DIGITAL SPEED AND ACCURACY TRADEOFFS IN CONTINUOUS KALMAN FILTERING FOR NONLINEAR SYSTEMS

By

JERRY ALTON GASTON

Bachelor of Science

Oklahoma State University

Stillwater, Oklahoma

1971

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
December, 1973

m

DIGITAL SPEED AND ACCURACY TRADEOFFS IN

CONTINUOUS KALMAN FILTERING

FOR NONLINEAR SYSTEMS

Thesis Approved:

James R. Rowland
Thesis Adviser

Bennett Basore

Rao Yarlagadda

D. D. Durham
Dean of the Graduate College

PREFACE

I would like to express my sincere appreciation to my thesis adviser and committee chairman, Dr. James R. Rowland, for his extensive guidance and encouragement throughout this study. I would also like to thank the other committee members, Dr. Bennett Basore and Dr. Rao Yarlagadda, for their comments on the work.

Mrs. Jaquita Icenhower deserves special thanks for typing this thesis, and a deep appreciation goes to my parents for their support and encouragement. In conclusion, I express gratitude to my wife for her understanding and help in preparing the rough draft.

TABLE OF CONTENTS

n

LIST OF TABLES

Table | Page

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

Due to the rapid expansion in recent years of such diverse areas as navigation and orbit determination, communication and control theory, and operations research, the need has arisen for increased sophistication in dealing with problems caused by random noise in nonlinear dynamical systems. The digital computer has become an invaluable tool in the simulation of such nonlinear stochastic systems and their associated optimal filtering algorithms. The filtering problem is to determine the optimal estimate of the state variables of continuous-time nonlinear dynamical systems from continuous noisy output observations. The digital computer, when used in the simulation of such systems corrupted by random input and measurement disturbances, introduces new discretization problems in obtaining an accurate, yet computationally fast, implementation of the particular filter under consideration. The proper choice of sampling rate, integration scheme, and number of on-line computations should result in an improved performance of the stochastic system under consideration. Since exact solutions to nonlinear filtering problems lead to infinite-dimensional computational algorithms, approximate filters become necessary in most cases. Although complicated approximate nonlinear filters are available, extensions of the basic Kalman filter are used most often in practice. The proper choice of the approximate filter for the particular system under consideration and the corresponding number of operations to be performed on-line become important when operations are to be implemented on the digital computer.

The objective of this research was to investigate tradeoff possibilities for the digital implementation of two versions of the continuous Kalman filter for nonlinear dynamical systems. A detailed examination of tradeoffs between complexity, accuracy, and computational speed was made. The two particular filtering algorithms considered were the variational Kalman filter and the extended Kalman filter. An examination of these two algorithms pointed out distinct differences as integration techniques, step sizes, and system nonlinearities were varied.

The approach to the problem was to develop a general digital computer program to implement the variational Kalman filter and the extended Kalman filter with provisions for handling changes in integration techniques and step sizes. Since the structural differences between the two algorithms are small, the programming effort may be accomplished with a single general computer program. This research was performed to provide some insight into the problem of which filter to use for a given application, which step size would likely result in the most efficient treatment of the system being considered, and how severe a plant nonlinearity each method would handle effectively.

Background

The emphasis has shifted in the past two decades from the design of filters in the frequency domain, e.g. Wiener filtering, to time-domain design, e.g. Kalman filtering. For linear, time-varying systems with gaussian noise disturbances, the optimal filter in the mean-square error sense is the Kalman filter. There are a great number and variety of derivations of Kalman filtering algorithms to be found in the literature which are very useful in new applications in the area of stochastic control systems. The early work of Kalman (1) and Kalman and Bucy (2) dealt with a linear unbiased minimum-error-variance algorithm for state estimation using an "orthogonal projection lemma". More recent methods for developing the continuous linear algorithm include a direct derivation from the Wiener-Hopf integral equation (3), a direct application of the matrix minimum principle to minimize the variance of the estimation error (4), and considerations of the continuous problem as the limiting case of the discrete problem as the sampling interval is reduced to zero (5,6). Sims and Melsa (7) considered the use of fixed-configurations for supoptimal linear estimation.

Certain applications, such as frequency and phase modulation, having inherently nonlinear observations models, and the nonlinear dynamic message models for many vehicle guidance and control problems are treated by extending the Kalman filter for approximate filtering in nonlinear systems. These problems may be treated by linearizing about a nominal trajectory, by the invariant imbedding sequential estimation procedure introduced by Bellman (8), or by conditional mean estimation for nonlinear systems (9,10). These minimum-error-variance filtering algorithms represent a form of "linearized" Kalman filter for the nonlinear case (11), whereas maximum a posteriori estimation, derived from nonlinear two-point boundary-value problems, uses "running-time" invariant imbedding (12). The linearized Kalman filter (13), a heuristic extension of Kalman and Bucy's earlier work (2), is a minimum-variance filter based on linearization about a nominal trajectory. If one assumes the conditional-mean estimate is known and the message and observation models are expanded about the conditional-mean estimate, the resulting first-order conditional-mean filter is called the extended Kalman filter (5,14). A limited amount of research comparing these filters can be found in the literature (15,16).

Since data representing continuous random processes must often, as in certain Kalman filtering applications, be processed and analyzed in terms of discrete sample values (17), some continuous-to-discrete conversion errors are introduced. If the sampling update rate for filtering is considered to be constant, then the problem remaining is that of choosing an appropriate sampling interval. Sampling at points too close together results in an unnecessarily large cost of calculations. On the other hand, sampling intervals too large lead to aliasing problems, which are potential sources of error. The question becomes one of choosing the Nyquist frequency as a lower limit of the sampling rate. The choice will lead to a suitable compromise between accuracy and speed to yield an efficient but accurate Kalman filtering algorithm.

Because of the rather important role played by the modern computer in the analysis, design, development, and control of a wide variety of systems, it is appropriate to consider the numerical techniques that have been developed for system simulation. In performing the basic operations involved in simulating a continuous system on a digital computer, certain approximations which introduce various types of errors are inherent.

One of the most important areas of concern in the simulation of differential equations on the digital computer is the choice of numerical integration techniques. The extension of the State Transition Method (18) to certain nonlinear systems by means of a quasi-linear

n

approach was considered by Giese (19). Also, Wait (20) introduced a generalization of the state-space method using piecewise linear approximations to handle nonlinear systems by using interpolations and extrapolations of system inputs. The similarity of the general state transition method of Liou (21) to the conventional fourth-order Runge-Kutta method (RK4) was pointed out by Mastascusa (22). Benyon (23) investigated the proper selection of integration formulas based on computing speed and accuracy for the simulation of guided missiles and similar systems. He compared predictor, predictor-corrector, and Runge-Kutta methods of different orders, both theoretically and experimentally. Rowland and Holmes (24) investigated digital integration techniques for nonlinear dynamical systems by introducing a variational approach and making certain numerical approximations which yielded improvements in both accuracy and execution time. The approximations involved in their method were shown to be effective for mildly nonlinear systems, and an improvement over RK4 in both speed and accuracy in certain cases was demonstrated for the variational method developed.

Several comparisons between Runge-Kutta (25) and predictor-corrector formulas (26) have been made. Both Hildebrand (27) and Kopal (28) treated Runge-Kutta (RK) methods and their derivation and use extensively. While the step sizes in RK methods are easy to change because of their self-starting feature, error estimates are not readily available. On the other hand, predictor-corrector methods yield error estimates readily, but must employ another means of starting, such as RK formulas. Merson (29) has devised a modified Runge Kutta method from which error estimates may be made by increasing derivative evaluations. Rice (30) and Forrington (31) have described the limited use of mixed step-length "split Runge-Kutta" and "split predictor-corrector" methods, respectively. Two major sources of error introduced in the digital solution of differential equations are truncation errors and roundoff errors. Blum (32) and Gill (33) devised schemes for reducing the effects of roundoff errors. A method developed to give an estimate of truncation error was described by Lance (34) as the Runge-Kutta-Merson (RKM) method, a modification of the basic fourth-order Runge-Kutta method having a slightly smaller truncation error. While requiring five iterations per step, RKM allows one to maximize the step while the truncation error is maintained within specified limits.

It can readily be observed that much work has been done in estimating the state of noise-corrupted nonlinear systems, resulting in several very effective algorithms for state estimation in nonlinear system. There have been extensive investigations of a large variety of numerical integration techniques designed to handle different types of systems. However, an investigation is needed to determine which integration technique or combination of techniques would be most effective to implement a particular filtering algorithm.

Kalman Filtering For Nonlinear Systems

Consider the nth-order class of nonlinear dynamical systems driven by white noise with noise-corrupted observations defined by

$$\dot{\underline{x}}(t) = \underline{f}\,[\underline{x}(t),t] + B(t)\underline{w}(t) \tag{1.1}$$

$$\underline{z}(t) = \underline{h}\,[\underline{x}(t),t] + \underline{v}(t) \tag{1.2}$$

where $\underline{w}(t)$ is an r-vector of zero-mean white Gaussian noise processes with a covariance matrix

$$\text{cov}\,\{\underline{w}(t),\underline{w}(\tau)\} = Q(t)\,\delta_D(t-\tau) \tag{1.3}$$

where $\delta_D(\cdot)$ is the Dirac delta function. The observation is an m-vector corrupted by additive, zero-mean, white Gaussian noise with a covariance matrix

$$\text{cov}\,\{\underline{v}(t),\ \underline{v}(\tau)\} = R(t)\ \delta_D\ (t-\tau) \tag{1.4}$$

Both Q(t) and R(t) are symmetric, positive-definite matrices. It is assumed that $\underline{w}(t)$, $\underline{v}(t)$ and $\underline{x}(t_0)$ are all uncorrelated.

Given the set of data $\underline{z}(t)$, the problem is to apply the variational and extended versions of the Kalman filter to determine an estimate of the state of the system.

The variational Kalman filter is based on incremental linearization about a nominal trajectory. Both $\underline{f}[\underline{x}(t),t]$ and $\underline{h}[\underline{x}(t),t]$ are expanded in a Taylor series about the nominal deterministic trajectory given by

$$\dot{\underline{x}}_N(t) = \underline{f}[\underline{x}_N(t),t] \tag{1.5}$$

For the variational Kalman filter, A(t) and C(t) are given by

$$\begin{aligned} A(t) &\triangleq \left.\frac{\partial \underline{f}[\underline{x}(t),t]}{\partial \underline{x}(t)}\right|_{\underline{x}(t) = \underline{x}_N(t)} \\ C(t) &\triangleq \left.\frac{\partial \underline{h}[\underline{x}(t),t]}{\partial \underline{x}(t)}\right|_{\underline{x}(t) = \underline{x}_N(t)} \end{aligned} \tag{1.6}$$

The resulting linearized message and observation models are

$$\underline{\delta \dot{x}}(t) = A(t)\ \underline{\delta x} + B(t)\underline{w}(t) \tag{1.7}$$

$$\underline{\delta \dot{z}}(t) = C(t)\ \underline{\delta x} + \underline{v}(t) \tag{1.8}$$

where $\underline{\delta x}$ and $\underline{\delta z}(t)$ are the deviations from the nominal trajectory and nominal observation, respectively. If these deviations are small, then the higher order terms of the Taylor series expansion may be neglected to yield (1.6) - (1.7). Therefore, the variational Kalman filtering algorithm may be written as

$$\dot{\delta\hat{\underline{x}}}(t) = A(t)\ \delta\hat{\underline{x}}(t) + K(t)\ (\delta\underline{z}(t) - B(t)\ \delta\hat{\underline{x}}(t)) \tag{1.9}$$

$$K(t) = P(t)\ C^T(t)R^{-1}(t) \tag{1.10}$$

where the error variance equation is given as

$$\begin{aligned}\dot{P}(t) = A(t)P(t) + P(t)A^T(t) - P(t)C^T(t)R^{-1}(t)C(t)P(t) \\ + B(t)Q(t)B(t)\end{aligned} \tag{1.11}$$

The total linearized state estimate is

$$\hat{\underline{x}}(t) = \underline{x}_N(t) + \delta\hat{\underline{x}}(t) \tag{1.12}$$

The extended Kalman filter may be obtained by assuming that the conditional-mean estimate $\hat{\underline{x}}(t)$ is known and used to expand the message and observation models in a Taylor series. The linearized coefficient matrices A(t) and C(t) for the extended Kalman filter are

$$\begin{aligned} A(t) &\triangleq \left.\frac{\partial \underline{f}[\underline{x}(t),t]}{\partial \underline{x}(t)}\right|_{\underline{x}(t) = \hat{\underline{x}}(t)} \\ C(t) &\triangleq \left.\frac{\partial \underline{h}[\underline{x}(t),t]}{\partial \underline{x}(t)}\right|_{\underline{x}(t) = \hat{\underline{x}}(t)} \end{aligned} \tag{1.13}$$

For the linear case, it is clear that both of these algorithms reduce to the linear Kalman filtering equations. It should be pointed out that the time-varying gain K(t) may be precomputed for the variational filter algorithm. This fact becomes especially important in cases where on-line computational time is critical.

Approach To The Problem

The initial effort was to develop a computer program that would accurately simulate the nonlinear system considered, using an accurate integration method for all investigations. Thereafter, a second portion of the program would implement each filter by using one of the several integration methods under investigation with a provision for adjusting the step size for speed and accuracy tradeoff studies. A comparison of the accuracies of both filters is desirable as a basis of comparison for accuracy versus computational speed and later for several integration methods. Since no comparisons have been made in the literature for stochastic filters when implemented by different integration methods, the results of such a comparison will provide useful information in cases where an optimal choice of both the type of filter and the method of integration is desirable for time-critical computations. Some insight into the effects of system nonlinearities and input noise on filter performance might be derived from the results in Chapters II and III.

The curves of Chapters II and III will illustrate at what point it becomes advantageous to use the more computationally complex extended filter over the variational filter for a given set of operating conditions. Chapter IV will provide an idea as to how computationally complex an integration algorithm to use for given accuracy or computational speed requirements. All the above points of interest will provide valuable information about the variational Kalman filter or the extended Kalman filter when computational accuracy and speed are of importance.

Thesis Outline

Following this introduction to the problem, a brief description of the digital computer program used to implement the two filtering algorithms, along with preliminary simulation results comparing accuracies, is presented in Chapter II. Chapter III sets forth considerations of accuracy versus computational speed as the integration step size is adjusted to represent data rates consistent with the amount of on-line calculations performed. Tradeoffs between accuracy and computational speed are investigated with respect to several different numerical integration techniques in Chapter IV. Conclusions and recommendations for further research are presented in Chapter V.

CHAPTER II

PRELIMINARY DEVELOPMENT AND RESULTS

The first step in the investigation of the implementation of Kalman filtering algorithms for nonlinear applications is the development of an appropriate digital computer program. Only the two continuous versions of the Kalman filters under examination were implemented. Because the methods of simulating the system, filter algorithm, and integration schemes are of great importance in such a study, a program description will be provided along with preliminary results of accuracy comparisons for the variational and extended Kalman filters.

Digital Computer Program Description

The initial part of this research effort was necessarily the development of a multipurpose digital simulation program. To obtain the flexibility necessary to examine the effectiveness of both the variational and the extended Kalman filters, a flexible computer program was implemented by using several generalized subroutines. A comprehensive listing of the Fortran program is given in the appendix.

Since the structural differences between the two algorithms are small, a single program was used to implement both filters with only minor changes. The simulation program was capable of implementing the variational filter (KALF = VAR) or the extended filter (KALF = EXT) for any nonlinear system by inserting system equations in Subroutine XEQN by inserting system equations in state variable form into Subroutine XEQN. Partial derivatives of $\underline{f}[\underline{x}(t),t]$ with respect to each state variable must be calculated and then defined as the elements of A(t) in Subroutine JAC. Subroutines RUNGK and RK4 are both fourth-order Runge-Kutta integration subroutines and Subroutines RK2 and RK2P are both second-order Runge-Kutta integration subroutines. The integration of filter equations by the second-order Adams-Bashforth formula may be accomplished with Subroutines AB2 and AB2P. Subroutine RNG generates two series of random numbers which become the input and measurement noises. Subroutine DELXH and Subroutine RSL are used to calculate the filter equations for $\delta\dot{\hat{\underline{x}}}(t)$ and $\dot{\underline{P}}(t)$, respectively. The flow chart shown in Figure 1 depicts the basic features of the program operations for both filters. It illustrates that the program may generate a single sample function (MCR = 1) or any number of Monte Carlo runs (MCR > 1), where the sample mean and sample variance of $\underline{XE}(t)$ are both calculated. $\underline{XE}(t)$, the error in the estimate, is determined from the equation

$$\underline{XE}(t) = \underline{x}(t) - \hat{\underline{x}}(t) \tag{2.1}$$

The sample mean and sample variance for the first state of the system are defined as

$$X1BAR = \frac{\sum_{M1=1}^{MCR} (XE_1(t))_{M1}}{MCR} \tag{2.2}$$

$$VARX1 = \frac{\sum_{M1=1}^{MCR} (XE_1(t) - X1BAR)_{M1}{}^2}{MCR} \tag{2.3}$$

Corresponding relationships, X2BAR and VARX2, are defined for $x_2(t)$. The average RMS error RMSER , which is used extensively in this chapter, is computed in the program from the equation

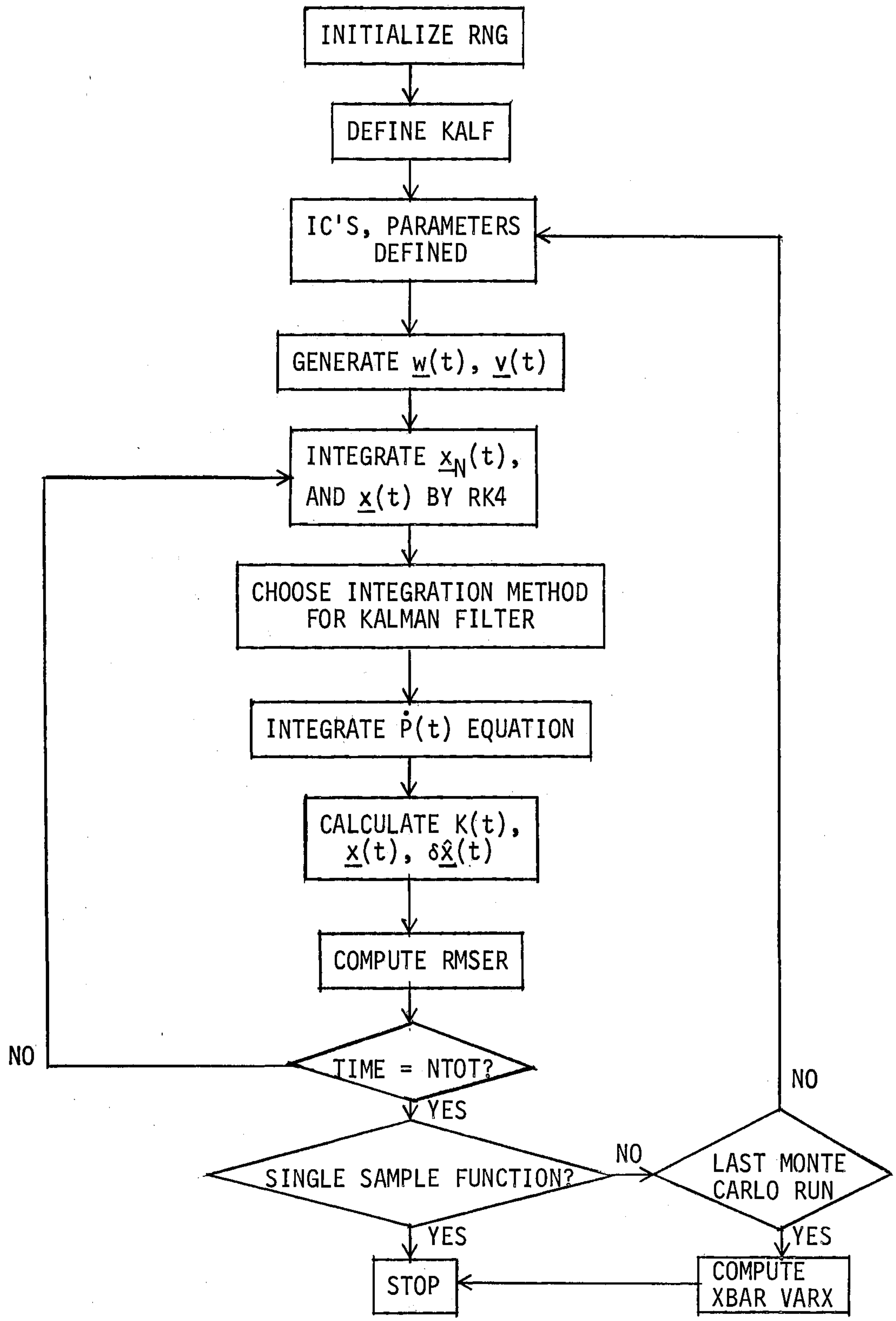

Figure 1. Flow chart for Generalized Computer Program

$$\text{RMSER} = \sqrt{\sum_{LN=1}^{NTOT} \left[x_1(LN) - \hat{x}_1(LN)\right]^2 / NTOT} \tag{2.4}$$

for a total number of samples NTOT.

The nonlinear system used in the computer program for all comparisons made is given by

$$\begin{aligned} \dot{x}_1(t) &= -2\,x_1(t) + \alpha x_2^3(t) \\ \dot{x}_2(t) &= -x_2(t) + w(t) \end{aligned} \tag{2.5}$$

This system was used by Rowland and Holmes (35) for error propagation studies and was found to be acceptable as a nonlinear reference system. Figure 2 shows a second-order nonlinear circuit described mathematically by

$$\begin{aligned} \dot{v}_o &= -\frac{1}{R_2\,C_2}\,v_o + \frac{K}{R_2\,C_2}\,v_1{}^3 \\ \dot{v}_1 &= -\frac{1}{R_1\,C_1}\,v_1 + \frac{1}{R_1\,C_1}\,v_s \end{aligned} \tag{2.6}$$

where $R_1C_1 = 1$, $R_2C_2 = \frac{1}{2}$, and the source $v_s(t)$, applied for all $t \geq 0$, is a zero-mean Gaussian white noise process with variance Q. Identifying v_o as x_1 and v_1 as x_2 with the given parameters yields (2.5) directly from (2.6) and the nonlinear circuit of Figure 2. The observation model in all cases considered was given by

$$z(t) = x_1(t) + v(t) \tag{2.7}$$

where the variance of the zero-mean white noise process v(t) was 0.1 for all time. A block diagram of the system in (2.5) with its associated Kalman filtering algorithms is shown in Figure 3.

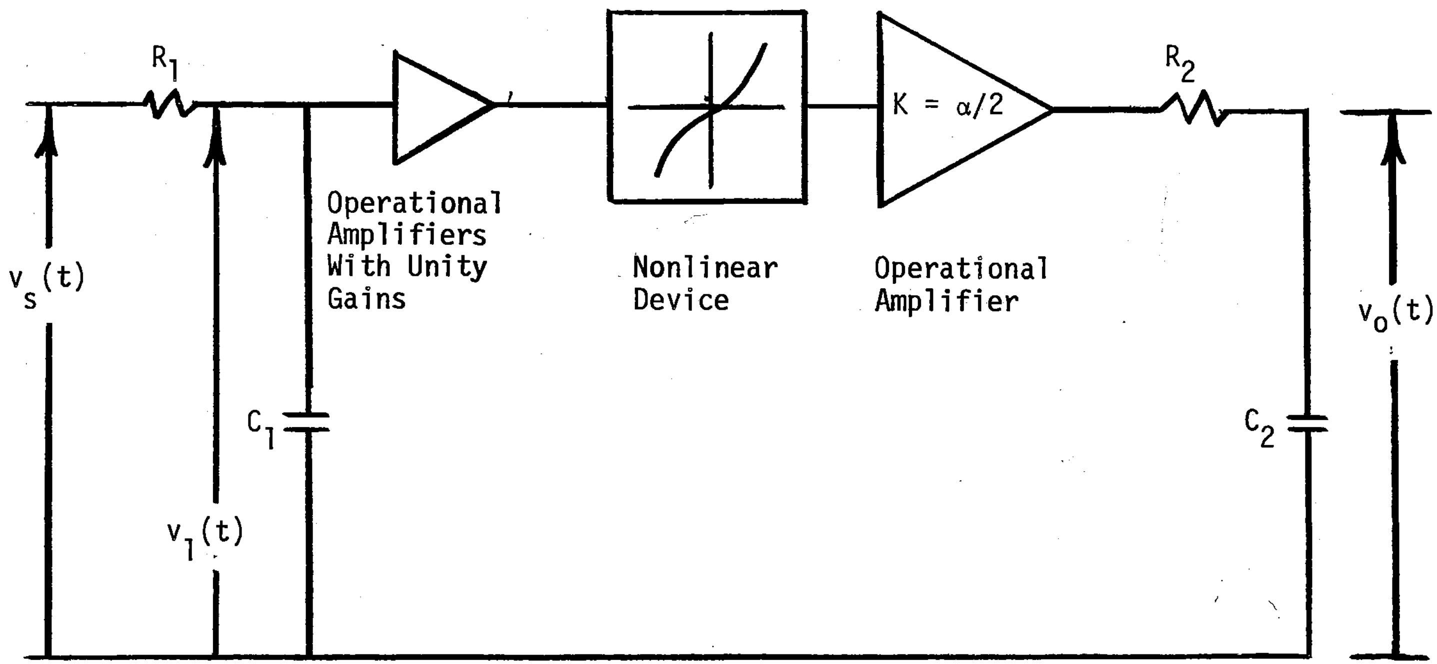

Figure 2. Nonlinear Circuit Described by Equation (2.6)

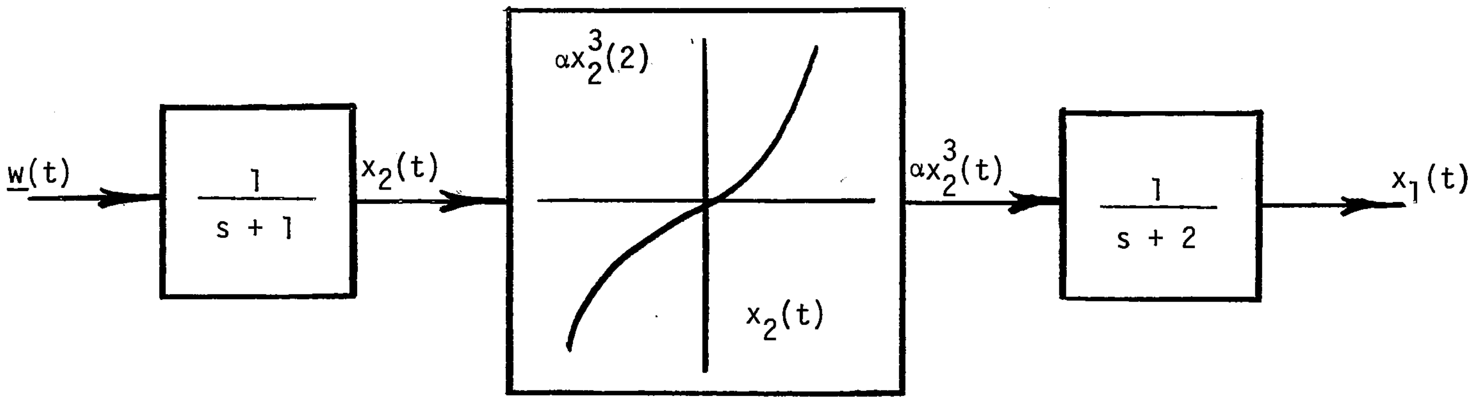

Figure 3. Block Diagram of the Second-Order Nonlinear System Described by Equation (2.5)

Numerical Comparisons Based on Accuracy

For accuracy comparisons of the two algorithms, a standard fourth-order Runge-Kutta integration method (RK4) was used. The data rate was held constant for accuracy comparisons, in this chapter and computational speed comparisons are described in Chapter III. Parameters under consideration were the magnitude of the system nonlinearity (α), the order of the nonlinearity (AEXP), and the variance of the input noise (Q). Because of a more accurate reference trajectory, i.e. the new estimate of the state as that estimate became available, the linearity assumptions were more accurate for the extended Kalman filter than for the variational filter. Due to this relinearization, large initial estimation errors were not allowed to propagate in time. Figures 4 and 5 show sample functions of sample error variance for x_1, $\hat{x}_1(t)$, $x_1(t)\big|_{var}$ and $x_1(t)\big|_{ext}$ versus time for the conditions of α=0.5, Q = 10, and AXEP = 3. Because of the better accuracy of the extended Kalman filter and the fact that no penalty is associated in this chapter for the additional on-line operations, Figures 6, 7, and 8 indicate an improved performance for the extended Kalman filter in the average RMS error defined by (2.4).

It should be noted that Figure 4 represents single sample functions and should be viewed as typical behavior for the state estimate of the system. More valid plots of estimate errors would show Monte Carlo averages for many sample functions. Figure 5 shows error variance results obtained by using twenty-five Monte Carlo runs with a constant step size of 0.05 for both filters. These results point up an area to be considered in Chapter III, i.e. the adjusting of the step size of

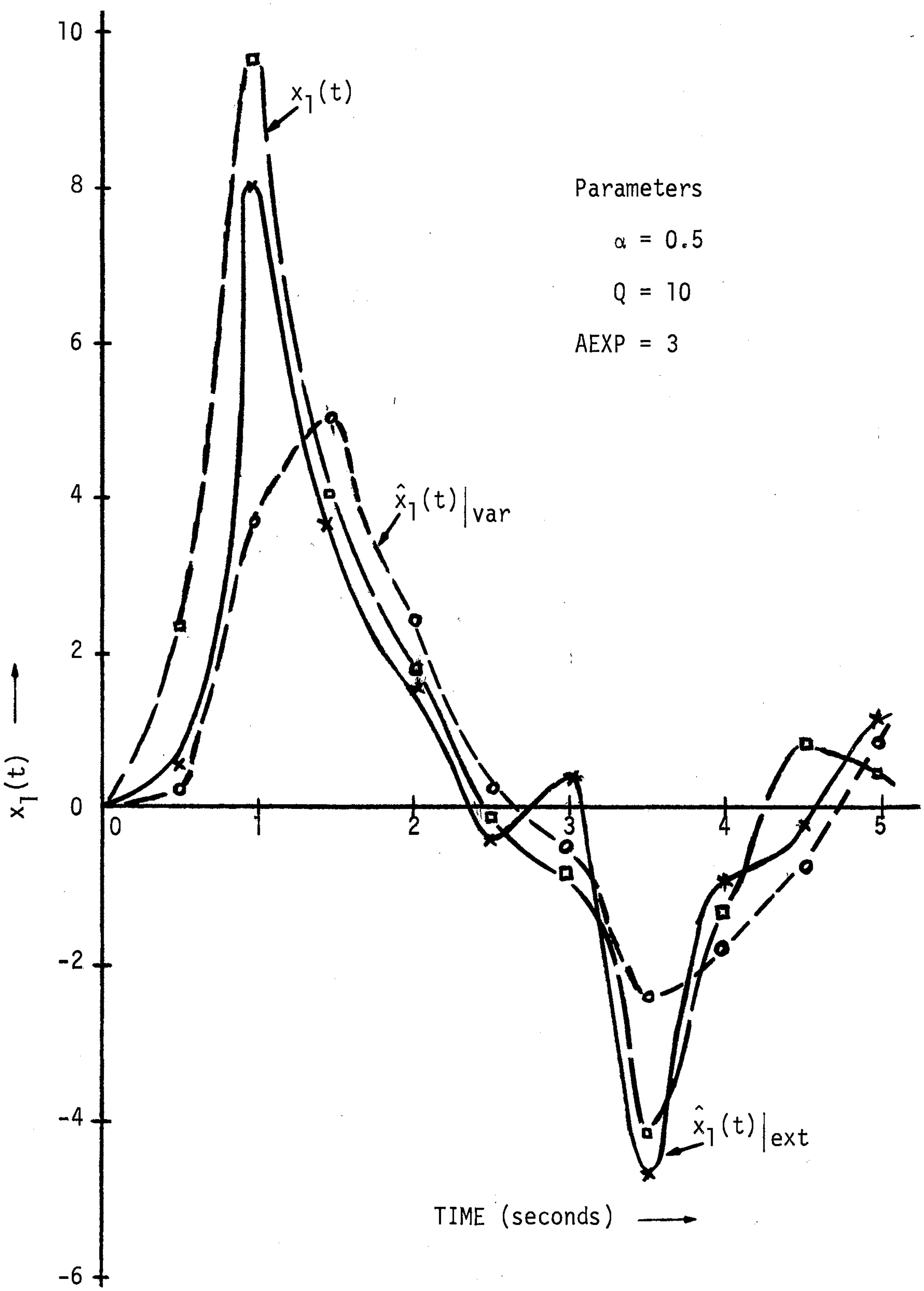

Figure 4. The Variable $x_1(t)$ and Estimates of $x_1(t)$ Versus Time

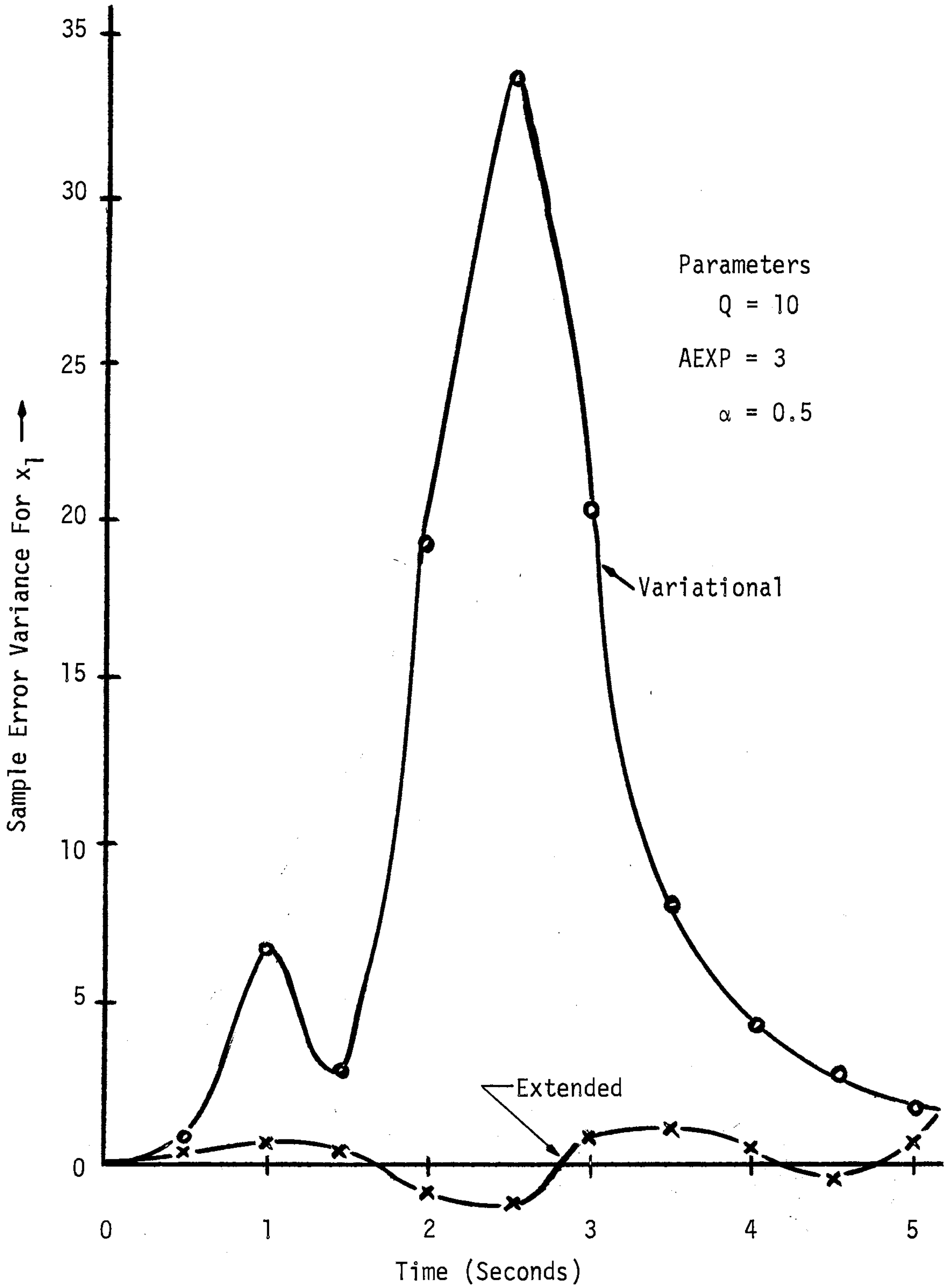

Figure 5. Plots of Sample Error Variance for x_1 Versus Time

both filters or the penalizing of the extended filter for having more on-line operations per step than the variational filter, whose filter gain and error variance may be precomputed.

Tradeoff Possibilities

It is clear from Figures 6, 7, and 8 that for all cases considered, the unpenalized extended Kalman filter performed better than the variational Kalman filter. It should be noted, however, that there exists a penalty of increased computational time for the extended filter and that for smaller values of input noise and nonlinearities the variational filter could possibly be used for increased speed. It can also be seen from Figures 6, 7, and 8 that as conditions become more harsh, the error in the estimate of $x_1(t)$ for the variational filter becomes much more pronounced when compared to that of the extended filter. It is conjectured that there exists a certain operating point beyond which the extended Kalman filter should be used as opposed to the variational Kalman filter. This point may be determined by the degree of accuracy required for the particular case being considered here. It is apparent that for some given set of parameters with the step sizes adjusted to give comparable performances in computational speed for both filters, the extended Kalman filter should be used for its greater accuracy in dealing with harsher nonlinear conditions. These tradeoffs will be considered further in the following chapter.

Summary

Comparisons of the two filters based entirely upon accuracy have been made in this chapter for a particular second-order nonlinear

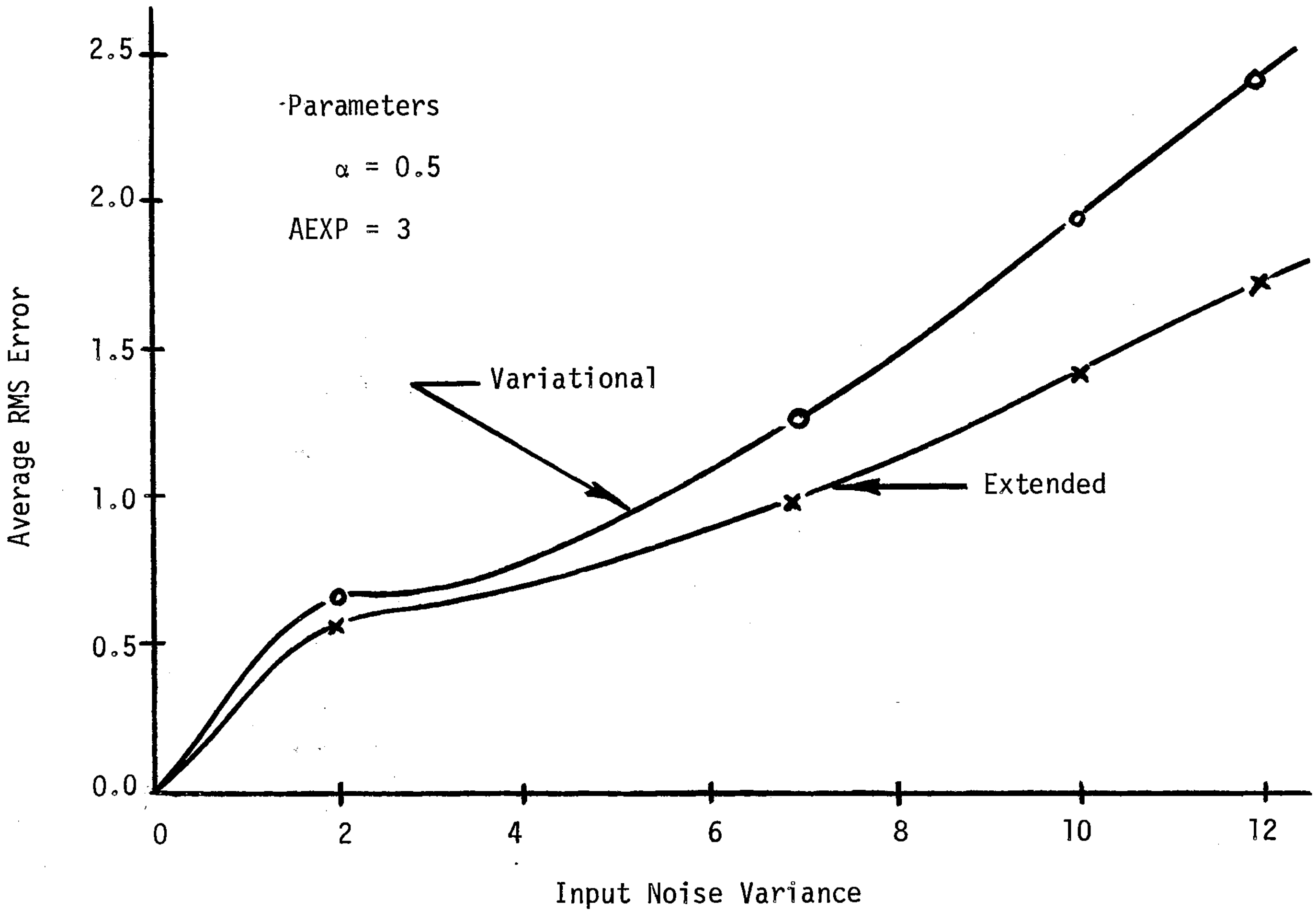

Figure 6. Average RMS Error Versus Variance of Input Noise (Q)

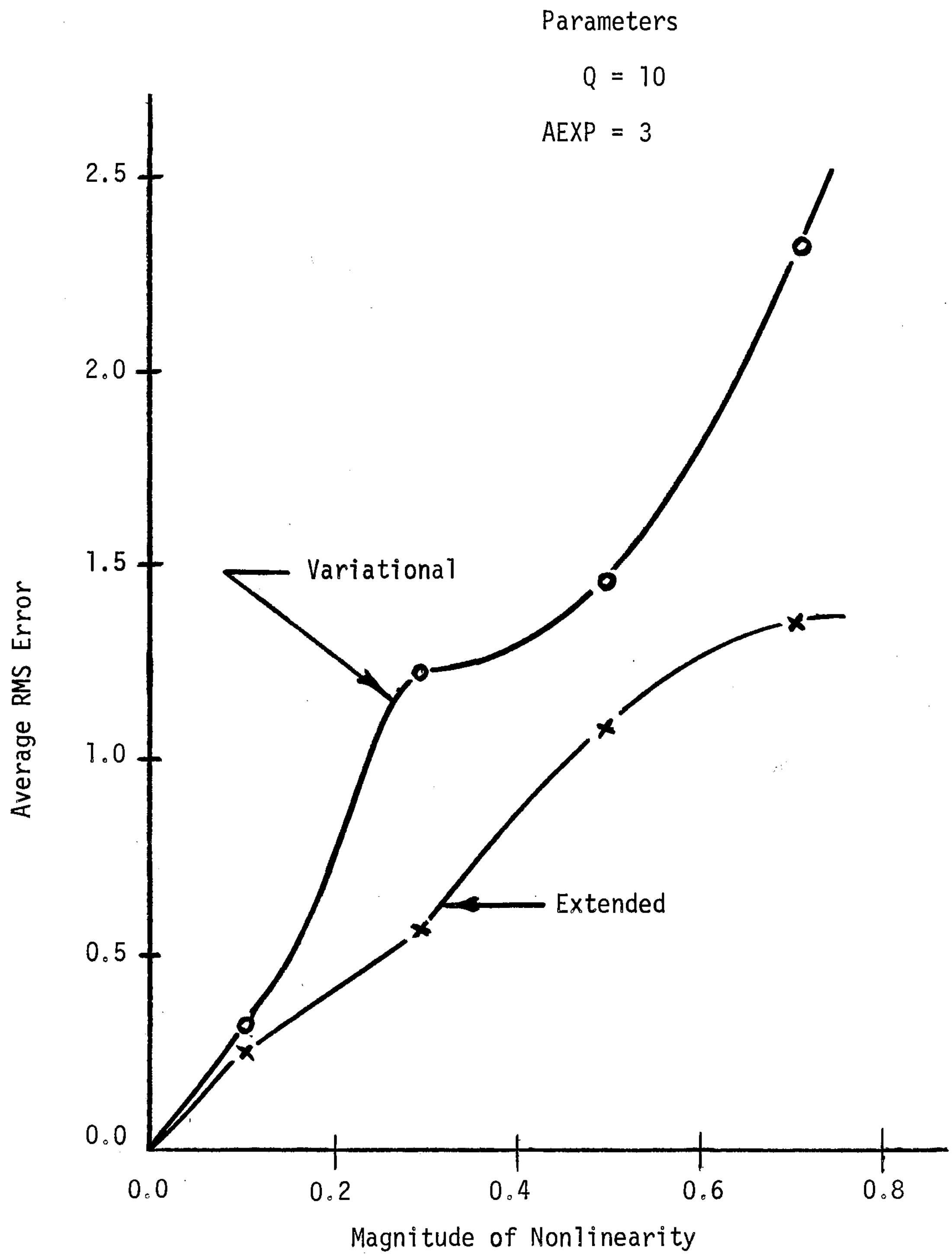

Figure 7. Average RMS Error Versus Magnitude of Nonlinearity

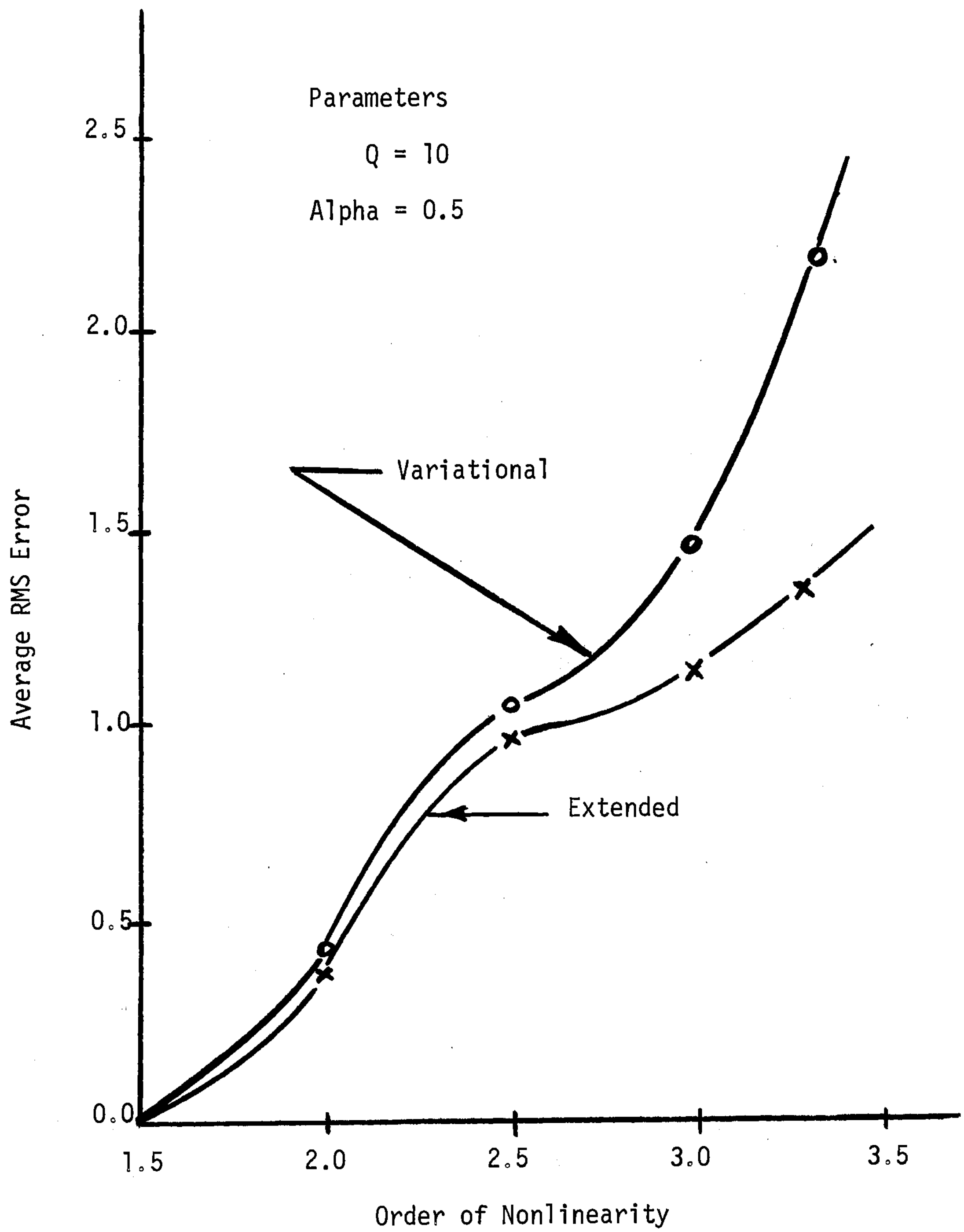

Figure 8. Average RMS Error Versus Order of Nonlinearity

system. The standard RK4 integration formula was used throughout with a nominal step size of 0.05 seconds. The error in the estimates of the state $x_1(t)$ was examined in the average RMS sense. It was shown that, for the particular system considered, the higher the input noise and the greater the nonlinearity, the more pronounced was the advantage of the extended Kalman filter in computational accuracy. The need for including a penalty for decreased computational speed and a subsequent accuracy versus speed study was introduced for consideration in the next chapter.

CHAPTER III

COMPUTATIONAL SPEED AND FILTERING PERFORMANCE

To make a fair comparison of the extended Kalman filter and the variational Kalman filter on the basis of computational speed and accuracy, a suitable adjustment was made in the step size at which the more computationally complex algorithm operated. Once this was done, it was possible to outline the conditions under which it was advantageous to utilize each of the two filtering algorithms. The important system characteristics which comprise these conditions are input noise, order of nonlinearity, and magnitude of nonlinearity. The RMS error in the estimate of $x_1(t)$ was again used as a measure of filtering performance as these parameters were varied for both filters. These comparisons form a basis for the consideration in Chapter IV of several single-step and multi-step integration methods when used to implement both filters.

Number of Computer Operations

When totaling the number of on-line operations for each filter, some relationship between a multiplication and an addition must either be calculated for the particular computer being used or assumed in the more general case. Benyon's assumption (23) that a multiplication takes approximately twice as long as an addition was used in this chapter. It was also assumed that those operations that could be computed off-line were precomputed and stored for subsequent use. Those quantities which varied from interval to interval, but were held constant over a particular interval, were computed only once per interval and then stored for later use in the same interval. The integration of $\dot{P}(t)$ and $\delta\dot{\hat{x}}(t)$ by RK4 and the calculation of K(t) by multiplying P(t) by a precomputed constant made up the operations from which the total number of weighted on-line operations was determined for the extended Kalman filter. This number was reduced substantially for the variational Kalman filter, which required that only $\delta\dot{\hat{x}}(t)$ be integrated on-line.

It was determined that for an nth order system with r-dimensional input noise and m-dimensional observation observation noise, the number of multiplications involved in Equation (1.10) is n^2m and the number of additions is $(n-1)nm$ for the above assumptions. Equation (1.11) requires $[n(n + 1)(3n + r)/2 + n(r^2 + n) + n(n + 1)/2]$ multiplications and $[n(n + 1)(3n + r-4)/2 + nr^2 - nr + n^3 - n^2 + 9 + n(n +1)/2]$ additions. Finally, it is noted that Equation (1.9) involves $[n^2(m + 2) + n]$ multiplications and $n(n^2 + 2)$ additions.

Numerical Results

Numerical comparisons showing the number of on-line operations required for the variational and extended Kalman filters for a second-order system of the form given in Figure 3 are provided in Table 1. These number of operations were computed by assuming that the RK4 integration formula was to be used. While the elements of the A(t) matrix are a function of the state estimates for the extended filter, the number of operations cannot easily be found for the general case.

Taking into account the zeroes in the matrices A(t), H(t), and B(t), for the second-order system of Equation (2.5), the weighted total number of operations for the extended Kalman filter is 292. This corresponding number is 116 for the variational Kalman filter. The ratio of the number of weighted operations performed on-line for the extended filter to the number of weighted operations performed on-line for the variational filter is then 2.52. Thus, for the curves of Figures 9, 10, and 11 a step size of HF = 0.05 seconds was used for the variational filter and a step size 2.5 times that number, i.e. HF = 0.125 seconds, was used for the extended filter. As in Chapter II, the system equations were integrated at the smaller step size of HH = 0.025 seconds to more closely approximate the realtime system. In addition, all the curves of this chapter are for the same sequence of random numbers, generated every 0.025 seconds. Moreover, each curve represents a set of single sample functions. The system equations were integrated out to five seconds and the RMS error determined from an average of twenty values computed every 0.25 seconds. Figure 9 illustrates the effect of input noise on filter performance for a single sample function. It appears that for the system of Figure 3 and input noise $Q < 8$, the variational Kalman filter consistently out-performed the extended filter. For $Q > 8$ the added complexity of the extended filter was beneficial in spite of the penalty of a larger step size. Figures 10 and 11 support the results of Figure 9 in that the variational filter performance is superior to that of the extended filter for mild conditions, i.e. small magnitude of nonlinearity and small order of nonlinearity. Figures 10 and 11 also identify conditions for more harsh nonlinearities, where the extended filter increasingly outperformed the variational filter.

TABLE I

A COMPARISON OF THE NUMBER OF ON-LINE OPERATIONS REQUIRED FOR THE TWO FILTERS

	General Case for n = 2, r = 1, m = 1		Special Case with Zero Elements Included for n = 2, r = 1, m = 1	
	Mult.	Add	Mult.	Add
Integration	30	27	13	12
Integration	14	12	10	9
Calculation of K	4	2	2	0
Totals for the Variational Kalman Filter	14	12	10	9
Totals for the Extended Kalman Filter	48	41	25	16
Totals for the Variational Kalman Filter Using RK4	56	48	40	36
Totals for the Extended Kalman Filter Using RK4	180	158	94	84

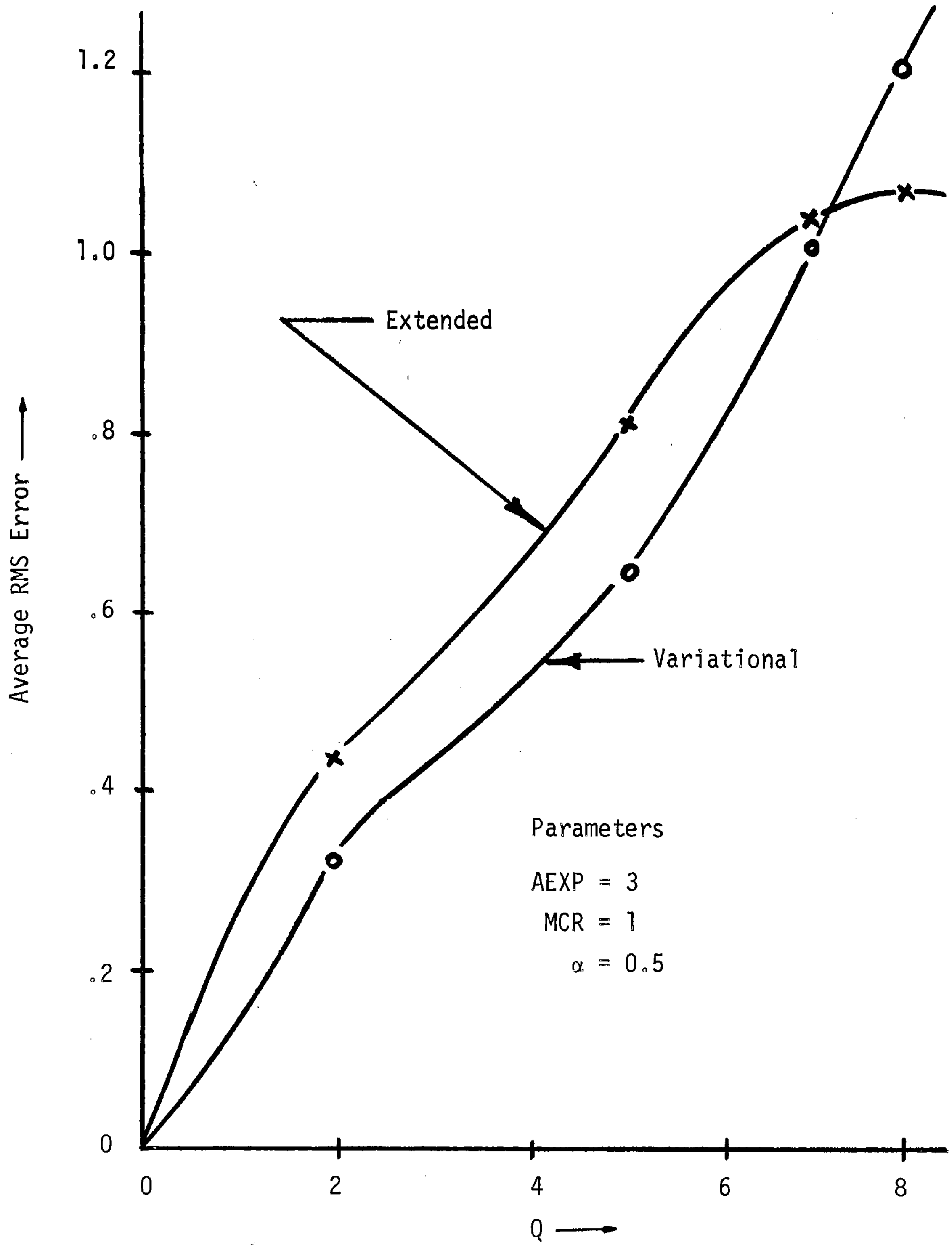

Figure 9. Average RMS Error Versus Input Noise Variance Q

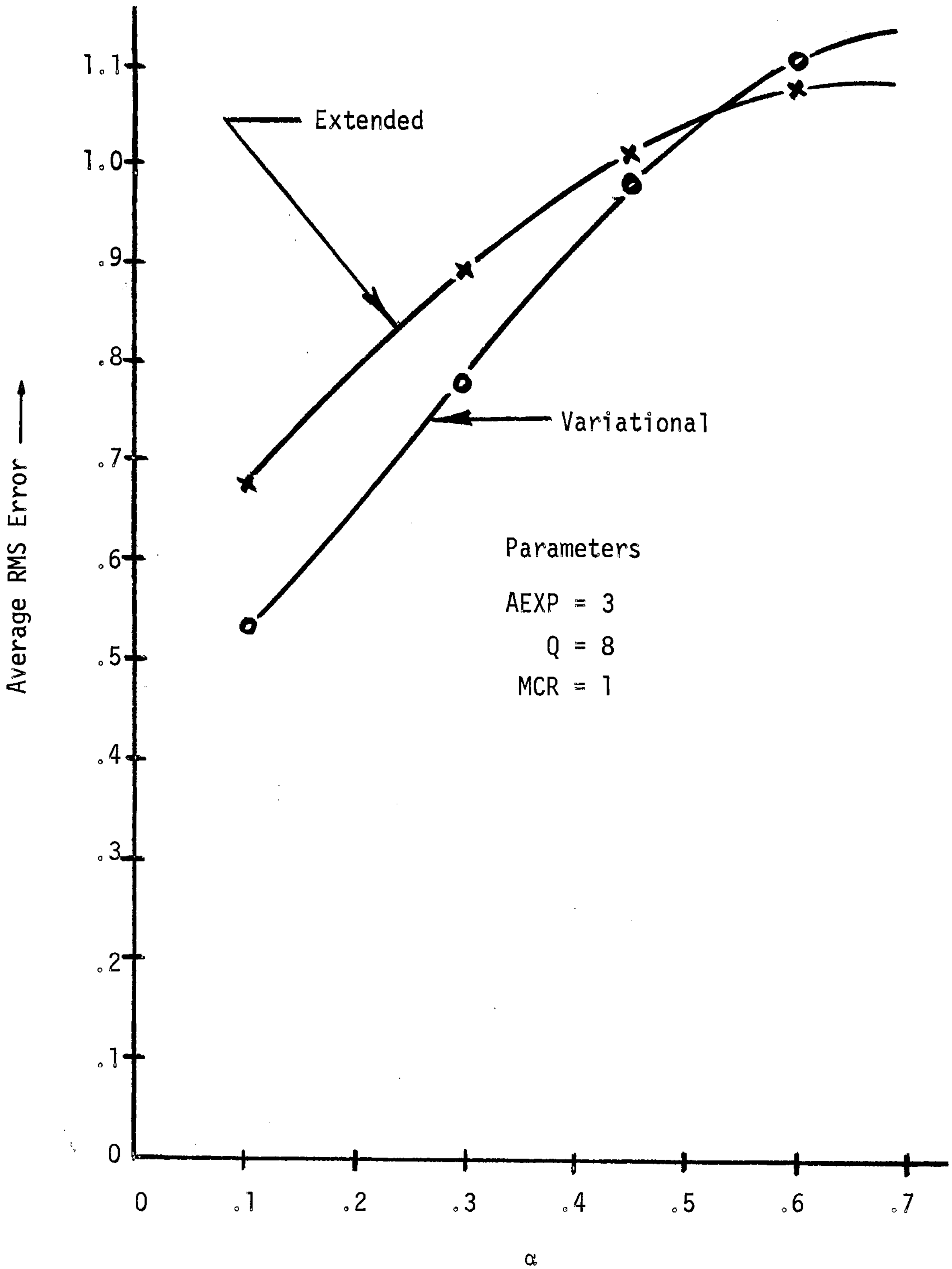

Figure 10. Average RMS Error Versus Magnitude of Nonlinearity

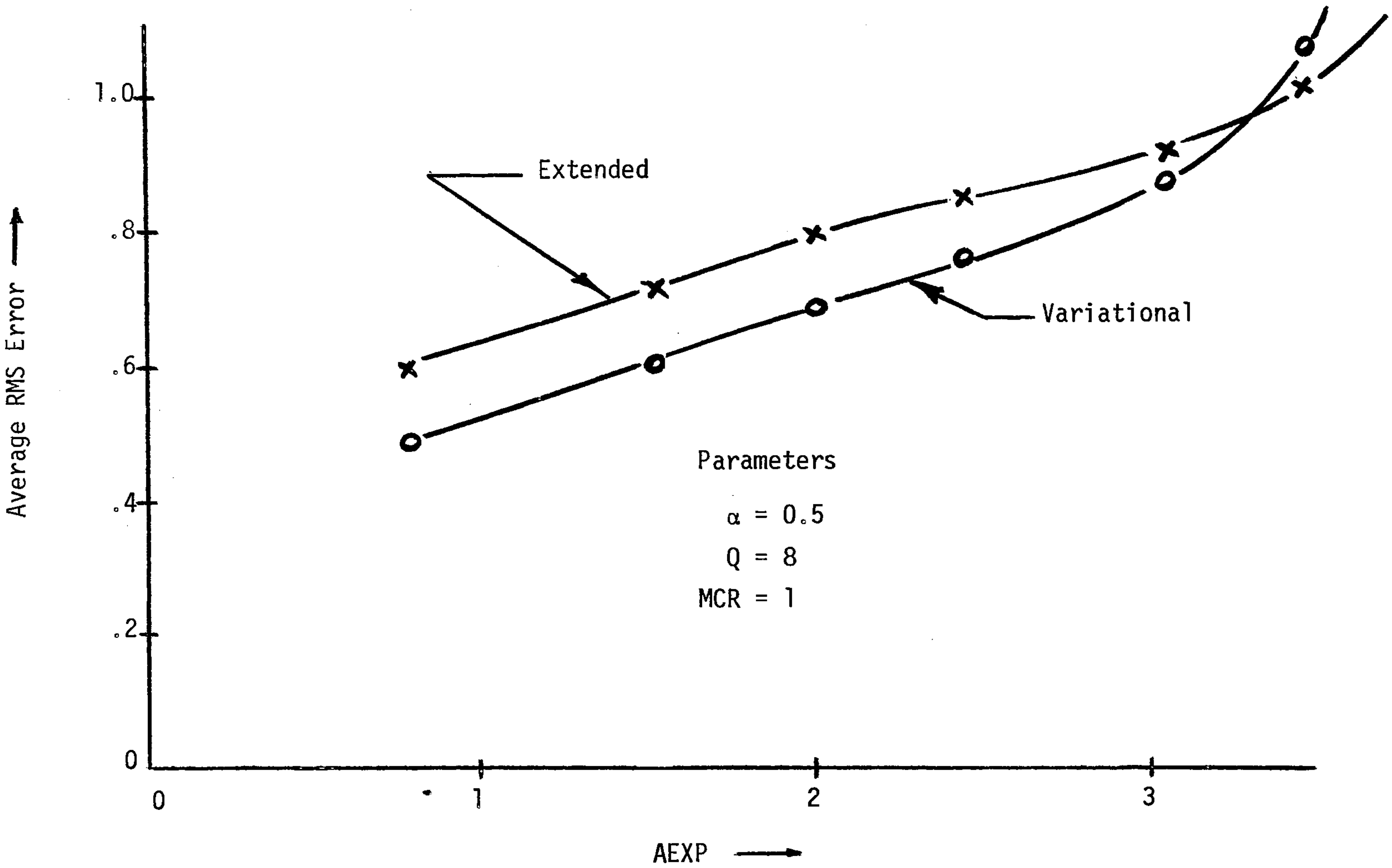

Figure 11. Average RMS Error Versus Order of Nonlinearity (AEXP)

Summary

An indication of the added complexity of the extended Kalman filter over the variational filter was identified in this chapter for both a general second-order system and for the specific second-order system considered in Chapter II. A ratio of the number of on-line computer operations was determined to penalize the extended filter by increasing the filter step size to yield filtering algorithms of approximately equal computational speed. A comparison of the two filtering algorithms based on both computational speed and accuracy was then made. Therefore a basis has been developed from which further studies involving the use of several well known integration methods may be made in Chapter IV.

CHAPTER IV

ON-LINE KALMAN FILTERING USING DIFFERENT NUMERICAL INTEGRATION FORMULAS

This chapter is concerned with the on-line simulation of the nonlinear stochastic system described in Chapter II using both the extended and the variational Kalman filtering algorithms described in Chapter I. In particular, several different integration methods are investigated using the step sizes for each filter as determined in Chapter III for algorithms of equal computational speeds. It is assumed that there are two computers available, a fast computer and a relatively slow computer. Step sizes for three different integration algorithms were determined by considering the total number of weighted on-line operations. Comparisons involving simulations of the two computers were made between single-step methods (RK4 and RK2). The more accurate of these methods was compared with a multistep method (AB2) for the slower computer. Results were verified by ensemble-averaging 100 Monte Carlo runs.

Single-Step Integration Formulas

Integration formulas may be classified as either single-step or multistep methods. One important group of single-step algorithms are the Runge-Kutta (RK) methods. Because they involve only first-order derivative evaluations, they are computationally simpler than the higher-order Taylor formulas but produce results equivalent in accuracy to higher-order formulas. For differential equations of the form $d\underline{x}/dt = \underline{f}(\underline{x},t)$, RK methods require the evaluation of $\underline{f}(\underline{x},t)$ at two, three, and four values of t on the interval $t_i \leq t \leq t_{i+1}$ for second, third, and fourth order approximations, respectively. Two of the several available RK methods were used to implement the filtering equations for both the extended and variational filtering algorithms in this chapter. The first was a second-order Runge-Kutta formula (RK2) which requires two derivative evaluations and two projections as shown geometrically in Figure 12. The algorithm is given by

$$\underline{x}_{i+1} = \underline{x}_i + \frac{h}{2}(\underline{m}_0 + \underline{m}_1) \tag{4.1}$$

where h is the step size and

$$\begin{aligned} \underline{m}_0 &= \underline{f}(\underline{x}_i, t_i) \\ \underline{m}_1 &= \underline{f}(x_i + \underline{m}_0 h, t_{i+1}) \end{aligned} \tag{4.2}$$

The expression for $\underline{m}_1$ in Equation (4.2) uses a predicted value of $\underline{x}$ at the endpoint of the interval, i.e. $\underline{x}_i + h\underline{m}_0$. The corrector equation in (4.1) utilizes the average between the initial slope ($\underline{m}_0$) and a projected slope (m_1) for proceeding over the entire interval.

The second Runge-Kutta method used in this chapter approximates the derivative at four points over the interval $[t_i, t_{i+1}]$, instead of only two. Figure 13 geometrically describes this fourth-order method (RK4) given by

$$\underline{x}_{i+1} = \underline{x}_i + \frac{h}{6}(\underline{m}_0 + 2\underline{m}_1 + 2\underline{m}_2 + \underline{m}_3) \tag{4.3}$$

where

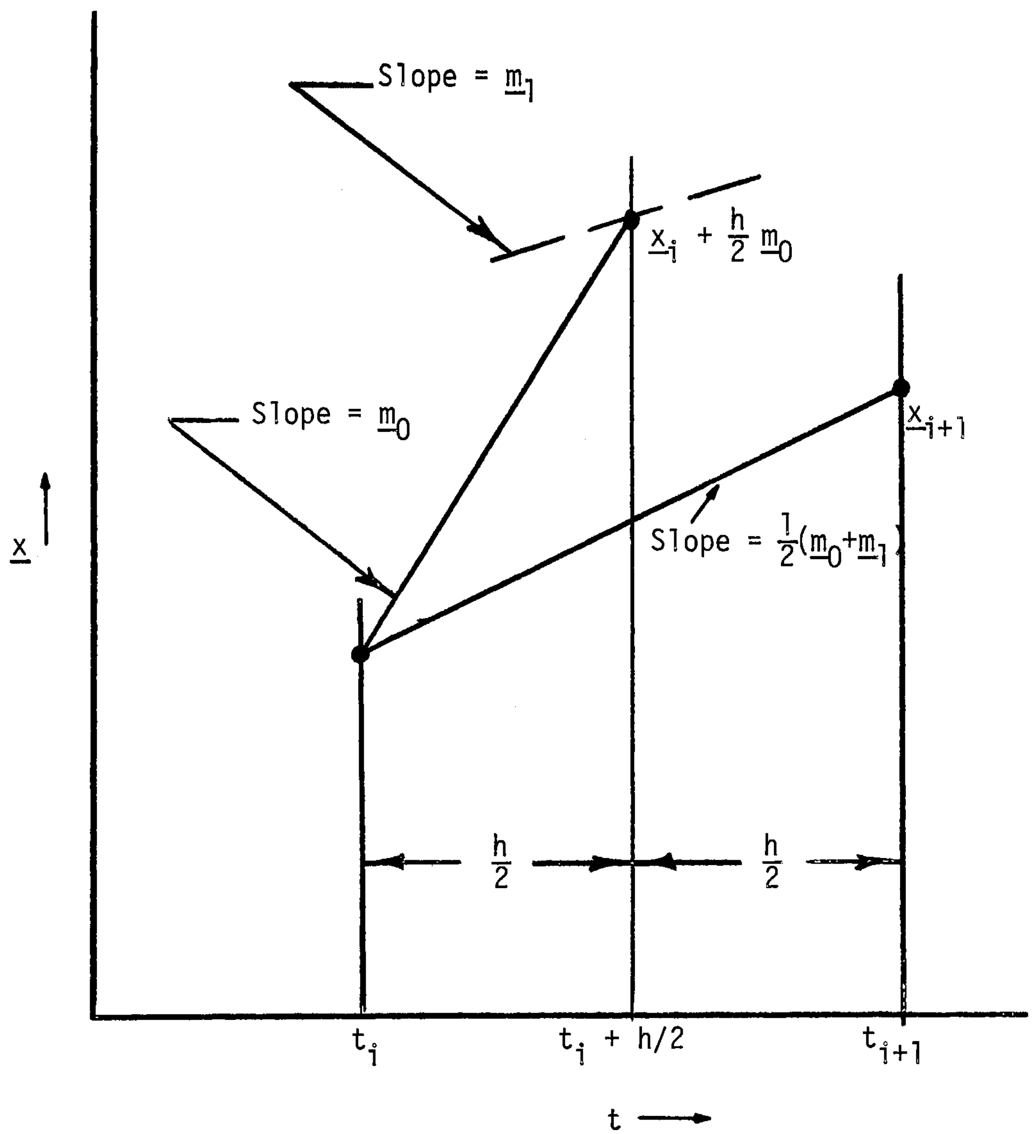

Figure 12. Geometrical Interpretation of Equations (4.1) and (4.2)

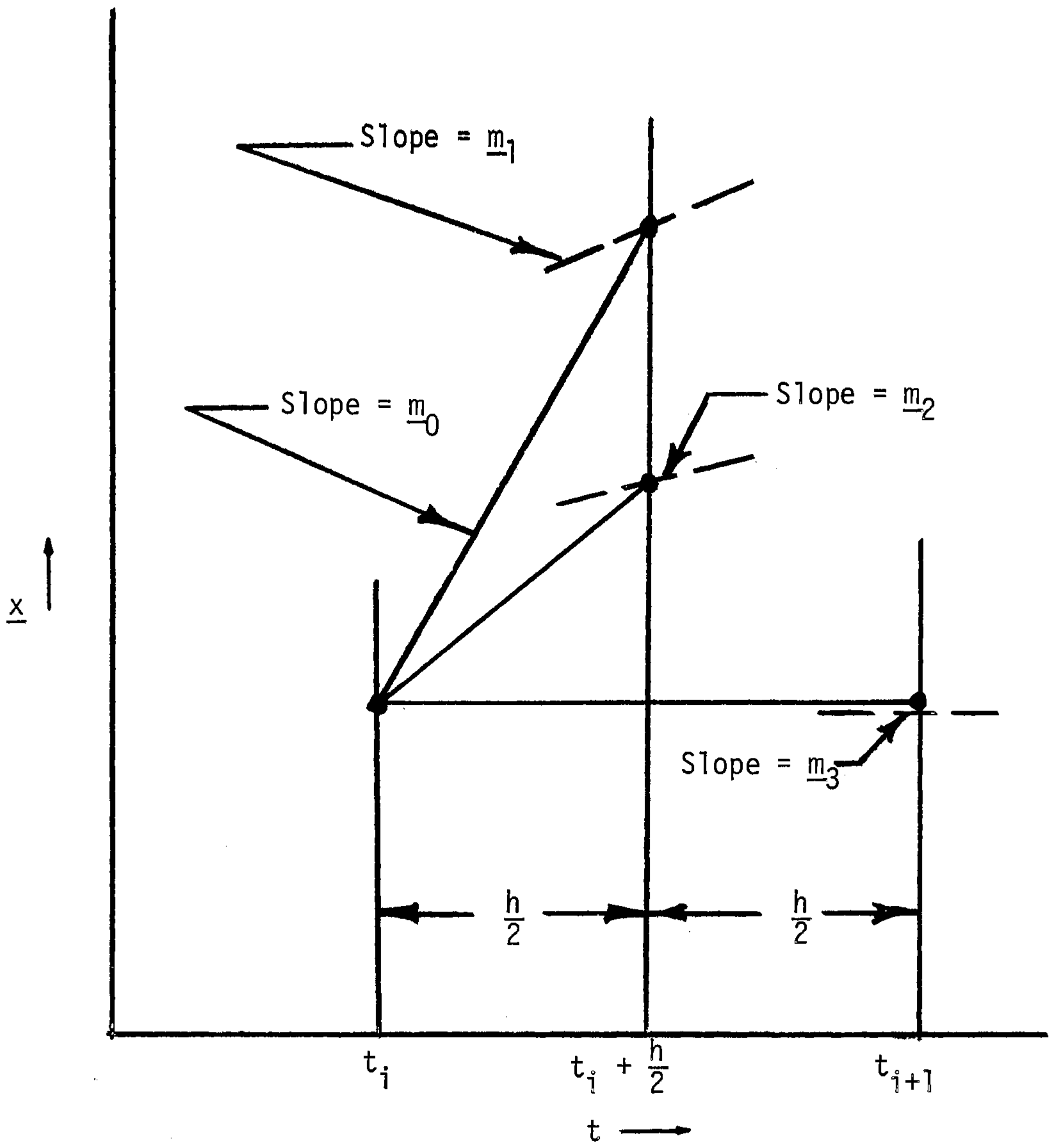

Figure 13. Geometrical Interpretation of Equations (4.3) and (4.4)

$$\underline{m}_0 = \underline{f}(\underline{x}_i, t)$$

$$\underline{m}_1 = \underline{f}(\underline{x}_i + \frac{\underline{m}_0 h}{2}, t_i + \frac{h}{2}) \tag{4.4}$$

$$\underline{m}_2 = \underline{f}(\underline{x}_i + \frac{\underline{m}_1 h}{2}, t_i + \frac{h}{2})$$

$$\underline{m}_3 = \underline{f}(\underline{x}_i + \underline{m}_2 h, t_{i+1})$$

Numerical Results Using Single-Step Methods

Because of its extensive use in a wide variety of areas and because of its high degree of accuracy, the RK4 integration method used exclusively in Chapters II and III was also used in the research described in this chapter as a basis of comparisons with other integration methods. The other single step method chosen (RK2) provides a clear comparison between a lower-order integration method and a higher-order method with integration step sizes properly adjusted for differences in the number of on-line computations. As noted earlier, the total number of weighted operations involved in implementing the variational filter for the system in Equation (2.5) using RK4 is 116. For the extended filter, this corresponding number is 292. When the filter equations are integrated for (2.5) by RK2, the total number of weighted operations is 58 for the variational Kalman filter and 148 for the extended Kalman filter as indicated in Table II.

It was assumed that for the nonlinear system equations considered, the filter can be operated on-line for a filter step size HF = .05 seconds with 120 weighted operations per step. Let this computer be designated as Computer I. These conditions then allow the variational

TABLE II

TOTAL NUMBER OF OPERATIONS AND CORRESPONDING STEP SIZES FOR THE THREE INTEGRATION METHODS

	Total Number of Weighted Operations		Step Size for Integration of Filter Equation	
	Variational Kalman Filter	Extended Kalman Filter	Variational Kalman Filter	Extended Kalman Filter
Fourth-Order Runge-Kutta	116	292	Computer I: HF=.05 sec. Computer II: HF=.10 sec.	Computer I: HF=.125 sec. Computer II: HF=. 25 sec.
Second-Order Runge-Kutta	58	148	Computer I: HF=.025 sec. Computer II: HF=.05 sec.	Computer I: HF=.0625 sec. Computer II: HF=.125 sec.
Second-Order Adams-Bashforth	31	84	HF = .025 sec.	

filter, using RK4 and having 116 weighted operations per step, to be operated at HF = .05 seconds. The extended filter using RK4 must then be operated at HF = .125 seconds as shown in Table II. The step sizes for both filters using RK2 was found in the same manner to be HF = .025 seconds for the variational filter and HF = .0625 seconds for the extended filter. In order to compare performance at the same points in time, such as at every 0.125 seconds, the error was determined every other step for the extended filter using RK2. Because of its obvious significance in filtering applications, only the input noise variance was varied in this work on different integration methods. Numerical results for RK2 with Computer I are shown in Figure 14. Figures 9 and 14 illustrate that the lower order integration method (RK2) was superior to the higher order integration method (RK4) used with either filter for Computer I.

Another computer (II) operating at a slower speed implementing both filters using both RK2 and RK4 was introduced to give further insight into the favorable operating conditions of each filter-integration method combination. Computer II can perform only 60 weighted operations for a step size HF = .05 seconds and must therefore be operated at HF = .01 seconds for RK4 to perform the same number of operations as Computer I in the same total time. Step sizes used for Computer II are given in Table II, and numerical results are shown in Figures 15 and 16. These curves illustrate that only a slightly lower average RMS error was obtained when using RK2 with both filters. The combinations of RK2 with the variational filter and RK2 with the extended filter both yielded convergent numerical results for higher input noise ($Q \leq 7$) than the RK4-filter combinations, which became

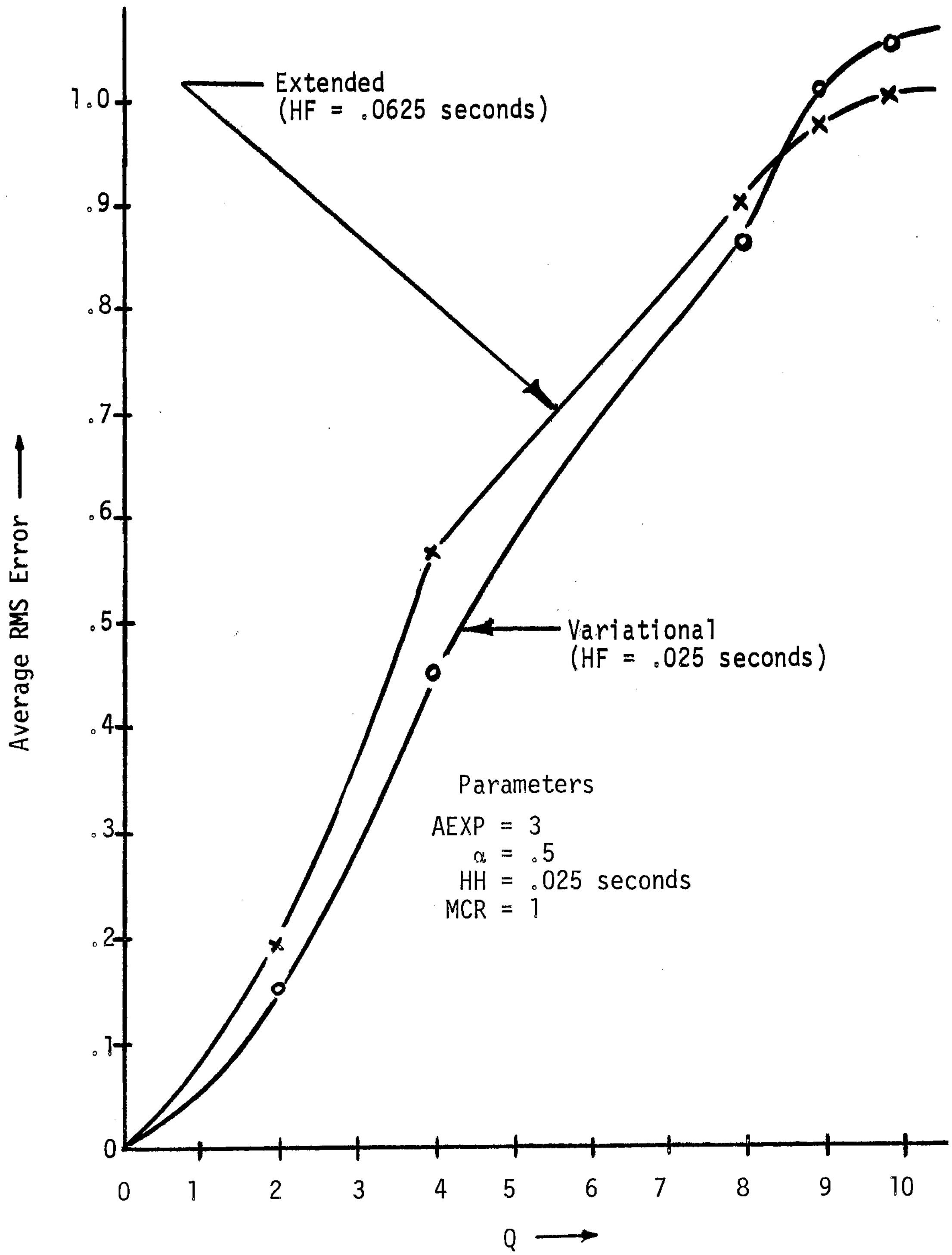

Figure 14. Average RMS Error Versus Input Noise Variance (Q) with Computer I Using RK2

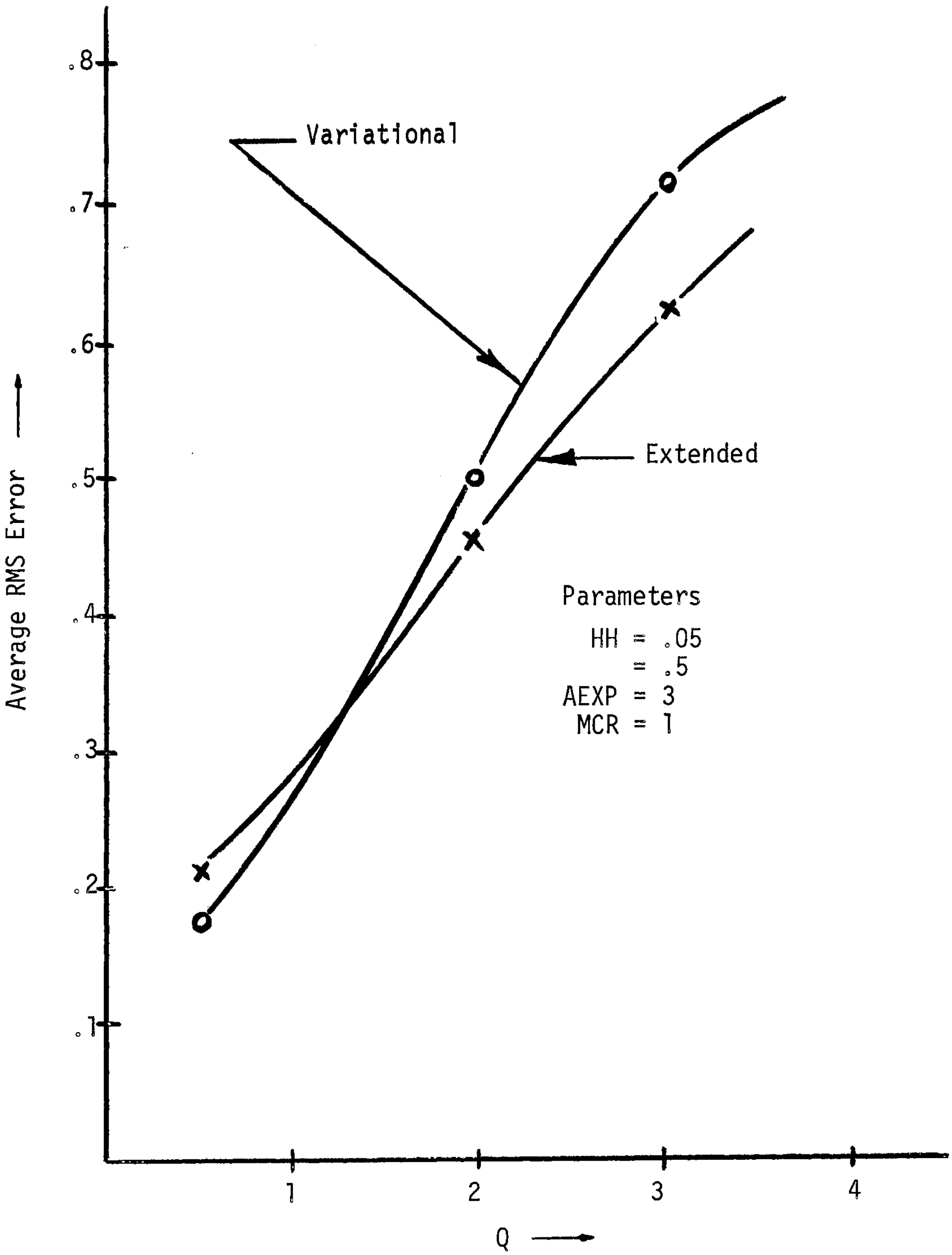

Figure 15. Average RMS Error Versus Input Noise Variance (Q) Using RK4 for Computer II

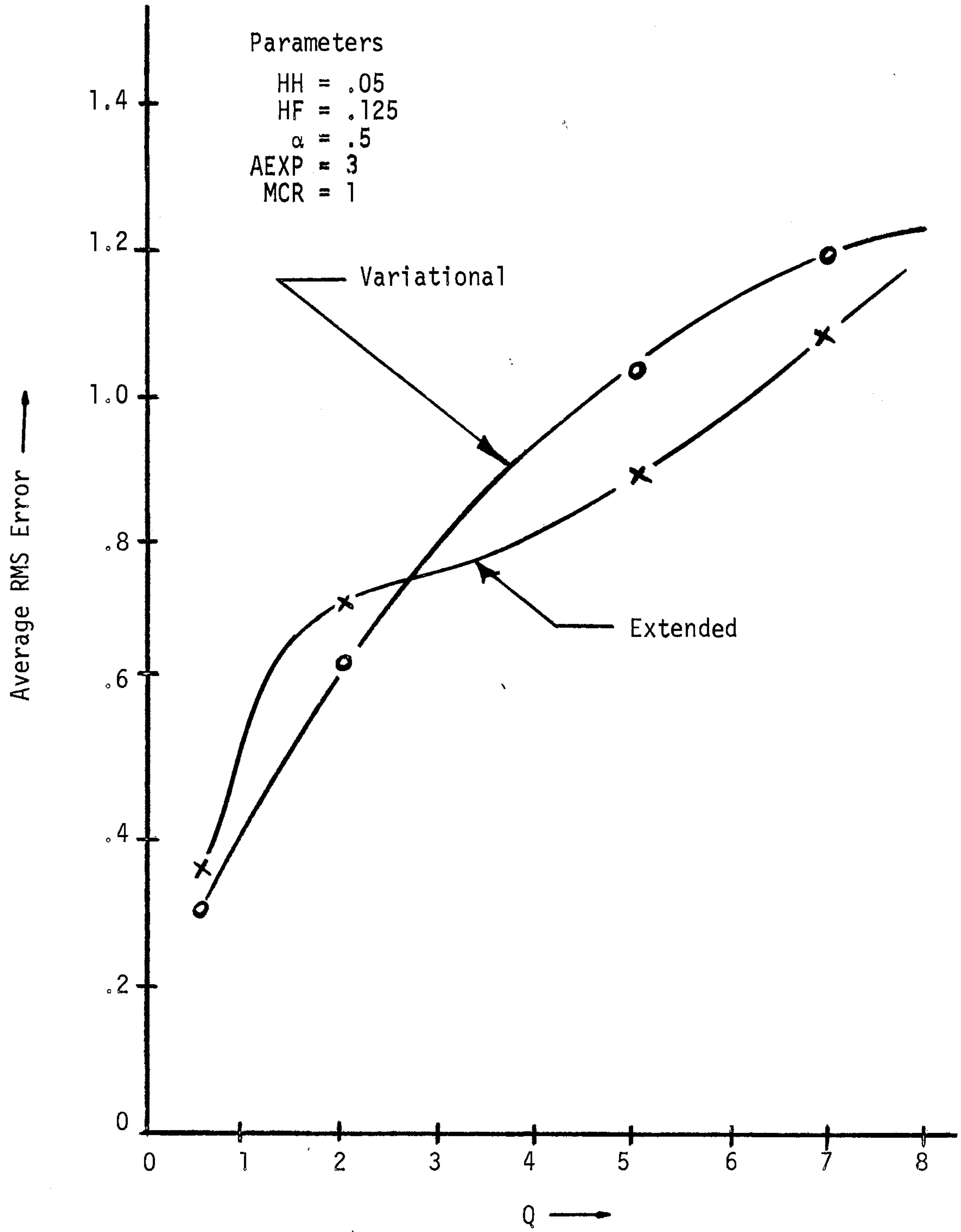

Figure 16. Average RMS Error Versus Input Noise Variance (Q) Using RK2 for Computer II

divergent for Q > 3. Comparing the results in Figure 14 with the results in Figure 15 for the moderate input noise of Q = 2 shows that the average RMS error of 0.413 for the RK2-variational filter combination is the lowest error of the four combinations. Other combinations yielded errors from 6.8% to 22.4% greater than this lowest error.

As shown in Chapter III, it is apparent that the extended filter performs better than the variational for much higher values of input noise when Computer I was used. The cross-over point beyond which the variational filter performed better than the extended filter occurred for much milder input noise conditions when the larger step size corresponding to Computer II was used for the filter equations.

To verify the results of these single sample functions for moderate input noise, Monte Carlo ensemble averages were run for all four combinations for Computer II. For 100 Monte Carlo runs, the sample error variances for the variational and extended filters using both RK2 and RK4 are plotted in Figure 17. A slightly lower input noise variance (Q) was used because of occasional numerical instability problems over the 100 runs. The RK2-variational filter combination was again clearly superior to the other three combinations. Although the differences between the RK4 combinations were smaller, the RK2-extended filter combination generally performed better over the entire five second solution than either filter with the RK4 method. The performances of the RK4 combinations were both more erratic with higher variances of error than either RK2 combination. Therefore, the lower-order method was consistently superior to the higher-order method for the larger step size (Computer II), while the differences were not so pronounced at the smaller step size (Computer I).

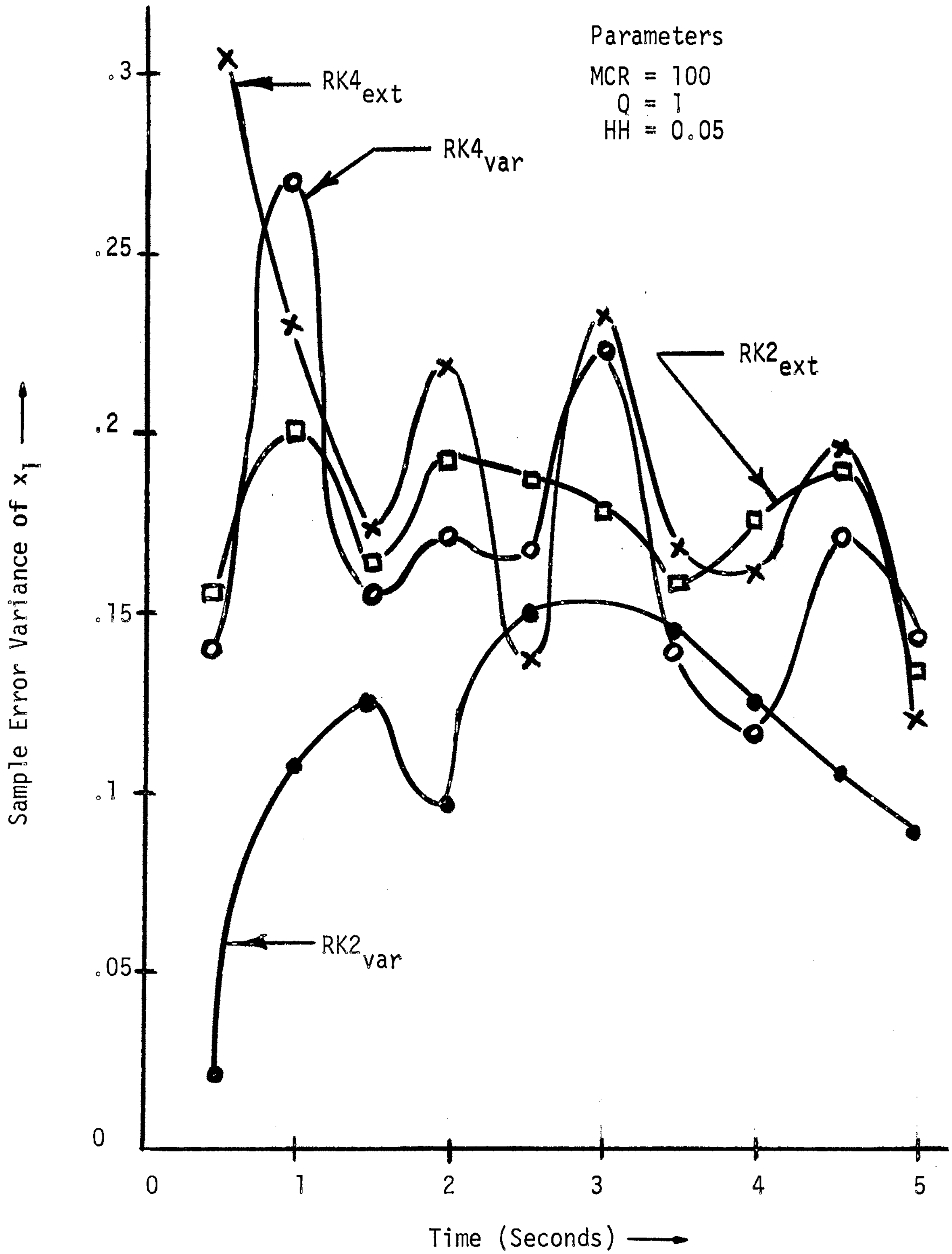

Figure 17. Sample Error Variances Using RK2 and RK4

Multistep Integration Formulas

Integration formulas which require information not only at t_i but also outside the integration interval under consideration $[t_i, t_{i+1}]$ are referred to as multistep methods. A disadvantage of these methods is the requirement of additional information to start the procedure. However, these methods do require considerably less computational time than single-step methods. Multistep methods include predictor, corrector, and predictor-corrector methods.

A simple multistep predictor method called the Adams-Bashforth second-order method (AB2) has been chosen for comparison with the two single-step methods considered earlier in this chapter. This particular multistep method utilizes a single past slope as shown in Figure 18. A polynomial is formed and extended to time t_{i+1} to determine $\underline{x}_{i+1}$ to form the equation

$$\underline{x}_{i+1} = \underline{x}_i + \frac{h}{2}\left(3\dot{\underline{x}}_i - \dot{\underline{x}}_{i-1}\right) \tag{4.5}$$

Equation (4.5) is used to update $\underline{x}$, and $\dot{\underline{x}}$ is updated by the equation

$$\dot{\underline{x}}_{i+1} = \underline{f}(\underline{x}_{i+1}, t_{i+1}) \tag{4.6}$$

Together, Equations (4.5) and (4.6) mathematically form the second-order Adams-Bashforth (AB2) method.

Numerical Results Using Multi-Step Formulas

Once again the initial step in obtaining numerical results for comparison purposes was to determine the weighted number of operations and the associated filter integration step size. This information is tabulated in Table II. Because the variational filter requires 31

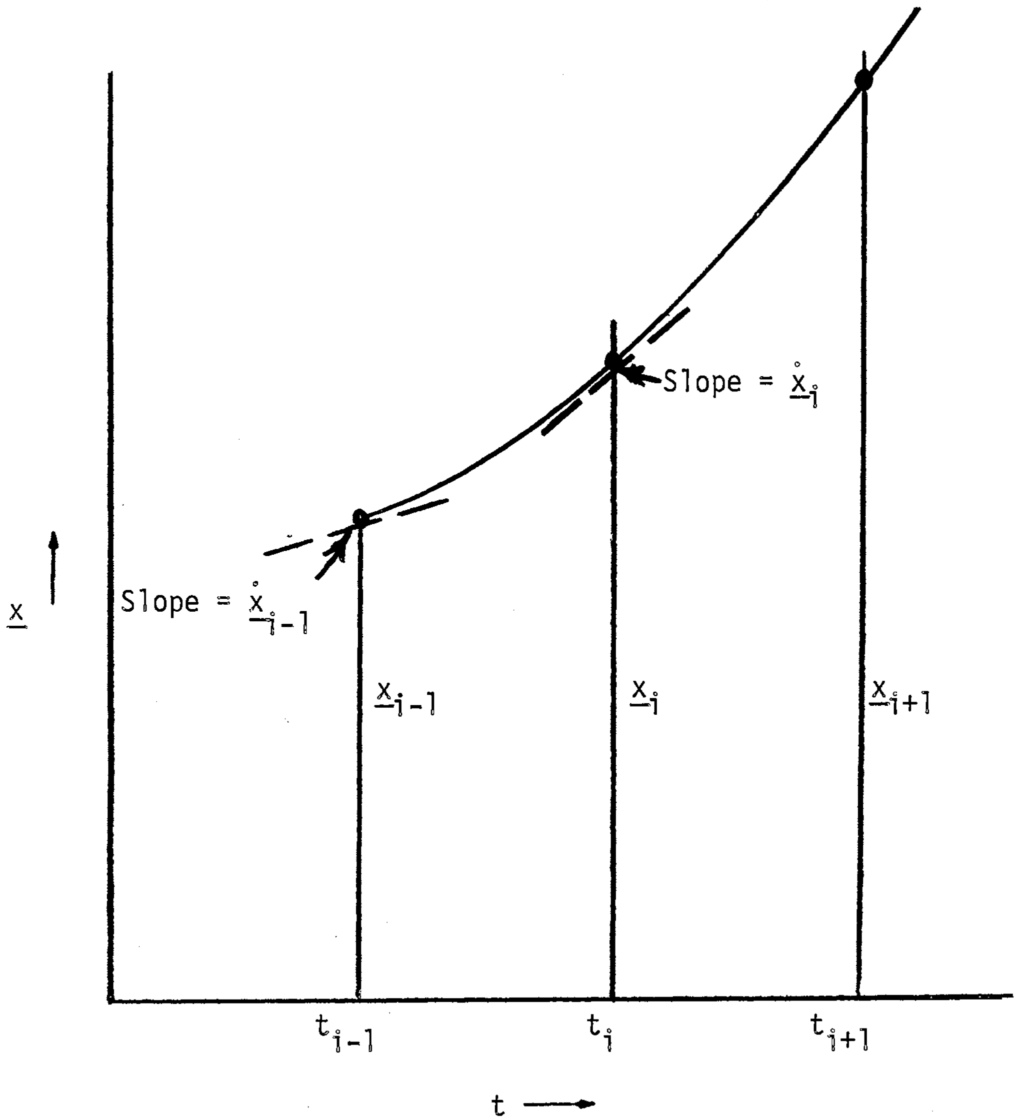

Figure 18. Second-Order Adams-Bashforth Method

weighted operations per step when integrated by AB2, a step size of HF = .025 seconds was used with Computer II. Figure 19 shows how the AB2 variational filter combination performed for different input noise variances (Q). Comparing Figures 15, 16, and 19 demonstrates that the second-order multistep method (AB2) operated with much greater accuracy for all values of input noise variance considered than did either RK4-filter combination. A similar comparison with the curves of the RK2-filter combinations was inconclusive for a set of single sample functions, even though the same set of random number was used in every case. This observation indicated that a Monte Carlo ensemble average was needed to provide more conclusive results. Such a comparison with both RK2-variational filter and RK2-extended filter combinations has been made in Figure 20. This more valid set of results for 100 Monte Carlo runs shows that the AB2-variational filter combination, at the same computational speed as the RK2 methods, was clearly superior to both RK2 methods.

The AB2 extended filter combination was not implemented due to difficulties in making comparisons at the same points in time. However, it is expected that its performance relative to the AB2-variational filter combination would have been similar to the relative performances of the RK2 and RK4 combinations.

Summary

Both single-step and multi-step methods were examined in conjunction with both the variational Kalman filter and the extended Kalman filter for a variety of operating conditions. A "fast" computer (Computer I) and a "slow" computer (Computer II) were used to simulate

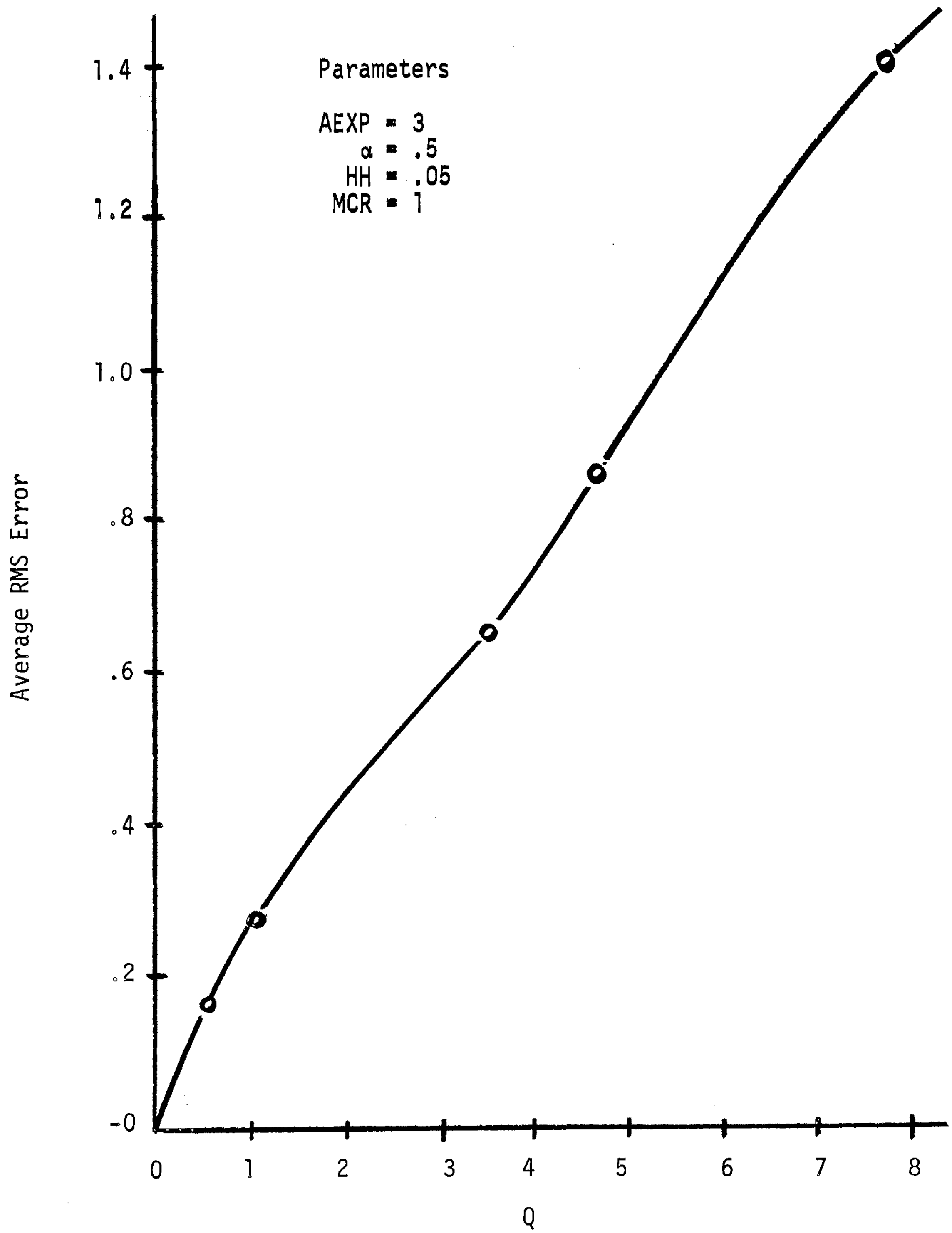

Figure 19. Average RMS Error Versus Input Noise Variance (Q) Using AB2

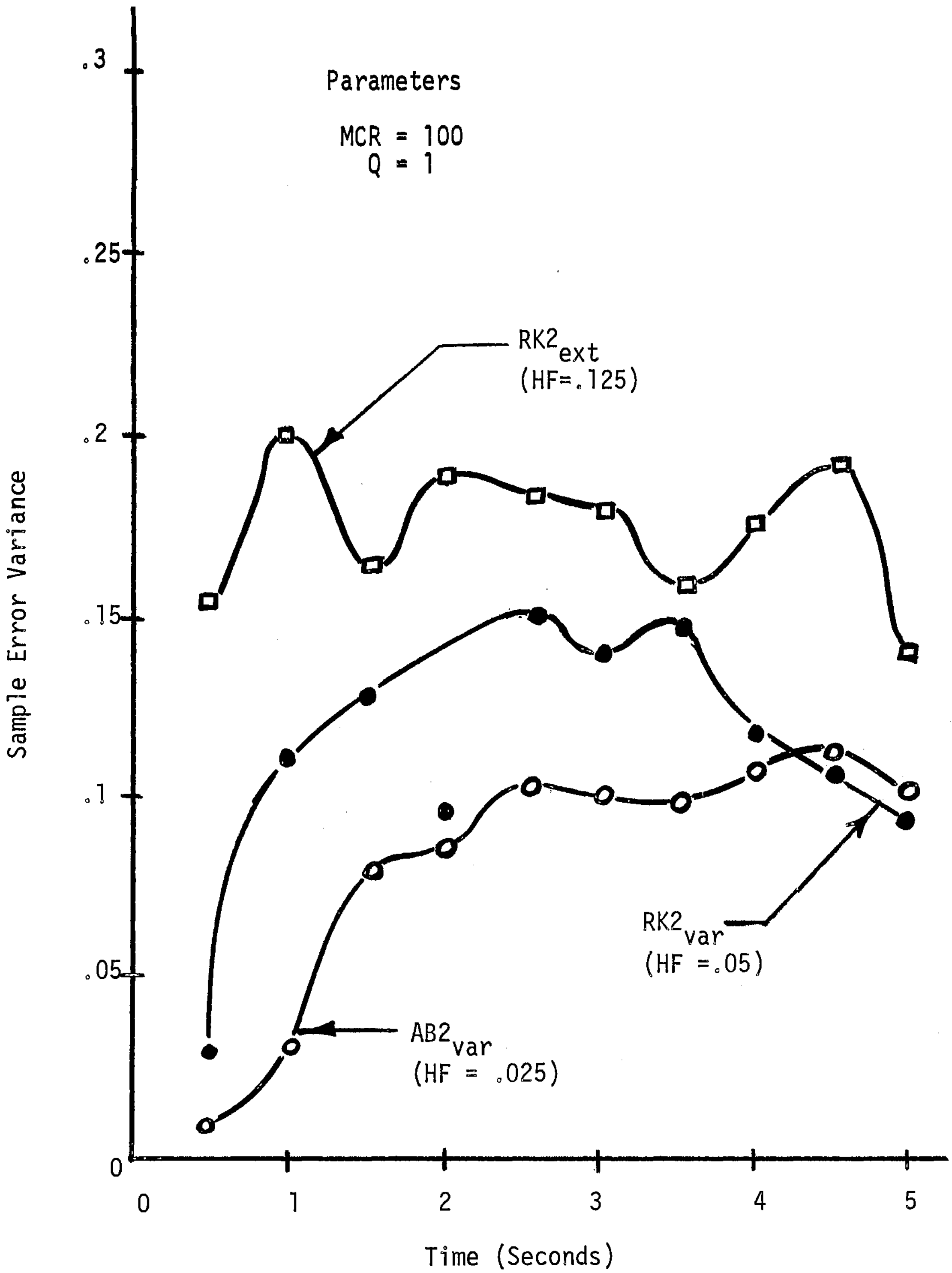

Figure 20. Sample Error Variance Using RK2 and AB2

the several integration method-filter combinations considered. It was determined that the lower order single-step method (RK2) was markedly better than the higher-order method (RK4) for Computer II with both filter combinations. The improved performance was only marginal at the smaller step size (Computer I) for a mild input noise. After adjusting step sizes to obtain filter-integration method combinations of approximately equal computational speed, it was shown that the multi-step method (AB2) was clearly more accurate than either of the single-step methods for both filters.

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

A general purpose digital computer simulation program was developed to carry out the research in this thesis. This program allowed the implementation of both the variational Kalman filter and the extended Kalman filter for a variety of significant operating conditions and filter integration methods. All simulation results are for a particular nonlinear stochastic system corrupted by zero-mean white Gaussian noise. Both filters were first compared for computational accuracy and computational speed as the input noise variance, order of nonlinearity, and magnitude of nonlinearity were varied. The results indicated that the extended filter is far superior to the variational filter in terms of a lower average RMS error when operating at the same step size. It was shown that for all values of input noise and system nonlinearities considered, the extended filter should be used where accuracy is the only important consideration. This conclusion is even more significant for the more harsh conditions of higher input noise, higher order nonlinearity, and larger nonlinearity, where the extended filter with its relinearization each step became increasingly more accurate than the variational filter. It was then concluded that a decision as to which filter to use for a given application should be determined by considering both accuracy and computational speed requirements. For highly nonlinear conditions the extended Kalman filter should be utilized, but for mild operating conditions the computationally simpler variational Kalman filter is more desirable. In making comparisons based on both computational speed and accuracy, the number of weighted operations for each filter was computed to determine the step size at which each filtering algorithm should be operated. Once this had been accomplished, it was found that there existed a set of conditions beyond which the variational Kalman filter should be used, since the extended filter had been penalized for its excessive number of on-line computations. For all but the most severely nonlinear conditions, the variational Kalman filter proved to have a lower average RMS error than the extended Kalman filter for the same sequence of random input noise and with the integration of both system equations and filter equations by the fourth-order Runge-Kutta method. These results provided a basis for comparison of several single-step and multistep integration methods when used to integrate the filter equations for both a slow computer and a fast computer. First, it was necessary to again adjust the step sizes for each integration method as a function of the number of operations each must perform every step and then to compare errors at the same points in time. It was determined that for a smaller step size, a lower-order integration method should be used for a lower estimation error and for a more stable solution. These conclusions were supported by a Monte Carlo ensemble average of 100 runs. The second-order multistep method was found to be much more accurate than either single-step method used to integrate each filter at the larger step size.

Recommendations for Further Study

The general purpose digital computer simulation program developed for this research may be easily adapted to a large variety of nonlinear stochastic systems and integration methods for further investigation of their combined use. Investigation of higher-order, more complex systems with smaller time constants would provide a broader area of preferred operation for each nonlinear filter considered in this research effort. More valid results might be provided by making large numbers of Monte Carlo ensemble averages rather than single sample functions.

Another area which might be of importance in estimating the states of a nonlinear stochastic system using different integration methods is that of considering other nonlinear filters described in the literature. Additional problems such as truncation error, roundoff error, mixed step length, numerical instability, and special methods for the linear parts of a system (23) might be considered for combinations of integration methods and filtering algorithms of interest.

SELECTED BIBLIOGRAPHY

(1) Kalman, R. E. "A New Approach to Linear Filtering and Prediction Problems." J. Basic Engineering, Transactions ASME, Vol. 82-D (March 1960), pp. 35-45.

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

(3) Sage, A. P. Optimum Systems Control. Englewood Cliffs, N. J.: Prentice-Hall, 1968.

(4) Athans, M. and E. Tse. "A Direct Derivation of the Optimal Linear Filter Using the Maximum Principle." IEEE Transactions on Automatic Control, Vol. AC-12, No. 6 (December 1969), pp. 690-698.

(5) Sage, A. P. and J. L. Melsa. Estimation Theory with Applications to Communications and Control. New York: McGraw-Hill Book Co., 1971.

(6) Meditch, J. S. Stochastic Optimal Linear Estimation and Control. New York: McGraw-Hill Book Co., 1969.

(7) Sims, C. S. and J. L. Melsa. "Specific Optimal Estimation." IEEE Transactions on Automatic Control, Vol. AC-14, No. 2 (1969), pp. 183-186.

(8) Bellman, R., et al. "Invariant Imbedding and Nonlinear Filtering Theory." Rand Corporation, Report No. 4374-PR, December 1964.

(9) Wonham, W. M. "Some Applications of Stochastic Differential Equations to Optimal Nonlinear Filtering." J. SIAM Control, Ser. A. Vol. 2, No. 3 (1965), pp. 347-369.

(10) Bucy, R. S. "Nonlinear Filtering Theory." IEEE Transactions on Automatic Control, Vol. AC-10, No. 2 (April 1965), pp. 198-199.

(11) Frost, P. A. "Nonlinear Estimation in Continuous Time Systems." Stanford University, Systems Theory Lab. Tech. Report 6304-4, U. S. Air Force Contract F33615-67-C-1245, May 1968.

(12) Sage, A. P. "Maximum A-posteriori Filtering And Smoothing Algorithms." International Journal of Control, Vol. 11, No. 3 (May 1970), pp. 641-658.

(13) Jazwinski, A. H. Stochastic Processes and Filtering Theory. New York: Academic Press, 1970.

(14) Kushner, H. J. "Approximations to Optimal Nonlinear Filters." IEEE Transactions on Automatic Control, Vol. AC-12, No. 5 (October 1967), pp. 546-556.

(15) Wishner, R. P., J. A. Tabaczynski and M. Athans. "A Comparison of Three Nonlinear Filters." Automatica, Vol. 5, No. 4 (July 1969), pp. 487-496.

(16) Schwartz, L. and E. B. Stear. "A Computational Comparison of Several Nonlinear Filters." IEEE Transactions on Automatic Control, Vol. 13, No. 1 (February 1968), pp. 83-86.

(17) Bendat, J. S. and A. G. Piersol. Random Data: Analysis and Measurement Procedures. New York: John Wiley and Sons, Inc., 1971.

(18) Martens, H. R. "A Comparative Study of Digital Integration Methods." Simulation, Vol. 12, No. 2 (February 1969), pp. 87-94.

(19) Giese, C. "State Variable Difference Method for Digital Simulation." Simulation, Vol. 8, No. 5 (May 1967), pp. 263-271.

(20) Wait, J. V. "State-Space Methods for Designing Digital Simulations of Continuous Fixed Linear Systems." IEEE Transactions on Electronics Computers, Vol. EC-16 (June 1967), pp. 351-355.

(21) Liou, M. L. "A Novel Method of Evaluating Transient Response." Proceedings IEEE, Vol. 54 (January 1966), pp. 20-23.

(22) Mastascusa, E. J. "A Relation Between Liou's Method and the Fourth-Order Runge-Kutta Method for Evaluating Transient Response." Proceedings IEEE (Letters) (May 1969), pp. 803-804.

(23) Benyon, P. R. "A Review of Numerical Methods for Digital Simulation." Simulation, Vol. 11, No. 5 (November 1968), pp. 219-238.

(24) Rowland, J. R. and W. M. Holmes. "A Variational Approach to Digital Integration." IEEE Transactions on Computers, Vol. C-20, No. 8 (August 1971), pp. 894-900.

(25) Ralston, A. and H. S. Wilf. Eds. Mathematical Methods for Digital Computers. New York: Wiley and Sons, 1960.

(26) Milne, W. E. Numerical Solution of Differential Equations. New York: Wiley and Sons, 1953.

(27) Hildebrand, F. R. Introduction to Numerical Analysis. New York: McGraw-Hill, 1956.

(28) Kopal, Z. Numerical Analysis. London: Chapman and Hall, 1955.

(29) Merson, R. H. "The Stability of the Runge-Kutta Method of Solution of Linear Differential Equations." Royal Aircraft Establishment Technical Note GW320, June 1954.

(30) Rice, J. R. "Split Runge-Kutta Method for Differential Equations." Journal of Research of the National Bureau of Standards, Vol. 64B, No. 3 (July - September 1960), pp. 151-170.

(31) Forrington, C. V. D. "Extensions of the Predictor-Corrector Method for the Solution of Systems of Ordinary Differential Equations." Math. of Comp., Vol. 16, No. 78 (April 1961), pp. 176-187.

(32) Blum, E. K. "A Modification of the Runge-Kutta Fourth-Order Method." Math. of Comp., Vol. 16, No. 78 (April 1962), pp. 80-84.

(33) Gill, S. "Process for the Step-by-Step Integration of Differential Equations in an Automatic Digital Computating Machine." Proc. Cambridge Phitos. Soc., Vol. 47, Part 1 (1951), pp. 96-108.

(34) Lance, G. N. Numerical Methods for High Speed Computers. London: Illiffe, 1960.

(35) Rowland, J. R. and W. M. Holmes. "Statistical Analysis Techniques for Error Propagation in Large-Scale Missile Systems." U. S. Army Missile Command, Technical Rpt. No. RG-TR 71-19, August 1971.

APPENDIX

COMPUTER PROGRAM FOR SIMULATING NONLINEAR STOCHASTIC SYSTEMS USING CONTINUOUS KALMAN FILTERS

The general purpose digital computer simulation program listed here has several different operational modes. Any one of several integration methods may easily be used for both the system equations and filter equations separately with provisions for integrating the system at a different step size (HH) than the step size used for the filter equation (HF). The system equations, defined in subroutine XEQN, may be altered by changing the order of the nonlinearity (AEXP) or the magnitude of the nonlinearity (ALPHA). There is also a provision for a single sample function (MCR = 1), or Monte Carlo ensemble averaging over any desired number of runs (MCR = number of runs). The sample variance of each state of the system (VARX, VARX2) is computed from a ten-point average. Either filter may be implemented by setting KALF = VAR or KALF = EXT for the variational Kalman filter and the extended Kalman filter, respectively.

```
C
C     THIS PROGRAM SIMULATES A NONLINEAR STOCHASTIC SYSTEM
C     USING ANY ONE OF SEVERAL INTEGRATION METHODS WITH
C     EITHER THE VARIATIONAL KALMAN FILTER OR THE EXTENDED
C     KALMAN FILTER.
C
C     KALF=EXT   EXTENDED KALMAN FILTER
C     KALF=VAR   VARIATIONAL KALMAN FILTER
C     MCR=1   SINGLE SAMPLE FUNCTION
C     MCR=  DESIRED NUMBER OF MONTE CARLO RUNS
C     ALPHA=  WEIGHTING FACTOR FOR MAGNITUDE OF NONLINEARITY
C     AEXP=  ORDER OF NONLINEARITY
C     NS=  ORDER OF SYSTEM
C
      INTEGER RS,VAR,EXT
      REAL K
      DIMENSION DXP(2),DPP(2,2)
      DIMENSION XN(4),XND(4),DP(4,4),P(4,4),X(4),DX(4),XH(4),DXH(4),HTRI
     *(4,2),W(2),W1(2),XHT(4),VANDW(2)
      DIMENSION SUMX1(20),SSQX1(20),X1BAR(20),VARX1(20),
     *SUMX2(20),SSQX2(20),X2BAR(20),VARX2(20),XE(2)
      COMMON B(4,2),K(4,2),H(2,4),R(2,2),HT(4,2),Q(2,2),KUTTA,BB,HH,V,G,
     *VN(2),MS,RS,KALF,NS,RI(4,4),ALPHA,AEXP,HF
      COMMON/BLOCK2/IX,DUM
      COMMON/BLOCK3/DPP,DXP
      DATA WMEAN1,WMEAN2,VMEAN1,VMEAN2/4*0.0/
      DATA VAR,EXT/3HVAR,3HEXT/
      DATA W1/2*0.0/
      DATA KC,KC1/5,5/
      IX=31571
      DUM=0.1
      AEXP=3.
      KALF=EXT
      NTOT=20
      HH=0.05
      HF=0.025
      MTOT=20
      MCR=100
      DO 501 M1=1,10
      SUMX1(M1)=0.
      SUMX2(M1)=0.
      SSQX2(M1)=0.
  501 SSQX1(M1)=0.
C     SYSTEM CONSTANTS, IC'S
      NS=2
      MS=1
      RS=1
      DO 502 MC=1,MCR
      RMSER=0.0
      DO 789 I=1,NS
      DXP(I)=0.0
      DO 789 J=1,NS
  789 DPP(I,J)=0.0
      DO 5 I=1,MS
```

```
      DO 4 J=1,MS
      RI(I,J)=0.0
    4 R(I,J)=0.0
      DO 5 L=1,NS
      H(I,L)=0.
    5 HTRI(L,I)=0.0
      DO 2 I=1,RS
      DC 6 L=1,NS
    6 B(L,I)=0.
      W(I)=0.
      DO 2 J=1,RS
    2 Q(I,J)=0.
      DO 7 I=1,NS
      VN(I)=0.0
      XHT(I)=0.0
      XE(I)=0.0
      DO 7 J=1,NS
    7 P(I,J)=0.
      XHT(2)=0.1
      B(2,1)=1.0
      ALPHA=0.5
      Q(1,1)=1.0
      R(1,1)=0.1
      SIGV1=SQRT(R(1,1)/HF)
      SIGW1=SQRT(Q(1,1)/HH)
      X(1)=0.0
      X(2)=0.1
      XN(1)=0.0
      XN(2)=0.1
      XH(1)=0.0
      XH(2)=0.0
      H(1,1)=1.0
      DO 15 IY=1,MS
      DO 15 ID=1,NS
   15 HT(ID,IY)=H(IY,ID)
      RI(1,1) = 1.0/R(1,1)
      DO 16 ID=1,NS
      DO 16 IY=1,MS
      HTRI(ID,IY) = 0.0
      DO 16 IJ=1,MS
   16 HTRI(ID,IY)= HTRI(ID,IY) + HT(ID,IJ)*RI(IJ,IY)
      HIM=HH
      IF(KALF.EQ.EXT) KC=2
      LN=0
      DO 10 M1=1,10
      DO 30 IDO=1,2
      MNG=0
      DO 55 KAC=1,KC
C
C     ***** INTEGRATE SYSTEM EQUATIONS & NOMINAL TRAJECTORY BY RK4  *****
C
      HH=HIM
      IF(MNG.EQ.0) KC1=2
      IF(MNG.EQ.0) MNG=1
```

```
      IF(MNG.EQ.1) KC1=3
      IF(MNG.EQ.1) MNG=0
      IF(KALF.EQ.REG) KC1=1
      DO 257 KAB=1,KC1
      LN=LN+1
C
C     *****  GENERATE INPUT AND MEASUREMENT NOISE  *****
C
      CALL RNG(SIGW1,SIGV1,WMEAN1,VMEAN1,VANDW)
      VN(1)=VANDW(2)
      W(1)=VANDW(1)
C
C     *****  INTEGRATE SYSTEM EQUATIONS BY RK4  *****
C
      DO 3 KUTTA=1,4
      CALL XEQN(XN,XND,W1)
   26 CALL XEQN(X,DX,W)
    3 CALL RK4(XN,XND,X,DX)
  257 CONTINUE
      DO 55 IAB2=1,2
      DO 25 IJ=1,NS
      DO 25 JI=1,MS
      K(IJ,JI)=0.
      DO 25 IK=1,NS
   25 K(IJ,JI)=P(IJ,IK)*HTRI(IK,JI) + K(IJ,JI)
      IF(KALF.EQ.EXT) HH=HF
      IF(KALF.EQ.REG) HH=0.025
      IF(KALF.EQ.REG) CALL RSC(XN,DP,P)
      IF(KALF.EQ.EXT) CALL RSC(XHT,DP,P)
      CALL AB2P(P,DP)
      CALL DELXH(XH,DXH,VN,XN,X,XHT)
      CALL AB2(XH,DXH)
      DO 28 LI=1,NS
      XHT(LI)=XH(LI) + XN(LI)
   28 XE(LI)=X(LI)-XHT(LI)
   55 CONTINUE
      RMSER=RMSER+XE(1)*XE(1)
      TIME=LN*HH
   30 CONTINUE
      SUMX1(M1)=SUMX1(M1)+X(1)-XHT(1)
      SUMX2(M1)=SUMX2(M1)+X(2)-XHT(2)
      SSQX1(M1)=SSQX1(M1)+(X(1)-XHT(1))*(X(1)-XHT(1))
      SSQX2(M1)=SSQX2(M1)+(X(2)-XHT(2))*(X(2)-XHT(2))
      GO TO 10
      WRITE(6,300) TIME
  300 FORMAT(1X,'TIME = ',F5.3/)
      DO 12 IQ=1,NS
   12 WRITE(6,400) X(IQ), XN(IQ), XH(IQ)   ,XHT(IQ),XE(IQ)
  400 FORMAT(29X,5(2X,F10.7))
      WRITE(6,200) ((P(I,J),J=1,NS),I=1,NS)
   10 CONTINUE
      RMSER=SQRT(RMSER/MTOT)
      WRITE(6,969) RMSER
  969 FORMAT (37X,F20.6)
```

```
  200 FORMAT(5X,2F12.7/5X,2F12.7/)
  502 CONTINUE
C
C     *****  CALCULATE SAMPLE MEAN AND VARIANCE  *****
C
      DO 40 M1=1,10
      XM1=MCR
      X1BAR(M1)=SUMX1(M1)/XM1
      X2BAR(M1)=SUMX2(M1)/XM1
      VARX1(M1)=SSQX1(M1)/(XM1-1.)-(XM1/(XM1-1.))*X1BAR(M1)*X1BAR(M1)
      VARX2(M1)=SSQX2(M1)/(XM1-1.)-(XM1/(XM1-1.))*X2BAR(M1)*X2BAR(M1)
      TIME=100.*HH
      WRITE(6,41) TIME,X1BAR(M1),VARX1(M1)
   40 WRITE(6,42) X2BAR(M1),VARX2(M1)
   41 FORMAT(10X,F6.3,2(5X,F15.8))
   42 FORMAT(16X,2(5X,F15.8))
      RETURN
      END
```

```
      SUBROUTINE JAC(A,XH)
      DIMENSION XH(4),A(4,4)
      COMMON B(4,2),K(4,2),H(2,4),R(2,2),HT(4,2),Q(2,2),KUTTA,BB,HH,V,G,
     *VN(2),MS,RS,KALF,NS,RI(4,4),ALPHA,AEXP,HF
      DO 10 I=1,NS
      DO 10 J=1,NS
   10 A(I,J)=0.
      A(1,1)= -2.0
      A(2,1)= 0.0
      A(2,2)= -1.0
      A(1,2)=1. +AEXP*ALPHA*ABS(XH(2))**(AEXP-1.)
      RETURN
      END
```

```
      SUBROUTINE RSC(X,DP,P)
      INTEGER RS
      DIMENSION AT(4,4),AP(4,4),PS(4,4),DP(4,4),PAT(4,4),XP(4,4),X(4),HP
     *(2,4),PHT(4,4),PHRHP(4,4),BT(2,4),P(4,4),BQ(4,2),BQBT(4,4),PHTRI(4
     *,2),A(4,4)
      COMMON B(4,2),K(4,2),H(2,4),R(2,2),HT(4,2),Q(2,2),KUTTA,BB,HH,V,G,
     *VN(2),MS,RS,KALF,NS,RI(4,4),ALPHA,AEXP,HF
      CALL JAC(A,X)
      DO 19 I=1,NS
      DO 19 J=1,NS
   19 AT(I,J) = A(J,I)
      DO 43 I=1,NS
      DO 43 J=1,NS
   43 PS(I,J)=P(I,J)
      DO 34 I=1,NS
      DO 34 J=1,NS
      AP(I,J)=0.
      DO 34 L=1,NS
   34 AP(I,J)=A(I,L)*PS(L,J) + AP(I,J)
      DO 35 I=1,NS
      DO 35 J=1,NS
      PAT(I,J)=0.
      DO 35 L=1,NS
   35 PAT(I,J)= PS(I,L)*AT(L,J) + PAT(I,J)
      DO 5 I=1,NS
      DO 5  J=1,MS
    5 HT(I,J) = H(J,I)
      DO 10 I=1,NS
      DO 10 J=1,MS
      PHT(I,J)=0.
      DO 10 L=1,NS
   10 PHT(I,J) = P(I,L)*HT(L,J) + PHT(I,J)
      DO 15  I=1,MS
      DO 15 J=1,NS
      HP(I,J)=0.
      DO 15 L=1,NS
   15 HP(I,J) = H(I,L)*P(L,J) + HP(I,J)
      DO 20 I=1,NS
      DO 20 J=1,MS
      PHTRI(I,J)=0.
      DO 20 L=1,MS
   20 PHTRI(I,J) = PHT(I,L)*RI(L,J) + PHTRI(I,J)
      DO 25 I=1,NS
      DO 25 J=1,NS
      PHRHP(I,J)=0.
      DO 25 L=1,MS
   25 PHRHP(I,J) = PHTRI(I,L)*HP(L,J) + PHRHP(I,J)
      DO 30 I=1,NS
      DO 30 J=1,RS
      BQ(I,J)=0.
      DO 30 L=1,RS
      BT(J,I) = B(I,J)
   30 BQ(I,J) = B(I,L)*Q(L,J) + BQ(I,J)
      DO 33 I=1,NS
```

```
      DO 33 J=1,NS
      BQBT(I,J)=0.
      DO 33 L=1,RS
   33 BQBT(I,J) = BQ(I,L)*BT(L,J) + BQBT(I,J)
      DO 36 I=1,NS
      DO 36 J=1,NS
   36 DP(I,J)=AP(I,J) + PAT(I,J) - PHRHP(I,J)    + BQBT(I,J)
      RETURN
      END
```

```
      SUBROUTINE RNG(SIGX1,SIGX2,XMEAN1,XMEAN2,X)
      COMMON/BLOCK2/IX,DUM
      DIMENSION X(2)
      IX=19971*IX
      IX=MOD(IX,1048576)
      U=ABS(IX*1.0/1048576.)
      Z1=SQRT(-2.0*ALOG(DUM))
      X(1)=Z1*COS(6.28318*U)*SIGX1 + XMEAN1
      X(2)=Z1*SIN(6.28318*U)*SIGX2 + XMEAN2
      DUM=U
      RETURN
      END
```

```
      SUBROUTINE AB2P(P,DP)
      DIMENSION P(4,4),DP(4,4),DPP(2,2),DXP(2)
      COMMON B(4,2),K(4,2),H(2,4),R(2,2),HT(4,2),Q(2,2),KUTTA,BB,HH,V,G,
     *VN(2),MS,RS,KALF,NS,RI(4,4),ALPHA,AEXP,HF
      COMMON/BLOCK3/DPP,DXP
      DO 10 I=1,NS
      DO 10 J=1,NS
      P(I,J)=P(I,J) + HF*(1.5*DP(I,J) - 0.5*DPP(I,J))
   10 DPP(I,J)=DP(I,J)
      RETURN
      END
```

```
      SUBROUTINE RK4( X1,DX1,X2,DX2)
      DIMENSION X1(4),DX1(4),X2(4),DX2(4),DXA(4),DXB(4),XA(4),XB(4)
      COMMON B(4,2),K(4,2),H(2,4),R(2,2),HT(4,2),Q(2,2),KUTTA,BB,HH,V,G,
     *VN(2),MS,RS,KALF,NS,RI(4,4),ALPHA,AEXP,HF
      GO TO(10,30,50,70), KUTTA
   10 DT= 0.5*HH
      DO 20 I=1,NS
      XA(I)=X1(I)
      DXA(I)=DX1(I)
      X1(I)=X1(I) + DT*DX1(I)
      XB(I)=X2(I)
      DXB(I)=DX2(I)
   20 X2(I)=X2(I) + DT*DX2(I)
      RETURN
   30 DO 40 I=1,NS
      DXA(I)=DXA(I) +  DX1(I) + DX1(I)
      X1(I)=XA(I) + DT*DX1(I)
      DXB(I)=DXB(I) +  DX2(I) + DX2(I)
   40 X2(I)=XB(I) + DT*DX2(I)
      RETURN
   50 DO 60 I=1,NS
      DXA(I)=DXA(I) +  DX1(I) + DX1(I)
      X1(I)= XA(I) + HH*DX1(I)
      DXB(I)=DXB(I) +  DX2(I) + DX2(I)
   60 X2(I)= XB(I) + HH*DX2(I)
      RETURN
   70 VH= HH*0.1666667
      DO 80 I=1,NS
      X1(I)=XA(I) + VH*(DXA(I) + DX1(I))
   80 X2(I)=XB(I) + VH*(DXB(I) + DX2(I))
      RETURN
      END
```

```
      SUBROUTINE AB2(X,DX)
      DIMENSION X(2),DX(2),DXP(2) ,DPP(2,2)
      COMMON B(4,2),K(4,2),H(2,4),R(2,2),HT(4,2),Q(2,2),KUTTA,BB,HH,V,G,
     *VN(2),MS,RS,KALF,NS,RI(4,4),ALPHA,AEXP,HF
      COMMON/BLOCK3/DPP,DXP
      DO 10 I=1,NS
      X(I)=X(I) + HF*(1.5*DX(I) - 0.5*DXP(I))
   10 DXP(I)=DX(I)
      RETURN
      END
```

```
      SUBROUTINE DELXH(XH,DXH,VM,XN,X,XHT)
      INTEGER RS,REG,EXT
      REAL K,KH,KVM
      DIMENSION XH(4),DXH(4),X(4),A(4,4),VM(2),KH(4,4),KVM(4),XN(4)
      DIMENSION XHT(4)
      COMMON B(4,2),K(4,2),H(2,4),R(2,2),HT(4,2),Q(2,2),KUTTA,BB,HH,V,G,
     *VN(2),MS,RS,KALF,NS,RI(4,4),ALPHA,AEXP,HF
      DATA REG,EXT/3HREG,3HEXT/
      DO 2 I=1,NS
      DO 2 J=1,NS
      KH(I,J)=0.
      DO 2 L=1,MS
    2 KH(I,J)=K(I,L)*H(L,J) + KH(I,J)
      DO 3 I=1,NS
      KVM(I)=0.
      DO 3 J=1,MS
    3 KVM(I)=KVM(I) + K(I,J)*VM(J)
      IF(KALF.EQ.REG) CALL JAC(A,XN)
      IF(KALF.EQ.EXT) CALL JAC(A,XHT)
      DO 4 I=1,NS
      DXH(I)=0.
      DO 4 J=1,NS
    4 DXH(I)=DXH(I) +(A(I,J)-KH(I,J))*XH(J) + KH(I,J)*(X(J)-XN(J))
      DO 5 L=1,NS
    5 DXH(L)= DXH(L) + KVM(L)
      RETURN
      END
```

```
      SUBROUTINE XEQN(XD,XMD,WNOISE)
      INTEGER RS
      DIMENSION XD(4),XMD(4),WNOISE(2),BW(4)
      COMMON B(4,2),K(4,2),H(2,4),R(2,2),HT(4,2),Q(2,2),KUTTA,BB,HH,V,G,
     *VN(2),MS,RS,KALF,NS,RI(4,4),ALPHA,AEXP,HF
      DO 10 I=1,NS
      BW(I)=0.
      DO 10 J=1,RS
   10 BW(I)=  BW(I)+ B(I,J)*WNOISE(J)
C
C     SYSTEM EQUATIONS
      XMD(1)=-2.*XD(1)+BW(1)+XD(2)+ALPHA*XD(2)*ABS(XD(2))**(AEXP-1.)
      XMD(2)=  -XD(2) + BW(2)
      RETURN
      END
```

VITA

Jerry Alton Gaston

Candidate for the Degree of

Master of Science

Thesis: DIGITAL SPEED AND ACCURACY TRADEOFFS IN CONTINUOUS KALMAN FILTERING FOR NONLINEAR SYSTEMS

Major Field: Electrical Engineering

Biographical:

Personal Data: Born in Woodward, Oklahoma, April 22, 1949, the son of Mr. and Mrs. John Gaston.

Education: Graduated from Woodward High School, Woodward Oklahoma, in May, 1967. Received Bachelor of Science in Electrical Engineering degree from Oklahoma State University, Stillwater, Oklahoma in July, 1971. Completed the requirements for Master of Science degree in December, 1973.

Professional Experience: Employed by Oklahoma State University as a graduate assistant from February, 1972 to August, 1972; employed by the Boeing Company, Wichita, Kansas, working in the Navigation and Weapon Delivery Department from August, 1972 to September, 1973.

Professional Organizations: Member of the Institute of Electrical and Electronics Engineers, Member of Triangle Fraternity.