Covariance Analysis Algorithm for Interconnected Systems

Robert D. Curley*
Integrated Systems, Inc., Santa Clara, California 95054
and
Victor H. L. Cheng†
NASA Ames Research Center, Moffett Field, California 94035
and
Ching-An Lin‡
National Chiao Tung University, Hsinchu, Taiwan 30043, Republic of China

We present in this paper a covariance analysis algorithm for propagation of signal statistics in arbitrarily interconnected nonlinear hybrid systems, applied to six-degree-of-freedom systems specifically in our work. The algorithm uses statistical linearization theory to linearize the nonlinear subsystems, and the resulting linearized subsystems are considered in the original interconnection framework for propagation of the signal statistics. We shall refer to some nonlinearities commonly encountered in six-degree-of-freedom space-vehicle models to illustrate the limitations of this method, along with problems not encountered in standard deterministic simulation analysis. Moreover, we shall exhibit the performance of the algorithm numerically by comparing results from such techniques to results from Monte-Carlo analysis results, both applied to a simple two-dimensional space-intercept problem.

I. Introduction

A GENERAL nonlinear continuous-time time-invariant dynamical system is modeled as

$$\dot{x} = f_1(x,u) \tag{1a}$$

$$y = f_2(x,u) \tag{1b}$$

where $u:\mathbb{R} \to \mathbb{R}^{ni}$ is the input as a function of time, $x:\mathbb{R} \to \mathbb{R}^{ns}$ is the state, $y:\mathbb{R} \to \mathbb{R}^{no}$ is the output, and $f_1:\mathbb{R}^{ns} \times \mathbb{R}^{ni} \to \mathbb{R}^{ns}$ and $f_2:\mathbb{R}^{ns} \times \mathbb{R}^{ni} \to \mathbb{R}^{no}$ are the two respective functions defining the state and output equations. When a system is memoryless, i.e., no state x is modeled, the output is related to the input simply by

$$y = f_2(u) \tag{2}$$

As a special case, a linear dynamical system would have the (f_1, f_2) representation simplified to the (A,B,C,D) representation, as in

$$\dot{x} = Ax + Bu \tag{3a}$$

$$y = Cx + Du \tag{3b}$$

where A, B, C, D are matrices of appropriate dimensions, possibly time varying.

Any stochastic variable can be partitioned into its mean (denoted by an overhead bar) and a zero-mean random component (denoted by a tilde), e.g., the vector x can be decomposed into

$$x = \bar{x} + \tilde{x} \tag{4}$$

where the mean of x is given by the expected value

$$\bar{x} = Ex \tag{5}$$

The covariance of x represents the second moment and is defined as

$$E\tilde{x}\tilde{x}^T \tag{6}$$

These first two moments are particularly useful in the analysis of stochastic processes because they are sufficient to completely characterize the statistical properties of linear stochastic processes involving variables with Gaussian distributions. Consequently, the assumption of Gaussian-distributed variables is often invoked when performing system analysis, which may otherwise be impossible or at best extremely difficult to carry out. Such an assumption is often justified through the Central Limit Theorem, but caution should be taken to avoid abusing the theory when the assumption cannot be justified.

Propagation of the first two moments of the state vector x of a linear system described by the (A,B,C,D) representation is rather straightforward. Specifically, square-root algorithms are available to provide efficient propagation of the covariance matrix with improved accuracy over direct propagation.[1] For a nonlinear system, however, the problem is not as trivial. One possible solution—the one we advocate in the current program—is to approximate the nonlinear system by a linearized one. The statistical linearization technique that is used takes into account the statistical properties of the input and state variables involved, including the covariance between them.[2-6] The resulting linearized model actually contains two parts: a representation $(\bar{A},\bar{B},\bar{C},\bar{D})$ that relates the mean quantities, and a representation $(\tilde{A},\tilde{B},\tilde{C},\tilde{D})$ for the zero-mean components and, consequently, for propagation of the covariance matrix. It should be observed that even when the original nonlinear system is time-invariant, the resulting linearized model is, in general, time-varying.

The basic methodologies of statistical linearization and covariance propagation are essentially the same as those employed in the Covariance Analysis Describing Function Technique (CADET).[5] The current covariance analysis algorithm development effort can be perceived as an extension of CADET, most notably in the sense of implementation. As its name implies, CADET is a technique, and computer imple-

Presented as Paper 87-2583 at the AIAA Guidance, Navigation, and Control Conference, Monterey, CA, Aug. 17-19, 1987; received Nov. 3, 1988; revision received Oct. 24, 1989.

*Research Scientist, Advanced Systems Division.
†Research Scientist, Aircraft Guidance and Navigation Branch; formerly, Senior Research Scientist, Integrated Systems, Inc.
‡Associate Professor, Department of Control Engineering; formerly, Research Scientist, Integrated Systems, Inc. Member AIAA.

mentations of it have been limited to special software developed for particular examples involving only single-input nonlinearities. On the other hand, the current covariance analysis program is meant to be a complete package applicable to a large class of interconnected, multi-input, nonlinear hybrid (i.e., continuous-time and discrete-time) systems, requiring minimal effort by the users to implement their system models. In addition, square-root algorithms are used to propagate the statistics with improved efficiency and accuracy.

The rest of this paper is organized as follows: the statistical linearization results in Ref. 2 are summarized in Sec. II; Sec. III formulates the linearized system interconnection and statistics propagation; the covariance analysis algorithm resulting from the formulations of Secs. II and III is presented in Sec. IV; Sec. V illustrates some limitations and peculiarities of the algorithm; Sec. VI contains a simple space-intercept example, with which the covariance analysis algorithm is compared to Monte-Carlo analysis, followed by some concluding remarks in Sec. VII.

II. Statistical Linearization Results

The objective of statistical linearization is to find equivalent linear functions for the nonlinear functions $f_1(\cdot,\cdot)$ and $f_2(\cdot,\cdot)$ in Eq. (1), or for $f_2(\cdot)$ in Eq. (2). The statistical linearization process can, thus, be formulated for a general nonlinear function $f: \mathbb{R}^n \to \mathbb{R}^m$, where n and m are positive integers. Let ξ be a vector of random variables with mean $\bar{\xi} := E(\xi)$ and covariance matrix $P := E(\xi-\bar{\xi})(\xi-\bar{\xi})^T$. Define $\tilde{\xi} := (\xi-\bar{\xi})$ as the zero-mean random part of ξ. The statistical linearization process seeks equivalent gain matrices $\bar{N}_f$ and $\tilde{N}_f$ of size $m \times n$ to approximate the nonlinear function f, such that the mean square of approximation error is minimized, i.e.,

$$\min_{\bar{N}_f,\tilde{N}_f} E[f(\xi)-\bar{N}_f\bar{\xi}-\tilde{N}_f\tilde{\xi}]^T[f(\xi)-\bar{N}_f\bar{\xi}-\tilde{N}_f\tilde{\xi}] \tag{7}$$

The resulting matrices $\bar{N}_f$ and $\tilde{N}_f$ are called statistical equivalent gains.

Note that for the general state equation

$$\dot{x} = f_1(x,u) \tag{8a}$$

the linearization process would yield, as an approximation,

$$\dot{\bar{x}} \approx \bar{N}_{f1}\begin{bmatrix}\bar{x}\\ \bar{u}\end{bmatrix} = [\bar{A}\ \bar{B}]\begin{bmatrix}\bar{x}\\ \bar{u}\end{bmatrix} = \bar{A}\bar{x}+\bar{B}\bar{u}$$

where the matrix $\bar{N}_{f1}$ has been partitioned into $\bar{A}$ and $\bar{B}$. Similarly, $\tilde{N}_{f1}$ can be partitioned into submatrices $\tilde{A}$ and $\tilde{B}$ to give

$$\dot{\tilde{x}} \approx \tilde{A}\tilde{x}+\tilde{B}\tilde{u} \tag{8b}$$

Although the foregoing discussion is centered around continuous-time systems, its extension to the discrete-time case is straightfoward and is omitted here.

It is immediately obvious that the expected value in Eq. (7) is equivalent to the sum of the expected values for the m individual squared terms. Consequently, the minimization problem can be separately studied as m simpler minimization problems; hence, without loss of generality, we only need to consider the case with $m=1$, and so $\bar{N}_f$ and $\tilde{N}_f$ are $1 \times n$ matrices. Formulas for these matrices have been derived in Ref. 2, indicating their dependence on $\bar{\xi}$ and P as follows:

$$\bar{N}_f = \frac{\bar{\xi}^T E f(\xi)}{\bar{\xi}^T\bar{\xi}} \tag{9}$$

$$\tilde{N}_f = E[\tilde{\xi}^T f(\xi)]P^{-1} \tag{10}$$

The expected values in Eqs. (9) and (10) can be obtained using the usual definition since the joint probability density of the stochastic vector ξ is completely characterized by $\bar{\xi}$ and P.

Such statistically linearized gains for a class of commonly encountered nonlinearities, primarily piecewise-linear odd functions, have been tabulated in the form of random-input describing functions.[4,6]

III. System Interconnection and Statistics Propagation

In this section, we shall first formulate the system equations for an arbitrary interconnection of N linear time-invariant finite-dimensional systems, and then apply this formulation to the propagation of statistics in the statistically linearized system structure. For simplicity, we only consider the continuous-time case in this section. Let u_i, x_i, y_i be, respectively, the input, state, and output vectors of the ith subsystem, and let (A_i,B_i,C_i,D_i) be a representation of the subsystem. Note that if the ith system is memoryless, then A_i, B_i, and C_i do not exist. If we form the composite vectors

$$u = \begin{bmatrix}u_1\\ u_2\\ \cdot\\ \cdot\\ \cdot\\ u_N\end{bmatrix}, \quad x = \begin{bmatrix}x_1\\ x_2\\ \cdot\\ \cdot\\ \cdot\\ x_N\end{bmatrix}, \quad y = \begin{bmatrix}y_1\\ y_2\\ \cdot\\ \cdot\\ \cdot\\ y_N\end{bmatrix} \tag{11}$$

then we can describe the N systems in parallel in Eq. (3), where A, B, C, D are obtained by combining the collection of A_i, B_i, C_i, D_i matrices, appropriately. It is easy to see that A and D would be block-diagonal matrices obtained from the collection of A_i and D_i, respectively. In addition, B and C also appear to be block-diagonal, except that when the ith block is memoryless, the corresponding rows of B and columns of C are absent.

Our representation of an interconnected system involves the property that every input component of a subsystem is either connected to an output component of some subsystem or an exogenous input signal. Such interconnected systems would certainly include feedback loops as a possibility, but standard assumptions are made to ensure that any such loops would not cause inconsistency algebraically. Based on this representation, the composite input vector u can be expressed as a function of the composite output vector y and the exogenous input vector y_0 as

$$u = Ky + K_0y_0 \tag{12}$$

where

K = interconnection matrix relating subsystem outputs to inputs
K_0 = exogenous input connection matrix

Combining Eq. (12) with Eq. (3b) leads to the output equation

$$y = Cx + D(Ky + K_0y_0) \tag{13}$$

Solving explicity for the outputs in terms of the state and exogenous input yields

$$y = (I-DK)^{-1}Cx + (I-DK)^{-1}DK_0y_0 \tag{14}$$

Eq. (14) can be substituted back into Eq. (12) to determine the input vector

$$u = K[(I-DK)^{-1}Cx + (I-DK)^{-1}DK_0y_0] + K_0y_0 \tag{15}$$

which, upon rearranging and using the relation

$$(I-KD)^{-1} = [K(I-DK)^{-1}D + I]$$

yields

$$u = K(I-DK)^{-1}Cx + (I-KD)^{-1}K_0y_0 \tag{16}$$

Equations (14) and (16) provide the means for computing the subsystem output and input composite vectors, given the composite state and exogenous input vectors. The state equation in (3a) can also be expressed in terms of the composite state vector and exogenous input vector in view of Eq. (16), leading to

$$\dot{x} = [A + BK(I - DK)^{-1}C]x + B(I - KD)^{-1}K_0 y_0 \quad (17)$$

By defining the closed-loop matrices

$$A_{cl} = A + BK(I - DK)^{-1}C \quad (18a)$$

$$B_{cl} = B(I - KD)^{-1}K_0 \quad (18b)$$

$$C_{cl} = (I - DK)^{-1}C \quad (18c)$$

$$D_{cl} = (I - DK)^{-1}DK_0 \quad (18d)$$

the interconnected system is given by Eqs. (17) and (14) in the abbreviated form:

$$\dot{x} = A_{cl}x + B_{cl}y_0 \quad (19a)$$

$$y = C_{cl}x + D_{cl}y_0 \quad (19b)$$

In contrast to deterministic simulations, Eq. (16) is required in the covariance analysis algorithm to provide the means for computing the input statistics to each subsystem, which is, in turn, required for statistical linearization of the subsystem.

Now consider the case of N interconnected nonlinear subsystems. With the input and state statistics known for the ith subsystem, we can perform statistical linearization on it to obtain the two representations $(\bar{A}_i,\bar{B}_i,\bar{C}_i,\bar{D}_i)$ and $(\tilde{A}_i,\tilde{B}_i,\tilde{C}_i,\tilde{D}_i)$ as outlined in Sec. II. Then the composite vectors can be formed as in Eq. (11), with the corresponding composite matrices $(\bar{A},\bar{B},\bar{C},\bar{D})$ and $(\tilde{A},\tilde{B},\tilde{C},\tilde{D})$. With these matrices, the derivation in Eqs. (12–19) follows, except in this case there is a set of equations for the mean terms and another set for the zero-mean random terms. Let

$m_I = \bar{u}$ = composite-input mean vector
$m_S = \bar{x}$ = composite-state mean vector
$m_O = \bar{y}$ = composite-output mean vector
$m_X = \bar{y}_0$ = exogenous-input mean vector
$\Sigma_I = E[\tilde{u}\tilde{u}^T]$ = composite-input covariance matrix
$\Sigma_S = E[\tilde{x}\tilde{x}^T]$ = composite-state covariance matrix
$\Sigma_O = E[\tilde{y}\tilde{y}^T]$ = composite-output covariance matrix
$\Sigma_X = E[\tilde{y}_0\tilde{y}_0^T]$ = exogenous-input covariance matrix
$\Sigma_{IS} = E[\tilde{u}\tilde{x}^T]$ = covariance matrix between composite input and state

Then the output statistics can be determined from Eq. (19b) by

$$m_O = \bar{C}_{cl}m_S + \bar{D}_{cl}m_X \quad (20a)$$

$$\Sigma_O = \tilde{C}_{cl}\Sigma_S\tilde{C}_{cl}^T + \tilde{D}_{cl}\Sigma_X\tilde{D}_{cl}^T \quad (20b)$$

since the composite-state vector x is assumed to be uncorrelated with the exogenous-input vector y_0.

For improved computational efficiency and accuracy, the lower-triangular square-root of Σ_O can be computed by performing an orthonormal transformation with column operations to reduce the following composite matrix to lower-triangular form.[1,7]

$$[\tilde{C}_{cl}\Sigma_S^{1/2} \quad \tilde{D}_{cl}\Sigma_X^{1/2}]$$

where $\Sigma^{1/2}$ represents a matrix square-root of Σ.

The input statistics can be determined from Eq. (16) in a similar manner

$$m_I = \bar{K}_{IS}m_S + \bar{K}_{IX}m_X \quad (21a)$$

$$\Sigma_I = \tilde{K}_{IS}\Sigma_S\tilde{K}_{IS}^T + \tilde{K}_{IS}\Sigma_X\tilde{K}_{IS}^T \quad (21b)$$

where

$$K_{IS} = K(I - DK)^{-1}C, \quad K_{IX} = (I - KD)^{-1}K_0 \quad (22)$$

are both defined for both the mean and zero-mean random components. The lower-triangular square-root of Σ_I can be computed similar to the output term. Furthermore, the statistical linearization will also require the covariance term Σ_{IS}, which can be determined from Eq. (16) as

$$\Sigma_{IS} = \tilde{K}_{IS}\Sigma_S \quad (23)$$

Propagation of the state statistics using Eq. (19a) directly would involve the differential Riccati equation, which prohibits the use of the square-root algorithm. However, Eq. (19a) can be discretized first to obtain the algebraic equa-

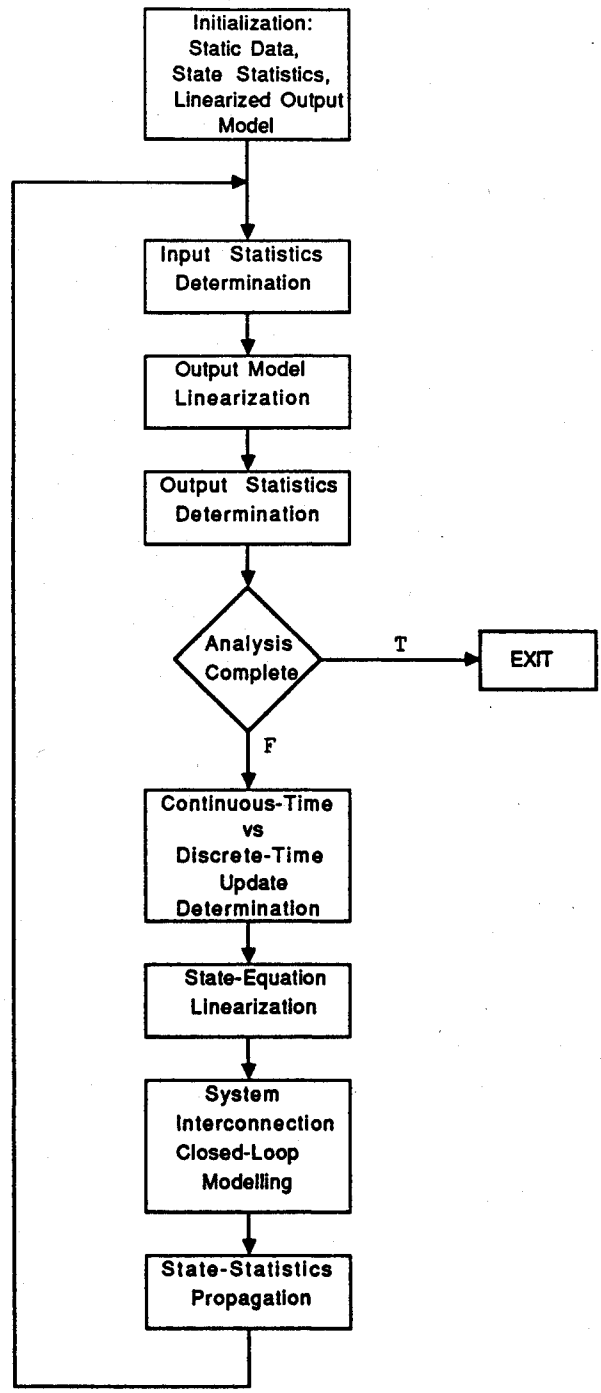

Fig. 1 Covariance-analysis algorithm flowchart.

tion for state transition. This involves computation of the discrete-time matrices

$$A_d = e^{A_{cl}\Delta t} \tag{24a}$$

$$B_d = \left(\int_0^{\Delta t} e^{A_{cl}\tau}\, d\tau\right) B_{cl} \tag{24b}$$

for the mean and zero-mean random components, where Δt is the integration stepsize, which is not necessarily constant. Propagation of the state statistics thus involve

$$m_S^+ = \bar{A}_d m_S^- + \bar{B}_d m_X \tag{25a}$$

$$\Sigma_S^+ = \tilde{A}_d \Sigma_S^- \tilde{A}_d^T + \tilde{B}_d \Sigma_X \tilde{B}_d^T \tag{25b}$$

In both of the previous equations, the superscripts + and − denote quantities after and before the update, respectively. With Eq. (25b), the lower-triangular square-root of Σ_S can be propagated with the orthonormal transformation approach as for the output and input computations.

This integration approach using the state-transition matrix A_d assumes that the dynamics prescribed by A_{cl} is constant over the integration step, but the integration of the state statistics is otherwise exact. Validity of this approach vs the Runge-Kutta numerical integration recommended in CADET[5] depends on the system model and integration step size. We believe that the assumption of A_{cl} being constant over Δt is not a serious drawback; furthermore, if there is any disadvantage compared to the Runge-Kutta technique at all, it can be more than offset by the numerical superiority of the square-root algorithms.

IV. Covariance Analysis Algorithm

The formulations discussed in the last two sections have been implemented in a computer program, as depicted by the flowchart in Fig. 1. The user provides subsystem-model and interconnection information in the form of user-defined program modules that are to be called by the covariance analysis program. We shall address in the rest of this section the different steps involved in the covariance analysis program.

A. Initialization

The initialization step involves collection of static information for subsystem identification and interconnection, so that it can set up the work space for propagating the signal statistics. In addition, it also performs the functions of initializing the state statistics and obtaining an initial linearized output model before control is passed to the main analysis loop.

B. Input-Statistics Determination

This step represents the entry point of the main analysis loop. Here the composite subsystem-input statistics are computed according to the expressions derived in Eqs. (21–23).

C. Output-Model Linearization

With the available input and state statistics, the statistical equivalent gains are computed for all the subsystem output models [Eqs. (1b) and (2)], using the formulas derived in Sec. II [Eqs. (9) and (10)].

D. Output-Statistics Determination

With the updated linearized output model, the composite output statistics can be computed according to Eq. (20), based on the state and exogenous-input statistics, and the closed-loop matrices of Eqs. (18c) and (18d). At this step, other statistical properties such as "spherical error probable" (SEP) can be calculated, and conditions for analysis termination can be evaluated to test for exiting from the loop.

E. Continuous-Time vs Discrete-Time Update Determination

For the analysis program to handle hybrid systems involving both continuous-time and discrete-time subsystems, it has to be able to perform both continuous-time (interval) and discrete-time (instant) updates. This step decides whether an interval update or an instant update is appropriate for the current iteration. If the simulation time instant is coincident with the sampling instant of one or more discrete-time subsystems, an instant update is called for; otherwise, an interval update is required. In the latter case, when an interval update is commanded, the program still has to check if the analysis can be performed for the whole nominal integration step-size without violating any sample instants of some discrete-time component. Any such violation would necessitate the reduction of the step-size to end at the first encountered sample instant.

F. State-Equation Linearization

The statistical linearization of each subsystem state equation depends on the following three factors:

1) whether an interval or instant update is commanded by the previous step;

2) whether the subsystem is continuous-time or discrete-time;

3) whether the simulation time instant coincides with the sampling instant of the subsystem for the case where an instant update is commanded and the subsystem is discrete-time.

When an interval update is commanded, all continuous-time subsystems would have their state equations [Eq. (1a)] linearized according to formulas derived in Sec. II, and every discrete-time subsystem should provide an equivalent continuous-time model that forces the state variables to remain unchanged through the integration step, i.e., $A_i = 0$, $B_i = 0$. For the other case, when an instant update is commanded, all discrete-time systems with a coincident sampling instant would have their state equations [the discrete-time version of Eq. (1a)] linearized, whereas any other subsystem (including discrete-time systems that do not have a coincident sampling instant) should provide an equivalent discrete-time model that forces the state variables to remain unchanged at the update instant, i.e., $A_i = I$, $B_i = 0$.

G. System Interconnection: Closed-Loop Modeling

The closed-loop matrices for state propagation are computed by using Eqs. (18a) and (18b). If the current update is of the interval type, the matrices have to be converted to the discrete-time versions in Eqs. (24a) and (24b).

H. State-Statistics Propagation

The state statistics are propagated forward according to Eqs. (25a) and (25b), using the exogenous-input statistics and the closed-loop matrices determined in the last step. This is the last function of the analysis loop, resuming with the input statistics determination step described previously.

V. Algorithm Limitations

The covariance analysis technique has been conceived as an efficient alternative to Monte-Carlo analysis for evaluation of stochastic dynamical systems. Although Monte-Carlo analysis may require many deterministic simulation runs to produce meaningful ensemble statistics, the covariance analysis technique can produce the first two statistical moments in a single run. Furthermore, the covariance analysis technique is exact when applied to linear system, in contrast to Monte-Carlo analysis, which is a numerical process that requires many repetitions for the results to converge. Such advantages, however, do not relieve a competent engineer from exercising prudent judgment as to whether the covariance analysis technique is appropriate for a certain application. We shall discuss in this section some of the limitations plaguing the covariance analysis technique. In particular, the following section ad-

dresses some conceivable difficulties of covariance analysis itself, and the other subsections describe some peculiarities stemming from the need for statistical linearization when nonlinear subsystems are present.

A. Curse of Dimensionality

Whereas covariance analysis can be performed with one computer run as opposed to Monte-Carlo analysis, which requires many runs, covariance analysis is not always the more efficient alternative. First, it is obvious that there is more effort involved in setting up the covariance analysis problem. A much more important limitation, however, is that covariance analysis will consume more computation time in one run than Monte-Carlo analysis for problems with high-order systems.

Since covariance analysis requires the propagation of the covariance matrices, its complexity increases with the square of the system dimension, whereas Monte-Carlo analysis has only a linear increase in complexity.[8] Consequently, although covariance analysis may be computationally more efficient than Monte-Carlo analysis with low-order systems, the property is bound to turn around when the order increases sufficiently. Zarchan[8] has indicated that, based on some simplistic arguments, the break-even point between the two techniques is a 100th-order system when 100 repetitions are assumed for the Monte-Carlo technique. (This is probably a good rule of thumb as to the limiting order of a system for considering covariance analysis as a technique for studying the system.)

For instance, a ninth-order extended Kalman filter (EKF) for target state estimation is quite often studied in intercept vehicles. Such a filter, in fact, has an order much higher than nine, since it also has to propagate a 9×9 covariance matrix. Even with symmetry considerations, the matrix elements still constitute a 45th-order component, boosting the order of the EKF alone to 54. The EKF formulation itself involves linearization functions. To convert this formulation to covariance analysis by using statistical linearization techniques would be unrealistically complex.

B. Multi-Input Nonlinearities

Reference 2 contains some simplifying techniques for computing the gains in Eqs. (9) and (10) when certain assumptions are satisfied. Moreover, in our study of space-intercept vehicles, there are not many multi-input nonlinearities that cannot be broken down into single-input ones. However, there are a couple of extremely common multi-input nonlinearities that are often taken for granted in deterministic simulations: multiplication and division. Multi-input statistical linearization has to be performed whenever a multiplication or division between signals is encountered, and there are many of these operations performed in the analysis of a six-degree-of-freedom model, e.g., matrix multiplications in coordinate transformations and matrix inversions in some filter designs.

C. Ill-Conditioned Nonlinearities

The form of linearization function adopted in the objective function of Eq. (7) will inevitably allow some nonlinearities to be more amenable to statistical linearization than others. For instance, it is difficult to statistically linearize functions that do not pass through the origin when the input signal has zero mean: e.g., the cosine function, which is used very often in transformation computations, when subject to the linearization formula in Eq. (9) produces a gain term

$$\bar{N}_{\cos} = \frac{1}{\bar{\xi}} \cos\bar{\xi} e^{\frac{-\sigma^2}{2}}$$

that diverges as the input mean $\bar{\xi}$ approaches zero. To implement the cosine nonlinearity, a phase shift by $\pi/2$ can first be added to the input mean and then the equivalent statistical gain is computed using formulas for the sine function. Solution of this problem via a phase shift is unique to this trigonometric function since a shifting back is not required. Other nonlinearities, such as step functions, cannot be evaluated for zero-mean conditions. As the variance approaches zero, the gain to the random component approaches zero, and the gain to the mean component approaches the value defined by the deterministic function normalized with respect to the mean. Finally, functions for which closed-form solutions do not exist must be evaluated using numerical integration techniques. Such functions include the reciprocal function $f(\xi)=1/\xi$ and the square-root function $f(\xi)=\sqrt{\xi}$.

There are also nonlinearities that do not fall within the statistical linearization framework of Secs. I and II: nonlinearities that have memory but do not lend themselves to finite-dimensional dynamical system modeling as Eq. (1). Nonlinearities of this kind include hysterisis and bistable switches. It is sometimes possible to model these nonlinearities as discrete-time feedback systems with appropriately defined memoryless nonlinearities.

It is clear that, due to the need for statistical linearization, many systems that are otherwise simple to implement in a simulation may require a substantial amount of effort in redefining them in terms of equivalent covariance analysis models.

D. Computational Algorithms

With the increasing availability of digital computers, modern flight vehicles inevitably are experiencing the incorporation of digital controllers with software that contain vast amounts of numerical computations and logic. Complicated algorithms that engineers feel comfortable in incorporating into digital computers may turn out to be infeasible for converting into a form suitable for statistical linearization and covariance analysis. Firstly, computer memory is inexpensive and it only takes, for instance, four bytes of memory to increase the order of the subsystem by one. Secondly, logic

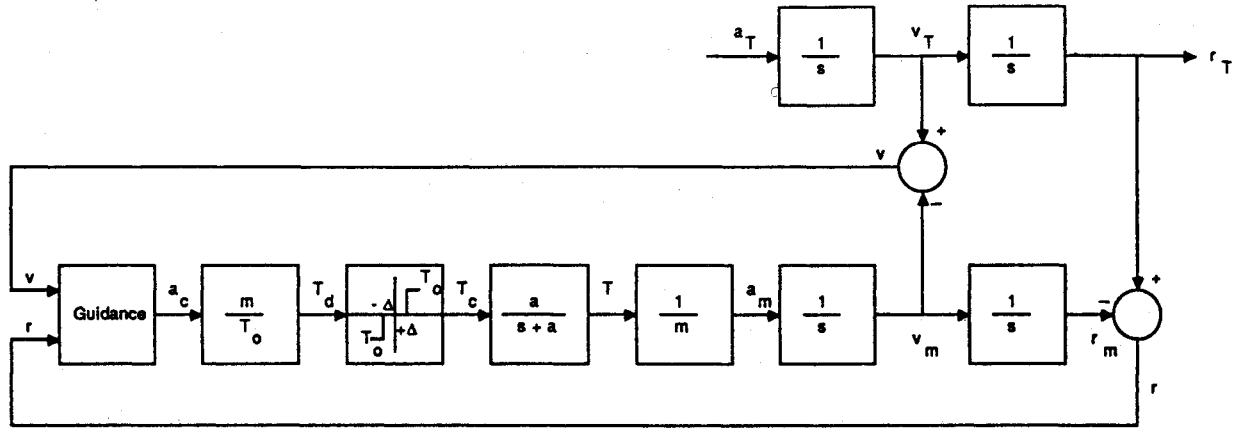

Fig. 2 Control system model.

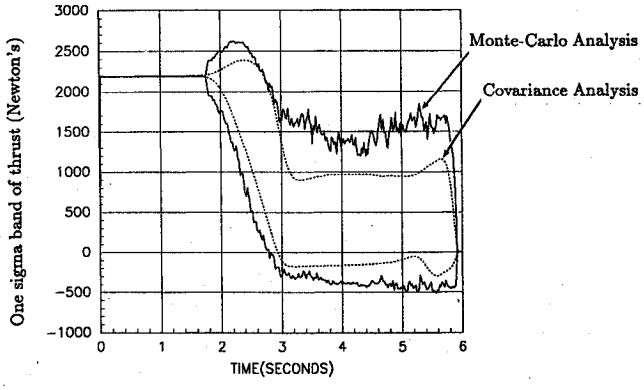

Fig. 3 One sigma band (i.e., mean ± standard deviation) of thrust (Newtons).

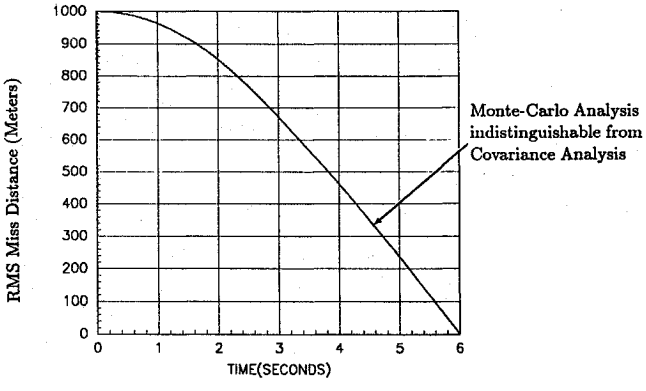

Fig. 4 RMS miss distance.

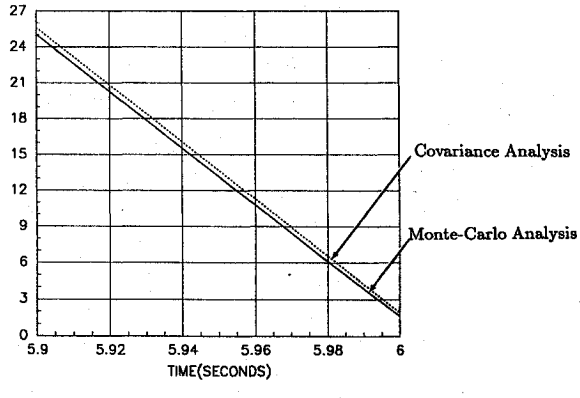

Fig. 5 RMS miss distance: final 0.1 s of engagement.

built into these controllers can rarely be modeled with simple nonlinear functions. Hence, the class of software algorithms tend to be inappropriate for covariance analysis evaluation.

VI. Example

A planar interceptor-target engagement model was used to study this covariance analysis algorithm with comparison to Monte-Carlo analysis results for the same engagement. Control is available only in one direction, given by thrust T, and there is no attitude control. The block diagram implementing the simulation model for this example is shown in Fig. 2. This system was modeled using four subsystems with five states (assuming constant mass of the interceptor): interceptor thrust T, interceptor position r_m, interceptor velocity v_m, target position r_T, and target velocity v_T. The only nonlinearity in this example is the thruster level on-off control, which was modeled as a relay with dead-zone. White Gaussian noise is superimposed upon a constant acceleration level (to represent the measured target accelertion a_T), which is the only external input to the system.

A Monte-Carlo analysis of the same system was performed for an ensemble size of 200 simulations. Both the covariance analysis and Monte-Carlo analysis used integration stepsizes of 5 ms for an engagement duration of 6 s. The equations for each deterministic simulation comprising the Monte-Carlo analysis were integrated using a fourth-order Runge-Kutta method, with statistics accumulated at the end of each simulation. Figure 3 shows the one-sigma band (mean ± standard deviation) of the thrust for the two analyses. Figures 4 and 5 compare the computed rms miss distance over the entire interval and over the final 0.1 s, respectively. The rms miss distance results indeed show positively that the covariance analysis technique can be an efficient alternative to Monte-Carlo analysis, if applied carefully.

VII. Conclusion

We have presented in this paper a covariance analysis algorithm that uses statistical analysis techniques for the study of arbitrarily interconnected nonlinear systems, which can include both continuous-time and discrete-time subsystems. We have discussed in much detail the implementation of the algorithm. Specifically, statistical linearization has to be performed on each subsystem. Interconnection of all the linearized subsystems leads to a closed-loop model, with which the first two statistical moments of the variables are propagated. We are using square-root algorithms and state-transition matrices for integration to achieve good numerical behavior in the statistics propagation.

We have also made a fair effort in identifying the limitations of the algorithm. In particular, although the algorithm can be a more efficient alternative to the Monte-Carlo technique for studying stochastic systems, such is not the case for high-order systems. We recommend that users restrict the system order to not much higher than 50 with this technique. Furthermore, potential users should familiarize themselves with the other limitations discussed in the paper, whereas some nonlinearities may render the algorithm cumbersome to use, others may actually cause numerical difficulties.

Acknowledgment

The work here described was supported by the U.S. Air Force Armament Laboratory, Eglin Air Force Base, Florida, under Contract F08635-85-C-0249.

References

[1]Bierman, G. J., *Factorization Methods for Discrete Sequential Estimation,* Academic, New York, 1977.

[2]Lin, C. A., and Cheng, V. H. L., "Statistical Linearization Based Formulation of Covariance Analysis for Interconnected Nonlinear Systems," Integrated Systems, Inc., Santa Clara, CA, TM 5529-01, Jan. 1986; prepared under Contract F08635-85-C-0249, with USAF Armament Lab., Eglin AFB, FL.

[3]Cheng, V. H. L., and Lin, C. A., "Top-Level Algorithm and Preliminary Database Design for Covariance Analysis Tool," Integrated Systems, Inc., Santa Clara, CA, TM 5529-02, 1986; prepared under Contract F08635-85-C-0249, with USAF Armament Lab., Eglin AFB, FL.

[4]Gelb, A., and Vander Velde, W. E., *Multiple-Input Describing Functions and Nonlinear System Design,* McGraw-Hill, New York, 1968.

[5]Gelb, A., and Warren, R. S., "Direct Statistical Analysis of Nonlinear Systems—CADET," AIAA Paper 72-875, Aug., 1972.

[6]Taylor, J. H., "Random-Input Describing Functions for Multi-Input Nonlinearities," *International Journal of Control,* Vol. 23, No. 2, 1976, pp. 277–281.

[7]Golub, G. H., and Van Loan, C. F., *Matrix Computations,* Johns Hopkins Univ. Press, Baltimore, MD, 1983.

[8]Zarchan, P., "Efficient Computerized Methods of Statistical Analysis," *Proceedings of the 10th Annual Pittsburgh Conference, Modelling and Simulation, Pt. 2: Systems and Control,* Vol. 10, Pittsburg, PA, April 25–27, 1979, pp. 577–581.