ON-LINE HYBRID COMPUTER IMPLEMENTATION
OF STOCHASTIC FILTERING ALGORITHMS

By

WILLARD MORRIS HOLMES

Bachelor of Electrical Engineering

Auburn University

Auburn, Alabama

1960

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
MASTER OF SCIENCE
July, 1973

OKLAHOMA
STATE UNIVERSITY
LIBRARY

NOV 16 1973

ON-LINE HYBRID COMPUTER IMPLEMENTATION

OF STOCHASTIC FILTERING ALGORITHMS

Thesis Approved:

James R. Rowland
Thesis Adviser

Paul A. McCollum

Craig S. Sims

D. D. Durham
Dean of the Graduate College

867495

PREFACE

This study is concerned with the hybrid computer simulation of stochastic systems operating in a time-critical environment. The primary objective is to determine the effects of time-critical hybrid computer operations on systems with random parameters operating in an unmodeled noisy environment. The stochastic model chosen for this study is a linear system with the linear Kalman filter and a nonlinear system with related variational and extended Kalman filters.

The author wishes to express his appreciation to his major adviser and thesis committee chairman, Dr. James R. Rowland, for his guidance and assistance throughout this study. Appreciation is also expressed to other committee members; Professor Paul A. McCollum and Dr. Craig S. Sims, for their assistance during this study. The author gives a special note of appreciation to Mr. Jess B. Huff and Dr. Kelly V. Grider of the Guidance and Control Directorate, U.S. Army Missle Command, for their assistance in making this program of study and research possible. A note of thanks is given to Mrs. Shirley Rexroat for her excellent typing of the final copy.

Finally, special gratitude is expressed to my wife, Liz, and our daughters, Beth and Barbara, for their understanding, encouragement and many sacrificies.

TABLE OF CONTENTS

LIST OF FIGURES

Figure		Page

CHAPTER I

INTRODUCTION

For more than a decade the areas of computer simulation and optimal control theory have made explosive contributions to the existing body of theoretical and technological information related to the engineering sciences. In particular, the use of hybrid computers as a simulation tool and optimal filtering algorithms for stochastic estimation have given specific impetus for scientific development. The application of hybrid computers and filtering techniques to practical engineering problems have resulted in increased economy and improved performance. In addition, the investigation of large-scale engineering problems of a more complex nature has now become possible.

The motivation for recent improvements in hybrid computer simulation has been to increase the capabilities for handling sophisticated problems that could not be simulated with previously existing analog or digital computers. Typical problems included realtime simulation of missile systems with hardware-in-the-loop. The objective has been to combine the inherent speed and parallel computation of the analog computer with the improved accuracy and dynamic range of the digital computer. The application of hybrid computers to aerospace trajectory optimization problems, air defense missile analysis in a time-critical environment and Monte Carlo simulation studies have demonstrated the usefulness of such a tool. However, hybrid computation has introduced new dimensions of complexities to the analysis problem, such as time delay due to digital computation, signal sampling, data quantization and data reconstruction.

The continuing effort to obtain improved performances from engineering systems, particularly those operating in noisy environments, has introduced a new era in the development of optimization techniques. Using stochastic optimal estimation theory, improved performance can be obtained for those dynamical systems subjected to random input and measurement disturbances. This better performance typically requires accurate modeling of the physical system in terms of time-varying parameters, plant dynamics, and the statistical nature of the environment. Significant in the development of optimal filtering and smoothing theory has been the study, evaluation and implementation of suboptimal filters. Furthermore, economic considerations and realtime operation requirements have provided the motivation for developing fast and efficient methods for handling the enormous computational requirements of a fully implemented optimal filter.

The hybrid computer implementation of the optimal estimation algorithm establishes a basis for improved dynamical system performance resulting from increased computational efficiency. This improved computation permits faster update rates for the estimation algorithm, which results in reduced mean-square error for optimal and suboptimal filters in a realtime environment.

Background

The optimal selection of system parameters is one of the most important problems in systems engineering. The use of hybrid computers with Monte Carlo methods has been the most practical approach for investigating complex nonlinear dynamical systems with random parameters, random inputs and random initial conditions. Parameter selection in a multiparameter system using a hybrid computer has been studied by Bekey (1), and dynamical systems with random disturbances have been investigated by Korn (2,3) and Steinmetz (4).

The advantage of combined operation of the analog and digital computers has been realized only by overcoming a unique set of problems characterizing hybrid computers. Specifically, digitization problems (5,6) include compensation for the inherent time delay due to digital computation, sampled-data problems, analog-to-digital quantization and digital-to-analog data reconstruction. Furthermore, analog related problems (7,8) include the limited bandwidth of the analog computer components, delays due to analog computer mode switching and reset time and delays caused by digital-to-analog conversion. Difficulties encountered with hybrid computer simulation for time-critical environments have been identified by Fineberg and Serlin (9). An equally important, but less obvious, problem is the partitioning of a given system between the analog computer and the digital computer in a hybrid system (10).

Recent advances in analog and digital computer technologies have minimized many of the basic problems that characterize the hybrid computer. Bedient and Dike (11) described the configuration of a modern hybrid computer using a Control Data CDC 6400 digital computer, four Comcor Ci-5000 analog computers and associated linkage equipment. Further development by Soma, Crunkleton and Lord (12) resulted in analog-digital computer interface improvements and related software for improved hybrid computer simulation. Moreover, Graycon, Nolby and

Sanson (13) presented a detailed description of the actual operation of a high performance computing system for a time-critical application. Comprehensive error analyses of hybrid computer systems were performed by Karplus (14) and Mitchell (15). Miura and Iwata (16) studied the effects of delays due to digital computer execution in a hybrid computer. Furthermore, Mitchell (17) determined the effects of digital computer compensation of the computational delay. Vidal, Karplus and Kaludjlan (18) investigated the use of sensitivity coefficients for correcting quantization errors, while Gelman (19) proposed a method of corrected inputs for improved hybrid simulation. However, the dilemma of problem partitioning between the analog and digital computer remains with the hybrid computer user. At present there are no clear guidelines available to expedite this decision. However, a reasonable initial partitioning effort involves a general knowledge of hybrid computation, a detailed knowledge of the particular hybrid computer system to be used and a working knowledge of the dynamics of the given engineering problem.

Recent activities in optimal filtering theory, initiated by Kalman (20) and Kalman and Bucy (21) in the early sixties, have established a new basis for optimization techniques. An increasing body of literature on Kalman filtering provides a means for obtaining improved performances for engineering systems in a realtime environment with stochastic disturbances. Mendel and Gieseking (22) compiled a bibliography in excess of 900 references on the linear-quadratic-Gaussian problem of which optimal linear filtering is an integral part.

Since the earliest applications of Kalman filtering, the need for improved implementation including more efficient computation has become a primary consideration. Practical applications have been reported by

Gains (23) and by Schmidt, Weinberg and Lukesh (24) and theoretical presentations have been made by Meditch (25), Jazwinski (26), Sage and Melsa (27) and Bryson and Ho (28). Recently, Mendel (29) reviewed the practicality of a fully implemented Kalman filter from the computational viewpoint, and Simon and Stubberud (30) investigated the possibility of reducing the order of the filter equations to obtain a more efficient computation. Brown and Sage (31) demonstrated the effects of simplifying assumptions on the statistical information or modeling errors in plant dynamics, which typically results in a suboptimal filter. In addition, Fitzgerald (32), Schlee, Standish and Toda (33), and Price (34) investigated divergence of the Kalman filter for various error sources.

Applications of the Kalman filter to larger and increasingly more complex engineering systems operating in time-critical environments requires special considerations with regard to practical implementations. Bierman (35) and Friedland (36) discussed implementation problems for the discrete Kalman filter, and Nishimura (37) investigated error bounds for the continuous filter applications. Bucy, Merritt and Miller (38) demonstrated the enormous computational advantage in using hybrid computers for a particular nonlinear estimation problem. In particular, the results of this effort showed that one hybrid computer with 250 integrators and multipliers operates at speeds equivalent to forty-nine CDC 6600 digital computers operating in parallel. However, their results were obtained in an all-digital environment by synthesizing the hybrid computer on a digital computer. Hybrid computer implementation of Kalman filtering for systems with stochastic parameters operating in a time-critical environment has not been sufficiently investigated to determine the relative advantages over an all-digital implementation. It is significant to note that only three documents, Bucy et al. (38), Tacker (39,40), on the time-critical hybrid computation of Kalman filtering were located in a computerized search of the libraries of the U.S. Army Document Center and the National Aeronautics and Space Administration. These two libraries include over two million research documents, engineering reports and scientific abstracts. This extensive search indicates the minimal amount of investigation that has been directed toward using the hybrid computer as a tool for improved computation in Kalman filtering.

System and Filter Descriptions

Consider an nth-order linear, time-varying, dynamical system S described by

$$S: \quad \dot{\underline{X}}(t) = \underline{A}(t)\,\underline{X}(t) + \underline{B}(t)\,\underline{W}(\tau) \qquad (1.1)$$

$$\underline{y}(t) = H(t)\,\underline{X}(t) \qquad (1.2)$$

where $\underline{X}(t)$ is the n-dimensional state vector of S, W(t) is an r-vector input, A(t) is an n x n matrix, B(t) is the n x r gain matrix, $\underline{y}(t)$ is the m-vector output, and H(t) is the output matrix of S. It is assumed that the input $\underline{W}(t)$ to S is a vector-valued white-noise Gaussian process with zero-mean given as

$$E\{\underline{W}(t)\} = 0 \quad \text{for all } t \qquad (1.3)$$

and covariance matrix denoted as

$$\mathrm{cov}[\underline{W}(t);\, \underline{W}(\tau)] = E\{\underline{W}(t)\,\underline{W}(\tau)^T\} = Q_{\underline{W}}(t)\,\delta(t-\tau) \qquad (1.4)$$

where $\delta(\cdot)$ is the Dirac delta function and $Q_{\underline{W}}(t)$ is an $r \times r$ symmetric positive semidefinite matrix. Let t_o denote the initial time and $\underline{X}(t_o)$ the initial state vector of S. It is assumed that $\underline{X}(t_o)$ is a vector-valued Gaussian random variable, independent of $\underline{W}(t)$, with known mean

$$E\{\underline{X}(t_o)\} \triangleq \eta_{\underline{X}}(t_o) \tag{1.5}$$

and known covariance matrix

$$\begin{aligned} \mathrm{cov}[\underline{X}(t_o);\underline{X}(t_o)] = E\{[\underline{X}(t_o) - \eta_{\underline{X}}(t_o)] \\ [\underline{X}(t_o) - \eta_{\underline{X}}(t_o)]^T\} \triangleq P(t_o). \end{aligned} \tag{1.6}$$

Suppose the output $\underline{y}(t)$ can be observed only in the presence of white Gaussian noise. Therefore, let the observed signal be denoted as

$$\underline{z}(t) = H(t)\ \underline{X}(t) + \underline{V}(t) \tag{1.7}$$

where $\underline{v}(t)$ is a vector of Gaussian white-noise processes with zero means

$$E\{\underline{V}(t)\} = \underline{0} \quad \text{for all } t \tag{1.8}$$

and covariance matrix

$$\mathrm{cov}[\underline{V}(t);\underline{V}(\tau)] = E\{\underline{V}(t)\ \underline{V}^T(\tau)\} = Q_{\underline{V}}(t)\ \delta(t-\tau) \tag{1.9}$$

where $Q_{\underline{V}}(t)$ is symmetric positive definite. Furthermore, it is assumed that $\underline{V}(t)$, $\underline{W}(t)$ and $\underline{X}(t_o)$ are independent.

The Kalman Filter

In general the state of S is not available for measurement. Therefore, the problem is to obtain an estimate of the state $\underline{X}(t)$ in the sense of least mean-square error. This estimate $\hat{\underline{X}}(t)$ must be obtained by using the noise-corrupted signal $\underline{z}(t)$. The objective is to construct a filter F to accept the available data $\underline{z}(t)$ in realtime and produce a vector-valued signal $\hat{\underline{X}}(t)$ such that the error to be minimized in some sense is defined as

$$\underline{e}(t) = \underline{X}(t) - \hat{\underline{X}}(t). \qquad (1.10)$$

The resulting filter for such a system is directly related to the imposed constraints. Here the filter will be constrained to be linear and time-varying. It is required that the estimate of the state of S be unbiased and, moreover, that the estimate be a minimum-variance estimate.

The derivation of the Kalman filtering algorithm has been developed by various authors as previously indicated. A different derivation or variation of the algorithm usually reflects the particular intended application. For completeness, the general results for the optimum linear continuous filtering algorithm are included here. The optimal linear time-varying filter for the system S given in (1.1) and (1.2) is

$$\dot{\hat{\underline{X}}}(t) = A(t)\ \hat{\underline{X}}(t) + K(t)\ [\underline{z}(t) - H(t)\ \hat{\underline{X}}(t)] \qquad (1.11)$$

with an error covariance matrix differential equation

$$\dot{P}(t) = A(t)\ P(t) + P(t)\ A^T(t) - P(t)\ H^T(t)\ Q_{\underline{V}}^{-1}(t)\ H(t)\ P(T) + $$
$$+ B(t)\ Q_{\underline{W}}(t)\ B^T(t). \qquad (1.12)$$

The time-varying gain K(t) is given as

$$K(t) = P(t)\ H^T(t)\ \ Q_{\underline{V}}^{-1}(t) \qquad (1.13)$$

where P(t) is the n x n error covariance matrix. The remaining terms in

(1.11), (1.12) and (1.13) have been defined previously for the dynamical system. An important result of the linear optimal time-varying filter is that the time-varying gain K(t) can be precomputed. The non-realtime results can be used during the actual system operation and optimal estimation procedures. This also implies that the P(t) calculation will not be required during on-line operation, which is a very important consideration for time-critical computation or simulation operations. However, as will be shown later, for certain applications of the filtering algorithm for nonlinear systems, this efficiency of operation is not possible. A dynamic simulation of the linear time-varying system model and the linear optimal filter is shown in Figure 1.

Extensions to Nonlinear Systems

The stated results and operating conditions of the previous section were developed for linear systems. If the system involved is nonlinear, then the use of the Kalman filtering algorithm requires a different set of constraints and operating conditions. The application of the filtering algorithm to nonlinear systems has been treated by a number of authors, such as Sage and Melsa (27) and Jazwinski (26). The two particular extensions of the linear filtering algorithm of interest are the linearized, or variational filter, and the extended filter. In the first case the filtering algorithm is applied to linearized variations of the system about a nominal trajectory $\underline{x}_N(t)$ obtained by replacing all noise disturbances by their mean values. Jazwinski defined this configuration of the Kalman filter as the "variational filter". The extended filter involves the application of the filtering algorithm to variations about the estimate of the system state $\hat{\underline{X}}(t)$. For continuity the

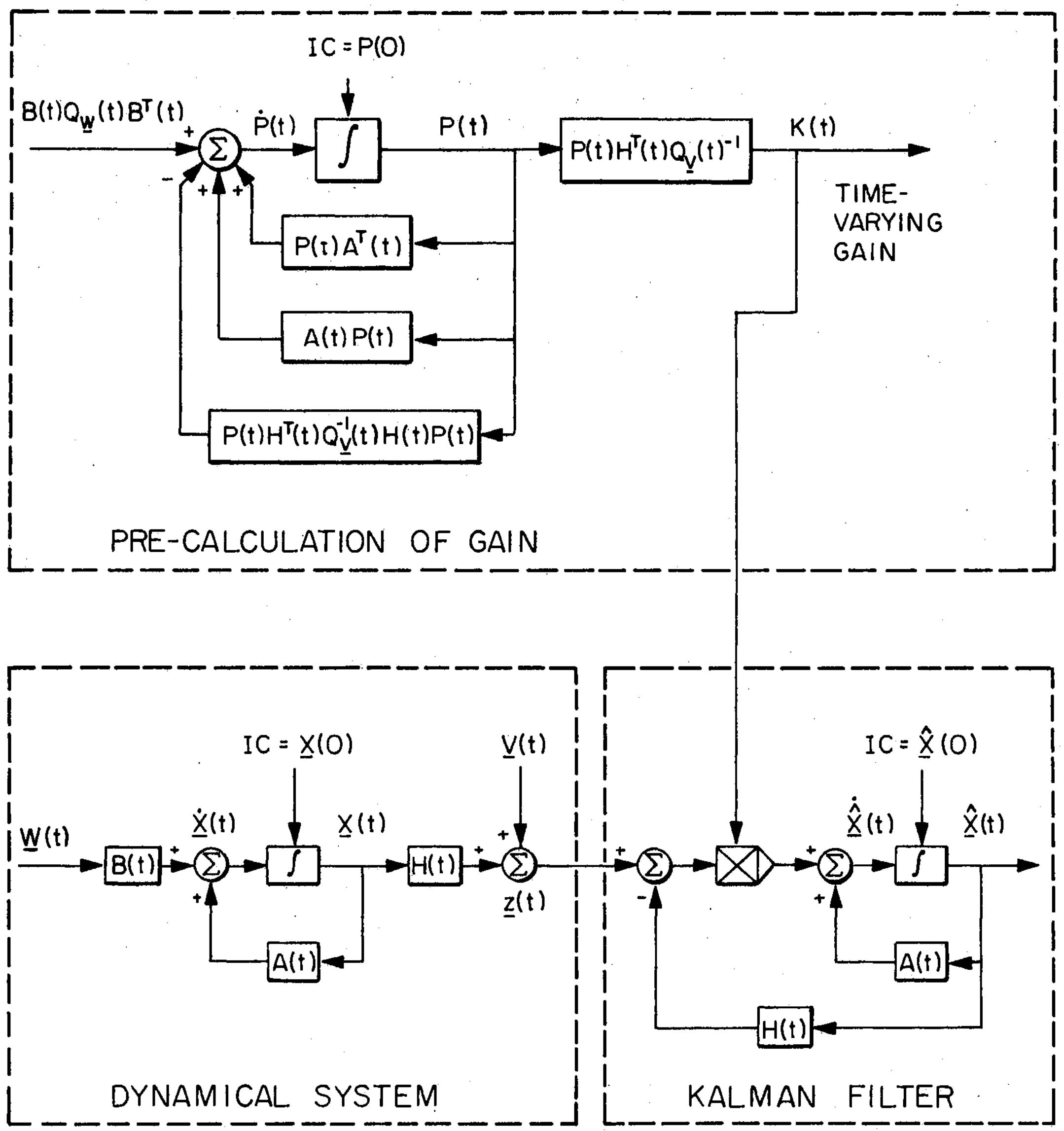


Figure 1. Block Diagram for the Linear Dynamical System and Related Kalman Filter.

results for these applications of the two particular filter configurations are included here.

The Variational Kalman Filter

Consider a nonlinear, continuous, time-varying system described by the nonlinear differential equation

$$\dot{\underline{X}}(t) = \underline{f}(\underline{X}(t),\underline{W}(t),t) \tag{1.14}$$

with a nonlinear observation of the system states given as the vector

$$\underline{z}(t) = \underline{h}(\underline{X}(t),t) + \underline{V}(t) \tag{1.15}$$

where t is time, $\underline{X}(t)$ is the vector of states, $\underline{W}(t)$ is a vector of white noise processes described by (1.3) and (1.4). A deterministic reference or nominal trajectory $\underline{X}_N(t)$ is obtained from

$$\dot{\underline{X}}_N(t) = \underline{f}(\underline{X}_N(t),\eta_{\underline{W}}(t),t) \tag{1.16}$$

where $\underline{X}_N(t)$ is a vector of the nominal states. Here the deviation or variation from the nominal trajectory is defined as

$$\underline{\delta X}(t) = \underline{X}(t) - \underline{X}_N(t) \tag{1.17}$$

and

$$\underline{\delta \dot{X}}(t) = \dot{\underline{X}}(t) - \dot{\underline{X}}_N(t). \tag{1.18}$$

Using expressions (1.16), (1.17) and (1.18), the nonlinear system functional $\underline{f}(\cdot)$ can be expanded in a Taylor series to obtain the differential equation that gives the variation between the actual state and the nominal state to first order.

$$\delta\dot{\underline{X}}(t) = \left.\frac{\partial \underline{f}[\underline{X}(t),\underline{W}(t),t]}{\partial \underline{X}(t)}\right|_{\substack{\underline{X}_N(t)\\ \eta_{\underline{W}}(t)}} \delta\underline{X}(t) + \left.\frac{\partial \underline{f}[\underline{X}(t),\underline{W}(t),t]}{\partial \underline{W}(t)}\right|_{\substack{\underline{X}_N(t)\\ \eta_{\underline{W}}(t)}} \delta\underline{W}(t) \tag{1.19}$$

The variation of the observation vector evaluated about the nominal is

$$Z(t) = \left.\frac{\partial \underline{h}[\underline{X}(t),t]}{\partial \underline{X}(t)}\right|_{\underline{X}_N(t)} \delta\underline{X}(t) + \underline{V}(t) \quad . \tag{1.20}$$

The results shown in equation (1.19) are in the same form as the expression for the linear system given in equations (1.1) and (1.2) where

$$A(t) \triangleq \left.\frac{\partial \underline{f}[\underline{X}(t),\underline{W}(t),t]}{\partial \underline{X}(t)}\right|_{\substack{\underline{X}_N(t)\\ \eta_{\underline{W}}(t)}} \tag{1.21}$$

and

$$B(t) \triangleq \left.\frac{\partial \underline{f}[\underline{X}(t),\underline{W}(t),t]}{\partial \underline{W}(t)}\right|_{\substack{\underline{X}_N(t)\\ \eta_{\underline{W}}(t)}} \quad . \tag{1.22}$$

By evaluating the new expressions for A(t) and B(t) at the nominal value of the sytem state, the linear filtering algorithm can be applied to nonlinear systems. An important characteristic is retained in this particular application. Since the nominal trajectory is deterministic and can be pre-computed, the time-varying gain K(t) can also be pre-computed. In Figure 1 the value of $\underline{X}_N(t)$ would be subtracted out ot the summing point for $\underline{z}(t)$ to yield $\delta\underline{Z}(t)$. This modification results in the filter output being $\delta\hat{\underline{X}}(t)$, which requires that the nominal $\underline{X}_N(t)$ be added to the filter output to give the estimated state $\hat{\underline{X}}(t)$.

The Extended Kalman Filter

The results for the extended filtering algorithm are derived by the same procedure as for the variational case. The one essential difference in the expressions in (1.19), (1.20), (1.21) and (1.22) is that the nominal $\underline{X}_N(t)$ is replaced by the estimate of the state $\hat{\underline{X}}(t)$. The required results are given as

$$A(t) \triangleq \left.\frac{\partial \underline{f}[\underline{X}(t),\underline{W}(t),t]}{\partial \underline{X}(t)}\right|_{\substack{\hat{\underline{X}}(t)\\ \eta_{\underline{W}}(t)}} \tag{1.23}$$

$$B(t) \triangleq \left.\frac{\partial \underline{f}[\underline{X}(t),\underline{W}(t),t]}{\partial \underline{W}(t)}\right|_{\substack{\hat{\underline{X}}(t)\\ \eta_{\underline{W}}(t)}} \tag{1.24}$$

$$\underline{\delta z}(t) \triangleq \left.\frac{\partial \underline{h}[\underline{X}(t),t]}{\partial \underline{X}(t)}\right|_{\hat{\underline{X}}(t)} \tag{1.25}$$

A very important difference exists between the variational and extended filtering algorithms with regard to time-critical applications. For the extended case the estimated trajectory or state about which the variations occur is no longer deterministic and cannot be precomputed, which prevents the pre-calculation of the time-varying gain K(t). Therefore, the error covariance matrix must be computed on-line. This requirement places severe constraints on the time-critical application of the extended filter. It should also be pointed out that the precalculated nominal $\underline{X}_N(t)$ in Figure 1 enters into the simulation as described above for the variational application.

Time-Critical Hybrid Computer Operation

The increased complexity of simulation requires that a precise definition of realtime and time-critical operations be established. The most general definition of realtime operation with digital computers applies to such operation as on-line banking systems, airline reservation systems and inventory control. The nature of these systems is such that a variable response time from the computer of a few seconds is not detrimental to the system operations. Therefore, any definition of realtime operations must be sufficient to include all realtime computer operations independent of the size of the simulator or the problem under study. A consistent working definition of time-critical hybrid computer operation is given by Fineberg and Serlin (9) and Graycon, Nolby and Sanson (13). In particular, an application is said to be time-critical if it demands a response from the digital computer within a fixed time after it has received a stimulus. This required response time is at least one or two orders of magnitude shorter than previously described for on-line realtime operations. Furthermore, for a time-critical operation, not a single omission of the stimulus-response cycle is permitted during the entire operation.

Any discussion of a time-critical hybrid computer operation with hardware-in-the-loop must be referenced to the concept of frame time (41). The frame time generally can be considered to be the total contiguous time required for the digital computer to complete the required digital operation for one iteration of the simulation. The maximum length of this time period is determined by the dynamics of the system, sampling rates, error budgets and related factors. Figure 2 shows some

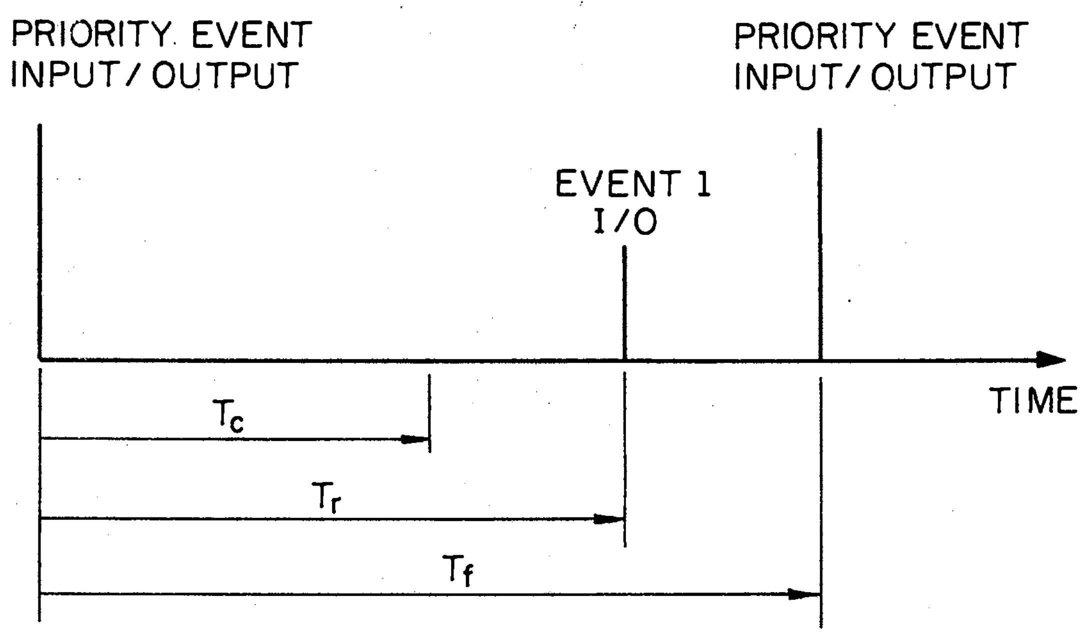


Figure 2. Frame Time Consideration in Hybrid Computation.

of the parameters for an ideal time-critical frame time. The priority or stimulus events are generated by real world events and may typically be a realtime clock with period T_f. The period T_c is the amount of time the digital computer takes to perform the actual calculation, and T_r is the time in which the real world demands a response from the digital computer and the output is required. In all cases, T_r must be equal to or less than T_f, and the ideal situation is that $T_c = T_r = T_f$. However, for any but the simplest problem this ideal situation is not achieved.

Due to the dynamics in a typical problem, multi-rate sampling is usually necessary and requires calculation and output with different priority levels. Multi-rate sampling and a variable interval within the frame time makes any compensation for computational delay difficult, if not impossible. Without compensation the error in the simulation result increases very rapidly. Shown in Figure 3 is an example of a realistic time-critical hybrid computer frame time. A dynamical subsystem may require an output response at T_{r1}, but only requires computational time S1. The calculation for a second subsystem can begin immediately after completing the calculation S1. This calculation continues until time for a stimulus response for subsystem S1, shown as Event 1. The calculation for S2 can be interrupted for the Event 1 input-output since its stimulus-response cycle is T_{r2} or greater. The distribution of calculation time over multiple sub-frames of the basic frame time is required to achieve an efficient computer operation. However, the allowable distribution is directly dependent upon the partitioning of the dynamical system between the analog and digital computers. The partitioned configuration contributes to the random noise in the hybrid computer operation indirectly through sampling rate errors, quantization, data

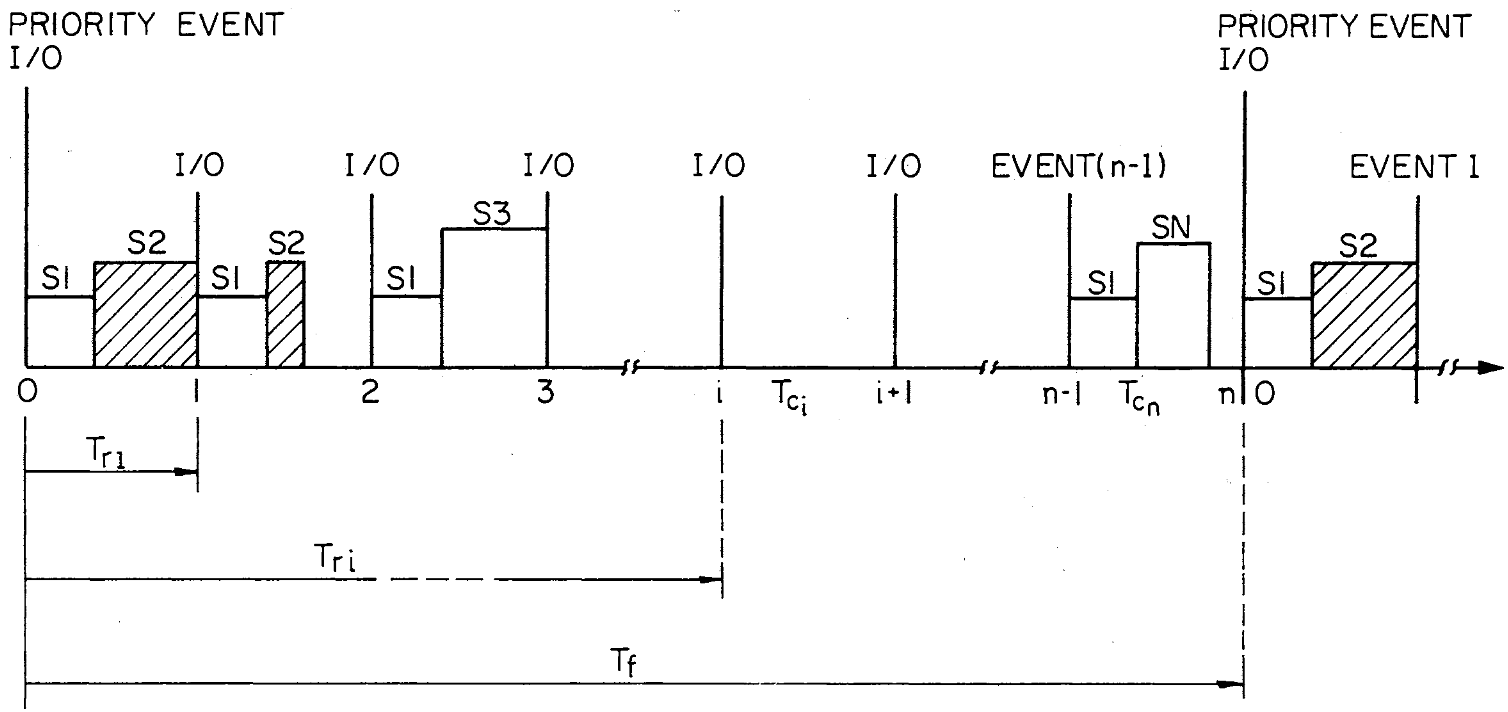


Figure 3. A Time-Critical Hybrid Computer Frame Time for Multi-Rate Systems.

reconstruction, truncation, roundoff, time skew and uncompensated time delays. This dependency emphasizes the importance of the partitioning process. The difficulties in obtaining time-critical hybrid computer results increases when the dynamical system includes stochastic processes.

A realtime hybrid computer operation is typically viewed as the synchronized exchange of data between the analog and digital computers through an interface. A time-critical operation with hardware included adds a new dimension of complexity to the total simulation. All interfaces, either for hardware or simulation, act as an error source and each input channel is a potential source of unknown random disturbances. Figure 4 shows a typical configuration of a time-critical hybrid computer with possible noise sources indicated. Two noise sources used for hybrid operations in this thesis research were a zero-mean Gaussian distribution to represent hybridization noise and a zero-mean uniformly distributed noise source to represent the sampling errors.

Research Objectives

The initial objective of this research is to determine the effects of the time-critical hybrid computer implementation of continuous stochastic filtering algorithms. Secondly, for any degradation that might occur in the stochastic filter performance due to this implementation, modifications will be made to obtain an improved operation. This research will be accomplished by developing an all-digital computer simulation of a time-critical hybrid computer operation. The Kalman filtering algorithm will be implemented with the time-critical simulation and the effects of hybridizing the noisy system and filtering algorithm

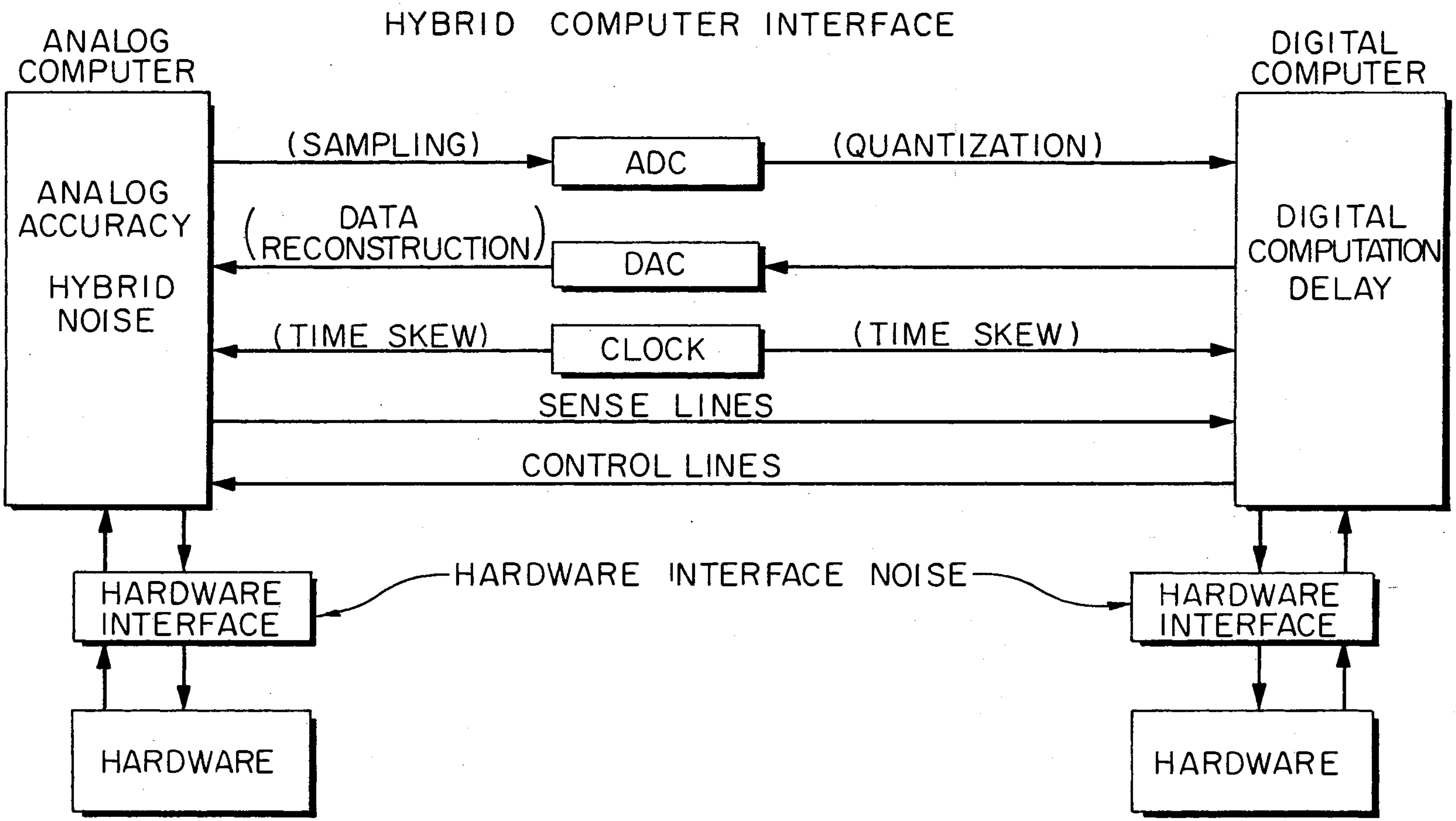


Figure 4. An Operating Time-Critical Hybrid Computer Showing Noise Sources.

will be established. Monte Carlo evaluations will be performed for comparison of results.

Thesis Outline

Following this introductory material on time-critical hybrid operations for stochastic systems, the implemented all-digital and time-critical simulation program is described at the beginning of Chapter II. Simulated hybrid computer results with the Kalman filter as a particular stochastic algorithm for linear systems are then given. In Chapter III a variation of the stochastic algorithm is used for a nonlinear system to determine the effects of time-critical operation in a more complex simulation model. Chapter IV included hybrid simulation refinements for frame time compensation, Kalman gain modification and problem partitioning. Conclusions and recommendations are presented in Chapter V.

CHAPTER II

TIME-CRITICAL SIMULATION OF LINEAR CONTINUOUS SYSTEMS

The success of simulating any time-critical operation is directly related both to the mathematical representation of the physical system and to the simulation of these models. Therefore, it is important to discuss the approach in developing the simulation used in this research. The time-critical system operation was simulated by developing an all-digital computer program. The simulation was designed to include linear and nonlinear systems with related filter configurations. Only that part of the total simulation that pertains to linear systems will be discussed in this chapter. The results of the simulation program presented is for a particular linear system and the related Kalman filter. The features of the simulation program for nonlinear systems and related filter configurations will be discussed in Chapter III. A complete Fortran listing of the total computer program is included in the Appendix.

Non-Time-Critical Simulation Operation

The total simulation program has seven distinct operational modes: time-critical (ITC=1), non-time-critical (ITC=0), Monte Carlo operation (MCR=Number of runs), single sample function (MCR=1), linear system and Kalman filter (KFNRL=1), nonlinear system with variational filter (KFNRL=2), and nonlinear system with extended filter (KFNRL=3).

Additional options may be specified for some of the major modes of operation, e.g. for time-critical operation the hybrid sampling update rate and frame time may be specified with TSAMP. The integer value associated with TSAMP designates the number of non-time-critical integration time intervals to be included in the hybrid update frame time. The operational modes of primary interest here are time-critical, non-time-critical and Monte Carlo operation for a linear system and related Kalman filter.

Time-critical operation implies the synchronous operation of an analog computer and a digital computer with the associated frame time. For the purpose of this research the non-time-critical mode implies that the simulation process is in parallel, analogous to analog computer operations in which the concept of frame time does not apply. In the hybrid or time-critical mode, the analog simulation may be represented by a non-time-critical operation. The totally non-time-critical simulation represents the continuous system without any of the associated effects of hybridization. The results from this mode of simulation may be used as basis of comparison for the time-critical hybrid computer operation. Since the non-time-critical mode of operation corresponds to a very accurate analog computer simulation, the digital computer integration step size must be chosen to minimize the effects of digital computer integration. Acceptable hybrid computer results may typically be obtained if the particular system variable partitioned on the digital computer is sampled at least ten times per cycle for the highest frequency of interest. This lower bound often requires some compensation for the sampling error reflected as digital computer execution time. For a realistic simulation of the analog operation, an integration interval corresponding to forty samples per cycle was selected for use with a fourth-order Runge-Kutta (RK4) integration algorithm.

Shown in Figure 5 is a flow diagram of the dominant operational features of the non-time-critical simulation. The majority of the operations indicated are common to other modes of the simulation. The pre-calculation of a nominal trajectory $\underline{X}_N(t)$ systematizes the filtering procedures for linear systems with non-zero-mean disturbances and for nonlinear systems. The value of $\underline{X}_N(t)$ is subtracted and added at the appropriate points for the Kalman filter operation. Generating $\underline{X}_N(t)$ is an off-line process, and no significant penalty is incurred for time-critical operations.

The time-varying Kalman gain GK(t) is calculated with the same precision as the nominal trajectory values. The gain calculation is shown separately from the nominal trajectory calculation since one particular nonlinear problem configuration to be considered in Chapter III does not permit the pre-calculation of GK(t). It should be noted that the pre-calculation of the gain is by-passed in that particular nonlinear mode.

The random disturbances for the non-time-critical mode consist of two independent, zero-mean Gaussian white noise processes for W(t) and V(t). The values are simulated with a pseudo-random number generator by using a multiplicative congruential method with a recurrence formula of the form

$$Z_{k+1} = A\ Z_k\ (\text{MODULO}\ \ M). \tag{2.1}$$

The scalar constants A and M are selected to insure good statistical properties. The random sequence obtained from (2.1) is approximately

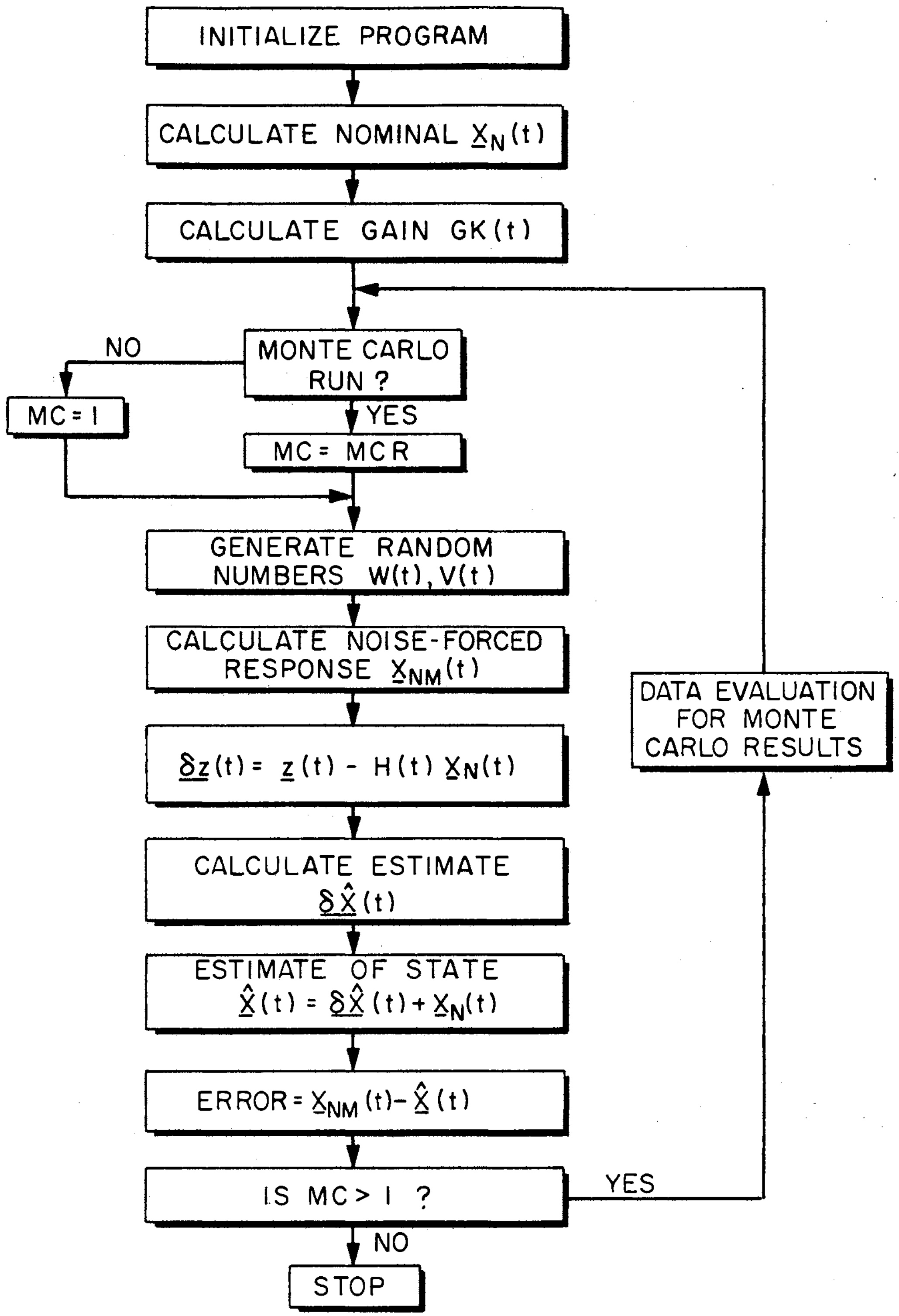


Figure 5. Flow Chart for Non-Time-Critical Simulation.

uniformly distributed on the unit interval (0,1). Box and Muller (42) developed an exact approach for transforming two independent random variables which are uniformly distributed on the unit interval to a pair of independent random variables with zero-mean, unity variance Gaussian distributions. The closed-form relation developed by Box and Muller is

$$G_1 = (-2 \ \mathrm{LOG}_e \ U_1)^{\frac{1}{2}} \mathrm{COS} \ 2\pi \ U_2$$
$$G_2 = (-2 \ \mathrm{LOG}_e \ U_1)^{\frac{1}{2}} \mathrm{SIN} \ 2\pi \ U_2 \tag{2.2}$$

where U_1 and U_2 are uniformly distributed, and G_1 and G_2 are Gaussianly distributed random variables. The resulting number sequence satisfies the requirements necessary for the digital representation of Gaussian white noise. This particular generator has been evaluated and found to be slightly better than the central limit method of averaging twelve uniformly distributed numbers (43). Using this pseudo-random Gaussian number generator, any particular variance may be obtained by multiplying the generated number sequence by the desired variance. The use of random numbers as disturbance input sample functions for dynamical systems requires special considerations for variance representation. It is necessary to determine a discrete representation for continuous noise processes. Rowland and Gupta (44) have shown that the relationship between variances of discrete and continuous noise processes is

$$Q_{Wd} = \frac{Q_W}{DT} \tag{2.3}$$

where Q_{Wd} and Q_W are the variance of the discrete and continuous cases, respectively, and DT is the sampling interval.

Using the generated random numbers as the disturbance inputs with specified means and variances, the measurement value $\underline{z}(t)$ is obtained by summing the output of the noise-forced response $H(t)\underline{X}_{NM}(t)$ with the measurement noise $\underline{V}(t)$. The mean value of $\underline{W}(t)$ determines the pre-calculated nominal $\underline{X}_N(t)$. The linear system response for an equivalent zero-mean Gaussian noise is obtained by

$$\underline{\delta z}(t) = H(t)(\underline{X}_{NM}(t) - \underline{X}_N(t)). \quad (2.4)$$

The value of $\underline{\delta z}(t)$ is the forcing function to the Kalman filter for estimating the state $\underline{\delta X}(t)$. The total estimate for non-zero-mean Gaussian noise is obtained by summing the nominal solution $\underline{X}_N(t)$ and the quantity $\underline{\delta X}(t)$ such that

$$\hat{\underline{X}}(t) = \underline{X}_N(t) + \underline{\delta X}(t). \quad (2.5)$$

The error in the estimate of the state is defined as

$$EPS = \underline{X}_{NM}(t) - \hat{\underline{X}}(t). \quad (2.6)$$

Verification of means and variances for linear and nonlinear systems is achieved by using a selected number of Monte Carlo runs. The sample means and sample variances of the typical system variables of interest in this simulation are defined by

$$RX1AVG = \frac{\sum_{MC=1}^{MCR} (X_{NM1}(t))}{MCR}$$

$$VARX1 = \frac{\sum_{MC=1}^{MCR} (X_{NM1}(t) - RX1AVG)^2}{(MCR - 1)} \quad (2.7)$$

For consistency the program variable names are shown. The sample variance VARX1 may be computed by using running sums according to

$$RX2AVG = \frac{\sum_{MC=1}^{MCR} (X_{NM1}(t))^2}{(MCR - 1)}$$

$$VARX1 = RX2AVG - (RX1AVG)^2 \left(\frac{MCR}{MCR-1}\right) \quad (2.8)$$

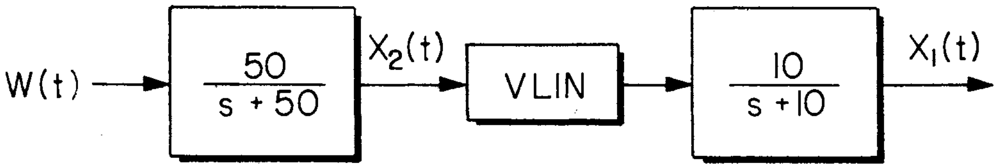
Similar expressions for the sample means and sample variances of all system variables and filter estimates are utilized in the program of the Appendix.

Non-Time-Critical Linear System Simulation

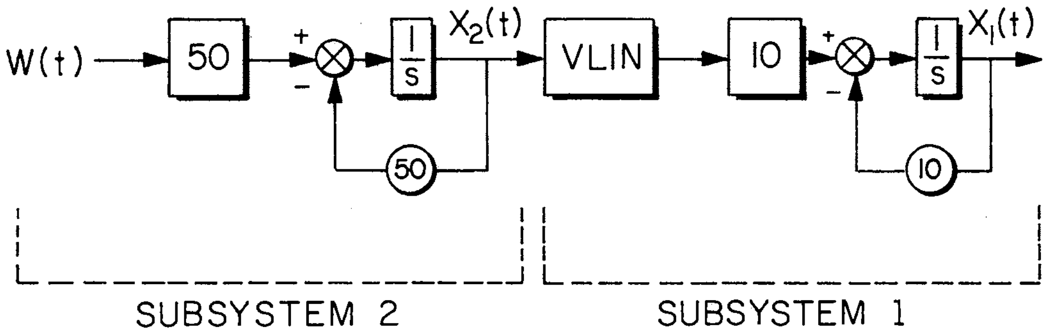
The block diagram of the second-order linear system used in the simulation program is shown in Figure 6. This system may be expressed as a set of first-order linear differential equations.

$$\begin{bmatrix} \dot{X}_1(t) \\ \dot{X}_2(t) \end{bmatrix} = \begin{bmatrix} -10.0 & 10.0VL1N \\ 0 & -50.0 \end{bmatrix} \begin{bmatrix} X_1(t) \\ X_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 50.0 \end{bmatrix} \underline{W}(t). \quad (2.9)$$

The particular configuration for the second-order system was chosen for two reasons. The system can represent two separate physical systems operationally interfaced with a gain VL1N, and error sources can be introduced to Subsystem 1 without affecting Subystem 2. Secondly, for purposes of partitioning, the system frequencies should be uncoupled and somewhat separated. These system frequencies essentially determine an acceptable hybrid computer update rate and frame time. Subsystem 2 is chosen with ω_2 = 50 radians per second with a resulting frequency of



(a) BLOCK DIAGRAM OF SECOND-ORDER SYSTEM



(b) SIMULATION MODEL OF SECOND-ORDER SYSTEM

Figure 6. Second-Order Linear System Chosen for Non-Time-Critical and Time-Critical Simulations.

$f_2 = 7.96$ cycles/sec. The non-time-critical operation of 40 integration intervals per cycle or 320 samples per second requires an integration step size of 0.003125 seconds. Similarly, Subsystem 1 with $\omega_1 = 10$ radians per second requires an integration step of not greater than 0.015625 seconds.

The effect of time-critical operation on the selected system and related Kalman filter was established by comparing time-critical performance with non-time-critical results. For further evaluation of operating in an increased noisy environment, results were obtained for two operating conditions of the stochastic system. The first condition utilized an input Gaussian noise with a mean of 10.0 and a variance of unity. A second operating condition was chosen with increased input disturbances of the same mean and a variance of 5. The measurement noise in all cases was Gaussianly disturbed with a mean of zero and a variance of 0.5. Sample functions for the two conditions are shown in Figures 7 and 8. The noise-forced system response is denoted by $x_{NM1}(t)$ and the estimate of the system state by $\hat{x}_1(t)$. Further comparisons are shown in Figures 9 and 10 by ensemble-averaging 100 Monte Carlo runs.

Time-Critical Simulation Operation

Time-critical operation implies that the simulation model has been partitioned between the analog and the digital computer. Correct simulation of a time-critical operation requires that the non-time-critical program as described in the previous section must be expanded. The changes must include provision for multi-rate integration and the effect of sampling rate errors for continuous data transmitted to the digital

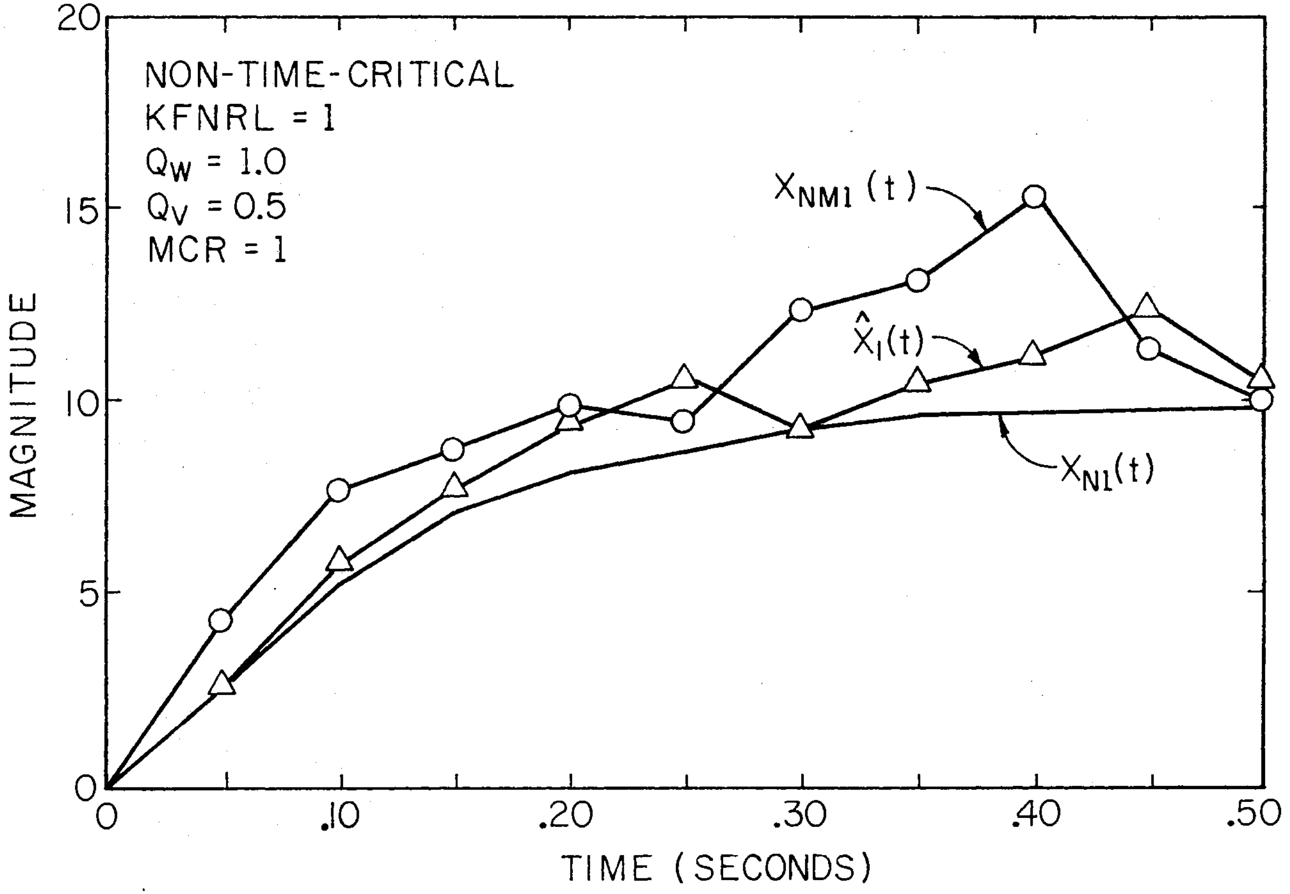


Figure 7. Sample Functions of System Variables for Small Input Disturbances.

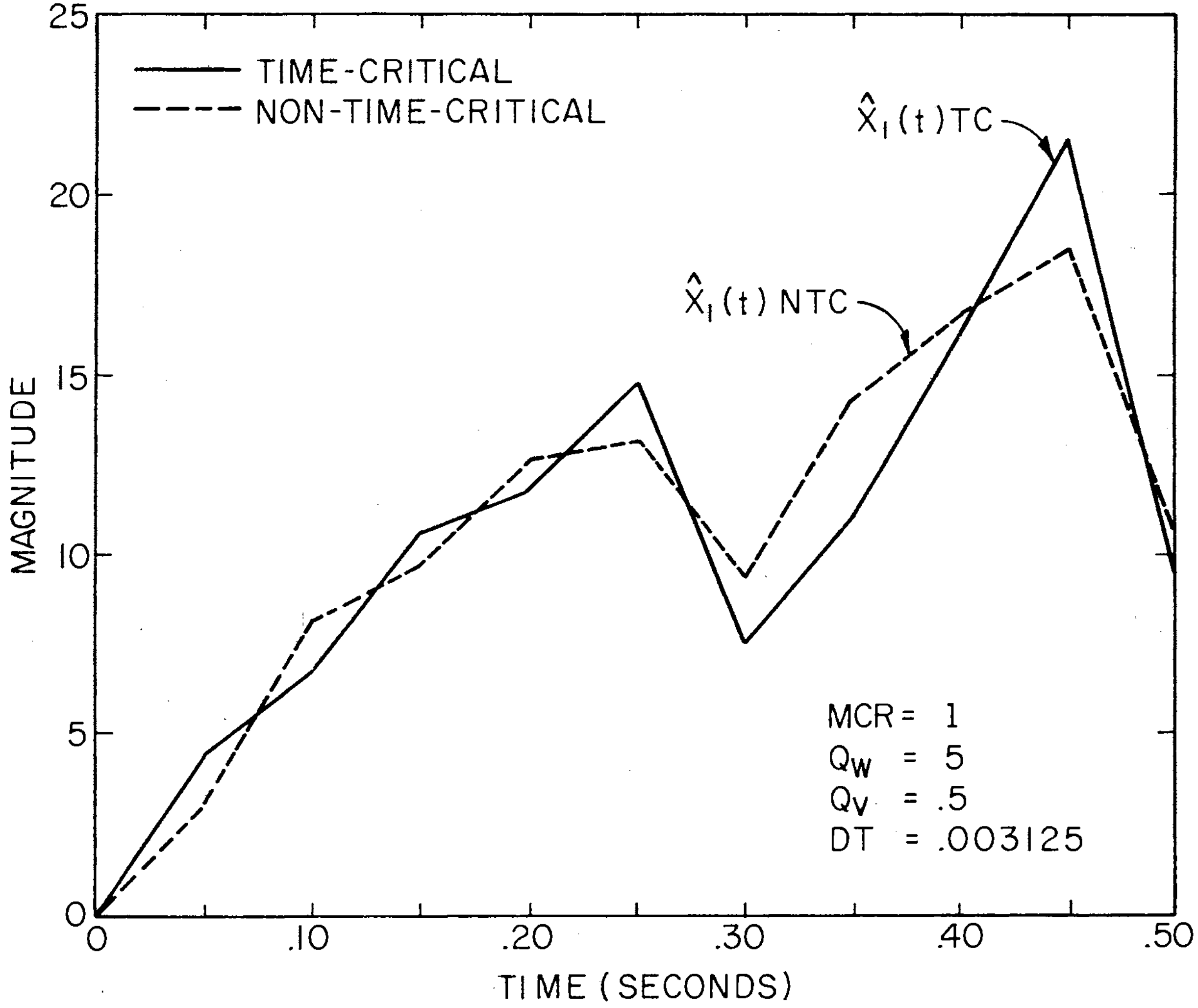


Figure 8. Sample Function for Large Input Disturbances.

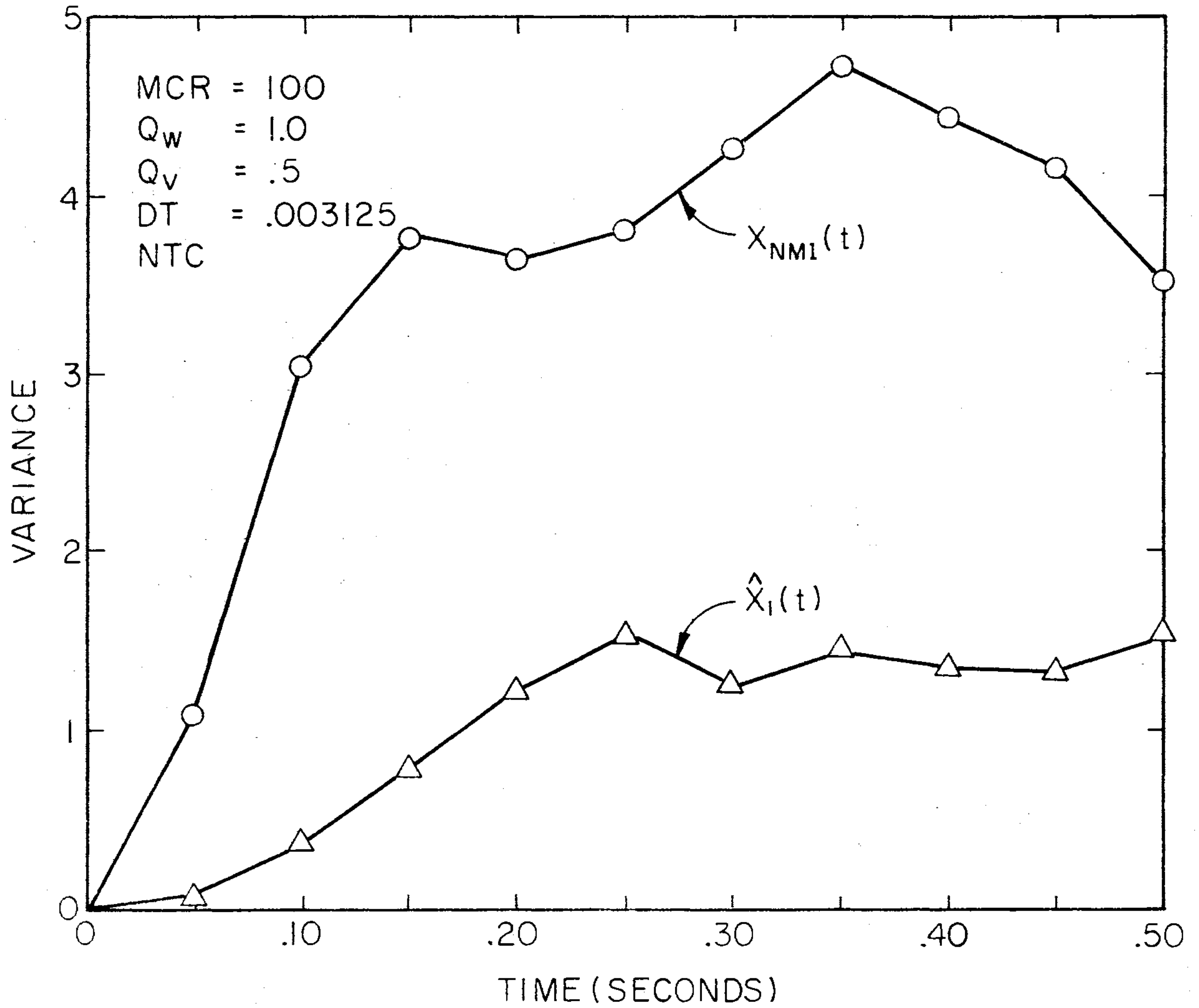


Figure 9. Non-Time-Critical Variances With Small Input Disturbances.

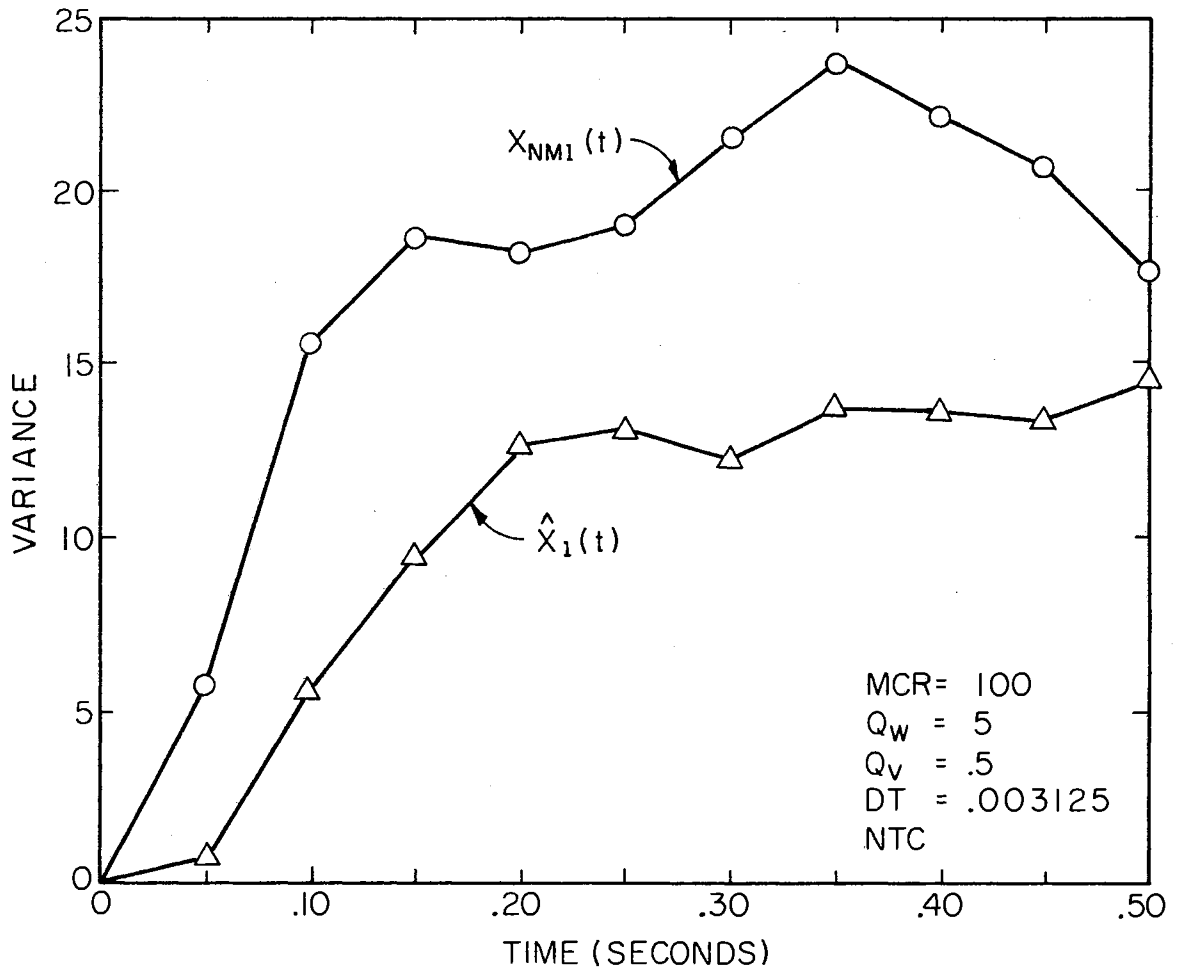


Figure 10. Non-Time-Critical Variances With Large Input Disturbances.

computer. Figure 11 shows a block diagram of the dominant operating features of the time-critical mode of the simulation program. The operations discussed in the previous section have been indicated collectively. Without any specific guideline, the initial partitioning choice may be conditioned only on elementary considerations of the system dynamics, i.e. the higher frequency system models are placed on the analog computer and the slower dynamics on the digital computer. Since one of the objectives of this simulation is to determine the effects of errors introduced by hybridization, the initial partitioning need not be completely arbitrary. For purposes of comparing various partitioning configurations, an initial choice was made to use a large digital computer frame time and a minimum contribution to hybridization errors. This is achieved by placing Subsystem 1 on the digital computer and performing all required filter calculations on the digital side. The time-varying filter gain GK(t) is pre-computed and does not contribute to the frame time. This partitioned configuration requires only that the input to Subsystem 1 be corrupted by quantization and hybrid system noise. A simulation diagram of the partitioned time-critical configuration is shown in Figure 12. The effects of partitioning without hybrid system and quantization noise were initially determined. Minimum sampling of ten samples per cycle of the highest frequency of interest required an ideal frame time of 12.5 milliseconds. This was due directly to the total system dynamics being included in the Kalman filter. Direct comparisons of the effects of time-critical operation were made with Monte Carlo results.

Simulations of time-critical deterministic systems typically use the mean of the variables of interest to determine the correctness of

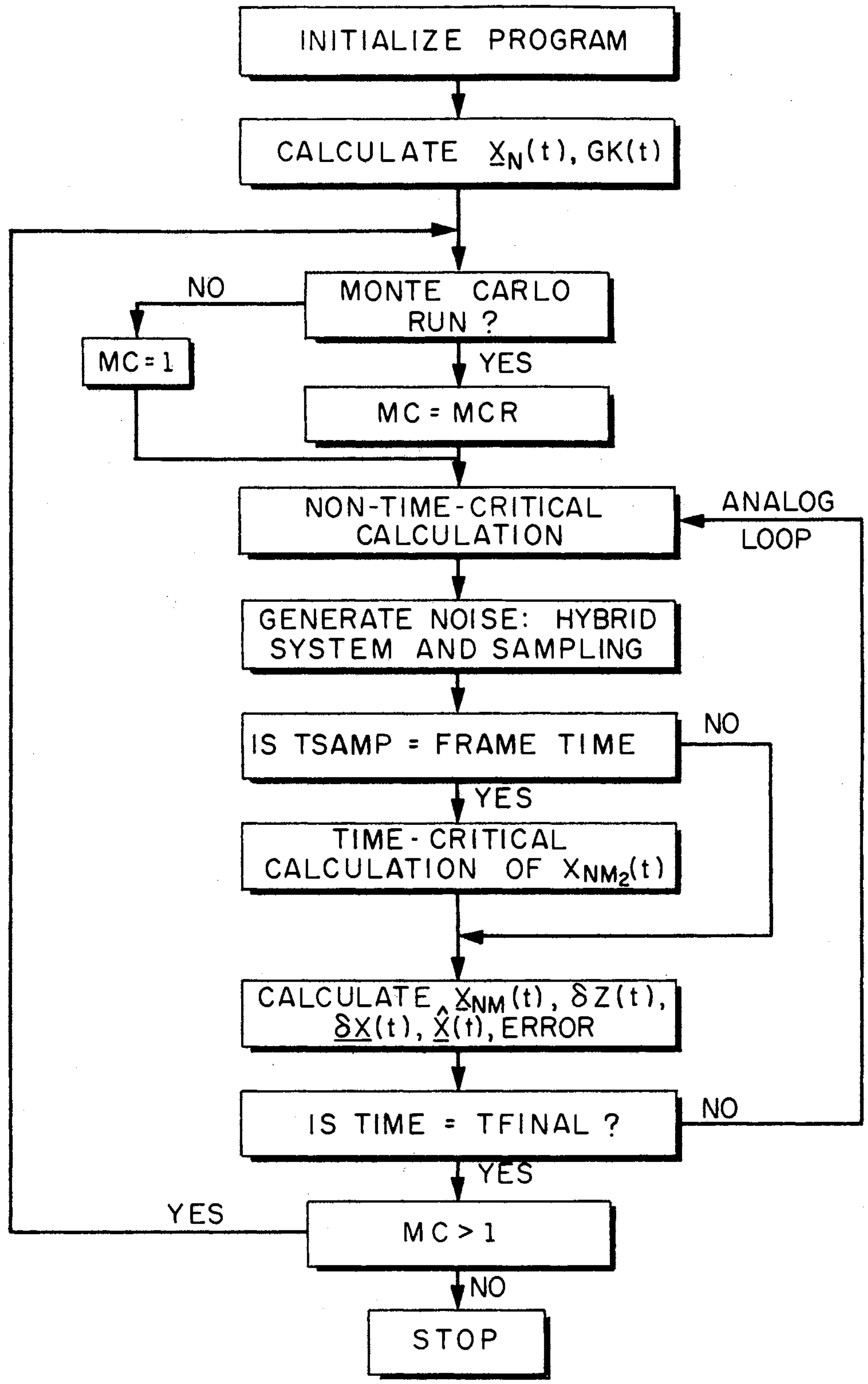


Figure 11. Flow Chart for Time-Critical Simulation.

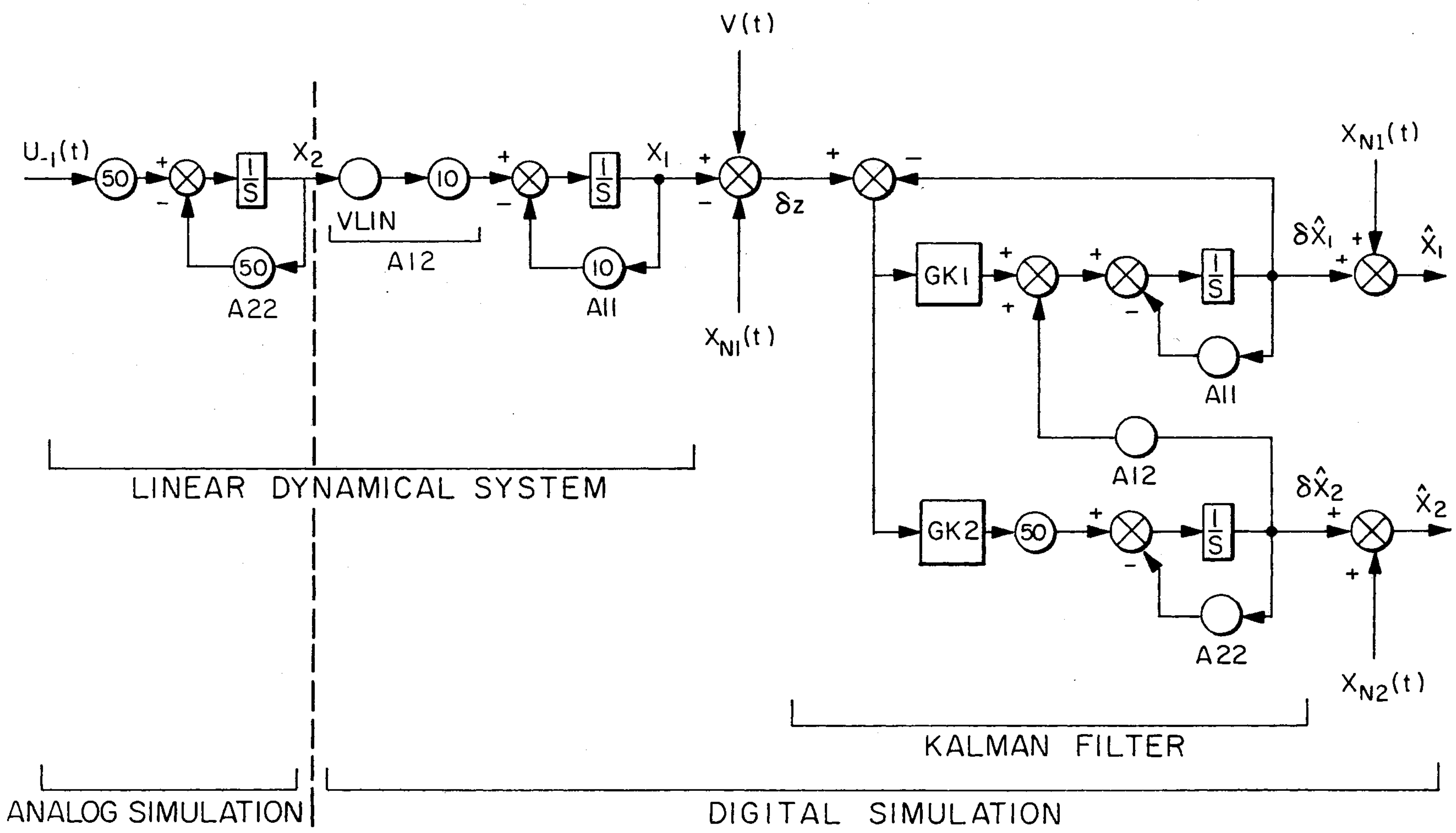


Figure 12. Partitioning for Time-Critical Simulation.

the simulation and to evaluate particular error sources. The effects of partitioning on the Monte Carlo mean for this system is minimal as shown in Figure 13. However, the system with stochastic processes must include other considerations, such as the variation of the variables of interest. Recall that equation (2.3) gave the relationship between the discretized and continuous variances as

$$Q_{Wd} = \frac{Q_W}{DT} . \qquad (2.3)$$

The step size DT used for the continuous non-time-critical operation was 0.003125 seconds. The importance of using the correct time step is shown with results in Figure 14. Curve A is the non-time-critical results where all operations used the same size step. Straightforward partitioning of the problem and using the noise generated for the non-time-critical operations resulted in Curve B. Correct variance modification using the hybrid frame time in generating noise for use on the digital computer resulted in Curve C. The difference in Curve B and Curve C emphasizes the importance of the hybrid computer frame time in generating random functions for hybrid simulation. The correct implementation of the variance modification becomes more complex for multirate updating. This error also emphasizes the desirability of fixed-interval sampling as opposed to variable intervals obtained with adaptive sampling techniques. The difference in Curve A and Curve C in Figure 14 is attributed to sampling rate errors.

Included in the objectives of this research is the determination of the effects of hybrid computer implementation of stochastic systems operating in a time-critical environment. Inherent in such operating

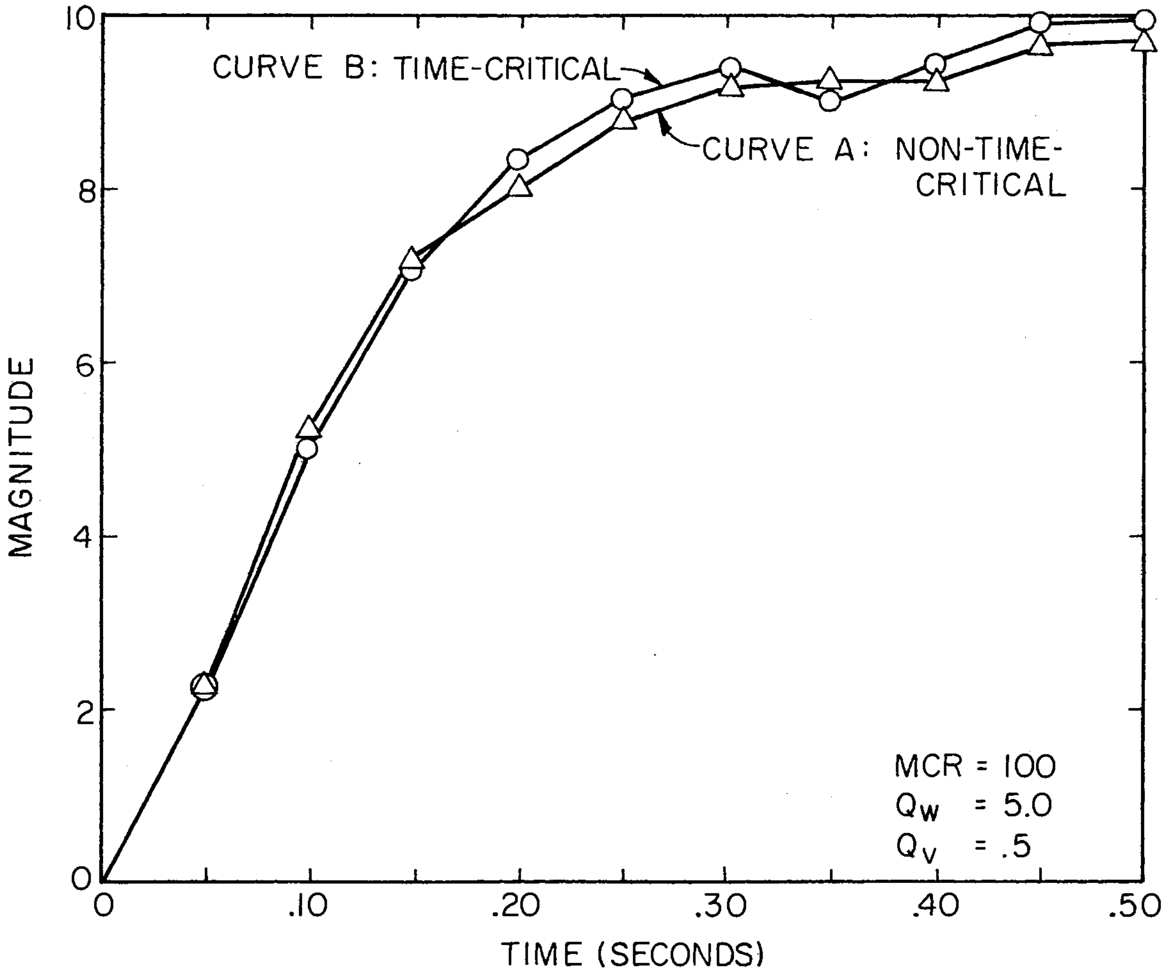


Figure 13. Effects of Hybrid Partitioning on the State Estimate.

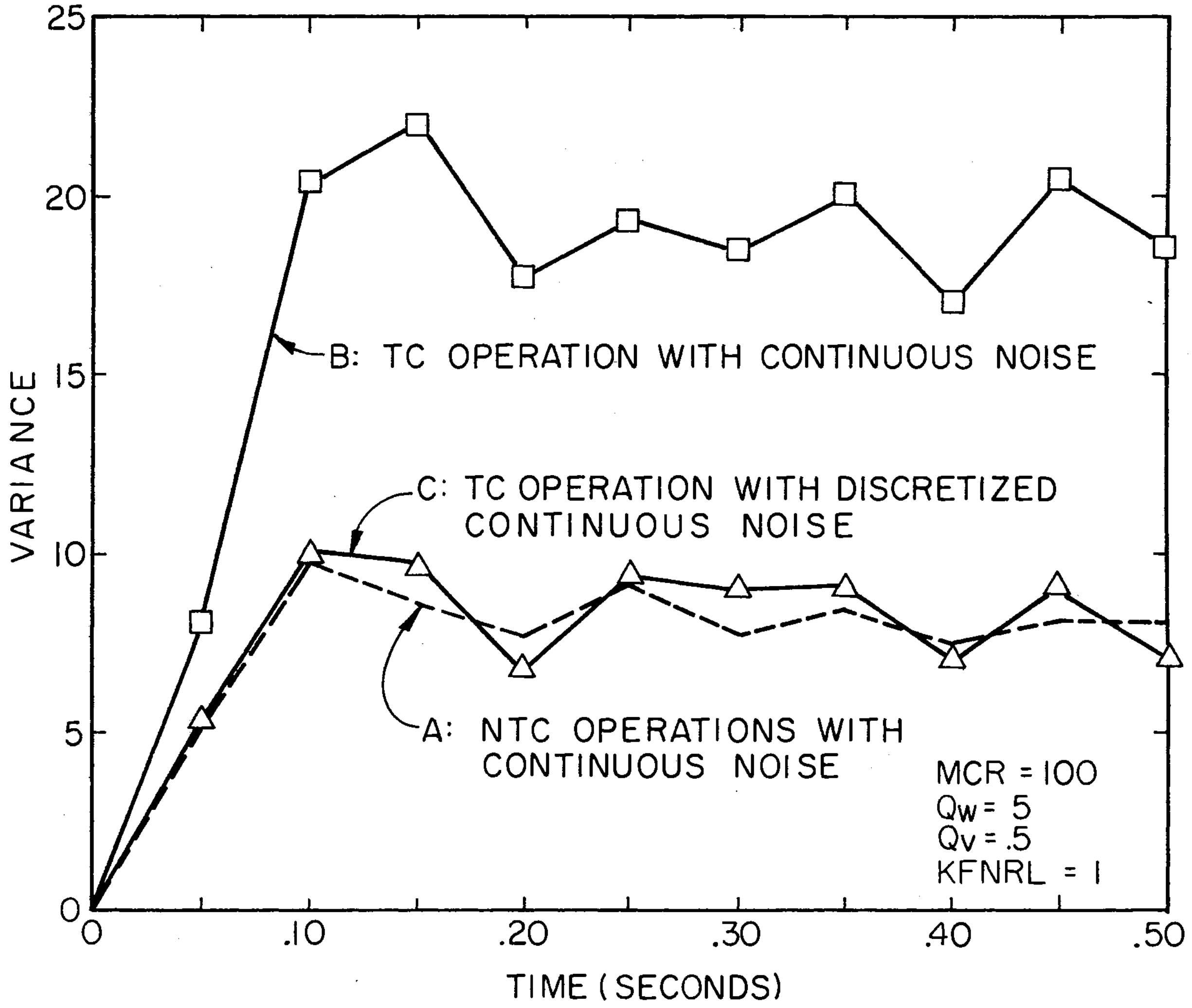


Figure 14. Variance of Error for Non-Time-Critical and Time-Critical Operations.

conditions are random disturbances not specifically included in the stochastic system model. Here hybrid noise is used to represent the ground or electrical noise typical in an actual hybrid computer system. For purposes of this research, a zero-mean Gaussian white noise process was used to simulate the hybrid system noise. Particular variances are specified depending on the simulating operation condition. A uniformly distributed white noise process with amplitude proportional to one-half the hybrid update rate was used to represent sampling noise. The effects of hybrid noise on the partitioned configuration is shown in Figure 15. Curve A is the partitioned time-critical result of including hybrid noise with a variance $Q_H = 1.0$. Curve C shows the effects of including both hybrid and sampling noise.

The ideal uncompensated frame time of 12.5 milliseconds was used to determine the effects of hybrid computer partitioning. The actual digital computer execution time was measured as approximately 16 milliseconds. This time would have to be reduced to achieve realtime operation. Further reductions would be necessary if any improvement with increased hybrid update rates are to be accomplished. The options available for possible improved operations are (1) use a faster digital computer, (2) utilize more efficient programming in terms of machine or assembly language programs, and (3) repartition the total system model between the analog and digital computer with careful consideration given to noise generation process in the partitioned model and noise sources not included in the model. Due to fixed resources, the first option is usually not available. Combinations of options (2) and (3) are typically used where tradeoffs must be accomplished to achieve a realistic time-critical simulation.

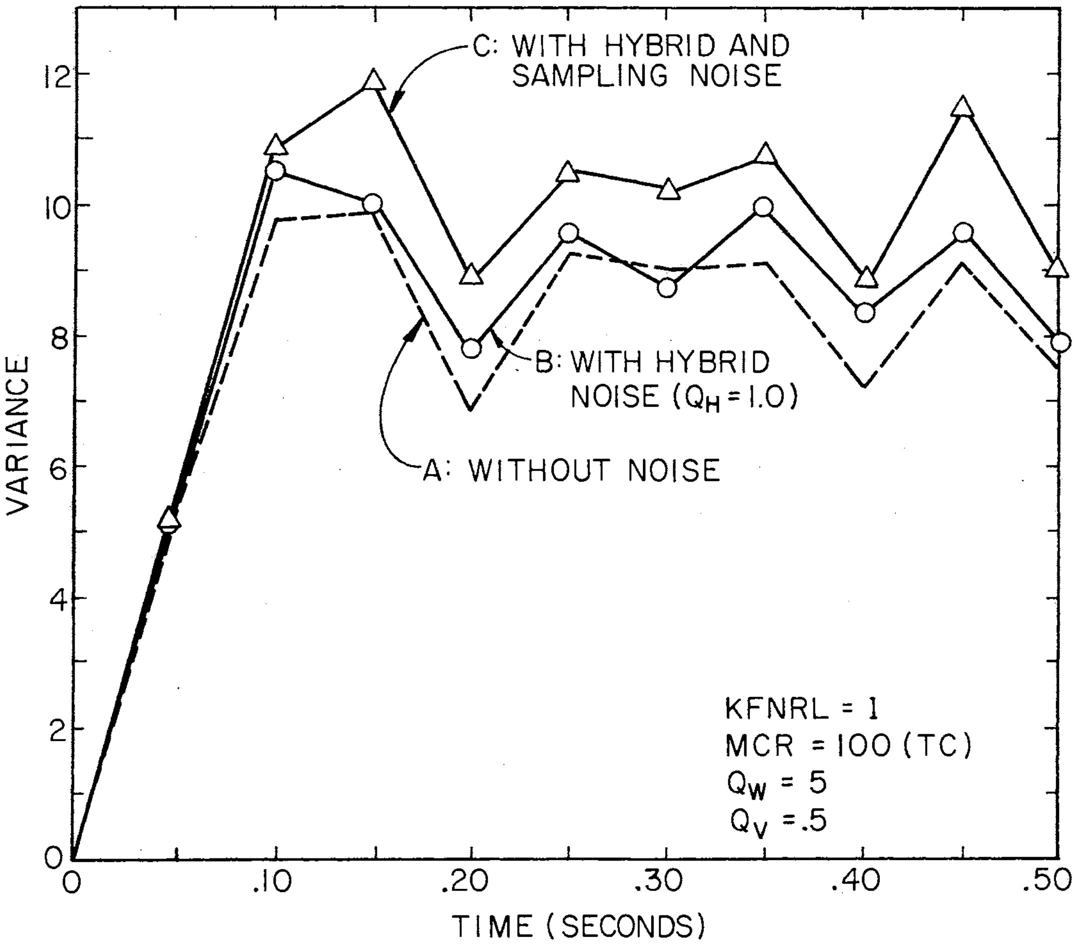


Figure 15. Effects of Hybrid and Sampling Noise on Error Variance.

Summary

A straightforward time-critical simulation of a linear stochastic system was developed in this chapter. Sample functions and Monte Carlo results showed that the particular partitioning process resulted in a large increase in the variance of the system variables. The increased variation due directly to partitioning made the error contribution of hybrid and sampling noise less obvious. Time-critical simulation of a stochastic system requires that special considerations be given to the hybrid update rates, system partitioning and random noise generated for the systems on the digital and analog computers. These and other options form a basis for hybrid simulation refinements to be discussed in a later chapter.

CHAPTER III

TIME-CRITICAL SIMULATION OF NONLINEAR CONTINUOUS SYSTEMS

This chapter deals with the time-critical simulation of nonlinear stochastic systems with the variational and the extended Kalman filter algorithms as given in Chapter I. In particular, the simulation program and results for linear systems developed in Chapter II are extended to a nonlinear system modeled by nonlinear differential equations involving a cubic nonlinearity. The effects of time-critical operations are determined for a particular nonlinear system using the variational and extended Kalman filters. The time-critical operations are compared with non-time-critical results using 100 Monte Carlo runs.

Non-Time-Critical Simulation Operation

Expanding the previously described simulation program for nonlinear systems operations for use with the variational and the extended filter requires three prime considerations. First, capabilities are needed to evaluate partial derivatives of the nonlinear system along a nominal trajectory as indicated by equations (1.19) through (1.22). Secondly, it is necessary to evaluate the system partial derivatives along an estimate of the system states according to equations (1.23) through (1.25). The third consideration is for on-line calculations of the Kalman gains GK(t) for extended filter operations. The required on-line calculation must be included for both the time-critical and non-time-critical modes. The effect of time skew as an additional error source can be minimized by giving careful attention to implementing the covariance and gain calculation in the extended filter time-critical operation. The Monte Carlo mode of operations and other options remain essentially the same as previously described.

Nonlinear Systems Operation

The nonlinear system with a cubic nonlinearity selected for the research is shown in Figure 16 and is described by a set of nonlinear differential equations as

$$\begin{aligned} \mathring{X}_1(t) &= 10(-X_1(t) + \text{VLIN}\ X_2(t) + \text{ALPHA}\ X_2^{\,3}(t)) \\ \mathring{X}_2(t) &= 50(-X_2(t) + W(t)). \end{aligned} \qquad (3.1)$$

The required partial derivatives for this system from equations (1.21) and (1.23) are

$$A(t) = \begin{bmatrix} -10.0 & 10(\text{VLIN} + 3.0\ \text{ALPHA}\ X_2^{\,2}(t) \\ 0 & -50.0 \end{bmatrix} \qquad (3.2)$$

where A(t) is evaluated along the nominal $\underline{X}_N(t)$ for the variational filter and along the estimate of the state for the extended mode. The system was modeled to achieve a weighted combination of the linear and nonlinear effects by selecting values for VLIN and ALPHA. This option allowed the determination of the effect of time-critical operations for various degrees of nonlinearities.

The chosen nonlinearity made the system very responsive to small signals or noise. A harsh nonlinear operating condition was achieved

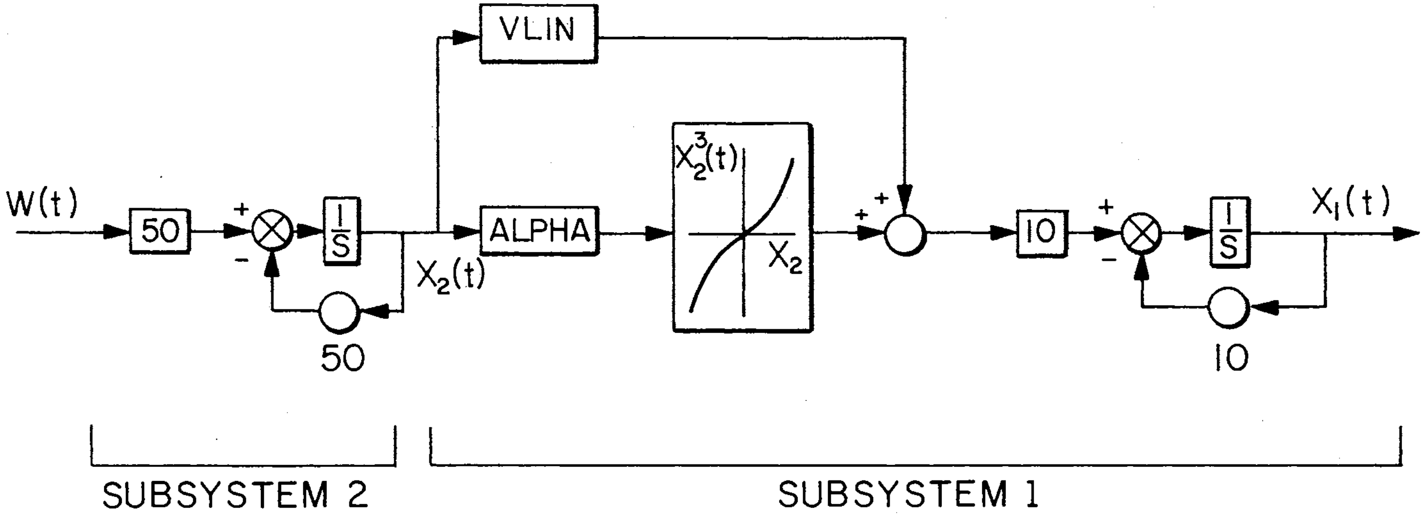


Figure 16. Nonlinear System Chosen for Non-Time-Critical and Time-Critical Simulation.

with ALPHA = 0.1, VLIN = 0, Q_W = 1.0 and Q_V = 0.5. Monte Carlo results for the variational filter in the non-time-critical mode are shown in Figure 17. The magnitude of the resulting error variance indicates that this particular nonlinear mode of operation may not be suitable for initially determining the effects of hybridization. A mild nonlinear operation condition was achieved by choosing ALPHA = 0.01 and VLIN = 0.2 with the same input noise conditions. Sample functions for this non-time-critical operation with the variational and the extended filter are shown in Figure 18. The extended filter showed an improved performance even for this mild nonlinear operating condition. Monte Carlo results for the selected operating conditions are shown in Figure 19 with the extended filter showing improved error conditions. These results were used for comparing the effects of hybridization for time-critical operations.

Time-Critical Simulation Operation

The initial choice of partitioning was the same as in the previous chapter with Subsystem 1 for non-time-critical (analog) operations and Subsystem 2 on the digital computer. This configuration placed the nonlinear element on the digital computer. Maintaining the same approximate accuracy for nonlinear as for linear systems in digital computations generally requires a higher sampling or update rate. However, for this initial effort using the mild nonlinear condition the same sampling rate was used as with the linear system. The effects of partitioning on the operation of the variational and extended filters were shown with Monte Carlo results. Figure 20 shows the effects of time-critical operation on the variational filter, and Figure 21 shows

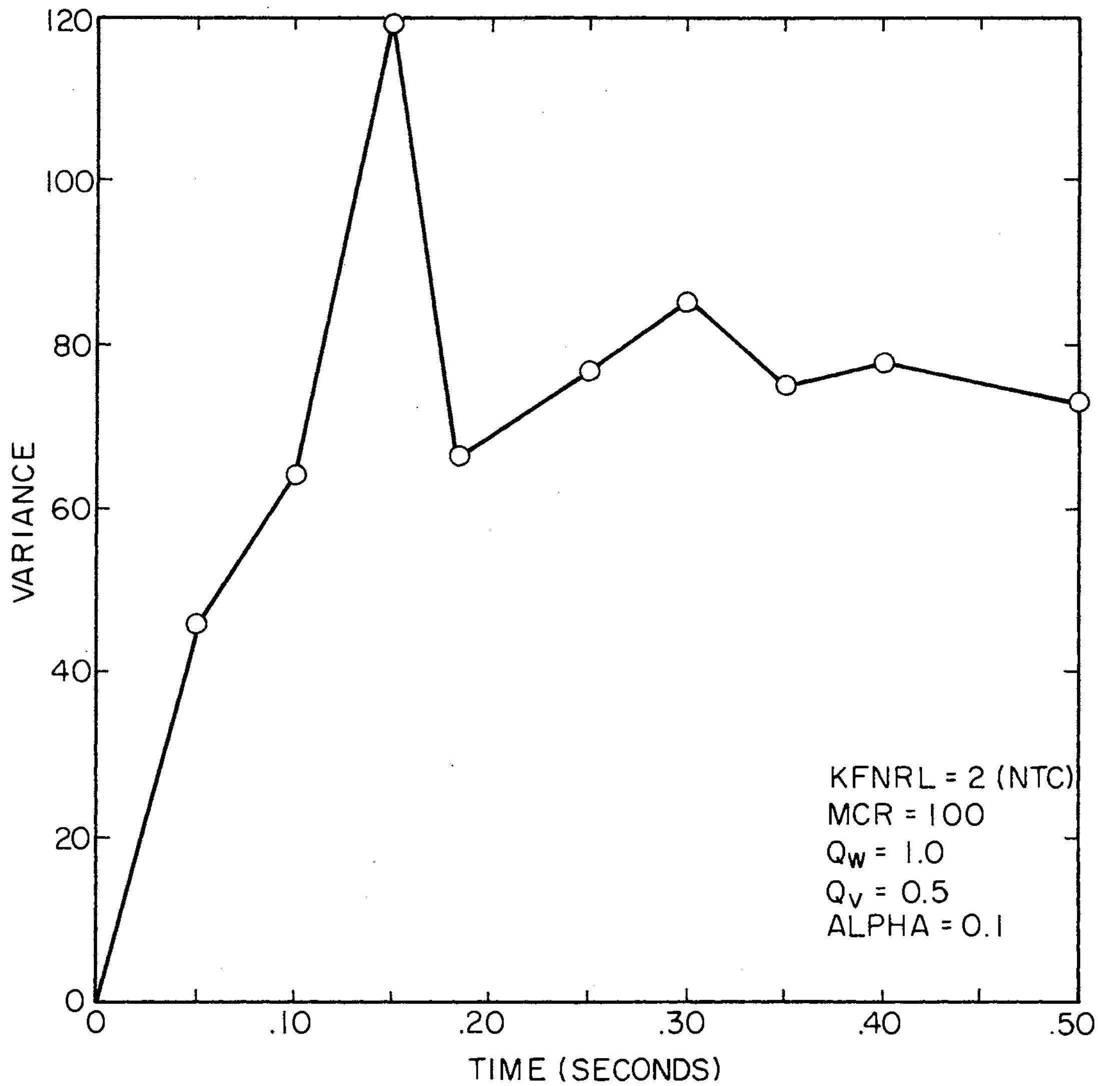


Figure 17. Error Variance for Harsh Nonlinear Operations.

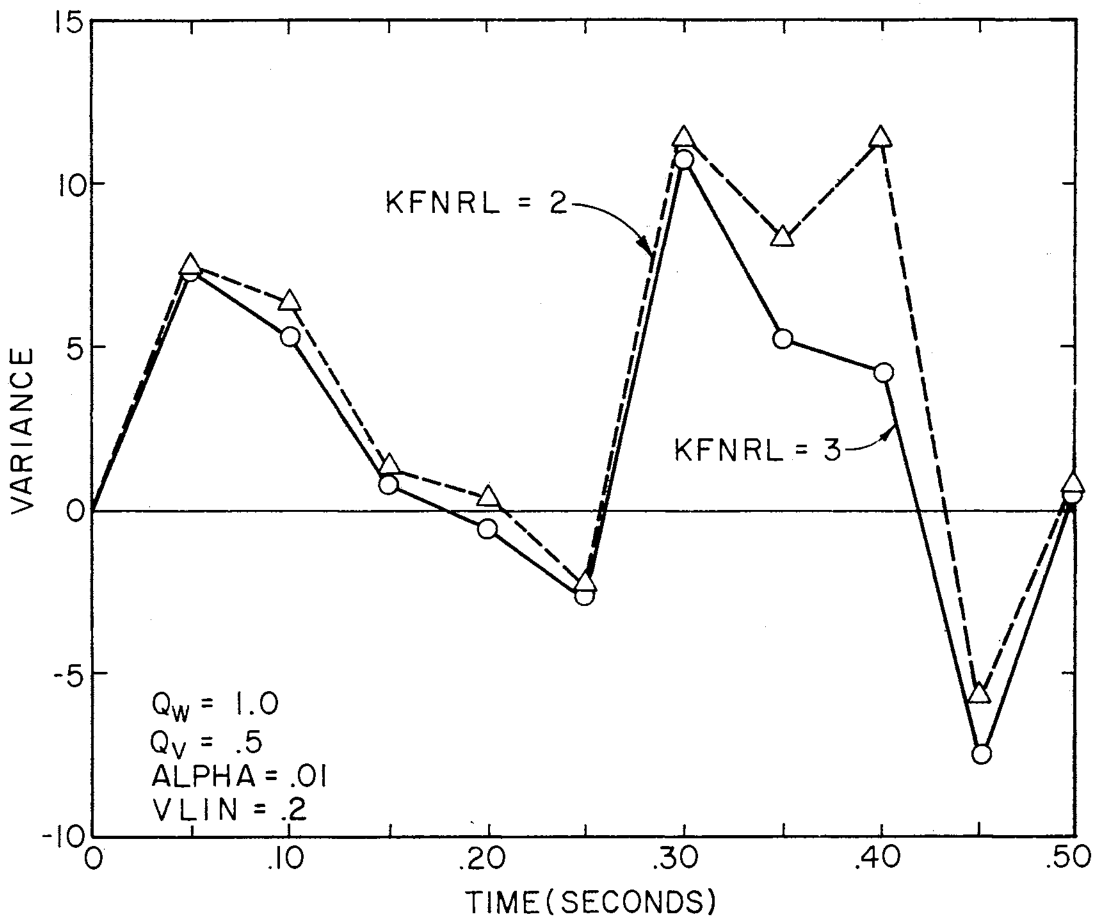


Figure 18. Sample Functions for Non-Time-Critical Operations for the Variational and Extended Filters.

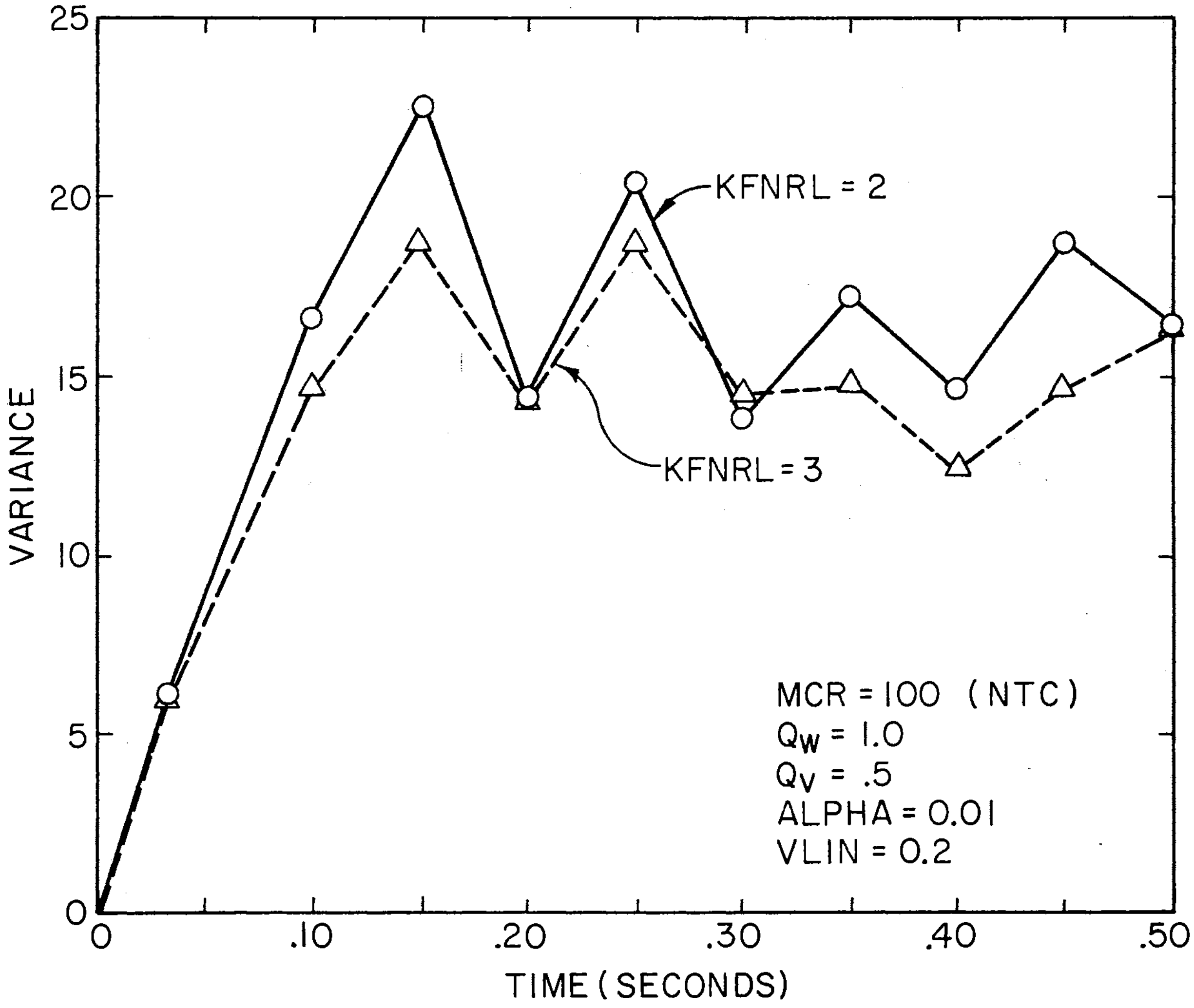


Figure 19. Non-Time-Critical Error Variance for Variational and Extended Filter Operations.

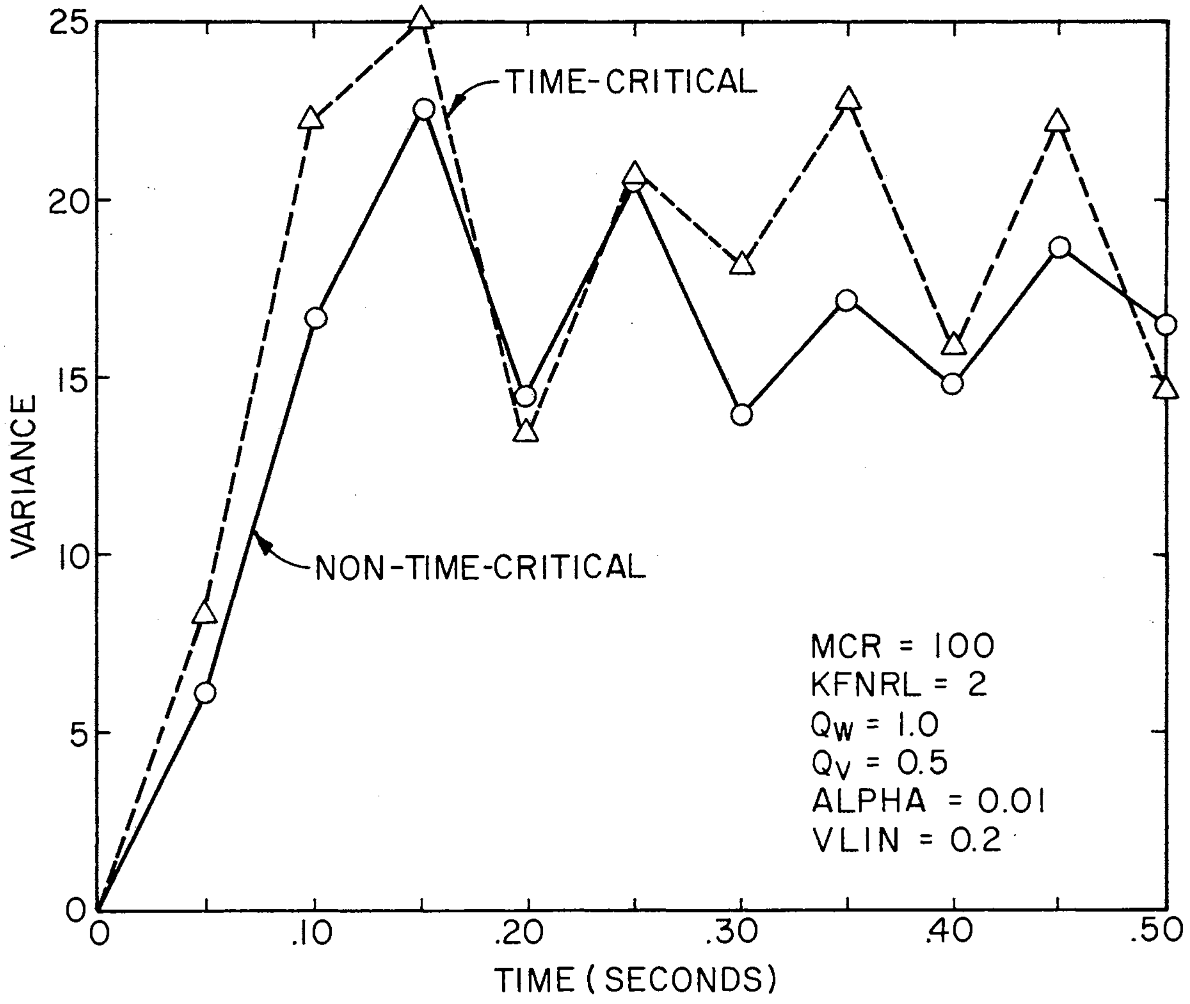


Figure 20. Effects of Hybrid Computer Partitioning on the Variational Filter.

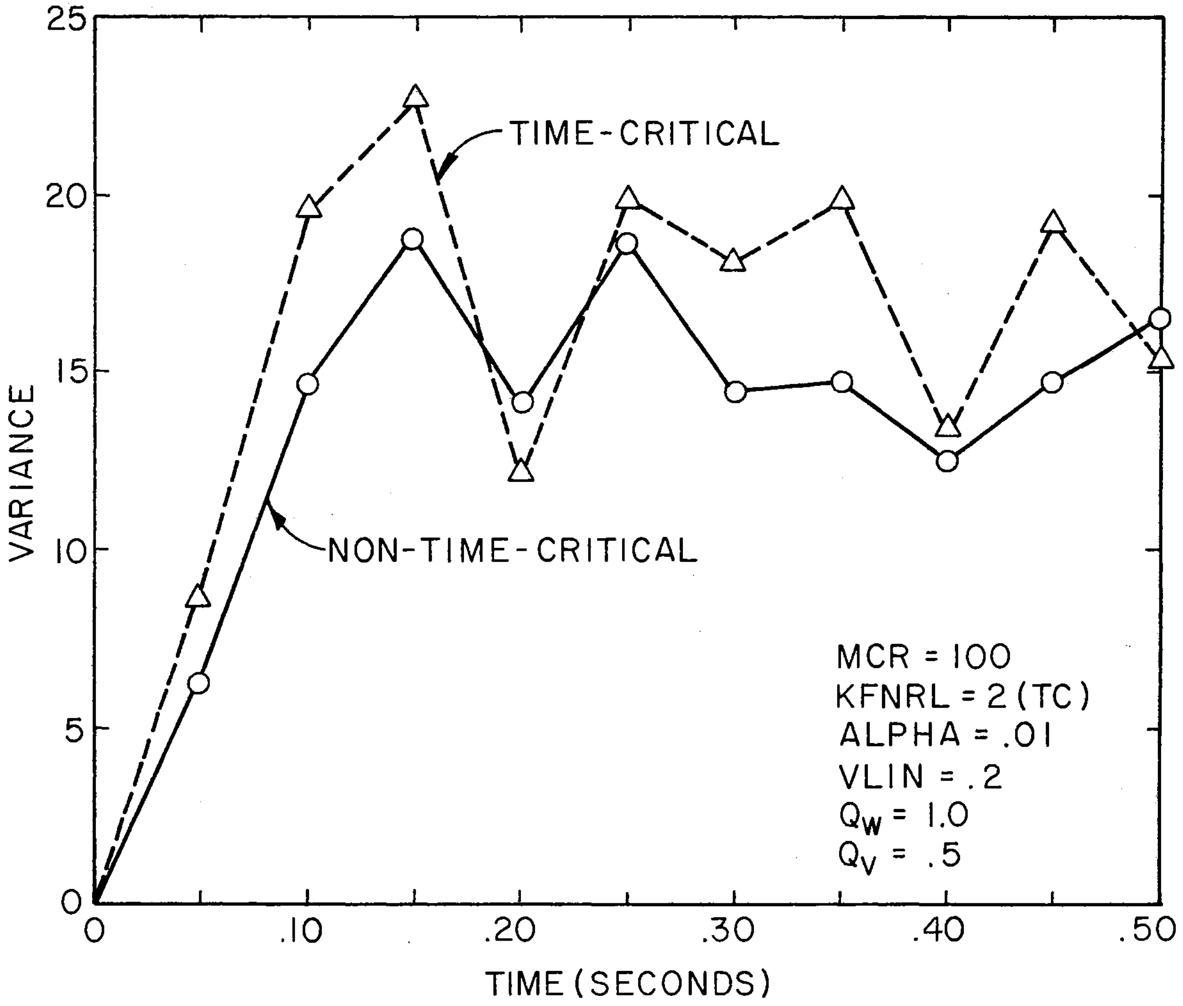


Figure 21. Effects of Hybrid Computer Partitioning on the Extended Filter.

the results for the extended filter.

The selection of a suitable operating noisy environment for this nonlinear system was by trial and error. What appeared to be suitable operating conditions were obtained in sample functions with $Q_H = 1.0$. However, Monte Carlo results showed a maximum error variance of 390 for what was selected as a mild nonlinear condition. This discrepancy emphasizes that single sample functions are not always a reliable indicator of the range of errors to be encountered in a stochastic nonlinear system. A zero-mean Gaussian white noise process with a variance $Q_H = 0.10$ was selected for a suitable representation of hybrid noise. As previously discussed, uniformly distributed white noise was used to represent the sampling error. This noise was a function of the sampling or update interval and did not change for the simulation described here. Figure 22 shows the Monte Carlo results of the variational filter operating in a time-critical environment with both modeled and unmodeled disturbances. Curve A shows the partitioned results with only modeled noises with variances $Q_W = 1.0$ and $Q_V = 0.5$. Curve B shows the results of adding unmodeled hybrid noise to the system variable transmitted to the digital computer. Curve C shows the results of adding both hybrid and sampling noise to the simulation. Similar results are shown for the extended filter in Figure 23.

Summary

A description of the time-critical simulation program for linear systems as expanded to include nonlinear systems and related Kalman filters has been presented in this chapter. The expanded simulation program was used with a particular nonlinear system with a cubic

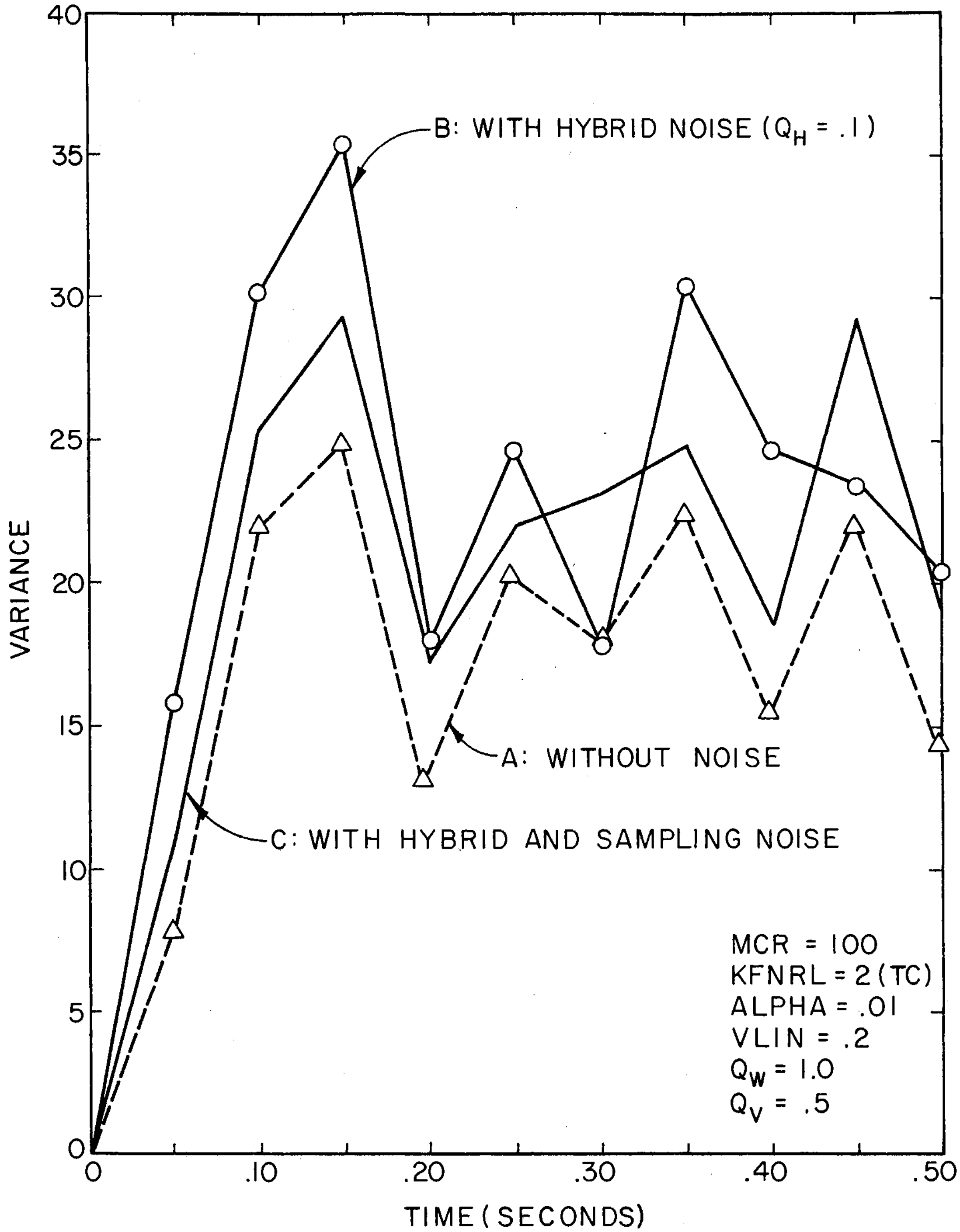


Figure 22. Effects of Hybrid and Sampling Noise on the Time-Critical Operation of the Variational Filter.

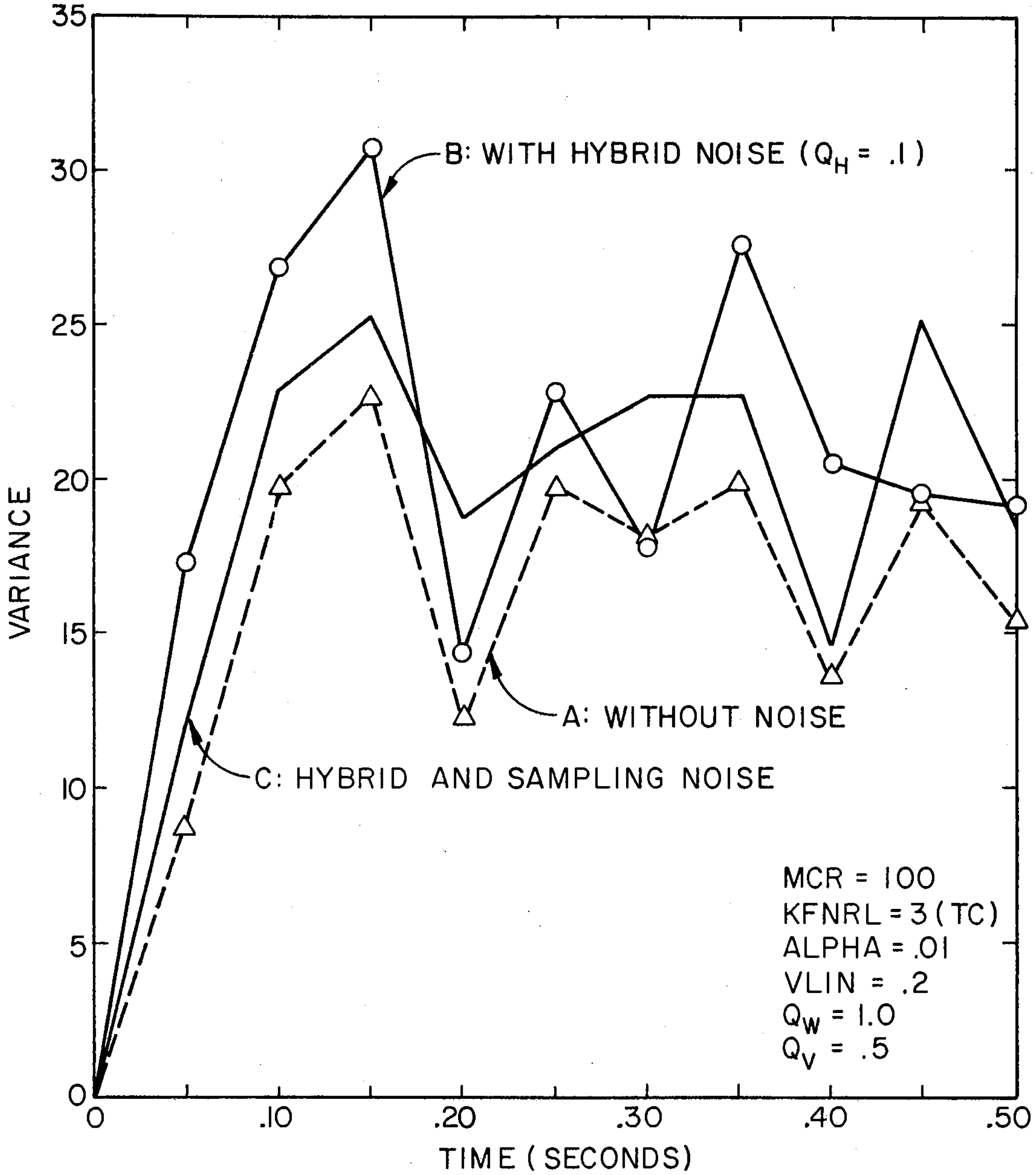


Figure 23. Effects of Hybrid and Sampling Noise on the Time-Critical Operation of the Extended Filter.

nonlinearity for both the variational filter and the extended filter. Results from 100 Monte Carlo runs were used to show the effects of hybrid computer partitioning and operation in a time-critical environment. Monte Carlo results were also used to show the effects of hybrid and sampling noise not modeled in the stochastic system. These results were obtained with an ideal hybrid computer frame time and without any compensation for the effects of the frame time. The simulation frame time was 12.5 milliseconds, while the actual digital computer time required for the variational filter was approximately 20 milliseconds, and the extended filter requirement was approximately 32 milliseconds. While these total results are significant, there is a need to consider further refinements to achieve improved operations.

CHAPTER IV

HYBRID SIMULATION REFINEMENTS

Program refinements and improvements for time-critical simulation which were investigated are described in this chapter. In particular, frame time compensation was applied to the time-critical stochastic simulation in the straightforward manner typically applied to deterministic systems. The reduction of the error variance was investigated by modifying the Kalman gains calculated in Chapters I and II to include the effects of hybridization noise. Additional improvements in the error variance were obtained by repartitioning the simulation model between the analog and digital computers.

Frame Time Compensation

One of the most serious errors introduced into hybrid computer simulation is due to the time delays inherent in digital computation. The problem is significantly more complex with closed-loop systems. The effects of delay due to digital computation have been extensively investigated for deterministic systems (6,14-19). The time delay error is serious enough in most hybrid systems to justify continuous compensation. The effect of the time delay τ is to cause the input to the analog or digital computer to arrive τ seconds late. One method of compensation is to use a prediction scheme to offset the delay.

A widely used compensation technique is to include in the simulation an approximation to the ideal predictor $e^{\tau s}$ to eliminate the time delay. The first two terms of a Taylor series expansion of the ideal prediction

$$e^{\tau s} = 1 + \tau s + \frac{1}{2}\tau^2 s^2 + \ldots \quad (4.1)$$

are used as an approximate prediction shceme, where s denotes the derivative operator. The variable to be delayed is passed through this predictor filter for approximate delay compensation. Specific implementation of the delay compensation is determined, to a large extent, by the specific problem being simulated. Bekey and Karplus (8) have given as three general approaches to time delay compensation implementation (1) the modification of the analog computer input by the addition of a voltage corresponding to $\tau s\underline{Y}(s)$. (2) the modification of the analog computer output by the addition of a term $\tau s\underline{X}(s)$, and (3) modification of the output of the digital computer by a term corresponding to $\tau sY(s)$. Generally, if the analog output is obtained from an integration, the second method is simple and straightforward to use. This method is referred to as the predistortion method since the analog output is distorted with the predictor before input to the time delay of digital computation. Monte Carlo results for the straightforward application of the predistortion technique to the variational filter are shown in Figure 24. Curve A depicts the time-critical operation with no frame time compensation, and Curve B shows the results obtained by using the predistortion technique. The increase in error variance is attributed to the derivative term used for prediction not representing an average or typically expected value across the sampling interval. This effect is due directly to the stochastic nature of the system. Improved error

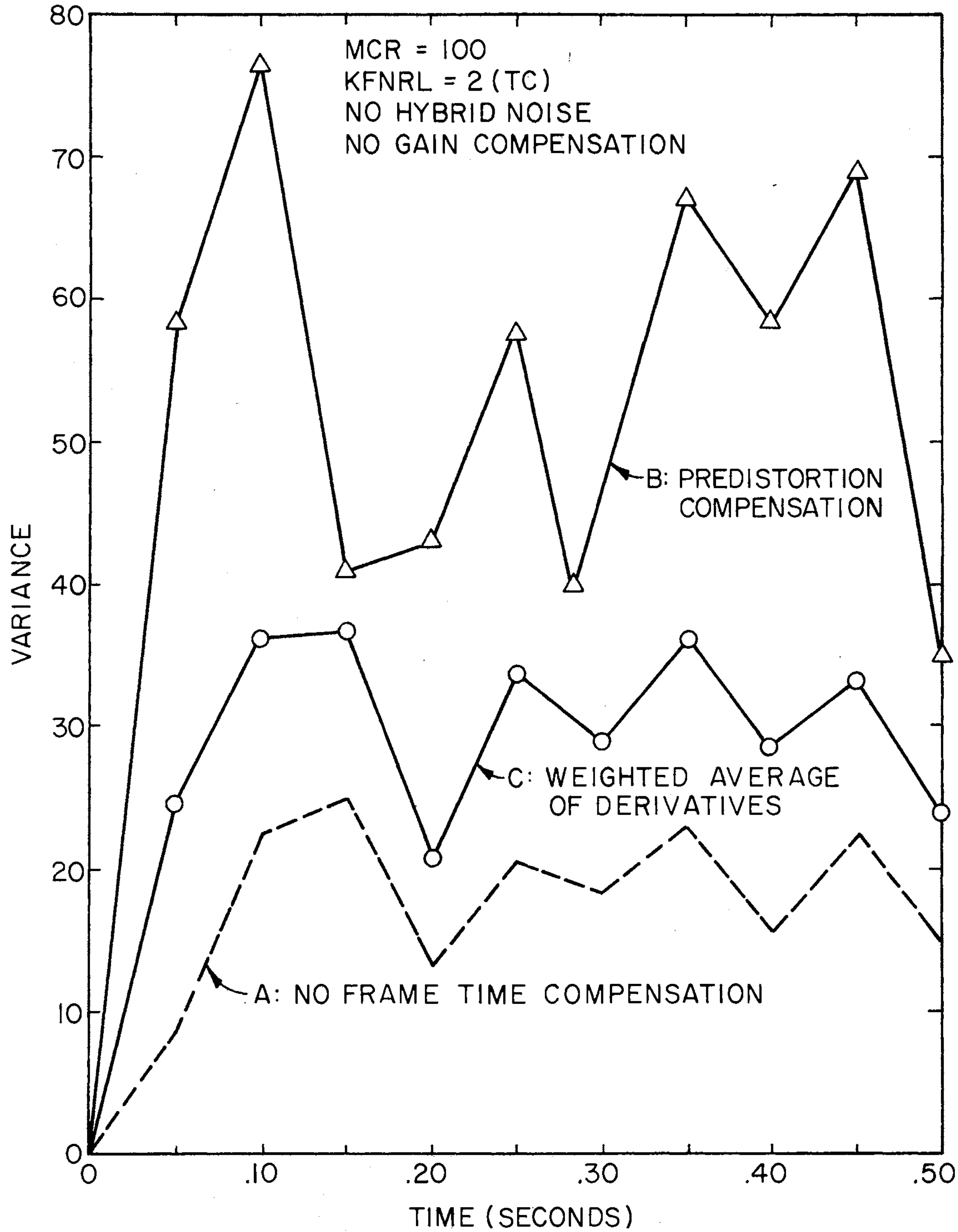


Figure 24. Effects of Frame Time Compensation on the Variational Filter.

variance results were obtained when a weighted average was used for the derivative term. Since the hybrid frame time was four integration intervals of the non-time-critical analog side, four values of the derivative terms were used to obtain an average. The results are shown as Curve C in Figure 24. These results show that the straightforward application of the predistortion technique does not give the desired results for the stochastic system used in this simulation. Furthermore, additional considerations must be given to the derivative term used when white noise disturbances have not been smoothed either through the system dynamics or other smoothing operations such as an averaging or filtering circuit. Similar Monte Carlo results are shown for the extended filter in Figure 25.

Kalman Gain Modification

The effects of the operating environment on the dynamical system not included in the model are designated as modeling errors. For the purpose of this research, hybrid system and sampling noise has been included in the simulation but not in the model of the dynamical system and related filters. The effects of these modeling errors have been discussed in Chapters II and III. One way to compensate for the unknown noise parameter is to include a fictitious noise source in the system model. If possible an estimate of the required statistics for the fictitious noise source can be obtained by a direct measurement of the operating environment. Otherwise, assumed values of the required statistics must be made and results verified by simulation evaluation.

Essentially, what was required to include the fictitious noise sources in the model was to determine particular values to be inserted

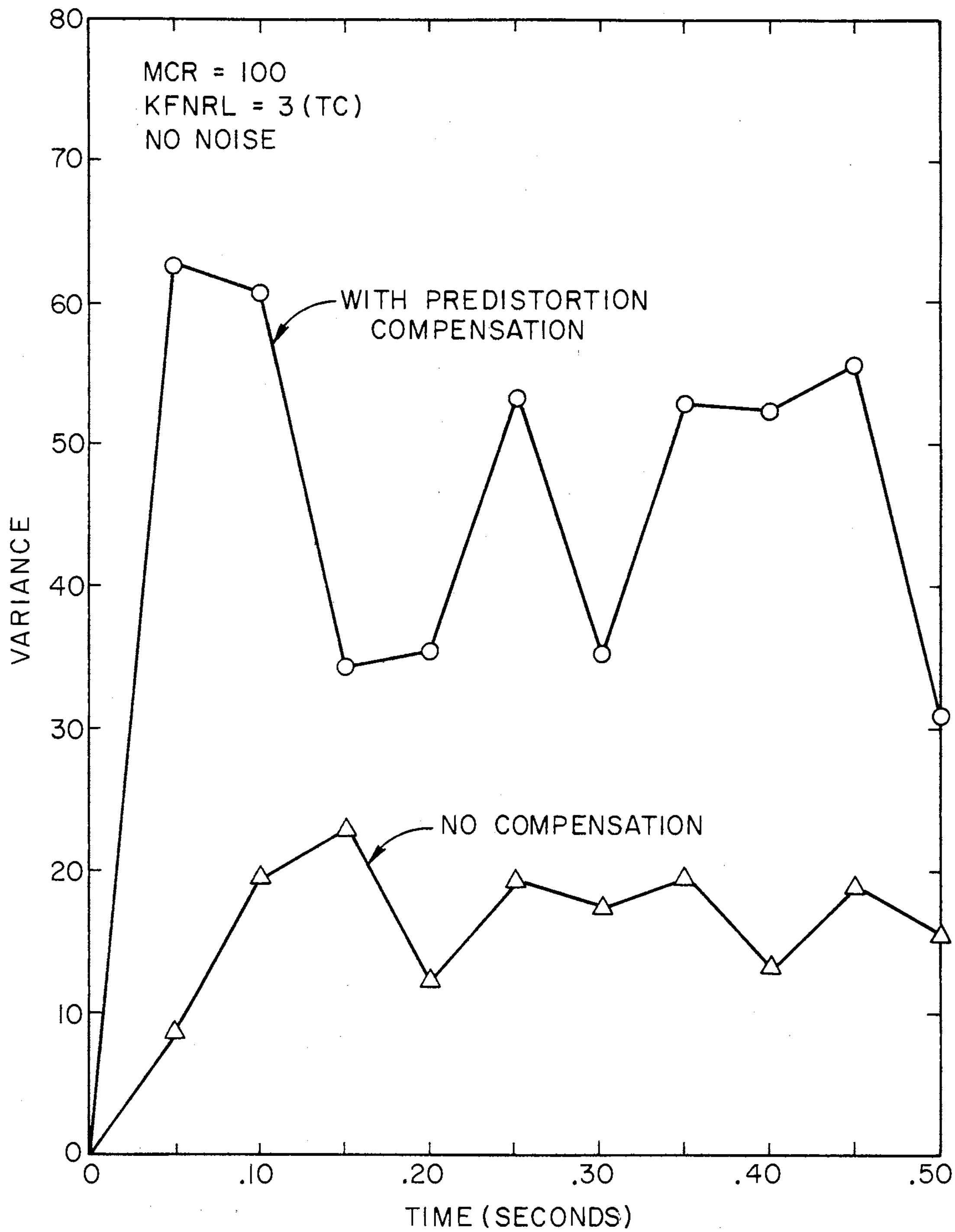


Figure 25. Effects of Frame Time Compensation on the Extended Filter.

in the noise coefficient matrix B(t) and the noise input covariance matrix $Q_{\underline{W}}(t)$. These matrices were used in the calculation of $B(t)Q_{\underline{W}}(t) B^T(t)$ in (1.12) for the error covariance matrix equation. The particular system used in this research had the modeled noise as input to Subsystem 2, which resulted in a nonzero value only in the (2,2) position of BQBT. The unmodeled noise was simulated as being added to the input of Sybsystem 1. The required expression for B(t) was obtained by using (1.21)-(1.24) to yield

$$\text{BQBT }(1,1) = V_K(t)*A(1,2)*A(1,2) \qquad (4.2)$$

where $V_K(t)$ is the variance of the fictitious noise source. Equation (4.2) was evaluated along the nominal trajectory $\underline{X}_N(t)$ for the variational filter and about the estimate of the state $\hat{\underline{X}}(t)$ for the extended filter.

The mean and variance of the hybrid noise used in this simulation was known and included in the initial effort for improved performance with gain modification. The Monte Carlo results for the variational filter are shown in Figure 26. As shown with Curve B the improvement in error variance was minimal. The variance of the fictitious error source included in the model was increased to $V_K = 0.4$, while the variance of the hybrid noise in the simulation remained at $Q_H = 0.1$. The Monte Carlo results from this operating condition are shown in Figure 27. An overall improved error variance of approximately five percent was achieved. Improved results were obtained with variances of the modeled fictitious noise source greater than the variance of the simulation noise source. This indicates that the effective variance of the noise in the hybrid simulation was greater than the simulated hybrid

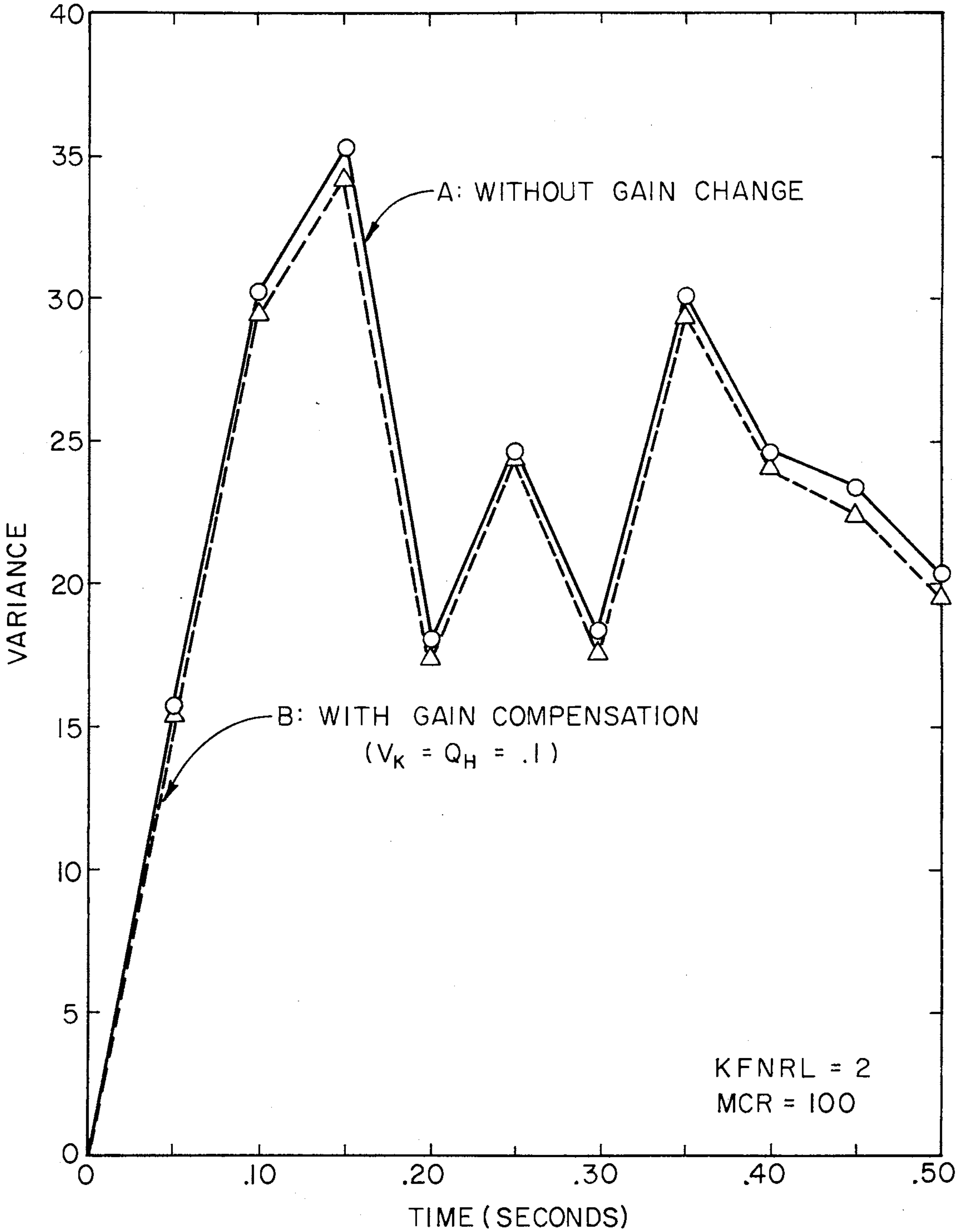


Figure 26. Effects of approximate compensation for hybrid noise by Kalman gain modification for the variational filter.

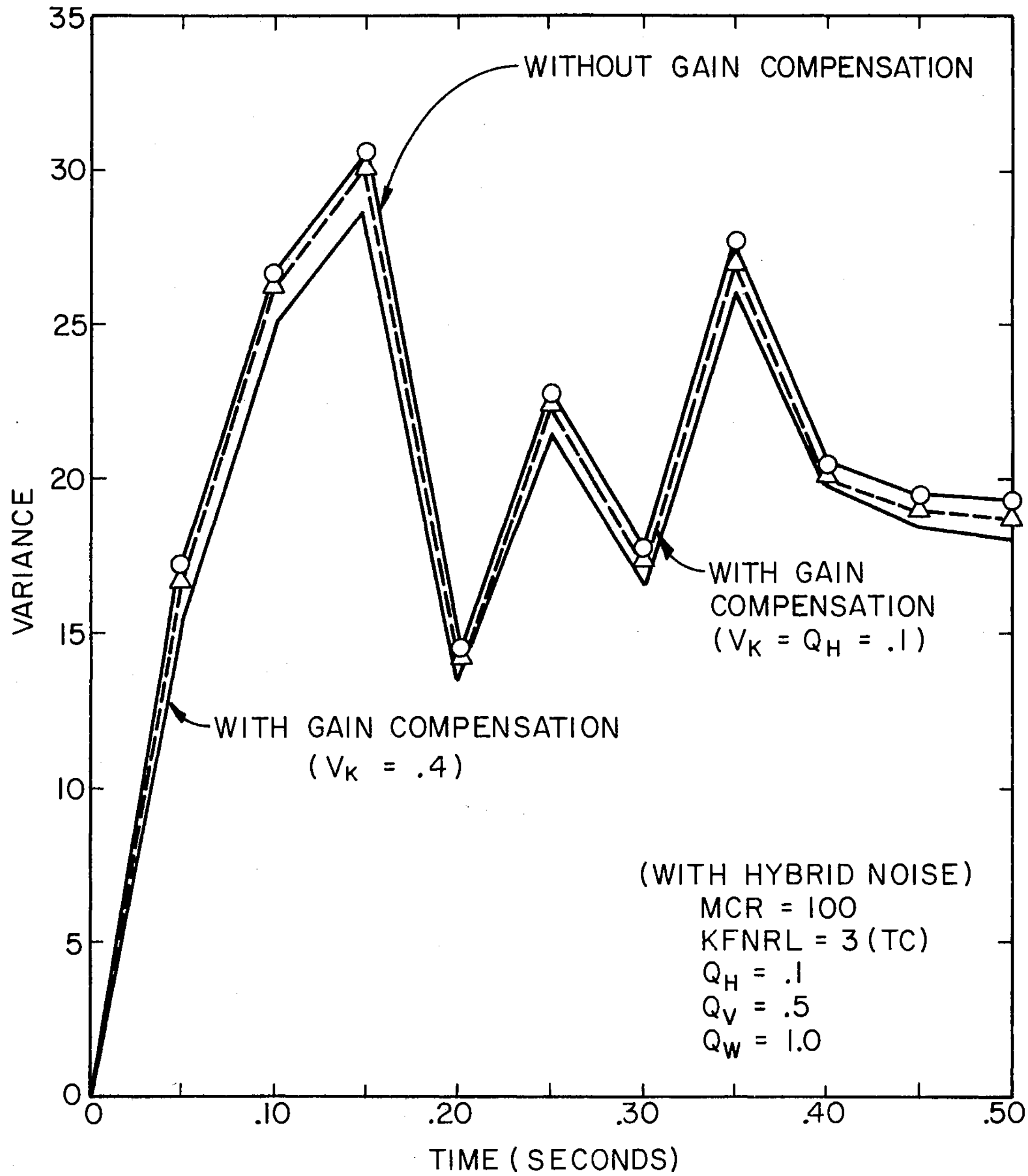


Figure 27. Effects of Approximate Compensation for Hybrid Noise by Kalman Gain Modifications for the Extended Filter.

noise, which results directly from the error introduced by partitioning and frame time delay. The conclusions consistent with the results shown in Figures 20 and 21 showing errors introduced by partitioning only.

Simulation Model Partitioning

The partitioning of a dynamical system between the analog and digital computer for time-critical operations remains one of the most important areas in hybrid computer simulation. The overall choice is influenced by the actual hardware-in-the-loop, the simulator and the performance requirements. With little or no guidelines other than a knowledge of the dynamical system and a detailed understanding of the simulator, an acceptable simulation of a complex system may be achieved only after several partitioning efforts. However, the final choice is determined by the sensitivity of the partitioned system to artificially introduced error sources and the system performance error budget.

Observing Figure 12, partitioning the Kalman filter with the high speed dynamics (that portion with A22) on the analog computer is the most obvious first choice. Curve B in Figure 28 shows the Monte Carlo results of such a partitioned choice without other compensation considerations. An approximate partitioned choice was achieved with the high speed dynamics on the analog side with continuous calculations of $\delta\hat{X}_2(t)$ and $\delta\dot{\hat{X}}_2(t)$ and the value of these variables being used as initial conditions on the digital calculation for the same quantities over the hybrid sampling interval. The improvement in error variance is shown in Curve C in Figure 28.

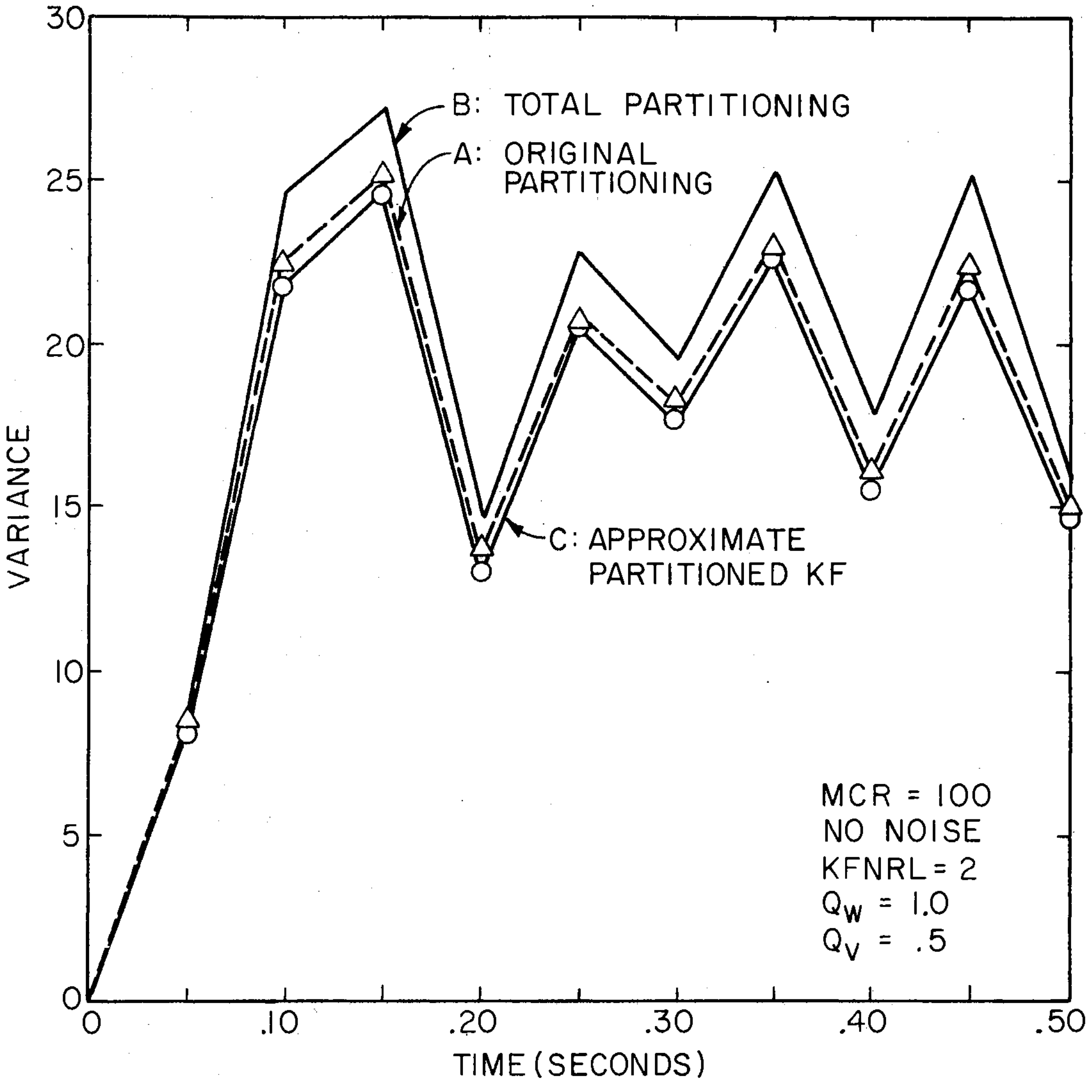


Figure 28. Effects of Partitioning the Kalman Filter for the Variational Mode.

Summary

A number of compensation techniques have been examined for improved time-critical hybrid computer operation for a particular nonlinear stochastic system. Straightforward application of frame time compensation as typically used for deterministic system did not give the desired results for the particular partitioned stochastic system. Results showed that a smoother derivative was required for a white noise input to the sampled digital simulation. While the Kalman gain modification was effective in reducing errors due to unmodeled noise sources, the estimated parameters must be obtained by simulation since the hybridization introduced apparent error sources not included in either the model or the simulation. Approximate re-partitioning of the simulation model between the analog and digital computer did result in an improved operating condition.

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

This research was accomplished by developing an all-digital computer simulation of a time-critical hybrid computer operation. The developed simulation program applied the methodology of time-critical hybrid computer operation to a stochastic estimation problem. The total simulation includes a linear stochastic system with the related Kalman filter and a nonlinear system with variational and extended filters. This research was accomplished by using continuous and continuous-discrete models of a particular stochastic system. Results obtained using the continuous, analog, or non-time-critical mode of operations were used as references for performance comparisons with the partitioned system operating with both modeled and unmodeled random disturbances. The modeled random parameters were included in the dynamical system operations while the unmodeled random noise sources were included in the simulation operating environment.

The significance of correctly implemented random disturbances was demonstrated with the linear system and related Kalman filter. The continuous-discrete representation of the random disturbance function was directly related to the hybrid computer frame time. Monte Carlo results showed that the error introduced by the direct partitioning process was greater than the error associated with the sampling noise. However, the combined effort of hybrid and sampling noise resulted in a significant error increase in the time-critical operating environment.

The importance of problem partitioning and a correctly modeled noisy environment was demonstrated with a particular nonlinear stochastic system. Monte Carlo results for non-time-critical and time-critical operations were compared for variation in error. The cubic nonlinearity chosen for this simulation required a small attenuation factor which emphasized the sensitivity of this system to unmodeled random disturbances. The difficulty in compensating for hybridization error in stochastic systems was shown by the direct application of various techniques typically used in the hybrid computer simulation of deterministic systems. Straightforward application of the predistortion method for frame time compensation resulted in an increase in error variance. This increase resulted from the direct input of white noise to the digital computer. Monte Carlo results showed that additional processing of the input signals to the digital computer is required for using this technique. Improved time-critical operating conditions were achieved by modifying the Kalman gains. This was accomplished by including in the dynamical system model a fictitious noise source representing the hybrid system noise. Additional improvements were obtained by repartitioning the system model between the analog and digital computers.

Recommendations For Further Work

The all-digital simulation program developed for this research is flexible and can be readily expanded to investigate more complex time-critical hybrid computer simulations. The particular stochastic estimation problem chosen for this simulation included only open-loop operation, or one-way communication between the analog and digital computers. This mode of operation eliminates the effect of accumulated phase error that exists in any closed-loop hybrid computer operation. The effect of hybrid operations on a closed-loop stochastic estimation problem is a logical extension of the results obtained in this research. A particular important closed-loop stochastic estimation problem might include a Kalman filter in the feedback loop.

Particularly important in any further research in the time-critical hybrid computer simulation of stochastic system is frame time compensation. The predistortion method used here includes the effect of a first-order hold circuit for which the results were not satisfactory. The results obtained with the minimal amount of sample averaging indicates that smoothing circuits might improve the time-critical operation. Another option for improving the frame time compensation is to increase the sampling rate. For a fixed simulation configuration, a reduced frame time could be achieved by the use of improved programming techniques such as an assembly or machine language. In addition, further frame time reductions could be achieved by using more efficient digital integration algorithms. An accuracy versus speed tradeoff for various algorithms should give options for improved operations. Any reduction in frame time requirements will directly reduce the error due to partitioning.

Precise guidelines for problem partitioning between the analog and digital computers are not available for either deterministic or stochastic systems. However, the results of this research clearly indicates that the partitioned problem should not have wide band white noise going directly to the digital computer. Further research is needed to establish some basis for partitioning a stochastic system with modeled and unmodeled random disturbances. This need is especially important for estimation problems that include the error covariance calculation as an on-line operation.

Improved operations in the hybrid noise environment were obtained by including a fictitious noise source in the dynamical system model, but the results from this effort were not conclusive. The partitioning process introduces an equivalent noise source in the hybrid computer operation that has not been included in the system model or simulation environment. However, improved operations were obtained when the variance of the fictitious noise was much greater than the hybrid noise. An equivalent noise representation of the error due to partitioning would give significant insight into the time-critical hybrid computer simulation of stochastic systems.

SELECTED BIBLIOGRAPHY

1. Bekey, George A., "Optimization of Multiparameter System by Hybrid Computer Techniques," Simulation, Fall 1963, pp. 19-32.

2. Korn, Granino A., "Hybrid-Computer Techniques for Measuring Statistics From Quantized Data," Simulation, April 1965, pp. 229-239.

3. Korn, Granino A., Random Process Simulation and Measurements, McGraw-Hill, 1967.

4. Steinmetz, H. L., "Automated Statistical Hybrid Simulation of Missile Trajectory Dispersions," Simulation, October 1970, pp. 167-177.

5. Schmidt, Herman, Electronic Analog/Digital Conversion, Van Nostrand Reinhold Company, 1967.

6. Vidal, Jacques and Walter J. Karplus, "Characterization and Compensation of Quantization Errors in Hybrid Computer Systems," IEEE International Convention Record, New York, Part III, pp. 236-241.

7. Rubin, A. I., "Analog/Hybrid-What It Was, What It Is, What It May Be," AFIPS Conference Proceedings, Fall Joint Computer Conference, Vol. 37 (1970), pp. 641-652.

8. Bekey, George A. and Walter J. Karplus, Hybrid Computation, John Wiley and Son Book Company, 1968.

9. Fineberg, Mark S. and Omri Serlin, "Multiprogramming for Hybrid Computation," Fall Joint Computer Conference, 1967, pp. 1-13.

10. Rowland, James R. and Willard M. Holmes, "Dynamic Programming for the Partitioning of Numerical Simulations With Hybrid Computer Applications," IEEE Transactions on Computers, May 1973, pp. 516-523.

11. Bedient, Conrad K. and Larry L. Dike, "The Lockheed Hybrid System-A Giant Step," Proceedings of the Fall Joint Computer Conference, 1968, pp. 663-682.

12. Soma, Gerald N., Joseph D. Crunkleton and Robert E. Lord, "A Priority Interrupt Oriented Hybrid Executive," Proceedings of the Fall Joint Computer Conference, 1968, pp. 683-699.

13. Graycon, T. J., R. A. Nolby and F. J. Sanson, "A High Performance Computing System for Time-Critical Application," Fall Joint Computer Conference, 1971, pp. 549-560.

14. Karplus, Walter J., "Error Analysis of Hybrid Computer Systems," Simulation, February 1966, pp. 120-136.

15. Mitchell, E. E. L., "A Generalized Hybrid Simulation for an Aerospace Vehicle," IEEE Transactions on Electronic Computers, Vol. EC-15, No. 3, June 1966, pp. 304-313.

16. Miura, T. T. and J. Iwata, "Effects of Digital Execution Time in a Hybrid Computer," Proceedings of the Fall Joint Computer Conference, 1963, pp. 251-265.

17. Mitchell, Ewart Edward Leslie, "The Effects of Digital Compensation for Computational Delay In A Hybrid Loop," Proceedings FJCC, 1967, pp. 103-108.

18. Vidal, J., W. J. Karplus and G. Kaludjlan, "Sensitivity Coefficients For The Correction of Quantization Errors In Hybrid Computer Systems," International Symposium on Sensitivity Analysis, JFAC, Dubrounik, Yugoslavia, September 1964, pp. 197-208.

19. Gelman, R., "Corrected Inputs-A Method For Improved Hybrid Simulation," Proceedings of the Fall Joint Computer Conference, 1963, pp. 267-276.

20. Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," J. Basic Engineering, Transactions ASME, March 1960, pp. 35-45.

21. Kalman, R. E. and R. S. Bucy, "New Results in Linear Filtering and Prediction Theory," J. Basic Engineering, Transactions ASME, Vol. 83D, March 1961, pp. 95-108.

22. Mendel, Jerry M. and Darrell L. Gieseking, "Bibliography on the Linear-Quadratic-Gaussian Problem," IEEE Transactions on Automatic Control, December 1971, pp. 847-869.

23. Gains, Harry T., "Flight Test of a Kalman Filter Doppler Aided Strap-Down Inertial Navigator," NAECON 1971 Record, pp. 91-99.

24. Schmidt, Stanley F., John W. Weinberg and John Lukesh, "Application of Kalman Filtering to the C-5A Guidance and Control System," Agardograph No. 139, Theory and Application of Kalman Filtering, Edited by C. T. Leondes, Chapter 13.

25. Meditch, J. S., Stochastic Optimal Linear Estimation and Control, McGraw-Hill Book Company, New York, 1969.

26. Jazwinski, Andrew H., Stochastic Processes and Filtering Theory, Academic Press, 1970.

27. Sage, Andrew P., and James M. Melsa, Estimation Theory: With Application to Communication and Control, McGraw-Hill, New York, 1970.

28. Bryson, Arthur E. and Yu-Chi Ho, Applied Optimal Control, Blaisdell Publishing Company, 1969.

29. Mendel, Jerry M., "Computational Requirements For A Discrete Kalman Filter," IEEE Transactions on Automatic Control, Vol. AC-16, No. 6, December 1971, pp. 748-758.

30. Simon, K. W. and A. R. Stubberud, "Reduced Order Kalman Filter," International Journal of Control, 1969, Vol. 10, No. 5, pp. 501-509.

31. Brown, R. J. and A. P. Sage, "Analysis of Modeling and Bias Errors In Discrete-Time State Estimation," IEEE Transactions on Aerospace and Electronics, Vol. AES-7, No. 2, March 1971, pp. 340.

32. Fitzgerald, Robert J., "Divergence of the Kalman Filter," IEEE Transactions on Automatic Control, Vol. AC-16, No. 6, December 1971, pp. 736-747.

33. Schlee, F. H., C. J. Standish and N. F. Toda, "Divergence in the Kalman Filter," AIAA Journal, Vol. 5, No. 6, 1967, pp. 1114-1120.

34. Price, Charles F., "An Analysis of the Divergence Problem in the Kalman Filter," IEEE Transactions on Automatic Control, December 1968, pp. 699-702.

35. Bierman, G. J., "Steady-State Covariance Computation For Discrete Linear Systems," JACC of the American Automatic Control Council, 1971, August 11-13, pp. 811-819.

36. Friedland, Bernard, "A Review of Recursive Filtering Algorithms," Spring Joint Computer Conference, 1972, pp. 163-180.

37. Nishimura, T., "Error Bounds of Continuous Kalman Filters and the Application to Orbit Determination Problems," IEEE Transactions on Automatic Control, June 1967, pp. 268-275.

38. Bucy, R. S., M. J. Merritt and D. S. Miller, "Hybrid Computer Synthesis of Optimal Discrete Nonlinear Filters," Proceedings of the Symposium on Nonlinear Estimation Theory and Its Applications, September 1971, pp. 59-87.

39. Tacker, Edgar C. and Thomas D. Linton, "Hybrid Simulation of an Optimal Stochastic Control System" Report No. THEMIS-LSU-T-TR-37, Louisiana State University, Baton Rouge, August 1970.

40. Tacker, Edgar C, "Computer Evaluation of Stochastic and Deterministic Controller Design For A Chemical Reactor," Louisiana State University, Baton Rouge, Louisiana, Report No. LSU-T-TR-44, February 1971.

41. Holmes, Willard, Charles Will and A. Jolly, "Time-Critical Hybrid Computer Simulation With Hardware-In-The-Loop Operation," U.S. Army Missile Command, Report RG-73-1, 1973.

42. Box, G. E. P. and M. E. Muller, "A Note On The Generation of Normal Deviates," Annals of Mathematical Statistics, Vol. 28, 1958, pp. 610-611.

43. Rowland, James R. and Willard M. Holmes, "Statistical Analysis Technique for Error Propagation in Large-Scale Missile Systems," U.S. Army Missile Command, Report No. RG-TR-71-19, April, 1971.

44. Rowland, James R. and Vijayendra M. Gupta, "Digital Simulation for Monte Carlo Analysis," Proceedings of the Fifteenth Midwest Symposium on Circuit Theory, University of Missouri, Rolla, Missouri, Vol. 1, Paper V.3, May 4-5, 1972.

APPENDIX

COMPUTER PROGRAM FOR SIMULATING TIME-CRITICAL HYBRID COMPUTER OPERATIONS FOR STOCHASTIC SYSTEMS

The digital computer simulation includes seven distinct operational modes: time-critical (ITC=1), non-time-critical (ITC=0), Monte Carlo operation (MCR=Number of Runs), single sample function (MCR=1), linear system and Kalman filter (KFNRL=1), nonlinear system with variational filter (KFNRL=2), and nonlinear system with extended filter (KFNRL=3). Additional options may be specified for some of the major modes. In particular, TSAMP = (hybrid computer frame time), ALPHA and VLIN are the weighting factor for the nonlinear and linear signals.

Standard Monte Carlo techniques are used to obtain statistical results from program operations. Pre-calculation of the Kalman gains are automatically bypassed for the extended filter mode of operation.

```
      DIMENSION DUN(1)
      DIMENSION RNU1(5),RNU2(5),RNGU(5)
      DIMENSION RNUM(10)
      DIMENSION AX2(10),DX2(10)
      DIMENSION DHVEC(10),HVEC(10)
      DIMENSION PDVEC(10),PVEC(10)
      DIMENSION DELZ(3,3),PHK(3,3)
      DIMENSION XE(2),DXN(10),XNM(10),DXEM(10),DXDEM(10),GKI(3,3)
      DIMENSION XN2(201),GK1(201),GK2(201),PT1(3,3),PH(3,3),XNN(10)
      DIMENSION DXDE(10),UFCT(2),UINPUT(1),HK(3,3),PN(3,3),XN1(201)
      DIMENSION DXM(10),GK(3,3),H(3,3),XNO(1),XN(10),HT(3,3),DXE(10)
      DIMENSION BT(3,3),QBT(3,3),PAT(3,3),X(10),DX(10),XINIT(2),CX(10)
      DIMENSION C(3,3),AT(3,3),AP(3,3),PD(3,3),Q(3,3),BQBT(3,3)
      DIMENSION RMEAN1(20),RMEAN2(20),TSAMP(10),A(3,3),P(3,3),B(3,3)
      DIMENSION VVSKF(10),DXDM(10),XNI(10),XA(10),DXA(10)
      DIMENSION XHSUM1(10),XHSUM2(10),XHSQ1(10),XHSQ2(10),DVSKF(10)
      DIMENSION QI(3,3),PHQI(3,3),P11(201),EPSM1(10),EPSSQ1(10)
      DIMENSION AEKF(201)
C
C
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C
C     THIS PROGRAM SIMULATES A HYBRID COMPUTER OPERATING IN
C     A TIME-CRITICAL ENVIROMENT AND A NON-TIME CRITICAL
C     OPERATIONAL MODE REPRESENTING AN ANALOG COMPUTER SIMULATION
C     THE PROGRAM HAS SEVEN DISTINCT MODES OF OPERATION
C
C     MCR=1 SINGLE SAMPLE FUNCTION
C     MCR= NUMBER OF SIMULATION RUNS REQUIRED IN MONTE CARLO STATISTIC
C     ITC=1 TIME CRITICAL OR HYBRID COMPUTER MODE
C     ITC=0  NON-TIME CRITICAL
C     KFNRL=1  LINEAR SYSTEM WITH LINEAR KALMAN FILTER
C     KFNRL=2  NONLINEAR SYSTEM WITH VARIATIONAL KALMAN FILTER
C     KFNRL=3  NONLINEAR SYSTEM WITH EXTENDED KALMAN FILTER
C
C     OTHER OPTIONS INCLUDES
C     ITSAMP= NUMBER OF NON-TIME CRITICAL STEP SIZES INCLUDED
C     IN THE HYBRID FRAME TIME
C     VLIN= WEIGHTING VALUE ON LINEAR SIGNAL COUPLING
C     SUBSYSTEM2 AND SUBSYSTEM1
C     ALPHA= WEIGHTING VALUE ON CUBIC NONLINEARITY COUPLING
C     SUBSYSTEM1 AND SUBSYSTEM2
C
C     SEPERATE RANDOM NUMBER GENERATORS ARE USED FOR GENERATING
C     THE WHITE NOISE PROCESS FOR  THE MODELED SYSTEM NOISE
C     AND UNMODLED HYBRID SYSTEM AND SAMPLING NOISE
C
C
C                             WILLARD M. HOLMES
C
C
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C
C
```

```
      CCMMON INDEX3
      COMMON DXM
      COMMON MX
      CCMMON TINIT,TFINAL,IPRINT
      COMMON KUTTA,DT,NX,X,DX,TIME,DXA,DK(2)
      COMMON XINIT
      COMMON UFCT
      COMMON PAT,AP,C,PH,PQHI,HT,HK,PN,PT1
      CCMMON AT
      CCMMON DUM(1),IX,DUM1,XNORM(2)
      READ (5,1013) NX
      READ (5,1013) IPRINT
      READ (5,1041) DT
 1041 FORMAT (F10.6)
      READ (5,1012) TINIT,TFINAL,XINIT(1),XINIT(2)
      WRITE (6,1003)
 1003 FORMAT (1H1,T5,'INITIAL CONDITIONS')
      WRITE (6,1001) NX,DT,TINIT,TFINAL,IPRINT,XINIT(1),XINIT(2)
 1001 FORMAT (1H0,5X,I2,3X,F8.6,3X,F8.6,F10.6,I3,3X,F10.6,F10.6)
      READ (5,1013) MX
 1013 FCRMAT (I2)
      READ (5,1012) ((A(I,J),I=1,NX),J=1,NX)
      READ (5,1012) ((P(IP,JP),IP=1,NX),JP=1,NX)
 1012 FCRMAT (4F10.6)
      READ (5,1014)((B(IB,JB),IB=1,NX),JB=1,MX)
 1014 FORMAT (4F10.6)
      READ (5,1031) ((Q(IQ,JC),IQ=1,MX),JQ=1,MX)
 1031 FORMAT (4F10.6)
      WRITE (6,1018) NX,MX,IPRINT
 1018 FORMAT (1H0,5X,'NX=',I2,5X,'MX=',I2,5X,'IPRINT=',I2)
      WRITE (6,1020)
 1020 FCRMAT (1H0,5X,'P MATRIX')
      WRITE (6,1016)((P(JP,KP),JP=1,NX),KP=1,NX)
 1016 FORMAT (1H0,5X,4E16.8)
      WRITE (6,1021)
 1021 FORMAT (1H0,5X,'B MATRIX')
      WRITE (6,1017)((B(KB,LB),KB=1,NX),LB=1,MX)
 1017 FORMAT (1H0,10X,4F10.6)
      WRITE (6,1033)
 1033 FORMAT (1H0,5X,'Q MATRIX')
      WRITE (6,1032) ((Q(KQ,LQ),KQ=1,MX),LQ=1,NX)
 1032 FORMAT (1H0,5X,4E16.8)
      WRITE (6,1034)
 1034 FCRMAT (1H0,5X,'Q MATRIX   BY ROWS')
      WRITE (6,1035) Q(1,1),Q(1,2),Q(2,1),Q(2,2)
 1035 FCRMAT (1H0,10X,4E16.8)
C
C     *************************
C     NX= ORDER OF SYSTEM
C     MX= NUMBER OF SYSTEM INPUTS
C     INPUT MATRIX DATA MUST BE BY COLUMN,A11,A21,A12,A22,ETC
C
C     ********************
C
```

```
C
      HDT=.5*DT
      H(1,1)=1.0
      H(1,2)=0.0
      H(2,1)=0.
      H(2,2)=0.0
      UFCT(1)=0.0
      UFCT(2)=0.
      TNOM=0.0
      XN1(1)=0.0
      ERROR1=0.0
      ERROR2=0.0
      GK1(1)=0.0
      GK2(1)=0.0
      GK(1,2)=0.0
      GK(2,2)=0.0
C     **********************************
C
C     FIRST ELEMENT IN XN1 AND XN2 IS SET EQUAL TO THE
C     INITIAL CONDITIONS ON XN(1)AND XN(2)
C
C     **********************************
      ICYCLE=(TFINAL-TINIT)/DT+.5
      ICAL=ICYCLE/IPRINT
      DO 28 ILL=1,20
      RMEAN1(ILL)=0.0
      RMEAN2(ILL)=0.0
   28 CONTINUE
      DO 29 K16=1,10
      XHSUM1(K16)=0.0
      XHSUM2(K16)=0.0
      XHSQ1(K16)=0.0
      XHSQ2(K16)=0.0
      EPSM1(K16)=0.0
      EPSSQ1(K16)=0.0
   29 CONTINUE
      DUM1=0.
      DUM2=0.
      CALL TRANSA (B,NX,MX,BT)
      CALL MATMUL (Q,BT,MX,NX,MX,QBT)
      CALL MATMUL (B,QBT,NX,NX,MX,BQBT)
      WRITE (6,996) BQBT(1,1),BQBT(1,2),BQBT(2,1),BQBT(2,2)
      IPRT=IPRINT
C     *******************************************
C
C     FOR APPLICATION OF VARIOUS KALMAN FILTER CONFIGURATION TO LINEAR
C     AND NONLINEAR SYSTEMS ,THE VARIABLE  KFNRL  HAS THE VALUES
C     KFNRL=1  APPLICATION OFKALMAN FILTER TO LINEAR SYSTEMS
C     KFNRL=2  REGULAR APPLICATION OF KALMAN FILTER TO VARIATION
C     AROUND A NOMINAL TRAJECTCRY
C     KFNRL=3  APPLICATION OF THE EXTENDED KALMAN FILTER
C
C
C     **********************************
```

```
      ITC=1
      MCR=100
      MX=2
      KFNRL=2
      ALPHA=.01
      VLIN=.2
      VK=.1
      ITSAMP=4
      UINP=10.
      QV=.5
      TSAM1=ITSAMP
      HSAMP=TSAM1*DT
      KCUNT=0
      TDIG=0.
      DDT=.5*HSAMP
      QI(1,1)=2.0
      QI(1,2)=0.
      QI(2,1)=0.
      QI(2,2)=0.
      WRITE (6,1019)
 1019 FORMAT (1H0,5X,'A MATRIX')
      A(1,1)=-10.
      A(1,2)=10.0*VLIN
      A(2,1)=0.0
      A(2,2)=-50.0
      WRITE (6,1024) A(1,1),A(1,2),A(2,1),A(2,2)
 1024 FORMAT (1H0,5X,'A(1,1)='F10.6,3X,'A(1,2)='F10.6,3X,'A(2,1)='F10.6
     1,3X,'A(2,2)='F10.6)
      QW=Q(1,1)
      ZW=QW/DT
      IF(ITC.EQ.1) GO TO 481
      ZV=QV/DT
  481 CONTINUE
      IF (ITC.EQ.0) GO TO 482
C     ****************************************************************
C     KALMAN GAIN MODIFICATION FOR IMPROVED TIME CRITICAL OPERATION
      BQBT(1,1)=VK*A(1,2)*A(1,2)
C
C     ****************************************************************
      QH=.1
      ZV=QV/(TSAM1*DT)
      ZH=QH/(TSAM1*DT)
      SIGH=SQRT(ZH)
  482 CONTINUE
      SIGV=SQRT(ZV)
      SIGW=SQRT(ZW)
      DFT=0.0
      RHYBN=0.
      RSAMPN=0.
      WRITE (6,775) KFNRL,MCR,QW,QV,QH
  775 FORMAT (1H0,5X,'KFNRL='I1,3X,'MCR='I2,3X,'QW='F10.6,3X,'QV='F10.6,
     12X,'QH='F10.6)
      WRITE (6,776) ALPHA,VLIN,ITSAMP,ITC,HSAMP
  776 FORMAT (1H0,5X,'ALPHA='F10.6,2X,'VLIN='F10.6,2X,'ITSAMP='I2,
```

```
     12X,'ITC='I1,2X,'HSAMP='F10.6)
      WRITE (6,777) ZW,ZV,ZH
 777  FORMAT (1H0,5X,'ZW='F10.6,2X,'ZV='F10.6,2X,'ZH='F10.6)
C     ******************************************
C
C     CALCULATION CF NOMINAL SYSTEM TRAJECTORY
C
C     **********************************
      INDEX1=1
      INDEX4=0
      XN(1)=XINIT(1)
      XN(2)=XINIT(2)
      XN1(1)=XINIT(1)
      XN2(1)=XINIT(2)
      WRITE (6,770)
 770  FORMAT (1H1,10X,'NOMINAL SYSTEM RESPONSE')
      DO 700 INOM=1,ICAL
      WRITE (6,1113) TNOM,XN(1),XN(2)
      DO 701 INOUT=1,IPRT
      DO 8 KUTTAN=1,4
      GO TO (12,22,12,22),KUTTAN
   22 CONTINUE
      TNOM=TNOM+HDT
   12 CONTINUE
      XNORM(1)=0.
      CALL SYSEQ (UINP,XNORM,VLIN,ALPHA,DX,XN)
      CALL RUNK (KUTTAN,DT,DUM2,NX,DX,XN)
    8 CONTINUE
      INDEX1=INDEX1+1
      XN1(INDEX1)=XN(1)
      XN2(INDEX1)=XN(2)
 701  CONTINUE
 700  CONTINUE
      WRITE (6,1113) TNOM,XN(1),XN(2)
 1113 FORMAT (1H0,5X,'TNOM='F10.6,3X,'XN1(INDEX1)='E16.8,3X,
     1'XN2(INDEX1)='E16.8)
      IF(KFNRL.GT.1) GO TO 14
      WRITE (6,1027)
 1027 FORMAT (1H0,10X,'***LINEAR SYSTEM APPLICATION OF KALMAN FILTER*')
   14 CONTINUE
C     ******************************************
C
C     PRE-CALCULATION OF CCVARIANCE AND KALMAN GAIN
C
C     ******************************************
      IF(KFNRL.EQ.3) GO TO 469
      TCOV=0.
      DC 445 K18=1,10
      PDVEC(K18)=0.0
      PVEC(K18)=0.0
 445  CONTINUE
      DO 446 K19=1,NX
      DO 446 K20=1,NX
      P(K19,K20)=0.0
```

```
 446  CCNTINUE
      INDEX1=1
      DO 440  IPK=1,ICAL
      DO 441  IPOUT=1,IPRT
      DO 442  KUTTAP=1,4
      GC TO (13,23,13,23),KUTTAP
  23  CONTINUE
      TCOV=TCOV+HDT
  13  CONTINUE
      GO TO (4,5),KFNRL
 5    CONTINUE
      XNI(1)=XN1(INDEX1)
      XNI(2)=XN2(INDEX1)
      CALL AA(XN2,INDEX1,ALPHA,VLIN,A)
C     ****************************************************************
C     KALMAN GAIN MODIFICATION FOR IMPROVED TIME CRITICAL OPERATION
      BQBT(1,1)=VK*A(1,2)*A(1,2)
C
C     ****************************************************************
  4   CONTINUE
      CALL TRANSA (A,NX,NX,AT)
      CALL COVPD (P,A,QI,B,H,NX,MX,BQBT,AT,PD)
      PDVEC(1)=PD(1,1)
      PDVEC(2)=PD(1,2)
      PDVEC(3)=PD(2,2)
      LP=3
      CALL RUNK (KUTTAP,DT,DUM1,LP,PDVEC,PVEC)
      P(1,1)=PVEC(1)
      P(1,2)=PVEC(2)
      P(2,1)=PVEC(2)
      P(2,2)=PVEC(3)
 442  CONTINUE
      INDEX1=INDEX1+1
      CALL MATMUL (P,H,NX,MX,NX,PHK)
      CALL MATMUL (PHK,QI,NX,MX,MX,GK)
      GK1(INDEX1)=GK(1,1)
      GK2(INDEX1)=GK(2,1)
 441  CONTINUE
      WRITE (6,443) TCOV,GK1(INDEX1),GK2(INDEX1)
 443  FORMAT (1H0,5X,'TCOV='F10.6,3X,'GK1(INDEX1)='E16.8,3X,
     1'GK2(INDEX1)='E16.8)
      WRITE (6,444) P(1,1),P(1,2),P(2,1),P(2,2)
 444  FORMAT (1H0,5X,'P(1,1)='E16.8,2X,'P(1,2)='E16.8,2X,
     2'P(2,1)='E16.8,2X,'P(2,2)='E16.8)
 440  CCNTINUE
 469  CONTINUE
C     **********************************
C
C     INITILIZATION OF RANDOM NUMBER GENERATORS
C     OUTSIDE OF MONTE CARLO LOOP
C
C     **********************************
      XNORM(1)=0.
      XNORM(2)=0.0
```

```
      IX=31571
      JX=31571
      DUM(1)=0.1
      DUN(1)=.1
      DO 100 MC=1,MCR
C     ********************************************
C
C     INITIAL CONDITIONS FOR MONTECARLO          *****************
C
C     ********************************
      DO 665 KP1=1,NX
      DO 665 KP2=1,NX
      P(KP1,KP2)=0.0
      DELZ(KP1,KP2)=0.0
  665 CONTINUE
C   *********************************************
C
C  FIRST ELEMENT IN XN1 AND XN2 ARRAY CONTAINS I C ON X NOMINAL
C
C     *********************************
      GK(1,1)=0.0
      GK(2,1)=0.0
      INDEX3=0
      DO 10 JK=1,NX
      X(JK)=XINIT(JK)
  10  CONTINUE
      DO 27 K17=1,10
      DVSKF(K17)=0.0
      VVSKF(K17)=0.0
  27  CONTINUE
      XNM(1)=XINIT(1)
      XNM(2)=XINIT(2)
      AX2(1)=XINIT(2)
      AX2(2)=XINIT(1)
      DXM(1)=0.0
      DXM(2)=0.0
      DXEM(1)=0.0
      DXEM(2)=0.0
      HVEC(4)=0.0
      HVEC(5)=0.0
      HVEC(6)=0.0
      TIME=TINIT
      T=TINIT
      INDEX2=1
      XHAT1=XINIT(1)
      XHAT2=XINIT(2)
C     *********************************
C
C     NONLINEAR SYSTEM VARIATION ABOUT A NOMINAL TRAJECTORY
C
C     *********************************
      IF(MCR.GT.1) GO TO 1111
      WRITE(6,882)
 882  FORMAT (1H1,5X,'SYSTEM RESPONSE AND KALMAN GAIN CALCULATION FOR
```

```
     2SINGLE RUN')
 1111 CONTINUE
      DO 50 IPL=1,ICAL
      IF(MCR.GT.1) GO TO 561
      WRITE (6,989) T,XHAT1,XHAT2,GK1(INDEX2),GK2(INDEX2)
      WRITE(6,990) ERROR1,ERROR2
      WRITE(6,991) XNM(1),DELZ(1,1),XNM(2)
      WRITE(6,995) DXEM(2),XNORM(1),XNORM(2),DXDEM(1),DXDEM(2)
      WRITE(6,993) XN1(INDEX2),DXEM(1),RHYBN,RSAMPN
      WRITE (6,996) BQBT(1,1),BQBT(1,2),BQBT(2,1),BQBT(2,2)
 561  CONTINUE
      DO 25 I=1,IPRT
C     *********************************
C  INITIAL CONDITION GENERATES FIRST ELEMENT IN GK1 AND GK2
C     *********************************
      GO TO (38,39,40),KFNRL
  40  CONTINUE
      XNI(1)=XHAT1
      XNI(2)=XHAT2
      AEKF(2)=XHAT2
      CALL AA(AEKF,2,ALPHA,VLIN,A)
      CALL TRANSA (A,NX,NX,AT)
C     ***************************************************************
C
C     KALMAN GAIN MODIFICATION FOR IMPROVED TIME CRITICAL OPERATION
      BQBT(1,1)=VK*A(1,2)*A(1,2)
C
C     ***************************************************************
      GO TO 41
  39  CONTINUE
      CALL AA(XN2,INDEX2,ALPHA,VLIN,A)
  38  CONTINUE
  41  CONTINUE
      CALL RANDOM (DUM,IX,XNORM)
      XNORM(1)=XNORM(1)*SIGW
      XNORM(2)=XNORM(2)*SIGV
      IF(ITC.EQ.1) GO TO 456
      DO 2 KUTTA=1,4
      GO TO (11,21,11,21),KUTTA
  21  TIME=TIME+HDT
  11  CONTINUE
      CALL SYSEQ (UINP,XNORM,VLIN,ALPHA,DXDM,XNM)
      DELZ(1,1)=XNM(1)+XNORM(2)-XN1(INDEX2)
      CALL KALMAN (DELZ,GK,H,A,NX,MX,DXEM,DXDEM)
      IF(KFNRL.LT.3) GO TO 447
      CALL COVPD (P,A,QI,B,H,NX,MX,BQBT,AT,PD)
 447  CONTINUE
      CALL ARTOVE (DXDM,DXDEM,PD,XNM,DXEM,P,DVSKF,VVSKF)
C**************************************
C
C     LX IS EQUAL TO THE NUMBER OF ELEMENTS IN VECTOR DVSKF
C
C**************************************
      IF (KFNRL.LT.3) GO TO 778
```

```
      LX=7
  778 CONTINUE
      IF (KFNRL.EQ.3) GO TO 779
      LX=4
 779  CONTINUE
      CALL RUNK (KUTTA,DT,DUM1,LX,DVSKF,VVSKF)
      CALL VETOAR (VVSKF,XNM,DXEM,P)
  2   CONTINUE
 456  CONTINUE
      IF(ITC.EQ.0) GO TO 460
C     ******************************************
C
C     BEGIN TIME CRITICAL OPERATIONAL LOOP
C
C     ITC=1,TIME-CRITICAL OR HYBRID OPERATIONS
C     ITC=0, NON-TIME CRITICAL OR TOTAL ANALOG
C     ITSAMP= NUMBER OF NON-TIME CRITICAL TIME STEPS IN THE
C     DIGITAL COMPUTER FRAME TIME FOR HYBRID OPERATION
C     ALPHA= WEIGHTING ON NONLINEAR EFFECTS
C     VLIN= WEIGHTING FACTOR OF SUB2 OUTPUT TO SUB1 INPUT
C
      DO 457 KUTTAA=1,4
      GO TO (458,459,458,459),KUTTAA
 459  TIME=TIME+HDT
 458  CONTINUE
      DXA2=50.0*(-XNM(2)+UINP+XNORM(1))
      DX2(1)=DXA2
      LA=1
      CALL RUNK(KUTTAA,DT,DUM1,LA,DX2,AX2)
      XNM(2)=AX2(1)
      DXEM(2)=AX2(2)
 457  CONTINUE
      KOUNT=KOUNT+1
      IF(KOUNT.LT.ITSAMP) GO TO 460
      KOUNT=0
C
C     HYBRID SYSTEM AND SAMPLING NOISE FOR TC OPERATION
      IRN=2
      CALL RNGEN (DUN,JX,1,IRN,RNU2)
      RSAMPN=RNU2(1)*DDT
      IRN=3
      CALL RNGEN(DUN,JX,1,IRN,RNGU)
      RHYBN=RNGU(1)*SIGH
      XNM(2)=XNM(2)+RHYBN  +RSAMPN
      DFT=0.
      DO 451 KUTTAD=1,4
      GO TO (452,453,452,453),KUTTAD
 453  TDIG=TDIG+DDT
 452  CONTINUE
      DHX1=10.0*(-XNM(1)+VLIN*XNM(2)+ALPHA*XNM(2)*XNM(2)*XNM(2))
      HX1=XNM(1)
      DELZ(1,1)=XNM(1)+XNORM(2)-XN1(INDEX2)
      KX=1
      CALL KALMAN (DELZ,GK,H,A,NX,KX,DXEM,DXDEM)
```

```
      IF (KFNRL.LT.3) GC TC 780
      CALL COVPD (P,A,QI,B,H,NX,MX,BQBT,AT,PD)
      DHVEC(4)=PD(1,1)
      DHVEC(5)=PD(1,2)
      DHVEC(6)=PD(2,2)
      HVEC(4)=P(1,1)
      HVEC(5)=P(1,2)
      HVEC(6)=P(2,2)
      LH=6
 780  CONTINUE
      IF(KFNRL.EQ.3) GO TO 782
      LH=3
 782  CONTINUE
      DHVEC(1)=DXDEM(1)
      HVEC(2)=DXEM(2)
      DHVEC(3)=DHX1
      HVEC(1)=DXEM(1)
      DHVEC(2)=DXDEM(2)
      HVEC(3)=HX1
      CALL RUNK(KUTTAD,HSAMP,DUM1,LH,DHVEC,HVEC)
      DXEM(1)=HVEC(1)
      DXEM(2)=HVEC(2)
      HX1=HVEC(3)
      XNM(1)=HVEC(3)
      P(1,1)=HVEC(4)
      P(1,2)=HVEC(5)
      P(2,1)=HVEC(5)
      P(2,2)=HVEC(6)
 451  CONTINUE
      XNM(2)=AX2(1)
 460  CONTINUE
C     ***********************************************
C     END OF TIME CRITICAL OPERATIONAL LOOP
C
C     ***********************************************
      ATI=(IPL-1)*IPRT+I
      T=TINIT+ATI*DT
      TIME=T
      INDEX2=INDEX2+1
C     ************************************
C     CALCULATION OF GAIN  K=PHQI
C     ************************************
      IF(KFNRL.LT.3) GO TO 448
      CALL MATMUL (P,H,NX,MX,NX,PHK)
      CALL MATMUL (PHK,QI,NX,MX,MX,GK)
      GK1(INDEX2)=GK(1,1)
      GK2(INDEX2)=GK(2,1)
 448  CONTINUE
      IF(KFNRL.EQ.3) GO TO 449
      GK(1,1)=GK1(INDEX2)
      GK(2,1)=GK2(INDEX2)
 449  CONTINUE
C****************************************************
C
```

```
      XHAT1=DXEM(1)+XN1(INDEX2)
      XHAT2=DXEM(2)+XN2(INDEX2)
   25 CONTINUE
      ERROR1=XNM(1)-XHAT1
      ERROR2=XNM(2)-XHAT2
      IF(MCR.EQ.1) GO TO 49
C     ********************************
      INDEX3=IPRT*IPL+1
      XHAT1=DXEM(1)+XN1(INDEX3)
      XHAT2=DXEM(2)+XN2(INDEX3)
      EPS1=XNM(1)-XHAT1
      TSAMP(IPL)=TIME
      RMEAN1(IPL)=RMEAN1(IPL)+XNM(1)
      RMEAN2(IPL)=RMEAN2(IPL)+XNM(1)*XNM(1)
      XHSUM1(IPL)=XHSUM1(IPL)+XHAT1
      EPSM1(IPL)=EPSM1(IPL)+EPS1
      XHSQ1(IPL)=XHSQ1(IPL)+XHAT1*XHAT1
      EPSSQ1(IPL)=EPSSQ1(IPL)+EPS1*EPS1
   49 CONTINUE
   50 CONTINUE
  100 CONTINUE
      IF(MCR.GT.1) GO TO 48
      WRITE (6,989) T,XHAT1,XHAT2,GK1(INDEX2),GK2(INDEX2)
  989 FORMAT (1H0,3X,'T='F8.6,2X,'XHAT1='E15.7,2X,'XHAT2='E15.7,2X,
     1'GK1(INDEX2)='E15.7,2X,'GK2(INDEX2)='E15.7)
      WRITE(6,990) ERROR1,ERROR2
  990 FORMAT (1H0,13X,'ERROR1='E15.8,2X,'ERROR2='E15.8)
      WRITE(6,991) XNM(1),DELZ(1,1),XNM(2)
  991 FORMAT (1H0,13X,'XNM(1)='E15.8,2X,'DELZ(1,1)='E15.8,2X,
     1'XNM(2)='E15.8)
      WRITE(6,995) DXEM(2),XNORM(1),XNORM(2),DXDEM(1),DXDEM(2)
  995 FORMAT(1H0,3X,'DXEM(2)='F15.7,2X,'XNORM(1)='F15.7,2X,2X,F15.7,2X,
     1'DXDEM(1)='F15.7,3X,F15.7)
      WRITE(6,993) XN1(INDEX2),DXEM(1),RHYBN,RSAMPN
  993 FORMAT(1H0,3X,'XN1(INDEX2)='F15.7,2X,'DXEM(1)='F15.7,2X,
     1'RHYBN='F10.6,2X,'RSAMPN='F10.6)
      WRITE (6,996) BQBT(1,1),BQBT(1,2),BQBT(2,1),BQBT(2,2)
  996 FORMAT (1H0,10X,'BQBT(1,1)='E16.8,2X,'BQBT(1,2)='E16.8,2X,
     1'BQBT(2,1)='E16.8,2X,'BQBT(2,2)='E16.8)
   48 CONTINUE
C     ********************************
      IF(MCR.EQ.1) GO TO 149
C     STATISTICS ONLY AT SAMPLE POINTS OBTAINED FROM
C     LAST VALUE IN PRINT LOOP
      WRITE (6,883)
  883 FORMAT(1H1,5X,'MONTE CARLO RESULTS FOR NONLINEAR SYSTEM RESPONSE
     3WITH EXTENDED KALMAN FILTER')
      WRITE (6,99)
   99 FORMAT (1H1,3X,'TSAMP',T22,'RX1AVG',T36,'VARX1',T50,'XHVAR1',T63,'
     1EPMEN1',T80,'EPVAR1',T93,'XHMEN1')
      XCR=MCR
      BETA=XCR/(XCR-1.0)
      DO 150 IMC=1,ICAL
      RX1AVG=RMEAN1(IMC)/XCR
```

```
      RX2AVG=RMEAN2(IMC)/(XCR-1.0)
      EPMEN1=EPSM1(IMC)/XCR
      VARX1=RX2AVG-RX1AVG*RX1AVG*BETA
      XHMEN1=XHSUM1(IMC)/XCR
      EPMSQ1=EPSSQ1(IMC)/(XCR-1.0)
      XHMSQ1=XHSQ1(IMC)/(XCR-1.0)
      XHVAR1=XHMSQ1-XHMEN1*XHMEN1*BETA
      EPVAR1=EPMSQ1-EPMEN1*EPMEN1*BETA
      WRITE (6,880) TSAMP(IMC),RX1AVG,VARX1,XHVAR1,EPMEN1,EPVAR1,XHMEN1
880   FORMAT (1H0,3X,F8.6 ,2X,E15.7,2X,E15.7,2X,E15.7,2X,E15.7,2X,E15.7,
     42X,E15.7)
  150 CONTINUE
149   CONTINUE
      STOP
      END
```

```
C
C     SYSTEM EQUATIONS, EXPRESSED IN N FIRST ORDER DIFFERENTIAL
C     EQUATIONS.  NX=NUMBER OF EQUATIONS OR THE ORDER OF SYSTEM
C
      SUBROUTINE SYSEQ(UINP,XNORM,VLIN,ALPHA,DX,X)
      DIMENSION DX(2),X(10),XNORM(2)
      DX(1)=10.0*(-X(1)+VLIN*X(2)+ALPHA*X(2)*X(2)*X(2))
      DX(2)=50.0*(-X(2)+UINP+XNORM(1))
      RETURN
      END


      SUBROUTINE AA (X,I,ALPHA,VLIN,A)
      DIMENSION X(201),A(3,3)
C     THIS ROUTINE IS USED TO REPLACE VALUES IN A MATRIX
C     FOR NONLINEAR SYSTEM
      A(1,1)=-10.0
      A(1,2)=10.0*(VLIN+3.0*ALPHA*X(I)*X(I))
      A(2,1)=0.0
      A(2,2)=-50.0
      RETURN
      END


      SUBROUTINE RANDOM (DUM,IX,XNORM)
      DIMENSION DUM(1),XNORM(2)
C     *********************************
C
C     MULTIPLICATIVE DIGITAL NUMBER GENERATOR WITH M=2 TO THE 20TH POWER
C     THIS RANDOM NUMBER GENERATOR ASSUMES A UNITY VARIANCE AND ZERO
C     MEAN. REQUIREMENTS OTHER THAN UNITY VARIANCE AND ZERO MEAN IS
C     APPLIED TO THE VARIABLE XNORM IN MAIN PROGRAM
C
C     *********************************
C
      IY=19971*IX
      IPY=IY/1048576
      IX=IY-IPY*1048576
      AX=IX
      U=AX/1048576.
      IF(U)5,5,6
    5 U=-U
    6 CONTINUE
      IX=IY
C
C     TRANSFORM TO NORMAL
C
      Z=SQRT(-2.0*ALOG(DUM(1)))
      XNORM(1)=Z*COS(6.28318*U)
      XNORM(2)=Z*SIN(6.28318*U)
      DUM(1)=U
      RETURN
      END
```

```
      SUBROUTINE RUNK (KUTTA,DT,DUM1,LX,DX,X)
      DIMENSION XA(10),DXA(10)
      DIMENSION DK(3,3)
      DIMENSION X(10)
      DIMENSION DX(10)
      HDT=.5*DT
      GO TO (10,30,50,70),KUTTA
10    DO 20 I=1,LX
      XA(I)=X(I)
      DXA(I)=DX(I)
      X(I)=X(I)+HDT*DX(I)
      DUM1=X(I)
20    CONTINUE
      RETURN
30    DO 40 I=1,LX
      DXA(I)=DXA(I)+DX(I)+DX(I)
      X(I)=XA(I)+HDT*DX(I)
40    CONTINUE
      RETURN
 50   DO 60 I=1,LX
      DXA(I)=DXA(I)+DX(I)+DX(I)
60    X(I)=XA(I)+DT*DX(I)
      RETURN
70    VDT=DT*.1666667
      DO 80 I=1,LX
80    X(I)=XA(I)+VDT*(DXA(I)+DX(I))
      RETURN
      END


      SUBROUTINE TRANSA (A,IAV,KAV,AI)
      DIMENSION A(3,3),AI(3,3)
      DO 775 JV=1,IAV
      DO 775 KV=1,KAV
      AI(KV,JV)=A(JV,KV)
775   CONTINUE
      RETURN
      END


      SUBROUTINE MATADD (A,B,M,N,C)
      DIMENSION A(3,3),B(3,3),C(3,3)
      DO 20 I=1,M
      DO 20 J=1,N
      C(I,J)=A(I,J)+B(I,J)
20    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MATSUB (A,B,M,N,C)
      DIMENSION A(3,3),B(3,3),C(3,3)
      DO 20 I=1,M
      DO 20 J=1,N
      C(I,J)=A(I,J)-B(I,J)
   20 CONTINUE
      RETURN
      END

      SUBROUTINE MATMUL (A,P,II,JJ,KK,C)
      DIMENSION A(3,3),P(3,3),C(3,3)
C     ********************************
C     II=NO. ROWS IN MATRIX A
C     JJ=NC.COLUMNS IN MATRIX P
C     KK=NO.COLUMNS IN A AND ROWS IN P
C     RESULTS MATRIX C HAS II ROWS AND JJ COLUMNS
C     ********************************
      DO 10 I=1,II
      DO 10 J=1,JJ
      C(I,J)=0.0
      DO 10 K=1,KK
      C(I,J)=C(I,J)+A(I,K)*P(K,J)
   10 CONTINUE
      RETURN
      END

      SUBROUTINE COVPD (P,A,QI,B,H,NX,MX,BQBT,AT,PD)
      DIMENSION P(3,3),A(3,3),QI(3,3),B(3,3),H(3,3),BQBT(3,3),PD(3,3)
      DIMENSION AT(3,3)
      DIMENSION PHQI(3,3),HT(3,3),HK(3,3),PN(3,3),PT1(3,3)
      DIMENSION PAT(3,3),AP(3,3),C(3,3),PH(3,3)
      CALL MATMUL (P,AT,NX,NX,NX,PAT)
      CALL MATMUL (A,P,NX,NX,NX,AP)
      CALL MATADD (AP,PAT,NX,NX,C)
      CALL MATMUL (P,H,NX,MX,NX,PH)
      CALL MATMUL (PH,QI,NX,MX,MX,PHQI)
      CALL TRANSA (H,NX,MX,HT)
      CALL MATMUL (PHQI,HT,NX,NX,MX,HK)
      CALL MATMUL (HK,P,NX,NX,NX,PN)
      CALL MATADD (C,BQBT,NX,NX,PT1)
      CALL MATSUB (PT1,PN,NX,NX,PD)
      RETURN
      END
```

```
SUBROUTINE ARTOVE (DXDM,DXDEM,PD,XNM,DXEM,P,DVSKF,VVSKF)
DIMENSION DXDM(10),DXDEM(10),PD(3,3),XNM(10),DXEM(10),P(3,3)
DIMENSION DVSKF(10),VVSKF(10)
DVSKF(1)=DXDM(1)
DVSKF(2)=DXDM(2)
DVSKF(3)=DXDEM(1)
DVSKF(4)=DXDEM(2)
DVSKF(5)=PD(1,1)
DVSKF(6)=PD(1,2)
DVSKF(7)=PD(2,2)
VVSKF(1)=XNM(1)
VVSKF(2)=XNM(2)
VVSKF(3)=DXEM(1)
VVSKF(4)=DXEM(2)
VVSKF(5)=P(1,1)
VVSKF(6)=P(1,2)
VVSKF(7)=P(2,2)
RETURN
END

SUBROUTINE VETOAR (VVSKF,XNM,DXEM,P)
DIMENSION VVSKF(10),XNM(10),DXEM(10),P(3,3)
XNM(1)=VVSKF(1)
XNM(2)=VVSKF(2)
DXEM(1)=VVSKF(3)
DXEM(2)=VVSKF(4)
P(1,1)=VVSKF(5)
P(1,2)=VVSKF(6)
P(2,1)=VVSKF(6)
P(2,2)=VVSKF(7)
RETURN
END

SUBROUTINE KALMAN (DELZ,GK,H,A,NX,MX,XHAT,XHATD)
DIMENSION HXHAT(3,3),ZINOV(3,3),CORRE(3,3),PRED(3,3)
DIMENSION DELZ(3,3),GK(3,3),H(3,3),A(3,3),XHAT(10),XHATD(10)
CALL MATMUL (H,XHAT,NX,MX,NX,HXHAT)
CALL MATSUB (DELZ,HXHAT,NX,MX,ZINOV)
CALL MATMUL (GK,ZINOV,NX,MX,NX,CORRE)
CALL MATMUL (A,XHAT,NX,MX,NX,PRED)
CALL MATADD (PRED,CORRE,NX,MX,XHATD)
RETURN
END
```

```
      SUBROUTINE RNGEN(DUM,JX,NUM,IRN,RNUM)
      DIMENSION RNUM(NUM),DUM(1)
C     **********************************
C
C     MULTIPLICATIVE DIGITAL NUMBER GENERATOR WITH M=2 TO THE 20TH POWER
C     DISTRIBUTION ARE GENERATED CONDITIONED ON INTEGER IRN
C     IRN=1, UNIFORMLY DISTRIBUTED ON THE INTERVAL (0,1)
C     IRN=2,UNIFORMLY DISTRIBUTED ON THE INTERVAL (-1,1)
C     IRN=3,NORMAL DISTRIBUTION,ZERO MEAN,UNITY VAR.
C
C     **********************************
C
      DO 7 I=1,NUM
      IY=1366853*JX
      JX=IY
      IF(IY.LT.0)IY=IY+2147483647+1
      U=IY*.4656613E-9
C
C     TRANSFORM TC NORMAL
C
      Z=SQRT(-2.0*ALOG(DUM(1)))
      XNORM=Z*COS(6.28318*U)
      DUM(1)=U
      GC TO (1,2,3),IRN
    1 RNUM(I)=U
      GO TO 10
    2 RNUM(I)=2.0*U-1.0
      GO TO 10
    3 RNUM(I)=XNORM
   10 CONTINUE
    7 CONTINUE
      RETURN
      END
```

VITA

Willard Morris Holmes

Candidate for the Degree of

Master of Science

Thesis: ON-LINE HYBRID COMPUTER IMPLEMENTATION OF STOCHASTIC FILTERING ALGORITHMS

Major Field: Electrical Engineering

Biographical:

Personal Data: Born in Butler County, Alabama, February 1, 1936.

Education: Graduated from Crenshaw County High School, Highland Home, Alabama, in May 1954. Received Bachelor of Electrical Engineering from Auburn University, Auburn, Alabama, in June 1960. Completed the requirements for Master of Science Degree in July 1973.

Professional Experience: 1960-1965: Employed with aerospace contractors, Huntsville, Alabama, to perform design and development of guidance and control of Saturn V aerospace vehicle.

1965-1972: U.S. Army Millile Command, Redstone Arsenal, Research and Development Laboratories, Huntsville, Alabama. Performed design and development of air defense missile systems using analytical and computer simulation techniques.

1972-1973: On leave to attend Oklahoma State University, Stillwater, Oklahoma.

Professional Organizations: Registered Professional Engineer, State of Alabama, Member of the Institute of Electrical and Electronics Engineers, Member of Simulation Councils, Inc.