Scientific Report No. 13

Control Theory Group

THE UNIVERSITY OF TENNESSEE

DEPARTMENT OF ELECTRICAL ENGINEERING

METHODS FOR INJECTION ERROR ANALYSIS

AND THEIR COMPARISON

by

J. D. Irwin and J. C. Hung

Supported by National Aeronautics and Space Administration

under Grant NsG-351

August 1, 1966

METHODS FOR INJECTION ERROR ANALYSIS AND THEIR COMPARISON

J. D. Irwin J. C. Hung

Department of Electrical Engineering
The University of Tennessee

32291

SUMMARY

Statistical techniques for the analysis of missile injection errors are studied in detail. The commonly used direct and adjoint methods are reviewed and extended. It is shown that the determination of the covariance matrix is equivalent to the determination of two transformation matrices for both methods. In general, the adjoint method is more efficient. But for a special case the direct method could be preferable depending upon the relative dimension of the system state and the error source. Two examples are given to verify the results. The techniques presented can equally well be applied to a wide variety of control system problems.

INTRODUCTION

In space flight, the desired trajectory of the missile, called the reference or nominal trajectory, is usually predetermined. Then a guidance and control scheme is used to keep the spacecraft flying along this reference trajectory. However, the actual trajectory of the missile, in general, deviates from the reference trajectory due to a large number of system errors. The trajectory errors at any time, such as engine burnout, are called "injection errors." The type of system errors involved will depend upon the hardware used in the system. Some typical system errors are listed in the following.

1. Gyro drifts,
2. accelerometer errors,
3. platform initial alignment errors,
4. airborne computer errors,
5. misalignment of thrust axis,
6. inexact knowledge of burnout time,
7. deviation of thrust from its nominal value,
8. misalignment of the missile's initial state, etc.

The precise injection errors of the trajectory cannot be evaluated due to uncertainties in error sources. Therefore we are forced to settle for the next best description, namely, the statistical knowledge of the injection errors.

Since the statistical knowledge of the error sources is obtained from preflight laboratory tests, the statistical knowledge of a certain trajectory deviation can usually be determined. This knowledge is indispensable for successful space flight operation. It indicates the probability of mission success, the probable range of the target, and it provides the information required for safety precautions.

Two methods are commonly used for statistical trajectory analysis, namely, the direct method[2,3] and the adjoint method.[3,4] Some analysts favor the former while others favor the latter.

The purpose of this paper is: first, to generalize these two methods and formulate them in the simple vector-matrix notation; and, secondly, to make a comparison between them. It will be shown that even though both methods result in the same amount of information, one method will be more efficient than the other depending upon the problem.

FORMULATION OF THE PROBLEM

Consider a missile system which is launched at time t_o. The dynamics of the trajectory deviations due to system errors is represented by a time-varying vector differential equation

$$\dot{\underline{x}}(t) = \underline{F}\left[\underline{x}(t),\ t,\ \underline{e}(t)\right] \tag{1}$$

which is, in general, nonlinear. In Eq. (1) $\underline{x}$ is an n-vector whose components are n state variables of the trajectory and $\underline{e}$ is an m-vector whose components represent m system errors. The state variables may represent any dynamical quantities, such as, position deviation components, velocity deviation components, mass, mass rate, fuel, fuel rate, etc.

In most practical cases, the trajectory deviation is small enough so that Eq. (1) can be satisfactorily approximated by its linear perturbation equation about $\underline{x} = 0$, which has the form

$$\dot{\underline{x}}(t) = A(t)\ \underline{x}(t) + B(t)\ \underline{e} \tag{2}$$

where A is an n x n matrix and B is an n x m matrix whose ij-elements are, respectively,

$$\left.\begin{aligned} a_{ij}(t) &= \frac{\partial F_i}{\partial x_j} \\ b_{ij}(t) &= \frac{\partial F_i}{\partial e_j} \end{aligned}\right\} \quad . \tag{3}$$

It is assumed that the random processes of the system errors are independent and stationary, with a zero mean Gaussian distribution.

$$f(\underline{e}) = \frac{1}{\sqrt{\det S}(2\pi)^{m/2}} \quad \exp\left[-\frac{1}{2}\ \underline{e}^T S^{-1} e\right] \tag{4}$$

where $S = E\left[\underline{e}\ \underline{e}^T\right]$ = covariance matrix of $\underline{e}$.

The zero mean assumption is valid, since if the random system errors have non-zero means they should be corrected before launch.

It will be shown later that the stationary assumption for the error processes is practical and imposes little if any limitation on our methods.

The assumption that the errors are Gaussianly distributed has both practical and theoretical justifications. In practice, the statistics of the individual errors can be satisfactorily approximated by Gaussian distributions. This is equivalent to approximating non-linear systems by linear systems. Furthermore, since the number of system errors is large and the errors are independent of one another, the central limit theorem asserts that the sums of these errors approaches a gaussian distribution in the limit regardless of the distribution of the individual system errors.

Our problem is to evaluate the statistical injection errors at a time $t_1 > t_o$.

COVARIANCE MATRIX OF INJECTION ERRORS

Finding the statistical injection errors amounts to determining their statistical distribution. Since the error dynamics are linear and the error sources are Gaussian with zero mean, the injection errors at any time $t_1 > t_o$ also have a Gaussian distribution with zero mean[6]

$$f(\underline{x}) = \frac{1}{\sqrt{\det M}\,(2\pi)^{n/2}} \exp\left[- \frac{1}{2} \underline{x}^T M^{-1} \underline{x} \right] \tag{5}$$

where $M = E\left[\underline{x}\,\underline{x}^T \right]$ = covariance matrix of injection errors.

Eq. (5) shows that once the covariance matrix M is determined, the complete distribution is specified. Therefore, our problem is reduced to the determination of M at $t_1 > t_o$.

DIRECT METHOD

Beginning the analysis, the solution of Eq. (2) is

$$\underline{x}(t) = G(t,t_o)\underline{x}(t_o) + \int_{t_o}^{t} G(t,\tau)B(\tau)\underline{e}(\tau)d\tau \tag{6}$$

where $G(t,t_o)$ is the transition matrix, a function of two variables.

Due to the stationary assumption the error vector $\underline{e}$ is constant in time, so that (6) reduces to

$$\underline{x}(t) = G(t,t_o)\underline{x}(t_o) + P(t)\underline{e} \tag{7}$$

where

$$P(t) = \int_{t_o}^{t} G(t,\tau)B(\tau)d\tau \quad . \tag{8}$$

Notice that $P(t)$ is an n x m transformation matrix.

In all practical cases, $\underline{x}(t_o)$ and $\underline{e}$ are uncorrelated. The covariance matrix of the injection errors is therefore

$$\begin{aligned} M(t) &= E\left[\underline{x}(t)\underline{x}^T(t)\right] \\ &= G(t,t_o)M_oG^T(t,t_o) + P(t)S\,P^T(t) \end{aligned} \tag{9}$$

where

$$\left.\begin{aligned} M_o &= E\left[\underline{x}(t_o)\underline{x}^T(t_o)\right] = \text{covariance matrix of initial deviations} \\ S &= E\left[\underline{e}\,\underline{e}^T\right] = \text{covariance matrix of system errors} \end{aligned}\right\} \quad . \tag{10}$$

Therefore the problem of determining M reduces to the determination of the matrices $G(t,t_o)$ and $P(t)$.

As shown in (8), $P(t)$ requires the knowledge of $G(t,\tau)$ as a function of τ. In general, for a time-varying system G cannot be obtained analytically. However, for a specific time $t = t_1$, the elements of the constant matrix $P(t_1)$ can be obtained by applying the method of numerical solution to (2) from t_o to t_1 m times as follows.

Let $\underline{x}(t_o) = 0$ and let all $e_i = 0$, except $e_k = 1$. Then (2) becomes

$$\dot{\underline{x}} = A(t)\underline{x} + \underline{b}_k(t) \qquad x(t_o) = 0 \tag{11}$$

where $\underline{b}_k$ is the k-th column vector of B(t). Integrating (11) from t_o to t_1 gives

$$\underline{x} = \underline{p}_k = \text{k-th column of } P(t_1) \quad . \tag{12}$$

Or, in expanded form,

$$\begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} = \begin{bmatrix} p_{1k} \\ p_{2k} \\ \cdot \\ \cdot \\ \cdot \\ p_{nk} \end{bmatrix} \quad . \tag{13}$$

If the system error vector has m components, then m integrations are needed to obtain the complete matrix P(t).

A similar procedure is used to obtain the matrix G. Let $\underline{e} = 0$ and let all $x_i(t_o) = 0$, except $x_k(t_o) = 1$. Then (2) becomes

$$\dot{\underline{x}} = \underline{a}_k \tag{14}$$

where $\underline{a}_k$ is the k-th column vector of A(t). Obtaining the numerical solution of (14) from t_o to t_1, gives

$$\underline{x} = \underline{g}_k = \text{k-th column of } G(t_1,t_o) \quad . \tag{15}$$

For n-th order trajectory dynamics, n integrations are required to obtain the complete matrix $G(t_1,t_o)$.

<u>A Speical Case</u> An important special case occurs when $\underline{x}(t_o) = 0$. Under this condition, the covariance matrix of the injection error is simply

$$M(t) = P(t)S\,P^T(t) \quad , \tag{16}$$

and the n integrations required for G are no longer needed.

ADJOINT METHOD

In the adjoint method, we first form the adjoint equation[5] of (2)

$$\dot{\underline{\lambda}} = - A^T(t)\,\underline{\lambda} \tag{17}$$

where $\underline{\lambda}$ is an n-vector. Using equations (2) and (17), the following is easily obtained

$$\frac{d}{dt}\ \underline{\lambda}^T\underline{x} = \underline{\lambda}^T B(t)\underline{e}\ . \tag{18}$$

Integrating (13) from t_o to t_1

$$\underline{\lambda}^T(t_1)\underline{x}(t_1) = \underline{\lambda}^T(t_o)\underline{x}(t_o) + \int_{t_o}^{t_1} \underline{\lambda}^T B(t)\underline{e}\ dt\ . \tag{19}$$

Letting

$$\left.\begin{array}{ll} \lambda_i(t_1) = 0 & i \neq k \\ \lambda_k(t_1) = 1 & \end{array}\right\} \quad k = 1\text{---}n \tag{20}$$

and noting that $\underline{e}$ is constant, so (19) becomes

$$x_k(t_1) = \underline{\lambda}_k^T(t_o)\underline{x}(t_o) + \underline{p}_k^T(t_1)\underline{e} \qquad k = 1\text{---}n \tag{21}$$

where

$$\underline{p}_k^T(t_1) = \int_{t_o}^{t_1} \underline{\lambda}_k^T(t)B(t)dt \qquad k = 1\text{---}n \tag{22}$$

is a row vector of dimension m.

The vector form of (21) is

$$\underline{x}(t_1) = G(t_1,t_o)\underline{x}(t_o) + P(t_1)\underline{e} \tag{23}$$

where

$$G(t_1,t_o) = \begin{bmatrix} \underline{\lambda}_1^T(t_o) \\ \underline{\lambda}_2^T(t_o) \\ \cdot \\ \cdot \\ \cdot \\ \underline{\lambda}_n^T(t_o) \end{bmatrix} \tag{24}$$

$$P(t_1) = \begin{bmatrix} \underline{p}_1^T(t_1) \\ \underline{p}_2^T(t_1) \\ \cdot \\ \cdot \\ \cdot \\ \underline{p}_n^T(t_1) \end{bmatrix} \quad . \tag{25}$$

By comparing (23) to (7), we see that the matrices G and P in one equation should be identical to those in the other.

Eqs. (23) and (10), once again, give the same covariance matrix (9) of the injection errors, which is

$$M(t_1) = E\left[x(t_1)x^T(t_1)\right]$$

$$= G\ M_o\ G^T + P\ S\ P^T \quad . \tag{9}$$

Therefore, again the problem reduces to the determination of G and P.

From the above derivation we see that each row of the matrix G can be obtained by using the numerical solution method on (17) from t_1 to t_o backward in time with initial conditions (20). Then each row of P is obtained by integrating (22). This procedure must be repeated n times, with n different initial conditions, to give the complete description of G and P.

<u>The Special Case</u> For the same special case considered before, $\underline{x}(t_o) = 0$,

$$M(t_1) = P(t_1)\ S\ P^T(t_1) \quad . \tag{27}$$

However, note that one must still solve (17) to get $\underline{\lambda}_k^T(t)$ and then integrate (22) n times to get a row of P.

COMPARISON

A comparison of the two methods is shown below.

Direct Method

D-1. Given: $\dot{\underline{x}} = A\underline{x} + B\underline{e}$ (i)

M_o and S

D-2. To obtain the k-th column of P, compute the solution of (i) with the initial conditions

$$e_i = \delta_{ik} \quad i = 1\text{---}n$$ (ii)

$$\underline{x}(0) = 0$$

D-3. To obtain the k-th column of G, compute the solution of (i) with the initial conditions

$$\underline{e} = 0$$ (iii)

$$x_i = \delta_{ik} \quad i = 1\text{---}n$$

D-4. Perform D-2 m times and D-3 n times to give P and G.

D-5. $M(t_1) = G\,M_o G^T + P\,S\,P^T$. (iv)

Adjoint Method

A-1. Given: $\dot{\underline{x}} = A\underline{x} + B\underline{e}$ (i)

M_o and S

are therefore

$$\dot{\underline{\lambda}} = -A^T\underline{\lambda}.$$ (ii)

A-2. To obtain the k-th row of G, compute the solution of (ii), backward in time with initial conditions

$$\lambda_i(t_1) = \delta_{ik}$$ (iii)

A-3. To obtain the k-th row of P, use the value of $\underline{\lambda}(t)$ obtained in A-2 to evaluate the integral

$$\underline{p}_k^T = -\int_{t_1}^{t_o} \underline{\lambda}_k^T(t)B(t)dt \ .$$ (iv)

A-4. Perform A-2 and A-3 n times to give G and P.

A-5. $M(t_1) = G\,M_o G^T + P\,S\,P^T$ (v)

Figures 1 and 2 are diagrams which illustrate the sequence of computations required for each method.

Several remarks can be made:

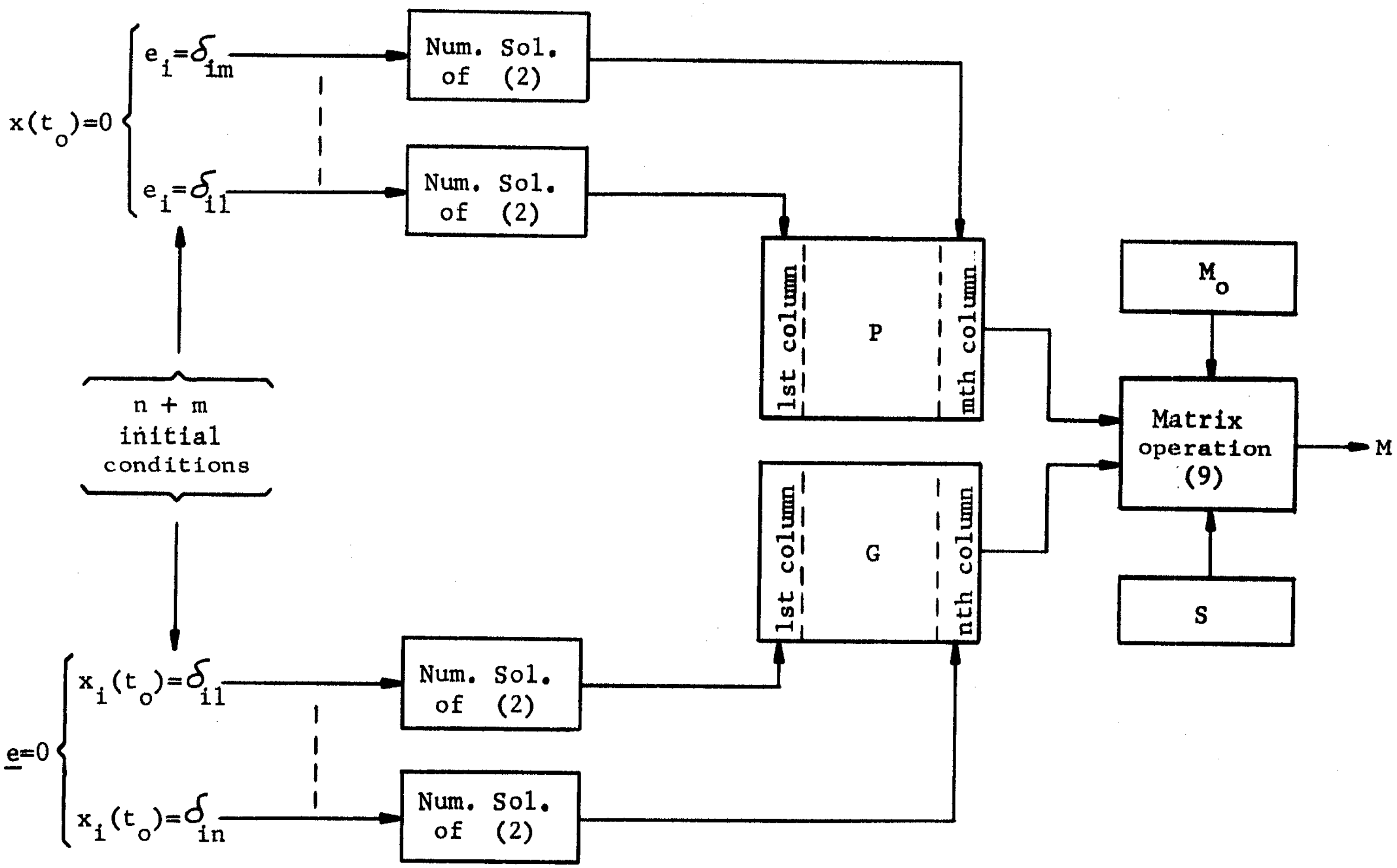

Figure 1. Diagram of direct method.

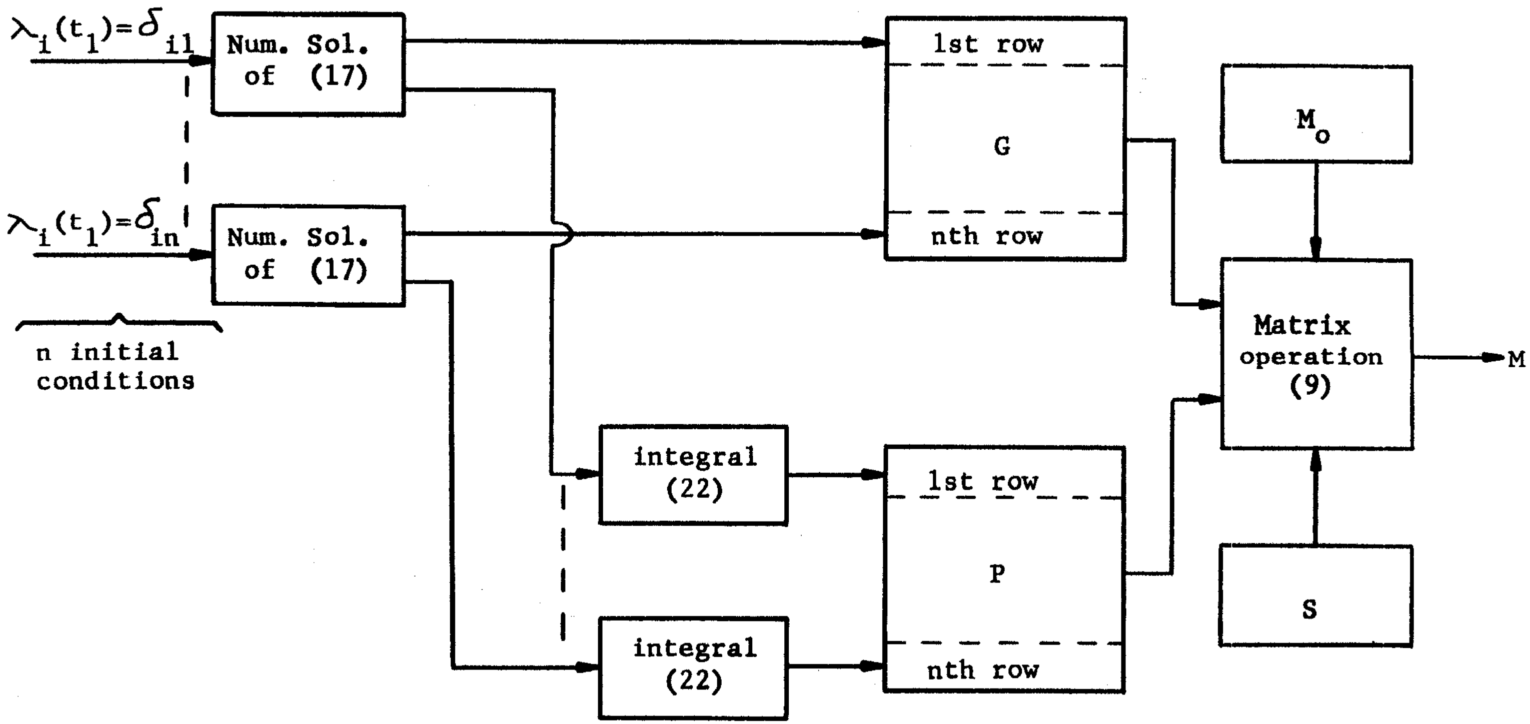

Figure 2. Diagram of adjoint method.

1. In both methods the determination of the covariance matrix of the injection errors is reduced to the determination of the transformation matrices G and P.

2. In the direct method, $\underline{m + n}$ numerical solutions are needed to give P and G, m for P and n for G.

3. In the adjoint method, $\underline{n}$ numerical solutions are needed to give G and to provide data for the n integrations which lead to P.

4. Since the system differential equation is more complex than its adjoint differential equation due to the forcing term $B\underline{e}$, the complexity of each numerical solution involved in either D-2 or D-3 is approximately equivalent to each combined solution of A-2 and A-3. Therefore we conclude that when $\underline{x}(t_o) \neq 0$, the adjoint method is always more efficient than the direct method.

5. For the special case $\underline{x}(t_o) = 0$, only P is needed. The direct method requires only m numerical solutions while the adjoint method still requires n. Therefore the choice of the method depends upon two numbers n and m, which represent the dimension of state $\underline{x}$ and the dimension of the error source $\underline{e}$ respectively. When $n \gg m$, direct method is preferable; but when $m \gg n$, the adjoint method is certainly more efficient.

6. For the special case $\underline{e} = 0$, only G is needed. Both methods require n numerical solutions and either method is as good as the other.

7. We have assumed, when formulating the problem, that the random processes of the system errors be stationary. This assumption imposes little limitation on the methods. For all practical cases, the time-varying characteristics of error processes can be handled by writing

$$\underline{e}(t) = c(t)\underline{e}' \tag{28}$$

where $\underline{e}'$ is a constant q-vector; $q \geq m$ in general. The time-varying matrix c(t) is m x q. By defining

$$B'(t) = B(t)\ c(t) \tag{29}$$

the forcing term of Eq. (2) becomes

$$B'(t)\ \underline{e}' \ . \tag{30}$$

EXAMPLES

Example 1. Consider a simplified error dynamics of a space vehicle given by the following set of differential equations

$$\left.\begin{aligned} \ddot{y} + \frac{u}{r^3}\,y - \frac{3u}{r^5}\left[\,Y^2 y + Y(2 + r_o)z\,\right] &= e_1 + A_2 e_2 t \\ \ddot{z} + \frac{u}{r^3}\,z - \frac{3u}{r^5}\left[\,Y(2 + r_o)y + (Z + r_o)^2 z\,\right] &= e_2 - A_1 e_2 t \end{aligned}\right\} \quad (31)$$

where Y, Z - nominal trajectory position

y, z - position errors of Y, Z, respectively

$r = \sqrt{r^2 + (Z + r_o)^2}$ - radius from earth's center to vehicle

$u = g_o r_o^2$ - gravitational constant

$g_o = 9.81$ meters/second^2

$r_o = 6.37 \times 10^6$ meters

A_1, A_2 - nominal sensed acceleration along Y and Z respectively

e_1, e_2 - bias in Y- and Z-accelerometers respectively

e_3 - constant platform drift rate about x-axis

t - time variable.

All errors e_1, e_2 and e_3 are random, uncorrelated, and have zero means. Their standard deviations are

$$\sigma_{e_1} = \sigma_{e_2} = 10^{-4} \text{ meter/sec}^2$$

$$\sigma_{e_3} = 5 \times 10^{-6} \text{ radian/sec}$$

The initial trajectory error is taken as zero.

Defining the state variables

$$x_1 = y, \qquad x_2 = \dot{y}, \qquad x_3 = z, \qquad x_4 = \dot{z}, \tag{32}$$

the state equation of (31) is

$$\dot{\underline{x}} = A(t)\underline{x} + B(t)\,\underline{e} \qquad \underline{x}(0) = 0 \tag{33}$$

where $\underline{x} = \left[x_1, x_2, x_3, x_4 \right]^T$

$\underline{e} = \left[e_1, e_2, e_3 \right]^T$

$$A(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \dfrac{3uY^2}{r^5} - \dfrac{u}{r^3} & 0 & \dfrac{3uY(Z + r_o)}{r^5} & 0 \\ 0 & 0 & 0 & 1 \\ \dfrac{3uY(Z + r_o)}{r^5} & 0 & \dfrac{3u(Z + r_o)^2}{r^5} - \dfrac{u}{r^3} & 0 \end{bmatrix} \tag{34}$$

$$B(t) = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & A_2 t \\ 0 & 0 & 0 \\ 0 & 1 & -A_1 t \end{bmatrix} . \tag{35}$$

Notice that $n = 4$ and $m = 3$. Values of Y, Z, A_1 and A_2 as functions of time are contained in Appendix 1.

Both direct and adjoint methods were applied to find the transformation and covariance matrices at the final time $t = 853.6$ sec, with the initial time being $t = 0$. The computations were carried out by an IBM-7040 digital computer. Appendix 1 contains the computer programs for the two methods in Fortran language and all the data. The numerical results are given below.

Direct Method:

$$P = \begin{bmatrix} 0.34014471\ 06 & 0.21969810\ 05 & 0.15577654\ 09 \\ 0.77391915\ 03 & 0.15434290\ 03 & -0.91316429\ 06 \\ 0.20111238\ 05 & 0.42568219\ 06 & -0.13046488\ 10 \\ 0.13661971\ 03 & 0.11185155\ 04 & -0.45767998\ 07 \end{bmatrix}$$

$$M = \begin{bmatrix} 0.60782003\ 06 & -0.35535729\ 04 & -0.50806797\ 07 & -0.17823240\ 05 \\ 0.35535729\ 04 & 0.20852953\ 02 & 0.29784779\ 05 & 0.10448703\ 03 \\ 0.50806797\ 07 & 0.29784779\ 05 & 0.42554526\ 08 & 0.14928269\ 06 \\ 0.17823240\ 05 & 0.10448703\ 03 & 0.14928269\ 06 & 0.52369010\ 03 \end{bmatrix}$$

Adjoint Method:

$$P = \begin{bmatrix} 0.33980686\ 06 & 0.21823815\ 05 & 0.15780548\ 09 \\ 0.77376799\ 03 & 0.15387262\ 03 & -0.91028112\ 06 \\ 0.19978623\ 05 & 0.42534883\ 06 & -0.12998004\ 10 \\ 0.13642535\ 03 & 0.11184063\ 04 & -0.45736717\ 07 \end{bmatrix}$$

$$M = \begin{bmatrix} 0.62372363\ 06 & -0.35885207\ 04 & -0.51277296\ 07 & -0.18043053\ 05 \\ 0.35885207\ 04 & 0.20721517\ 02 & 0.29580402\ 05 & 0.10408595\ 03 \\ 0.51277296\ 07 & 0.29580402\ 05 & 0.42238837\ 08 & 0.14862628\ 06 \\ 0.18043053\ 05 & 0.10408595\ 03 & 0.14862629\ 06 & 0.52297451\ 03 \end{bmatrix}$$

Note that the following notation is being used above:

$0.34014471\ 06 = 0.34014471 \times 10^6$.

Computer time (including loading time and execution time):

direct method - 0.40 minute

adjoint method - 0.41 minute.

Comparing the numerical values, we see that both methods led to the same results. However, the direct method is more efficient in this case since it consumed less time. This is what we expected, since here $\underline{x}(0) = 0$ and $n \gg m$. The difference in time is not very much in this example due to the small difference between n and m.

<u>Example 2.</u> Consider a second order system with six system errors.

$$\dot{\underline{x}} = A\underline{x} + B\underline{e}$$

where $\underline{x} = \left[x_1, x_2\right]^T$, $\underline{e} = \left[e_1, e_2, e_3, e_4, e_5, e_6\right]^T$

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} .$$

Assume $\underline{x}(0) = 0$ and assume that the system errors are independent with standard deviations

$$\sigma_{e_1} = \sqrt{0.1} \qquad \sigma_{e_2} = \sqrt{0.05} \qquad \sigma_{e_3} = \sqrt{0.01}$$

$$\sigma_{e_4} = \sqrt{0.2} \qquad \sigma_{e_5} = \sqrt{0.06} \qquad \sigma_{e_6} = \sqrt{0.04} .$$

The problem is to evaluate the transformation matrix and covariance matrix by both methods.

Notice, the system is time-invariant, so the solution can easily be obtained analytically. However, to demonstrate the main theme of this paper, we still use the numerical solution technique and let the digital computer carry out the computations. In this system $m = 6$ and $n = 2$.

The result, for the direct method, is

$$P^T = \begin{bmatrix} 0.49000891\ 02 & 0.99999254\ 00 \\ 0.50000862\ 02 & 0. \\ 0.99001753\ 02 & 0.99999254\ 00 \\ 0.49000891\ 02 & 0.99999254\ 00 \\ 0.50000862\ 02 & 0. \\ 0.99001753\ 02 & 0.99999254\ 00 \end{bmatrix}$$

$$M = \begin{bmatrix} 0.14854030\ 04 & 0.19650208\ 02 \\ 0.19650208\ 02 & 0.34999478\ 00 \end{bmatrix} ,$$

and for the adjoint method

$$P^T = \begin{bmatrix} 0.48985892\ 02 & 0.99994580\ 00 \\ 0.50000000\ 02 & 0. \\ 0.98985891\ 02 & 0.99994580\ 00 \\ 0.48985892\ 02 & 0.99994580\ 00 \\ 0.50000000\ 02 & 0. \\ 0.98985891\ 02 & 0.99994580\ 00 \end{bmatrix}$$

$$M = \begin{bmatrix} 0.14847956\ 04 & 0.19643997\ 02 \\ 0.19643997\ 02 & 0.34996206\ 00 \end{bmatrix} .$$

The computer times are:

direct method - 12.69 minutes

adjoint method - 2.61 minutes.

Here, as we expected, the adjoint method is much faster. Appendix 2 contains the complete program for the problem.

CONCLUSION

The techniques for statistical analysis of the injection errors of a missile have been studied in detail. Two commonly used methods, namely, the direct and adjoint methods, have been reviewed and extended. It has been demonstrated that, under the assumption of Gaussian random processes and linear error dynamics, all the necessary information is contained in covariance matrix of the injection errors. It has further been shown that for both direct and adjoint methods, the determination of the covariance matrix is equivalent to the determination of two transformation matrices.

A comparison of the two methods has revealed that, in general, the adjoint method is more efficient. But for the special case, where the initial state deviation is zero, the direct method could be preferable depending upon the relative dimension of the system state and the error sources.

Several remarks have been made to emphasize the main points and two examples have been given to verify the theoretical conclusion.

The techniques presented here are not limited to the injection errors of a missile; they can be applied equally well to a wide variety of control system problems.

ACKNOWLEDGMENTS

The research reported here was supported by National Aeronautics and Space Administration through Grant NsG-351. The authors thank Mr. H. C. Desai for his help in preparing the computer result for the examples, and Mrs. Shirley Kirk for managing the manuscript. The computer time was supported by National Science Foundation.

REFERENCES

1. Macomber, G. R. and M. Fernandez, "Inertial Guidance Engineering," Prentice-Hall, Inc., Englewood Cliff, N. J., 1962.

2. Norton, A. R. M., "The Statistical Analysis of Space Guidance Systems," Tech. Memo. No. 33-15, Jet Propulsion Lab., 1960.

3. Irwin, J. D. and J. C. Hung, "The Theory and Application of Statistical Design Techniques to Problems of Guidance, Control and Navigation," Scientific Report No. 11, Control Theory Group, E. E. Dept., University of Tennessee, August, 1965.

4. Peske, A. and M. Ward, "Techniques for Error Analysis of Trajectories," Guidance and Control, Edited by R. E. Robertson and J. S. Farrion, Academic Press, New York, 1962.

5. Coddington, E. A. and N. Levinson, "Theory of Ordinary Differential Equations," McGraw-Hill Book Co., New York, 1955.

6. Cramer, H., "Mathematical Methods of Statistics," Princeton University Press, Princeton, N. J., 1946.

APPENDIX I

DATA AND PROGRAMMING FOR EXAMPLE I

TIME RECORD OF Y, Z, A1 AND A2

TIME	Y	Z	A1	A2
+.000000000E+00	+.000000000E+00	+.000000000E+00	+.000000000E+00	+.120799580E+02
+.800000001E+01	+.327080001E+04	+.774000002E+02	+.000000000E+00	+.125616170E+02
+.160000000E+02	+.654130001E+04	+.330600000E+03	+.724415412E-01	+.130824120E+02
+.240000000E+02	+.981730001E+04	+.793200002E+03	+.434657900E-00	+.136702070E+02
+.320000000E+02	+.131214000E+05	+.150270000E+04	+.100800650E+01	+.143013510E+02
+.400000000E+02	+.164903000E+05	+.249970000E+04	+.187102800E+01	+.149689500E+02
+.480000000E+02	+.199790000E+05	+.382670000E+04	+.309198121E+01	+.156259500E+02
+.560000001E+02	+.236652000E+05	+.552560001E+04	+.458397441E+01	+.161591550E+02
+.640000001E+02	+.276427000E+05	+.762870001E+04	+.614840531E+01	+.162030730E+02
+.713540002E+02	+.316410000E+05	+.992180001E+04	+.756227761E+01	+.159052600E+02
+.720000001E+02	+.320104000E+05	+.101390000E+05	+.771688391E+01	+.159357100E+02
+.780000001E+02	+.356041001E+05	+.122832000E+05	+.938704022E+01	+.165115960E+02
+.820980002E+02	+.382483001E+05	+.138857000E+05	+.106870600E+02	+.170085680E+02
+.880000001E+02	+.423758001E+05	+.164090000E+05	+.126661420E+02	+.176696090E+02
+.960000001E+02	+.486854000E+05	+.202724000E+05	+.154503620E+02	+.184029930E+02
+.104000000E+03	+.559806001E+05	+.246898000E+05	+.183388170E+02	+.190295710E+02
+.112000000E+03	+.644447001E+05	+.297017000E+05	+.213673500E+02	+.196411870E+02
+.120000000E+03	+.742712002E+05	+.353485000E+05	+.246218340E+02	+.203317860E+02
+.128000000E+03	+.856682002E+05	+.416762000E+05	+.282318200E+02	+.211937300E+02
+.136000000E+03	+.988667002E+05	+.487418001E+05	+.323945421E+02	+.223380760E+02
+.138517000E+03	+.103433100E+06	+.511287000E+05	+.338601490E+02	+.227771100E+02
+.144000000E+03	+.113865900E+06	+.564438001E+05	+.168718450E+02	+.113493850E+02
+.144517000E+03	+.114876600E+06	+.569476001E+05	+.178761430E+02	+.114078160E+02
+.144517000E+03	+.114876600E+06	+.569476001E+05	-.266089220E-01	-.168971780E-01
+.150017000E+03	+.125651100E+06	+.621644001E+05	-.135619980E-01	-.816433102E-02
+.150017000E+03	+.125651100E+06	+.621644001E+05	-.135679770E-01	-.816433102E-02
+.150017000E+03	+.125651100E+06	+.621644001E+05	+.496544971E+01	+.410229881E+01
+.160017000E+03	+.145476400E+06	+.711239001E+05	+.510871041E+01	+.440107741E+01
+.160017000E+03	+.145476400E+06	+.711239001E+05	+.511245431E+01	+.440430270E+01
+.172000000E+03	+.169878900E+06	+.811763001E+05	+.529025981E+01	+.440034891E+01
+.220000000E+03	+.275100500E+06	+.114040800E+06	+.607381949E+01	+.435132581E+01
+.244000000E+03	+.332519900E+06	+.126059800E+06	+.651405425E+01	+.430235131E+01
+.268000000E+03	+.393416000E+06	+.135139000E+06	+.699152672E+01	+.423283871E+01
+.292000000E+03	+.458015300E+06	+.141261600E+06	+.751065367E+01	+.413883071E+01
+.316000000E+03	+.526565201E+06	+.144393900E+06	+.807656900E+01	+.401557851E+01
+.340000000E+03	+.599337201E+06	+.144482300E+06	+.869529904E+01	+.385736490E+01
+.364000000E+03	+.676629501E+06	+.141451000E+06	+.937399866E+01	+.365727650E+01
+.388000000E+03	+.758771501E+06	+.135197800E+06	+.101212867E+02	+.340690530E+01
+.412000000E+03	+.846217602E+06	+.125589900E+06	+.109477198E+02	+.309594860E+01
+.436000000E+03	+.939103700E+06	+.112458200E+06	+.118667132E+02	+.271165290E+01
+.460000000E+03	+.103815360E+07	+.955905001E+05	+.128944549E+02	+.223801900E+01
+.484000000E+03	+.114378920E+07	+.747227002E+05	+.140537226E+02	+.165462470E+01
+.508000001E+03	+.125659120E+07	+.495272000E+05	+.153741100E+02	+.934814501E+00
+.532000001E+03	+.137723020E+07	+.195985000E+05	+.168970932E+02	+.428068001E-01
+.556000000E+03	+.150648250E+07	-.155671000E+05	+.186823604E+02	-.107111450E+01
+.580000001E+03	+.164527210E+07	-.565997000E+05	+.208191602E+02	-.247897360E+01
+.604000001E+03	+.179471800E+07	-.104296800E+06	+.234469741E+02	-.429048751E+01
+.628000001E+03	+.195621670E+07	-.159686500E+06	+.267958717E+02	-.668325901E+01
+.639915002E+03	+.204140940E+07	-.190446800E+06	+.288409837E+02	-.817465852E+01
+.639915002E+03	+.204140940E+07	-.190446800E+06	+.697197991E+01	-.197613050E+01
+.651000001E+03	+.212257910E+07	-.220835200E+06	+.704132551E+01	-.222347550E+01
+.675000001E+03	+.229998250E+07	-.291349401E+06	+.717816951E+01	-.278689420E+01
+.699000000E+03	+.247959870E+07	-.368553000E+06	+.729599551E+01	-.338884470E+01
+.723000001E+03	+.266133980E+07	-.452741900E+06	+.739434251E+01	-.403012590E+01
+.747000001E+03	+.284510480E+07	-.544227400E+06	+.747341411E+01	-.471222231E+01
+.771000001E+03	+.303077950E+07	-.643336801E+06	+.753407482E+01	-.543766181E+01
+.795000000E+03	+.321823771E+07	-.750414201E+06	+.757780721E+01	-.621042841E+01
+.819000001E+03	+.340734230E+07	-.865822500E+06	+.760664521E+01	-.703643602E+01
+.831000000E+03	+.350246650E+07	-.926769802E+06	+.761624471E+01	-.747191681E+01
+.843000000E+03	+.359794680E+07	-.989945801E+06	+.762309291E+01	-.792409181E+01
+.853633000E+03	+.368282860E+07	-.104782650E+07	+.762716282E+01	-.833993871E+01

A PROGRAM FOR DIRECT METHOD

```
$IBFTC MAIN
C     *     *     *     *     *     *     *     *     *     *
      DIMENSION T(61), Y(61), Z(61), R(61), A21(61),
     1   A23(61), A41(61), A43(61), B23(61),B43(61),
     2   A1(61), A2(61)
      RO = 6370000.0
      GO = 9.81
      READ 500, (T(I), Y(I), Z(I), A1(I), A2(I), I = 1,61)
 500  FORMAT(5(E15.8, 1X))
      PMU = GO*RO**2
      DO 10 I = 1,61
      R(I) = SQRT(Y(I)**2 + (Z(I) + RO)**2)
      A21(I) = 3.*PMU*Y(I)**2/R(I)**5 - PMU/R(I)**3
      A23(I) = 3.*PMU*Y(I)*(Z(I) + RO)/R(I)**5
      A41(I) = 3.*PMU*Y(I)*(Z(I) + RO)/R(I)**5
      A43(I) = 3.*PMU*(Z(I) + RO)**2/R(I)**5 - PMU/R(I)**3
      B23(I) = A2(I)*T(I)
      B43(I) = -A1(I)*T(I)
 10   CONTINUE
C     INITIAL CONDITIONS
      DIMENSION E(3),X1(3,61),X2(3,61),X3(3,61),X4(3,61)
      DO 50 I = 1,3
      DO 20 J = 1,3
      IF(I-J) 7,8,7
 8    E(J) = 1.C
      GO TO 20
 7    E(J) = 0.0
 20   CONTINUE
      X1(I,1) = 0.0
      X2(I,1) = 0.0
      X3(I,1) = 0.0
      X4(I,1) = 0.0
C     SOLUTIONS OF DIFFERENTIAL EQUATIONS
      X1(I,2) = X1(I,1)
      X2(I,2) = X2(I,1) +8.0*(E(1) +B23(1)* E(3))
      X3(I,2) = X3(I,1)
      X4(I,2) = Y4(I,1) + 8.0* (E(2)  + B43 (1) *E(3))
      N = 1
   15 X1(I,2) = X1(I,1) + 4.0*(X2(I,1) + X2(I,2))
      X2(I,2) = 4.0*(A21(2)*X1(I,2) +A23(2)*X3(I,2)+2.*E(1)
     1   + B23(2)*E(3) + B23(1)*E(3) + A21(1)*X1(I,1) +
     2   A23(1)*X3(I,1))+ X2(I,1)
      X3(I,2) = X3(I,1) + 4.0*(X4(I,1) + X4(I,2))
      X4(I,2) = 4.0*(A41(2)*X1(I,2) +A43(2)*X3(I,2)+2.*E(2)
```

```
     1    + B43(2)*E(3) + B43(1)*E(3) + A41(1)*X1(I,1) +
     2    A43(1)*X3(I,1))+ X4(I,1)
      N = N+1
      IF(N-8)15,16,16
16    DO 30 K = 3,61
      HT = (T(K) - T(K-1))*0.5
      X1(I,K) = X1(I,K-1) + HT*(X2(I,K-1) +X2(I,K-2))
      X2(I,K) = X2(I,K-1) + HT*(X1(I,K-1)*A21(K-1) +2.0*E(1)
     1  +A23(K-1)*X3(I,K-1) + B23(K-1)*E(3) +B23(K-2)*E(3)
     2    + A21(K-2)*X1(I,K-2) + A23(K-2)*X3(I,K-2))
      X3(I,K) = X3(I,K-1) + HT*(X4(I,K-1) +X4(I,K-2))
      X4(I,K) = X4(I,K-1) + HT*(X1(I,K-1)*A41(K-1) +2.0*E(2)
     1  +A43(K-1)*X3(I,K-1) + B43(K-1)*E(3) +B43(K-2)*E(3)
     2    + A41(K-2)*X1(I,K-2) + A43(K-2)*X3(I,K-2))
      N = 1
   17 X1(I,K) = X1(I,K-1) + HT*(X2(I,K-1) +X2(I,K))
      X2(I,K) = X2(I,K-1) + HT*(X1(I,K-1)*A21(K-1) +2.0*E(1)
     1    + A23(K-1)*X3(I,K-1) + B23(K-1)*E(3) + B23(K)*E(3)
     2    + A21(K)*X1(I,K) + A23(K)*X3(I,K))
      X3(I,K) = X3(I,K-1) + HT*(X4(I,K-1) +X4(I,K))
      X4(I,K) = X4(I,K-1) + HT*(X1(I,K-1)*A41(K-1) +2.0*E(2)
     1    + A43(K-1)*X3(I,K-1) + B43(K-1)*E(3) + B43(K)*E(3)
     2    + A41(K)*X1(I,K) + A43(K)*X3(I,K))
      N = N + 1
      IF( N - 8) 17, 30, 30
30    CONTINUE
50    CONTINUF
      DIMENSION P(4,3)
      DO 40 I = 1,3
      P(1,I) = X1(I,61)
      P(2,I) = X2(I,61)
      P(3,I) = X3(I,61)
      P(4,I) = X4(I,61)
40    CONTINUE
C     COVARIENCE MATRIX OF STATE VECTOR ERROR
      DIMENSION ER(3,3),C(4,3) ,D(4,4)
      READ 61,((ER(I,J),I=1,3),J=1,3)
61    FORMAT (5E8.0/4E8.0)
      DO 70 I = 1,4
      DO 70 J = 1,3
      C(I,J) = 0.0
      DO 70 K = 1,3
      C(I,J) = C(I,J) + P(I,K)*ER(K,J)
70    CONTINUE
      DO 80 I = 1,4
      DO 80 J = 1,4
```

```
      D(I,J) = 0.0
      DO 80 K = 1,3
      D(I,J) = D(I,J) + C(I,K)* P(J,K)
 80   CONTINUE
      PRINT 201, ((P(I,J), J = 1,3), I = 1,4)
      PRINT 202, ((D(I,J), J = 1,4), I = 1,4)
      PRINT 203,((ER(I,J), J = 1,3), I = 1,3)
 201  FORMAT(1H1,38H TRANSFORM MATRIX P BY DIRECT METHOD  /
     1   ///3(2X,E15.8,3X))
 202  FORMAT(////38H COVARIENCE MATRIX OF STATE VECTOR    ,
     1   23H ERROR BY DIRECT METHOD///4(2X,E15.8,3X))
 203  FORMAT(////38H COVARIENCE MATRIX OF SOURCE VECTOR   ,
     1   6H ERROR///3(2X,E15.8,3X))
      CALL EXIT
      END
$ENTRY
      DATA CARDS
 +.1E-07 +.0E+00 +.0E+00 +.0E+00 +.1E-07
 +.0E+00 +.0E+0C +.0E+00+.25E-10
$IBSYS
```

NOTE-1 DATA CARDS INCLUDES THE TIME RECORD OF Y, Z, A1 AND A2 WHICH IS LISTED IN THE BEGINNING OF APPENDIX-I.

A PROGRAM FOR ADJOINT METHOD

```
$IBFTC MAIN
C     *     *     *     *     *     *     *     *     *     *
      DIMENSION T(61), Y(61), Z(61), R(61), A21(61),
     1   A23(61), A41(61), A43(61), B23(61),B43(61),
     2   A1(61), A2(61)
      RO = 6370000.0
      GO = 9.81
      READ 500, (T(I), Y(I), Z(I), A1(I), A2(I), I = 1,61)
 500  FORMAT(5(E15.8, 1X))
      PMU = GO*RO**2
      DO 10 I = 1,61
      R(I) = SQRT(Y(I)**2 + (Z(I) + RO)**2)
      A21(I) = 3.*PMU*Y(I)**2/R(I)**5 - PMU/R(I)**3
      A23(I) = 3.*PMU*Y(I)*(Z(I) + RO)/R(I)**5
      A41(I) = 3.*PMU*Y(I)*(Z(I) + RO)/R(I)**5
      A43(I) = 3.*PMU*(Z(I) + RO)**2/R(I)**5 - PMU/R(I)**3
      B23(I) = A2(I)*T(I)
      B43(I) = -A1(I)*T(I)
 10   CONTINUE
      DIMENSION XLE(4,70)
C     INITIAL CONDITIONS
      DO 100 I = 1,4
      DO 90 J = 1,4
      IF(I-J)2,1,2
 1    XLE(J,61) = 1.0
      GO TO 90
 2    XLE(J,61) = 0.0
 90   CONTINUE
C     SOLUTIONS OF DIFFERENTIAL EQUATIONS
      XLE(1,60) = XLE(1,61)+(A21(61)*XLE(2,61)
     1   + A41(61)*XLE(4,61))*10.6333
      XLE(2,60) = XLE(2,61) + XLE(1,61)*10.633
      XLE(3,60) = XLE(3,61)+(A23(61)*XLE(2,61)
     1   + A43(61)*XLE(4,61))*10.6333
      XLE(4,60) = XLE(4,61) + XLE(3,61)*10.633
      N = 1
   21 XLE(1,60) = XLE(1,61) +5.3165*(A21(61)*XLE(2,61)
     1   + A41(61)*XLE(4,61)+ A21(60)*XLE(2,60)
     2   + A41(60)*XLE(4,60))
      XLE(2,60) = XLE(2,61) +5.1365*(XLE(1,61) + XLE(1,60))
      XLE(3,60) = XLE(3,61) +5.1365*(A23(61)*XLE(2,61)
     1   + A43(61)*XLE(4,61)+ A23(60)*XLE(2,60)
     2   + A43(60)*XLE(4,60))
      XLE(4,60) = XLE(4,61) +5.1365*(XLE(3,61) + XLE(3,60))
```

```
      N = N +1
      IF(N-8)21,22,22
22    DO 60 K = 1,59
      N = 60 - K
      DT = (T(N+1)-T(N))*0.5
      XLE(1,N) = XLE(1,N+1)+DT*(A21(N+2)*XLE(2,N+2)
     1   + A41(N+2)*XLE(4,N+2) + A21(N+1)*XLE(2,N+1)
     2   + A41(N+1)*XLE(4,N+1))
      XLE(2,N) = XLE(2,N+1)+DT*(XLE(1,N+2) + XLE(1,N+1))
      XLE(3,N) = XLE(3,N+1)+DT*(A23(N+2)*XLE(2,N+2)
     1   + A43(N+2)*XLE(4,N+2) + A23(N+1)*XLE(2,N+1)
     2   + A43(N+1)*XLE(4,N+1))
      XLE(4,N) = XLE(4,N+1)+DT*(XLE(3,N+2) + XLE(3,N+1))
      M = 1
   23 XLE(1,N) = XLE(1,N+1) + DT*(A21(N)*XLE(2,N) +A41(N)*
     1   XLE(4,N)+A21(N+1)*XLE(2,N+1) +A41(N+1)*XLE(4,N+1))
      XLE(2,N) = XLE(2,N+1) + DT*(XLE(1,N) + XLE(1,N+1))
      XLE(3,N) = XLE(3,N+1) + DT*(A23(N)*XLE(2,N) + A43(N)*
     1   XLE(4,N)+A23(N+1)*XLE(2,N+1) +A43(N+1)*XLE(4,N+1))
      XLE(4,N) = XLE(4,N+1) + DT*(XLE(3,N) + XLE(3,N+1))
      M = M + 1
      IF(M-8)23,23,60
60    CONTINUE
C     TRAPIZOIDAL RULE IS USED FOR INTREGTRATION
      DIMENSION Q(4,3)
      Q(I,1) = 0.0
      Q(I,2) = 0.0
      Q(I,3) = 0.0
      DO 110 K = 1,60
      FT = (T(K+1) - T(K))*0.5
      Q(I,1) = Q(I,1)+(XLE(2,K) +XLE(2,K+1))*FT
      Q(I,2) = Q(I,2)+(XLE(4,K) +XLE(4,K+1))*FT
      Q(I,3) = Q(I,3)+(B23(K)*XLE(2,K)+B43(K)*XLE(4,K) +
     1   B23(K+1)*XLE(2,K+1) + B43(K+1)*XLE(4,K+1))*FT
110   CONTINUE
100   CONTINUE
      DIMENSION ER(3,3),C(4,3) ,D(4,4)
      READ 61,((ER(I,J),I=1,3),J=1,3)
61    FORMAT (5E8.0/4E8.0)
      DO 70 I = 1,4
      DO 70 J = 1,3
      C(I,J) = 0.0
      DO 70 K = 1,3
      C(I,J) = C(I,J) + Q(I,K)*ER(K,J)
70    CONTINUE
      DO 80 I = 1,4
```

```
      DO 80 J = 1,4
      D(I,J) = 0.0
      DO 80 K = 1,3
      D(I,J) = D(I,J) + C(I,K)* Q(J,K)
 80   CONTINUE
      PRINT 201, ((Q(I,J), J = 1,3), I = 1,4)
      PRINT 202, ((D(I,J), J = 1,4), I = 1,4)
      PRINT 203,((ER(I,J), J = 1,3), I = 1,3)
 201  FORMAT(1H1,38H TRANSFORM MATRIX Q BY ADJOINT METHOD /
     1   ///3(2X,E15.8,3X))
 202  FORMAT(////38H COVARIENCE MATRIX OF STATE VECTOR    .
     1   24H ERROR BY ADJOINT METHOD///3(2X,E15.8,3X))
 203  FORMAT(////38H COVARIENCE MATRIX OF SOURCE VECTOR   .
     1   6H ERROR///3(2X,E15.8,3X))
      CALL EXIT
      END
$ENTRY
      DATA CARDS
 +.0E+00 +.0E+00 +.0E+00+.25E-10
 +.1E-07 +.0E+00 +.0E+00 +.0E+00 +.1E-07
$IBSYS
```

NOTE-1 DATA CARDS INCLUDES THE TIME RECORD OF Y, Z, A1 AND A2 WHICH IS LISTED IN THE BEGINNING OF APPENDIX-I.

APPENDIX II

PROGRAMMING FOR EXAMPLE II

A PROGRAM FOR DIRECT METHOD

```
$IBFTC MAIN
C     *     *     *     *     *     *     *     *     *     *
      DIMENSION E(6),P(2,6)
      DO 10 I = 1,6
      DO 20 J = 1,6
      IF(I-J)1,2,1
 2    E(J) = 1.0
      GO TO  20
 1    E(J) = 0.0
 20   CONTINUE
      T = 0.0
      X1= 0.0
      X2= 0.0
      H = 0.001
 50   T = T + H
      AK1 = (E(2) + E(3) + E(5) + E(6) + X2)*H
      BK1 = (E(1) + E(3) + E(4) + E(6) - X2)*H
      AK2 = (E(2) + E(3) + E(5) + E(6) + X2 + 0.5*BK1)*H
      BK2 = (E(1) + E(3) + E(4) + E(6) - X2 - 0.5*BK1)*H
      AK3 = (E(2) + E(3) + E(5) + E(6) + X2 + 0.5*BK2)*H
      BK3 = (E(1) + E(3) + E(4) + E(6) - X2 - 0.5*BK2)*H
      AK4 = (E(2) + E(3) + E(5) + E(6) + X2 + BK3)*H
      BK4 = (E(1) + E(3) + E(4) + E(6) - X2 - BK3)*H
      X1 = X1 + (AK1 + 2.*AK2 + 2.*AK3 + AK4)/6.0
      X2 = X2 + (BK1 + 2.*BK2 + 2.*BK3 + BK4)/6.0
      IF(T-49.9999)50,50,51
 51   P (1,I) = X1
      P (2,I) = X2
   10 CONTINUE
      PRINT 100, (P(1,I), I=1,6), (P(2,I),I=1,6)
 100  FORMAT(1H1,38H TRANSFORM MATRIX BY DIRECT METHOD     /
     1   //6(2X,E15.8,3X))
C     COVARIENCE MATRIX OF STATE VECTOR ERROR
      DIMENSION )(2,2),C(2,6),ER(6,6)
      READ 200,((ER(I,J),I=1,6),J=1,6)
 200  FORMAT (12F5.0)
      DO 70 I = 1,2
      DO 70 J = 1,6
      C(I,J) = 0.0
      DO 70 K = 1,6
```

```
      C(I,J) = C(I,J) + P(I,K)*ER(K,J)
 70   CONTINUE
      DO 80 I = 1,2
      DO 80 J = 1,2
      D(I,J) = 0.0
      DO 80 K = 1,6
      D(I,J) = D(I,J) + C(I,K)*P(J,K)
 80   CONTINUE
      PRINT 300, ((D(I,J),J=1,2),I=1,2)
 300  FORMAT(////38H COVARIENCE MATRIX OF STATE VECTOR     ,
     1   23H ERROR BY DIRECT METHOD///2(2X,E15.8,3X))
      CALL EXIT
      END
$ENTRY
 0.1  0.0  0.0  0.0  0.0  0.0  0.0 0.05  0.0  0.0  0.0  0.0
  0.0 0.0 0.01  0.0  0.0  0.0  0.0  0.0  0.0 0.2   0.0  0.0
  0.0 0.0  0.0  0.0 0.06  0.0  0.0  0.0  0.0  0.0  0.0 0.04
$IBSYS
```

A PROGRAM FOR ADJOINT METHOD

```
$IBFTC MAIN
C     *     *     *     *     *     *     *     *     *     *
      DIMENSION  XLEMDA(2)  ,Q(2,6)
      DO 60 I = 1,2
      DO 50 J = 1,2
      IF(I-J)11,12,11
 12   XLEMDA(J) = 1.0
      GO TO 50
 11   XLEMDA(J)   = 0.0
 50   CONTINUE
      T = 50.0
      H =0.001
      U = XLEMDA(1)
      V = XLEMDA(2)
      FUNY(V) = (U-V)*H
      Q(I,1) = V*H*0.5
 75   T = T - H
      CK1 = FUNY(V)
      CK2 = FUNY(V + CK1*0.5)
      CK3 = FUNY(V + CK2*0.5)
      CK4 = FUNY(V + CK3 )
      V = V + (CK1 + 2.*CK2 + 2.*CK3 + CK4)/6.0
      IF (T-H)81,82,82
 82   Q(I,1) = Q(I,1) + V*H
      GO TO 75
 81   Q(I,1) = Q(I,1) + V*H*0.5
      Q(I,2) = U*50.00
      Q(I,3) = U*50.00 + Q(I,1)
      Q(I,4) = Q(I,1)
      Q(I,5) = Q(I,2)
      Q(I,6) = Q(I,3)
 60   CONTINUE
      PRINT 100, (Q(1,I), I=1,6), (Q(2,I),I=1,6)
 100  FORMAT(1H1,39H TRANSFORM MATRIX BY ADJOINT METHOD   /
     1   ///6(2X,E15.8,3X))
C     COVARIENCE MATRIX OF STATE VECTOR ERROR
      DIMENSION D(2,2),C(2,6),ER(6,6)
      READ 200,((ER(I,J),I=1,6),J=1,6)
 200  FORMAT (12F5.0)
      DO 70 I = 1,2
      DO 70 J = 1,6
      C(I,J) = 0.0
      DO 70 K = 1,6
      C(I,J) = C(I,J) + Q(I,K)*ER(K,J)
```

```
 70    CONTINUE
       DO 80 I = 1,2
       DO 80 J = 1,2
       D(I,J) = 0.0
       DO 80 K = 1,6
       D(I,J) = D(I,J) + C(I,K)*Q(J,K)
 80    CONTINUE
       PRINT 300, ((D(I,J),J=1,2),I=1,2)
 300   FORMAT(////38H COVARIENCE MATRIX OF STATE VECTOR
      1   24H ERROR BY ADJOINT METHOD////6(2X,E15.8,3X))
       CALL EXIT
       END
$ENTRY
 0.1  0.0  0.0  0.0  0.0  0.0  0.0 0.05 0.0  0.0  0.0  0.0
  0.0 0.0 0.01  0.0  0.0  0.0  0.0  0.0 0.0 0.2  0.0  0.0
  0.0 0.0  0.0  0.0 0.06  0.0  0.0  0.0 0.0  0.0  0.0 0.04
$IBSYS
```