Modern Systems Analysis, Inc.
1235 WINES DRIVE
ANN ARBOR, MICHIGAN 48103

Telephone (313)
761-7295

9-14103-F
NASA CR-
140332

FINAL REPORT

for

CONTRACT NAS 9-14103

NEIGHBORING OPTIMAL GUIDANCE THEORY AND COMPUTER PROGRAM

by

William F. Powers

(NASA-CR-140332) NEIGHBORING OPTIMAL GUIDANCE THEORY AND COMPUTER PROGRAM Final Report (Modern Systems Analysis, Inc., Ann Arbor) 34 p HC $3.75 CSCL 09B

N75-12631

Unclas
G3/61 03611

October 1974

Modern Systems Analysis, Inc.
1235 WINES DRIVE
ANN ARBOR, MICHIGAN 48103

Telephone (313)
761-7295

9-14103-F

FINAL REPORT

for

CONTRACT NAS 9-14103

NEIGHBORING OPTIMAL GUIDANCE THEORY AND COMPUTER PROGRAM

by

William F. Powers

October 1974

TABLE OF CONTENTS

1. INTRODUCTION

This report is concerned with the description of a computer program for solving a certain class of optimal control problems known as the linear-quadratic problem (LQP).[1,5,6*] The LQP is sometimes referred to as the neighboring optimal guidance problem[1,3] even though it is applicable to both guidance and control problems. The distinguishing feature of the LQP is that it can be solved without iteration (whereas general optimization problems usually require iterative numerical techniques). Thus, it is useful in the initial portion of guidance and control design for determining approximate feedback gains and giving insight into the systems.

The computer program described in Appendix A solves the following problem:

$$\text{Minimize:}\quad J=\tfrac{1}{2}x_f^T S_f x_f+\tfrac{1}{2}\int_{t_o}^{t_f}[x^T A(t)x+2x^T N(t)u+u^T B(t)u]dt \tag{1.1}$$

$$\text{Subject To:}\quad \dot{x}=F(t)x+G(t)u,\ x(t_0)=x_0 \tag{1.2}$$

$$Mx_f=\Psi\quad , \tag{1.3}$$

where x is an n-vector, u is an m-vector, and Ψ is a $k\leq n$ -vector. The initial and final times t_0 and t_f are assumed to be known, and the matrix B(t) is assumed to be symmetric and positive definite on $[t_o,t_f]$. It is a necessary condition that B(t) be at least positive semi-definite; the case when B(t) is not positive definite is called the singular case

*Numbers indicate references listed in Section 5.

and its solution is presented in Ref. 7. If one is only interested in using the computer program, then one can proceed immediately to the self-contained Appendix A.

In Section 2, the solution for the problem defined by Eqs. (1.1)-(1.3) will be solved <u>without recourse to optimal control theory methods</u> (i.e., without using the calculus of variations or the Pontryagin maximum principle). To demonstrate the essential features of the justification, the quantities Ψ and N(t) will be assumed to be zero in Section 2. The case with Ψ and N(t) included will be justified in Section 3 and optimal control theory will be utilized. It should be noted that the usual application of Eqs. (1.1)-(1.3) is with Ψ and N(t) equal to zero, so Section 2 should be sufficient justification for most of the cases which arise in applications.

2. METHOD JUSTIFICATION: WITHOUT OPTIMAL CONTROL THEORY

In this section the optimal feedback control for the problem defined by Eqs. (1.1)-(1.3) with $M=0$ and $N(t)=0$ will be determined, i.e.,

Minimize: $$J=\tfrac{1}{2}x_f^T S_f x_f+\tfrac{1}{2}\int_{t_0}^{t_f}[x^T A(t)x+u^T B(t)u]dt \tag{2.1}$$

Subject To: $$\dot{x}=F(t)x+G(t)u,\ x(t_0)=x_0 \tag{2.2}$$

That is, we wish to determine the control

$$u=f(t,x) \tag{2.3}$$

i.e., a feedback control function, which causes J (the performance index) to be minimized. In Section A.2 of Appendix A the typical origin of the terms in Eqs. (2.1) and (2.2) is discussed. Also, as noted in Section 1, $B(t)$ is assumed to be symmetric and positive definite (the nonsingular case).

We could use the calculus of variations or the maximum principle to solve this problem, however knowledge of variational theory is required. Instead, we shall employ a "trick" which is employed in many types of optimization analysis. This trick involves the introduction of an arbitrary function with specified continuity and differentiability properties, which will be chosen later to help us out. (Such a trick is also used when one introduces Lagrange multipliers into an optimization problem, i.e., they are first treated as arbitrary functions and then particular functional forms are chosen to aid in the solution of the problem.)

Let $S(t)$ be an <u>arbitrary</u> differentiable, symmetric nxn matrix function.

<u>Property</u>: $\frac{1}{2}\int_{t_0}^{t_f}\frac{d}{dt}[x^TS(t)x]dt-\frac{1}{2}[x^TS(t)x]_{t_0}^{t_f} \equiv 0.$ (2.4)

<u>Proof</u>: The integral is an exact differential in x^TSx, so:

$$\frac{1}{2}\int_{t_0}^{t_f}d[x^TSx]-\frac{1}{2}[x^TSx]_{t_0}^{t_f} = \frac{1}{2}[x^TSx]_{t_0}^{t_f} - \frac{1}{2}[x^TSx]_{t_0}^{t_f} \equiv 0.$$

If Eq. (2.4) is added to Eq. (2.1), the problem is not changed because Eq. (2.4) is identically zero. Thus, performing this addition:

$$J=\frac{1}{2}x_f^TS_fx_f - \frac{1}{2}[x^TS(t)x]_{t_0}^{t_f} + \frac{1}{2}\int_{t_0}^{t_f}[x^TAx+u^TBu+\frac{d}{dt}(x^TSx)]dt.$$

Note that S_f is a <u>given</u> nxn matrix, whereas $S(t_f)$ is (as of now) just an arbitrary nxn matrix function. Combining terms outside the integral and differentiating under the integral results in:

$$J=\frac{1}{2}x_f^T(S_f-S(t_f))x_f+\frac{1}{2}x_0^TS(t_0)x_0$$

$$+\frac{1}{2}\int_{t_0}^{t_f}[x^TAx+u^TBu+\dot{x}^TSx+x^T\dot{S}x+x^TS\dot{x}]dt$$

But, $\dot{x}=Fx+Gu$ (note this is the point where the constraints get into the problem), so:

$$J=\frac{1}{2}x_f^T[S_f-S(t_f)]x_f+\frac{1}{2}x_0^TS(t_0)x_0+\frac{1}{2}\int_{t_0}^{t_f}[x^TAx+u^TBu+x^TF^TSx+u^TG^TSx$$

$$+x^T\dot{S}x+x^TSFx+x^TSGu]dt$$

or,

$$J=\tfrac{1}{2}x_f^T[S_f-S(t_f)]x_f+\tfrac{1}{2}x_o^TS(t_o)x_o+\tfrac{1}{2}\int_{t_o}^{t_f}[x^T(A+F^TS+\dot{S}+SF)x+x^TSGu$$

$$+u^TG^TSx+u^TBu]dt \tag{2.5}$$

Note that the integrand is a quadratic form in x and u, and that (as of now) S is an arbitrary matrix. We shall now choose S(t) in such a way that the optimal control is obvious.

If we can write the integral in J as:

$$\int_{t_o}^{t_f}(Kx+Lu)^T(Kx+Lu)dt \tag{2.6}$$

with L^{-1} existing, then $u=-L^{-1}Kx$ is the optimal control because the integrand is the square of (Kx+Lu), which implies zero is the smallest value the integrand can take and $u=-L^{-1}Kx$ causes the integrand to equal zero.

Let us now expand Eq. (2.6) and equate it to the integrand of Eq. (2.5); this will then imply how we should choose S(t) to get the obvious control solution form of Eq. (2.6):

$$\int_{t_o}^{t_f}(Kx+Lu)^T(Kx+Lu)dt=\int_{t_o}^{t_f}[x^TK^TKx+x^TK^TLu+u^TL^TKx+u^TL^TLu]dt \tag{2.7}$$

Equating terms with the integrand of Eq. (2.5) implies

$$K^TK=A+F^TS+\dot{S}+SF \tag{2.8}$$

$$K^TL=SG \tag{2.9}$$

$$L^TK=G^TS \tag{2.10}$$

$$L^TL=B \tag{2.11}$$

Since B is symmetric and invertible, L is also symmetric and invertible, i.e.,

$$LL=B \Rightarrow L^2=B \Rightarrow L \equiv B^{\frac{1}{2}}.$$

By Eqs. (2.9) and (2.10)

$$L^TK=G^TS \Rightarrow K=L^{-1}G^TS=B^{-\frac{1}{2}}G^TS. \tag{2.12}$$

Finally, by Eqs. (2.8) and (2.12):

$$(B^{-\frac{1}{2}}G^TS)^T(B^{-\frac{1}{2}}G^TS)=A+F^TS+\dot{S}+SF$$

or

$$\dot{S}+SF+F^TS+A=SGB^{-\frac{1}{2}}B^{-\frac{1}{2}}G^TS \ ,$$

where the symmetry of $B^{-\frac{1}{2}}$ and S has been used. Then,

$$\dot{S}=-SF-F^TS-A+SGB^{-1}G^TS \tag{2.13}$$

Since x_f depends implicitly upon u, we can remove the implicit u-term from J by choosing:

$$S(t_f)=S_f \tag{2.14}$$

To summarize, then, if one defines the Riccati equation:

$$\dot{S}=-SF-F^TS-A+SGB^{-1}G^TS$$

with boundary condition:

$$S(t_f)=S_f \ ,$$

the quantity J may be written as:

$$J=\tfrac{1}{2}x_o^TS(t_o)x_o+\int_{t_o}^{t_f}(B^{-\frac{1}{2}}G^TSx+B^{-\frac{1}{2}}u)^T(B^{-\frac{1}{2}}G^TSx+B^{-\frac{1}{2}}u)dt \ . \tag{2.15}$$

Then, the term outside of the integral is independent of u (since x_o is specified and $S(t_o)$ is well-defined by the solution of Eqs. (2.13) and (2.14)) and the smallest possible value of the integral is zero, and the integral is zero if:

$$B^{-\frac{1}{2}}G^TSx+B^{-\frac{1}{2}}u=0$$

$$\therefore \quad u=-B^{-1}G^TSx \ . \tag{2.16}$$

Equation (2.16) defines the optimal feedback control, and $J=\frac{1}{2}x_o^T S(t_o)x_o$ is the value of the performance index due to the optimal control.

Example: Min.: $J=\frac{1}{2}\int_0^T (ax^2+bu^2)dt$ (2.17)

Sub. to: $\dot{x}=u$, $x(o)=x_o$, T specified (2.18)

Note that the system is a linear, time-invariant system, and if classical linear control were employed here, a linear feedback control with constant gains would be the typical result. The usual result with LQP theory is a linear feedback control with time-varying gains. However, by letting $t_f=T$ be large, the gains are approximately constant. (Constant gains may be obtained by choosing $T=\infty$; see Ref. 5.)

Let us now compute the solution of the problem defined by Eqs. (2.17) and (2.18). As noted above, if T is finite, then time varying gains are obtained. However, we should expect the gain to approach a constant as T becomes large. The solution to this problem is defined by Eqs. (2.13), (2.14), and (2.16):

$$u=-B^{-1}G^T Sx \tag{2.19}$$

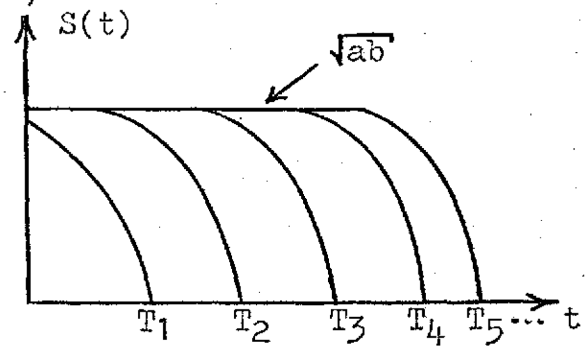
with: $B^{-1}=\frac{1}{b}$, $G=1$, and S is the solution of the Riccati equation:

$$\dot{S}=-a+\frac{S^2}{b} \quad .$$

The solution of the Riccati equation is:

$$S(t)= \sqrt{ab}\left[\frac{1-e^{2\sqrt{a/b}\,(t-T)}}{1+e^{2\sqrt{a/b}\,(t-T)}}\right] \tag{2.20}$$

Note that for $T \gg t$, $S(t) \approx \sqrt{ab}$. Plots of various S(t) as T varies are shown to the right; S(t) is basically constant except for a transient near $t_f = T$. This behavior is also typical of more complicated time-invariant systems.



Thus, for this problem, $S(t) \approx \sqrt{ab}$ if $T \gg t$ and the approximate optimal linear feedback control is (from Eq. (2.19))

$$u=-(\tfrac{1}{b})(1)\sqrt{ab}\,x=-\sqrt{a/b}\,x \quad . \tag{2.21}$$

This result agrees with intuition in that: (i) u is negative if x is positive, which implies that the control attempts to drive the state x to zero; (ii) u is proportional to a/b . The latter result implies that if $a \gg b$, then there is more weighting on the state in the performance index and the result is a large control value to maintain a small value of x. On the other hand, if $a \ll b$, then there is more weighting on the control in the performance index and the result is a small control value (relative to the value of x).

3. METHOD JUSTIFICATION: WITH OPTIMAL CONTROL THEORY

In this section the optimal control problem defined by Eqs. (1.1)-(1.3) will be solved in general. The solution technique is well known, and is similar to the developments in References 1 and 6. The underlying optimization theory is discussed in Refs. 1, 5, 6.

To develop the desired solution it is convenient to adjoin the terminal conditions, Eq. (1.3), to the performance index, Eq. (1.1), with the constant Lagrange multiplier vector q, and the differential equations, Eq. (1.2), with the time-varying (in general) Lagrange multiplier vector p. Then, the augmented performance index is:

$$J=\tfrac{1}{2}x_f^T S_f x_f+q^T(Mx_f-\Psi)+\tfrac{1}{2}\int_{t_o}^{t_f}[x^T Ax+2x^T Nu+u^T Bu$$

$$+p^T(Fx+Gu-\dot{x})]dt. \tag{3.1}$$

The Hamiltonian for this problem is

$$H=\tfrac{1}{2}(x^T Ax+2x^T Nu+u^T Bu)+p^T(Fx+Gu) \quad , \tag{3.2}$$

and the resultant necessary conditions of optimality are:

$$\dot{p}=-H_x \tag{3.3}$$

$$H_u=0 \tag{3.4}$$

$$p_f=S_f x_f+M^T q \quad , \tag{3.5}$$

where H_x is considered as an (nx1) vector and H_u is an (mx1) vector. Since the problem is nonsingular (i.e., $B(t)$ is positive definite), the control, u, may be eliminated from the problem by Eq. (3.4), i.e.,

$$H_u = N^T x + Bu + G^T p = 0 \tag{3.6}$$

which implies

$$u = -B^{-1}(N^T x + G^T p). \tag{3.7}$$

Then, upon substitution into Eqs. (1.2) and (3.3), we have

$$\dot{x} = Fx - GB^{-1}(N^T x + G^T p)$$

$$\dot{p} = -Ax + NB^{-1}(N^T x + G^T p) - F^T p$$

or,

$$\dot{x} = (F - GB^{-1}N^T)x - GB^{-1}G^T p \tag{3.8}$$

$$\dot{p} = (NB^{-1}N^T - A)x + (NB^{-1}G^T - F^T)p \tag{3.9}$$

Since Eqs. (3.8), (3.9), (1.3), and (3.5) are linear in x, p, Ψ, and q, it can be shown that there must exist linear relationships among the variables, and we introduce the unknown (for now) matrices Q, R, S, and V involved in these relationships

$$p(t) = S(t)x(t) + R(t)q \tag{3.10}$$

$$\Psi = V(t)x(t) + Q(t)q. \tag{3.11}$$

(It can be shown that the resultant $S(t)$ is symmetric,[1,5] and we shall assume this now to ease the notation.) Thus, among the $2n+2k$ -variables x, p, q, and Ψ there exist $n+k$ independent variables, which by Eqs. (3.10) and (3.11) have been chosen to be x and q. Upon substitution of these relations into Eqs. (1.3) and (3.5) (i.e., the terminal boundary and transversality conditions), we obtain

$$Mx_f=V(t_f)x_f+Q(t_f)q$$

$$S(t_f)x_f+R(t_f)q=S_f x_f+M^T q$$

or

$$(M-V(t_f))x_f+Q(t_f)q=0 \tag{3.12}$$

$$(S(t_f)-S_f)x_f+(R(t_f)-M^T)q=0 \ , \tag{3.13}$$

which are identities in x_f and q. This implies that the coefficients of Eqs. (3.12) and (3.13) must vanish, and thus

$$V(t_f)=M \ , \quad Q(t_f)=0 \ , \quad S(t_f)=S_f \ , \quad R(t_f)=M^T \ . \tag{3.14}$$

Equations (3.14) define boundary conditions for the unknown matrices. If differential equations could be developed for the matrices, then the matrices could be computed by integrating the resultant equations backward with the boundary conditions of Eq. (3.14). We shall now determine such a set of differential equations by differentiating Eqs. (3.10) and (3.11), substituting the results into Eqs. (3.8) and (3.9), and then interpreting the resultant forms.

Before we make these computations, it is instructive to answer the question of why Eqs. (3.10) and (3.11) were introduced in the first place. Our goal is a feedback control, say

$$u=g(t,x,\Psi) \ . \tag{3.15}$$

We are guaranteed the existence of relationships of the type assumed in Eqs. (3.10) and (3.11) by properties of linear differential equations (i.e., Eqs. (3.8), (3.9)) with linear boundary conditions (i.e., Eqs. (1.3), (3.5)). Such relations are desirable because if S(t), R(t), V(t), and Q(t) can be determined, and if Q(t) is invertable, then the optimal feedback control can be computed from Eq. (3.7), i.e.,

$$u=-B^{-1}[N^Tx+G^TSx+G^TR(Q^{-1}\Psi-Q^{-1}Vx)] \tag{3.16}$$

We shall come back to this equation after determining the defining differential equations for S, R, V, and Q.

First differentiate Eq. (3.10) and substitute the result into Eq. (3.9), i.e.,

$$\dot{S}x+S\dot{x}+\dot{R}q=(NB^{-1}N^T-A)x+(NB^{-1}G^T-F^T)(Sx+Rq) \quad . \tag{3.17}$$

Then upon substitution for $\dot{x}$ (from Eq. (3.8)) and upon rearrangement, we obtain

$$[\dot{S}+SF+F^TS+A-(SG+N)B^{-1}(SG+N)^T]x$$

$$+[\dot{R}+(F^T-(SG+N)B^{-1}G^T)R]q=0 \tag{3.18}$$

Since x and q are the independent variables, Eq. (3.18) is an identity in x and q, and thus, the coefficients must vanish, which implies:

$$\dot{S}=-SF-F^TS-A+(SG+N)B^{-1}(SG+N)^T \quad , \quad S(t_f)=S_f \tag{3.19}$$

$$\dot{R}=[-F^T+(SG+N)B^{-1}G^T]R \quad , \quad R(t_f)=M^T \quad , \tag{3.20}$$

where the boundary conditions at t_f are obtained from Eq. (3.14).

The equations for Q and V are obtained by differentiating Eq. (3.11) (noting that Ψ is a constant) and substituting for $\dot{x}$, which gives

$$[\dot{V}+V(F-GB^{-1}(N^T+G^TS))]x+[\dot{Q}-VGB^{-1}G^TR]q=0 \tag{3.21}$$

This is also an identity in x and q which implies that the coefficients must vanish. Since the equation for $\dot{V}$ is the transpose of $\dot{R}$, $V(t_f)=M=R(t_f)^T$, and S is symmetric, it follows that:

$$V(t)=R(t)^T \quad , \tag{3.22}$$

so the variable V(t) is eliminated. The equation for Q, with boundary condition from Eq. (3.14), is

$$\dot{Q}=R^TGB^{-1}G^TR \ , \ Q(t_f)=0 \ . \tag{3.23}$$

The resultant optimal control is then:

$$u=-B^{-1}[N^T+G^T(S-RQ^{-1}R^T)]x-B^{-1}G^TRQ^{-1}\Psi \ , \tag{3.24}$$

where S, R, and Q can be determined by Eqs. (3.19), (3.20), and (3.23).

Note that Q occurs in the optimal control, Eq. (3.24), only in the product RQ^{-1}. This motivates one to develop differential equations for $(S-RQ^{-1}R^T)$ and RQ^{-1} (as opposed to S, R, and Q), and it can be shown[1,6] that the resultant differential equations are exactly like the $\dot{S}$ and $\dot{R}$ equations. However, one cannot use these until $t_f-\varepsilon$ ($\varepsilon > 0$) because Q^{-1} does not exist at t_f (since $Q(t_f)=0$). Thus, in LQP, the S, R, and Q equations are integrated backward for a small time increment, and then a switch-over to the direct computation of $S-RQ^{-1}R^T$ and RQ^{-1} is made. This, of course, saves computer time.

Finally, it should be noted that $S-RQ^{-1}R^T$ can become unbounded. This means that the proposed problem does not possess an optimal solution (or a unique optimal in very special cases), and the time at which $S-RQ^{-1}R^T$ becomes unbounded is called a <u>conjugate point</u>. The program, LQP, prints out the occurrence of a conjugate point and stops the computation. This is another reason for choosing $S_f>0$, $A(t)\geq 0$, $N(t)=0$, and no terminal conditions because then one is guaranteed that a unique optimal control exists[5] and no conjugate point can occur.

4. SUMMARY AND CONCLUSIONS

This report contains developments of the linear quadratic optimal control problem, one of which does not involve optimal control theory. The theory is applicable to the development of neighboring optimal feedback guidance gains, and is useful as a tool for synthesizing feedback control laws in general. A computer program which requires only the pertinent matrices of the linear quadratic problem is described in Appendix A, which also serves as a self-contained User's Guide.

Knowledge of optimal control theory is not necessary to use the computer program or to understand the development of the expression for the optimal feedback control (see Section 2). Thus, Section 2 and Appendix A may be learned in a relatively short period of time without any background in optimization theory.

The relationships between classical feedback control design and linear quadratic optimal control design were presented in a number of lectures to NASA-JSC and contractor personnel in July-August 1974 by W. F. Powers. Lecture Notes were handed out at the lectures and are available upon request from Modern Systems Analysis, Inc.

5. REFERENCES

1. Bryson, A. E., Jr., and Ho, Y. C., Applied Optimal Control, Ginn-Blaisdell, Waltham, Massachusetts, 1969, Chapters 5 and 6.

2. Powers, W. F., "Numerical Integration Routines for Near-Earth Operations," Final Report for NASA Contract NAS 9-12319 (Mod. No. 1S), September 1973.

3. Powers, W. F., "Techniques for Improved Convergence in Neighboring Optimal Guidance," AIAA Journal, Vol. 8, No. 12, 1970, pp. 2235-2241.

4. Levison, W. H., and Baron, S., "Manual Control Analysis of Vertical Situation Displays for STOL Aircraft," presented at the Eighth Annual Conference on Manual Control, Ann Arbor, Michigan, May 1972.

5. Athans, M., and Falb, P., Optimal Control, McGraw-Hill, New York, 1966, Chapter 9.

6. Citron, S. J., Elements of Optimal Control, Holt, Rinehart, and Winston, New York, 1969.

7. McDanell, J. P., and Powers, W. F., "New Jacobi-Type Necessary and Sufficient Conditions for Singular Optimization Problems," AIAA Journal, Vol. 8, No. 8, 1970, pp. 1416-1420.

8. IBM System/360, Scientific Subroutine Package, (360A-CM-03X) Version III, IBM Technical Publication Department, 1968.

APPENDIX A

USER'S GUIDE FOR LQP

LQP is a subroutine which solves the following optimization problem, which does not require iteration:

Minimize: $$J=\tfrac{1}{2}x_f^T S_f x_f+\tfrac{1}{2}\int_{t_o}^{t_f}[x^T A(t)x+2x^T N(t)u+u^T B(t)u]dt \qquad (A.1)$$

Subject To: $$\dot{x}=F(t)x+G(t)u \quad , \quad x(t_o)=x_o \qquad (A.2)$$

$$Mx_f=\Psi \qquad (A.3)$$

where x=n-vector, u=m-vector, Ψ =p-vector, and t_o and t_f are specified. The notation of Eqs. (A.1)-(A.3) is that of Ref. 1. The user need only supply a "MAIN" subroutine which defines the parameters of the problem and calls LQP. If any of the matrices A, N, B, F, or G are time-varying, then a second subroutine TIMVAR, which defines the time-varying matrices, must be supplied, also. Since LQP employs the numerical integration scheme DVDQ (which is a variable-stepsize, variable-order integrator; see Ref. 2), it is recommended that the time-varying matrices in TIMVAR be represented by cubic splines.

A.1 Basic Flow Of The Algorithm

As shown in Section 3, the solution of the optimal control problem defined above is:

$$u=-B^{-1}[N^T+G^T(S-RQ^{-1}R^T)]x-B^{-1}G^T RQ^{-1}\Psi \qquad (A.4)$$

where

$$\dot{S}=-SF-F^TS-A+(SG+N)B^{-1}(SG+N)^T\ ,\quad S(t_f)=S_f \tag{A.5}$$

$$\dot{R}=[-F^T+(SG+N)B^{-1}G^T]R\ ,\quad R(t_f)=M^T \tag{A.6}$$

$$\dot{Q}=R^TGB^{-1}G^TR\ .\quad ,\quad Q(t_f)=0 \tag{A.7}$$

If, as in most applications, Eq. (A.3) is not present, then the solution is defined by:

$$u=-B^{-1}[N^T+G^TS]x\ , \tag{A.8}$$

where S is still defined by Eq. (A.5). Since the latter problem requires much less integration and logical operations, it is advantageous to model the control problem without an Eq. (A.3) (if possible) and a flag exists in the program for this purpose (IFLAG1).

The flow of the computations is as shown in Figure A.1, i.e., the values for $S(t_f)$, $R(t_f)$, and $Q(t_f)$ are defined, numerical

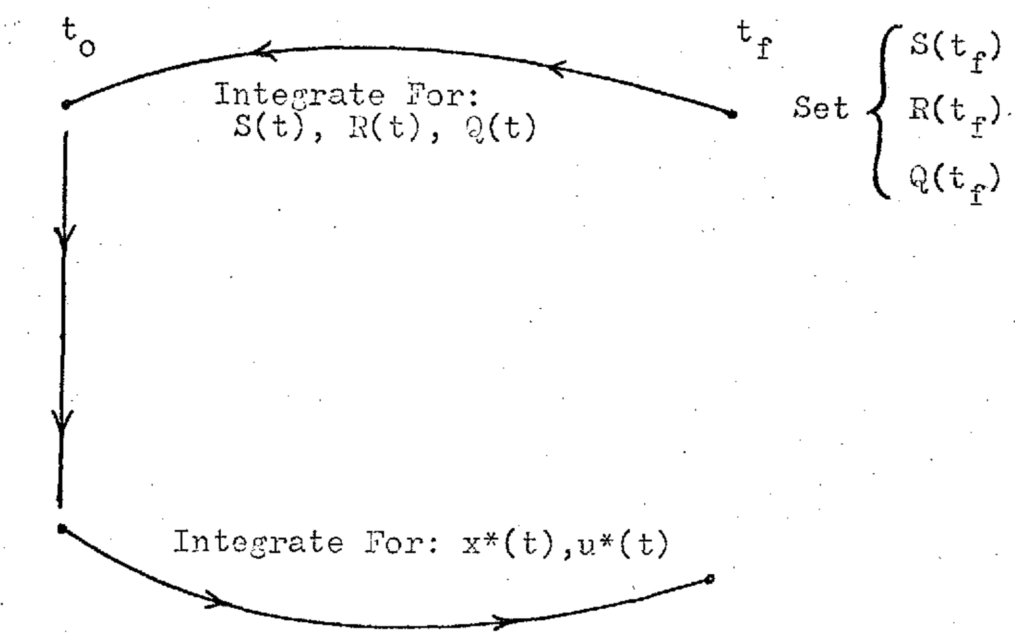


Figure A.1. Flow Of The Computations.

integration proceeds backward to t_o, and then the optimal state and control are defined by a forward integration.

A.2 Selection Of Weighting Matrices

In the optimization problem of Eqs. (A.1)-(A.3), the matrices F(t), G(t), M, and Ψ are defined by the process, e.g., F and G typically result from linearization about a nominal trajectory. If pure neighboring optimal guidance is to be used, then the matrices in Eq. (1) are also well-defined (e.g., see Refs. 1, 3). However, most applications will probably require the specification of the weighting matrices S_f, A(t), N(t), and B(t) by the guidance or control designer. In this section a "rule-of-thumb" for weighting matrix selection which has proved useful in a number of applications (Ref. 1, 4) will be presented.

To get started on a design, assume N(t)=0, i.e., no mixed state-control terms in Eq. (A.1). In most cases one will not have to employ a nonzero N-matrix at any time in the design. The only remaining matrices are S_f, A, and B which weight terminal state values, state trajectory values, and control values, respectively. If Eq. (A.2) results from linearization about a nominal trajectory (the usual case), then x and u actually represent deviations from the nominal. In such a case, one usually has some idea of the tolerable deviations for each variable. Thus, assume it is desired that:

$$|x_{i_f}| \le |\overline{x}_{i_f}| \quad (i=1,\ldots,n) \tag{A.9}$$

$$|x_i(t)| \le |\overline{x}_i(t)| \quad (i=1,\ldots,n) \tag{A.10}$$

$$|u_i(t)| \le |\overline{u}_i(t)| \quad (i=1,\ldots,m)\ , \tag{A.11}$$

i.e., the maximum deviation from the nominal value associated with x_1 is $\pm\,\overline{x}_{1f}$, and so on. Then, the smaller the value of $\overline{()}$ (e.g., $\overline{x}_{1f}$), the larger the weighting of $()$ (e.g., x_{1f}) should be in Eq. (A.1), and vice versa. A choice which satisfies this criterion is:

$$S_f = \begin{bmatrix} \frac{1}{\overline{x}_{1f}^2} & & & \bigcirc \\ & \frac{1}{\overline{x}_{2f}^2} & & \\ & & \ddots & \\ \bigcirc & & & \frac{1}{\overline{x}_{nf}^2} \end{bmatrix} \tag{A.12}$$

$$A(t) = \begin{bmatrix} \frac{1}{\overline{x}_1(t)^2} & & & \bigcirc \\ & \frac{1}{\overline{x}_2(t)^2} & & \\ & & \ddots & \\ \bigcirc & & & \frac{1}{\overline{x}_n(t)^2} \end{bmatrix} \tag{A.13}$$

$$B(t) = \begin{bmatrix} \frac{1}{\overline{u}_1(t)^2} & & & \bigcirc \\ & \frac{1}{\overline{u}_2(t)^2} & & \\ & & \ddots & \\ \bigcirc & & & \frac{1}{\overline{u}_m(t)^2} \end{bmatrix} \tag{A.14}$$

Typically the $\bar{x}_i(t)$ and $\bar{u}_i(t)$ values are constants; however in Shuttle reentry one may wish to change the weighting matrices from one flight phase to another (e.g., from the constant drag phase to the equilibrium glide phase and so on). One can then compute the resultant optimal control with LQP, and check to see if the resultant feedback control meets all specifications. If not, the weighting matrices should be modified, and, of course, the modifications are problem dependent. In any case, Eqs. (A.12)-(A.14) give, at least, a well-defined start to the feedback gain design process.

Finally, to save computer time, B^{-1} will be supplied to the program instead of B. Usually B^{-1} is easily calculated beforehand (if not, the computation can be done in TIMVAR).

A.3 LQP Argument List

In this section the variables employed in LQP will be listed along with their type (integer or double precision), dimension, and identification with the variables in Eqs. (A.1)-(A.3). A "MAIN" subroutine (to be discussed in the next section) is to be supplied by the user, and a call to LQP is made from MAIN. The CALL-statement is:

```
CALL  LQP(N,M,IP,IFLAG1,IFLAG2,TI,TF,EP,SF,A,DN,BINV,F,G,
      DM,PSI,X,SRQX,DSRQX,S,R,Q,U,XDOT,DT,KQ,YN,DUM1,
      DUM2,DUM3,DUM4,DUM5,DUM6,DUM7,DUM8,DUM9,DUM10,DUM11,
      XL,XLDOT,DUM12)
```

The variables in this call are defined as follows, where I=integer, D=double precision, E=single precision, and

$$K=n(n+1)/2 + np+p(p+1)/2+n \quad .$$

Program Variable	Problem Variable	Variable Type	Dimension
N	n	I	scalar
M	m	I	scalar
IP	p	I	scalar
IFLAG1	--	I	scalar
IFLAG2	--	I	scalar
TI	t_o	D	scalar
TF	t_f	D	scalar
EP	--	E	scalar
SF	S_f	D	N*(N+1)/2
A	A	D	N*(N+1)/2
DN	N	D	N*M
BINV	B^{-1}	D	M*(M+1)/2
F	F	D	N*N
G	G	D	N*M
DM	M	D	N*IP(If p=0,DIM=1)
PSI	Ψ	D	IP(If p=0,DIM=1)
X	x	D	N
SRQX	--	D	K
DSRQX	--	D	K
S	S	D	N*(N+1)/2
R	R	D	N*IP(If p=0,DIM=1)
Q	Q	D	IP*(IP+1)/2(If p=0, DIM=1)
U	u	D	M
XDOT	$\dot{x}$	D	N
DT	--	D	(17,K)

Program Variable	Problem Variable	Variable Type	Dimension
KQ	--	I	K
YN	--	D	K
DUM1	--	D	N*N
DUM2	--	D	N*N
DUM3	--	D	N*N
DUM4	--	D	N*M
DUM5	--	D	N*M
DUM6	--	D	N*M(If p=0,DIM=1)
DUM7	--	D	N*IP(If p=0,DIM=1)
DUM8	--	D	N*IP(If p=0,DIM=1)
DUM9	--	D	N*IP(If p=0,DIM=1)
DUM10	--	D	M
DUM11	--	D	M*(M+1)/2
XL	p(see Eq.3.9)	D	N
XLDOT	$\dot{p}$(see Eq.3.9)	D	N
DUM12	--	D	N

The variables above which are not problem variables are described below. Except for IFLAG1, IFLAG2, and EP, these variables are LQP and DVDQ "working variables" which are of no concern to the user except for DIMENSION statements (and the dimensions are well-defined in the list above).

IFLAG1: flag set by user indicating presence or absence of terminal conditions; =0 if terminal conditions present, and =1 if no terminal conditions.

IFLAG2: flag set by user indicating time-variability of matrices; =0 if A, B, DN, F, G are time-invariant, and =1 if at least one of these matrices is time-varying (in which case the user must supply a subroutine TIMVAR).

EP: absolute local error indicator for the numerical integration scheme DVDQ; this parameter is problem dependent, but a safe initial choice is 1.E-5. (See Ref. 2 for a more thorough description of EP.)

SRQX: contains the vector being integrated by DVDQ. If terminal conditions present, SRQX contains S, R, Q (expressed in vector form) going backward and S, R, Q, x going forward. If no terminal conditions, SRQX contains S going backward and x, p going forward.

DSRQX: contains the time derivative of SRQX.

DT: storage required for DVDQ.

KQ: storage required for DVDQ.

YN: storage required for DVDQ.

DUM1 through DUM12: dummy storage required for LQP matrix manipulations.

A.4 Matrix To Vector Conversions

Even though Eqs. (A.1)-(A.3) are written in matrix form, the computer program operates in a vector mode. (The only matrix dimension is for DT, which is part of the integrator, DVDQ). Thus, all matrices must be converted to vectors, and since some of the matrices are symmetric, considerable savings can be gained by distinguishing between general and symmetric matrices.

A number of subroutines from Ref. 8 are employed in LQP to perform the various matrix manipulations. These subroutines assume that the matrices have been converted to vectors column-by-column. That is, consider the three-by-three matrix A:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \qquad (A.15)$$

If A is a general matrix, then it will be converted to a 9-vector column-by-column, i.e.,

$$A(9) = [a_{11}\ a_{21}\ a_{31}\ a_{12}\ a_{22}\ a_{32}\ a_{13}\ a_{23}\ a_{33}]^T \ . \qquad (A.16)$$

(General Matrix Format)

If A is a symmetric matrix, then it will be converted to a 6-vector column-by-column of the <u>upper</u> triangular portion of the matrix, i.e.,

$$A(6) = [a_{11}\ a_{12}\ a_{22}\ a_{13}\ a_{23}\ a_{33}]^T \ . \qquad (A.17)$$

(Symmetric Matrix Format)

The various matrices are printed out in the same manner (i.e., as n^2 or $n(n+1)/2$ vectors in the format of Eqs. (A.16) or (A.17), respectively).

A.5 Example Problems

A number of simple examples will be presented in this section to illustrate the setup of MAIN and TIMVAR, typical printout, and the output for a problem with a conjugate point.

Example A.1: Let x be a 3-vector and u a 2-vector.

Minimize: $$J=\frac{1}{2}\int_0^1 (2x_3^2+u_1^2+u_2^2)dt \tag{A.18}$$

Subject To:

$$\dot{x}=\begin{bmatrix}0&1&0\\0&0&0\\0&0&1\end{bmatrix}x+\begin{bmatrix}0&0\\1&0\\0&1\end{bmatrix}u \tag{A.19}$$

$$x_1(0)=1 \ , \quad x_2(0)=2 \ , \quad x_3(0)=0 \tag{A.20}$$

$$\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}x(1)=\begin{bmatrix}0\\0\\1\end{bmatrix} \tag{A.21}$$

Equation (A.18) corresponds to Eq. (A.1); Eqs. (A.19) and (A.20) to Eq. (A.2); and Eq. (A.21) to Eq. (A.3). Since terminal conditions are present, IFLAG1=0; and since all of the matrices are time-invariant, IFLAG2=0. A value of EP=1.0×10^{-5} will be used for the absolute local error control in the integrator. A typical MAIN subroutine (the only information required by the user) is shown in Figure A.2. The development of MAIN involves the development of a well-defined DIMENSION-statement, data input, and a well-defined CALL to LQP. (Also, note the REAL EP statement because EP must be single precision to avoid difficulties on UNIVAC computers.)

The program begins the backward integration of S, Q, and R at t=1, and the printout is shown in Figure A.3. Since only $\tilde{S}=S-RQ^{-1}R^T$ and $\tilde{W}=RQ^{-1}$ are needed to define the optimal feedback gains, a switch to the $\tilde{S}$ and $\tilde{W}$ system is made at t=0.95. (Such a switch is possible only if $Q^{-1}(0.95)$ exists; if $Q^{-1}(0.95)$ does not exist, then the program will stop, usually indicating an abnormal problem.) From t=0.95 to t=0, the $\tilde{S}$ and $\tilde{W}$ matrices are printed out in the format of Eqs. (A.17) and (A.16), respectively, since $\tilde{S}$ is symmetric and $\tilde{W}$ is a general matrix. The first few forward integrations from t=0 are shown in Figure A.4, and the values of the resultant optimal state and control are added to the printout. One can compare the backward and forward values of S and W to aid in the choice of a value for EP which gives the desired accuracy.

<u>Example A.2:</u> Let x and u be scalars.

Minimize: $$J=\frac{1}{2}\int_0^{\frac{3\pi}{2}} u^2 dt$$

Subject To: $$\dot{x}=x+u$$

$$x(0)=0\ , \quad x(3\pi/2)=1.0$$

This optimal control problem possesses a conjugate point at $t=\pi/2$, which implies that there does not exist an optimal feedback control on the interval $[0, 3\pi/2]$. The program detects the possibility of a conjugate point when the numerical integration scheme begins to decrease the stepsize to a very small value (which is necessary to get accurate values of $\tilde{S}$ and $\tilde{W}$ since $|\tilde{S}| \longrightarrow \infty$). The output of the program denoting that this behavior is occurring at $t=\pi/2$ is shown in Figure A.5.

<u>Example A.3:</u> Let x and u be scalars.

Minimize: $$J=2x_f^2+\frac{1}{2}\int_0^1 u^2dt$$

Subject To: $$\dot{x}=t^2x+tu \quad , \quad x(0)=1$$

This optimal control problem possesses time variable dynamics, and, thus, the subroutine TIMVAR is required. A typical TIMVAR subroutine for this problem is shown in Figure A.6.

```
      IMPLICIT REAL*8(A-H,O-Z)
      REAL EP
      DIMENSION SF(6),A(6),DN(6),BINV(3),F(9),G(3),DM(9),PSI(3),X(3),
     * SRQX(24),DSRQX(24),S(6),R(9),Q(6),U(2),XDOT(3),DT(17,24),KQ(24),
     * YN(24),DUM1(9),DUM2(9),DUM3(9),DUM4(6),DUM5(6),DUM6(6),DUM7(9),
     * DUM8(9),DUM9(9),DUM10(2),DUM11(3),XL(3),XLDOT(3),DUM12(3)
C
      N=3
      M=2
      IP=3
      IFLAG1=0
      IFLAG2=0
      TI=0.D0
      TF=1.D0
      EP=1.E-5
      DO 1 I=1,6
    1 SF(I)=0.D0
      DO 2 I=1,5
    2 A(I)=0.D0
      A(6)=2.D0
      DO 3 I=1,6
    3 DN(I)=0.D0
      BINV(1)=1.D0
      BINV(2)=0.D0
      BINV(3)=1.D0
      DO 4 I=1,9
    4 F(I)=0.D0
      F(4)=1.D0
      F(9)=1.D0
      DO 5 I=1,6
    5 G(I)=0.D0
      G(2)=1.D0
      G(6)=1.D0
      DO 6 I=1,9
    6 DM(I)=0.D0
      DM(1)=1.D0
      DM(5)=1.D0
      DM(9)=1.D0
      PSI(1)=0.D0
      PSI(2)=0.D0
      PSI(3)=1.D0
      X(1)=1.D0
      X(2)=2.D0
      X(3)=0.D0
      CALL LQP(N,M,IP,IFLAG1,IFLAG2,TI,TF,EP,SF,A,DN,BINV,F,G,DM,
     * PSI,X,SRQX,DSRQX,S,R,Q,U,XDOT,DT,KQ,YN,DUM1,DUM2,DUM3,DUM4,DUM5,
     * DUM6,DUM7,DUM8,DUM9,DUM10,DUM11,XL,XLDOT,DUM12)
      END
```


Figure A.2 A Typical MAIN Subroutine For Example A.1.

```
                    BACKWARD INTEGRATION BEGINS AT T=    1.000

    T=   1.000
    ---------

    S MATRIX:          .000000          .000000          .000000   ...

    R MATRIX:         1.000000          .000000          .000000   ...

    R MATRIX:          .000000         1.000000

    Q MATRIX:          .000000          .000000          .000000   ...

    T=    .950
    ---------

    S MATRIX:          .000000          .000000          .000000   ...

    R MATRIX:         1.000000          .050000          .000000   ...

    R MATRIX:          .000000         1.048559

    Q MATRIX:        -.000042         -.001250         -.050000   ...


                    SUCCESSFUL SWITCH TO TILDE SYSTEM AT T=     .950

    T=    .950
    ---------

    S(TILDE):    96417.826274      2410.445657        80.261141   ...

    W(TILDE):   -96417.826274     -2410.445657          .000000   ...

    W(TILDE):         .000000       -19.975022

    T=    .902
    ---------

    S(TILDE):    12954.518904       631.532797        41.043634   ...

    W(TILDE):   -12954.518904      -631.532797          .000000   ...

    W(TILDE):         .000000       -10.207822
```

Figure A.3   Backward Integration Printout For Example A.1, With Switch At t=0.95.

FORWARD INTEGRATION STARTS

T= .000

S(TILDE):	12.000006	6.000003	4.000002 ...
W(TILDE):	-12.000006	-6.000003	.000000 ...
W(TILDE):	.000000	-.632677	
STATE:	1.000000	2.000000	.000000 ...
CONTROL:	-14.000007	.632677	

T= .047

S(TILDE):	13.886301	6.613351	4.199477 ...
W(TILDE):	-13.886301	-6.613351	.000000 ...
W(TILDE):	.000000	-.690923	
STATE:	1.079635	1.362075	.030086 ...
CONTROL:	-12.860006	.604733	

Figure A.4 Forward Integration Printout For Example A.1.

```
T=   1.865
---------

S(TILDE):          -2.300845

W(TILDE):          -3.448993

T=   1.532
---------

S(TILDE):          -15.361161

W(TILDE):          -16.391674
```



```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                                              !
! DVDQ SUSPENDS EXECUTION WITH IFLAG = 7 ,     !
!                                              !
! IF IFLAG = 7 (MINIMUM STEPSIZE EXCEEDED),    !
!                                              !
! LIKELY A CONJUGATE POINT EXISTS AT           !
!                                              !
!              T =  1.57342                    !
!                                              !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

Figure A.5 Printout Denoting The Occurrence Of A Conjugate Point In Example A.2.

```
      SUBROUTINE TIMVAR(T,N,M,A,DN,BINV,F,G)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(1),DN(1),BINV(1),F(1),G(1)
C
      F(1)=T*T
      G(1)=T
      RETURN
      END
```

Figure A.6 A Typical TIMVAR Subroutine For Example A.3.