K= 1,N
              DO 8 L= 1,N
              IF(DABS(PSI(K,L)).LE.EP1)GO TO 8
              IF(DABS(UPP).GE.DABS(PSI(K,L)*1.E-5)) GO TO 6
        8     CONTINUE
              GO TO 25
        6     POWER=POWER+10
              GO TO 14
       25     IF(NW.EQ.2) GO TO 28
              DO 9 J= 1,N
              DO 9 K= 1,N
        9     INTEGA(J,K)=T*INTEGA(J,K)
              IF(NW.EQ.3) GO TO 28
              CALL MULTIQ(INTEGA,B,D,N,N,M)
              IF(NW.NE.4) GO TO 28
       24     WRITE(6,36)
          36 FORMAT('1'///20X,'THE INPUT AND OUTPUT FOR SUBROUTINE DISCO'//)
              WRITE(6,21)
          21 FORMAT(///21X,'THE A MATRIX.'//)
              DO 10 I= 1,N
       10     WRITE(6,20)(A1(I,J),J=1,N)
              WRITE(6,11)
          11 FORMAT(///21X,'THE B MATRIX.'//)
              DO 22 I= 1,N
       22     WRITE(6,20)(B(I,J),J=1,M)
```

```
      WRITE(6,16)
   16 FORMAT(///21X,'THE DISCRETE SYSTEM MATRIX PHI.'//)
      DO 17 I = 1,N
17    WRITE(6,20)(PSI(I,J),J=1,N)
      WRITE(6,19)
   19 FORMAT(///21X,'THE DISCRETE INPUT MATRIX D.'//)
      DO 23 I = 1,N
23    WRITE(6,20)(D(I,J),J=1,M)
   20 FORMAT(22X,1P3E17.6)
      WRITE(6,26)EPS,ANORM,POWER
   26 FORMAT(///22X,'EPS   =',1PE15.6/22X,'ANORM=',1PE15.6/22X,
     1'POWER=',I4)
28    RETURN
      END
```

```
HESIS*AYZAN1(1).SUB7
              SUBROUTINE SIFIR(SI,N3,N4)
              IMPLICIT REAL*8(A-H,O-Z)
              DIMENSION SI(10,10)
              DO 15 I=1,N3
              DO 15 J=1,N4
       15     SI(I,J)=0.0
              RETURN
              END
```

```
IS*AYZAN1(1).SUB8
              SUBROUTINE TOPLA(Q,E,M3,M4)
              IMPLICIT REAL*8(A-H,O-Z)
              DIMENSION Q(10,10),E(10,10)
              DO 4 K=1,M3
              DO 4 L=1,M4
        4     Q(K,L)=Q(K,L)+E(K,L)
              RETURN
              END
```

```
IS*AYZAN1(1).SUB9
          SUBROUTINE SIMFOR(H,AI,AI1,AI2,AI3,M5,M6)
          IMPLICIT REAL*8(A-H,O-Z)
          DIMENSION AI(10,10),AI1(10,10),AI2(10,10),AI3(10,10)
          DO 7 I=1,M5
          DO 7 J=1,M6
    7     AI3(I,J)=(H/6.)*(AI(I,J)+AI1(I,J)+4.*AI2(I,J)+2.*AI3(I,J))
          RETURN
          END
```

```
SIS*AYZAN1(1).SUB10
          SUBROUTINE TRANS(A,AT,N1,N2)
          IMPLICIT REAL*8(A-H,O-Z)
          DIMENSION A(10,10),AT(10,10)
          DO 15 I= 1,N1
          DO 15 J=1,N2
       15 AT(J,I) = A(I,J)
          RETURN
          END
T,S AYZAN1.SUB11,.SUB12,.SUB13,.SUB14,.SUB15,.SUB16,.SUB17,.SUB18,.SUB19
```

```
ESIS*AYZAN1(1).SUB11
              SUBROUTINE LOADM(DULL,OPM,M1,M2)
              IMPLICIT REAL*8(A-H,O-Z)
              DIMENSIONDULL(10,10),OPM(10,10)
              DO 8 I=1,M1
              DO 8 J=1,M2
        8     OPM(I,J)=DULL(I,J)
              RETURN
              END
```

```
IS*AYZAN1(1).SUB12
              SUBROUTINE RICATI($,$,$)
              IMPLICIT REAL*8(A-H,O-Z)
              DIMENSION RHAT(10,10),QHAT(10,10),EM(10,10),RINV(10,10)
              DIMENSION EMT(10,10),DUMY(10,10),D(10,10),DRM(10,10),PHI(10,10)
              DIMENSION TETHA(10,10),RMT(10,10),GAMA(10,10),DTR(10,10)
              DIMENSION TETATR(10,10),S(10,10),P(10,10,10),PK(10,10),TP(10,10
              DIMENSION XK(10,10),X(10),XX(10,1),G1(10,10),G2(10,10),G3(10,10
              DIMENSION G4(10,10),G(10,10,10),U(10,1),UI(10,1),UOPT(10,10)
              DIMENSION D1(10,1),D2(10,1),R(10,10),A1(10,10),B(10,10),DI(10,1
              DIMENSION RIM(10,10),XKT(1,10),COST(1,1)
              COMMON TIP1,R,QHAT,EM,RHAT
              COMMON/DIC/T,IR,JW,B,PHI,D,DI
              COMMON/EIDC/A1,N
              COMMON/ICFR/S,X,IFT
               COMMON /OMAR/ XK,UOPT
              CALL LOADM(RHAT,RINV,IR,IR)
              LEO=1
              CALL INVERS($69,RINV,DET,IR,LEO)
              GO TO 21
        69    RETURN 1
        21    CALL TRANS(EM,EMT,N,IR)
              CALL MULTIQ(RINV,EMT,DUMY,IR,IR,N)
              CALL MULTIQ(D,DUMY,DRM,N,IR,N)
              CALL SUBTRT(PHI,DRM,TETHA,N,N)
              CALL MULTIQ(EM,DUMY,RMT,N,IR,N)
              CALL SUBTRT(QHAT,RMT,GAMA,N,N)
              CALL TRANS(D,DTR,N,IR)
              CALL TRANS(TETHA,TETATR,N,N)
              M=IFT+1
              CALL CIZERO(N,S,KS)
              IF(KS.EQ.N**2) GO TO 1
              JANE=-1
              DO 2 I=1,N
              DO 2 J=1,N
        2     P(I,J,M)=S(I,J)
              L=1
        15    K=M-L+1
              DO 4 I=1,N
              DO 4 J=1,N
        4     PK(I,J)=P(I,J,K)
              GO TO 12
         1    JANE=1
              CALL LOADM(S,PK,N,N)
        12    CONTINUE
              CALL MULTIQ(DTR,PK,DUMY,IR,N,N)
              CALL MULTIQ(DUMY,D,TP,IR,N,IR)
              CALL TOPLA(TP,RHAT,IR,IR)
              CALL INVERS($79,TP,DET,IR,LEO)
              GO TO 31
        79    RETURN 2
        31    CALL MULTIQ(TP,DTR,DUMY,IR,IR,N)
              CALL MULTIQ(DUMY,PK,TP,IR,N,N)
              CALL MULTIQ(D,TP,DUMY,N,IR,N)
              CALL MULTIQ(PK,DUMY,TP,N,N,N)
              CALL SUBTRT(PK,TP,DUMY,N,N)
              CALL MULTIQ(DUMY,TETHA,TP,N,N,N)
              CALL MULTIQ(TETATR,TP,DUMY,N,N,N)
              IF(JANE)17,18,18
        18    CALL TOPLA(DUMY,GAMA,N,N)
              CALL SUBTRT(DUMY,PK,G1,N,N)
```

```
      CALL LOADM(DUMY,PK,N,N)
      DO 16 I=1,N
      DO 16 J=1,N
      G1(I,J)=G1(I,J)*G1(I,J)
 16   IF(G1(I,J).GE.1.E-8) GO TO 12
      GO TO 19
 17   K=K-1
      DO 3 I=1,N
      DO 3J=1,N
 3    P(I,J,K)=DUMY(I,J)+GAMA(I,J)
      L=L+1
      IF(L.LE.M) GO TO 15
C     PRINT OUT P MATRIX
      WRITE(6,5)
    5 FORMAT('1',22X,'SOLUTION OF MATRIX RICCATI DIFFERENCE'/
     122X,'EQUATION P MATRICES ARE,'//)
      DO 6 I=1,M
      JJ=I-1
      WRITE(6,7)JJ,JJ
      DO 6 J=1,N
 6    WRITE(6,8)(P(J,JK,I),JK=1,N)
      GO TO 23
 19   PRINT *,'     STEADY STATE SOLUTION OF RICCATI MATRIX EQUATION.'
      DO 22 I=1,N
 22   WRITE(6,8)(PK(I,J),J=1,N)
    7 FORMAT(///22X,'AT THE SAMPLING INSTANT',I3,2X,'P(',I2,')
     1  MATRIX,'//)
    8 FORMAT(22X,1P3E17.6)
C     COMPUTE TIME VARYING FEEDBACK GAIN MATRICES,CONTROL& STATE VECT
 23   K=1
      DO 10 I=1,N
 10   XK(I,1)=X(I)
      PRINT *,'INITIAL  STATES  '
      WRITE(6,8)(XK(J,K),J=1,N)
 60   K=K+1
      DO 20 I=1,N
      XX(I,1)=X(I)
      IF(JANE)24,25,25
 24   CONTINUE
      DO 20 J=1,N
 20   PK(I,J)=P(I,J,K)
 25   IF(K.NE.2) GO TO 42
      CALL MULTIQ(DTR,PK,DUMY,IR,N,N)
      CALL MULTIQ(DUMY,D,G2,IR,N,IR)
      CALL MULTIQ(DUMY,TETHA,G3,IR,N,N)
      CALL TOPLA(G2,RHAT,IR,IR)
      CALL INVERS($89,G2,DET,IR,LEO)
      GO TO 41
 89   RETURN 3
 41   CALL MULTIQ(G2,G3,G4,IR,IR,N)
      CALL TOPLA(RIM,G4,IR,N)
      CALL LOADM(RIM,G1,IR,N)
 42   K=K-1
      IF(JANE)26,27,27
 26   DO 30 I=1,IR
      DO 30 J=1,N
 30   G(I,J,K)=G1(I,J)
C     PRINT OUT"FEEDBACK GAIN MATRIX G."
      JA=K-1
      WRITE(6,9)JA,JA
    9 FORMAT(///22X,'AT THE SAMPLING TIME',2X,'FEED BACK GAIN
```

```
     1'/22X,'MATRIX G(',I3,')'//)
      DO 11 I=1,IR
   11 WRITE(6,8)(G(I,J,K),J=1,N)
      GO TO 39
   27 JA=K-1
      IF(K.NE.2)GO TO 39
      PRINT *,'    STEADY STATE FEEDBACK GAIN MATRIX.'
      DO 28 I=1,IR
   28 WRITE(6,8)(G1(I,J),J=1,N)
C     COMPUTE 'CONTROL VECTOR U.'
   39 CONTINUE
      MM=1
      CALL MULTIQ(G1,XX,U,IR,N,MM)
      DO 40 I=1,IR
      UI(I,1)=-U(I,1)
   40 UOPT(I,K)=-U(I,1)
C     PRINT OUT 'CONTROL VECTOR U'.
      WRITE(6,13)JA,JA
   13 FORMAT(///22X,'AT THE SAMPLING TIME',I3,2X,'OPTIMUM
     1'/22X,'CONTROL VECTOR UOPT(',I2,')')
      WRITE(6,8)(UOPT(I,K),I=1,IR)
C     COMPUTE 'STATE VECTOR X'
      CALL MULTIQ(D,UI,D1,N,IR,MM)
      CALL MULTIQ(PHI,XX,D2,N,N,MM)
      K=K+1
      DO 50 I=1,N
      X(I)=D2(I,1)+D1(I,1)
   50 XK(I,K)=X(I)
C     PRINT OUT 'STATE VECTOR X.'
      KM1=K-1
      WRITE(6,14)JA,KM1
   14 FORMAT(///22X,'AT THE SAMPLING TIME',I3,2X,'NEXT STATE X('
     1,I2,') COMPUTED AS'//)
      WRITE(6,8)(XK(J,K),J=1,N)
      IF(K.NE.IFT+1)GO TO 60
C        COMPUTE COST
      IF(JANE)43,44,44
   43 DO 47 I=1,N
      DO 47 J=1,N
   47 PK(I,J)=P(I,J,1)
   44 CALL MULTIQ(PK,XK,DUMY,N,N,MM)
      CALL TRANS(XK,XKT,N,MM)
      CALL MULTIQ(XKT,DUMY,COST,MM,N,MM)
      PRINT*, '  OPTIMUM SOLUTION IS OBTAINED WITH THE FOLLOWING COST
      WRITE(6,35) COST(1,1)
   35 FORMAT(//,22X,'COST=',G12.5)
      RETURN
      END
```

```
IS*AYZAN1(1).SUB13
        SUBROUTINE INVERS($,A,DETER,N,JALE)
        IMPLICIT REAL*8(A-H,O-Z)
        DIMENSION IROW(10),JCOL(10),JORD(10),Y(10),A(10,10),A2(10,10)
        DATA EPS/1.D-20/
        DO 5 J=1,N
        DO 5 K=1,N
 5      A2(J,K)=A(J,K)
        DETER=1.
        DO18K=1,N
        KM1=K-1
        PIVOT=0.
        DO11I=1,N
        DO11J=1,N
        IF(K.EQ.1) GO TO 9
        DO 8 ISCAN=1,KM1
        DO 8 JSCAN=1,KM1
        IF(I.EQ.IROW(ISCAN)) GO TO 11
        IF(J.EQ.JCOL(JSCAN)) GO TO 11
 8      CONTINUE
 9      IF(DABS(A(I,J)).LE.DABS(PIVOT)) GO TO 11
        PIVOT=A(I,J)
        IROW(K)=I
        JCOL(K)=J
 11     CONTINUE
        IF(DABS(PIVOT).GT.EPS) GO TO 13
        RETURN 1
 13     IROWK=IROW(K)
        JCOLK=JCOL(K)
        DETER=DETER*PIVOT
        DO 14 J=1,N
 14     A(IROWK,J)=A(IROWK,J)/PIVOT
        A(IROWK,JCOLK)=1./PIVOT
        DO 18 I=1,N
        AIJCK=A(I,JCOLK)
        IF(I.EQ.IROWK) GO TO 18
        A(I,JCOLK)=-AIJCK/PIVOT
        DO 17 J=1,N
 17     IF(J.NE.JCOLK) A(I,J)=A(I,J)-AIJCK*A(IROWK,J)
 18     CONTINUE
        DO 20 I=1,N
        IROWI=IROW(I)
        JCOLI=JCOL(I)
 20     JORD(IROWI)=JCOLI
        INTCH=0
        NM1=N-1
        DO 22 I=1,NM1
        IP1=I+1
        DO 22 J=IP1,N
        IF(JORD(J).GE.JORD(I)) GO TO 22
        JTEMP=JORD(J)
        JORD(J)=JORD(I)
        JORD(I)=JTEMP
        INTCH=INTCH+1
 22     CONTINUE
        IF(INTCH/2*2.NE.INTCH) DETER=-DETER
        DO 28 J=1,N
        DO 27 I=1,N
        IROWI=IROW(I)
        JCOLI=JCOL(I)
 27     Y(JCOLI)=A(IROWI,J)
```

```
      DO 28 I=1,N
   28 A(I,J)=Y(I)
      DO 30 I=1,N
      DO 29 J=1,N
      IROWJ=IROW(J)
      JCOLJ=JCOL(J)
   29 Y(IROWJ)=A(I,JCOLJ)
      DO 30 J=1,N
   30 A(I,J)=Y(J)
      IF(JALE.EQ.1) GO TO 84
      WRITE(6,150)
  150 FORMAT('1',10X,'THE INPUT & OUTPUT FOR SUBROUTINE INVERS.'///)
      WRITE(6,151)N,EPS
  151 FORMAT(10X,'N   = ',I4/10X,'EPS= ',1PE12.4)
      WRITE(6,199)
  199 FORMAT(10X,'GIVEN MATRIX IS, '//)
      DO 4 I=1,N
    4 WRITE(6,200)(A2(I,J),J=1,N)
  200 FORMAT('0',5X,1P10E12.4)
      WRITE(6,201)DETER
  201 FORMAT('0',5X,' DETER= ',F12.6//10X,'THE INVERSE MATRIX.'//)
      DO 83 K=1,N
   83 WRITE(6,200)(A(K,L),L=1,N)
   84 RETURN
      END
```

```
$IS*AYZAN1(1).SUB14
          SUBROUTINE SUBTRT(A,B,C,M,N)
          IMPLICIT REAL*8(A-H,O-Z)
          DIMENSION A(10,10),B(10,10),C(10,10)
          DO 1 I=1,M
          DO 1 J=1,N
    1     C(I,J)=A(I,J)-B(I,J)
          RETURN
          END
```

```
$IS*AYZAN1(1).SUB15
      SUBROUTINE CONTR(5,PHI,D,N,IR)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION PHI(10,10),D(10,10),PHK(10,10,10),QC(10,10,10)
      DIMENSION PH(10,10),OP(10,10),QCON(10,10),QCT(10,10)
      DATA EPS/1.D-20/
      CALL DEVRET(PHI,PHK,N,N,1)
      CALL POPHI(PHI,PHK,N,KENT)
      DO 20 K=1,KENT
      KP1=K+1
      DO 25 I=1,N
      DO 25 J=1,N
   25 PH(I,J)=PHK(I,J,K)
      CALL MULTIQ(PH,D,OP,N,N,IR)
      CALL DEVRET(OP,QC,N,IR,KP1)
   20 CONTINUE
      CALL DEVRET(D,QC,N,IR,1)
      DO 15 K=1,N
      IOWA=(K-1)*IR+1
      JERSEY=K*IR
      DO 30 I=1,N
      M=0
      DO 30 J=IOWA,JERSEY
      M=M+1
   30 QCON(I,J)=QC(I,M,K)
   15 CONTINUE
      NIR=N*IR
      IF(NIR.GT.N) GO TO 35
      JALE=-1
      CALL INVERS($7,QCON,DEOCM,N,JALE)
      IF(DABS(DEOCM).GT.EPS) GO TO 8
      GO TO 7
   35 CALL TRANS(QCON,QCT,N,NIR)
      CALL RANKT(QCT,N,NIR,IRANK)
      IF(N-IRANK)7,8,7
    8 WRITE(6,5)
    5 FORMAT(///22X,'SYSTEM IS COMPLETELY CONTROLLABLE.')
      GO TO 9
    7 WRITE(6,6)
    6 FORMAT(///22X,'SYSTEM IS UNCONTROLLABLE.'//)
      RETURN 1
    9 RETURN
      END
```

```
IS*AYZAN1(1).SUB16
          SUBROUTINE DEVRET(B,P,N1,N2,K)
          IMPLICIT REAL*8(A-H,O-Z)
          DIMENSION P(10,10,10),B(10,10)
          DO 10 I=1,N1
          DO 10 J=1,N2
   10     P(I,J,K)=B(I,J)
          RETURN
          END
```

```
SIS*AYZAN1(1).SUB17
      SUBROUTINE RANKT(A,N,NIR,IRANK)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(10,10),IP(10)
      DATA EPS/1.D-20/
      DO 20 J=1,N
      JM1=J-1
      PIVOT =0.
      DO 10 I=1,NIR
      IF(J.EQ.1) GO TO 5
      DO 25 JSCAN=1,JM1
      IF(I.EQ.IP(JSCAN)) GO TO 10
   25 CONTINUE
    5 IF(DABS(A(I,J)).LE.DABS(PIVOT)) GO TO 10
      PIVOT=A(I,J)
      IP(J)=I
      IF(DABS(PIVOT).GT.EPS) GO TO 15
   10 CONTINUE
      IF(DABS(PIVOT).LE.EPS) GO TO 20
   15 DO 30 K=1,N
   30 A(IP(J),K)=A(IP(J),K)/PIVOT
      DO 20 I=1,NIR
      AIJ=-A(I,J)
      IF(I.EQ.IP(J)) GO TO 20
      DO 35 K=1,N
   35 A(I,K)=A(I,K)+AIJ*A(IP(J),K)
   20 CONTINUE
      IRANK=0
      DO 40 I=1,NIR
      DO 45 J=1,N
      IF(DABS(A(I,J)).GT.EPS) GO TO6
   45 CONTINUE
      GO TO 40
    6 IRANK=IRANK+1
   40 CONTINUE
      RETURN
      END
```

```
IS*AYZAN1(1).SUB18
            SUBROUTINE OBSERV($,PHI,C,N,M)
            IMPLICIT REAL*8(A-H,O-Z)
            DIMENSION PHI(10,10),CN(10,10),PHK(10,10,10),QOBS(10,10,10)
            DIMENSION QO(10,10),QOT(10,10),PH(10,10),PHT(10,10)
            DIMENSION DUMY(10,10)
            DATA EPS/1.D-20/
            CALL DEVRET(PHI,PHK,N,N,1)
            CALL POPHI(PHI,PHK,N,KENT)
            CALL DEVRET(C,QOBS,N,M,1)
            DO 20 K=1,KENT
            KP1=K+1
            DO 25 I=1,N
            DO 25 J=1,N
     25     PH(I,J)=PHK(I,J,K)
            CALL TRANS(PH,PHT,N,N)
            CALL MULTIQ(PHT,C,DUMY,N,N,M)
            CALL DEVRET(DUMY,QOBS,N,M,KP1)
     20     CONTINUE
            DO 15 K=1,N
            IOWA=(K-1)*M+1
            JERSEY=K*M
            DO 30 I=1,N
            M=0
            DO 30 J=IOWA,JERSEY
            M=M+1
         30 QO(I,J)=QOBS(I,M,K)
     15     CONTINUE
            NBM=N*M
            IF(NBM.GT.N) GO TO 35
            JALE=-1
            CALL INVERS($7,QO,DEOM,N,JALE)
            IF(DABS(DEOM).GT.EPS) GO TO 8
            GO TO 7
     35     CALL TRANS(QO,QOT,N,NBM)
            CALL RANKT(QOT,N,NBM,IRANK)
            IF(N-IRANK)7,8,7
          8 WRITE(6,1)
          1 FORMAT(//22X,'SYSTEM IS COMPLETELY OBSERVABLE.')
            GO TO 9
          7 WRITE(6,2)
          2 FORMAT(//22X,'SYSTEM IS UNOBSERVABLE.'///)
            RETURN 1
      9     RETURN
            END
```

```
SIS*AYZAN1(1).SUB19
              SUBROUTINE POPHI(PHI,PHK,N,KENT)
              IMPLICIT REAL*8(A-H,O-Z)
              DIMENSION PHI(10,10),PHK(10,10,10),PH(10,10),OP(10,10)
              CALL LOADM(PHI,PH,N,N)
              KENT=N-1
              DO 10 K=2,KENT
              CALL MULTIQ(PHI,PH,OP,N,N,N)
              CALL DEVRET(OP,PHK,N,N,K)
              CALL LOADM(OP,PH,N,N)
        10    CONTINUE
              RETURN
              END
T,S AYZAN1.SUB20,.SUB21,.SUB22
```

```
:SIS*AYZAN1(1).SUB20
         SUBROUTINE DEFNIT($,N,B)
         DIMENSION A(10,10),B(10,10),DOQF(10)
         DATA EPS/1.E-6/
         DO 15 K=1,N
         LEO=-1
         DO 20 I=1,K
         DO 20 J=1,K
      20 A(I,J)=B(I,J)
         CALL INVERS($2,A,DET,K,LEO)
         DOQF(K)=DET
      15 CONTINUE
         KONT=0
         KOUNT=0
         DO 1 K=1,N
         IF(DABS(DOQF(K))-EPS)3,5,5
       5 KOUNT=KOUNT+1
         IF(DOQF(K))2,3,4
       4 KONT=KONT+1
       1 CONTINUE
         IF(KONT-KOUNT)6,7,6
       2 PRINT *,'     INTRODUCED MATRIX IS NOT POSITIVE (SEMI) DEFINITE
      10 RETURN 1
       3 GO TO 2
       6 PRINT *,'     INTRODUCED MATRIX IS POSITIVE SEMI DEFINITE.'
         GO TO 8
       7 IF(KONT.EQ.N)GO TO 9
         PRINT *,'     DEFINITENESS OF INTRODUCED MATRIX IS NOT DETERMIN
         GO TO 10
       9 PRINT *,'     INTRODUCED MATRIX IS POSITIVE DEFINITE.'
       8 RETURN
         END
```

```
IS*AYZAN1(1).SUB21
      SUBROUTINE CIZERO(N,S,KSN)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION S(10,10)
      DATA EPS/1.D-20/
      KSN=0
      DO 1 I=1,N
      DO 1 J=1,N
      IF(DABS(S(I,J)).LE.EPS)KSN=KSN+1
    1 CONTINUE
      RETURN
      END
```

SIS*AYZAN1(1).SUB22

```
      SUBROUTINE SAMPLE(S)
      IMPLICIT REAL*8(A-H,O-Z)
      COMPLEX COP,EIG
      COMMON/EIDC/A1,N
      COMMON/EIC/EPS,KMAX,COP(11),EIG(10),JJJ,NU
      COMMON TIP1,R,QHAT,EM,RHAT
      COMMON/DIC/T,IR,JW,B,PHI,D,DI
      COMMON/CWO/QX,KN
      DIMENSION QHAT(10,10),EM(10,10),RHAT(10,10),A1(10,10)
     1,B(10,10),PHI(10,10),D(10,10),DI(10,10),QX(10,10),
     UEV(10),CEV(10),DUMY(10,10),RINV(10,10),EMT(10,10),RM
     1(10,10),QMR(10,10),AE(3),CTR(3)
      JJJ=1
      DO 1 I=1,N
      IF(AIMAG(EIG(I)).LT.EPS) GO TO 1
      GO TO 2
    1 CONTINUE
      DO 3 I=1,N
    3 EV(I)=REAL(EIG(I))
      AEV=EV(1)
      DO 4 K=2,N
    4 AEV=AMAX1(AEV,EV(K))
      T=1./(IK*AEV)
      GO TO 5
    2 DO 6 I=1,N
    6 CEV(I)=AIMAG(EIG(I))
      CE=CEV(1)
      DO 7 K=2,N
    7 CE=AMAX1(CE,CEV(K))
      PI=355./113.
      TMAX=PI/CE
      DO 12 I=1,N
   12 EV(I)=CABS(EIG(I))
      EVM=EV(1)
      DO 13 I=2,N
   13 EVM=AMAX1(EVM,EV(I))
      T=1./(25.*EVM)
      XT=TMAX/T
      IT=TMAX/T
      IF((XT-IT).EQ.0.) T=0.89*T
      GO TO 8
    5 IF(NU.NE.0) GO TO 8
      ALPHA=0.25
      CALL LOADM(A1,DUMY,N,N)
      DO 9 J=1,3
      T=ALPHA*T
      TIP1=T
      KN=1
      CALL INTBSR
   16 CALL LOADM(QHAT,A1,N,N)
      JJJ=1
      CALL EIGEN
      DO 11 I=1,N
   11 EV(I)=CABS(EIG(I))
      AEV=EV(1)
      DO 10 I=2,N
   10 AEV=AMAX1(AEV,EV(I))
      AE(J)=AEV
    9 ALPHA=ALPHA+0.5
      CALL LOADM(DUMY,A1,N,N)
```

```
      GO TO 17
    8 ALPHA=0.25
      DO 22 K=1,3
      T=ALPHA*T
      TIPI=T
      DO 33 KN=1,3
   33 CALL INTBSR
      CALL LOADM(RHAT,RINV,IR,IR)
      JALE=1
      CALL INVERS($15,RINV,DET,N,JALE)
      GO TO 28
   15 RETURN1
   28 CALL TRANS(EM,EMT,N,IR)
      CALL MULTIQ(EM,RINV,DUMY,N,IR,IR)
      CALL MULTIQ(DUMY,EMT,RM,N,IR,N)
      CALL SUBTRT(QHAT,RM,QMR,N,N)
      IF(K.NE.1) GO TO 19
      CALL LOADM(A1,DUMY,N,N)
   19 CALL LOADM(QMR,A1,N,N)
      JJJ=1
      CALL EIGEN
      DO 14 I=1,N
   14 EV(I)=CABS(EIG(I))
      DE=EV(1)
      DO 18 J=2,N
   18 DE=AMAX1(DE,EV(J))
      AE(K)=DE
   22 ALFA=ALFA+0.5
      CALL LOADM(DUMY,A1,N,N)
   17 WRITE(6,25)(AE(J),J=1,3)
   25 FORMAT(//22X,'NORMALIZED COST V',(22X,G12.5))
      ALFA=0.25
      T=T/1.25
      DO 30 I=1,3
      T=T*ALFA
      CTR(I)=AE(I)/T
   30 ALFA=ALFA+0.5
      IF(CTR(1).GT.CTR(2)) IM=2
      IF(CTR(IM).GT.CTR(3)) IM=3
      T=AE(IM)/CTR(IM)
      WRITE(6,39)T
   39 FORMAT(///22X,'OPTIMUM SAMPLING PERIOD:T =',G12.5)
      TIPI=T
      RETURN
      END
```