SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

1

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/CI/NR 86- 148D	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Modeling Continuous-Time Random Processes In Digital Computer Simulations of Physical Systems		5. TYPE OF REPORT & PERIOD COVERED /~~THESIS~~/DISSERTATION 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Tom Lance Riggs, Jr.		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS AFIT STUDENT AT: Auburn University		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE 1986 13. NUMBER OF PAGES 100
MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASS 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

DTIC
ELECTE
SEP 17 1986
S D
B

DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

APPROVED FOR PUBLIC RELEASE: IAW AFR 190-1

Lynn Wolaver 5 Sept 86
LYNN E. WOLAVER
Dean for Research and
Professional Development
AFIT/NR

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

ATTACHED ...

AD-A171 909

~~DTIC~~ FILE COPY

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

1118

DISSERTATION ABSTRACT

MODELING CONTINUOUS-TIME RANDOM PROCESSES IN DIGITAL COMPUTER SIMULATIONS OF PHYSICAL SYSTEMS

Tom Lance Riggs, Jr.

Doctor of Philosophy, August 27, 1986
(M.E.E., Auburn University, 1977)
(B.E.E., Auburn University, 1976)

111 Typed Pages

Directed by Charles L. Phillips

This dissertation addresses the problem of determining the correct relationship between the statistics of a continuous random process and the statistics of a discrete random process used to simulate the continuous random process. The findings of this research are directly applicable to the general field of digital simulation of physical systems described by ordinary differential equations.

It is shown that to ensure a faithful digital simulation of a continuous random process, the noise statistics of the random number generator must be set to values drastically different from the noise statistics of the continuous random process. Further, it is established that the

11/8

relationship between the continuous and discrete statistics is a function of the integration method used in the digital simulation.

The proper functional relationship between the discrete and continuous noise statistics is derived for

1. the class of Runge-Kutta integrators,
2. the 4th order Adams-Bashforth integrator, and
3. the Adams-Moulton corrector formula.

The derived relationships are applied to a specific problem and are demonstrated by simulation. The simulation results are compared to exact solutions. Additionally, the requirement for proper operation of a variable-step-size algorithm is developed.

MODELING CONTINUOUS-TIME RANDOM PROCESSES
IN DIGITAL COMPUTER SIMULATIONS
OF PHYSICAL SYSTEMS

Tom Lance Riggs, Jr.

A Dissertation
Submitted to
the Graduate Faculty of
Auburn University
in Partial Fulfillment of the
Requirements for the
Degree of
Doctor of Philosophy

Auburn, Alabama
August 27, 1986

148

MODELING CONTINUOUS-TIME RANDOM PROCESSES IN DIGITAL COMPUTER SIMULATIONS OF PHYSICAL SYSTEMS

Tom Lance Riggs, Jr.

Certificate of Approval:

Dallas W. Russell
Dallas W. Russell
Professor, Electrical Engineering

Charles L. Phillips
Charles L. Phillips, Chairman
Professor, Electrical Engineering

John E. Cochran Jr
John E. Cochran, Jr.
Professor, Aerospace Engineering

Joseph S. Boland, III
Joseph S. Boland, III
Professor, Electrical Engineering

Norman J. Doorenbos
Dean, Graduate School

VITA

Tom Lance Riggs, Jr., son of Tom Lance and Ferne Marie (Lowry), was born in Washington, D.C. on April 21, 1947. He attended Arlington County Public Schools and graduated from Washington-Lee High School, Arlington, Virginia, in 1965. He enlisted in the United States Air Force in 1966. After seven years of service as an electronics technician, the Air Force selected him for a college educational program and sent him to Auburn University to study electrical engineering. After completing his university education at Auburn (Bachelor of Electrical Engineering, March 1976 and Master of Electrical Engineering, June 1977), he was commissioned a 2nd Lieutenant in the United States Air Force. From 1977 to 1978, Lieutenant Riggs worked as a systems analyst and a research associate at the Guided Weapons Division of the Air Force Armament Laboratory. From 1978 to 1980, Lieutenant Riggs served as the Director for Research on the Advanced Guidance and Estimation for Tactical Missiles research program. In 1980, Lieutenant Riggs was assigned to the Department of Astronautics, United States Air Force Academy as an academic instructor. At the academy, he was promoted to the rank of Captain and achieved the position of Assistant Professor of Astronautical Engineering. In 1983, the Air Force selected Captain Riggs for a Doctoral program and sent him to the University of Texas at Austin to study aerospace engineering and then to Auburn University to complete his degree in electrical engineering. After graduation, Captain Riggs will return to the academy to serve on the Faculty. Captain Riggs married Patricia Frances, daughter of John William and Margaret May (Donnelson) Davis of Manchester, England, in December of 1969. They have three children, Dawn Michelle, Karen Nicole, and Kimberly Annette.

1118

DISSERTATION ABSTRACT

MODELING CONTINUOUS-TIME RANDOM PROCESSES IN DIGITAL COMPUTER SIMULATIONS OF PHYSICAL SYSTEMS

Tom Lance Riggs, Jr.

Doctor of Philosophy, August 27, 1986
(M.E.E., Auburn University, 1977)
(B.E.E., Auburn University, 1976)

111 Typed Pages

Directed by Charles L. Phillips

This dissertation addresses the problem of determining the correct relationship between the statistics of a continuous random process and the statistics of a discrete random process used to simulate the continuous random process. The findings of this research are directly applicable to the general field of digital simulation of physical systems described by ordinary differential equations.

It is shown that to ensure a faithful digital simulation of a continuous random process, the noise statistics of the random number generator must be set to values drastically different from the noise statistics of the continuous random process. Further, it is established that the

11/5

relationship between the continuous and discrete statistics is a function of the integration method used in the digital simulation.

The proper functional relationship between the discrete and continuous noise statistics is derived for

1. the class of Runge-Kutta integrators,
2. the 4th order Adams-Bashforth integrator, and
3. the Adams-Moulton corrector formula.

The derived relationships are applied to a specific problem and are demonstrated by simulation. The simulation results are compared to exact solutions. Additionally, the requirement for proper operation of a variable-step-size algorithm is developed.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

The design of modern technological systems is a well defined logical process involving many phases beginning with a statement of need and culminating in hardware. The success of the final product is critically dependent on the successful completion of each design phase. In each of the phases, analysis plays a fundamental role, whether the goal of the analysis is to test the potential payoff of a feasible solution, or simply to further the engineer's understanding of the problem. The importance of an accurate analysis, regardless of the design phase for which it is performed, can not be overemphasized.

To perform an accurate analysis, the analyst has a number of useful and time-proven tools. One of the most useful tools for analysis is based on the modern engineer's ability to describe a physical process using mathematical equations. The implementation of those equations forms a basis for a simulation of the physical system. That simulation allows the engineer to exercise the model of the physical system in a controlled manner. Of course, the accuracy of an analysis performed by simulation is directly dependent on the accuracy of the assumed mathematical model of the physical system.

Until recently, simulation was performed primarily on analog computers. Analog computers consist of a number of specialized electronic devices that respond to electrical inputs in a manner analogous to the response of physical devices to physical inputs. The interconnection of the analog computer devices in a way that represents the physical process leads to an analog simulation of the physical system. Once completed there will be a one-to-one correspondence between the variables describing the physical process and the variables of the analog simulation. This direct correspondence makes the analog simulation an excellent tool for studying the physical system without physically realizing that system. However, the initial cost, the cost of maintenance, the cost of operation and an inherent difficulty in reprogramming make analog computers unattractive.

As an alternative, modern digital computers offer easy reprogramming at relatively low costs. Additionally, digital computers effectively solve multiple problems simultaneously, do not have to be dedicated to simulation and provide the ability to store vast amounts of data. The combination of computational speed, programming flexibility, and sophisticated numerical algorithms for the solution of complex mathematical equations has made simulation via digital computers a common practice over the last fifteen years.

Many physical processes occur in the environment that can not be precisely modeled in a deterministic manner. Examples of such processes are wind gusts, electronic sensor noise, future target maneuvers, the weather and the economy. Although such processes can not be modeled deterministically, they can be modeled as continuous random processes based on statistical data. To model a physical process as a random process, one only needs an adequate statistical representation of the physical process. This representation can be derived from previous observation data or simply defined based on knowledge of the physical limits of the process. Although such a derived or defined model can not precisely predict the behavior of the physical system, it can predict the statistical behavior of the system or, in other words, how the system will behave "on the average". Modeling the non-deterministic system in this manner will provide useful information, much more so than simply ignoring the noisy process. Including such random models in a simulation will certainly enhance its accuracy by making it more faithful to the physical system.

Review of Random Variables

Before proceeding further, it will be useful to define and show a number of useful mathematical properties related to random variables. To do this, let $\underline{x}$ be an n-dimensional vector of random variables. All of the information known about $\underline{x}$ will be embodied in its probability density function (pdf) or, if the pdf is not available, in its probability distribution function.[1] For convenience, assume the pdf of $\underline{x}$ is available and let it be denoted by $f_{\underline{x}}(\cdot)$.

$f_{\underline{x}}(\cdot)$ can be used to compute the "expected value" of some function of $\underline{x}$, where the "expected value" is a numerical value obtained by averaging the outcomes of an experiment over an ensemble of trials. Specifically, if the m-dimensional vector function $\underline{z}(\cdot)$ is a function of $\underline{x}$ such that

$$\underline{z}(\cdot) = \underline{\theta}[\underline{x}(\cdot)]$$

where $\underline{\theta}(\cdot)$ is continuous, then the expectation of $\underline{z}$, denoted as $E[\underline{z}]$, is

$$E[\underline{z}] \triangleq \int_{-\infty}^{\infty} \underline{\theta}(\underline{\xi}) f_{\underline{x}}(\underline{\xi}) d\underline{\xi}$$

Since by definition, the expectation is an integral, it is a linear operation. Therefore, if

$$\underline{w}(\cdot) = A\underline{z}(\cdot) = A\underline{\theta}[\underline{x}(\cdot)]$$

where A is a constant matrix, then

$$E[\underline{w}] = E[A\underline{z}] = \int_{-\infty}^{\infty} A\underline{\theta}(\underline{\xi}) f_{\underline{x}}(\underline{\xi}) d\underline{\xi} = A\int_{-\infty}^{\infty} \underline{\theta}(\underline{\xi}) f_{\underline{x}}(\underline{\xi}) d\underline{\xi} = AE[\underline{z}] \qquad (1)$$

Additionally, if

$$\underline{w}(\cdot) = \underline{z}_1(\cdot) + \underline{z}_2(\cdot) = \underline{\theta}_1[\underline{x}(\cdot)] + \underline{\theta}_2[\underline{x}(\cdot)]$$

then

$$E[\underline{w}] = E[\underline{z}_1 + \underline{z}_2] = \int_{-\infty}^{\infty} [\underline{\theta}_1(\underline{\xi}) + \underline{\theta}_2(\xi)] f_{\underline{x}}(\underline{\xi}) d\underline{\xi}$$

$$= \int_{-\infty}^{\infty} \underline{\theta}_1(\underline{\xi}) f_{\underline{x}}(\underline{\xi}) d\underline{\xi} + \int_{-\infty}^{\infty} \underline{\theta}_2(\underline{\xi}) f_{\underline{x}}(\underline{\xi}) d\underline{\xi}$$

$$= E[\underline{z}_1] + E[\underline{z}_2] \qquad (2)$$

Particular functions of $\underline{x}$ are used to characterize $f_{\underline{x}}(\cdot)$. The expected value of these particular functions are called the moments, or statistics, of $\underline{x}$. Generally, an infinite number of moments are needed to completely characterize $f_{\underline{x}}(\cdot)$. Of special interest in stochastic analyses are the first two moments of $\underline{x}$. The first moment is called the mean of $\underline{x}$ and the second moment is called the autocorrelation matrix of $\underline{x}$.

To generate the first moment of $\underline{x}$, or the mean of $\underline{x}$, let $\underline{g}(\underline{x}) = \underline{x}$. The mean of $\underline{x}$, denoted as $\underline{m}$, will be computed by

$$\underline{m} = E[\underline{x}] = \int_{-\infty}^{\infty} \underline{\xi} f_{\underline{x}}(\underline{\xi}) d\underline{\xi}$$

Note that $\underline{m}$ is not a random variable, but rather a deterministic quantity. This fact is true for all the statistics of a random variable.[1]

To generate the autocorrelation matrix of $\underline{x}$, let $\underline{g}(\underline{x}) = \underline{x}\underline{x}^T$, where the superscript-T denotes a vector transpose operation. The autocorrelation matrix of $\underline{x}$, denoted as Ψ, will be computed by

$$\Psi = E[\underline{x}\underline{x}^T] = \int_{-\infty}^{\infty} \underline{\xi}\underline{\xi}^T f_{\underline{x}}(\underline{\xi}) d\underline{\xi}$$

A statistic closely related to the autocorrelation matrix is the covariance matrix, denoted as P. Like the autocorrelation matrix, the covariance matrix is a second moment. P is the second *central* moment of $\underline{x}$. The covariance matrix is defined as

$$P \triangleq E[(\underline{x}-\underline{m})(\underline{x}-\underline{m})^T] = \int_{-\infty}^{\infty} (\underline{\xi} - \underline{m})(\underline{\xi} - \underline{m})^T f_{\underline{x}}(\underline{\xi}) d\underline{\xi}$$

Using the facts that $E[\cdot]$ is a linear operation and $\underline{m}$ is a deterministic quantity, the relationship between Ψ and P can be derived in a straight forward manner. Consider,

$$P = E[(\underline{x}-\underline{m})(\underline{x}-\underline{m})^T] = E[\underline{x}\underline{x}^T - \underline{x}\underline{m}^T - \underline{m}\underline{x}^T + \underline{m}\underline{m}^T]$$

From equation (2) it is seen that

$$E[\underline{x}\underline{x}^T-\underline{x}\underline{m}^T-\underline{m}\underline{x}^T+\underline{m}\underline{m}^T] = E[\underline{x}\underline{x}^T] - E[\underline{x}\underline{m}^T] - E[\underline{m}\underline{x}^T] + E[\underline{m}\underline{m}^T]$$

Since $\underline{m}$ is a deterministic vector, from equation (1) it is seen that

$$E[\underline{x}\underline{m}^T] = E[\underline{x}]\underline{m}^T = \underline{m}\underline{m}^T;$$

$$E[\underline{m}\underline{x}^T] = \underline{m}E[\underline{x}^T] = \underline{m}\underline{m}^T;$$

$$\text{and} \quad E[\underline{m}\underline{m}^T] = \underline{m}\underline{m}^T.$$

Using these relationships, then

$$P = E[\underline{x}\underline{x}^T] - \underline{m}\underline{m}^T = \Psi - E[\underline{x}]E[\underline{x}^T] \qquad (3)$$

Note that if $\underline{m} = \emptyset$ then $P = \Psi$.

A useful relation is statistical independence. Two random variables are defined to be statistically independent if their joint probability density function is equal to the product of their individual density functions.

Definition of Independence

Given two random variables, u and v, then u and v are statistically independent if [2]

$$f_{uv}(u,v) = f_u(u)f_v(v)$$

where

$f_{uv}(u,v) \triangleq$ joint pdf of u and v;

$f_u(u) \triangleq$ pdf of u;

and $f_v(v) \triangleq$ pdf of v.

Uncorrelated and Orthogonal Random Variables

Two random variables, u and v, are uncorrelated if [2]

$$E[uv] = E[u]E[v].$$

They are orthogonal if [2]

$$E[uv] = \emptyset.$$

Theorem 1

If two random variables, u and v, are independent, they are also uncorrelated.[1]

Proof:

$$E[uv] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} uvf_{uv}(u,v)dudv = \int_{-\infty}^{\infty} uf_u(u)du \int_{-\infty}^{\infty} vf_v(v)dv$$

$$= E[u]E[v]$$

Theorem 2

If two n-dimensional random variable vectors, $\underline{x}$ and $\underline{y}$, are uncorrelated, then the covariance of their sum equals the sum of their covariances.[1]

Proof:

$$E[((\underline{x}-\underline{m}_x) + (\underline{y}-\underline{m}_y))((\underline{x}-\underline{m}_x) + (\underline{y}-\underline{m}_y))^T] =$$

$$E[(\underline{x}-\underline{m}_x)(\underline{x}-\underline{m}_x)^T] + E[(\underline{x}-\underline{m}_x)(\underline{y}-\underline{m}_y)^T] + E[(\underline{y}-\underline{m}_y)(\underline{x}-\underline{m}_x)^T]$$

$$+ E[(\underline{y}-\underline{m}_y)(\underline{y}-\underline{m}_y)^T]$$

but

$$E[(\underline{x}-\underline{m}_x)(\underline{x}-\underline{m}_x)^T] = P_x;$$

$$E[(\underline{y}-\underline{m}_y)(\underline{y}-\underline{m}_y)^T] = P_y;$$

$$E[(\underline{x}-\underline{m}_x)(\underline{y}-\underline{m}_y)^T] = E[\underline{x}\underline{y}^T] - \underline{m}_x E[\underline{y}^T] - E[\underline{x}]\underline{m}_y^T + \underline{m}_x\underline{m}_y^T = 0;$$

and

$$E[(\underline{y}-\underline{m}_y)(\underline{x}-\underline{m}_x)^T] = E[\underline{y}\underline{x}^T] - \underline{m}_y E[\underline{x}^T] - E[\underline{y}]\underline{m}_x^T + \underline{m}_y\underline{m}_x^T = 0;$$

Therefore,

$$E[((\underline{x}-\underline{m}_x) + (\underline{y}-\underline{m}_y))((\underline{x}-\underline{m}_x) + (\underline{y}-\underline{m}_y))^T] = P_x + P_y$$

Theorem 2 can easily be extended to provide the relationship that if the random vector,

$$\underline{x} = \sum_{i=1}^{n} \underline{u}_i$$

then

$$P_x(\underline{x}) = \sum_{i=1}^{n} P_{ui} \qquad (4)$$

Likewise, from equation (1) and Theorem 2, if

$$\underline{x} = \sum_{i=1}^{n} A_i \underline{u}_i$$

then

$$P_x(\underline{x}) = \sum_{i=1}^{n} A_i P_{ui} A_i^T \qquad (5)$$

Definition of a Stochastic Process

A process that contains an element of chance is formally called a stochastic process. A stochastic process will be described by some function that contains one or more random variables. The statistics of the stochastic process will be a function of not only the statistics of its random variables but also the describing function itself. Some examples of stochastic processes are

$X(t)$ = A sin(ωt) where ω is a random variable with a known pdf.

$Y(t)$ = some noiselike signal with no deterministic structure.

$\dot{x}(t)$ = w(t) where w(t) is a random varible with a uniform distribution.

Ingredients of Stochastic Simulation

Analog Simulation

In order to include a random process in an analog simulation, one needs, in addition to the mathematical model of the process, an electronic device that will generate wide-bandwidth noise. Wide-bandwidth noise is noise that contains power components over a very large range of frequencies. Ideally, the noise generator should produce noise that has a frequency spectrum of constant amplitude over an infinite range of frequencies. Such noise is said to have an infinite bandwidth with a constant spectral density function and is termed "white noise".

The desire to produce white noise comes from the fact that the mathematics involved with analyzing systems containing white noise is greatly simplified. For instance, the autocorrelation function, $\Psi(\tau)$, for a white noise vector is of the form [3]

$$\Psi(\tau) = Q\ \delta(\tau)$$

where

$\delta(\tau) \triangleq$ Dirac delta function

and Q is the spectral density matrix of the noise

The Dirac delta function is a defined mathematical function that has infinite amplitude, zero width, and occurs at $\tau = 0$.

As seen from its autocorrelation function, white noise is not correlated in time. In other words, knowing the value of the noise at any time provides no information concerning the value of the noise at any other time. This uncorrelatedness-in-time property as well as the constant spectral amplitude property, vastly simplifies the mathematics involved with processing the noise statistics. Thus, from a mathematical sense, a white noise model is a very attractive model. However, white noise is physically not possible. If this isn't obvious from the fact that it contains power components at an infinite number of frequencies, consider the argument that white noise changes it's value by an infinite amount in zero time. One might ask how such a noise model could be justifiably used in a simulation of a real system. There are two justifications.

First, if a physical noise generator has a fairly flat spectral density over a range of frequencies that is much greater than the bandwidth of the system that is driven by the noise source, then the effect of the bandlimited noise on the system is approximately the same as if it were driven by white noise. Figure 1 demonstrates this point graphically.[1] Second, virtually any spectral density function can be shaped from a white noise spectral density function by processing the white noise through a shaping filter.[1]

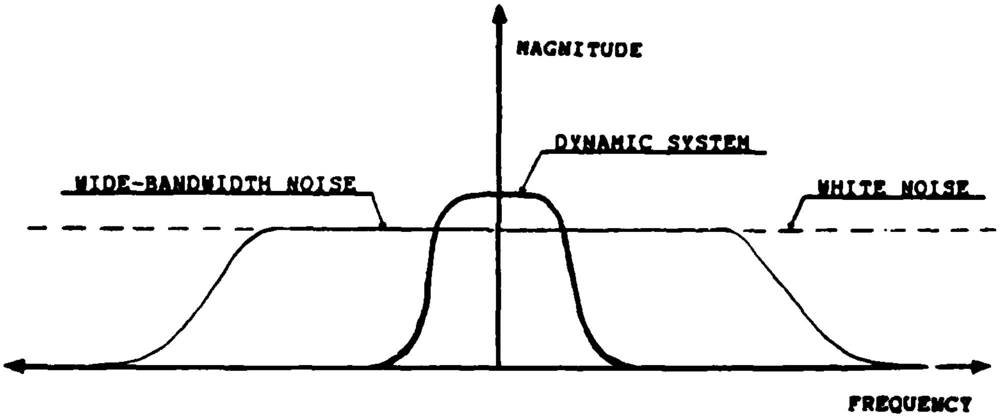

Figure 1. Frequency Spectrums of Interest

Therefore, the model of the true noise source can be simulated by a white noise source driving a shaping filter. The white noise source would in reality be a wide-bandwidth noise generator that has a bandwidth much greater than the bandwidth of the shaping filter. Since an analog simulation uses continuous analog devices and models to simulate the continuous noise process, there is a one-to-one correspondence between the statistics of the simulated noise process and the statistics of the actual noise process.

This one-to-one correspondence is indeed a nice relationship because it precisely defines the required noise statistics to be used in the simulation in order to accurately model the assumed physical noise process. Obviously, it would be equally attractive if this one-to-one correspondence held for a digital simulation. Unfortunately, it does not.

Digital Simulation

To simulate a continuous random process in a digital simulation, one needs the same functional ingredients as is required in an analog simulation; that is, a noise generator, a shaping filter, and knowledge of the relationship between the statistics of the continuous noise process and the statistics of the digital simulation model. Unlike an analog computer, a digital computer is a device that is only capable of producing a finite number of conditions or states. Because a digital computer is a finite state machine, it is not possible to generate a truly continuous random process. The best that can be achieved is the generation of a repeatable sequence of numbers that within the finite window of a sample space appears to be random. However, if the sample is large enough, this "random" sequence of numbers is usually adequate for simulating continuous noise.

Methods for generating random sequences on a digital computer (commonly called random number generators) are bountiful. Many of these random number generators provide excellent and predictable statistical properties. The most commonly used random number generators produce a random sequence that has a uniform distribution over some range of numbers. This uniform distribution can be shaped into most desired noise distributions through the use of shaping filter algorithms in much the same way as is done in analog computers.

The final ingredient needed to simulate continuous noise in a digital simulation is the knowledge of the relationship between the statistics of continuous noise process and the digital or discrete noise process. Unlike in analog simulation, there is not a one-to-one correspondence between these statistical properties. In fact, this dissertation will show that the relationship is complex and a factor of not only the continuous noise statistics and the dynamics of the system, but also of the numerical algorithms used to solve the equations of motion.

The simplest situation in which a relationship would be required is the case when one is attempting to simulate a continuously available measurement of a state that is corrupted by additive white noise. This situation is described mathematically by

$$z_c(t) = x(t) + v_c(t)$$

where

$z_c(t)$ = continuous measurement as a function of time, t;

$x(t)$ = a deterministic function or state;

and $v_c(t)$ = a continuous white noise function.

The discrete measurement equation for this case is

$$z_d(t_i) = x(t_i) + v_d(t_i)$$

where

$z_d(t_i)$ = discrete measurement taken at time equal to t_i;

$x(t_i)$ = the value of $x(t)$ at $t = t_i$;

and $v_d(t_i)$ = a discrete noise term applied at $t = t_i$.

In a simulation that contains this type of model, it is necessary that the statistics of $z_d(t_i)$ equal the statistics of $z_c(t)$ evaluated at $t = t_i$, for all t_i. Since $x(t)$ is deterministic and the noise is additive, this requirement will be met if the statistics of $v_d(t_i)$ equal the statistics of $v_c(t)$ evaluated at $t = t_i$, for all t_i. Maybeck [1] develops the relationship to meet this requirement. A summary of Maybeck's development is given below.

Assume the $v_c(t)$ is a zero-mean Gaussian noise with

$$E[v_c(t)v_c(t+\tau)] = R_c\delta(\tau)$$

where

$E[\cdot]$ denotes the statistical expectation;

R_c = spectral density of $v_c(t)$;

and $\delta(\tau)$ denotes the Dirac delta function.

If $v_d(t_i)$ is a zero-mean Gaussian sequence with

$$E[v_d^2(t_i)] = R_d(t_i) = R_c/\Delta t_i \tag{6}$$

where $\Delta t_i \triangleq t_{i+1} - t_i$

and $E[v_d(t_i)v_d(t_j)] = 0$ for $i \neq j$

then, in the limit as $\Delta t_i \rightarrow 0$, equation (6) will reach the strength of $R_c\delta(0)$. Hence, the desired relationship between the covariance of the random number generator and the spectral density of the continuous noise process is given by equation (6). Note that even in this simple case that includes no dynamics, there is is not a one-to-one statistical correspondence. Discretization has introduced a functional relationship between the statistics of the simulated noise process and the noise process of the simulator. The relationship given by equation (6) is the simplest relationship that will be developed in this study. When the process equations contain dynamics, the functional relationships will become more complex and, without simplifying assumptions, will also be explicit functions of the dynamics.

Dissertation Brief

Specifically, the class of problems that will be addressed in this dissertation are linear first-order differential systems that are driven by Gaussian white noise. Relationships will be derived for the accurate solution of this class of problems in digital simulations that use a number of popular integration methods. It will be shown that each integration method yields a different relationship for the noise statistics. It will further be shown that a number of popular integration methods that are often preferred in deterministic simulations are impractical for simulations of stochastic systems.

The general problem will be described and analytically developed in Chapter II. The necessary functional relationships between the continuous noise process statistics and the discrete noise statistics for the integrators of interest will be derived in Chapter III. An error analysis for the results obtained in Chapter III will be developed in Chapter IV. The results from this analysis will define the limits of validity for a simplifying assumption made in Chapter II. The analytical findings of this research will be demonstrated through simulation, comparing the numerical results to known analytical solutions. The simulation results will be given in Chapter V. The research findings will be summarized and recommendations will be made in Chapter VI.

II. PROBLEM DESCRIPTION

This research is applied to the analysis of problems belonging to the class of linear time-invariant stochastic differential systems. Mathematically, this class of problems is described by

$$\dot{\underline{x}}(t) = A\underline{x}(t) + B_u\underline{u}(t) + B_w\underline{w}(t) \qquad (7)$$

where

$\underline{x}(t) \triangleq$ n-dimensional state vector;

$\dot{\underline{x}}(t) = \dfrac{d\underline{x}(t)}{dt}$;

$\underline{u}(t) \triangleq$ p-dimensional deterministic input vector;

$\underline{w}(t) \triangleq$ m-dimensional random input vector;

$A \triangleq$ nxn system matrix of constant elements;

$B_u \triangleq$ nxp input matrix of constant elements;

and $B_w \triangleq$ nxm input matrix of constant elements.

For problems in the form of equation (7), the state vector can be described by the summation of two n-dimensional vectors, one purely deterministic and the other purely stochastic. Thus,

$$\underline{x}(t) = \underline{x}_u(t) + \underline{x}_w(t) \qquad (8)$$

where

$\underline{x}_u(t)$ = deterministic state vector

and $\underline{x}_w(t)$ = stochastic state vector.

Substituting equation (8) into equation (7) yields

$$\dot{\underline{x}}_u(t) + \dot{\underline{x}}_w(t) = [A\underline{x}_u(t) + B_u\underline{u}(t)] + [A\underline{x}_w(t) + B_w\underline{w}(t)] \quad (9)$$

Clearly from equation (9) the problem of analyzing the system can be separated into a deterministic analysis problem and a stochastic analysis problem. Since the stochastic analysis problem is the one of interest, it will be assumed, without loss of generality, that $\underline{u}(t) = 0$ and $\underline{x}_u(0) = 0$. By making these simplifying assumptions and dropping the subscripts, one finds the stochastic system to be described by

$$\dot{\underline{x}}(t) = A\underline{x}(t) + B\underline{w}(t) \quad (10)$$

Since $\underline{w}(t)$ is a non-deterministic function, equation (10) can not be solved explicitly. The best that one can do is solve for the statistics of $\underline{x}(t)$, which are deterministic quantities. If $\underline{w}(t)$ is Gaussian, the first-order and second-order statistics, mean and covariance, fully describe the statistical properties of $\underline{w}(t)$.[1,2] Further, since the system is linear, the statistics of $\underline{x}(t)$ will also be Gaussian.[1] Thus, one only needs to determine the mean and covariance of $\underline{x}(t)$ to determine the complete statistics of $\underline{x}(t)$. If $\underline{w}(t)$ is non-Gaussian then higher order statistics will be needed for a complete statistical description.[2] However, even for non-Gaussian distributions, the system analyst is often only interested in determining the first two statistical orders due to the physical meaning of those statistics. For convenience, it will be assumed that $\underline{w}(t)$ has a Gaussian distribution.

Determination of the Statistics of the State Vector

Since equation (10) is a stochastic differential equation, normal solution techniques can not be used to determine $\underline{x}(t)$. Another approach will be used in order to find the form of $\underline{x}(t)$ that will allow the determination of the statistics of $\underline{x}(t)$. To this end, define a random variable vector, $\underline{\beta}(t)$, such that

$$\underline{\beta}(t) = \int_0^t \underline{w}(\tau)\, d\tau$$

where $\underline{w}(\tau)$ is a zero-mean Gaussian white noise vector with a spectral density matrix equal to Q. $\underline{\beta}(t)$ is called a Weiner or Brownian-motion process.[1,2,3] The statistics of $\underline{\beta}(t)$ can easily be determined. The mean or average of Brownian-motion is

$$E[\underline{\beta}(t)] = E\left[\int_0^t \underline{w}(\tau)\, d\tau \right] = \int_0^t E[\underline{w}(\tau)]\, d\tau = 0$$

The autocorrelation matrix of $\underline{\beta}(t)$ is

$$E[\underline{\beta}(t)\underline{\beta}^T(t)] = E\left[\int_0^t \underline{w}(\tau)\, d\tau \int_0^t \underline{w}^T(\sigma)\, d\sigma \right]$$

$$= \int_0^t \int_0^t E[\underline{w}(\tau)\underline{w}^T(\sigma)]\, d\tau d\sigma$$

As presented in Chapter I, the $E[\underline{w}(\tau)\underline{w}^T(\sigma)] = Q\delta(\tau-\sigma)$, therefore

$$E[\underline{\beta}(t)\underline{\beta}^T(t)] = \int_0^t\int_0^t Q\delta(\tau-\sigma)\, d\tau d\sigma = \int_0^t Q\, d\sigma = Qt$$

Using this defined function, $\underline{\beta}(t)$, equation (10) can be rewritten in another form that can be solved for the statistics of $\underline{x}(t)$. Since equation (10) is a linear differential equation in time, it can be rewritten as [1]

$$d\underline{x}(t) = A\underline{x}(t)dt + Bd\underline{\beta}(t)$$

This equation can be integrated to yield

$$\underline{x}(t) = \underline{x}(t_0) + \int_{t_0}^t A\underline{x}(\tau)\, d\tau + \int_{t_0}^t Bd\underline{\beta}(\tau)$$

The problem at hand is to determine a stochastic process $\underline{x}(\cdot)$ that satisfies this integral equation. In section 4.8 of Maybeck [1], it is shown that the following equation is such a process.

$$\underline{x}(t) = \Phi(t,t_0)\underline{x}(t_0) + \int_{t_0}^t \Phi(t,\tau)Bd\underline{\beta}(\tau) \qquad (11)$$

where $\Phi(t,t_0)$ is the state transition matrix that satisfies $\dot{\Phi}(t,t_0) = A\Phi(t,t_0)$ and $\Phi(t_0,t_0) = I$, where I is the identity matrix and A is the system matrix in equation (10). Note that $\underline{x}(t)$ is a stochastic process, thus, its solution is non-deterministic; however, using the knowledge of the statistics of $\underline{\beta}(t)$, the statistics of $\underline{x}(t)$ can be determined.

Mean of the State Vector

The mean of $\underline{x}(t)$, denoted $E[\underline{x}(t)]$, is described by

$$E[\underline{x}(t)] = \Phi(t,t_0)E[\underline{x}(t_0)] + E\left[\int_{t_0}^{t} \Phi(t,\tau)Bd\underline{\beta}(\tau) \right]$$

where $\Phi(t,t_0) = \epsilon^{A(t-t_0)}$

$$= I + A*(t-t_0) + \frac{A^2*(t-t_0)^2}{2!} + \ldots \qquad (12)$$

and t and t_0 denote arbitrary values of time. In equation (12), I is an nxn identity matrix. Since $E[\underline{\beta}(t)] = 0$, the stochastic integral is also zero-mean and

$$E[\underline{x}(t)] = \Phi(t,t_0)E[\underline{x}(t_0)] \qquad (13)$$

Second Moment

The autocorrelation matrix of $\underline{x}(t)$ can be determined using equation (11) and equation (2) to write $E[\underline{x}(t)\underline{x}^T(t)]$ as the sum of four separate expectations. Using the definition of $\underline{\beta}(t)$, $\underline{x}(t_0)$ is independent of $\underline{\beta}(t)$. From Theorem 1 and the fact that the mean of the stochastic integral is zero, the expected value of two cross terms is zero. For example,

$$E\left[\int_{t_0}^{t} \Phi(t,\tau)Bd\underline{\beta}(\tau)\underline{x}^T(t_0)\right] = E\left[\int_{t_0}^{t} \Phi(t,\tau)Bd\underline{\beta}(\tau)\right]E[\underline{x}^T(t_0)] = 0$$

Therefore, the autocorrelation matrix of $\underline{x}(t)$ will be

$$E[\underline{x}(t)\underline{x}^T(t)] = \Phi(t,t_0)E[\underline{x}(t_0)\underline{x}^T(t_0)]\Phi^T(t,t_0)$$
$$+ \int_{t_0}^{t} \Phi(t,\tau)BQ(\tau)B^T\Phi^T(t,\tau)d\tau$$

The covariance of $\underline{\mathbf{x}}(t)$, denoted as $P(t)$, can be derived directly from the autocorrelation matrix of $\underline{\mathbf{x}}(t)$ by the relationship given in equation (3). Specifically,

$$E[\underline{\mathbf{x}}(t)\underline{\mathbf{x}}^T(t)] = P(t) + E[\underline{\mathbf{x}}(t)]E[\underline{\mathbf{x}}^T(t)]$$

and $$E[\underline{\mathbf{x}}(t_0)\underline{\mathbf{x}}^T(t_0)] = P(t_0) + E[\underline{\mathbf{x}}(t_0)]E[\underline{\mathbf{x}}^T(t_0)]$$

Evaluating the autocorrelation matrix of $\underline{\mathbf{x}}(t)$ using these relationships and equation (13), yields

$$P(t) = \Phi(t,t_0)P(t_0)\Phi^T(t,t_0) + \int_{t_0}^{t} \Phi(t,\tau)BQ(\tau)B^T\Phi^T(t,\tau)\,d\tau \qquad (14)$$

Notice that equation (14) is a deterministic equation as is equation (13).

Analysis Approach

A stochastic analysis of this system involves solving for the statistics of $\underline{\mathbf{x}}(t)$; specifically, for the mean and covariance of $\underline{\mathbf{x}}(t)$. For low-order linear systems, equation (13) and equation (14) can be solved analytically; but for large-order linear systems and of course nonlinear systems, simulations are normally used to find the statistics of $\underline{\mathbf{x}}(t)$. In a digital simulation, a random number generator is used to generate a random sequence, $\underline{\mathbf{w}}_D$, that will simulate $\underline{\mathbf{w}}(t)$. By repetitively running the simulation using a large number of sample sequences from the random number generator and statistically averaging the results, an approximation of $E[\underline{\mathbf{x}}(t)]$ and $P(t)$ can be generated. This analysis method is commonly called a Monte Carlo analysis.

In a digital simulation, $\underline{x}(t)$ will be represented by a discrete state vector $\underline{x}_D$ and the propagation of $\underline{x}(t)$ will be approximated by the propagation of $\underline{x}_D$ using discrete difference equations. In order to insure a faithful simulation, the statistics of $\underline{x}_D(t_k)$ must be approximately equal to the statistics of $\underline{x}(t)$ evaluated at $t = t_k$ for all t over the range of time of interest. This can be accomplished on an incremental basis assuming that the statistics of $\underline{x}_D(t_\emptyset)$ equal the statistics of $\underline{x}(t_\emptyset)$ where $t_\emptyset$ is some arbitrary initial time. By insuring that the propagation of the statistics of $\underline{x}_D$ approximates the propagation of the statistics of $\underline{x}$ from one update time to the next and assuming that accumulated errors remain small, then the requirement for a faithful simulation will be realized.

Using equation (13), the equation that describes the propagation of the mean of $\underline{x}(t)$ from $t = t_i$ to $t = t_{i+1}$ where $t_{i+1} = t_i + h$ will be

$$E[\underline{x}(t_{i+1})] = \Phi(h,\emptyset)E[\underline{x}(t_i)] \qquad (15)$$

Likewise, from equation (14), the covariance of $\underline{x}(t)$ will be propagated from $t = t_i$ to $t = t_{i+1}$ by

$$P(t_{i+1}) = \Phi(h,\emptyset)P(t_i)\Phi^T(h,\emptyset) + \int_\emptyset^h \Phi(h,\tau)BQB^T\Phi^T(h,\tau)\, d\tau \qquad (16)$$

Development of the Discrete System Model

In the digital simulation, the discrete state $\underline{x}_D$ would be propagated from t_i to $t_{i+1} = (t_i + h)$ by [1]

$$\underline{x}_D(t_{i+1}) = \Phi_D(h,0)\underline{x}_D(t_i) + B_D\underline{w}_D(t_i) \qquad (17)$$

where

$\Phi_D(\cdot,\cdot)$ is a discrete state transition matrix

and B_D is a discrete input matrix.

$\Phi_D(\cdot,\cdot)$ will not only be a function of the A matrix in equation (10) and B_D will not only be a function of the A and B matrices in equation (10), but these two functions will also be dependent on the integration method used in the simulation.

Determination of the Discrete State Statistics

Mean of the Discrete State Vector

If the mean of $\underline{w}_D$ is defined to be zero for all t_i, then, from equation (17), it is seen that the mean of $\underline{x}_D$ will be propagated from t_i to t_{i+1} by

$$E[\underline{x}_D(t_{i+1})] = \Phi_D(h,0)E[\underline{x}_D(t_i)] \qquad (18)$$

Second Moment of the Discrete State Vector

The autocorrelation matrix of $\underline{x}_D$ can be derived directly from equation (17) by determining $E[\underline{x}_D(t_{i+1})\underline{x}_D^T(t_{i+1})]$. Since the discrete state $\underline{x}_D(t_i)$ can not be influenced by the input $\underline{w}_D(t_i)$ (present state conditions can only be influenced by past inputs and initial state conditions), $\underline{x}_D(t_i)$ is independent of $\underline{w}_D(t_i)$. Therefore, by Theorem 1 and the fact that $E[\underline{w}_D(t_i)] = 0$, the expected value of the two cross terms will be zero. Thus,

$$E[\underline{x}_D(t_i)\underline{w}_D^T(t_i)] = E[\underline{w}_D(t_i)\underline{x}_D^T(t_i)] = 0$$

and

$$E[\underline{x}_D(t_{i+1})\underline{x}_D^T(t_{i+1})] = \Phi_D(h,0)E[\underline{x}_D(t_i)\underline{x}_D^T(t_i)]\Phi_D^T(h,0) + B_D Q_D B_D^T$$

where $Q_D = E[\underline{w}_D(t_i)\underline{w}_D^T(t_i)]$ for all t_i.

The discrete covariance matrix, $P_D(\cdot)$, can be determined from the autocorrelation matrix of $\underline{x}_D$ by using the following relationships

$$E[\underline{x}_D(t_{i+1})\underline{x}_D^T(t_{i+1})] = P_D(t_{i+1}) + E[\underline{x}_D(t_{i+1})]E[\underline{x}^T(t_{i+1})]$$

$$E[\underline{x}_D(t_i)\underline{x}_D^T(t_i)] = P_D(t_i) + E[\underline{x}_D(t_i)]E[\underline{x}^T(t_i)]$$

and $E[\underline{x}_D(t_{i+1})] = \Phi_D(h,0)E[\underline{x}_D(t_i)] = E[\Phi_D(h,0)\underline{x}_D(t_i)]$.

Using these relationships,

$$P_D(t_{i+1}) = \Phi_D(h,0)P_D(t_i)\Phi_D^T(h,0) + B_D Q_D B_D^T \qquad (19)$$

Development of Conditions for a Faithful Simulation

By direct comparison of equation (18) to equation (15), it is clear that in order for the propagation of the mean of the discrete state vector to approximate the propagation of the mean of the continuous state vector, $\Phi_D(h,0)$ must be approximately equal to $\Phi(h,0)$. This is the same requirement needed to insure digital simulations of deterministic systems are faithful. Thus, it will be assumed that the numerical integrator used in the digital simulation will insure that

$$\Phi_D(h,0) \approx \Phi(h,0)$$

Making this assumption, a comparison of equation (19) to equation (16) yields that the propagation of the discrete covariance will be approximately equal to the propagation of the continuous covariance if

$$B_D Q_D B_D^T = \int_0^h \Phi(h,\tau) B Q B^T \Phi^T(h,\tau)\, d\tau \qquad (20)$$

Selection of an integration method that will satisfy $\Phi_D(h,0) \approx \Phi(h,0)$ will insure that $E[\mathbf{x}_D(t_i)]$ approximates $E[\mathbf{x}(t)]$ evaluated at $t = t_i$. However, additional requirements are needed to ensure that the second-order statistics are faithfully simulated. Thus, equation (20) describes the essence of the problem. If $\Phi(\cdot,\cdot)$ is known then the right side of equation (20) can be solved either in closed form or numerically. However, in general $\Phi(\cdot,\cdot)$ will not be known. Of course, the general case is the one of interest. What is known, in general, is that as h approaches zero, $\Phi(h,0)$ approaches the identity matrix. In previous works of this type, this fact was used routinely to approximate the state transition matrix with the identity matrix in order to evaluate the right side of equation (20). [1,3,5] At the outset, the analyses developed in this work will also use this relationship. However, because of the practical significance of this convenient assumption, later in the error analyses section, Chapter IV, this assumption will be fully investigated.

If h is selected reasonably small such that

$$h * |\lambda_{max}| \ll 1$$

where $|\lambda_{max}|$ is the magnitude of the largest eigenvalue of A, then

$$\Phi(h,0) \approx I \tag{21}$$

Using equation (21), evaluation of equation (20) yields

$$B_D Q_D B_D^T = hBQB^T \tag{22}$$

In a digital simulation used to perform a Monte Carlo analysis of this type system, it is necessary to select an h that will satisfy equation (21) and it is necessary to adjust the covariance of the random number generator such that equation (22) is satified. It will be shown in Chapter IV of this dissertation that the need for an exceptionally small h imposed by equation (21) can be relaxed for certain integration methods.

One additional problem is that B_D, the discrete input matrix, is not known. B_D will be a function of the integration method used in the simulation. For each integrator of interest, B_D will be determined in order to derive the necessary conditions that will satisfy equation (22). Griffith [4] showed that if an Euler integrator, which will be defined in Chapter III, is used in the simulation, then

$$B_D = hB \tag{23}$$

He then concluded using equations (22) and (23) that for a simulation employing an Euler integrator

$$Q_D = Q/h \tag{24}$$

Griffith then applied his analytical findings to a problem that used an Adams-Bashforth integrator (also to be defined in Chapter III). Griffith's numerical results seemed to show that equation (24) was also a valid relationship for the Adams-Bashforth method. It will be shown analytically in this work that Griffith's numerical findings are valid only under very restricted conditions. Further, it will be shown that each for integration method there exists a unique function relating the continuous input noise statistics to the random number generator's noise statistics.

III. INTEGRATOR ANALYSES

The numerical algorithms that will be considered in this study are designed to approximate the solution to ordinary differential equations and can be divided into two major categories; fixed-step methods and multistep methods. Fixed-step methods utilize derivative information that is computed or approximated within the current integration step interval, whereas multistep methods utilize derivative information accumulated over a number of steps. There are advantages and disadvantages to each type of integration method. The type of algorithm that an analyst chooses to use in a given situation is usually the one that he feels will provide accurate and timely solutions. Many integration techniques have been studied in detail to assist the analyst in making an intelligent choice. However, most of these studies have been performed with application to deterministic systems and not to stochastic systems. This dissertation shows that when applied to solving stochastic differential equations, each integration method will affect the relationship of the noise generator statistics to the continuous noise process statistics. In this chapter, that relationship between the statistics will be developed for a number of widely used integration algorithms.

Fixed-Step (Runge-Kutta) Methods

Given a differential system of the form

$$\dot{\underline{x}} = \underline{f}(\underline{x}, t) \qquad (25)$$

a general Runge-Kutta method approximates the solution $\underline{x}$ at time $t = (i+1)h$, denoted as $\underline{x}_{i+1}$, by [5]

$$\underline{x}_{i+1} = \underline{x}_i + (\alpha_1 \underline{k}_1 + \alpha_2 \underline{k}_2 + \ldots + \alpha_n \underline{k}_n) \qquad (26)$$

where

$$\underline{k}_1 = h^*\underline{f}(\underline{x}_i \ , \ ih) \qquad (27)$$

and

$$\underline{k}_j = h^*\underline{f}(\underline{x}_i + a_j \underline{k}_{j-1}, ih + b_j h) \text{ for } j = 2,n \qquad (28)$$

where α_1, α_j, a_j, b_j are dependent on the particular Runge-Kutta method and $\underline{f}(\cdot,\cdot)$ is defined in equation (25).

Euler Integrator

An Euler integrator can be classified as a Runge-Kutta integrator of order one. For the system of interest,

$$f(\underline{x}_i, ih) = A\underline{x}_i + B\underline{w}_i \qquad (29)$$

Applying equation (29) to equations (26) - (28) with n = 1, the Euler integrator will approximate the solution of $\underline{x}(t)$ by

$$\underline{x}_{i+1} = (I + \alpha_1 hA)\underline{x}_i + \alpha_1 hB\underline{w}_{D1}$$

where $\underline{w}_{D1}$ is a discrete noise input vector at the evaluation of $f(\cdot,\cdot)$. Note that for $\alpha_1 = 1$,

$$(I + \alpha_1 hA) = (I + hA) = \Phi_D(h,0)$$

where $\Phi_D(h,0)$ is a first-order truncation of $\Phi(h,0)$ given by equation (12). Therefore, the update equation for the

Euler integrator can be written as

$$\underline{x}_{i+1} = \Phi_D(h,0)\underline{x}_i + \alpha_1 h B \underline{w}_{D1} \qquad (30)$$

The discrete autocorrelation matrix for an Euler integrator can be determined directly from equation (30).

$$E[\underline{x}_{i+1}\underline{x}_{i+1}^T] = \Phi_D(h,0)E[\underline{x}_i\underline{x}_i^T]\Phi_D^T(h,0) + \alpha_1 hBE[\underline{w}_{D1}\underline{x}_i^T]\Phi_D^T(h,0)$$
$$+ \Phi_D(h,0)E[\underline{x}_i\underline{w}_{D1}^T]\alpha_1 hB^T + (\alpha_1 h)^2 BE[\underline{w}_{D1}\underline{w}_{D1}^T]B^T$$

Using the relationship that the statistics of the states at present time are independent of the statistics of the present inputs, and the fact that $E[\underline{w}_{Di}] = 0$ for all i, results in

$$E[\underline{w}_{D1}\underline{x}_i^T] = E[\underline{w}_{D1}]E[\underline{x}_i^T] = 0$$
$$\text{and} \quad E[\underline{x}_i\underline{w}_{D1}^T] = E[\underline{x}_i]E[\underline{w}_{D1}^T] = 0$$

Therefore,

$$E[\underline{x}_{i+1}\underline{x}_{i+1}^T] = \Phi_D(h,0)E[\underline{x}_i\underline{x}_i^T]\Phi_D^T(h,0) + (\alpha_1 h)^2 BE[\underline{w}_{D1}\underline{w}_{D1}^T]B^T$$

From the autocorrelation matrix, the covariance matrix can be determined using the relationships

$$E[\underline{x}_{i+1}\underline{x}_{i+1}^T] = P_D(t_{i+1}) - E[\underline{x}_{i+1}]E[\underline{x}_{i+1}^T]$$
$$= P_D(t_{i+1}) - \Phi_D(h,0)E[\underline{x}_i]E[\underline{x}_i^T]\Phi_D^T(h,0)$$
$$\text{and} \quad E[\underline{x}_i\underline{x}_i^T] = P_D(t_i) - E[\underline{x}_i]E[\underline{x}_i^T]$$

Doing so results in

$$P_D(t_{i+1}) = \Phi_D(h,0)P_D(t_i)\Phi_D^T(h,0) + \alpha_1^2 h^2 BQ_D B^T \qquad (31)$$

where $Q_D = E[\underline{w}_{D1}\underline{w}_{D1}^T]$. Recognizing that equation (31) is in the form of equation (19), provides the relationship

$$B_D Q_D B_D^T = \alpha_1^2 h^2 BQ_D B^T \qquad (32)$$

Equating equation (32) to equation (22) yields the statistical relationship

$$Q_D = Q/(\alpha_1^2 h) \qquad (33)$$

For $\alpha_1 = 1$, equation (33) is the same result that Griffith derived. [4] Note that equation (33) is independent of the system dynamics. This is a nice relationship because it means that equation (33) can be applied to any digital simulation that uses an Euler integrator.

2nd Order Runge-Kutta Integrator

A second-order Runge-Kutta integrator approximates the solution to $\underline{x}(t)$ at $t = (i+1)h$ by the folowing algorithm.

$$\underline{x}_{i+1} = \underline{x}_i + (\alpha_1 \underline{k}_1 + \alpha_2 \underline{k}_2) \qquad (34)$$

where

$$\underline{k}_1 = h\underline{f}(\underline{x}_i, ih) = h(A\underline{x}_i + B\underline{w}_{D1}) \qquad (35)$$

and

$$\begin{aligned}\underline{k}_2 &= h\underline{f}(\underline{x}_i+a_1\underline{k}_1, (i+b_1)h) = h(A(\underline{x}_i+a_1\underline{k}_1) + B\underline{w}_{D2}) \\ &= [hA + a_1(hA)^2]\underline{x}_i + a_1h^2AB\underline{w}_{D1} + hB\underline{w}_{D2} \qquad (36)\end{aligned}$$

where $\underline{w}_{D1}$ and $\underline{w}_{D2}$ are the discrete noise inputs from the random number generator at the respective evaluations of $\underline{f}(\cdot,\cdot)$. Evaluating equation (34) using (35) and (36) and combining terms yields

$$\begin{aligned}\underline{x}_{i+1} = {} & [I + (\alpha_1+\alpha_2)hA + a_1\alpha_2(hA)^2]\underline{x}_i \\ & + h(\alpha_1 I + a_1\alpha_2 hA)B\underline{w}_{D1} + \alpha_2 hB\underline{w}_{D2} \qquad (37)\end{aligned}$$

The most commonly used 2nd order Runge-Kutta method defines $\alpha_1 = \alpha_2 = 0.5$ and $a_1 = b_1 = 1$. With these defined constants note that

$$[I + (\alpha_1+\alpha_2)hA + a_1\alpha_2(hA)^2] = [I + hA + 0.5(hA)^2] = \Phi_D(h,0)$$

where $\Phi_D(h,0)$ is a second-order truncation of $\Phi(h,0)$ given by equation (12). Therefore, equation (37) can be rewritten as

$$\mathbf{x}_{i+1} = \mathbf{\Phi}_D(h,0)\mathbf{x}_i + B_{D1}\mathbf{w}_{D1} + B_{D2}\mathbf{w}_{D2} \qquad (38)$$

where

$$B_{D1} = h(\alpha_1 I + a_1\alpha_2 hA)B$$

and

$$B_{D2} = \alpha_2 hB$$

Since $\mathbf{w}_{Di}$ represents a zero-mean white noise sequence, $E[\mathbf{w}_{Di}\mathbf{w}_{Dj}^T] = E[\mathbf{w}_{Di}]E[\mathbf{w}_{Dj}^T] = 0$ for all $i \neq j$. Further, since the present state, $\mathbf{x}_i$, can not be a function of present or future inputs, $\mathbf{x}_i$ is independent of $\mathbf{w}_{D1}$ and $\mathbf{w}_{D2}$. By the same procedure used to develop the discrete covariance equation given by equation (19), equation (38) will be used to derive the discrete covariance equation for the 2nd order Runge-Kutta algorithm. That procedure results in

$$P_D(t_{i+1}) = \mathbf{\Phi}_D(h,0)P_D(t_i)\mathbf{\Phi}_D^T(h,0) + B_{D1}Q_{D1}B_{D1}^T + B_{D2}Q_{D2}B_{D2}^T \qquad (39)$$

where

$$Q_{D1} = E[\mathbf{w}_{D1}\mathbf{w}_{D1}^T]$$

and

$$Q_{D2} = E[\mathbf{w}_{D2}\mathbf{w}_{D2}^T]$$

Note that there are two $B_D Q_D B_D^T$ terms in equation (39). This is due to the sampling of the noise generator twice per update, once per derivative-function evaluation. If it is assumed that Q_D is constant over the update interval, then $Q_{D1} = Q_{D2} = Q_D$.

Using the relationship given by equation (5), it is possible to define

$$B_D Q_D B_D^T = B_{D1} Q_D B_{D1}^T + B_{D2} Q_D B_{D2}^T$$

From the definition of B_{D1} and B_{D2} given in equation (38),

$$B_{D1} Q_D B_{D1}^T = \alpha_1^2 h^2 B Q_D B^T + \alpha_1 \alpha_2 a_1 h^3 (B Q_D B^T A^T + A B Q_D B^T) + \alpha_2^2 h^4 (A B Q_D B^T A^T)$$

and $B_{D2} Q_D B_{D2}^T = \alpha_2^2 h^2 B Q_D B^T$

Therefore, for the 2nd order Runge-Kutta integrator,

$$B_D Q_D B_D^T = (\alpha_1^2 + \alpha_2^2) h^2 B Q_D B^T + \alpha_1 \alpha_2 a_1 h^3 (B Q_D B^T A^T + A B Q_D B^T) + \alpha_2^2 h^4 (A B Q_D B^T A^T) \qquad (40)$$

Notice that $B_D Q_D B_D^T$ is a function of the system dynamics, represented by the A matrix. Since the relationship between Q_D and Q will be derived using $B_D Q_D B_D^T$ in equation (22), it is desirable to make $B_D Q_D B_D^T$ independent of the dynamics so the relationship between Q_D and Q will be independent of the dynamics. To accomplish this, the same assumption that was made in arriving at the expression on the right side of equation (22) will be made again to obtain an approximation to equation (40) that is independent of the dynamics. Specifically, assume that h is selected reasonably small, such that

$$B_D Q_D B_D^T \approx (\alpha_1^2 + \alpha_2^2) h^2 B Q_D B^T$$

For $\alpha_1 = \alpha_2 = 0.5$,

$$B_D Q_D B_D^T = h^2 B Q_D B^T \sum_{i=1}^{2} (\alpha_i)^2 = 0.5 h^2 B Q_D B^T \qquad (41)$$

From equations (41) and (22), it is seen that for the 2nd order Runge-Kutta integrator

$$Q_D = 2Q/h \tag{42}$$

A comparison of equation (42) to equation (33) indicates that the relationship between Q_D and Q is dependent on the integration method.

4th Order Runge-Kutta Integrator

A 4th order Runge-Kutta integrator is in the form of

$$\underline{x}_{i+1} = \underline{x}_i + (\alpha_1\underline{k}_1 + \alpha_2\underline{k}_2 + \alpha_3\underline{k}_3 + \alpha_4\underline{k}_4) \tag{43}$$

where

$$\underline{k}_1 = h\underline{f}(\underline{x}_i, ih);$$

$$\underline{k}_2 = h\underline{f}(\underline{x}_i + a_1\underline{k}_1, (i + b_1)h);$$

$$\underline{k}_3 = h\underline{f}(\underline{x}_i + a_2\underline{k}_2, (i + b_2)h);$$

and $\underline{k}_4 = h\underline{f}(\underline{x}_i + a_3\underline{k}_3, (i + b_3)h)$

Evaluating the $\underline{k}_i$'s in equation (43) for the system given by equation (29) yields

$$\underline{k}_1 = h[A\underline{x}_i + B\underline{w}_{D1}] \tag{44}$$

$$\underline{k}_2 = h[(I + a_1hA)A\underline{x}_i + a_1hAB\underline{w}_{D1} + B\underline{w}_{D2}] \tag{45}$$

$$\underline{k}_3 = h[(I + a_2hA + a_2a_1h^2A^2)A\underline{x}_i + a_2a_1h^2A^2B\underline{w}_{D1} + a_2hAB\underline{w}_{D2} + B\underline{w}_{D3}] \tag{46}$$

$$\underline{k}_4 = h[(I + a_3hA + a_3a_2h^2A^2 + a_3a_2a_1h^3A^3)A\underline{x}_i + a_3a_2a_1h^3A^3B\underline{w}_{D1} + a_3a_2h^2A^2B\underline{w}_{D2} + a_3hAB\underline{w}_{D3} + B\underline{w}_{D4}] \tag{47}$$

where the $\underline{w}_{Di}$'s are inputs from the random number generator at the respective evaluations of $\underline{f}(\cdot,\cdot)$. Substituting equations (44) - (47) into equation (43) results in

$$\begin{aligned}\underline{x}_{i+1} = [(I &+ (\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4)hA \\ &+ (a_1\alpha_2 + a_2\alpha_3 + a_3\alpha_4)h^2A^2 \\ &+ (a_2a_1\alpha_3 + a_3a_2\alpha_4)h^3A^3 + (a_3a_2a_1\alpha_4)h^4A^4]\underline{x}_i \\ &+ [\alpha_1 I + a_1\alpha_2 hA + a_2a_1\alpha_3h^2A^2 \\ &\quad + a_3a_2a_1\alpha_4h^3A^3]B\underline{w}_{D1} \\ &+ [\alpha_2 I + a_2\alpha_3hA + a_3a_2\alpha_4h^2A^2]B\underline{w}_{D2} \\ &+ [\alpha_3 I + a_3\alpha_4hA]B\underline{w}_{D3} + [\alpha_4 I]B\underline{w}_{D4} \qquad (48)\end{aligned}$$

A popular 4th order Runge-Kutta integrator uses $\alpha_1 = \alpha_4 = 1/6$; $\alpha_2 = \alpha_3 = 1/3$; $a_1 = a_2 = b_1 = b_2 = 1/2$; and $a_3 = b_3 = 1$. Using these constants to evaluate the state transition part of equation (48) results in

$$I + hA + 0.5(hA)^2 + (hA)^3/6 + (hA)^4/24 = \Phi_D(h,0) \qquad (49)$$

Note that equation (49) is a fourth-order truncation of $\Phi(h,0)$ given by equation (12). Using equation (49), equation (48) can be rewritten as

$$\underline{x}_{i+1} = \Phi_D(h,0)\underline{x}_i + B_{D1}\underline{w}_{D1} + B_{D2}\underline{w}_{D2} + B_{D3}\underline{w}_{D3} + B_{D4}\underline{w}_{D4} \qquad (50)$$

where

$B_{D1} = [\alpha_1 I + a_1\alpha_2hA + a_2a_1\alpha_3h^2A^2 + a_3a_2a_1\alpha_4h^3A^3]B;$

$B_{D2} = [\alpha_2 I + a_2\alpha_3hA + a_3a_2\alpha_4h^2A^2]B;$

$B_{D3} = [\alpha_3 I + a_3\alpha_4hA]B;$

and $B_{D4} = [\alpha_4 I]B$

Equation (50) can be used to determine the second moment function for this algorithm. A number of statistical independence relationships will simplify the math. Since the input noise is a zero-mean white noise sequence, $E[\underline{w}_{Di}\underline{w}_{Dj}^T] = 0$ for all $i \neq j$. Further, the present state is independent of present and future inputs, therefore, $E[\underline{x}_i\underline{w}_{Dk}^T] = 0$ for k = 1 to 4. Using these statistical relationships, the discrete covariance matrix for the 4th order Runge-Kutta algorithm can be derived from equation (50) by the same process used to derive equation (19) from equation (17). The result is

$$P_D(t_{i+1}) = \Phi_D(h,0)P_D(t_i)\Phi_D^T(h,0) + B_{D1}Q_{D1}B_{D1}^T + B_{D2}Q_{D2}B_{D2}^T + B_{D3}Q_{D3}B_{D3}^T + B_{D4}Q_{D4}B_{D4}^T \qquad (51)$$

where $Q_{Dk} = E[\underline{w}_{Dk}\underline{w}_{Dk}^T]$ for k = 1 to 4.

Note that $P_D(t_{i+1})$ has four $B_{Di}Q_{Di}B_{Di}^T$ terms, one from each sample of the noise generator at the four evaluations of $f(\cdot,\cdot)$.

To rewrite equation (51) in the form of equation (19), the $B_{Di}Q_{Di}B_{Di}^T$ terms will be manipulated into one equivalent $B_DQ_DB_D^T$ term. To do this, first assume that Q_D is constant over the update interval, leading to $Q_{D1} = Q_{D2} = Q_{D3} = Q_{D4} = Q_D$. Now, using this assumption and the general relationship given by equation (5), define

$$B_DQ_DB_D^T = B_{D1}Q_DB_{D1}^T + B_{D2}Q_DB_{D2}^T + B_{D3}Q_DB_{D3}^T + B_{D4}Q_DB_{D4}^T$$

From the B_{Di} definitions given in equation (50),

$$\begin{aligned} B_{D1}Q_DB_{D1}^T = {} & \alpha_1^2h^2[BQ_DB^T] + a_1\alpha_1\alpha_2h^3[ABQ_DB^T + BQ_DB^TA^T] \\ & + a_1^2\alpha_2^2h^4[ABQ_DB^TA^T] \\ & + a_2a_1\alpha_1\alpha_3h^4[A^2BQ_DB^T + BQ_DB^T(A^T)^2] \\ & + a_3a_2a_1\alpha_1\alpha_4h^5[A^3BQ_DB^T + BQ_DB^T(A^T)^3] \\ & + a_1^2a_2\alpha_2\alpha_3h^5[A^2BQ_DB^TA^T + ABQ_DB^T(A^T)^2] \\ & + (a_2a_1\alpha_3)^2h^6[A^2BQ_DB^T(A^T)^2] \end{aligned}$$

$$
\begin{aligned}
&+ a_1^2 a_2 a_3 \alpha_2 \alpha_4 h^6 [A^3 B Q_D B^T A^T + A B Q_D B^T (A^T)^3] \\
&+ a_1^2 a_2^2 a_3 \alpha_3 \alpha_4 h^7 [A^3 B Q_D B^T (A^T)^2 + A^2 B Q_D B^T (A^T)^3] \\
&+ (a_3 a_2 a_1 \alpha_4)^2 h^8 [A^3 B Q_D B^T (A^T)^3]
\end{aligned}
$$

$$
\begin{aligned}
B_{D2} Q_D B_{D2}^T = {} & \alpha_2^2 h^2 [B Q_D B^T] + a_2 \alpha_2 \alpha_3 h^3 [A B Q_D B^T + B Q_D B^T A^T] \\
&+ a_2^2 \alpha_3^2 h^4 [A B Q_D B^T A^T] \\
&+ a_3 a_2 \alpha_2 \alpha_4 h^4 [A^2 B Q_D B^T + B Q_D B^T (A^T)^2] \\
&+ a_2^2 a_3 \alpha_3 \alpha_4 h^5 [A^2 B Q_D B^T A^T + A B Q_D B^T (A^T)^2] \\
&+ (a_3 a_2 \alpha_4)^2 h^6 [A^2 B Q_D B^T (A^T)^2] \\
B_{D3} Q_D B_{D3}^T = {} & \alpha_3^2 h^2 [B Q_D B^T] + a_3 \alpha_3 \alpha_4 h^3 [A B Q_D B^T + B Q_D B^T A^T] \\
&+ a_3^2 \alpha_4^2 h^4 [A B Q_D B^T A^T] \\
B_{D4} Q_D B_{D4}^T = {} & \alpha_4^2 h^2 [B Q_D B^T]
\end{aligned}
$$

and $B_D Q_D B_D^T = B_{D1} Q_D B_{D1}^T + B_{D2} Q_D B_{D2}^T + B_{D3} Q_D B_{D3}^T + B_{D4} Q_D B_{D4}^T$ (52)

Notice that $B_D Q_D B_D^T$ is a function of the system dynamics as it was with the 2nd order Runge-Kutta algorithm. As with the 2nd order algorithm, it is desireable to make $B_D Q_D B_D^T$ independent of the dynamics so the relationship between Q_D and Q will be independent of the dynamics. To accomplish this, the same assumption that was made in arriving at the expression on the right side of equation (22) will be made again to obtain an approximation to equation (52) that is independent of the dynamics. Specifically, assume that h is selected reasonably small, such that

$$B_D Q_D B_D^T \approx (\alpha_1^2 + \alpha_2^2 + \alpha_3^2 + \alpha_4^2) h^2 [B Q_D B^T]$$

For $\alpha_1 = \alpha_4 = 1/6$ and $\alpha_2 = \alpha_3 = 1/3$

$$B_D Q_D B_D^T = h^2 B Q_D B^T \sum_{i=1}^{4} (\alpha_i)^2 = h^2 B Q_D B^T / 3.6$$

With this approximation to $B_D Q_D B_D^T$ and equation (22), the relationship between Q_D and Q for this 4th order Runge-Kutta integrator is

$$Q_D = 3.6Q/h \tag{53}$$

nth Order Runge-Kutta

The trend established in the above analyses can be used to extend the results to an nth order Runge-Kutta integrator. Consider that the coefficients used in an nth order Runge-Kutta integrator are selected to ensure an nth order truncation of a Taylor series expansion of the state transition matrix.[6] Therefore, the integrator will always provide a $\underline{\Phi}_D(h,0)$ that approximates the state transition matrix, $\underline{\Phi}(h,0)$, over some interval h. Based on this fact, the state update equation for the system will be in the form of

$$\underline{x}_{i+1} = \underline{\Phi}_D(h,0)\underline{x}_i + \sum_{k=1}^{n} B_{Dk}\underline{w}_{Dk} \tag{54}$$

where

$$B_{Dk} = \left[\sum_{j=k}^{n} \alpha_j \left[\prod_{p=1}^{j-1} a_p\right](hA)^{j-k}\right]B$$

where

$$\prod_{p=m}^{q} a_p \triangleq \begin{cases} 1 & q < m \\ a_m & q = m \\ a_m \cdot a_{m+1} \cdots a_q & q > m \end{cases}$$

and $\underline{\Phi}_D(h,0)$ is an nth order truncation of equation (12). The expression in equation (54) for determining B_{Dk} was derived by noticing the sequence of coefficients in the B_D terms in equation (50), equation (38) and equation (30).

The discrete covariance equation for a system solved by an nth order Runge-Kutta will be

$$P_D(t_{i+1}) = \Phi_D(h,0)P_D(t_i)\Phi_D^T(h,0) + \sum_{k=1}^{n} B_{Dk}Q_{Dk}B_{Dk}^T \qquad (55)$$

where $Q_{Dk} = E[\underline{w}_{Dk}\underline{w}_{Dk}^T]$. Assuming $Q_{Dk} = Q_D$ for all k from 1 to n and assuming that h is small, as assumed in equation (21), then, for an nth order Runge-Kutta integrator

$$B_DQ_DB_D^T \approx h^2BQ_DB^T\sum_{i=1}^{n}(\alpha_i)^2 = h^2BQ_DB^T/\Gamma \qquad (56)$$

where $\Gamma = 1/(\sum_{i=1}^{n}\alpha_i^2)$

From equation (56) and equation (22), the relationship between Q_D and Q will be

$$Q_D = \Gamma Q/h \qquad (57)$$

The correction factor, Γ, for commonly used Runge-Kutta methods is given in Table 1.

Table 1.

Runge-Kutta Correction Factors

Order	α_i coefficients	Corr. Factor
2	$\alpha_1 = .5$, $\alpha_2 = .5$	$\Gamma = 2$
3	$\alpha_1 = \alpha_3 = 1/6$, $\alpha_2 = 2/3$	$\Gamma = 2$
4	$\alpha_1 = 1/6$, $\alpha_2 = 1/3$ $\alpha_3 = 1/3$, $\alpha_4 = 1/6$	$\Gamma = 3.6$
4 (Gil)	$\alpha_1 = 1/6$, $\alpha_2 = (2 - \sqrt{2})/6$ $\alpha_3 = (2 + \sqrt{2})/6$, $\alpha_4 = 1/6$	$\Gamma = 18/7$ $= 2.57$
(5,6)	$\alpha_1 = 7/90$, $\alpha_2 = 0$ $\alpha_3 = 32/90$, $\alpha_4 = 12/90$ $\alpha_5 = 32/90$, $\alpha_6 = 7/90$	$\Gamma = 3.54$

Alternate Analysis Method

The analysis approach taken with Runge-Kutta methods becomes difficult to manage when applied to other integration methods. Jury [7] offers an alternative approach that is based on z-transforms and yields a steady-state value of the autocorrelation matrix. Although this new approach is more manageable in some cases, it is practically, though not mathematically, restricted to scalar equations. To develop and understand this alternative approach, $E[x^2]$ for a 2nd order Runge-Kutta integrator will be determined by the Jury method.

Jury shows that if a discrete scalar system driven by discrete white noise can be described by

$$X(z) = \frac{M(z)}{L(z)} W_D(z) = G(z)W_D(z) \tag{58}$$

where

z^{-1} is the discrete delay operator;

$X(z)$ is the z-transform of the state x_i;

$W_D(z)$ is the z-transform of the input random sequence with variance Q_D;

and $M(z)$ and $L(z)$ are polynomials in z.

then

$$E[x^2] = \frac{Q_D}{2\pi j} \oint_{-\pi}^{\pi} G(z)G(z^{-1})z^{-1}dz \tag{59}$$

where $j = \sqrt{-1}$

In reference [7], Table III of the Appendix, Jury provides an algorithm for evaluating the contour integral in equation (59). A summary of the algorithm follows.

Given equation (58) such that

$$G(z) = \frac{M(z)}{L(z)} = \frac{m_0 z^n + m_1 z^{n-1} + \ldots + m_n}{l_0 z^n + l_1 z^{n-1} + \ldots + l_n} \qquad (60)$$

then equation (59) can be evaluated to yield the steady-state value

$$E[x_1^2] = Q_D \frac{|\Omega_1|}{l_0 |\Omega|} \qquad (61)$$

where $|\cdot|$ denotes the matrix determinant operation. The matrices Ω and Ω_1 are square, have dimension n+1 (n is the order of L(z)), and have elements formed from the coefficients of L(z) and M(z). Specifically,

$$\Omega \triangleq \begin{bmatrix} l_0 & l_1 & l_2 & l_3 & \ldots & l_n \\ l_1 & l_0+l_2 & l_1+l_3 & l_2+l_4 & \ldots & l_{n-1} \\ l_2 & l_3 & l_0+l_4 & l_1+l_5 & \ldots & l_{n-2} \\ \cdot & & & & & \\ \cdot & & & & & \\ \cdot & & & & & \\ l_n & 0 & 0 & 0 & \ldots & l_0 \end{bmatrix} \qquad (62)$$

and Ω_1 is formed by replacing the first column of Ω with the vector

$$\begin{bmatrix} \Sigma_0^n m_i^2 \\ 2\Sigma_0^n (m_i m_{i+1}) \\ 2\Sigma_0^n (m_i m_{i+2}) \\ \cdot \\ \cdot \\ \cdot \\ 2m_0 m_n \end{bmatrix}$$

To apply Jury's algorithm to analyzing the 2nd order Runge-Kutta integrator, consider a scalar linear differential equation of the form

$$\dot{x}(t) = Ax(t) + Bw(t) = f(x,t) \tag{63}$$

Applying equation (63) to the Runge-Kutta routine given by equations (35) - (37) yields a difference equation in the form of

$$x_k = x_{k-2} + 0.5y_{k-1} + 0.5y_{k-2} \tag{64}$$

where

$$x_{k-2} = x(ih);$$

$$x_k \quad = x((i+1)h);$$

$$y_{k-2} = h[Ax(ih) + Bw_{D1}];$$

and $\quad y_{k-1} = h[A(y_{k-2} + x_{k-2}) + Bw_{D2}]$

where w_{D1} and w_{D2} are defined as in equation (35). Note that k increments two for one increment in ih. This is done in order to handle the two sequential inputs, w_{D1} and w_{D2}. Taking the z-transform of equation (64) yields

$$X(z) = \frac{(0.5z + 0.5 + 0.5hA)hBW(z)}{z^2 - \Phi_D} \tag{65}$$

where Φ_D is defined above equation (38). Note that equation (65) is in the form of equation (58); hence, the Jury algorithm can be applied directly. For this case $m_0 = 0$; $m_1 = 0.5$; $m_2 = 0.5(1 + hA)$; $l_0 = 1$; $l_1 = 0$; and $l_2 = -\Phi_D$. Evaluating the matrices Ω and Ω_1 using these coefficients provides

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

and

$$\Omega_1 = \begin{bmatrix} (0.5)^2+(0.5)^2(1+hA)^2 & 0 & -\Phi_D \\ 2(0.5)^2(1+hA)^2 & 1-\Phi_D & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Evaluating the determinates of Ω and Ω_1 results in

$$|\Omega| = (1 - \Phi_D)(1 - \Phi_D^2)$$

and $$|\Omega_1| = (0.5)^2[1 + (1 + hA)^2](1 - \Phi_D)$$

Therefore, from equation (61)

$$E[x_1^2] = \frac{[0.5 + 0.5(hA) + (0.5hA)^2]h^2B^2Q_D}{1 - \Phi_D^2} \qquad (66)$$

Manipulating equation (66) yields

$$E[x_1^2] = \Phi_D^2E[x_1^2] + [0.5 + 0.5hA + (0.5hA)^2]h^2B^2Q_D$$

Comparing this result to the steady-state value of the scalar form of equation (19) provides

$$B_D^2Q_D = [0.5 + 0.5hA + (0.5hA)^2]h^2B^2Q_D$$

which agrees with the scalar form equation (40) evaluated with $\alpha_1 = \alpha_2 = 0.5$ and $a_1 = 1$.

Adams-Bashforth Methods

Adams-Bashforth methods belong to a class of integration algorithms called multistep methods. Multistep methods have the form [8]

$$\underline{x}_{i+1} = \underline{x}_i + \sum_{j=0}^{n-1} \alpha_{j+1} \underline{f}(\underline{x}_{i-j}, (i-j)h) \qquad (67)$$

where the α_j's are coefficients dependent on the particular algorithm and $\underline{f}(\cdot,\cdot)$ is given in equation (29) evaluated at $t = (i-j)h$. Because of the nature of multistep methods, the analysis approach proposed by Jury will be used in determining $B_D^2 Q_D$.

The fourth-order Adams-Bashforth algorithm for the system given by equation (10) is [6]

$$\begin{aligned} \underline{x}_{i+1} = \underline{x}_i &+ h[\alpha_1(A\underline{x}_i + B\underline{w}_i) + \alpha_2(A\underline{x}_{i-1} + B\underline{w}_{i-1}) \\ &+ \alpha_3(A\underline{x}_{i-2} + B\underline{w}_{i-2}) + \alpha_4(A\underline{x}_{i-3} + B\underline{w}_{i-3})] \qquad (68) \end{aligned}$$

where $\alpha_1 = 55/24$; $\alpha_2 = -59/24$; $\alpha_3 = 37/24$; $\alpha_4 = -9/24$. Limiting equation (68) to the scalar case, transforming it to the z-domain and writing it in the form of equation (58) yields

$$X(z) = \frac{hB[\alpha_1 z^3 + \alpha_2 z^2 + \alpha_3 z + \alpha_4]W(z)}{z^4 - (1 + \alpha_1 hA)z^3 - \alpha_2 hAz^2 - \alpha_3 hAz - \alpha_4 hA} \qquad (69)$$

where X, W, A, and B are scalars. The application of equation (69) to the algorithm given by equations (60) - (62) is straight forward, but quite tedious, because it requires solving two polynomial matrices. A procedure was developed and a computer program was written to help facilitate the the computations. That procedure and program are provided in Appendix A. The results of the Jury Analysis for this problem is

$$E[x_1^2] = \frac{h^2B^2Q_D[1 - 6.34hA + 18.9(hA)^2 - 60.8(hA)^3 + \ldots]}{1 - \Phi_D^2}$$

leading to

$$B_D^2Q_D = h^2B^2Q_D[1 - 6.34hA + 18.9(hA)^2 - 60.8(hA)^3 + \ldots]$$

$B_D^2Q_D$ is actually a polynomial in hA of infinite order. The first few significant terms are provided above. It will be assumed that $|hA| < 1$, thereby making it possible to approximate the $B_D^2Q_D$ by a finite polynomial. For convenience, let

$$B_D^2Q_D = h^2B^2Q_D[1 - 6.34hA + 18.9(hA)^2 - 60.8(hA)^3] \qquad (70)$$

Using equation (70) to evaluate the left side of the scalar form of eqation (22) leads to the relationship

$$Q_D = Q/[h(1 - 6.34(hA) + 18.9(hA)^2 - 60.8(hA)^3)] \qquad (71)$$

Notice that equation (71) has a strong dependence on the system dynamics. However, equation (71) will reduce to equation (24) as hA approaches zero.

IV. ERROR ANALYSES

One is always concerned with errors in performing analyses of continuous systems via simulation. Digital simulation introduces a number of unique error sources. The effects of those errors are usually categorized into two areas; one pertaining to round-off errors (finite word-length errors) and one pertaining to truncation errors due to representing the system models by truncated infinite series. With respect to digital simulations of physical systems, round-off errors normally begin to influence the accuracy of the results when the integration step size, h, is decreased to the point that the changes in the system dynamics become smaller than the numerical accuracy of the digital computations. On the other hand, truncation errors typically increase as the step size becomes larger. Unacceptable truncation errors occur when $\Phi_D(h,0)$ no longer accurately approximates $\Phi(h,0)$. Usually, there is a region of possible values of h in which neither type error is significant. For efficiency reasons, the system analyst will normally select the largest value of h that will not introduce significant truncation errors and use a machine that provides adequate numerical accuracy.

In deterministic analyses, the order of the integration method is usually selected to minimize truncation errors. The type of nth order integration method is selected to maximize efficiency of solution. For instance, it is generally accepted that a 4th order Adams-Bashforth algorithm is more efficient than a 4th order Runge-Kutta algorithm.[6] In stochastic analyses, these same problems exist and hence, the same decisions must be made, but an additional error source is introduced due to the need to insure statistical accuracy. In Chapter II, it was shown that the first-order statistics of the noise corrupted states will be processed in the same way as deterministic states but higher order statistics will be processed in a unique manner. The second-order statistics will be processed by a solution to equation (14) and higher order statistics will likewise be processed by unique integral equations. For linear systems forced by Gaussian distributed noise, the first and second-order statistics fully describe the statistical behavior of the system.[1] However, if the noise is non-Gaussian or if the system is nonlinear, higher order statistics will have to be calculated.

Thus far, all of the analyses presented in this paper with regards to the processing of the second-order statistics have been based on the assumption that h would be selected small enough to allow one to approximate the state transition matrix by the identity matrix. In practical terms, this means that the step size required to accurately compute the second-order statistics must be much smaller than is required to compute the first-order statistics as well as the deterministic states. This step size requirement will translate into much more costly analyses. Further, it could lead to the introduction of round-off errors. Since the step size assumption that allowed the use of equation (21) was made for convenience, it will be useful to see if the step size requirement imposed by that assumption can be relaxed. To perform this investigation, a more useful form of equation (20) will be needed.

Analysis Approach

If one restricts the problem to linear systems or to well behaved (analytic) nonlinear systems that can be accurately modeled through some linearization process, then $\Phi(\cdot,\cdot)$ can be represented by the infinite series given in equation (12). Likewise, the integral expression in equation (20) can be represented by an infinite series. To develop the series representation of equation (20), let

$$BQB^T = \Lambda \tag{73}$$

The series form of equation (12) is

$$\Phi(h,\tau) = \sum_{i=0}^{\infty} \frac{A^i(h-\tau)^i}{i!} \tag{74}$$

Using equations (73) and (74) yields

$$\Phi(h,\tau)BQB^T\Phi^T(h,\tau) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \frac{A^i\Lambda(A^T)^j(h-\tau)^{i+j}}{i!j!} \tag{75}$$

Equation (75) is a polynomial of infinite order with the independent variable τ. Since A and Λ are independent of

τ, Equation (75) can easily be integrated with respect to τ and evaluated over the limits 0 to h as in equation (20) to yield

$$\int_0^h \Phi(h,\tau)\Lambda\Phi^T(h,\tau)d\tau = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\frac{A^i\Lambda(A^T)^j}{i!j!}\int_0^h (h-\tau)^{i+j}d\tau$$

$$= h\sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\frac{A^i\Lambda(A^T)^j(h)^{i+j}}{(i+j+1)i!j!} \qquad (76)$$

Equation (76) is a quadratic polynomial in hA of infinite order. If h is selected reasonably small such that $h^*|\lambda_{max}| < 1$, where $|\lambda_{max}|$ is the magnitude of the largest eigenvalue of A, then equation (76) can be reasonably approximated by a finite series of order n. A truncated expression of equation (76) can be used to determine the order of magnitude of the local error in the discrete covariance computations. The appropriate order of truncation will be a function of the integration method and its selection will be based on the order of the $B_DQ_DB_D^T$ expression determined for the integration method. Thus, at this point, the analysis becomes integration method dependent.

Integrator Analyses

In the analyses to follow, equation (76) will be truncated at various orders in h. For notational brevity, an nth order truncation of equation (76) means that i will step from 0 to n and j will step from 0 to n, however all terms resulting from the condition [(i+j) > n] will be neglected.

Euler Integrator

For the Euler integrator, from equation (32),

$$B_D Q_D B_D^T = h^2 B Q_D B^T \qquad (77)$$

Equating the right side of equation (77) to a first-order truncation of equation (76), yields

$$h^2 B Q_D B^T = h(BQB^T + 0.5h(ABQB^T + BQB^T A^T)) \qquad (78)$$

If $Q_D = Q/h$, as proposed by Griffith[4], then equation (78) will be valid if the first-order terms in h on the right side of equation (78) are much smaller than the zeroth-order term. However, if it is assumed (as Griffith did) that the state transition matrix can be approximated by the identity matrix (a zeroth-order truncation of equation (12)) then equation (24) will be a valid relationship that can be used to satisfy equation (78). Making that assumption translates into a smaller step size requirement. Therefore, it is concluded that the local error that results in using equation (24) to relate Q_D to Q for the Euler integrator is of order h as compared to a local integration error of order h^2.[6]

2nd Order Runge-Kutta

With $\alpha_1 = \alpha_2 = 0.5$, equation (40) evaluates to

$$B_D Q_D B_D^T = 0.5h^2(BQ_D B^T + 0.5h(ABQ_D B^T + BQ_D B^T A^T) + h^2(ABQ_D B^T A^T/2)) \qquad (79)$$

The right side of equation (76) truncated at 2nd order is

$$h(BQB^T + 0.5h(ABQB^T + BQB^T A^T) + h^2(ABQB^T A^T/3 + (A^2 BQB^T + BQB^T (A^T)^2)/6)) \qquad (80)$$

By direct inspection, it is seen that the right side of equation (79) will approximate expression (80) with local error on the order of h^2, if $Q_D = 2Q/h$. This compares to a local integration error on the order of h^3.[6]

4th Order Runge-Kutta

For notational convenience let

$$\Lambda = BQB^T$$

$$\text{and} \quad \Lambda_D = BQ_DB^T$$

For the most commonly used 4th order Runge-Kutta integrator with $\alpha_1 = \alpha_4 = 1/6$; $\alpha_2 = \alpha_3 = 1/3$; $a_1 = a_2 = 0.5$; and $a_3 = 1$, equation (52) yields

$$\begin{aligned} B_DQ_DB_D^T = \frac{h^2}{3.6}\Bigg[& \Lambda_D + .5[A\Lambda_D + \Lambda_DA^T]h + \frac{1}{6.667}[A^2\Lambda_D + \Lambda_D(A^T)^2]h^2 \\ & + \frac{1}{3.333}[A\Lambda_DA^T]h^2 + \frac{1}{40.}[A^3\Lambda_D + \Lambda_D(A^T)^3]h^3 \\ & + \frac{1}{10.}[A^2\Lambda_DA^T + A\Lambda_D(A^T)^2]h^3 + \frac{1}{20.}[A^2\Lambda_D(A^T)^2]h^4 \\ & + \frac{1}{40.}[A^3\Lambda_DA^T + A\Lambda_D(A^T)^3]h^4 \\ & + \frac{1}{80.}[A^3\Lambda_D(A^T)^2 + A^2\Lambda_D(A^T)^3]h^5 \\ & + \frac{1}{160}[A^3\Lambda_D(A^T)^3]h^6 \Bigg] \end{aligned} \qquad (81)$$

The right side of equation (76) truncated at sixth-order is

$$\begin{aligned} h\Bigg[& \Lambda + .5[A\Lambda + \Lambda A^T]h + \frac{1}{6}[A^2\Lambda + \Lambda(A^T)^2]h^2 \\ & + \frac{1}{3}[A\Lambda A^T]h^2 + \frac{1}{24.}[A^3\Lambda + \Lambda(A^T)^3]h^3 \end{aligned}$$

$$\begin{aligned}
&+ \frac{1}{8}[A^2\Lambda A^T + A\Lambda(A^T)^2]h^3 + \frac{1}{20.}[A^2\Lambda(A^T)^2]h^4 \\
&+ \frac{1}{30.}[A^3\Lambda A^T + A\Lambda(A^T)^3]h^4 + \frac{1}{120}[A^4\Lambda + \Lambda(A^T)^4]h^4 \\
&+ \frac{1}{720}[A^5\Lambda + \Lambda(A^T)^5]h^5 + \frac{1}{72.}[A^3\Lambda(A^T)^2 + A^2\Lambda(A^T)^3]h^5 \\
&+ \frac{1}{144}[A^4\Lambda A^T + A\Lambda(A^T)^4]h^5 + \frac{1}{252}[A^3\Lambda_D(A^T)^3]h^6 \\
&+ \frac{1}{5040.}[A^6\Lambda + \Lambda(A^T)^6]h^6 + \frac{1}{336}[A^4\Lambda(A^T)^2 + A^2\Lambda(A^T)^4]h^6 \\
&+ \frac{1}{840}[A^5\Lambda A^T + A\Lambda(A^T)^5]h^6\Big]
\end{aligned} \tag{82}$$

A comparison of the right side of equation (81) to expression (82) reveals that equation (81) will approximate expression (82), with local error on the order of h^2, if

$$Q_D = \frac{3.6Q}{h}$$

Note that a 4th order Runge-Kutta has an error of the same order in h as a 2nd order Runge-Kutta. Based on this observation, one may ask if this means that a 2nd order Runge-Kutta should be preferred over a 4th order Runge-Kutta for efficiency versus accuracy reasons. Certainly not, for two reasons. First, the 4th order Runge-Kutta is still preferred for the deterministic state equations and for the first-order statistics of the non-deterministic states. Second, although the two integrators have covariance errors on the same order in h, the absolute error of the 4th order Runge-Kutta will be smaller than that of the 2nd order Runge-Kutta. To see this, consider that the absolute error in the 4th order integrator is a term by term difference between equation (81) and equation (76). Likewise, the absolute error in the 2nd order integrator is a term by term difference between equation (79) and equation (76). For comparison purposes, expression (81) contains all the useful information of equation (76) because both methods are equally in error above order six. Since the second-order integrator contains no information above order two and the 4th order integrator only deviates by a fractional amount in each common term through order six, it is concluded that the 4th order method has a smaller absolute error.

4th Order Adams-Bashforth Integrator

From equation (70), the scalar case of the 4th order Adams-Bashforth integrator provides

$$B_D^2 Q_D = h^2 B^2 Q_D [1 - 6.34hA + 18.9(hA)^2 - 60.9(hA)^3] \qquad (83)$$

A third-order truncation of the right side of equation (76) for the scalar case yields

$$hB^2 Q[1 + hA + 2(hA)^2/3 + (hA)^3/3] \qquad (84)$$

Since equation (83) and expression (84) are scalar functions, the relationship between Q_D and Q can be determined directly. Doing so, yields

$$Q_D = \frac{Q[1 + hA + 2(hA)^2/3 + (hA)^3/3]}{h[1 - 6.34hA + 18.9(hA)^2 - 60.9(hA)^3]} \qquad (85)$$

Casual inspection of equation (85) shows that equation (24) is only valid for $hA \ll 1$ when applied to the Adams-Bashforth integrator.

Scalar Analyses

It will be beneficial to examine the above results for the scalar case. Restricting the equations to scalar systems will allow the results to be demonstrated graphically by defining an error equation and plotting it as a function of the parameter, hA. The parameter, hA, is the product of the step size, h, with the dynamic coefficient, A. For the scalar case, A is the reciprocal of the system time-constant. The step size is selected such that the ratio of h to the system time-constant is less than one, typically one-tenth. Therefore, hA is typically 0.1 or less.

A scalar case of equation (20) shows that, ideally, for each integrator considered, the normalization function should result in the equality

$$\begin{aligned} B_D^2 Q_D &= \int_0^h \Phi^2(h,\tau) B^2 Q d\tau \\ &= B^2 Q \int_0^h \epsilon^{2A(h-\tau)} d\tau \\ &= B^2 Q(\epsilon^{2hA} - 1)/2A \end{aligned} \tag{86}$$

In each of the cases presented in this chapter,

$$B_D^2 = h^2 B^2 f(hA)/\Gamma$$

where f(hA) was a polynomial in hA, and Γ was a constant used to make the zeroth coefficient of f(hA) equal to 1. Further, in order to make the relationship between Q_D and Q independent of the system dynamics, it was proposed that for each of the cases, with the possible exception of the

Adams-Bashforth integrator,

$$Q_D = \Gamma Q/h$$

Therefore,

$$B_D^2Q_D = B^2Qhf(hA) \qquad (87)$$

Substituting equation (87) into equation (86), provides the ideal relationship

$$2hAf(hA) = \epsilon^{2hA} - 1$$

In reality, this relationship will not be exact because

$$f(hA) \neq (\epsilon^{2hA} - 1)/2hA$$

Based on this realization, an error function can be defined and evaluated for each integrator. Let

$$V(hA) = 2hAf(hA) - \epsilon^{2hA} + 1 \qquad (88)$$

where $V(hA)$ is the error function. Note that $V(hA) = 0$ for $hA = 0$. This function can be plotted versus hA for each of the f(hA) functions derived in these analyses. A number of practical assumptions will be made in plotting $V(hA)$ for each of the integrators. First, it will be assumed that the dynamic system being simulated is stable; thus, $A < 0$. Next, the range of hA will be restricted to values that would likely be considered for use in a simulation; specifically, $0 \leq |hA| \leq 0.2$.

Euler Integrator

From equation (77), for the Euler integrator, it is seen that $f(hA) = 1$. Figure 2 provides a plot of equation (88) evaluated with $f(hA) = 1$. Note, that using equation (24) to set the random number generator's covariance would result in less than a 10% error in the computed discrete covariance.

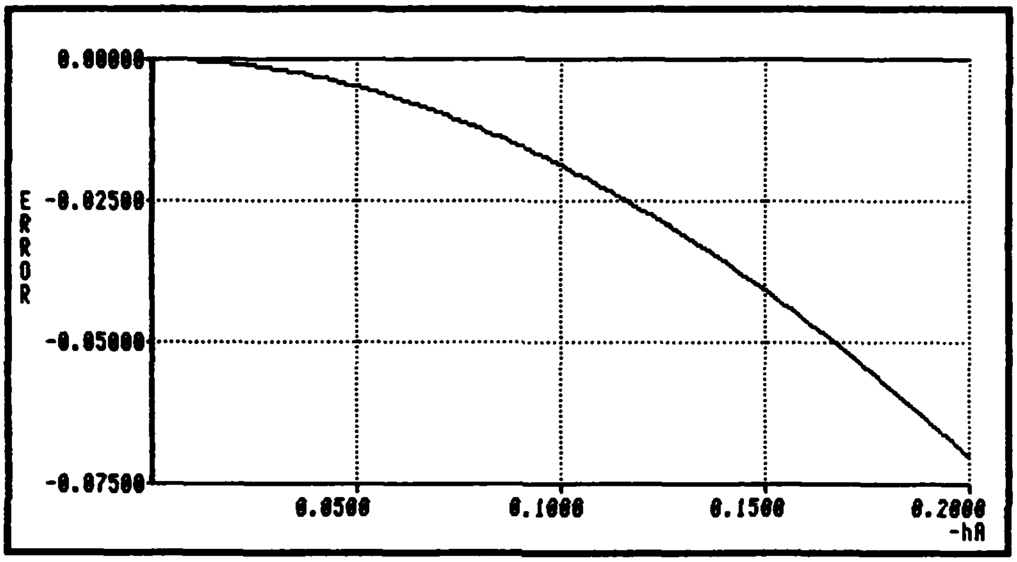

Figure 2. Error Function For Euler Integrator using QD = Q/h

2nd and 4th Order Runge-Kutta Integrators

For the 2nd order Runge-Kutta integrator, equation (79) provides

$$f(hA) = 1.0 + hA + 0.5(hA)^2$$

Likewise, for the 4th order Runge-Kutta integrator, equation (81) provides

$$f(hA) = 1.0 + hA + (3/5)(hA)^2 + (1/4)(hA)^3 + (1/10)(hA)^4 + (1/40)(hA)^5 + (1/160)(hA)^6$$

Figure 3 provides a comparison plot of $V(hA)$ evaluated with these two f(hA) functions. Note that, as predicted in the error analyses, the order of magnitude of the error is the same for the two methods and the absolute error of the 4th order Runge-Kutta integrator is smaller than that of the 2nd order Runge-Kutta integrator. However, both integrators provide accurate results for the second-order statistics when the proper statistical relationship is used.

Adams-Bashforth Integrator

There are several choices for the Q_D to Q normalizing function. Consider two choices. First, consider the one proposed by Griffith where $Q_D = Q/h$. The advantage of this choice is that the normalizing function is independent of the system dynamics. With this normalizing function,

$$f(hA) = 1.0 - 6.34hA + 18.9(hA)^2 - 60.9(hA)^3$$

The second choice would be to use

$$Q_D = Q/(h(1.0 - 6.34hA + 18.9(hA)^2 - 60.9(hA)^3))$$

and

$$f(hA) = 1.0$$

The obvious disadvantage of this choice is that the Q_D to Q normalizing function is dependent on the system dynamics. However, for analysis purposes, this choice must be considered. A comparison plot for these two f(hA) functions is given in Figure 4. Note from Figure 4, that the use of $Q_D = Q/h$ for the Adams-Bashforth integrator could result in errors on the order of 100 percent. Further, note that even when using the complex normalizing function, the errors are the same as would be obtained from using an Euler integrator. Besides this sensitivity to step size, there are other problems with the use of the Adams-Bashforth integrator for stochastic analyses. These other problems will be demonstrated in Chapter V.

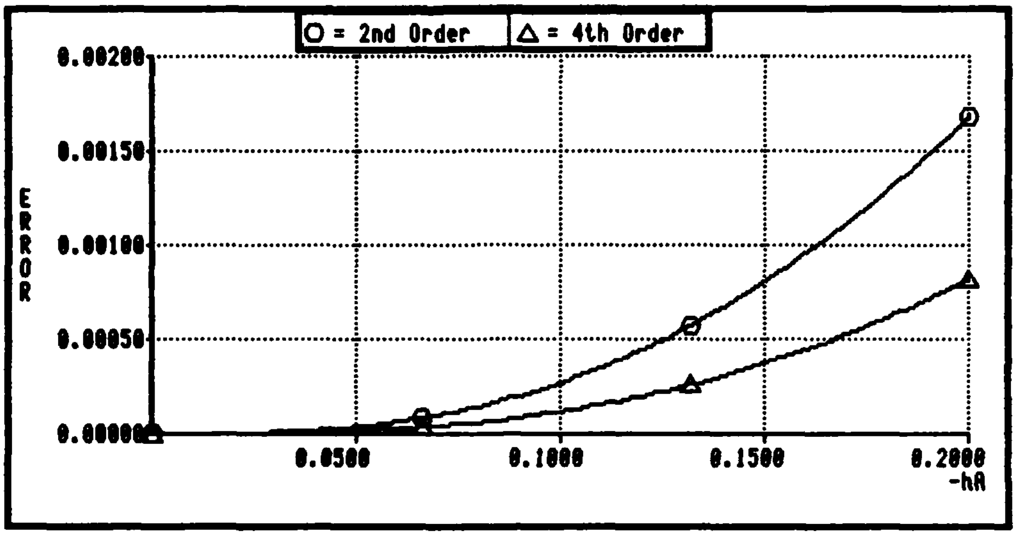

Figure 3. Error Functions for 2nd & 4th Order Runge-Kutta Int.

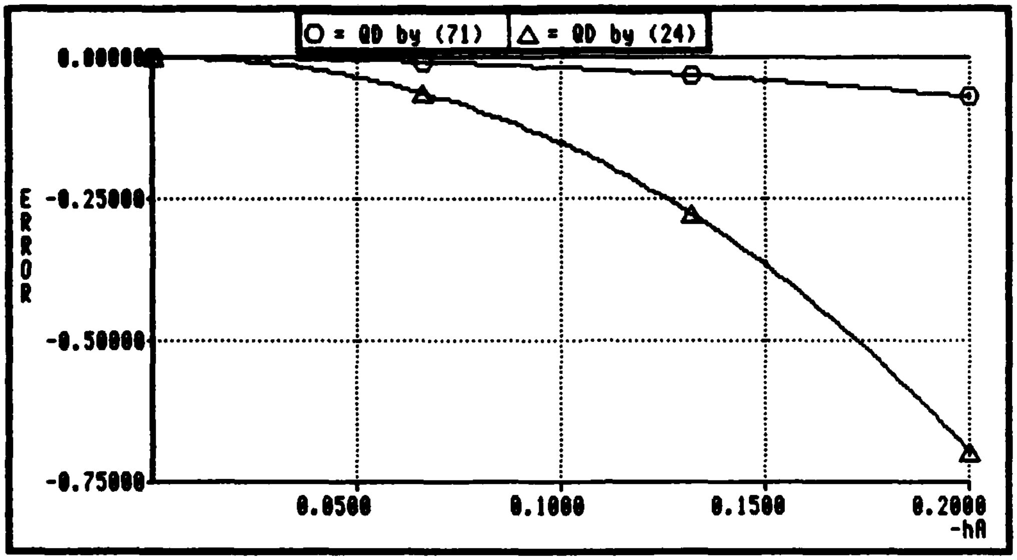

Figure 4. Error Function For Adams-Bashforth Integrator

V. NUMERICAL AND FURTHER ANALYSES

To demonstrate the significance of the findings in Chapters III and IV, a first-order linear differential stochastic equation will be analyzed numerically. Limiting the example to a linear scalar system will allow the covariance to be conveniently expressed analytically and the numerical results compared directly. For the example, the continuous system is described by

$$\dot{x}(t) = -2x(t) + w(t) \tag{89}$$

where w(t) is a normally distributed white random process with a spectral density equal to σ^2. For the example let $\sigma^2 = .01$.

Equation (14) will be used to analytically determine P(t). At the outset, assume that $t_0 = 0$ and $P(t_0) = 0$. With these assumptions, the scalar form of equation (14) applied to this system is

$$P(t) = \int_0^t \Phi^2(t,\tau)B^2Q(\tau)d\tau$$

where $\Phi(t,\tau) = \epsilon^{-2(t-\tau)}$; B = 1; and $Q(\tau) = \sigma^2$. Evaluating the right side of the covariance equation provides

$$P(t) = \sigma^2(1 - \epsilon^{-4t})/4 \tag{90}$$

To numerically analyze this system, a Monte Carlo simulation was written in FORTRAN 77 and executed using single precision arithmetic on a VAX 780 computer. One thousand executions were made per Monte Carlo simulation. The simulation was written so the integrator would be a module interfaced with the rest of the simulation through parameter I/O. Each integrator of interest was then coded as a module and incorporated in the simulation as needed. Thus, the only variable from one simulation program to another was the explicit integrator under investigation. A random number generator proposed by Marse and Roberts [9] was used to generate a uniform distributed random sequence. The uniformly distributed sequence was then shaped into an approximate Gaussian distributed sequence by direct application of the Central Limit Theorem involving the summation of twelve samples from the random number generator.[1,2] The random number generator routine is given in Appendix B.

Fixed-Step Method Results

Figures 5 through 10 show the results from the Monte Carlo analyses for various fixed-step integrators. The integrators were selected to exercise and test the analytical result given by equation (57) for a number of frequently used Runge-Kutta algorithms. Table 2 provides a summary of the methods tested as well as the step size and normalization function used in each test.

Table 2

Summary of Figures Providing Fixed-Step Results

FIGURE NUMBER	INTEGRATOR METHOD	STEP SIZE (seconds)	NORMALIZING FUNCTION
5	EULER	0.005	$Q_D = Q/h$
6	2nd Ord R-K	0.005	$Q_D = 2Q/h$
7	4th Ord R-K	0.005	$Q_D = 3.6Q/h$
8	4th Ord R-K	0.100	$Q_D = 3.6Q/h$
9	4th Order R-K-Gil	0.050	$Q_D = 18Q/7h$
10	R-K (5,6)	0.050	$Q_D = 3.54Q/h$

The step size used to generate the data in Figures 5, 6, and 7 was selected to enforce the original assumption that the state transition matrix would approximate the identity matrix. At a step size of 0.005 seconds, the state transition matrix for the system described by equation (89) is approximately equal to 0.99 over one step. The plots clearly demonstrate the validity of equation (57). The results also show that the use of $Q_D = Q/h$ for any Runge-Kutta method higher than first-order will introduce significant, and unnecessary errors into the simulation.

The error analyses in Chapter IV predicted that the normalization function for the Runge-Kutta methods would be fairly insensitive to a relaxation of the step size restrictions imposed by the assumption that the state transition matrix approximate the identity matrix. To test that prediction, the 4th order Runge-Kutta integrator-based simulation was re-run at a large step size of 0.1 seconds. Figure 8 shows the results from this test. It should be

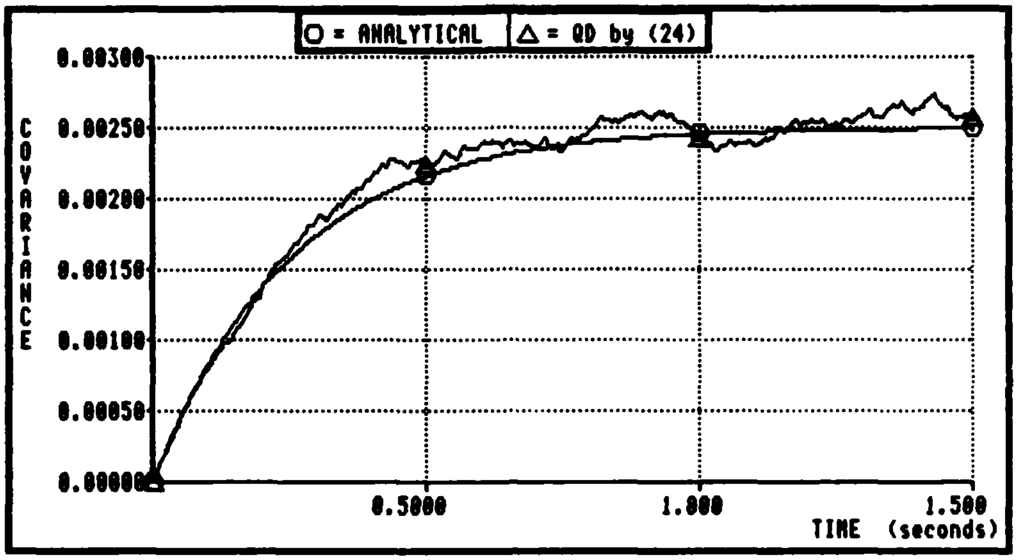

Figure 5. Euler Integrator Analysis

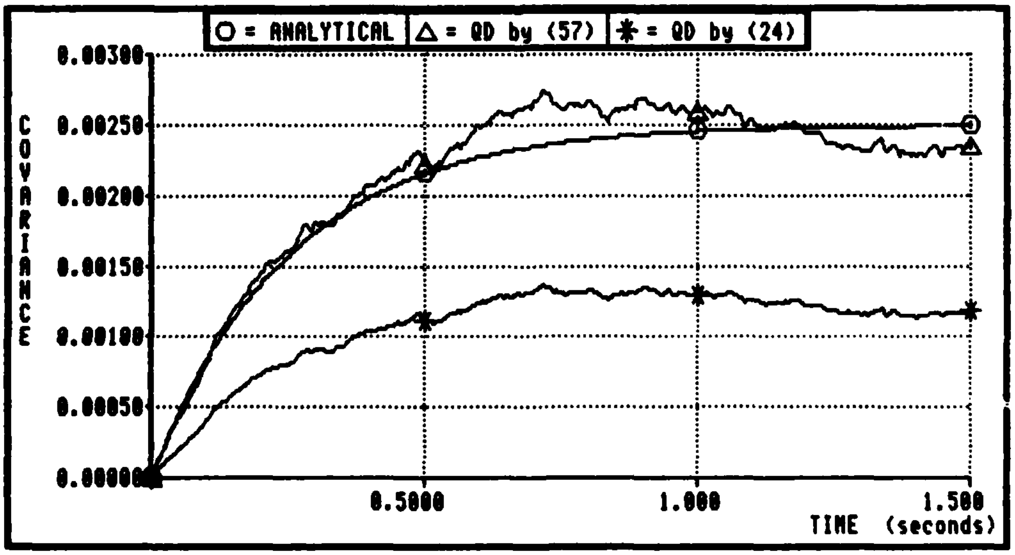

Figure 6. 2nd Order Runge-Kutta Analysis

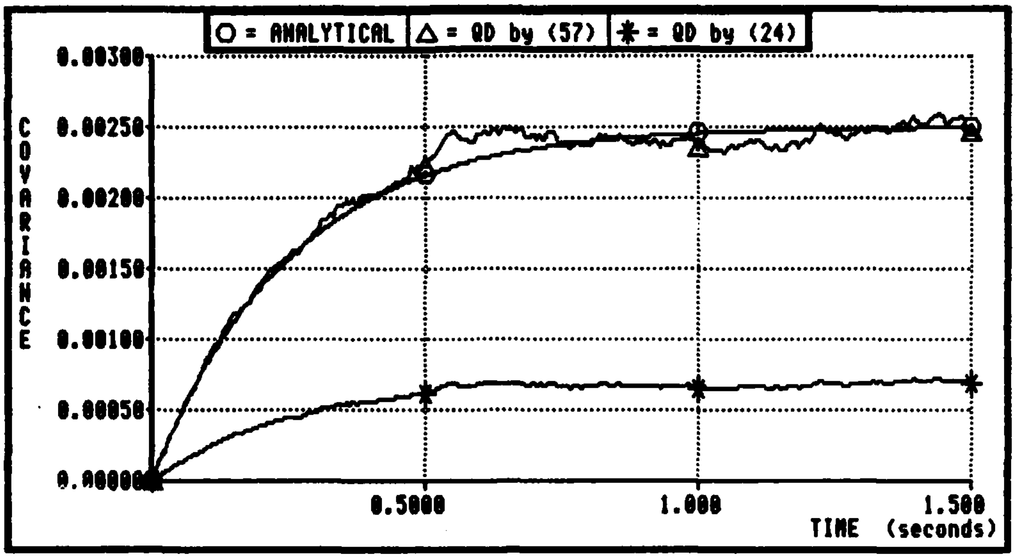

Figure 7. 4th Order Runge-Kutta Analysis

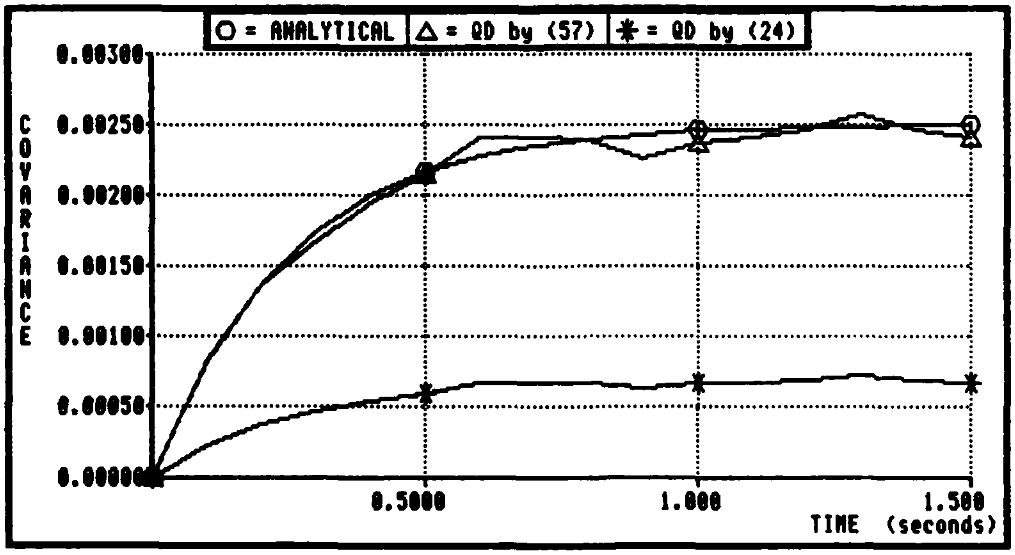

Figure 8. 4th Order Runge-Kutta Analysis (h = 0.1 seconds)

noted, that although this seems to be a large step size, for this example,

$$|\Phi_D(0.1,0) - \Phi(0.1,0)| < 3*10^{-6}$$

where $\Phi(\cdot,\cdot)$ is given by equation (12) and $\Phi_D(\cdot,\cdot)$ is given by equation (49). As seen by the plot, the normalization function still provides accurate numerical results.

Figure 9 shows the results for a Runge-Kutta-Gil method.[5] The Runge-Kutta-Gil method is a 4th order method that uses a non-conventional algorithm for calculating the intermediate update vectors. Gil derived his procedure to minimize finite word-length errors. The algorithm was tested to insure that equation (57) would hold for this nonconventional method. Clearly it does.

Figure 10 shows the results for another non-conventional method. This method is called a Runge-Kutta (5,6). [5] It is a sixth-order method and thus makes six evaluations of the derivative functions during each update. However, it only explicitly uses five of the intermediate update vectors to propagate the states over the interval. This algorithm, as well as the Runge-Kutta (7,8) method, is often used in situations needing high accuracy, such as the solution to orbital equations of motion. The correction factor, Γ, for the Runge-Kutta (5,6) integrator is given in Chapter III, Table 1. Figure 10 shows, once again, that equation (57) is a valid general solution to the problem.

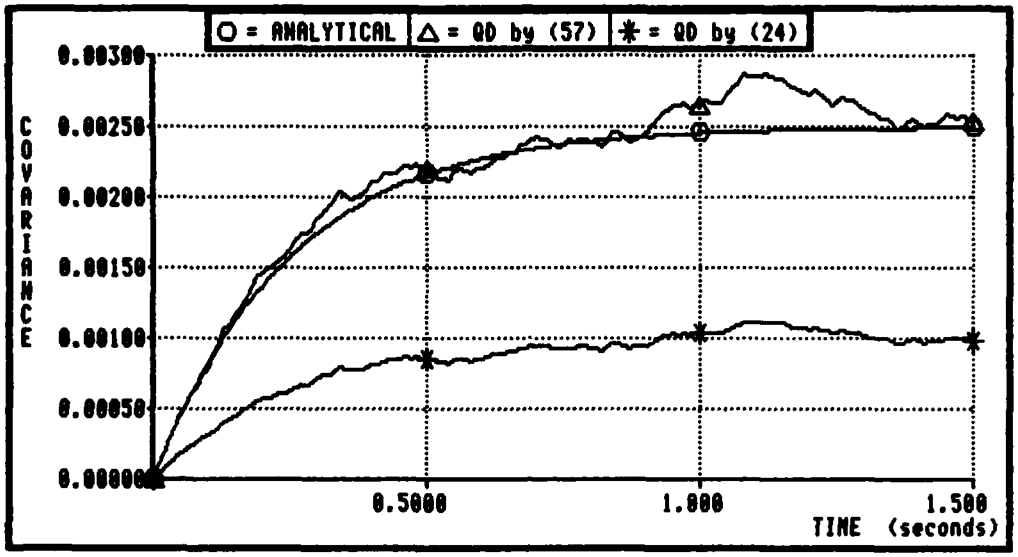

Figure 9. 4th Order Runge-Kutta-Gil Analysis

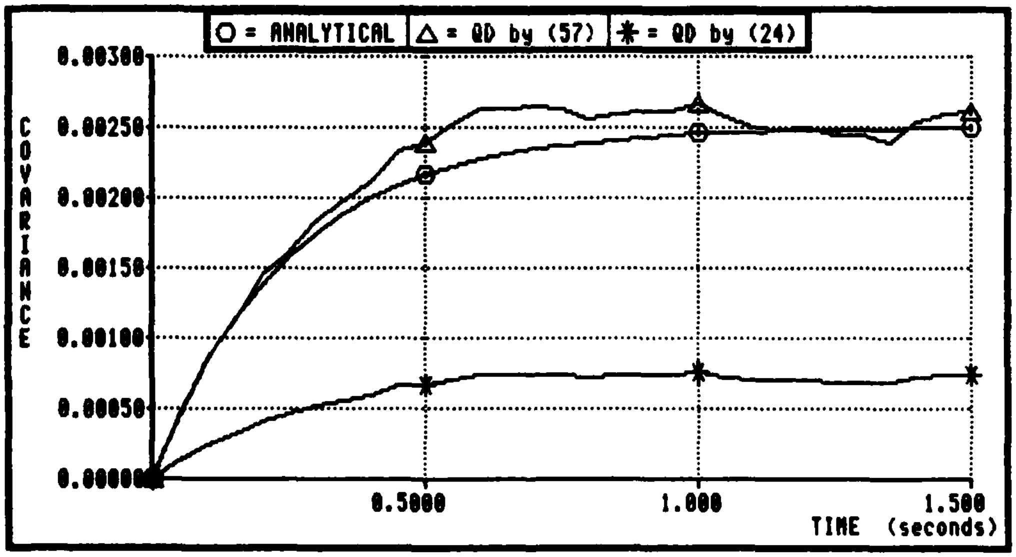

Figure 10. Runge-Kutta (5,6) Analysis

Adams-Bashforth Method

The Adams-Bashforth integrator selected for evaluation is a fourth-order method and is given by equation (68). The integrator was first tested at an integration step of 0.005 seconds. Equation (71) was used to determine the correct value of the discrete covariance. The results of this numerical analysis are provided in Figure 11. Figure 11 shows a comparison between the calculated state covariance when equation (71) was used, the calculated state covariance when equation (24) was used, as suggested by Griffith, and the analytical solution to the state covariance given by equation (90). Figure 11 demonstrates that equation (71) provides greater accuracy over equation (24). Examination of equation (71) shows that as the step size approaches zero, equation (71) will approach equation (24). Figure 12 demonstrates this relationship and provides validity to Griffith's numerical findings. For the case presented in Figure 11, the step size was selected to be 0.001 seconds.

An interesting problem occurred with the Adams-Bashforth integrator. This problem occurs only in stochastic analyses. Recall that the Adams-Bashforth method, as well as all multistep methods, are not self-starting. To get an nth order multistep integrator started, computation of the derivative function over the first (n-1) steps must be accomplished independent of the multistep integrator.

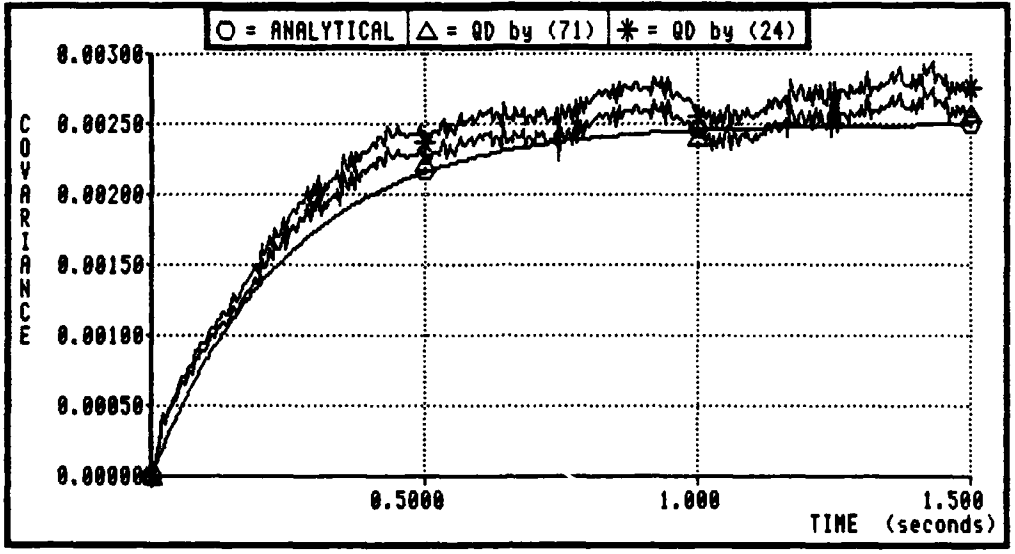

Figure 11. 4th Order Adams-Bashforth Analysis (h = 0.005 seconds)

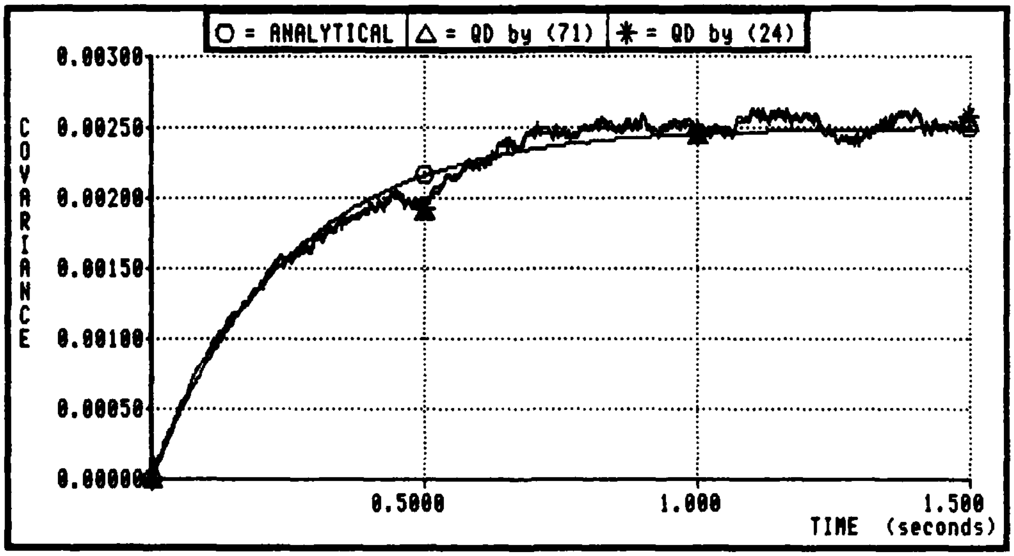

Figure 12. 4th Order Adams-Bashforth Analysis (h = 0.001 seconds)

Often a 4th order Runge-Kutta method is used to start a 4th order Adams-Bashforth integrator. The order of the two methods is chosen to be the same so that the numerical accuracy is of the same order of magnitude. The problem that occurs in stochastic analyses using the Adams-Bashforth method is due to the starting procedure.

In order for the starting integrator to provide accurate statistics over the starting interval, the covariance of the random number generator must be set with respect to the starting integrator. For instance, if a 4th order Runge-Kutta algorithm is used to start the multistep integrator, Q_D must be related to Q by $Q_D = 3.6Q/h$ over the starting interval. Once the four evaluations have occurred, control is turned over the Adams-Bashforth integrator and Q_D is then set in accordance with equation (71). The problem occurs during the first four updates made by the Adams-Bashforth integrator. The noise terms used in the evaluation of the derivative functions during the starting interval are mismatched to the noise terms needed by the Adams-Bashforth for accurate propagation of the states. This mismatch causes a large transient spike and introduces an error in the state calculation that propagates in time. Figure 13 illustrates this phenomenon. This problem appears only in the calculation of the stochastic states.

To remedy the problem, a number of potential solutions was tried resulting in varying degrees of success. The necessary requirements for an adequate solution were found to be that the starting procedure must provide accurate updates at the selected step size and the noise normalization function used by the starting integrator must be approximately the same as the one used by the Adams-Bashforth. The method that finally provided good results was a method involving the use of a number of different integration methods. At the first update an Euler method was used, followed by a 2nd order Adams-Bashforth integrator at the second update, followed by a 3rd order Adams-Bashforth integrator at the third update and finally the 4th order Adams-Bashforth integrator at the 4th and subsequent updates. The data used to construct Figures 11 and 12 were determined with the 4th order Adams-Bashforth integrator started by this method.

The error analyses in Chapter IV predicted that the Adams-Bashforth integrator would be sensitive to a violation of the assumption that the state transition matrix would approximate the identity matrix over the integration interval. To test this prediction, the Adams-Bashforth simulation was run at a step size of 0.025 seconds. At this step size $\Phi(h,0) \approx 0.95$. This step size was chosen because the total execution time realized for the simulation was 138 seconds which is slightly longer than that realized running the 4th order Runge-Kutta simulation (126 seconds) with a step size of Ø.1 seconds. Figure 14 shows the results from this test and verifies the prediction. A Comparison of Figure 14 to Figure 8 and considering equivalent execution times, indicates that there is no benefit in using an Adams-Bashforth integrator for stochastic simulations.

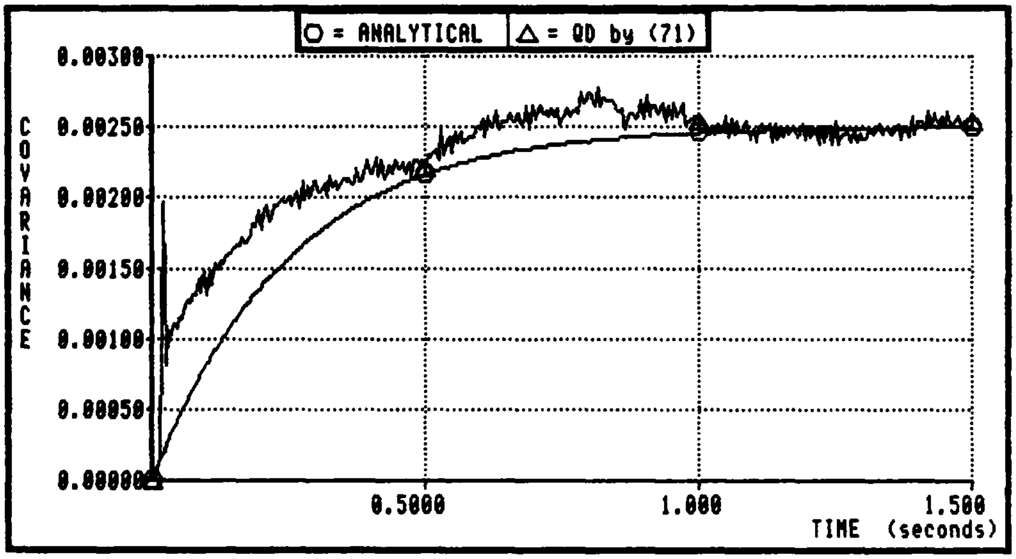

Figure 13. 4th Order Adams-Bashforth with Runge-Kutta Starter

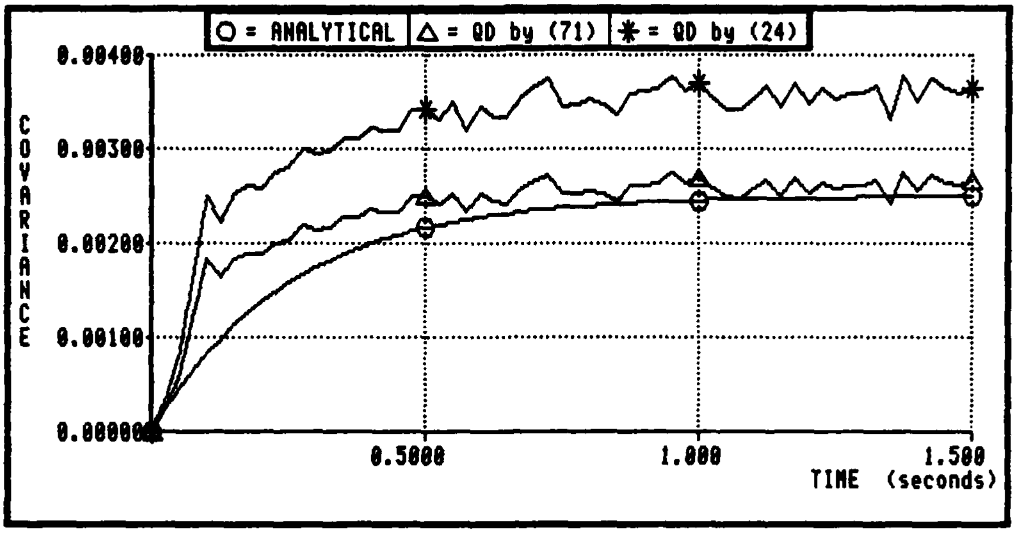

Figure 14. Adams-Bashforth Analysis (h = 0.025 seconds)

Other Integrators of Interest

Thus far, all of the analyses have dealt with conventional integrators. A number of more complex integration methods are frequently used in special situations. These more complex methods are usually selected in order to obtain highly accurate solutions at moderate computational costs. Two types of "advanced" algorithms that are of interest to this study are variable-step-size methods and predictor-corrector methods.

Variable-step-size Methods

Variable-step-size methods are useful for simulating systems that contain time-varying dynamics. In any simulation, the step size should be selected to handle the fastest dynamics in the system. However, if the dynamics are fast over one interval and then slow over another interval, a step size selected for the fast interval will cause wasted computations during the slow interval. Variable-step-size methods address this problem by computing some function that senses the accuracy of solution and adaptively adjusts the step size so the errors remain bounded below some defined tolerance level. Application of variable-step-size methods to stochastic systems presents some unique problems not encountered with deterministic systems.

As established in this dissertation, the magnitude of the noise added to the system states is a function of the step size. This fact translates into effectively discounting all variable-step-size methods based on multistep integration methods because a step size adjustment would require a total restart of the integrator using new noise terms. Therefore, at the outset, the search for candidate variable-step-size methods for this analysis will be restricted to Runge-Kutta methods. As seen in the analyses presented in Chapter III, each fixed-step integrator has a unique normalizing function for setting the random number generator's covariance. This fact makes the application of methods similar to the Runge-Kutta-Fehlburg integrator difficult to manage.

The Runge-Kutta-Fehlburg integrator is an efficient algorithm that calculates the state update by solving a fourth-order Runge-Kutta integrator and a fifth-order Runge-Kutta integrator in parallel. The fourth-order solution is used to propagate the states and the fifth-order solution is used to predict the error of the fourth-order solution. The efficiency of the Fehlburg method is found in the intermediate derivative evaluations. A fourth-order Runge-Kutta method requires four function evaluations and a fifth-order Runge-Kutta method requires six function evaluations. Therefore, a brute force implementation of a fourth and a fifth-order method would require ten function evaluations. Fehlburg derived a set of coefficients that would allow the fourth-order update formula to use the same derivative function evaluations as the fifth-order update; thus the entire process only requires six function evaluations as compared to ten by the brute force method.[6]

The difficulty with applying Fehlburg's algorithm to stochastic systems is due to the need to use one noise covariance for the fourth-order update and another for the fifth-order update. This means that the two update formulas can no longer share the derivative function evaluations. The requirement of ten function evaluations per step makes the computational costs prohibitively large. Any variable-step-size method that uses two update methods of different order to calculate the error function would introduce this same problem.

A variable-step-size method that does not have any of the above problems is one presented in reference 5. This method was developed by Colatz and is based entirely on a 4th order Runge-Kutta integrator. The error formula that Colatz derived is a function of the intermediate function evaluations. Specifically, the error function, ϵ, is

$$\epsilon = \left| \frac{K_2 - K_3}{K_1 - K_2} \right| \qquad (91)$$

wh-re the K_i's are given in equation (43). Colatz showed that the error function given by equation (91) should be less than a few hundredths in order to insure an accurate solution. The only difficulty in applying Colatz's formula was the situation where all the K_i's are nearly zero. This situation occurs in steady-state. Explicit implementation of equation (91) in computer code will cause numerical errors in this situation due to a division by small numbers. To circumvent that problem, equation (91) was implemented in the following form.

$$|K_2 - K_3| - \epsilon|K_1 - K_2| \leq 0 \qquad (92)$$

where ϵ was set to 0.01 . When equation (92) was used to predict the maximum step size for a deterministic system with the same dynamics as in equation (92), it predicted that $h < 0.1$ seconds. This is certainly a reasonable finding. However, when equation (92) was applied to the stochastic system given by equation (89), the results were clearly erroneous. Figure 15 is a plot of the left side of equation (92) versus time for the stochastic system solved at a step size of 0.005 seconds. As seen from equation (92), the evaluated expression must be less than or equal to zero in order to meet the accuracy condition. Figure 15 shows that this condition is never met indicating a need to reduce the step size. Although other analyses have shown that this step size is well within acceptable limits, the step size was reduced an order of magnitude to an $h = 0.0005$ seconds. Figure 16 shows the computed error function at this step size. Clearly, from Figures 15 and 16, the addition of noise to the system makes Colatz's algorithm useless. This is really not surprising when one considers that equation (91) is a function of stochastic state information, therefore equation (91) is stochastic. To gain useful information from this stochastic equation, a stochastic analysis would have to be performed. However, note that equation (91) is nonlinear making its statistics non-Gaussian and its analysis difficult. An alternative is to use deterministic state information in equation (91) or to use another approach to estimating the required step size. Regardless, this study strongly suggests that any error function used to make decisions with regards to step size adjustment must only be based on deterministic information.

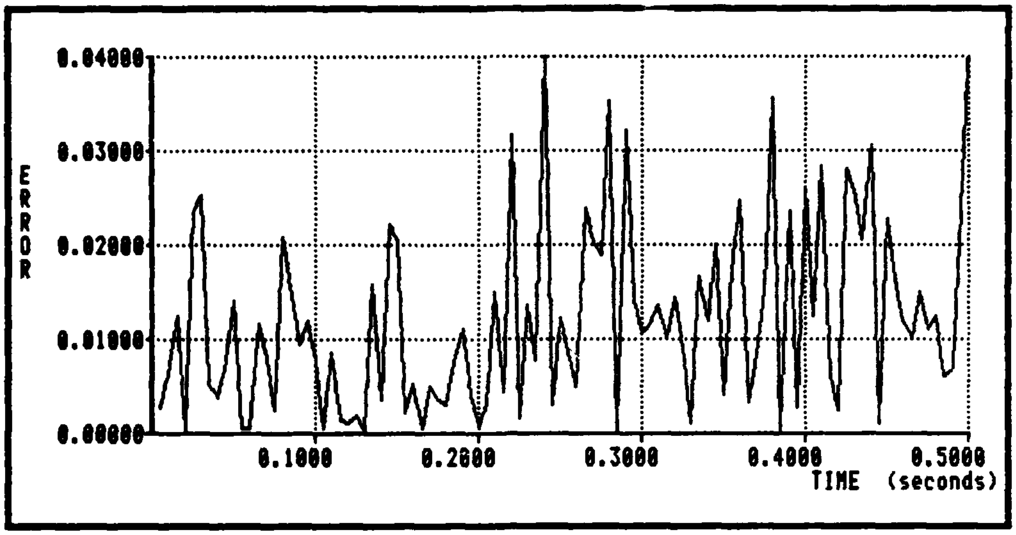

Figure 15. Error Function of Stochastic State (h=0.005 seconds)

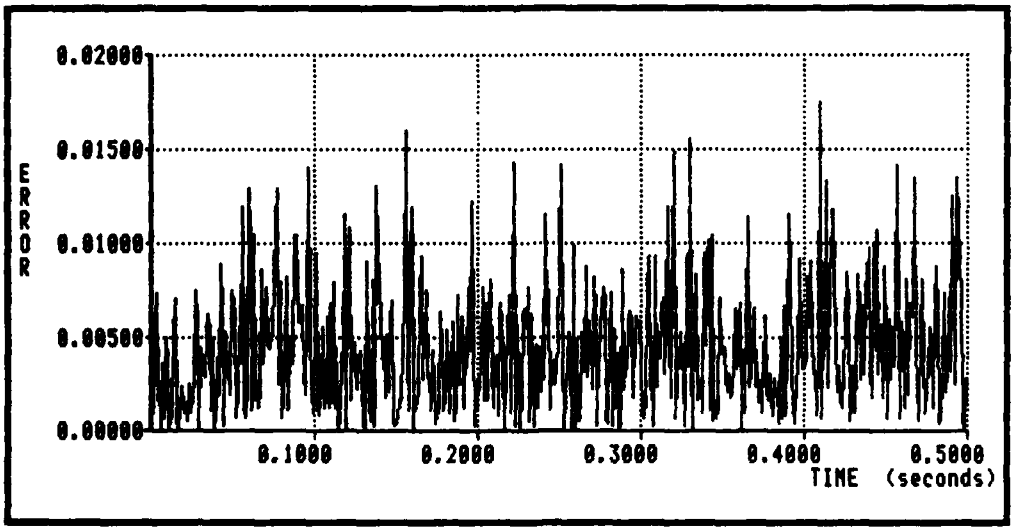

Figure 16. Error Function of Stochastic State (h=0.0005 seconds)

Adams-Moulton Method [6,8]

The Adams-Moulton method is a corrector formula for use in a predictor-corrector integration algorithm. It is probably the most widely used corrector method. Predictor-corrector formulas are iterative methods used to increase the accuracy of solution at relatively low computational costs. The Adams-Moulton formula requires derivative information at the next update increment in order to compute the state at that same increment. The predictor formula is used to provide that derivative information. The predictor-corrector pair can iterate on the solution for as many times as necessary. The iteration process can be terminated abruptly after some predetermined number of iterations or it can be terminated intelligently after reaching some defined convergence criterion. The equation that describes the Adams-Moulton corrector formula belongs to the class of multistep integration methods. Thus, it only requires one additional function evaluation per iteration. The specific equation is

$$\underline{x}_{i+1}^{(k)} = \underline{x}_i + h[\alpha_1 \underline{f}(\underline{x}_{i+1}^{(k-1)},(i+1)h) + \alpha_2 \underline{f}(\underline{x}_i,ih) + \alpha_3 \underline{f}(\underline{x}_{i-1},(i-1)h) + \alpha_4 \underline{f}(\underline{x}_{i-2},(i-2)h)] \quad (93)$$

where $\alpha_1 = 9/24$; $\alpha_2 = 19/24$; $\alpha_3 = -5/24$; $\alpha_4 = 1/24$; and k is the number of predictor-corrector iterations. Note that equation (93) is independent of the method used to obtain the prediction term, $\underline{x}_{i+1}^{(k-1)}$. Therefore, to analyze equation (93) on a per iteration basis, k can be set to 1.

Next, note that the noise properties of the corrector formula will only be an explicit function of four independent noise terms derived at the evaluation of $\underline{f}(\cdot,\cdot)$.

To analyze this system, the Jury method presented in Chapter III will be used. For this analysis, a difference equation will be required. A change of variable will be used to account for the iterations in equation (93).

Let $j = i + k = i + 1$. Using a scalar form of equation (10) to evaluate $f(\cdot,\cdot)$ in equation (90), then equation (93) will become

$$\begin{aligned} x_{j+1} = x_{j-1} + h[\alpha_1(Ax_j + Bw_j) + \alpha_2(Ax_{j-1} + Bw_{j-1}) \\ + \alpha_3(Ax_{j-2} + Bw_{j-2}) + \alpha_4(Ax_{j-3} + Bw_{j-3})] \end{aligned} \tag{94}$$

Taking the z-transform of equation (94), assuming $x_0 = 0$, yields

$$\begin{aligned} zX(z) = z^{-1}X(z) + h[\alpha_1(AX(z) + BW(z)) \\ + \alpha_2(AX(z) + BW(z))z^{-1} + \alpha_3(AX(z) + BW(z))z^{-2} \\ + \alpha_4(AX(z) + BW(z))z^{-3}] \end{aligned} \tag{95}$$

Multiplying through by z^3, collecting terms, and rearranging equation (95), results in

$$X(z) = \frac{(\alpha_1 z^3 + \alpha_2 z^2 + \alpha_3 z + \alpha_4)hBW(z)}{z^4 - \alpha_1 hAz^3 - (1 + \alpha_2 hA)z^2 - \alpha_3 hAz - \alpha_4 hA} \tag{96}$$

Applying equation (96) to the Jury analysis procedure given in Chapter III, in the same manner as equation (65), yields the steady-state expression

$$E[x^2] = \frac{0.5556(1 - .067hA + .012(hA)^2)h^2B^2Q_D}{V(hA)} \tag{97}$$

where

$$V(hA) = -1.333hA - 0.6111(hA)^2 + 0.0370(hA)^3 + 0.0069(hA)^4$$

Unresolved Inconsistency

V(hA) in equation (97) should equal $(1 - \Phi_D^2)$. It does not. When equation (97) was first developed, the numerator function was used to determine the Q_D to Q normalizing function without considering the inconsistency in the denominator function. To compound matters, the normalizing function worked well in the simulations. However, there is clearly an analytical error in this derivation. Oblivious to this error, the numerical analysis will be presented in order to show a number of important points regarding the application of the Adams-Moulton formula to stochastic analyses.

Numerical Findings

The numerator of equation (97) should equal $B_D^2Q_D$. Assuming it does, equating it to the scalar form of equation (20), and evaluating for small hA, leads to the relationship

$$Q_D = 1.8Q/h \qquad (98)$$

Figure 17 shows the results of applying equation (98) to a Adams-Moulton based simulation that uses a 4th order Adams-Bashforth predictor. To demonstrate that equation (98) is valid, independent of the predictor, Figure 18 shows the results of the Adams-Moulton simulation that uses a 4th order Runge-Kutta predictor. For the cases shown in

Figures 17 and 18, the algorithm was corrected once per update. Figure 19 shows the results of the simulation using the Adams-Bashforth predictor with 4 predictor-corrector iterations per update.

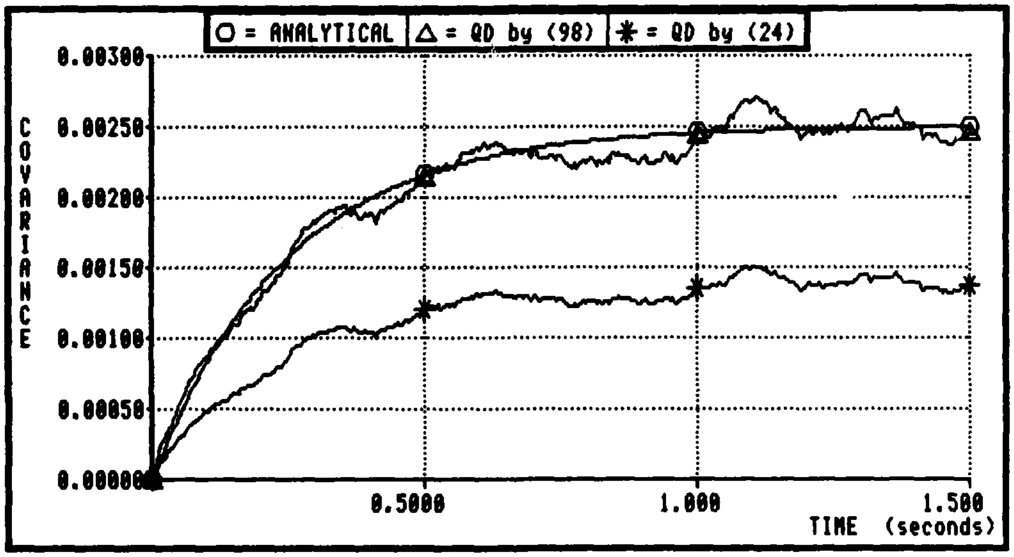

Figure 17. Adams-Moulton Corrector w/ Adams-Bashforth Predictor

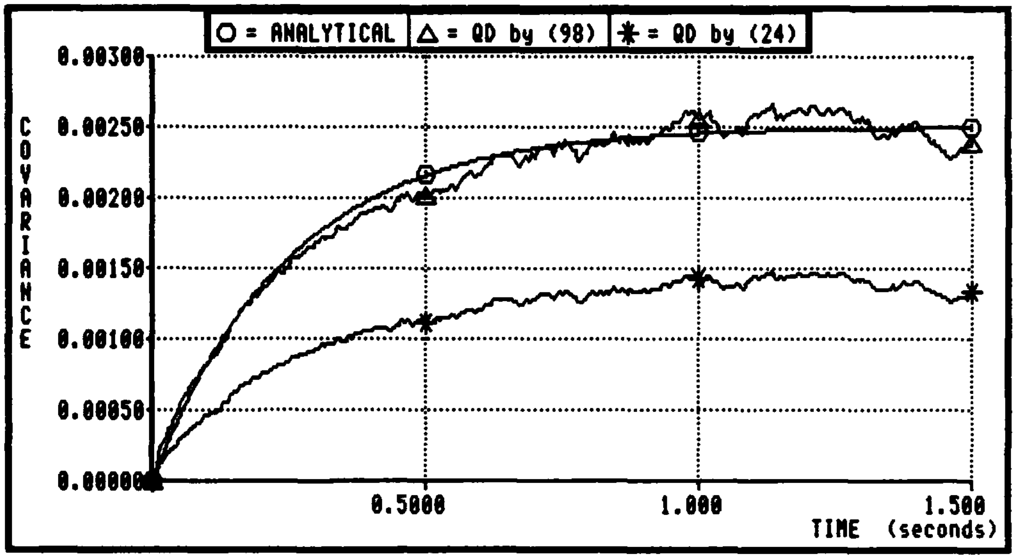

Figure 18. Adams-Moulton Corrector w/ Runge-Kutta Predictor

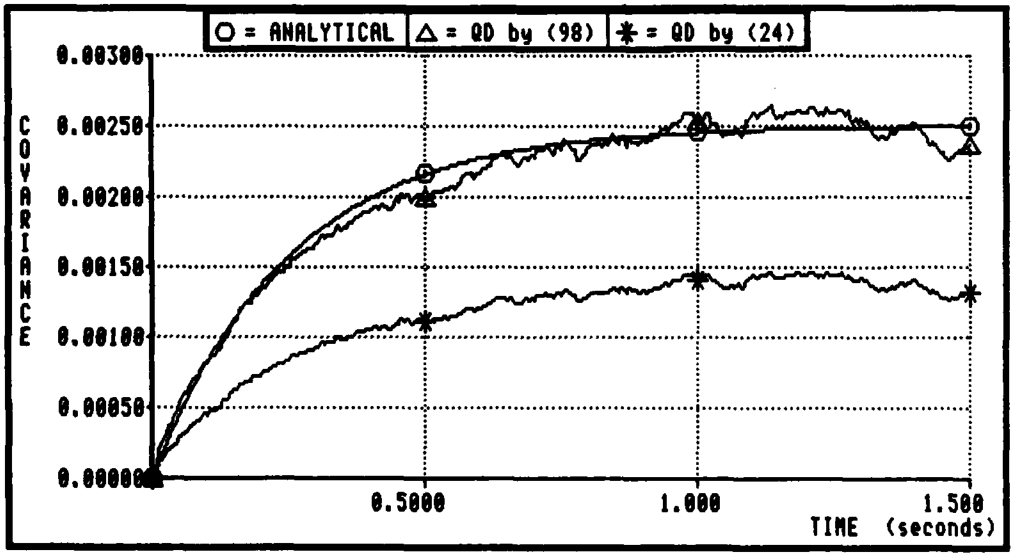

Figure 19. Adams-Moulton/Adams-Bashforth (4 Iterations)

VI. SUMMARY AND CONCLUSIONS

This dissertation addresses the problem of determining the correct relationship between the statistics of a continuous random process and the statistics of a discrete random process used to simulate the continuous random process. The findings of this research are directly applicable to the general field of digital simulation of physical systems described by ordinary differential equations.

It is shown that to ensure a faithful digital simulation of a continuous random process, the noise statistics of the random number generator must be set to values drastically different from the noise statistics of the continuous random process. Further, it is established that the relationship between the continuous and discrete statistics will be a function of the integration method used in the digital simulation.

The proper functional relationship between the discrete and continuous noise statistics was derived for

1. the class of Runge-Kutta integrators,
2. the 4th order Adams-Bashforth integrator, and
3. the Adams-Moulton corrector formula.

Additionally, the requirement for proper operation of a variable-step-size algorithm was developed.

The derived functional relationship between the discrete and continuous noise statistics for the class of Runge-Kutta integrators results in a unique normalizing factor for each Runge-Kutta method. The derivation provides a useful formula for calculating that factor. The use of the factor in the statistical relationship function will provide the proper setting of the random number generator's statistical parameters. The function is demonstrated to be accurate for five specific Runge-Kutta integrators.

It is shown, in the error analyses chapter, that all the integration methods considered yield normalization functions that, without simplifying assumptions, are dependent on the system dynamics. This dependency does, however, decrease with the integration step size. The rate of decrease varies for each integration method. It is shown that the derived normalization functions for the Runge-Kutta integrators are less sensitive to the system dynamics than the other examined methods.

In contrast to the Runge-Kutta methods, the derived function for the Adams-Bashforth integrator is the most sensitive to the system dynamics. The influence of the dynamics can be minimized by reducing the integration step size to a value much smaller than that required for deterministic analyses. However, the required step size for realizing an effective independence of the system dynamics is on the order of step size requirements for Euler integrators applied to deterministic systems. This extremely small step size requirement makes the Adams-Bashforth integrator impractical for stochastic simulations.

The Adams-Moulton formula's normalization function is derived independent of the predictor algorithm. It is shown that the derived function is valid when the Adams-Moulton corrector formula is used with a Runge-Kutta predictor as well as an Adams-Bashforth predictor. There is an analytical inconsistency in the Adams-Mouton derivation that has not been resolved. However, the numerical results show a number of useful findings for stochastic applications of this often used corrector formula.

Error functions needed to implement variable step size integration methods are discussed in detail. It is shown that when applied to a stochastic system, the error function for the method considered will be a stochastic function making its usefulness extremely limited. Numerical examples are presented to substantiate this finding.

Though this research investigated integration methods most commonly used in practice, it is certainly not exhaustive. However, the analysis procedures used in this dissertation are applicable to other integration methods that may be of interest. Without a doubt, the findings of this research establish the fact that the proper relationship between the statistics of the continuous random process and the statistics needed in the simulation for accurate modeling is explicitly dependent on the integration method used in the simulation. To ensure a faithful simulation, it is mandatory that the proper function be derived and validated for the specific integration method used in the simulation. Failure to do so will likely invalidate the simulation results.

REFERENCES

1. MAYBECK, PETER S.
STOCHASTIC MODELS, ESTIMATION, AND CONTROL , Volume 1, Academic Press, Inc., Orlando, FL (1984).

2. PAPOULIS, ATHANASIOS
PROBABILITY, RANDOM VARIABLES, AND STOCHASTIC PROCESSES , McGraw-Hill Book Company, New York, NY (1965).

3. BROWN, ROBERT G.
INTRODUCTION TO RANDOM SIGNAL ANALYSIS AND KALMAN FILTERING , John Wiley & Sons, New York, NY (1983).

4. GRIFFITH, JAMES M.
"Random Process Simulation Using a Continuous System Simulation Language". SIMULATION (October 1985). 193-195.

5. JAMES, M.L., SMITH, G.M., and WOLFORD, J.C.
APPLIED NUMERICAL METHODS FOR DIGITAL COMPUTATION, IEP - A Dun-Donnelly Publisher, New York, NY (1977).

6. CONTE, SAMUEL D. and deBOOR, CARL
ELEMENTARY NUMERICAL ANALYSIS, AN ALGORITHMIC APPROACH, McGraw-Hill, New York, NY (198Ø).

7. JURY, E.I.
THEORY AND APPLICATION OF THE Z-TRANSFORM METHOD, Robert E. Krieger Publishing Co., Huntington, NY (1973).

8. JOHNSTON, R.L.
NUMERICAL METHODS, A SOFTWARE APPROACH, John Wiley and Sons, New York, NY (1982).

9. MARSE, K. and ROBERTS, S.D.
"Implementing a Portable FORTRAN Uniform (Ø,1) Generator", SIMULATION, (October 1983) 135-139.

APPENDICES

APPENDIX A

IMPLEMENTATION PROCEDURE AND CODE FOR JURY ANALYSIS METHOD

Application of the Jury method for analyzing discrete integration algorithms of order three and higher is difficult without the aid of computational tools. The difficulty is due to the fact that the coefficients of the transfer functions that describe the integration processes are in variable form. For instance, consider the transfer function for the Adams-Bashforth integrator given by equation (69). The coefficients for the numerator polynomial, M(z), and the denominator polynomial, L(z), are

$$\begin{array}{ll} m_0 = 0 & l_0 = 1 \\ m_1 = \alpha_1 hB & l_1 = -(1 + \alpha_1 hA) \\ m_2 = \alpha_2 hB & l_2 = -\alpha_2 hA \\ m_3 = \alpha_3 hB & l_3 = -\alpha_3 hA \\ m_4 = \alpha_4 hB & l_4 = -\alpha_4 hA \end{array}$$

In the Jury algorithm given by equations (60) - (62), these coefficients will be used to form the matrices Ω and Ω_1 as in equation (62).

Recall from equation (61) that the steady-state value of the autocorrelation function will be determined by

$$E[x_1^2] = Q_D \frac{|\Omega_1|}{l_0 |\Omega|}$$

Thus, to solve the problem for the 4th order Adams-Bashforth integrator, the determinant operation must be accomplished for two fifth order square matrices. From equation (62), it is seen that the elements of Ω will be a linear combination of the coefficients l_0 through l_4.

Since the coefficients l_0 through l_4 are first-order polynomials in hA, the elements of Ω will also be first-order polynomials in hA. The determinant of Ω will, therefore, be a polynomial in hA of order five or less.

The matrix Ω_1 will be the same as Ω except for the first column. The elements in the first column of Ω_1 will be a combination of the square of the coefficients m_0 - m_4. Since all the coefficients of M(z) contain the common term hB, hB can be conveniently factored out. Therfore, Ω_1 will be a fifth-order square matrix with constants in the first column and first-order polynomials in hA in columns 2 - 5. Hence, the determinant of Ω_1 will be a polynomial in hA of order four or less.

Based on the above observations, it is expected that the steady-state covariance for the Adams-Bashforth integrator will have the following form

$$E[x_1^2] = h^2B^2Q_D\frac{|\Omega_1|}{l_0|\Omega|} = h^2B^2Q_D\frac{U(hA)}{V(hA)} \qquad (99)$$

where U(hA) is a polynomial in hA of order four or less and V(hA) is a polynomial in hA of order five or less.

The calculation of a determinant for a fifth order matrix of constants is tedious, but for a matrix of polynomials it is grueling task. However, Jury provides a set of equations that perform the determinant operation for the matrix given by equation (62). Though these equations ease the task, they only reduce the complexity slightly. The equations are lengthy and notationally involved. Because of the notational complexity in the equations, they were re-derived and validated to ensure there were no typographical errors. There were not. For brevity, Jury's equations will not be reproduced in this document. However, for those who are really interested, the computer code included in this appendix contains a direct implementation of Jury's equations (see Listing 1).

To determine U(hA) and V(hA), Jury's equations were coded in a computer program written in Pascal. The coefficients for L(z) and M(z) were coded as a function of the program variable hA. The program prompts the user for a value of hA, calculates the coefficients of L(z) and M(z), evaluates Jury's equations, and outputs the numerical value of U(hA) and V(hA). Evaluation of U(hA) and V(hA) at six distinct values of hA will provide enough data to uniquely determine the coefficients of U(hA) and V(hA). To provide a little extra confidence that the procedure was sound, seven points were actually used knowing that if the numerical data was valid, equivalent results would be obtained. For the Adams-Bashforth integrator, the results were

$$U(hA) = 1 - 5.6736hA + 9.5833(hA)^2 - 6.25(hA)^3 \tag{100}$$

$$\begin{aligned} V(hA) = {} & -2.0hA - 3.333(hA)^2 + 7.6736(hA)^3 \\ & -9.5833(hA)^4 + 6.25(hA)^5 \end{aligned} \tag{101}$$

The question that now arises is how do these results relate to the function needed to complete the analysis;

that is, $B_D^2Q_D$. Consider, that in steady-state, the state covariance equation is of the form

$$E[x_i^2](1 - \Phi_D^2) = B_D^2Q_D$$

which can be rewritten as

$$E[x_i^2] = \frac{B_D^2Q_D}{(1 - \Phi_D^2)} \qquad (102)$$

A comparison of equation (102) to equation (99) leads to the plausible relationships

$$B_D^2 = h^2B^2N(hA)U(hA) \qquad (103)$$

$$\text{and} \quad (1 - \Phi_D^2) = N(hA)V(hA) \qquad (104)$$

where N(hA) is a polynomial in hA that is common to both B_D^2 and $(1 - \Phi_D^2)$. To determine N(hA) additional information is required, such as Φ_D.

Derivation of Φ_D for the Adams-Bashforth Integrator

The discrete state transition matrix maps the states $\mathbf{x}_i$ into the states $\mathbf{x}_{i+1}$. Mathematically this is expressed as

$$\mathbf{x}_{i+1} = \Phi_D(h,0)\mathbf{x}_i \qquad (105)$$

where $\mathbf{x}_i = \mathbf{x}(t_i)$ and $\mathbf{x}_{i+1} = \mathbf{x}(t_i+h)$. In this derivation, all that is known about the specific problem will be used.

Consider the general 4th order Adams-Bashforth formula, where

$$\mathbf{x}_{i+1} = \mathbf{x}_i + h[\alpha_1\mathbf{f}(\mathbf{x}_i,t_i) + \alpha_2\mathbf{f}(\mathbf{x}_{i-1},t_{i-1}) + \alpha_3\mathbf{f}(\mathbf{x}_{i-2},t_{i-2}) + \alpha_4\mathbf{f}(\mathbf{x}_{i-3},t_{i-3})] \qquad (106)$$

where $\alpha_1 = 55/24$; $\alpha_2 = -59/24$; $\alpha_3 = 37/24$; and $\alpha_4 = -9/24$.

To develop the specific problem, consider

1. Restricting the problem to linear time-invariant differential systems given by

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) = \mathbf{f}(\mathbf{x},t) \qquad (107)$$

2. The state transition matrix is independent of the inputs, $\mathbf{u}(t)$. Therefore, let $\mathbf{u}(t) = 0$ for all t.

3. The continuous state transition matrix is given by

$$\mathbf{\Phi}(t_2,t_1) = \epsilon^{A(t_2 - t_1)} \qquad (108)$$

The derivation of the Adams-Bashforth formula makes the assumption that the present and past derivatives are known and are exact. The same assumption will be made in this derivation. With this assumption, equations (107) and (108) yield

$$\mathbf{f}(\mathbf{x}_i,t_i) = A\mathbf{x}_i$$
$$\mathbf{f}(\mathbf{x}_{i-1},t_{i-1}) = A\mathbf{x}_{i-1} = A\epsilon^{-Ah}\mathbf{x}_i$$
$$\mathbf{f}(\mathbf{x}_{i-2},t_{i-2}) = A\mathbf{x}_{i-2} = A\epsilon^{-2Ah}\mathbf{x}_i$$
$$\mathbf{f}(\mathbf{x}_{i-3},t_{i-3}) = A\mathbf{x}_{i-3} = A\epsilon^{-3Ah}\mathbf{x}_i$$

Evaluating equation (106) with these derivative functions, results in

$$\mathbf{x}_{i+1} = [I + \alpha_1 Ah + \alpha_2 Ah\epsilon^{-Ah} + \alpha_3 Ah\epsilon^{-2Ah} + \alpha_4 Ah\epsilon^{-3Ah}]\mathbf{x}_i \qquad (109)$$

Noting that equation (109) has the same structure as equation (105), it is concluded that

$$\mathbf{\Phi}_D(h,0) = I + \alpha_1 Ah + \alpha_2 Ah\epsilon^{-Ah} + \alpha_3 Ah\epsilon^{-2Ah} + \alpha_4 Ah\epsilon^{-3Ah} \qquad (110)$$

Notice that unlike the Runge-Kutta methods, the discrete state transition matrix for the Adams-Bashforth integrator is an infinite series. The question arises, does this state transition matrix approximate the Taylor series form of $\Phi(h,0)$? To answer this question, expand equation (110) in a series form ,collect terms and compare it to

$$\Phi(h,0) = \epsilon^{hA}$$

$$= I + Ah + (Ah)^2/2 + (Ah)^3/6 + (Ah)^4/24 + (Ah)^5/120 + \ldots \qquad (111)$$

Expanding the exponentials in equation (110) provides

$$\epsilon^{-hA} = I - hA + (hA)^2/2 - (hA)^3/6 + (hA)^4/24 + \ldots$$

$$\epsilon^{-2hA} = I - 2hA + 2(hA)^2 - 4(hA)^3/3 + 2(hA)^4/3 + \ldots$$

$$\epsilon^{-3hA} = I - 3hA + 9(hA)^2/2 - 9(hA)^3/2 + 27(hA)^4/8 + \ldots$$

Evaluating equation (110) using these expansions and the α_i's given in equation (106), results in

$$\Phi_D(h,0) = I + Ah + (Ah)^2/2 + (Ah)^3/6 + (Ah)^4/24 - 49(Ah)^5/120 + \ldots \qquad (112)$$

A comparison of equation (112) to equation (111) shows that $\Phi_D(h,0) \approx \Phi(h,0)$ with error on the order of h^5. With Φ_D in hand, the Jury analysis can be completed.

Determination of N(hA) and B_D^2

The general form of N(hA) is assumed to be

$$N(hA) = \gamma_0 + \gamma_1 hA + \gamma_2(hA)^2 + \gamma_3(hA)^3 + \ldots \qquad (113)$$

The γ_i coefficients in equation (113) can be computed directly from equation (104). Consider that

$$1 - \Phi_D^2 = -2hA - 2(hA)^2 - 4(hA)^3/3 - 2(hA)^4/3 + \ldots \qquad (114)$$

Taking the product of equation (113) with equation (101) and equating the resulting coefficients to the coefficients in equation (114) results in the following constraint equations

$$-2\gamma_0 = -2$$

$$-3.3333\gamma_0 - 2\gamma_1 = -2$$

$$7.6736\gamma_0 - 3.3333\gamma_1 - 2\gamma_2 = -4/3$$

$$-9.5833\gamma_0 + 7.6736\gamma_1 - 3.3333\gamma_2 - 2\gamma_3 = -2/3$$

and so on. Solving for γ_0 through γ_3 results in

$$N(hA) = 1 - 2(hA)/3 + 5.61(hA)^2 - 16.379(hA)^3 + \ldots \quad (115)$$

Solving for B_0^2 in equation (103) results in

$$B_0^2 = h^2B^2(1 - 6.34hA + 18.98(hA)^2 - 60.84(hA)^3 + \ldots) \quad (116)$$

Listing 1.

PASCAL Program Used to Evaluate Jury's Equations

```
Program Jury_Determinant;
   { Calculates U(hA) and V(hA) for the Adams-Bashforth
     integrator.  The coefficients 1Ø through 14 have been
     replaced by aØ through a4 to enhance readability }
var
   hA,aØ,a1,a2,a3,a4,e1,e2,e3,e4,e5,
   alfa1,alfa2,alfa3,alfa4,
   mØ,m1,m2,m3,m4,BØ,B1,B2,B3,B4 : real;
   QØ,Q1,Q2,Q3,Q4,BQ,AQ : real;

begin
   alfa1 := 55.Ø/24.Ø;
   alfa2 := -59.Ø/24.Ø;
   alfa3 := 37.Ø/24.Ø;
   alfa4 := -9.Ø/24.Ø;
               { Prompt User for input and Get It }
   write('Input hA ');  readln(hA);
               { Define Coefficients }
   mØ := Ø.Ø;
   m1 := alfa1;
   m2 := alfa2;
   m3 := alfa3;
   m4 := alfa4;
   aØ := 1.Ø;
   a1 := -(1.Ø + alfa1*hA);
   a2 := -alfa2*hA;
   a3 := -alfa3*hA;
   a4 := -alfa4*hA;
                { Jury's Equations }
   e1 := aØ + a2;
   e2 := a1 + a3;
   e3 := a2 + a4;
   e4 := aØ + a4;
   e5 := aØ + a2 + a4;

   BØ := mØ*mØ + m1*m1 + m2*m2 + m3*m3 + m4*m4;
   B1 := 2.Ø*(mØ*m1 + m1*m2 + m2*m3 + m3*m4);
   B2 := 2.Ø*(mØ*m2 + m1*m3 + m2*m4);
   B3 := 2.Ø*(mØ*m3 + m1*m4);
   B4 := 2.Ø*mØ*m4;

   QØ := aØ*(e1*e4 - a3*e2) + a4*(a1*e2 - e3*e4);
   Q1 := aØ*(a1*e4 - a2*a3) + a4*(a1*a2 - a3*e4);
   Q2 := aØ*(a1*e2 - a2*e1) + a4*(a2*e3 - a3*e2);
   Q3 := a1*(a1*e2 - e3*e4) - a2*(a1*e1 - a3*e3)
   + a3*(e1*e4 - a3*e2);
   Q4 := aØ*(e2*(a1*a4 - aØ*a3) + e5*(aØ*aØ - a4*a4))
   + (e2*e2 - e5*e5)*(a1*a1 - a1*a3 + (aØ - a4)*(e4 - a2));
```

Continuation of Listing 1.

```
            { BQ is U(hA) and AQ is V(hA) }
   BQ := aØ*BØ*QØ - aØ*B1*Q1 + aØ*B2*Q2 - aØ*B3*Q3 + B4*Q4;
   AQ := aØ*((aØ*aØ-a4*a4)*QØ - (aØ*a1 - a3*a4)*Q1
   + (aØ*a2 - a2*a4)*Q2 - (aØ*a3 - a1*a4)*Q3);

            { Output to terminal }
   writeln('For hA = ',hA);
   writeln('QØ = ',QØ,'  Q1 = ',Q1);
   writeln('Q2 = ',Q2,'  Q3 = ',Q3,'  Q4 = ',Q4);
   writeln('AQ = ',AQ,'  BQ = ',BQ);

            { Output to Printer }
   writeln(lst);
   writeln(lst,'For hA = ',hA);
   writeln(lst,'QØ = ',QØ,'  Q1 = ',Q1);
   writeln(lst,'Q2 = ',Q2,'  Q3 = ',Q3,'  Q4 = ',Q4);
   writeln(lst,'AQ = ',AQ,'  BQ = ',BQ);
   writeln(lst);
   writeln(lst);
end.
```

APPENDIX B

CODE FOR RANDOM NUMBER GENERATOR

USED IN SIMULATION

Listing 2.

FORTRAN Code of Random Number Generator

with Gaussian Shaping Algorithm

```
      REAL FUNCTION GAUSSN(SIG,SEED)
C
C******* FOR GOOD RESULTS USE AN INITIAL SEED = 824Ø64364
C
      INTEGER*4 SEED
      GNOIZ=Ø.
      DO 1Ø I=1,12
         GNOIZ = GNOIZ + URAND(SEED)
 1Ø   CONTINUE
      GAUSSN=SIG*(GNOIZ-6.Ø)
      RETURN
      END

      REAL FUNCTION URAND(SEED)
      INTEGER*4 B2E15,B2E16,MODLUS,HIGH15,HIGH31,LOW15,LOWPRD,
     &  MULT1,MULT2,OVFLOW,SEED
      DATA MULT1,MULT2/24112,26143/
      DATA B2E15,B2E16,MODLUS/32768,65536,2147483647/

      HIGH15 = SEED/B2E16
      LOWPRD = (SEED - HIGH15*B2E16)*MULT1
      LOW15  = LOWPRD/B2E16
      HIGH31 = HIGH15*MULT1 + LOW15
      OVFLOW = HIGH31/B2E15

      SEED = (((LOWPRD - LOW15*B2E16) - MODLUS) +
     &         (HIGH31 - OVFLOW*B2E15)*B2E16) + OVFLOW
      IF (SEED.LT.Ø) SEED = SEED + MODLUS

      HIGH15 = SEED/B2E16
      LOWPRD = (SEED - HIGH15*B2E16)*MULT2
      LOW15  = LOWPRD/B2E16
      HIGH31 = HIGH15*MULT2 + LOW15
      OVFLOW = HIGH31/B2E15

      SEED = (((LOWPRD - LOW15*B2E16) - MODLUS) +
     &          (HIGH31 - OVFLOW*B2E15)*B2E16) + OVFLOW
      IF (SEED.LT.Ø) SEED = SEED + MODLUS

      URAND = FLOAT(2*(SEED/256) +  1)/16777216.Ø
      RETURN
      END
```