CONFIDENCE INTERVAL ESTIMATION FOR OUTPUT OF DISCRETE-EVENT SIMULATIONS USING THE KALMAN FILTER

THESIS

Randall Barry Howard
Captain, USAF

AFIT GOR ENS 92M-15

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GOR/ENS/92M-15

CONFIDENCE INTERVAL ESTIMATION FOR OUTPUT OF DISCRETE-EVENT SIMULATIONS USING THE KALMAN FILTER

THESIS

Randall Barry Howard
Captain, USAF

AFIT/GOR/ENS/92M-15

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 1992	Master's Thesis

4. TITLE AND SUBTITLE
CONFIDENCE INTERVAL ESTIMATION FOR OUTPUT OF DISCRETE-EVENT SIMULATIONS USING THE KALMAN FILTER

5. FUNDING NUMBERS

6. AUTHOR(S)
Randall B. Howard, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GOR/ENS/92M-15

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)
Discrete-event simulation is computer modeling of stochastic, dynamic systems. The Kalman filter is a Bayesian stochastic estimation algorithm. Because of the correlated nature of simulation output, it is difficult to apply the methods of classical statistics directly when constructing confidence intervals of discrete-event simulation parameters. Through the determination of a dynamics equation and application of the Kalman filter to simulation output data, three new confidence interval construction techniques have been developed. One technique obtains an estimate of the mean value and its associated variance from an estimated Kalman filter. The second technique utilizes Multiple Model Adaptive Estimation (MMAE) techniques to obtain an estimate of the simulation output's mean value and its associated variance. The third technique also uses MMAE, but constructs a nonsymmetric confidence interval using the final MMAE filter probabilities.

The purpose of this research was twofold. The first objective was to explore these new confidence interval construction techniques based on the information provided by Kalman filters. The second objective was to contrast these Kalman filter approaches to several accepted approaches. Both of these objectives were achieved and excellent results were obtained. In particular, a Monte Carlo analysis demonstrated that the third technique produced intervals that achieved nominal coverage rates with, when compared to currently accepted techniques, smaller average half widths and lower variability.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Confidence-Intervals; Kalman Filters; Discrete-Event Simulation; Statistical Analysis; Computer Simulation	149
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

AFIT/GOR/ENS/92M-15

CONFIDENCE INTERVAL ESTIMATION FOR OUTPUT OF DISCRETE-EVENT SIMULATIONS USING THE KALMAN FILTER

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Operations Research

Randall Barry Howard, B.S.

Captain, USAF

March, 1992

THESIS APPROVAL

STUDENT: Captain Randall B. Howard CLASS: GOR 92-M

THESIS TITLE:

Confidence Interval Estimation
for Output of Discrete-Event Simulation
Using the Kalman Filter

DEFENSE DATE: 4 MAR 92

GRADE:

COMMITTEE:	NAME/DEPARTMENT	SIGNATURE
Advisor	Major Kenneth W. Bauer/ENS	Kenneth W. Bauer, Jr.
Reader	Doctor Peter S. Maybeck/ENG	Peter S. Maybeck

Preface

The purpose of this research has been to explore new methodologies for constructing confidence intervals based on the information provided by Kalman filters and to compare and contrast these methods to those currently used. The use of Kalman filters in the area of simulation output analysis is a new idea, and this effort has shown there is a great potential for its use. Both of the initial objectives of this study were met and several areas have been identified for follow-on studies.

This effort would not have been successful without the help of many people. First, I would like to thank Major Kenneth Bauer, my advisor, for his constant support and encouragement in this and other endeavors. I would also like to thank Dr. Peter Maybeck whose knowledge of the Kalman filter turned an extremely complex area of study into an understandable and enjoyable topic. Also, I would like to thank the many instructors I have had at the Air Force Institute of Technology for the various insights and contributions they have given me.

Captain Mark Gallagher deserves much recognition for the endless hours of support and guidance he has given me. Without him there were many hurdles I would never have passed. Finally, I want to thank my wonderful wife, Karen, who has supported me throughout this entire program. We have both survived the first year of marriage during a time when I was more often a student than a husband.

Randall Barry Howard

Table of Contents

List of Figures

List of Tables

Abstract

Discrete-event simulation is computer modeling of stochastic, dynamic systems. The Kalman filter is a Bayesian stochastic estimation algorithm. Because of the correlated nature of simulation output, it is difficult to apply the methods of classical statistics directly when constructing confidence intervals of discrete-event simulation parameters. Through the determination of a dynamics equation and application of the Kalman filter to simulation output data, three new confidence interval construction techniques have been developed. One technique obtains an estimate of the mean value and its associated variance from an estimated "optimal" Kalman filter. The second technique utilizes Multiple Model Adaptive Estimation techniques to obtain an estimate of the simulation output's mean value and its associated variance. The third technique also uses MMAE, but constructs a nonsymmetric confidence interval using the final MMAE filter probabilities.

The purpose of this research was twofold. The first objective was to explore these new confidence interval construction techniques based on the information provided by Kalman filters. The second objective was to contrast these Kalman filter approaches to several accepted techniques for confidence interval construction. Both of these objectives were achieved and excellent results were obtained. In particular, a Monte Carlo analysis demonstrated that the third technique produced intervals that achieved nominal coverage rates with, when compared to currently accepted techniques, smaller average half widths and lower variability.

CONFIDENCE INTERVAL ESTIMATION FOR OUTPUT OF DISCRETE-EVENT SIMULATIONS USING THE KALMAN FILTER

I. Introduction

Background

Frequently the Air Force needs to estimate various parameters associated with a system. These systems can include aerial refueling missions, customer lines at base commissaries, and staffing at medical center emergency rooms. The systems are often extremely complex and not well modeled by analytical formulas. However, computer simulation can often be used to analyze these systems [29:6]. Pritsker defines computer simulation as "the process of designing a mathematical-logical model of a real system and experimenting with this model on a computer" [38:6].

Pritsker [38:6] states the primary advantage of computer simulation is that it allows us to make inferences about a system without building it, or disturbing it, or destroying it. Thus, when compared to actual experimentation with a system, simulation is inexpensive, unobtrusive, and easy to implement. The Air Force, faced with decreasing budgets, must rely increasingly on simulation to analyze its systems objectively. Therefore, the Defense Science Board recently recommended using more simulations to analyze weapon system development and procurement systems [14:40]. In addition, the Department of Defense has included "Simulation and Modeling" in the twenty critical technologies for ensuring the long-term qualitative superiority of United State's weapon systems [7].

The primary purpose of the computer simulation is to allow the analyst to estimate various output parameters [45:268]. For example, the parameters may include the average number of minutes a customer will spend in line or the average number of aircraft a tanker can refuel. Because of the psuedo-random nature of the stochastic simulations, the parameters can not be exactly determined and therefore, must be estimated.

Law and Kelton state, "One of the most important but difficult problems encountered in a real-world simulation study is that of constructing a confidence interval (c.i.) for the steady state mean ..." [28:1221]. A confidence interval is a numeric interval that has a stated $(1-\alpha)$ confidence level. The confidence level is a probabilistic statement, such that, if one constructs a large number of confidence intervals, he can expect $(1-\alpha)$ percent of them to contain the true parameter. Most methods of constructing confidence intervals require an estimate of a value and its corresponding variance, and many analysts do not put enough effort into estimating these values [29:522]. Several techniques for estimating parameters and associated confidence intervals of computer simulation output data exist [29, 28, 45]. Conclusive evidence on which, if any, of the existing methods is the most accurate is hard to find. In addition, Kalman filters may provide information useful in constructing accurate confidence intervals.

According to Maybeck, the Kalman filter is "...an optimal recursive data processing algorithm" [30:4]. It can also be defined as a recursive mathematical technique that allows one to make inferences about the state of a stochastic process based on noise-corrupted measurements [30:3–7]. A Kalman filter state is the set of variables that characterize the relevant parameters of a system. Meinhold and Singpurwalla [36:123] observe the Kalman filter is mainly used in engineering applications and give examples of its use in aerospace tracking and underwater sonar. The techniques used in Kalman filters provide valuable information that may be exploited in the area of operations research, specifically, in the area of simulation output analysis.

Purpose

The purpose of this research was twofold. The first objective was to explore confidence interval construction methods based on the information provided by Kalman filters. The second objective was to compare and contrast these potential methods with several published techniques for confidence interval construction.

Scope of the Study

In operations research, simulations can be characterized in several ways. These characterizations include the following:

1. deterministic versus stochastic
2. static versus dynamic
3. continuous and/or discrete

Deterministic simulations have known input variables, while stochastic simulations have input variables that are functions of random inputs. Stochastic simulations can be broken into static and dynamic simulations. A simulation is described by a collection of variables known as the system state. In static simulations this state does not change. Examples of static stochastic simulations are distribution sampling and classic Monte Carlo techniques [46]. In dynamic simulations the state variables change over time. For example, one state variable of a commissary line simulation describes if the cashier is busy or idle, and this variable will obviously change over time. Before discussing discrete-event and continuous event simulations, it is necessary to define events. In simulation, an event occurs when a state changes. In a discrete-event simulation, events only occur at specific points in time [38:52], such as customer arrival or service completion. In continuous-event simulations, events occur continuously throughout simulated time [38:52]. This research dealt with simulation models which are stochastic, dynamic, and characterized by discrete events (generally called "discrete-event simulation").

Discrete-event simulations also can be characterized with regard to analysis of their output. The two different types are terminating and nonterminating (steady-state) simulations. In terminating simulations, the length of the simulation run is quantitatively defined. For example, one can estimate the number of customers in a store at the end of an eight hour day. In a nonterminating simulation one wants to estimate a parameter, which will reach a steady-state value, as the simulation length approaches infinity [26:89–90]. These nonterminating simulation parameters can be steady-state cyclic (following a periodic probability distribution) or constant steady-state (where the parameter attains a stationary probability distribution). This research focused on techniques for estimating parameters of nonterminating simulations that reach stationary steady-state probability distributions.

The analysis of nonterminating simulations involves the determination of the duration period of a transient phase of the simulation. The transient phase is the part of a simulation necessary for a system to move from the initial starting condition to effective steady-state conditions [38:43].

The output of the transient phase is not representative of the steady-state output and can bias the estimation of steady-state parameters. This research eliminated the problem of transient conditions by truncating a large part of the initial data.

According to Pritsker there are two distinct types of parameters that can be estimated from simulation output; he calls these cases "statistics based upon observations" and "statistics based on time-persistent variables" [38:36]. The first type, "statistics based upon observations", deal with those parameters associated with particular entities in the simulation. For example, one might be interested in how long a plane has to wait for aerial refueling. Values for this type of variable can only be recorded at certain points in the simulation (e.g., when a plane quits waiting and begins to refuel) and the analyst is generally interested in the average value for all the observations. The second type, "statistics on time-persistent variables", deal with variables that have a value throughout the simulation. For example, one might be interested in the number of planes waiting for refueling. The value of these parameters can change at any time, and one is interested in determining the average value over time. This research considered both types of parameters.

Objectives

To reach the goals of this effort several objectives had to be met. First, methods had to be developed to compute confidence intervals or similar intervals based on the use of Kalman filters. Computer routines were developed to process simulation output through Kalman filters. This software provided the necessary information for constructing confidence intervals. Application of the Kalman filter provides a vast amount of information about the systems and the parameters of interest. Specifically, a state-space representation, state estimates, the state estimate covariances, residuals, and computed residual covariances are available (3:9). By relating this information to the variance of the parameter estimates from the simulation output, new techniques for constructing confidence intervals were developed.

The next objective was to test these techniques. Computer simulation models of time series and queueing models with known analytical means were programmed. Computer routines were developed to compute confidence intervals based on the Kalman filter techniques and four widely applied techniques (nonoverlapping batch means, overlapping batch means, standardized

time series, and autoregressive).

The final objective was to compare the various methods of confidence interval estimation. A Monte Carlo analysis was conducted in which each method was used to calculate a large number of independent ninety-percent confidence intervals for the chosen parameter. The methods were compared and contrasted using the resulting actual coverage rates, interval half widths, and interval half-width variance.

Overview of Chapters

The following chapters will cover the thesis effort in detail. Chapter II provides a review of the following three areas: 1) the statistics involved in constructing confidence intervals, 2) currently applied construction techniques, and 3) performance criteria for confidence intervals. Chapter III provides an overview of Kalman filters and the associated information they provide. This chapter is a modified version of a research proposal submitted to the Air Force Office of Scientific Research [3].

Chapter IV develops the methodology of the proposed confidence interval construction approaches. Chapter V provides the Monte Carlo results and discusses their implications. Finally, Chapter VI provides a summary of the final conclusions drawn from this research and offers recommendations for further studies.

II. Confidence Intervals

The literature required to support this research includes publications in both the areas of confidence intervals and Kalman filters. This chapter will concentrate on confidence intervals. Specifically, the review will explore the following areas in relation to confidence intervals: 1) independent random variable statistics, 2) steady-state parameters in discrete-event simulations, 3) performance measures for confidence intervals, and 4) current construction procedures. The following chapter will concentrate on the Kalman filter.

Independent Random Variable Statistics

In classical statistics confidence intervals are easily calculated from estimates of the parameter and its associated variance. Following the development of Law [26:985–986], $Y_1, Y_2, \ldots, Y_n$ are n independent identically-distributed random variables with a population mean of μ and a population variance of σ^2. The sample mean

$$\bar{Y}(n) = \sum_{i=1}^{n} \frac{Y_i}{n}$$

is an unbiased point estimator of the population mean μ (i.e., $E[\bar{Y}(n) - \mu] = 0$). The sample variance

$$s^2(n) = \sum_{i=1}^{n} \frac{(Y_i - \bar{Y}(n))^2}{n-1} \tag{1}$$

is an unbiased estimator for σ^2 (i.e., $E[s^2(n)] = \sigma^2$). The variance σ^2 is a measure of how spread the individual Y_i's are from the point estimate $\bar{Y}(n)$. The point estimate $\bar{Y}(n)$ will have an associated variance $\sigma^2_{\bar{Y}(n)}$, which is a measure of the spread in mean estimates about the true mean. Since the Y_i's are independent, the variance of the sample mean is calculated as $\sigma^2_{\bar{Y}(n)} = \sigma^2/n$. Using Equation (1) above, an unbiased estimator for the variance of $\bar{Y}(n)$ is

$$\hat{\sigma}^2_{\bar{Y}(n)} = s^2(n)/n$$

If the Y_i's are normal random variables, then

$$\sqrt{n}\frac{\bar{Y}(n) - \mu}{s}$$

has a t distribution with $(n-1)$ degrees of freedom and

$$\bar{Y}(n) \pm t_{n-1,1-\alpha/2}\sqrt{s^2(n)/n} \tag{2}$$

where $t_{n-1,1-\alpha/2}$ is the upper $1-\alpha/2$ critical value for a t distribution with $n-1$ degrees of freedom, gives a $100(1-\alpha)\%$ confidence interval for μ.

Law gives an excellent interpretation of this confidence interval, "If one constructs a very large (an infinite) number of $100(1-\alpha)\%$ confidence intervals each based on n observations, the proportion of these confidence intervals that contain (cover) μ will be $1-\alpha$ and is called the *coverage* for the confidence interval" [26:986]. He then goes on to say that in actual practice the variables (Y_i's) will not be normal and the actual coverage may be less than the stated (nominal) $1-\alpha$ coverage. However, if n is large enough, the central limit theorem will ensure that the actual coverage will approach the nominal coverage [26:986].

Steady-State Parameters for Discrete-Event Simulation Data

Constructing confidence intervals for steady-state values of discrete-event simulation parameters is one of the primary objectives of this research. According to Heidelberger and Welch:

> In generating confidence intervals for steady-state characteristics there are two fundamental problems: (1) There is a transient phase during which the characteristics of the output sequence do not approximate the steady-state characteristics. (2) The output sequence is, in general, correlated and hence standard statistical techniques based on uncorrelated observations do not directly apply. [16:233]

There are several proposed techniques for estimating parameters and associated confidence intervals from a sequence of simulation output $\{y_1, y_2, \ldots, y_n\}$. All of these techniques require the deletion of the initial transient data. There are several truncation methods for deleting this transient data [29]. Based on different assumptions and techniques, each confidence interval method deals with the correlated nature of the output data differently. Law [26:997] explains that the methods can be placed into two main categories: fixed-sample size procedures and sequential procedures. Fixed-sample size procedures assume the analyst must compute a confidence interval with a given amount of data (either in one long run or several shorter runs). Sequential procedures allow the analyst to

change the length of a simulation run until obtaining a satisfactory confidence interval. This study is limited to fixed-sample size procedures.

The literature suggests five prominent methods based on fixed-sample size and a single long simulation run [26, 27, 20, 25, 28, 41]. The five procedures are 1) batch means, 2) overlapping batch means, 3) autoregressive time series, 4) standardized time series, and 5) spectral analysis.

Construction Techniques

This section provides a brief overview of five methods for confidence interval construction. Each of these techniques is designed for one long simulation run and assumes the initial transient is deleted. The various methods result from different approaches to account for the correlation in the output sequence.

Batch Means. To alleviate the problem of autocorrelation, the method of batch means makes one single run of m observations and divides the run into n "batches" each of k sequential observations. This method relies on an assumption that the process is covariance stationary [29:554–555]. Covariance stationarity means that all of the observations have the same expected value and all of the covariances between any two random variables separated by a set interval of time are equal [38:100]. Law and Carson [27] have demonstrated that, in general, for a large enough batch size this assumption holds true.

Following Welch's [45:307] discussion, most simulation models have positive autocorrelation that decreases as the lag between the random variables increases. Therefore, there exists a lag amount L such that correlations greater than L are essentially zero. Welch states, "If we partition this steady-state sequence up into contiguous subsequences of length $N \gg L$ then estimates derived from these subsequences can be considered to be approximately independent" [45:307].

The sample (batch) mean for the k observations in the nth batch is $\bar{y}_n(k)$. The grand mean can be found by averaging all of the batch means. Since the batch means are uncorrelated, the sample variance can be found using Equation (1), and an appropriate confidence interval can then be computed with Equation (2).

The primary advantages of this technique are its efficient use of the available data and its simplicity. However, the selection of batch size (and hence number of batches) is very difficult. Law and Kelton [28:1224] state the batch size must be large enough to satisfy the normality and independence assumptions. If the batch size is too small, the means of sequential batches will be correlated. Law and Kelton go on to say that correlated batch-means result in underestimating the variance of the mean estimate. This potential bias in the estimation of the variance causes the most errors in application of this method.

Batches which are too large cause a loss of information. The loss of information manifests itself in large confidence intervals with high variance [29:555]. Welch [45:309] recommends the subsequence length be $k \geq 5L$, but the choice of L is also difficult. Welch recommends estimation using an autocorrelation function to determine L. Fishman developed a procedure to test for independence between batches. Using the Von Neuman statistic, the batch size is iteratively increased until the test of independence of batches is passed. Fishman [9:514–515] recommended at least eight batches because the statistical test used to detect correlation has little power with fewer than eight batches. Schmeiser [40:560] found that fewer than ten batches resulted in very unstable intervals, and additional batches beyond thirty yield little improvement in the performance of the confidence intervals.

Overlapping Batch Means. Meketon and Schmeiser [37] introduced the method of overlapping batch means. Their method is very similar to the method of batch means, but they contend that batch size is more important than independence between batches. Therefore, they allow the batches to overlap and estimate variance using all $m - k + 1$ batch means of size k. Their paper shows that overlapping batch-means produce confidence intervals with 2/3 the variance of those constructed with nonoverlapping batch means. Kang and Goldsman [20:16] state that the method of overlapping batch means produces confidence intervals with smaller average half widths than the regular batch means method. After estimating the mean estimates variance, the computation of confidence intervals is similar to that of nonoverlapping batch means.

Autoregressive Time Series. The method of autoregressive time-series attempts to use the autocorrelation structure of the simulation process to determine a confidence interval [26:999].

Law and Kelton state, "the autoregressive method, developed by Fishman, assumes the process is covariance stationary and can be represented by a pth order autoregressive model" [28:1224]. An autoregressive (AR) model is a model in which the state at time n is a function of the state at one or more previous times plus a random error term [39:465]. For example, in a first-order AR process, the state at time n is a constant multiple of the state at time $n-1$ plus a random error term. Fishman's method determines a pth order, AR(p), model that "fits" the data, and the AR model provides the variance estimate of the parameter [10:247–252]. A confidence interval is constructed using the mean value of the simulation output and the calculated variance. Law [26:1000] gives a detailed description of this method.

Schriber and Andrews [41:348] generalized the methodology to allow for a moving-average component along with the autoregressive factors. The moving-average (MA) component of an autoregressive-moving average (ARMA) process allows the state of the process to depend on the past error terms (ϵ_n). For example, a second-order ARMA process is a model where the state at time n is $x_n = \phi_1 x_{n-1} + \phi_2 x_{n-2} + \theta_0 + \epsilon_n - \theta_1 \epsilon_{n-1} - \theta_2 \epsilon_{n-2}$. The ϕ_i's are the autoregressive coefficients, and the θ_i's are the moving-average coefficients.

Law and Kelton [28:1225] believe the major problem with this method is the validity of an AR or ARMA model for an arbitrary stochastic process. They state that, if the model is incorrect (e.g., the process is not covariance stationary or the wrong model is fit), the associated confidence intervals may have smaller coverages than desired. Pritsker [38:740] states that this technique has not produced reliable estimates of the mean's variance, probably because of non-stationary behavior of the time series or a non-normality of the individual observations. Another drawback of this method is that it requires an advanced understanding of time series.

Standardized Time Series. Schruben [42], in 1983, introduced a novel application of standardized time series models to simulation output data. Schruben's methodology assumes the process is strictly stationary and is a phi-mixing process. Law explains that "strictly stationary means that the joint distribution of $Y_{i_1+j}, Y_{i_2+j}, \ldots, Y_{i_n+j}$ is independent of j for all time indices $i_1, i_2, \ldots, i_n$" [26:1003]. Phi-mixing means that Y_i and Y_{i+j} become independent as j becomes large [26:1003]. This method separates the run into batches, exactly as in the method of batch means, and calculates individual batch means and a grand mean ($\bar{y}$) in the same way. Following the development of

Law:

> ...make one simulation run of length m and divide $Y_1, Y_2, \ldots, Y_m$ into n batches of size $k(m = nk)$. Let $\bar{Y}_j(k)$ be the sample mean of the k observations in the jth batch. The grand sample mean $\bar{Y}(m)$ is the point estimator for ν. Furthermore, if m is large, then $\bar{Y}(m)$, will be approximately normally distributed with mean ν and variance τ^2/m, where $\tau^2 = lim_{m\to\infty} mVar[\bar{Y}(m)]$. Let $A = [12/(k^3 - k)] \sum_{i=1}^{k} \{\sum_{i=1}^{l} [\bar{Y}_j(k) - Y_{i+(j-1)k}]\}^2$. For a fixed number of batches n, A will be asymptotically (as $k \to \infty$) distributed as τ^2 times a chi-square random variable with n degrees of freedom and asymptotically independent of $\bar{Y}(m)$. [26:1003]

Using the above definitions, a t distribution can be found and a confidence interval constructed using [26:1003]:

$$\bar{Y}(m) \pm t_{n,1-\alpha/2}\sqrt{A/mn}$$

The major advantage to this method is its computational ease [26:1004]. The major difficulty with this method is selecting the batch size. As in the method of batch means, there is a bias in the estimation of the variance. However, the method of standardized time series reduced the variance of estimates for simulation data when compared to batch means [5:208]. Glynn and Iglehart [12] have had good success in working with this method.

Spectral Analysis. The method of spectral analysis estimates the autocorrelation structure of simulation data and obtains an estimate of the variance of the sample mean based on the structure [26:1000]. The literature suggests two prominent variants of the spectral method developed by Heidelberger and Welch [16] and Fishman [10]. Both methods are based in the frequency domain rather than the time domain [16:233]. Law and Kelton [29:556–557] summarize the general logic behind both of these methods. They state that the method assumes the process is covariance stationary with mean μ. Under this assumption the variance can be mathematically estimated in a fashion similar to that of the autoregressive method. The method of spectral analysis estimates the variance using $Var[\bar{Y}(m)] = 2\pi g(0)$ as $m \to \infty$, where $g(\tau)$ is the spectrum of the process at frequency τ, and is defined by a Fourier transform. After estimating the variance, a confidence interval is constructed using Equation (2) where the sample mean estimates $\bar{Y}(n)$.

The method of spectral analysis has the advantage of having only the assumption of covariance stationarity [16:233]. Law and Kelton [26:1001] found that the spectral method is very complicated and sometimes produces confidence intervals with low actual coverage.

In summary, these five confidence interval techniques have varying approaches for dealing with the correlated output sequence of simulation output. These approaches range from approximating independence to estimating the correlation in either the time or frequency domain. Yet, none of these techniques constructs confidence intervals in a consistent and satisfactorily manner. In order to compare confidence intervals constructed with these and other techniques, it is necessary to review the various performance measures used to evaluate confidence intervals.

Performance Measures

The literature suggests many ways to evaluate and compare confidence intervals. The most widely accepted criteria are coverage frequency of the interval, expected width of the interval, and variance of the interval width [41, 13, 40, 21, 5, 20, 47, 26]. These measures are conflicting in nature. Therefore, if a researcher attempts to improve one measure, the action he takes might have a detrimental effect on another measure. It is the researcher's task to find a good compromise between the measures. To evaluate the actual measures of performance for a given method, the researcher applies a Monte Carlo strategy. A Monte Carlo strategy constructs a large number of independent confidence intervals for a parameter with an analytically known value, and then calculates the various measures.

Coverage Frequency. Schruben [42:1101] describes the coverage frequency as the percentage of confidence intervals that actually include the true value of the parameter. He goes on to say that the analyst would prefer the observed frequency (actual coverage) to be equal to the stated coverage of $(1 - \alpha)$ percent. Kang and Schmeiser [21:546] state that coverage is the primary criterion for evaluating confidence intervals. For a fixed number of observations n, it is desirable to have the actual coverage be as close to $1 - \alpha$ as possible. To calculate the coverage frequency, the analyst determines the percentage of times the individual confidence intervals actually contain the true value of the parameter. This measure of effectiveness also can be expressed as a probability of coverage [5:203].

Expected Width. Chen and Sargent point out that a wide confidence interval will have a better probability of covering the true value of a parameter. For example, they point out that the interval $0 \leq \sigma^2 \leq \infty$ will always contain the true value of σ^2, yet this interval tells the analyst nothing about the value that he did not already know (i.e., it is a positive value). Therefore, Chen and Sargent state, "the expected width of the interval is the most used measure" [5:203]. The expected width is the length of the confidence interval. Law [26:986] states that the interval half width (length/2) is another method of reporting this measure. He agrees with Chen and Sargent that in choosing between two confidence intervals with the same probability of coverage, the analyst prefers the one with the smallest half width because it offers more precision.

Variance of the Interval Width. A measure proportional to the variance of the interval width is the standard deviation of the individual interval half widths [42:1101]. The standard deviation gives a measure of the stability of the confidence interval [5:205]. Kang and Schmeiser [21:546] state that if a confidence interval technique results in intervals with a large width variance, its accuracy will be subject to frequent false indications. Methods subject to false indications produce some intervals that have low coverage and some that have high coverage. These false indications make it hard for the researcher to determine the true accuracy of a given interval. Therefore, the researcher prefers confidence-interval techniques that provide interval widths with small variances.

Summary

The output sequence from discrete-event simulations is usually correlated and standard statistical techniques can not be used to construct confidence intervals. Numerous methodologies have been proposed for constructing confidence intervals from a sequence of simulation output. Five of the most widely-used techniques are nonoverlapping batch means, overlapping batch means, autoregressive, standardized time series, and spectral analysis.

There are three main performance measures used to evaluate various confidence interval construction techniques. These performance measures are actual coverage rate, average half width, and standard deviation of the half widths. In general, one wants a technique that produces confidence intervals with 1) an actual coverage rate equal to, or near, the nominal coverage rate, 2) a small average half width, and 3) a low standard deviation of the half widths. It is hard to find conclusive

evidence on which, if any, of the available techniques produce the best confidence intervals for simulation output.

The next chapter discusses the foundations of the Kalman filter. Chapter IV then shows how Kalman filters may be used to construct meaningful confidence intervals for output of discrete-event simulations.

III. Kalman Filters

Kalman [18] and Kalman and Bucy [19] developed the Kalman filter based partially on the work of the German mathematician Gauss. Kalman's contribution was to combine the work Gauss extended with the state-space representation in linear system theory. Akaike [2] defines a state as "a condensed representation of information from the present and past, such that the future behavior of the system can completely be described by the knowledge of the present state and the future input." Since the Markov property means that some set of sufficient statistics at the current time provides as much information about the system as all the historical statistics, this definition of state-space representation is limited to Markov processes. The original derivation was based on the fact that the state estimate at time t_i based on all measurements through that time, $\hat{x}(t_i^+)$, is the orthogonal projection of the true state $x(t_i)$ onto the subspace spanned by the measurement history $Z(t_i)$ composed of measurements $z(t_1), z(t_2), \ldots, z(t_i)$ at each sample time $t_1, t_2, \ldots, t_i$ [30:235].

The Kalman filter is an optimal, recursive, next-step prediction algorithm. After initialization, the discrete Kalman filter continues with a two-step procedure. The first step propagates the best estimate of the state estimate through time. The second step uses the information contained in a noise-corrupted measurement to update the prediction. The algorithm repeats again by propagating this estimate to the next time interval. These steps of propagation and update are repeated for each of the available observations. Before showing the Kalman filter formulas, the underlying theory of stochastic processes will be discussed.

This effort deals only with univariate simulation output sequences. Therefore, the dynamics noise and its associated variance, the measurement noise and its associated variance, and the measurements are all scalars. The notation throughout this and subsequent chapters reflects this univariate case, but the Kalman filter concepts can easily be extended to the multivariate case. This entire chapter follows Maybeck's development of the Kalman filter [30, 31, 32].

Stochastic Processes

Simulation output is a discrete realization of a stochastic process. In this section, the stochastic process will be described or approximated by a linear stochastic difference equation. The stochastic process is defined in terms of the states that are estimated by the Kalman filter, which

may or may not be the simulation states or the simulation observations. The estimation of the stochastic difference equation parameters is dependent upon the filter-design states selected for the application.

Assume the underlying stochastic process can be described or approximated with a linear stochastic difference equation,

$$\mathbf{x}(t_{i+1}) = \mathbf{\Phi}(t_{i+1}, t_i)\mathbf{x}(t_i) + \mathbf{G}(t_i)w_d(t_i) \tag{3}$$

where $\mathbf{x}$ is the state vector

$\mathbf{\Phi}$ is the state transition matrix

$\mathbf{G}$ is the dynamics noise dispersion matrix

w_d is the dynamics noise

The discrete-time dynamics driving noise $w_d(t_i)$ is normally distributed with zero-mean and the following characteristics:

$$E[w_d(t_i)] = 0$$

$$E[w_d(t_i)w_d(t_j)] = Q_d(t_i) \text{ for } t_i = t_j$$

$$E[w_d(t_i)w_d(t_j)] = 0 \text{ for } t_i \neq t_j$$

The dynamics noise sequence $\{w_d\}$ is uncorrelated in time, or white.

Assuming equally-spaced samples and a time-invariant system model, the transition matrix is constant (i.e., $\mathbf{\Phi}(t_{i+1}, t_i) = \mathbf{\Phi}$). Fishman suggests that in simulation output applications, equally spaced can be in terms of simulated time or an index [8:282]. For example, customers through a queuing system could be numbered sequentially as an index. The transition matrix then represents the relationship between subsequent entities. In addition, since the dynamics of steady-state simulation output is being modeled, the covariance matrix $Q_d(t_i)$ for the discrete-time white noise sequence $w_d(t_i)$ and the distribution matrix $\mathbf{G}(t_i)$ are assumed to be constant matrices, Q_d and $\mathbf{G}$ respectively.

In addition, in order to apply the Kalman filter, measurements of the system must be available. In these applications, the simulation output can be used as discrete-time measurements of the filter-design system states [11]. The measurement model indicates the relationship between the

filter-design states and the measurements:

$$z(t_i) = \mathbf{H}(t_i)\mathbf{x}(t_i) + v(t_i) \tag{4}$$

where $z = y - \mu_y$ is the Kalman filter measurement

y is the simulation output

$\mathbf{H}$ is the measurement matrix

$\mathbf{x}$ is the vector of filter-design system states

v is the measurement noise

Assume that measurement noise sequence $\{v(t_i)\}$ is white, normally-distributed zero-mean stochastic sequence, such that,

$$E[v(t_i)] = 0$$
$$E[v(t_i)v(t_j)] = R(t_i) \text{ for } t_i = t_j$$
$$E[v(t_i)v(t_j)] = 0 \text{ for } t_i \neq t_j$$

The dynamics noise sequence $\{w_d(t_i)\}$ and the measurement noise sequence $\{v(t_i)\}$ are assumed to be uncorrelated.

If the measurement matrix $\mathbf{H}(t_i)$ is not a function of the states $\mathbf{x}(t)$, then the measurement equation is linear. In addition, for steady-state simulation output sequences, the measurement matrix $\mathbf{H}(t_i)$ and the variance of the measurement noise process $R(t_i)$ are assumed not to change with time. Therefore, these time-dependent matrices will be replaced with constant matrices, $\mathbf{H}(t_i) = \mathbf{H}$ and $R(t_i) = R$.

Linear Gaussian Time-Invariant Kalman Filter Algorithm

The discrete-time Kalman filter algorithm is shown below for a time-invariant, linear system with no control inputs, and normally-distributed (Gaussian) zero-mean discrete dynamics noise and measurement noise [30:275]. Discrete-time implies that the propagation and measurement updates occur only at set intervals, and linear implies that the values of the filter-design system states $\mathbf{x}(t_i)$ do not affect the values of the transition matrix $\mathbf{\Phi}$, dynamics noise dispersion matrix $\mathbf{G}$, or the

measurement matrix $\mathbf{H}$. Time-invariant means that the filter-design matrices $\mathbf{\Phi}$, $\mathbf{G}$, and $\mathbf{H}$ do not change throughout the stochastic process. Due to process stationarity, the covariance matrices of the noises Q_d and R do not change throughout the stochastic process.

Before showing the two stages of the Kalman filter algorithm, the notation, which again follows Maybeck [30, 31, 32], must be explained. Estimates are indicated by "hat" over the variable, such as, Kalman filter state estimates $\hat{\mathbf{x}}$. However, at each point in time, two estimates of the state and associated covariance matrices are encountered. The first is the estimate based on the propagated output of the dynamics equation, before the measurement information is incorporated at that sample time. These estimates from the propagation stage are labeled with a superscript minus sign $\hat{\mathbf{x}}(t_i^-)$. In contrast, the state estimate resulting from the measurement update step are labeled with a superscript plus sign $\hat{\mathbf{x}}(t_i^+)$. The associated covariance matrices have similar notation with $\mathbf{P}(t_i^-)$ and $\mathbf{P}(t_i^+)$ respectively.

Propagation Stage. Two equations comprise the propagation stage. The first formula determines the propagation through time of the estimated state vector. The second equation propagates the covariance matrix of the state variables through time. The propagation equation takes the best state estimate at the previous time $\hat{\mathbf{x}}(t_{i-1}^+)$ (or the initial estimate $\hat{\mathbf{x}}(t_0)$), and moves it through time by multipling by the transition matrix $\mathbf{\Phi}$. Therefore, the propagation equation is

$$\hat{\mathbf{x}}(t_i^-) = \mathbf{\Phi}\hat{\mathbf{x}}(t_{i-1}^+) \tag{5}$$

where $\hat{\mathbf{x}}$ is the estimated state vector

$\mathbf{\Phi}$ is the state transition matrix

This estimate results from the conditional expectation of Equation (3), conditioned on measurements through time t_{i-1}, since the dynamics driving noise $w_d(t_i)$ has a mean of zero.

The associated covariance matrix $\mathbf{P}(t_i^-)$ is calculated with the following formula:

$$\mathbf{P}(t_i^-) = \mathbf{\Phi}\mathbf{P}(t_{i-1}^+)\mathbf{\Phi}^T + \mathbf{G}Q_d\mathbf{G}^T \tag{6}$$

where $\mathbf{P}$ is the covariance matrix of $\hat{\mathbf{x}}$

Q_d is the variance of the discrete dynamics driving noise

$\mathbf{G}$ is the dynamics noise dispersion matrix

These two equations, one for the state estimate and the other for the associated covariance, complete the propagation stage.

Measurement Update Stage. After the propagation stage, the second step is the measurement update. The measurement update is actually a static estimation problem of combining two terms (a state estimate and a correction term $\mathbf{K}(t_i)[z_i - \mathbf{H}\hat{\mathbf{x}}(t_i^-)]$. The state estimate resulted from Equation (5) in the propagation stage. The correction to that state estimate is based upon the actual measurement and the measurement model, Equation (4). Both estimates are normally distributed with known covariance matrices. The Kalman filter gain provides the weighting between the two sets of information about the state

$$\mathbf{K}(t_i) = \mathbf{P}(t_i^-)\mathbf{H}^T[\mathbf{H}\mathbf{P}(t_i^-)\mathbf{H}^T + R]^{-1} \tag{7}$$

where $\mathbf{H}$ is the measurement or observation matrix

R is the measurement covariance noise matrix

$\mathbf{K}$ is the Kalman filter gain

The measurement update equations, shown below, determine the new state estimates and covariance matrix after the measurement at time t_i is incorporated. The updated state estimate is:

$$\hat{\mathbf{x}}(t_i^+) = \hat{\mathbf{x}}(t_i^-) + \mathbf{K}(t_i)[z_i - \mathbf{H}\hat{\mathbf{x}}(t_i^-)] \tag{8}$$

where z_i is the actual measurement realization at time t_i

The updated state estimate $\hat{\mathbf{x}}(t_i^+)$ is the previous state estimate $\hat{\mathbf{x}}(t_i^-)$ updated with the Kalman filter gain multiplied by the residuals provided by the measurement. The residual is obtained from the measurement by taking the actual measurement z_i and subtracting the best prediction of the measurement before the measurement was received. $\mathbf{H}\hat{\mathbf{x}}(t_i^-)$ is the prediction of the measurement based on the assumed measurement model, Equation (4), and the fact that the measurement noise

$v(t_i)$ has a mean of zero. Thus, the differences between the predicted measurements and the actual measurements are the residuals

$$r_i = z_i - \mathbf{H}\hat{\mathbf{x}}(t_i^-) \tag{9}$$

These residuals are critical in simulation applications of the Kalman filter.

Along with an updated state estimate, an updated covariance matrix is also calculated. Since the Kalman filter gain $\mathbf{K}(t_i)$ is the relative weight based on comparing the variances of the estimate from the propagation stage and the measurement model, this same gain is used to determine the reduction in variance resulting from incorporating the measurement information.

$$\mathbf{P}(t_i^+) = \mathbf{P}(t_i^-) - \mathbf{K}(t_i)\mathbf{H}\mathbf{P}(t_i^-) \tag{10}$$

Each propagation is followed by a measurement update, and then the two-stage cycle begins again.

Another important observation is that the state estimate covariances $\mathbf{P}(t_i^-)$ and $\mathbf{P}(t_i^+)$ do not depend on the actual measurements. If the system model $\{\mathbf{\Phi}, \mathbf{G}, \mathbf{H}\}$ are known, and the dynamics driving noise variance Q_d and the measurement noise variance R are known, then the Kalman filter gains $\mathbf{K}(t_i)$ and the covariances $\mathbf{P}(t_i^-)$ and $\mathbf{P}(t_i^+)$ are completely determined. Therefore, these values can be precomputed and stored prior to the actual running of the Kalman filter.

In addition, the Kalman filter gain $\mathbf{K}(t_i)$ and the covariance matrices $\mathbf{P}(t_i^-)$ and $\mathbf{P}(t_i^+)$ are functions of the initial estimate $\mathbf{P}(t_0)$ and the variance of the dynamics driving noise Q_d and the measurement noise variance R. Since Q_d and R are constant (as are $\mathbf{\Phi}, \mathbf{G}$, and $\mathbf{H}$) in these applications, $\mathbf{K}(t_i)$, $\mathbf{P}(t_i^-)$, and $\mathbf{P}(t_i^+)$ attain steady-state values as the contribution of $\mathbf{P}(t_0)$ decays.

Using the steady-state matrices, the two-step Kalman filter algorithm simplifies to only two equations. The first equation is the same propagation relationship, Equation (5), the second equation is the measurement update from Equation (8) with a constant Kalman filter gain

$$\hat{\mathbf{x}}(t_i^+) = \hat{\mathbf{x}}(t_i^-) + \mathbf{K}[z_i - \mathbf{H}\hat{\mathbf{x}}(t_i^-)]$$

The covariances matrices are not calculated since their steady-state values are used for all time.

After applying the Kalman filter to the simulation output data, not only are the estimates of the filter-design state vector and the residual vector at each time available, but so are the covariance matrices associated with these vectors. The covariance matrix $\mathbf{P}(t_i^+)$ represents the individual variances for each of the filter-design states and their associated covariances, conditioned on the measurements. This additional information may be used to improve simulation output analysis techniques.

System Identification

Before applying a Kalman filter to simulation output, the system dynamics and measurement equations must be determined. For engineering applications, these equations are developed by aggregating the effect of subsystem components and empirical testing. Since discrete-event simulations are used in applications where no analytically tractable solution exists, the dynamics and measurement equations must be deduced from the simulation output sequence.

Assume that the discrete stochastic process of simulation output, which has attained a stationary steady-state distribution, can be represented by a linear time-invariant discrete system with normally-distributed noise inputs. The dynamics equation, Equation (3), is determined by the transition matrix $\mathbf{\Phi}$, noise dispersion matrix $\mathbf{G}$, and the variance Q_d of the discrete, white, normally-distributed, zero-mean dynamics driving noise $w_d(t_i)$. The transition matrix $\mathbf{\Phi}$ relates the state vector $\mathbf{x}(t_i)$ at one time to the state vector at the next time. Similarly, the measurement or observation equation, Equation (4), is determined by specifying the measurement matrix $\mathbf{H}$ and the variance R of the discrete, white, normally-distributed, zero-mean measurement noise $v(t_i)$. Therefore, to specify the system equations completely, $\mathbf{\Phi}$, Q_d, $\mathbf{H}$, $\mathbf{G}$, and R must be known or estimated. However, if they are estimated, the Kalman filter application is no longer linear and optimality cannot be guaranteed.

Gallagher [11] has done extensive research on identifying appropriate systems for simulation outputs. The rest of this chapter utilizes the results of his work.

In this simulation output analysis application, the first step to apply a Kalman filter is to determine appropriate systems equations. Both Kelton and Law [22] and Schruben [43] model the

steady-state output as a constant mean plus noise

$$y_n = \mu_y + \eta_n \text{ for observations } n = 1, 2, \ldots, N \tag{11}$$

where η_n is noise with $E[\eta_n] = 0$. The subscript has been changed from i to n to represent the N simulation observations.

In a similar approach, the correlated noise could be modeled with a linear, dynamic, stochastic system where the noises and lagged noises would be the filter-design states. Therefore, the steady-state simulation output sequence $\{y_n\}$ could be approximated by a steady-state mean μ_y plus a correlated noise η_n. Since the steady-state process is being modeled, a time-invariant process appears appropriate.

Since the noises η_n are time correlated, an appropriate model must be determined and estimated. In time series analysis, an AR(2) process

$$\eta_n = \phi_1 \eta_{n-1} + \phi_2 \eta_{n-2} + w_d(t_n)$$

can model a mixture of damped exponentials [4:59]. Gallagher has suggested that an AR(2) model might be a robust model for simulation output [11].

In the proposed model, the simulation output is divided into a steady-state mean plus correlated noise. The linear shaping filter for the correlated noise will be modeled as an AR(2) with a two-state dynamics equation. The noise in simulations may have higher order effects than those modeled by an AR(2) process. By increasing the dynamics noise variance Q_d, a reduced order model may be sufficient. In addition, perhaps the "lack of fit" of the simple model can be estimated and incorporated as measurement noise. Thus, an increased measurement noise variance R, will indicate that the simulation output is of a higher order than two.

These autoregressive models can be written as discrete, linear, time-invariant, stochastic, dynamic systems with the appropriate choices of defining matrices. Let the filter-design system states $\mathbf{x}(t_n)$ be the autoregressive noises. The dynamics equation for an AR(2) noise process is

given by substituting

$$\mathbf{x}(t_n) = \begin{bmatrix} \eta_n \\ \eta_{n-1} \end{bmatrix}, \ \mathbf{\Phi} = \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix}, \text{ and } \mathbf{G} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{12}$$

in Equation (3).

The measurement model for this process is given by Equation (4) where $\mathbf{H} = \begin{bmatrix} 1 & 0 \end{bmatrix}$. The measurements z_i would equal the simulation output y_i corrected for their mean. However, the traditional autoregressive model does not include measurement noise $v(t_n)$, so either $v(t_n) = 0$, or equivalently, since the measurement noise already has a mean of zero, its variance is zero, $R = 0$. While not a traditional autoregressive process, nonzero measurement noise $v(t_n)$ may be incorporated into model [15:51-53]. In this application, since the simulation output y_n is known exactly, the $v(t_n)$ may represent a "lack of fit" from assuming the noise correlation is an AR(2) process. However, inclusion of measurement noise requires that the variance R of the measurement noise $v(t_n)$ must also be estimated.

Since the assumed model for the simulation output is given by Equation (11), the actual measurements of the filter-design states ($x_1(t_i) = \eta_n$), are determined by

$$z_n = y_n - \mu_y \tag{13}$$

The notation indicates that x_i is an element of the vector $\mathbf{x}$.

For this formulation, the $\mathbf{\Phi}$, $\mathbf{G}$, Q_d, $\mathbf{H}$ and R are time-invariant. Therefore, the Kalman filter gain $\mathbf{K}$ and state variance matrices $\mathbf{P}^-$ and $\mathbf{P}^+$ will attain steady-state values. These steady-state values can be determined by setting the the variance matrices at one time period equal to the variance matrix at the next time period. However, in this application, the steady-state solutions were estimated from the simulation output.

In order to implement this formulation of the Kalman filter, the system parameters must be estimated. The AR(2) formulation has five unknown parameters to estimate, which are the steady-state mean μ_y, the variance Q_d of the dynamics driving noise $w_d(t_n)$, the variance R of the measurement noise $v(t_n)$, and the autoregressive coefficients ϕ_1 and ϕ_2.

Parameter Estimation

As mentioned in the previous section five parameters must be estimated in order to implement the proposed Kalman filter. When estimating these parameters we are looking for the set of parameters that provides the "best" Kalman filter.

Least Squares Estimation. One scheme for estimating these parameters is based on the fact that the "best" Kalman filter should provide a minimum value for the sum of the squared residuals. The residuals are calculated with Equation (9). The least squares parameter estimate minimizes the sum of the squared residuals for all the observations. This method is commonly called least squares estimation, and its implementation will be discussed in more detail in Chapter IV.

Multiple Model Adaptive Estimation (MMAE). Another method for parameter estimation is Multiple Model Adaptive Estimation [31:131]. Multiple Model Adaptive Estimation (MMAE) as described by Maybeck [31, 33] can be applied to this estimation problem with the structure of the dynamics and measurement models imposed by *a priori* considerations. The MMAE approach is to approximate the unknown continuous parameter space with discrete points and run a Kalman filter at each combination of the discretized parameters. The MMAE estimation is accomplished by monitoring the residuals for each of the Kalman filters in the bank.

For a single unknown parameter, let "a" denote the parameter to be estimated. The continuous range of values for a will be discretized into K representative values. In most engineering applications, the grid discretization should be determined by the effect of varying the particular parameter [24, 34, 44, 23]. However, in this application, the analyst is not interested in how a change in the parameter effects the system. Instead, he is interested in obtaining a reliable estimate of the mean value of the parameter and its associated variance. The grid discretization is still, however, an important issue and it will be discussed in Chapter IV.

After the discretization of the parameter space is complete, let the probability that the parameter a assumes the value a_k conditioned on the measurement history prior to and including time t_n be $p_k(t_n) =$ Prob $(\mathbf{a} = a_k | Z(t_n) = Z_n)$, where Z_n is the measurement history up to time

t_n. Assuming a is limited to the K discrete values a_k, the probability is calculated as [34]:

$$p_j(t_n) = \frac{f_{z(t_n)|\mathbf{a},Z(t_{n-1})}(z_n|a_j, Z_{n-1}) \cdot p_j(t_{n-1})}{\sum_{k=1}^{K} f_{z(t_n)|\mathbf{a},Z(t_{n-1})}(z_n|a_k, Z_{n-1}) \cdot p_k(t_{n-1})} \tag{14}$$

The probabilities are calculated using the residuals $r_k(t_n) = z_n - \mathbf{H}_k\hat{x}_k(t_n)$ and their covariance matrices $A_k(t_n) = \mathbf{H}_k\mathbf{P}_k(t_n^-)\mathbf{H}_k^T + R_k$. Since these residuals are assumed jointly normally distributed

$$f_{z(t_n)|\mathbf{a},Z(t_{n-1})}(z_n|a_k, Z_{n-1}) = (2\pi)^{-1/2}|A_k(t_n)|^{-1/2} \exp\left\{-\frac{1}{2}r_k^2(t_n)A_k^{-1}(t_n)\right\} \tag{15}$$

since the measurements, z_n, are univariate. Since the residuals $r_k(t_n)$ for the "best" a_k should be small (relative to filter-computed residual variance $A_k(t_n)$, the "best" value of a_k will have a high probability assigned by the preceding $p_k(t_n)$ computation. Similarly, the residuals for a "mismatched" model should be large and the associated probability should be small [33].

In the recursion calculations of p_k in Equation (14), if any of the p_k became zero, it would remain zero thereafter. To prevent a set of parameters from getting prematurely discarded, the probabilities should be given a lower bound which depends on the number of discrete points in the parameter space [33]. This lower bound permits the probabilities to continue to adapt throughout the observation set.

The Bayesian minimum mean square error state estimate is the sum of the K discrete state estimates weighted by their associated probabilities:

$$\hat{x}(t_n^+) = \sum_{k=1}^{K} \hat{x}_k(t_n^+) \cdot p_k(t_n)$$

where each $\hat{\mathbf{x}}_k(t_n^+)$ is calculated using a discrete-time Kalman filter with the associated a_k value. Similarly, from Maybeck [31:132-133], the conditional mean of the unknown parameter a at time t_i is

$$\hat{\mathbf{a}}(t_i) = \sum_{k=1}^{K} a_k \cdot p_k(t_i) \tag{16}$$

and an indication of the precision of this estimate is given by the conditional covariance matrix of $\mathsf{a}(t_i)$,

$$\Sigma_{\mathsf{a}(t_i)|Z_i} = E\{[\mathsf{a} - \hat{a}(t_i)][\mathsf{a} - \hat{a}(t_i)]^T | Z(t_i) = Z_i\} = \sum_{k=1}^{K} [a_k - \hat{a}(t_i)][a_k - \hat{a}(t_i)]^T p_k(t_i) \qquad (17)$$

The MMAE parameter estimate, Equation (16), and associated variance, Equation (17), can be used in a method for constructing confidence intervals for the parameter.

Summary

The Kalman filter is an optimal, recursive, next-step prediction algorithm. The Kalman filter propagates estimates of various states through time and then, using available periodic measurements, updates these estimates. The Kalman filter also provides variance information on the states.

This chapter has provided the basic foundations of the discrete-time Kalman filter for a time-invariant linear system with no control inputs, and stationary, normally-distributed dynamics and measurement noises. In particular, the Kalman filter utilizes a linear stochastic difference equation, Equation (3), to represent the underlying system. The available measurements, which are univariate in this application, are related to the states using a measurement model, Equation (4).

This chapter has also shown how a Kalman filter can be used to model discrete-event simulation output. In order to implement the proposed Kalman filter, several parameters must first be estimated and this chapter discussed two potential methods for their estimation.

The following chapter provides the methodology for this effort. The methodology applies the Kalman filter relationships discussed in this chapter to the development of the proposed confidence interval techniques.

IV. Methodology

This chapter will cover the development of the Kalman filter techniques for confidence interval construction. Three different techniques based on the Kalman filter were explored in this research. The first technique is based on the estimated "best" Kalman filter. The mean value of the parameter is calculated as the average of all the observations and its associated variance is estimated using information provided by the Kalman filter. The other two techniques use MMAE to obtain an estimate of the mean value. One of the MMAE techniques uses a MMAE estimate of the variance to construct a confidence interval. The other MMAE technique uses a probabilistic approach to construct a confidence interval based on the MMAE filter's final probabilities. The three techniques are described in the following sections.

Kalman Filter Variance Approach

This method first requires that the optimal Kalman filter be estimated. This estimated Kalman filter is referred to as the "best" Kalman filter although its optimality can not be guaranteed. As discussed in Chapter III, five parameters must be estimated. Gallagher [11] discusses the estimation of these five parameters; the procedure discussed below is based on one alternative he discusses.

The first, and easiest, parameter to estimate is the mean value (μ_y) of the simulation output $\{y_1, y_2, \ldots, y_n\}$. Given a large sample size, the strong law of large numbers [29:292] suggests that the average of the observations ($\bar{y}$) is a good estimator for μ_y.

The variance Q_d of the dynamics driving noise $w_d(t_n)$, and the variance R of the measurement noise $v(t_n)$ are both directly related to the Kalman filter gain $\mathbf{K}$. The Kalman filter gain determines how much emphasis or weight should be placed on the incoming observations. If Q_d is large, suggesting an imprecise or noisy dynamics model, then the measurements are weighted more heavily, and thus, the magnitude of $\mathbf{K}$ is increased. On the other hand, if R is large, the measurements are corrupted with a large amount of noise and more emphasis is placed on the dynamics equations, by decreasing the magnitude of $\mathbf{K}$. When Q_d and R are both scalars, as they are in this case, it is only their ratio, not their individual magnitudes, that determines $\mathbf{K}$. This indeterminacy between

the two parameters makes it impossible to estimate them both at the same time. Instead, one must either fix one of them or estimate a scalar that represents their ratio.

Harvey [15:107] suggests one way to approach this problem of estimating both Q_d and R. The Kalman filter gain is solved in terms of the transition matrix parameters and a scalar rather than Q_d, R, $\mathbf{P}^-$, and $\mathbf{P}^+$. The transition matrix parameters and the scalar can be estimated using the sample autocorrelations and least squares estimation.

The Kalman filter gain can be solved without Q_d or R as follows. From the Kalman filter gain equation, Equation (7), with the AR(2) system matrices, Equation (12),

$$\mathbf{K} = \begin{bmatrix} P_{11}^- \\ P_{12}^- \end{bmatrix} \left(\frac{1}{P_{11}^- + R} \right)$$

which simplifies to

$$\mathbf{K} = \begin{bmatrix} \frac{P_{11}^-}{P_{11}^- + R} \\ \frac{P_{12}^-}{P_{11}^- + R} \end{bmatrix} \tag{18}$$

Now set the first element of the Kalman filter gain to an unknown scalar, k

$$K_1 = k = \frac{P_{11}^-}{P_{11}^- + R} \tag{19}$$

and solve for $\frac{P_{12}^-}{P_{11}^- + R}$ in terms of the other known or estimated parameters. From the covariance update equation, Equation (10),

$$\mathbf{P}^+ = \mathbf{P}^- - \mathbf{KHP}^- = \begin{bmatrix} P_{11}^- - \frac{(P_{11}^-)^2}{P_{11}^- + R} & P_{12}^- - \frac{P_{11}^- P_{12}^-}{P_{11}^- + R} \\ P_{12}^- - \frac{P_{11}^- P_{12}^-}{P_{11}^- + R} & P_{22}^- - \frac{(P_{12}^-)^2}{P_{11}^- + R} \end{bmatrix} \tag{20}$$

Substituting this result for $\mathbf{P}^+$ into the covariance propagation equation, Equation (6), yields

$$\mathbf{P}^- = \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} P_{11}^- - \frac{(P_{11}^-)^2}{P_{11}^- + R} & P_{12}^- - \frac{P_{11}^- P_{12}^-}{P_{11}^- + R} \\ P_{12}^- - \frac{P_{11}^- P_{12}^-}{P_{11}^- + R} & P_{22}^- - \frac{(P_{12}^-)^2}{P_{11}^- + R} \end{bmatrix} \begin{bmatrix} \phi_1 & 1 \\ \phi_2 & 0 \end{bmatrix} + \begin{bmatrix} Q_d & 0 \\ 0 & 0 \end{bmatrix} \tag{21}$$

Solving Equation (21) for P_{12}^- produces:

$$P_{12}^- = \phi_1 P_{11}^- - \frac{\phi_1 (P_{11}^-)^2}{P_{11}^- + R} + \phi_2 P_{12}^- - \frac{\phi_2 P_{11}^- P_{12}^-}{P_{11}^- + R} \tag{22}$$

Equation (19) yields $P_{11}^- = k(P_{11}^- + R)$. Using this result and Equation (22), a solution can be obtained for $\frac{P_{12}^-}{P_{11}^- + R} = K_2$

$$K_2 = \frac{P_{12}^-}{P_{11}^- + R} = \frac{\phi_1 k - \phi_1 k^2}{1 - \phi_2 + \phi_2 k} \tag{23}$$

Equations (19) and (23) determine the Kalman filter gain $\mathbf{K}$ in terms of the autoregressive coefficients and the scalar k.

For an AR(2) model with measurement noise the the values of ϕ_1 and ϕ_2 can be estimated using the lagged autocorrelation coefficients. The simulation output sequence is modeled as an AR(2) process with measurement noise. Since the filter-design state $x_1(t_i)$ models the AR(2) correlated noise term, the autocovariances, $\gamma_{x_1}(i)$, of $x_1(t_i)$ are the same as an AR(2) process without measurement noise [1:204-205]:

$$\gamma_{x_1}(0) = \left(\frac{1 - \phi_2}{1 + \phi_2}\right) \frac{Q_d}{(1 - \phi_2)^2 - \phi_1^2}$$

$$\gamma_{x_1}(1) = \frac{\phi_1}{1 - \phi_2} \gamma_{x_1}(0)$$

$$\gamma_{x_1}(2) = \phi_1 \gamma_{x_1}(1) + \phi_2 \gamma_{x_1}(0)$$

In the Kalman filter terms, the variance of $x_1(t_i)$ at lag zero is P_{11}^-, which therefore equals $\gamma_x(0)$, $P_{11}^- = \gamma_x(0)$.

Using the relationships from Equation (13) and (4), the output sequence y_n can be expressed as

$$\begin{aligned} y_n &= \mu_y + z_n \\ &= \mu_y + \mathbf{H}\mathbf{x}(t_n) + v(t_n) \\ &= \mu_y + x_1(t_n) + v(t_n) \end{aligned}$$

Since the measurement noise is assumed independent of the state vector

$$\begin{aligned} \mathrm{Var}\{y_n\} &= E\{[\mu_y + x_1(t_n) + v(t_n)]^2\} - \mu_y^2 \\ &= E\{x_1^2(t_n)\} + E\{v^2(t_n)\} \\ &= \left(\frac{1-\phi_2}{1+\phi_2}\right)\left(\frac{Q_d}{(1-\phi_2)^2-\phi_1^2}\right) + R \\ &= P_{11}^- + R \end{aligned} \tag{24}$$

The variance of y_n, Equation (24), is the autocovariance at lag zero $\gamma_y(0)$. Since the measurement noise is uncorrelated in time, the lagged autocovariances of the simulation output is the same as the filter-design states.

$$\gamma_y(1) = \gamma_{x_1}(1)$$

$$\gamma_y(2) = \gamma_{x_1}(2)$$

Thus, the theoretical autocorrelations of the measurements equal

$$\rho(1) = \frac{\gamma_y(1)}{\gamma_y(0)} = \frac{\phi_1}{1-\phi_2}\left(\frac{P_{11}^-}{P_{11}^- + R}\right)$$

$$\rho(2) = \frac{\gamma_y(2)}{\gamma_y(0)} = \phi_1\rho(1) + \phi_2\left(\frac{P_{11}^-}{P_{11}^- + R}\right)$$

Applying Equation (19),

$$\rho(1) = \frac{k\phi_1}{1-\phi_2}$$

$$\rho(2) = \phi_1\rho(1) + \phi_2 k$$

The theoretical autocorrelations ($\rho(i)$) can be estimated with the sample autocorrelation coefficients ($r(i)$) with the following equation:

$$r(i) = \hat{\rho}(i) = \frac{\sum_{t=i+1}^{N}(y_t - \bar{y})(y_{(t-i)} - \bar{y})}{\sum_{t=1}^{N}(y_t - \bar{y})^2} \tag{25}$$

The values of ϕ_1 and ϕ_2 can be estimated by simultaneously solving the following set of equations:

$$r(1) = \frac{k\hat{\phi}_1}{1-\hat{\phi}_2} \tag{26}$$

$$r(2) = \hat{\phi}_1 r(1) + \hat{\phi}_2 k \tag{27}$$

Thus, for a given value of $k = K_1$ one can determine the appropriate value for K_2, using Equation (23). Equation (25) can be used to calculate values for $r(1)$ and $r(2)$. Then, ϕ_1 and ϕ_2 can be estimated by simultaneously solving Equations (26) and (27). Since $k = \frac{P_{11}^-}{P_{11}^- + R}$, and P_{11}^- and R are variances, the value of k is bounded between 0 and 1. Therefore, by searching this range for k, the least squares estimate can be found.

Using the least squares estimate of k and the associated $\hat{\phi}_1$ and $\hat{\phi}_2$, along with the corresponding Kalman filter residuals, estimates of Q_d and R, and thus P^- and P^+, can be obtained. The Kalman filter with the least squares $\mathbf{K}$, and associated $\hat{\phi}_1$ and $\hat{\phi}_2$ will provide a sequence of residuals, $\{r_1, , r_2, \ldots, r_N\}$. Based on linear dynamics system theory, these residuals are assumed to be normally distributed with mean 0 and variance $\mathbf{HP}^-\mathbf{H}^T + R$. This variance is approximated by the mean squared residual value, $N^{-1}\sum_{i=1}^{N} r_i^2$. Therefore

$$N^{-1}\sum_{i=1}^{N} r_i^2 = \mathbf{HP}^-\mathbf{H}^T + R = \mathbf{P}_{11}^- + R \tag{28}$$

K_1 and K_2 are known values (of the "best" Kalman filter) and $N^{-1}\sum_{i=1}^{N} r_i^2$ is easily provided by processing the observations through the Kalman filter. Equations (19), (23), and (28) provide a system of three equations and three unknowns (P_{11}^-, P_{12}^-, and R). The estimated values of these unknowns can then be easily calculated. From the covariance matrix propagation equation for the AR(2) model, Equation (21), we obtain

$$Q_d = P_{11}^- - \left(\left(\phi_1\left(\phi_1 P_{11}^+ + \phi_2 P_{12}^+\right)\right) + \left(\phi_2\left(\phi_1 P_{12}^+ + \phi_2 P_{22}^+\right)\right)\right) \tag{29}$$

Since the propagation with $\mathbf{\Phi} = \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix}$, moves the element of the estimated state vector $\hat{x}_1(t_i^+)$ to the second element of the propagated state estimate vector $\hat{x}_2(t_{i+1}^-)$, their variances are equal

$$P_{22}^- = P_{11}^+ \tag{30}$$

Using Equations (30), (23), (28), and (20) we can solve Equation (29) for Q_d.

Once this "best" filter has been found an estimate of the process variance (σ_y^2) and mean variance ($\sigma_{\hat{m}u_y}^2$) can be found. By substituting the estimated parameters into Equation (24), an estimate of the variance of the output process is obtained.

Fishman [10] estimates the variance of the mean estimator of an AR(2) process as

$$\hat{\sigma}_{\bar{y}}^2 = \frac{\sigma_{pn}^2}{(1-\phi_1-\phi_2)^2} = \frac{Q_d}{(1-\phi_1-\phi_2)^2} \tag{31}$$

where σ_{pn}^2 is the variance of the process noise.

In order to estimate the variance of the mean estimator we use the fact that, by independence, the variance of the sum of the measurement noise terms is the sum of their variances. Therefore, using Equation (24),

$$\begin{aligned} \mathrm{Var}\{\hat{\mu}_y\} &= \mathrm{Var}\{\tfrac{1}{N}\textstyle\sum_{n=1}^{N} y_n\} \\ &= \mathrm{Var}\{\tfrac{1}{N}\textstyle\sum_{n=1}^{N} x_1(t_n)\} + \tfrac{1}{N^2}\textstyle\sum_{n=1}^{N}\mathrm{Var}\{v(t_n)\} \\ &= \mathrm{Var}\{\tfrac{1}{N}\textstyle\sum_{n=1}^{N} x_1(t_n)\} + \tfrac{R}{N} \end{aligned}$$

Using Fishman's approximation, Equation (31), for the variance of the autoregressive portion,

$$\mathrm{Var}(\hat{\mu}_y) \approx \frac{\sigma_{pn}^2}{N(1-\phi_1-\phi_2)^2} + \frac{R}{N} \tag{32}$$

Fishmans approximation for degrees of freedom d with $p = 2$ [10], is modified by subtracting one for the estimating of R,

$$d = \frac{N(1-\hat{\phi}_1-\hat{\phi}_2)}{4+4\hat{\phi}_2} - 1 \tag{33}$$

In summary, this approach searches over the admissible range of the scalar k to determine which value of k minimizes the sum of the squared residuals. By selecting a value of k, estimates are obtained for all of the necessary Kalman filter parameters. Using these estimated parameters, and the sum of the squared residuals, an estimate of the mean estimators variance is obtained from the modified Fishmans approximation, Equation (32) (Q_d is the estimate of σ_{pn}^2). A confidence interval is then constructed.

The estimation routine discussed above is only one of many possible estimation techniques.

Mehra [35] discusses several other estimation techniques including those based on maximum likelihood and linear least squares.

MMAE Approach

This technique relies on MMAE as discussed in Chapter III. Under the assumptions of linear models with white normally-distributed noises, the residuals for each of the k filters in the bank,

$$r_k(t_n) = z(t_n) - \mathbf{H}\hat{\mathbf{x}}_k(t_n^-)$$

are each a white, normally-distributed sequence with zero mean and variance of $A_k = \mathbf{H}\mathbf{P}_k^-\mathbf{H}^T + R_k$. Using these variances and the values of the residuals, MMAE calculates the probability that the unknown parameters $\mathbf{a}_k$ used in each filter are correct. In addition, using Equations (16) and (17), MMAE estimates the values and variances of the unknown parameters.

Assume there is only one unknown parameter, the simulation output mean μ_y. In this case, each a_k becomes a scalar value equal to an estimate of the mean. A bank of filters can be designed each having a unique value of $a_k = \hat{\mu}_y$. The $\hat{\mu}_y$ values can be centered on the average value of the simulation observations since it is a good estimator for the true mean. Each filter uses the same K_1, K_2, ϕ_1, ϕ_2, P_{11}^-, and R values (those estimated in the "best" filter from the first Kalman filter technique). The MMAE probabilities of each filter's mean value being the correct one are conditioned on the estimated parameters and the measurements. After processing all of the data through the bank of filters, an MMAE estimate of the mean value and its associated variance are obtained from Equations (16) and (17).

Before implementing this technique and constructing a confidence interval, several tactical issues must be decided. These issues are all interrelated and can have a significant impact on the MMAE estimate of a parameter's mean and variance. These issues are:

1. The spread between the filters with the smallest and largest mean estimate.
2. The number of filters in the bank.
3. The spacing between the filters.
4. The use of the MMAE variance estimate.

5. The initial or *a priori* MMAE filter probabilities.

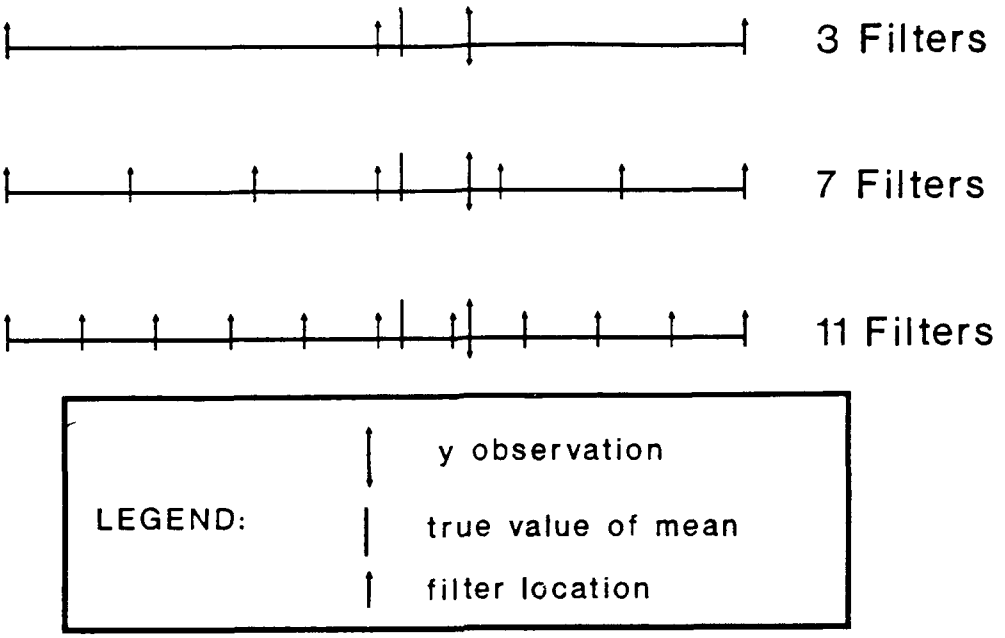
In most engineering applications, an extensive and detailed model, known as the truth model, can be generated. The truth model can be used to evaluate a filter's performance and provides many mathematical techniques for answering the issues listed above [24, 34, 44, 23]. However, discrete-event simulation is most often applied when no analytical solution is apparent. Therefore, no truth model is obvious and heuristics were used to answer these questions.

Issues 1, 2, and 3 are directly related, since each impacts the others. For Issue 1, the total spread of the filters can be too large or too small. If the spread is too large, then there will be one or more filters at each end of the bank that consistently ends up with a near-zero probability of having the correct mean value. Since these filters will provide no information, they are an unnecessary computational burden and should not be included. However, on the other hand, if the end filters have a significant probability, then an artificial upper bound on the estimated variance $\hat{\sigma}_{\tilde{y}}^2$ from Equation (17) is induced. This loss of information can seriously impact the ability to construct a meaningful confidence interval.

This research, as well as many previous studies [24, 34, 44, 23], has shown that the number of filters and the spacing between them are critical issues. In general, if the filters are spaced too far apart, then the probabilities of the filters being correct will concentrate on one or two filters. When this occurs it is impossible to obtain a reliable estimate of the variance. If the filters are spaced too close together there will be computational problems. Because of the random component in the measurements, two or more filters may have nearly equal MMAE probabilities. In other words, the filter probabilities may switch back and forth between several filters without stabilizing. If the filters are spaced too close together there is also an identifiability problem in which the residuals from the two "close" filters will be essentially indistinguishable.

Computational problems can also arise if the filters are spaced at an intermediate spacing (i.e. in between the two cases mentioned above). Figure 1 shows three possible spacings between consecutive filters in the MMAE approach. For each arrangement, as shown in Figure 1, the true, but unknown, value of the mean is the same, and the simulation observation y has the same value. The MMAE algorithm determines the probability of that particular observation coming from a process with each filter's mean estimate. For each simulation observation the filter closest (as

Figure 1. Filter Spacing



determined by the filter's mean value) to the observation, as indicated in Figure (1) will receive the highest probability of being the correct filter. For a simulation observation near the true mean, the filter with mean estimate closest to the true mean should receive the highest probability. The distance between the filter with the highest probability and the true mean is largest in the moderately-spaced filters ("7 Filters" in Figure 1). The moderately-spaced filters may, therefore, introduce a bias into the MMAE estimate of the mean value. However, widely-spaced filters ("3 Filters" in Figure 1) may lead to a crude MMAE estimate of the mean, and closely-spaced ("11 Filters" in Figure 1) filters may have problems with filter probabilities not stabilizing. Without a truth model, a hueristic approach was to use the smallest number of filters and as wide a total spread, as possible, without degrading the estimates obtained. This ensured no loss of information and computational efficiency.

To construct a confidence interval, one must assume or approximate the underlying distribution of the parameters estimator. Usually an estimate of the mean and its associated variance, along with an assumed probability distribution is used. For example, the classical statistics method

presented in Chapter II, Equation (2), assumes a t distribution. A confidence interval is then constructed using the estimate of the mean and its associated variance and an appropriate multiplier (e.g. a t critical value). The MMAE estimate may or may not be based on an assumed distribution. Two possible alternatives are 1) estimating the variance of the mean estimate with MMAE and assuming a t distribution, or 2) using the calculated final MMAE probabilities as a discrete approximation of the parameters distribution and construct a $(1-\alpha)$ percent "probability interval" by finding the range of filters that contains $(1-\alpha)$ percent of the probabilities. If all of the Kalman filter parameters were known, not estimated as in this application, the probability interval would have a different meaning than a typical confidence interval. Basically, the probability interval would be a stricter statement that one is $(1-\alpha)$ percent sure that the true value of a given parameter is contained in every constructed interval. However, since the MMAE approach used is based on several estimated parameters this "probability interval" will probably be closer to a confidence interval as discussed in Chapter II . One way of inducing a t distribution might involve uneven spacing between adjacent filters in an MMAE model. If a t distribution is assumed or induced, the degrees of freedom would be the number of filters minus one. One degree of freedom was lost since the filters are centered on an estimated mean.

Summary

This chapter provided the methodology for using the Kalman filter developed in the previous chapter in constructing confidence intervals. Three different construction techniques were proposed.

The first construction technique uses a single Kalman filter to provide an estimate of the mean and its associated variance. A routine for estimating the necessary Kalman filter parameters was discussed. After estimating these parameters an estimation of the mean and its variance can be found using modified forms of two approximations given by Fishman [10], shown in Equations (32) and (33). Using these estimates for the mean and its variance, a confidence interval can be constructed using Equation (2).

The other two techniques use MMAE. The MMAE approaches approximate the unknown continuous parameter space with discrete points and runs a Kalman filter at each of these points. After processing the observations through the bank of filters, estimates of both the mean and its

associated variance are obtained using Equations (16) and (17). The first MMAE technique uses these two estimates and Equation (2) to construct a confidence interval in the normal fashion. The second approach uses the final filter probabilities to estimate the region that contains $(1 - \alpha)$ percent of the probabilities. When using MMAE there are numerous tactical issues that must be decided (e.g., number of filters, and spacing between the filters). This chapter provided a discussion on the impact each of these issues can have on the estimates provided by MMAE.

The following chapter discusses the results of the Monte Carlo analysis; comparing these proposed confidence interval techniques with the widely-accepted techniques.

V. *Monte Carlo Analysis*

This chapter discusses the Monte Carlo analysis of the various confidence interval construction techniques. The four major areas discussed are confidence interval techniques, data selection, baseline Monte Carlo results, and fine tuning of the MMAE technique.

Confidence Interval Techniques

The first step in conducting a Monte Carlo analysis was to develop computer routines to calculate confidence intervals based on the current techniques discussed in Chapter II and the new techniques proposed in Chapter III. Computer routines for three (nonoverlapping batch means, overlapping batch means, and standardized time series) of the five techniques discussed in Chapter II were written in FORTRAN. The routine for nonoverlapping batch means is based on the methodology given by Kang and Goldsman [20:16]. The overlapping batch means code is based on Meketon and Schmeiser's methodology [37]. The standardized time series code uses Schruben's standardized sum methodology [42:1099]. A FORTRAN computer routine, written by Fishman [10], for estimating a parameters mean, variance, and associated degrees of freedom based on the autoregressive method, was modified to construct confidence intervals and to run on available computers. Spectral Analysis was not used due to its computational difficulties. Routines were developed for both of the Kalman filter techniques discussed in Chapter IV. Table 1 gives the names and purpose of each of the computer subroutines for confidence interval construction. The FORTRAN codes are contained in Appendix A.

Data Selection

To evaluate the confidence interval construction techniques, the models that would generate the data had to be selected. Models with analytical mean estimates were necessary to evaluate coverage rates. Two types of data were used in this study and both are discussed in the following sections.

AR(2) Data. Since the underlying system equations for the Kalman filter are estimated as an AR(2) process, data generated as an AR(2) process was chosen as one of the test cases to compare

Table 1. Computer Routines For Confidence Interval Construction

KFMAIN	Main program for both Kalman filter techiques.
ESTPAR	Estimates $\rho(1)$ and $\rho(2)$ from the data. Calculates sample autocorrelations.
SEARCH	Searches over the admissible range of k to find the value that minimizes the squared sum of residuals.
CHECK	Function that determines if, for a given a value of k, corresponding ϕ_1 and ϕ_2 values (as estimated by ESTPHI) are available in the stationary region for an AR(2) process. If they are, it calls KFRESID and returns the value of the residual sum of squares for a filter with those values.
ESTPHI	Given a value of k, estimates the corresponding ϕ_1 and ϕ_2 values.
KFRESID	Processes a set of data through a given Kalman filter and calculates squared sum of residuals.
GETPQR	Calculates the values of Q_d and R.
MAINB	Main program for the three techniques (NOBM, OBM, STDS) that require the selection of either number of batches or batch size (program calculates intervals for three different batch sizes per sample size).
NOBM	Calculates std dev for nonoverlapping batch means method.
OBM	Calculates std dev for overlapping batch means method.
STDS	Calculates std dev for standardized time series method.
MAINNB	Main program for techniques not requiring selection of batch size (currently only AUTOR).
AUTOR	Calculates std dev and t-crit for autoregressive method.
CI	Given sample size, number of batches, and method (e.g., NOBM), actually constructs c.i. and keeps track of statistics for Monte Carlo analysis.
SLAMCOM	Common block of variables used in most of the routines.
COMVAR	Common block of variables used in the Kalman filter routines.

Table 2. AR(2) Test Cases

Case	ϕ_1	ϕ_2	σ^2_{pn}	σ^2_{mn}
1	0.5	0.3	1.0	0.0
2	0.5	0.3	1.0	1.0
3	-1.0	-0.5	1.0	0.0
4	-1.0	-0.5	1.0	1.0
5	-0.5	0.3	1.0	0.0
6	-0.5	0.3	1.0	0.0

the different confidence interval techniques. AR(2) processes are easily modeled using FORTRAN routines and offer analytical solutions for mean values. The general formula for an AR(2) process was modified to allow for the addition of a measurement noise to each generated observation. Thus the process is modelled with the following two equations:

$$x_i = \xi + \phi_1 x_{i-1} + \phi_2 x_{i-2} + \epsilon(pn)_i$$

$$z_i = x_i + \epsilon(mn)_i$$

where ξ is a constant

ϕ_1 is the first autoregressive coefficient

ϕ_2 is the second autoregressive coefficient

$\epsilon(pn)$ is a white, Gaussian process noise with mean 0 and variance σ^2_{pn} (Q_d in the Kalman filter)

$\epsilon(mn)$ is a white, Gaussian measurement noise with mean 0 and variance σ^2_{mn} (R in the Kalman filter)

Six different AR(2) Cases were chosen for this study. The six cases and their associated parameters are summarized in Table (2). These cases (all of which are stationary) were chosen because they each exhibit different autocorrelation and partial autocorrelation functions and thus will subject the techniques to a wide range of output. If the Kalman filter parameters are correctly estimated, $\hat{Q}_d = \sigma^2_{pn}$ and $\hat{R} = \sigma^2_{mn}$. Since the Kalman filter can identify the presence of measurement noise and estimate its variance, the Kalman filter techniques should outperform the

Table 3. Computer Routines for AR(2) Data Generation

ARGEN	Generates data from an AR(2) process, and passes the output on to the baseline Monte Carlo analysis programs.
NORM	Generates Normal(0,1) pseudorandom variates.
STDEV	Calculates the mean, variance, and standard deviation of a sample.
MARGEN	Similar to ARGEN, but used for comparing the MMAE technique with various selections for number of filters and spacing of filters.

conventional techniques when a measurement noise is present. The odd numbered cases have no measurement noise, where the even numbered have measurement noise.

FORTRAN programs were written to generate the AR(2) data. Table 3 provides a brief description of the routines, which are contained in Appendix B. MARGEN is simply a modified version of ARGEN and is not contained in the appendix.

$M/M/1$ Data. The next type of data used to evaluate the techniques was $M/M/1$ queueing simulations. Sheldon and Ross [39:343] define queueing models as those models that deal with systems where entities arrive at a service area according to a random process. After arrival, they are either served immediately, or they wait in line until they can be served. After service, the customer leaves the system. The queueing models used in this study are $M/M/1$. These queues exhibit exponential (Markovian) distributions for both interarrival and service times and have a single server with infinite queue capacity [17:90]. $M/M/1$ queueing models are useful in simulation research because the output parameters have analytical solutions, and it is easy to model the queues as discrete-event simulations. However, $M/M/1$ queues have very complex statistics associated with them, and it is difficult to construct good confidence intervals for their parameters.

Four different $M/M/1$ cases were chosen for this study. The four cases and their associated parameters are summarized in Table (4). Average waiting time in the queue is a "statistic based upon observations", and average number in the queue is a "statistic based upon a time persistent variable". The traffic intensity is simply the ratio of arrival rate (how many entities arrive per unit time) to service rate (how many entities are serviced per unit time). Therefore, as traffic intensity increases the system becomes more "congested", the output sequence autocorrelation increases,

Table 4. $M/M/1$ Test Cases

Case	Output Parameter	Traffic Intensity
1	Average Waiting Time in Queue	0.5
2	Average Waiting Time in Queue	0.8
3	Average Number in Queue	0.5
4	Average Number in Queue	0.8

Table 5. Computer Routines for $M/M/1$ Data Generation

SLM	FORTRAN inserts for SLAM II model that generates waiting times.
INTLC	SLAM II FORTRAN insert that is called before each run to initialize certain parameters.
OTPUT	Routine in SLM
EVENT	SLAM II FORTRAN insert that actually records parameter of interest.
SLM2	Similar to SLM, but used for number in queue
MSLM	Similar to SLM, but used only for comparing the MMAE technique with various parameters (e.g. of filters).
MSLM2	Similar to SLM2, but used only for comparing the MMAE technique with various parameters (e.g. of filters).
SLCOD	SLAM II network statements for SLM and MSLM.
SLCOD2	SLAM II network statements for SLM2 and MSLM2.

and, in general, it is harder to estimate the true values of the systems parameters. Two different traffic intensities (0.5 and 0.8) were examined. These traffic intensities are typical of those seen in other confidence interval studies. For the traffic intensity of 0.5 the specific inter-arrival and service times used were 2 and 1. For the traffic intensity of 0.8 the specific service and inter-arrival times were 5/4 and 1.

The $M/M/1$ data was generated using SLAM II [38] with FORTRAN inserts. A brief description of the codes are given in Table 5. The SLAM II network statements and FORTRAN inserts for SLM, INTLC, OTPUT, EVENT, and SLCOD are contained in Appendix C. The remaining routines are simply modified versions of the others and are not included in the appendix.

Table 6. Computer Routines for Monte Carlo Analysis

INITIAL	Initializes parameters for the Monte Carlo analysis.
CONTROL	Changes the sample size for the analysis (program calculates intervals for three different sample sizes).
FIGURE	Calculates performance measures (e.g., actual coverage, average half width, etc.) after all runs are completed.
OUTPUT	Produces output files of data.
GETMX	After CONTROL selects data size this routine calculates the mean of that sample size.

Baseline Monte Carlo Anlaysis Results

This section presents the results of the baseline Monte Carlo analysis used to compare the various techniques for confidence interval construction. The methods are compared and contrasted based on their respective coverage rates, average half widths, and standard deviation of the half widths. As discussed in Chapter II, practitioners want a method that provides actual coverage equal to nominal coverage, while keeping the average half width as small as possible, and the standard deviation of the half width values as low as possible.

The Monte Carlo analysis involved 1000 runs on each case with a nominal coverage rate of 90 percent. Three different sample sizes were selected: 1280, 2560, and 5120. For those methods requiring a selection of batch size, three different sizes (for each sample size) were selected. These batch sizes correspond to 1/5, 1/10, and 1/20 of the total sample size. These sample sizes and batch sizes were selected to correspond to a similar analysis by Law and Kelton [29:563].

Computer routines were written in FORTRAN to conduct the Monte Carlo analysis. A brief description of these routines is given in Table 6, and the FORTRAN codes are contained in Appendix D.

Six tables are generated for each of the 10 models (6 AR(2) cases and 4 $M/M/1$ cases). The six tables present the following information:

1. Actual coverage rates.
2. Average half widths.
3. Standard deviation of half widths.

4. Estimated Kalman filter parameters.

5. Average bias of $\bar{y}$ (as estimated by the average of the simulation observations) and $\bar{y}$ (as estimated by MMAE) from the theoretical (analytically determined) value of μ_y.

6. The standard deviation of the mean estimates errors in Number 5.

The first three tables present the information for each of the methods as a function of data size and, where appropriate, batch size. The batch size can be determined by dividing the data size by the numbers (5, 10, and 20) listed under the methods. The information presented in the last three cases are not dependent on batch size. All of the tables in this section use the case numbers presented in Tables 2 and 4.

The baseline MMAE method used 21 filters (ten on each side of the center filter and the center filter). For the baseline MMAE method the filters were evenly spaced for computational ease. The two end filters were placed at $\bar{y} \pm 5\hat{\sigma}_{\bar{y}}$ where $\bar{y}$ is simply the grand average of the data. $\hat{\sigma}_{\bar{y}}$ is the estimated variance of $\bar{y}$ using the modified Fishman's approximation, Equation (32), as discussed in Chapter IV. The *a priori* probabilities of the filters were set at a triangular distribution in which the two end filters each had a probability of $\frac{1}{\left(\frac{N_f+1}{2}\right)^2}$ and the center filter had a probability of $\frac{1}{\left(\frac{N_f+1}{2}\right)}$; where N_f is the total number of filters used. This type of triangular distribution was hoped to provide a rough approximation of a normal distribution. A normal distribution was used since the technique utilizes a t-critical value in constructing a confidence interval.

The method names are abbreviated in the tables according to the following notation:

- NOBM — Nonoverlapping Batch Means
- OBM — Overlapping Batch Means
- STDS — Standardized Time Series
- AUTO — Autoregressive
- SKF — Kalman Filter using Modified Version of Fishman's Approximation for $\sigma_{\bar{y}}^2$
- MMAE — Multiple Model Adaptive Estimation with 21 filters evenly spaced $\pm 5\sigma_{\bar{y}}$ (as estimated above) around the MMAE's estimate of $\bar{y}$.

Table 7. Actual Coverage Rates (AR(2) Case 1)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	0.910	0.899	0.884	0.910	0.885	0.877	0.899	0.887	0.848	0.897	0.898	0.887
2560	0.899	0.895	0.891	0.905	0.890	0.888	0.905	0.881	0.869	0.896	0.897	0.885
5120	0.907	0.910	0.909	0.915	0.914	0.913	0.912	0.908	0.899	0.918	0.922	0.910

Note: With nominal rate of 0.9, estimation accuracy is $\approx \pm 0.016$ for 1000 runs.

AR(2) Results. The first results are from the six AR(2) cases described in Table 2. The results from each case are discussed below.

AR(2) Case 1 ($\phi_1 = 0.5, \phi_2 = 0.3, \sigma_{pn}^2 = 1.0, \sigma_{mn}^2 = 0.0$). The results from this model are presented in Tables 7–12. Table 7 indicates that virtually all of the methods achieved nominal coverage of 0.90. As expected, coverage generally increases as sample-size increases. With data size of 5120, SKF and AUTO gave the highest coverage, which is not surprising considering that both of these methods are based on autoregressive models and were applied to AR(2) data. NOBM, OBM, and STDS coverage tend to increase (sometimes marginally) as batch size increases (and thus number of batches decrease). For large sample size, OBM offered higher coverages than both NOBM and STDS. At small sample sizes, NOBM (particularly when used with a small number of batches) has the highest coverage rate. The estimation accuracy of ± 0.016 makes it impossible to draw conclusions as to which method is superior, since many of the coverage rates are equal within this accuracy. One should consider that an actual coverage rate that is higher than nominal coverage is not desired. Actual coverage rates should be examined in conjunction with average half widths since wide confidence intervals tend to increase coverage.

Table 8 shows that, in addition to coverage rates that exceed or are equal to nominal coverage, AUTO and SKF have tighter intervals than the other methods. For NOBM, OBM, and STDS, as the number of batches increase the average half widths decrease.

Table 9 indicates that for this case AUTO and SKF have small values, when compared to the other techniques, for the standard deviation of the half widths. This appears to be a major advantage of these methods. These tables also illustrate a point discussed by Law and Kelton

Table 8. Average Half Width (AR(2) Case 1)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	0.278	0.242	0.226	0.259	0.235	0.223	0.255	0.228	0.203	0.231	0.232	0.220
2560	0.194	0.173	0.164	0.182	0.168	0.161	0.186	0.168	0.156	0.163	0.164	0.157
5120	0.139	0.123	0.118	0.129	0.120	0.116	0.133	0.122	0.115	0.115	0.116	0.111

Table 9. Standard Deviation of Half Widths (AR(2) Case 1)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	0.103	0.060	0.038	0.077	0.050	0.033	0.085	0.052	0.033	0.028	0.025	0.024
2560	0.073	0.042	0.027	0.052	0.034	0.023	0.061	0.038	0.025	0.013	0.012	0.015
5120	0.052	0.030	0.019	0.036	0.024	0.015	0.044	0.028	0.019	0.007	0.006	0.009

[29:563]. Table 7 showed that as batch size increased so did coverage, but Table 9 shows that corresponding to this increase is an inherent instability in the intervals half widths.

The estimated Kalman filter parameters (from method SKF) and their associated standard deviations are listed in Table 10. The estimates for all of the parameters are very close to the values used to generate the data. Due to the lack of measurement noise, $\hat{K}_1$ is near 1.0 and the Kalman filter is placing a high weight on the incoming measurement and virtually disregarding the dynamics model. As sample size increases, the estimates become closer to the theoretical value and their standard deviation goes down. The estimated Q_d/R ratio for this case is approximately 150. The true ratio is 0 since there is no measurement noise R in this case. The estimated ratio is very large because R was estimated slightly higher than 0. As the estimate approaches infinity it indicates the estimated ratio is getting closer to the theoretical ratio.

Table 11 shows the average error of the estimated mean value from the theoretical value is virtually identical for both estimators. The average error is almost always near zero. Table 12 shows the standard deviation decreased as sample size increased. Tables 11 and 12 are indicative of the trends seen in the average errors of the mean estimators and standard deviation of these errors for all of the AR(2) cases. Therefore, for the remaining five AR(2) cases, these two tables are not

Table 10. Estimated Kalman Filter Parameters (AR(2) Case 1)

	Data Size	$\hat{A}$	$\hat{Q}_d$	$\hat{R}$	$\hat{K}_1$	$\hat{K}_2$	$\hat{\phi}_1$	$\hat{\phi}_2$
Mean	1280	0.9983	0.9821	0.0121	0.9879	0.0061	0.5054	0.2957
	2560	0.9995	0.9872	0.0092	0.9908	0.0046	0.5048	0.2977
	5120	0.9999	0.9910	0.0066	0.9934	0.0033	0.5032	0.2988
St. Dev.	1280	0.0408	0.0453	0.0161	0.0161	0.0080	0.0282	0.0273
	2560	0.0283	0.0318	0.0114	0.0114	0.0057	0.0203	0.0191
	5120	0.0195	0.0221	0.0082	0.0082	0.0041	0.0138	0.0129

Table 11. Average Mean Estimation Errors (AR(2) Case 1)

Data Size	$\bar{y}$	MMAE Estimate of μ_y
1280	0.0001	-0.0006
2560	0.0038	0.0035
5120	0.0022	0.0021

presented.

AR(2) Case 2 ($\phi_1 = 0.5, \phi_2 = 0.3, \sigma^2_{pn} = 1.0, \sigma^2_{mn} = 1.0$). The results from this model are presented in Tables 13–16. Table 13 shows that again most of the methods achieved nominal coverage. As expected, coverage generally increases as sample size increases. The same trends for sample size and batch size that were pointed out for Case 1 also are apparent. The coverage rates for NOBM, OBM, and STDS are approximately equal to those obtained for the first case. The coverage rates for AUTO increased with the addition of the measurement noise. A surprising result is that the coverage for SKF and MMAE increased when the measurement noise was added.

The change in coverage rates from Case 1 to this case can be explained further by looking at the average half widths in Table 14. The half widths for NOBM, OBM, and STDS either remained

Table 12. Standard Deviation of Mean Estimation Errors (AR(2) Case 1)

Data Size	$\bar{y}$	MMAE Estimate of μ_y
1280	0.1364	0.1361
2560	0.0985	0.0991
5120	0.0683	0.0683

Table 13. Actual Coverage Rates (AR(2) Case 2)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	0.902	0.897	0.884	0.902	0.886	0.881	0.888	0.881	0.853	0.859	0.907	0.899
2560	0.901	0.896	0.881	0.904	0.886	0.884	0.922	0.885	0.876	0.875	0.909	0.896
5120	0.912	0.903	0.912	0.913	0.912	0.910	0.903	0.907	0.899	0.901	0.933	0.922

Note: With nominal rate of 0.9, estimation accuracy is $\approx \pm 0.016$ for 1000 runs.

Table 14. Average Half Width (AR(2) Case 2)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	0.285	0.248	0.231	0.264	0.240	0.227	0.261	0.233	0.208	0.212	0.247	0.235
2560	0.198	0.176	0.167	0.186	0.171	0.165	0.189	0.171	0.159	0.154	0.176	0.168
5120	0.142	0.125	0.120	0.132	0.122	0.118	0.135	0.124	0.117	0.111	0.125	0.119

the same or increased marginally, SKF and MMAE half widths increased a larger percentage and AUTO half widths actually decreased. This seems to point out the benefit of the Kalman filter techniques to isolate measurement noise from the process noise. The same trends pointed out for the standard deviations of the half widths in Case 1 are evident in Table 15 for this case.

The estimated Kalman filter parameters (from method SKF) and their associated standard deviations are listed in Table 16. The filter tends to underestimate R and overestimate Q_d. The estimated ratio of Q_d/R is approximately 2.5, whereas the true ratio is 1. As expected, with the addition of measurement noise the Kalman filter gain, K, decreases. However, the gain still is

Table 15. Standard Deviation of Half Widths (AR(2) Case 2)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	0.105	0.061	0.039	0.078	0.050	0.033	0.086	0.053	0.034	0.030	0.031	0.030
2560	0.074	0.042	0.028	0.053	0.034	0.023	0.062	0.039	0.025	0.015	0.015	0.015
5120	0.052	0.031	0.019	0.037	0.024	0.016	0.044	0.028	0.019	0.008	0.008	0.007

Table 16. Estimated Kalman Filter Parameters (AR(2) Case 2)

	Data Size	$\hat{A}$	$\hat{Q}_d$	$\hat{R}$	$\hat{K}_1$	$\hat{K}_2$	$\hat{\phi}_1$	$\hat{\phi}_2$
Mean	1280	2.2344	1.4867	0.5975	0.7327	0.0982	0.4654	0.3019
	2560	2.2377	1.4836	0.6015	0.7312	0.0999	0.4691	0.3025
	5120	2.2389	1.4807	0.6046	0.7300	0.1006	0.4698	0.3035
St. Dev.	1280	0.0892	0.1435	0.1119	0.0488	0.0158	0.0440	0.0501
	2560	0.0632	0.0991	0.0776	0.0336	0.0111	0.0321	0.0363
	5120	0.0454	0.0682	0.0541	0.0232	0.0079	0.0232	0.0258

rather large due to the underestimation of R. Furthermore, the addition of measurement noise caused an underestimation of the value of ϕ_1.

AR(2) Case 3 ($\phi_1 = -1.0, \phi_2 = -0.5, \sigma^2_{pn} = 1.0, \sigma^2_{mn} = 0.0$). The results from this model are presented in Tables 17–20. Table 17 shows that virtually all of the methods achieved nominal coverage. There are some interesting comparisons between this case and the two previous cases. First, both SKF and MMAE are obtaining a smaller coverage than before. NOBM and OBM coverage marginally increased at the larger sample sizes and increased even more at the small sample size. In fact, NOBM and OBM had larger or equal actual coverage rates at smaller sample sizes than at larger sample sizes. STDS coverage greatly increased. In the previous cases, as batch size went up so did coverage, whereas in this case it is just the opposite; as batch size is reduced (and number of batches grows) coverage increases. These differences can be attributed to the nature of this particular AR(2) process. As discussed by Box and Jenkins [4:59], an AR(2) process with these autoregressive coefficients exhibit pseudo-periodic behavior and the negative correlation is induced. The random shocks cancel this effect when the observations are summed into batches. Even though SKF and MMAE coverage decreased, near nominal coverage is still achieved.

The unique nature of this AR(2) process is again evident in Tables 18 and 19. The average half widths are much smaller than in the previous cases. For example, the half width for MMAE with a sample size of 5120 is almost 0, yet the coverage is still equal to nominal coverage. These results are caused by the high negative correlation in the output sequence, which significantly reduces the variance of the mean estimates.

Table 17. Actual Coverage Rates (AR(2) Case 3)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	0.917	0.909	0.924	0.921	0.906	0.916	0.915	0.921	0.955	0.896	0.899	0.883
2560	0.902	0.902	0.903	0.915	0.905	0.909	0.914	0.913	0.921	0.891	0.892	0.880
5120	0.903	0.918	0.920	0.919	0.916	0.922	0.916	0.930	0.931	0.915	0.920	0.906

Note: With nominal rate of 0.9, estimation accuracy is $\approx \pm 0.016$ for 1000 runs.

Table 18. Average Half Width (AR(2) Case 3)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	0.023	0.021	0.021	0.022	0.020	0.020	0.022	0.022	0.023	0.018	0.018	0.018
2560	0.016	0.014	0.014	0.015	0.014	0.014	0.016	0.015	0.015	0.013	0.013	0.012
5120	0.011	0.010	0.010	0.010	0.010	0.010	0.011	0.010	0.010	0.009	0.009	0.009

In Table 20 we see that the Kalman filter estimates for $\hat{A}$, $\hat{Q}_d$, $\hat{R}$, $\hat{\mathbf{K}}_1$, and $\hat{\mathbf{K}}_2$ are very similar to those in Case 1 where there also was no measurement noise. The estimated Q_d/R ratio is approximately 500, indicating that the estimated ratio is more accurate than it was in Case 1. The estimates of the ϕ values are also very close to the values used to generate the data.

AR(2) Case 4 ($\phi_1 = -1.0, \phi_2 = -0.5, \sigma^2_{pn} = 1.0, \sigma^2_{mn} = 1.0$). The results from this model are presented in Tables 21–24. The addition of measurement noise to this AR(2) model caused trends different from those seen when measurement noise was added to Case 1. Table 21 shows that actual coverage rates for all of the techniques decreased with the addition of the mea-

Table 19. Standard Deviation of Half Widths (AR(2) Case 3)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	0.009	0.005	0.003	0.006	0.004	0.002	0.007	0.005	0.004	0.001	0.000	0.000
2560	0.006	0.003	0.002	0.004	0.003	0.002	0.005	0.003	0.002	0.001	0.000	0.001
5120	0.004	0.002	0.002	0.003	0.002	0.001	0.004	0.002	0.002	0.000	0.000	0.001

Table 20. Estimated Kalman Filter Parameters (AR(2) Case 3)

	Data Size	$\hat{A}$	$\hat{Q}_d$	$\hat{R}$	$\hat{K}_1$	$\hat{K}_2$	$\hat{\phi}_1$	$\hat{\phi}_2$
Mean	1280	0.9991	0.9906	0.0037	0.9963	-0.0038	-1.0078	-0.5066
	2560	0.9999	0.9938	0.0027	0.9973	-0.0027	-1.0054	-0.5047
	5120	1.0001	0.9956	0.0020	0.9980	-0.0020	-1.0044	-0.5038
St. Dev.	1280	0.0410	0.0420	0.0044	0.0044	0.0044	0.0262	0.0254
	2560	0.0283	0.0292	0.0029	0.0029	0.0029	0.0183	0.0178
	5120	0.0195	0.0200	0.0018	0.0018	0.0018	0.0128	0.0127

Table 21. Actual Coverage Rates (AR(2) Case 4)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	0.906	0.899	0.896	0.913	0.893	0.897	0.892	0.900	0.904	0.897	0.849	0.828
2560	0.897	0.894	0.904	0.901	0.902	0.910	0.895	0.909	0.903	0.914	0.855	0.840
5120	0.893	0.911	0.903	0.901	0.901	0.898	0.902	0.904	0.902	0.905	0.863	0.841

Note: With nominal rate of 0.9, estimation accuracy is $\approx \pm 0.016$ for 1000 runs.

surement noise. As in Case 3, actual coverage rates for NOBM, and OBM generally decreased as sample size increased.

Table 22 shows an increase in the average half widths due to the addition of measurement noise. The decrease in actual coverage rates and increase in half widths indicates that the addition of measurement noise had a detrimental impact on the negatively correlated AR data.

The estimated Kalman filter parameters (from method SKF) and their associated standard deviations are listed in Table 24. As in Case 2, the filter tends to underestimate R and overestimate

Table 22. Average Half Width (AR(2) Case 4)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	0.061	0.054	0.052	0.057	0.052	0.051	0.058	0.054	0.053	0.051	0.045	0.043
2560	0.043	0.038	0.037	0.040	0.037	0.036	0.041	0.038	0.037	0.036	0.032	0.030
5120	0.030	0.027	0.026	0.028	0.026	0.025	0.029	0.027	0.026	0.025	0.022	0.021

Table 23. Standard Deviation of Half Widths (AR(2) Case 4)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	0.022	0.013	0.008	0.015	0.010	0.007	0.018	0.012	0.008	0.004	0.002	0.002
2560	0.015	0.009	0.006	0.011	0.007	0.005	0.013	0.008	0.006	0.002	0.001	0.001
5120	0.011	0.006	0.004	0.008	0.005	0.003	0.009	0.006	0.004	0.001	0.000	0.000

Table 24. Estimated Kalman Filter Parameters (AR(2) Case 4)

	Data Size	$\hat{A}$	$\hat{Q}_d$	$\hat{R}$	$\hat{K}_1$	$\hat{K}_2$	$\hat{\phi}_1$	$\hat{\phi}_2$
Mean	1280	2.5549	1.4754	0.6733	0.7365	-0.1581	-0.9115	-0.4240
	2560	2.5557	1.4721	0.6761	0.7354	-0.1590	-0.9118	-0.4244
	5120	2.5589	1.4703	0.6787	0.7348	-0.1597	-0.9132	-0.4253
St. Dev.	1280	0.1059	0.1645	0.0970	0.0362	0.0189	0.0630	0.0532
	2560	0.0725	0.1159	0.0671	0.0254	0.0132	0.0442	0.0373
	5120	0.0522	0.0792	0.0475	0.0176	0.0091	0.0306	0.0258

Q_d. The estimated Q_d/R ratio of is approximately 2.2; again indicating that when measurement noise is present the ratio is estimated to be over twice it's theoretical value. As expected, with the addition of measurement noise, the Kalman filter gain, K, decreases. However, the gain still is rather large due to the underestimation of R. The addition of measurement noise caused an underestimation of the value of ϕ_1.

AR(2) Case 5 ($\phi_1 = -0.5, \phi_2 = 0.3, \sigma^2_{pn} = 1.0, \sigma^2_{mn} = 0.0$). The results from this model are presented in Tables 25–28. This AR(2) model has actual coverage rates, shown in Table 25, similar to those seen in Case 1.

The average half widths for this case, listed in Table 26 are, smaller than those found in Case 1. According to Box and Jenkins [4:59], an AR(2) process with coefficients similar to this case will exhibit an autocorrelation function that alternates in sign as it damps out. This alternating effect causes a reduction in the overall variance and thus tighter half widths.

The remaining tables for this case demonstrate the same trends as seen in Case 1. In particular, notice the tight half widths and low variability associated with the two Kalman filter

Table 25. Actual Coverage Rates (AR(2) Case 5)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	0.908	0.910	0.903	0.914	0.899	0.895	0.902	0.910	0.909	0.895	0.892	0.876
2560	0.904	0.897	0.891	0.911	0.901	0.898	0.908	0.905	0.901	0.892	0.895	0.882
5120	0.902	0.912	0.916	0.917	0.914	0.914	0.918	0.916	0.910	0.914	0.914	0.901

Note: With nominal rate of 0.9, estimation accuracy is $\approx \pm 0.016$ for 1000 runs.

Table 26. Average Half Width (AR(2) Case 5)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	0.047	0.042	0.040	0.044	0.041	0.039	0.044	0.042	0.041	0.038	0.038	0.036
2560	0.032	0.029	0.028	0.031	0.028	0.028	0.032	0.029	0.028	0.027	0.027	0.026
5120	0.023	0.021	0.020	0.022	0.020	0.020	0.022	0.021	0.020	0.019	0.019	0.018

techniques.

AR(2) Case 6 ($\phi_1 = -0.5, \phi_2 = 0.3, \sigma^2_{pn} = 1.0, \sigma^2_{mn} = 1.0$). The results from this model are presented in Tables 29–32. Table 29 shows that most of the methods achieved nominal coverage. With the addition of the measurement noise most of the actual coverage rates decreased, especially for the large sample size.

The average half widths for this case, in Table 30, have all increased from those in Case 5. The results are similar to those seen in Case 4. The remaining tables for this case show trends exactly like those found in Cases 2 and 4 where measurement noise was also present.

Table 27. Standard Deviation of Half Widths (AR(2) Case 5)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	0.017	0.010	0.007	0.013	0.008	0.005	0.015	0.009	0.006	0.003	0.002	0.002
2560	0.012	0.007	0.005	0.009	0.006	0.004	0.010	0.007	0.004	0.001	0.001	0.002
5120	0.009	0.005	0.003	0.006	0.004	0.003	0.007	0.005	0.003	0.001	0.000	0.001

Table 28. Estimated Kalman Filter Parameters (AR(2) Case 5)

	Data Size	$\hat{A}$	$\hat{Q}_d$	$\hat{R}$	$\hat{K}_1$	$\hat{K}_2$	$\hat{\phi}_1$	$\hat{\phi}_2$
Mean	1280	0.9984	0.9819	0.0123	0.9877	-0.0062	-0.5095	0.2942
	2560	0.9996	0.9875	0.0089	0.9910	-0.0045	-0.5064	0.2970
	5120	0.9999	0.9909	0.0067	0.9933	-0.0034	-0.5050	0.2981
St. Dev.	1280	0.0409	0.0456	0.0162	0.0162	0.0081	0.0296	0.0283
	2560	0.0282	0.0319	0.0114	0.0114	0.0058	0.0208	0.0198
	5120	0.0195	0.0224	0.0082	0.0083	0.0042	0.0142	0.0140

Table 29. Actual Coverage Rates (AR(2) Case 6)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	0.904	0.901	0.897	0.903	0.892	0.904	0.904	0.910	0.900	0.898	0.896	0.884
2560	0.898	0.901	0.906	0.908	0.908	0.908	0.908	0.903	0.910	0.896	0.906	0.886
5120	0.911	0.904	0.899	0.904	0.901	0.898	0.904	0.899	0.906	0.899	0.903	0.883

Note: With nominal rate of 0.9, estimation accuracy is $\approx \pm 0.016$ for 1000 runs.

Table 30. Average Half Width (AR(2) Case 6)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	0.074	0.065	0.063	0.068	0.063	0.061	0.070	0.065	0.063	0.060	0.060	0.057
2560	0.052	0.046	0.044	0.048	0.044	0.043	0.049	0.045	0.044	0.042	0.042	0.040
5120	0.037	0.032	0.031	0.034	0.031	0.031	0.035	0.032	0.031	0.030	0.030	0.029

Table 31. Standard Deviation of Half Widths (AR(2) Case 6)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	0.027	0.016	0.010	0.019	0.012	0.008	0.022	0.014	0.010	0.006	0.003	0.003
2560	0.019	0.011	0.007	0.014	0.009	0.006	0.016	0.010	0.007	0.002	0.002	0.001
5120	0.013	0.008	0.005	0.009	0.006	0.004	0.011	0.007	0.005	0.001	0.001	0.001

Table 32. Estimated Kalman Filter Parameters (AR(2) Case 6)

	Data Size	$\hat{A}$	$\hat{Q}_d$	$\hat{R}$	$\hat{K}_1$	$\hat{K}_2$	$\hat{\phi}_1$	$\hat{\phi}_2$
Mean	1280	2.2366	1.4978	0.5891	0.7367	-0.0981	-0.4701	0.2983
	2560	2.2378	1.4881	0.5976	0.7330	-0.0998	-0.4712	0.3006
	5120	2.2404	1.4876	0.5998	0.7323	-0.1003	-0.4713	0.3020
St. Dev.	1280	0.0904	0.1428	0.1123	0.0484	0.0163	0.0435	0.0491
	2560	0.0631	0.1015	0.0791	0.0343	0.0115	0.0306	0.0347
	5120	0.0452	0.0680	0.0533	0.0229	0.0078	0.0220	0.0259

$M/M/1$ Results. The next results are from the four $M/M/1$ cases shown in Table 4. The results from each of the cases are discussed in the following sections.

$M/M/1$ Case 1 (Average Waiting Time in Queue, $\rho = 0.5$). The results from this model are presented in Tables 33–38. The actual coverage rates, found in Table 33, are all close to nominal but there are several lower than those seen in the AR(2) cases. As mentioned earlier, this is due to the $M/M/1$ queue's inherent statistical complexity. As in the AR(2) Case 1 data, the coverage rates tend to increase as sample size increases and decrease as batch size decreases. NOBM and OBM provide the highest coverage rates, especially at the large sample size. STDS does not appear to perform well for this case, especially at the smaller sample sizes. Once again, however, it is important to note that, within the estimation accuracy many of the methods appear to perform equally well in regards to coverage rates.

The average half widths in Table 34 also show trends similar to those seen in AR(2) Case 1. The half widths get smaller as the sample size increases but larger as the batch size increases. Notice the tight confidence intervals provided by the two Kalman filter techniques. These tight intervals are responsible for the low coverage rates discussed above. The half widths' standard deviations in Table 35 exhibit similar patterns to those in AR(2) Case 1.

The estimated Kalman filter parameters in Table 36 provide considerable insight into the performance of the confidence intervals generated by the SKF technique. The Kalman filter is essentially fitting the $M/M/1$ output to an AR(1) process. It estimates the value of ϕ_1 to be about 0.75 and the value of ϕ_2 to be about 0.025 (essentially zero). As sample size increases, the value of ϕ_1 increases. The estimation has placed all of the noise in the Q_d term. Thus, with very little measurement noise, the Kalman filter almost entirely disregards the dynamics model by estimating the gain value, K_1, to be almost 1.0. The estimated ratio of Q_d/R for this and all of the following $M/M/1$ are above 700, again indicating that the estimation routine has not assigned a significant variance to the lack of fit term R.

The average mean estimation errors of the two mean estimators, see Table 37, offer some insight to the rather poor performance of the MMAE technique. For all sample sizes, we are underestimating the mean value with the MMAE estimator. This will cause a degradation in coverage, even if the variance is correctly estimated, and thus, the half widths are of the correct

Table 33. Actual Coverage Rates ($M/M/1$ Case 1)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	0.872	0.862	0.838	0.860	0.845	0.841	0.843	0.825	0.806	0.843	0.840	0.819
2560	0.888	0.870	0.877	0.893	0.872	0.871	0.858	0.857	0.846	0.874	0.870	0.844
5120	0.892	0.887	0.887	0.891	0.884	0.890	0.881	0.871	0.863	0.877	0.879	0.843

Note: With nominal rate of 0.9, estimation accuracy is $\approx \pm 0.016$ for 1000 runs.

Table 34. Average Half Width ($M/M/1$ Case 1)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	0.137	0.120	0.114	0.127	0.117	0.112	0.126	0.116	0.103	0.112	0.111	0.106
2560	0.102	0.090	0.085	0.095	0.087	0.084	0.094	0.085	0.081	0.081	0.081	0.078
5120	0.074	0.065	0.062	0.069	0.063	0.061	0.070	0.063	0.060	0.058	0.058	0.056

size. This underestimation of the mean is probably caused by an incorrect choice for either the number of filters or spacing of filters and will be addressed inthe MMAE section of this chapter.

Table 38 shows no significant difference between the standard deviation of the mean estimation errors. Both of the standard deviations are very small.

$M/M/1$ Case 2 (Average Waiting Time in Queue, $\rho = 0.8$). The results from this model are presented in Tables 39–44. The increased traffic intensity of 0.8 makes this case extremely complex from a statistical standpoint. The actual coverage rates, see Table 39, reflect this difficulty.

Table 35. Standard Deviation of Half Widths ($M/M/1$ Case 1)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	0.065	0.047	0.041	0.056	0.046	0.039	0.063	0.049	0.037	0.040	0.038	0.036
2560	0.044	0.033	0.027	0.036	0.029	0.026	0.045	0.033	0.027	0.024	0.021	0.025
5120	0.029	0.020	0.016	0.023	0.018	0.015	0.029	0.023	0.018	0.013	0.011	0.013

Table 36. Estimated Kalman Filter Parameters ($M/M/1$ Case 1)

	Data Size	$\hat{A}$	$\hat{Q}_d$	$\hat{R}$	$\hat{K}_1$	$\hat{K}_2$	$\hat{\phi}_1$	$\hat{\phi}_2$
Mean	1280	0.2897	0.2888	0.0006	0.9979	0.0015	0.7435	0.0166
	2560	0.2927	0.2921	0.0004	0.9985	0.0011	0.7479	0.0233
	5120	0.2952	0.2946	0.0004	0.9988	0.0009	0.7515	0.0274
St. Dev.	1280	0.0359	0.0360	0.0008	0.0029	0.0019	0.0559	0.0378
	2560	0.0255	0.0255	0.0003	0.0012	0.0008	0.0390	0.0269
	5120	0.0184	0.0184	0.0002	0.0007	0.0005	0.0282	0.0196

Table 37. Average Mean Estimation Errors ($M/M/1$ Case 1)

Data Size	$\bar{y}$	MMAE Estimate of μ_y
1280	-0.0055	-0.0109
2560	-0.0027	-0.0075
5120	-0.0001	-0.0052

All of the coverages are below nominal coverage and significantly less than the coverages for $M/M/1$ Case 1. A few of the coverage rates are close to nominal, but unlike the coverage rates with AR(2) data, there are many below it. As in the previous $M/M/1$ case, the coverage rates tend to increase as sample size increases and to decrease as batch size decreases. NOBM and OBM again provide the best coverage rates, especially at the large sample size. STDS still does not appear to perform well for $M/M/1$ data, especially at the smaller sample sizes.

The average half widths in Table 40 are much larger than those in $M/M/1$ Case 1, Table 33. This increase in average half widths is due to the increased traffic intensity. The trends in this table are the same as discussed for the previous case.

Again, the Kalman filter is modelling the $M/M/1$ output as an AR(1) process. It estimates

Table 38. Standard Deviation of Mean Estimation Errors ($M/M/1$ Case 1)

Data Size	$\bar{y}$	MMAE Estimate of μ_y
1280	0.0728	0.0772
2560	0.0528	0.0542
5120	0.0381	0.0409

Table 39. Actual Coverage Rates ($M/M/1$ Case 2)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	0.815	0.784	0.730	0.801	0.781	0.735	0.747	0.672	0.531	0.792	0.777	0.659
2560	0.857	0.843	0.814	0.852	0.826	0.802	0.820	0.765	0.674	0.830	0.824	0.716
5120	0.881	0.868	0.844	0.878	0.869	0.850	0.843	0.827	0.762	0.845	0.836	0.711

Note: With nominal rate of 0.9, estimation accuracy is $\approx \pm 0.016$ for 1000 runs.

Table 40. Average Half Width ($M/M/1$ Case 2)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	1.520	1.219	1.009	1.378	1.177	0.992	1.187	0.888	0.595	1.455	1.239	1.001
2560	1.201	1.018	0.884	1.096	0.971	0.866	1.056	0.823	0.621	0.990	0.930	0.851
5120	0.898	0.776	0.713	0.830	0.755	0.700	0.806	0.710	0.576	0.700	0.671	0.626

the value of ϕ_1 to be about 0.95 and the value of ϕ_2 to be about 0.01 (essentially zero). As sample size increases, the value of $\phi(1)$ increases. The estimation routines have again placed all of the noise in the Q_d term and thus the Kalman filter almost entirely disregards the dynamics model. The estimated value of Q_d increased with the increase in traffic intensity. As expected, this indicates greater variability in the output at the higher traffic intensity.

The average mean estimation errors of the MMAE estimator, as seen in Table 43, again show that, for all sample sizes the method is underestimating the mean value. In fact, the underestimation is even more pronounced for this traffic intensity. This will again cause a degradation in coverage.

Table 41. Standard Deviation of Half Widths ($M/M/1$ Case 2)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	1.165	0.755	0.508	0.959	0.701	0.492	0.730	0.374	0.167	2.246	1.100	0.534
2560	0.807	0.592	0.429	0.648	0.526	0.405	0.677	0.398	0.193	0.804	0.622	0.536
5120	0.525	0.391	0.314	0.435	0.357	0.297	0.471	0.342	0.206	0.365	0.318	0.320

Table 42. Estimated Kalman Filter Parameters ($M/M/1$ Case 2)

	Data Size	$\hat{A}$	$\hat{Q}_d$	$\hat{R}$	$\hat{K}_1$	$\hat{K}_2$	$\hat{\phi}_1$	$\hat{\phi}_2$
Mean	1280	1.1188	1.1167	0.0011	0.9990	0.0009	0.9433	0.0046
	2560	1.1232	1.1211	0.0011	0.9990	0.0009	0.9464	0.0086
	5120	1.1198	1.1177	0.0011	0.9990	0.0009	0.9482	0.0109
St. Dev.	1280	0.1213	0.1211	0.0002	0.0002	0.0001	0.0397	0.0319
	2560	0.0832	0.0830	0.0001	0.0001	0.0001	0.0292	0.0226
	5120	0.0584	0.0583	0.0001	0.0000	0.0000	0.0212	0.0161

Table 43. Average Mean Estimation Errors ($M/M/1$ Case 2)

Data Size	$\bar{y}$	MMAE Estimate of μ_y
1280	0.0007	-0.2885
2560	0.0266	-0.2416
5120	0.0129	-0.2206

Table 44 reports the standard deviations of the mean estimation errors. The data indicates that the MMAE estimator has a higher variance than the traditional estimator.

$M/M/1$ Case 3 (Average Number in Queue, $\rho = 0.5$). The results from this model are presented in Tables 45–50. The results in these tables indicate that, with a traffic intensity of 0.5, all of the methods perform the same for this "time-persistent statistic" as they did for the "statistic based or observations" (waiting time) in Case 1.

Table 44. Standard Deviation of Mean Estimation Errors ($M/M/1$ Case 2)

Data Size	$\bar{y}$	MMAE Estimate of μ_y
1280	0.9468	1.3941
2560	0.6827	0.9614
5120	0.4939	0.7796

Table 45. Actual Coverage Rates ($M/M/1$ Case 3)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	0.853	0.843	0.839	0.851	0.840	0.835	0.837	0.833	0.816	0.840	0.843	0.817
2560	0.882	0.877	0.877	0.883	0.873	0.862	0.849	0.859	0.849	0.864	0.861	0.837
5120	0.913	0.889	0.883	0.902	0.897	0.885	0.895	0.871	0.870	0.875	0.875	0.848

Note: With nominal rate of 0.9, estimation accuracy is $\approx \pm 0.016$ for 1000 runs.

Table 46. Average Half Width ($M/M/1$ Case 3)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	0.152	0.134	0.126	0.140	0.130	0.124	0.141	0.126	0.118	0.125	0.124	0.118
2560	0.113	0.099	0.093	0.104	0.095	0.092	0.103	0.095	0.089	0.090	0.090	0.086
5120	0.081	0.071	0.068	0.074	0.069	0.067	0.076	0.069	0.066	0.064	0.064	0.061

Table 47. Standard Deviation of Half Widths ($M/M/1$ Case 3)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	0.072	0.054	0.045	0.060	0.049	0.044	0.070	0.053	0.045	0.046	0.041	0.039
2560	0.048	0.033	0.027	0.038	0.029	0.025	0.045	0.035	0.028	0.023	0.021	0.022
5120	0.032	0.022	0.017	0.024	0.018	0.016	0.030	0.023	0.020	0.013	0.011	0.012

Table 48. Estimated Kalman Filter Parameters ($M/M/1$ Case 3)

	Data Size	$\hat{A}$	$\hat{Q}_d$	$\hat{R}$	$\hat{K}_1$	$\hat{K}_2$	$\hat{\phi}_1$	$\hat{\phi}_2$
Mean	1280	0.6135	0.6098	0.0026	0.9955	0.0028	0.6668	0.0210
	2560	0.6201	0.6176	0.0018	0.9971	0.0019	0.6746	0.0286
	5120	0.6211	0.6190	0.0014	0.9977	0.0015	0.6792	0.0314
St. Dev.	1280	0.0839	0.0847	0.0040	0.0072	0.0044	0.0694	0.0421
	2560	0.0589	0.0591	0.0022	0.0037	0.0023	0.0484	0.0314
	5120	0.0416	0.0417	0.0014	0.0024	0.0016	0.0346	0.0237

Table 49. Average Mean Estimation Errors ($M/M/1$ Case 3)

Data Size	$\bar{y}$	MMAE Estimate of μ_y
1280	-0.0040	-0.0084
2560	-0.0003	-0.0057
5120	-0.0007	-0.0053

Table 50. Standard Deviation of Mean Estimation Errors ($M/M/1$ Case 3)

Data Size	$\bar{y}$	MMAE Estimate of μ_y
1280	0.0823	0.0844
2560	0.0576	0.0620
5120	0.0409	0.0433

$M/M/1$ Case 4 (Average Number in Queue, $\rho = 0.8$). The results from this model are presented in Tables 51- 56. The data in Table 51 indicate that, for a traffic intensity of 0.8, the actual coverage rates for this time-persistent statistic are lower than they are for the statistic based on observations in Case 2. However, the trends in coverage rate among sample size and batch size remain the same. The half widths for average number in queue shown in Table 52 are larger than the half widths for average waiting time in the queue.

The information reported in Table 54 again shows that the SKF method is modeling the $M/M/1$ output as an AR(1) process. It estimates the value of ϕ_1 to be about 0.94 and the value of ϕ_2 to be about 0.01 (essentially zero). As sample size increases, the value of ϕ_1 increases. The Kalman filter has again placed all of the noise in the dynamics noise w_d and estimates no measurement noise v. By estimating the gain value, K_1, to be almost 1.0, the Kalman filter essentially disregards the dynamics model

The average mean estimation errors of the MMAE estimator, found in Table 55, again show that, for all sample sizes the method is underestimating the mean value. In fact, the underestimation is even more pronounced for this statistic than for average waiting time.

Table 51. Actual Coverage Rates ($M/M/1$ Case 4)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	0.798	0.769	0.709	0.777	0.747	0.698	0.744	0.644	0.499	0.769	0.750	0.622
2560	0.831	0.801	0.767	0.823	0.793	0.760	0.781	0.728	0.634	0.795	0.788	0.675
5120	0.853	0.823	0.812	0.842	0.829	0.809	0.805	0.790	0.733	0.808	0.802	0.694

Note: With nominal rate of 0.9, estimation accuracy is $\approx \pm 0.016$ for 1000 runs.

Table 52. Average Half Width ($M/M/1$ Case 4)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	1.589	1.300	1.086	1.453	1.246	1.060	1.307	0.963	0.651	1.565	1.317	1.044
2560	1.255	1.053	0.926	1.161	1.027	0.915	1.081	0.879	0.677	1.046	0.980	0.871
5120	0.926	0.802	0.737	0.854	0.778	0.728	0.844	0.732	0.614	0.729	0.699	0.664

Table 53. Standard Deviation of Half Widths ($M/M/1$ Case 4)

Data	NOBM			OBM			STDS			AUTO	SKF	MMAE
Size	5	10	20	5	10	20	5	10	20			
1280	1.173	0.817	0.562	1.061	0.779	0.551	0.866	0.435	0.195	2.295	1.044	0.552
2560	0.832	0.610	0.456	0.773	0.604	0.453	0.713	0.438	0.225	0.794	0.612	0.492
5120	0.526	0.393	0.321	0.456	0.382	0.318	0.497	0.363	0.229	0.367	0.307	0.339

Table 54. Estimated Kalman Filter Parameters ($M/M/1$ Case 4)

	Data Size	$\hat{A}$	$\hat{Q}_d$	$\hat{R}$	$\hat{K}_1$	$\hat{K}_2$	$\hat{\phi}_1$	$\hat{\phi}_2$
Mean	1280	1.4474	1.4447	0.0014	0.9990	0.0009	0.9363	0.0075
	2560	1.4458	1.4432	0.0014	0.9990	0.0009	0.9395	0.0117
	5120	1.4454	1.4428	0.0014	0.9990	0.0009	0.9421	0.0132
St. Dev.	1280	0.1425	0.1422	0.0004	0.0002	0.0002	0.0422	0.0336
	2560	0.1002	0.0999	0.0002	0.0001	0.0001	0.0319	0.0245
	5120	0.0726	0.0725	0.0001	0.0000	0.0000	0.0220	0.0174

Table 55. Average Mean Estimation Errors ($M/M/1$ Case 4)

Data Size	$\bar{y}$	MMAE Estimate of μ_y
1280	0.0017	-0.3962
2560	0.0022	-0.3415
5120	-0.0070	-0.2174

Table 56. Standard Deviation of Mean Estimation Errors ($M/M/1$ Case 4)

Data Size	$\bar{y}$	MMAE Estimate of μ_y
1280	1.0240	1.3701
2560	0.7611	1.0355
5120	0.5317	0.8013

MMAE Results

Examination of the baseline case for the MMAE estimator indicated that the technique was producing a biased estimate of the mean value and might not be providing an accurate estimate of its variance. This section discusses various attempts to explore and correct these problems.

Evenly-Spaced Filters. The first attempt to improve the MMAE method involved changing the number and spacing of filters. In addition to changes in the number and spacing of the filters, the *a priori* probabilities for the filters were changed to a uniform distribution. This change in the *a priori* probabilities was made for ease in coding the routines. The results, discussed below, indicate this change either caused no change in actual coverage rates or slightly improved them.

As in the baseline case, the spread value corresponds to the amount added to and subtracted from the center filter to determine the filters at each end of the bank. The remaining filters were placed at even increments between these end filters. In order to save computation time, the number of runs was cut to 250. With 250 runs, the estimation accuracy has decreased, and this should be taken into consideration when comparing results in this section to those of the baseline case.

Although the MMAE method was providing nominal coverage for the AR(2) cases, the effect of these changes on MMAEs performance was first tested on an AR(2) case. The AR(2) process from the baseline Case 1 was used. The first results, presented in Tables 57–61, are for 11, 21, and 31, filters all with spread values of $\pm$ 2, 5, and 8 times the variance of the mean estimate, Equation (32).

The results in Table 57 indicate that either 11 or 31 filters with a spread value of 5 or 8 times $\hat{\sigma}_{\bar{y}}$ provide the best coverage rates. This unique pattern of either a small, or large number of filters providing good results was discussed in Chapter IV. With small sample sizes, 11 filters provides the highest coverage. Nominal coverage was not obtained in several cases when 21 filters were used. Nominal coverage was not obtained in some of the cases using 11 or 31 filters at the smallest spread value.

The half widths reported in Table 59 indicate that, when 21 filters are used, the MMAE estimate of the mean value must not be a good estimator. The half widths are almost as large with 21 filters as they are for 11 or 31 filters yet coverage is drastically reduced. The data in Table 59

Table 57. MMAE Actual Coverage Rates (AR(2) Case 1)

Data	Spread ($2\hat{\sigma}_{\bar{y}}$)			Spread ($5\hat{\sigma}_{\bar{y}}$)			Spread ($8\hat{\sigma}_{\bar{y}}$)		
	11	21	31	11	21	31	11	21	31
1280	0.9360	0.7680	0.7720	0.9480	0.8800	0.9400	0.9480	0.8000	0.9320
2560	0.8640	0.8160	0.8000	0.9120	0.8600	0.9040	0.9120	0.8320	0.9000
5120	0.8960	0.7680	0.7840	0.9440	0.9120	0.9360	0.9440	0.8680	0.9320

Table 58. MMAE Average Half Width (AR(2) Case 1)

Data	Spread ($2\hat{\sigma}_{\bar{y}}$)			Spread ($5\hat{\sigma}_{\bar{y}}$)			Spread ($8\hat{\sigma}_{\bar{y}}$)		
	11	21	31	11	21	31	11	21	31
1280	0.2356	0.1954	0.2022	0.2578	0.2308	0.2518	0.2541	0.2110	0.2443
2560	0.1674	0.1373	0.1412	0.1811	0.1655	0.1723	0.1790	0.1567	0.1701
5120	0.1210	0.0960	0.0990	0.1280	0.1195	0.1210	0.1264	0.1151	0.1197

show that the MMAE estimate of the mean value does have larger estimation errors when 21 filters are used. Tables 59 and 61 indicate that with 21 filters, both the standard deviation of the half widths and the standard deviations of the mean estimation errors increase.

After looking at these results, it is obvious that the number of filters and spacing of filters has a significant impact on coverage rates. Since the small number of filters (11) and large number of filters (31) both provided similar coverage, it was decided to explore the use of fewer filters further. Without a truth model, a possible hueristic approach is to use as few filters with as large a total spread as possible without degrading performance. The smaller number of filters will provide computational efficiency and eliminate the problem of unstable probabilities among adjacent filters.

For the rest of the cases, it was decided to base the overall spread of the filters on a multiple

Table 59. MMAE Standard Deviation of Half Widths (AR(2) Case 1)

Data	Spread ($2\hat{\sigma}_{\bar{y}}$)			Spread ($5\hat{\sigma}_{\bar{y}}$)			Spread ($8\hat{\sigma}_{\bar{y}}$)		
	11	21	31	11	21	31	11	21	31
1280	0.0278	0.0382	0.0423	0.0319	0.0385	0.0454	0.0272	0.0434	0.0282
2560	0.0343	0.0267	0.0236	0.0182	0.0182	0.0177	0.0122	0.0230	0.0119
5120	0.0197	0.0175	0.0173	0.0110	0.0139	0.0123	0.0067	0.0133	0.0065

Table 60. MMAE Average Mean Estimation Errors (AR(2) Case 1)

Data	Spread ($2\hat{\sigma}_{\bar{y}}$)			Spread ($5\hat{\sigma}_{\bar{y}}$)			Spread ($8\hat{\sigma}_{\bar{y}}$)		
	11	21	31	11	21	31	11	21	31
1280	-0.0016	0.0352	0.0477	0.0010	0.0217	0.0055	-0.0028	0.0417	0.0005
2560	-0.0161	0.0045	0.0132	0.0026	0.0102	0.0059	0.0023	0.0187	0.0046
5120	-0.0105	0.0025	0.0108	0.0022	0.0049	0.0028	0.0019	0.0092	0.0025

Table 61. MMAE Standard Deviation of Mean Estimation Errors (AR(2) Case 1)

Data	Spread ($2\hat{\sigma}_{\bar{y}}$)			Spread ($5\hat{\sigma}_{\bar{y}}$)			Spread ($8\hat{\sigma}_{\bar{y}}$)		
	11	21	31	11	21	31	11	21	31
1280	0.1338	0.1469	0.1542	0.1371	0.1443	0.1345	0.1362	0.1537	0.1355
2560	0.1078	0.1029	0.1057	0.1046	0.1052	0.0994	0.1008	0.1099	0.0992
5120	0.0774	0.0771	0.0756	0.0685	0.0701	0.0679	0.0686	0.0731	0.0681

of $\hat{\sigma}_y^2$, Equation (24), instead of $\hat{\sigma}_{\bar{y}}$. $\hat{\sigma}_{\bar{y}}$ decreases as sample size increases, therefore, using it may have put the larger sample sizes at a disadvantage. Switching to a near constant spread between sample sizes also made comparisons easier. The variance of y was used instead of the standard deviation of y. This will cause the spread to increase further for processes with a great deal of variability (e.g. the $M/M/1$ queue), than for inherently stable processes (i.e. AR(2) processes).

Since the $M/M/1$ queue is widely used and much more statistically complex that the AR(2) process, $M/M/1$ data was used for the additional trials. Specifically, $M/M/1$ Case 1 (average waiting time in queue, $\rho = 0.8$) was used. The first $M/M/1$ MMAE trial used 5, 9, and 13 filters with spread values of $\pm$ 0.1, 0.3, and 0.5 times $\hat{\sigma}_y^2$.

Table 62 shows the actual coverage rates for the first $M/M/1$ MMAE trial. The table clearly

Table 62. MMAE Actual Coverage Rates ($M/M/1$ Case 2)

Data	Spread ($0.1\hat{\sigma}_y^2$)			Spread ($0.3\hat{\sigma}_y^2$)			Spread ($0.5\hat{\sigma}_y^2$)		
Size	5	9	13	5	9	13	5	9	13
1280	0.5120	0.6120	0.6000	0.5640	0.5360	0.6880	0.5560	0.5400	0.7360
2560	0.5920	0.6720	0.6360	0.5840	0.5320	0.6920	0.5720	0.5280	0.7520
5120	0.6480	0.6640	0.7280	0.3720	0.5320	0.7960	0.3760	0.5360	0.8240

Table 63. MMAE Average Half Width ($M/M/1$ Case 2)

Data	Spread ($0.1\hat{\sigma}_y^2$)			Spread ($0.3\hat{\sigma}_y^2$)			Spread ($0.5\hat{\sigma}_y^2$)		
Size	5	9	13	5	9	13	5	9	13
1280	0.9146	1.0350	1.1175	1.0987	1.0638	1.4741	1.1002	1.0391	1.5924
2560	0.8875	0.8827	0.9366	0.8374	0.7451	1.0647	0.8667	0.7192	1.1279
5120	0.7071	0.6264	0.6770	0.3937	0.5498	0.8247	0.3944	0.5418	0.8103

Table 64. MMAE Standard Deviation of Half Widths ($M/M/1$ Case 2)

Data	Spread ($0.1\hat{\sigma}_y^2$)			Spread ($0.3\hat{\sigma}_y^2$)			Spread ($0.5\hat{\sigma}_y^2$)		
Size	5	9	13	5	9	13	5	9	13
1280	0.6912	0.8044	0.9232	0.6939	0.8387	1.1124	0.7944	0.8635	1.1546
2560	0.6026	0.5288	0.6824	0.8020	0.5410	0.6189	0.8435	0.5554	0.6470
5120	0.4715	0.4152	0.3136	0.6259	0.4383	0.4402	0.6247	0.4502	0.4630

indicates that all of the actual coverage rates are significantly below the nominal rate. The table indicates that the actual coverage rates for each sample size are highest when the number of filters was equal to 13. When using 13 filters, coverage improved as the spread increased. As was expected, the switch to $\hat{\sigma}_y^2$ allows for better coverage as sample size increases. The only exception was when the filters were spaced too far apart (e.g. spacing of 0.3 $\hat{\sigma}_y^2$ with only 5 filters). This indicates that in these situations the probabilities are concentrating on one or two filters and the variance is underestimated.

Table 63 indicates that, as the coverage went up, so did the half widths. Table 64 indicates that as sample size increased, the standard deviation of the half widths decreased. A very positive result indicated in this chart is seen by examining the standard deviation of the half widths for sample size of 5120, 13 filters, with a spread of $0.5\hat{\sigma}_y^2$. The table indicates that, in addition to having the highest actual coverage rate (in Table 62), it also has the smallest standard deviation. This result seems corroborate the discussions from Chapter IV, that there is an optimal approach for spacing the filters that will offer actual coverage near nominal due to accurate and stable estimates of the variance.

Table 65 indicates that the problem of poor estimates of the mean value as seen in the baseline MMAE approach above can be eliminated with an appropriate combination of number of filters

Table 65. MMAE Average Mean Estimation Errors ($M/M/1$ Case 2)

Data	Spread ($0.1\hat{\sigma}_y^2$)			Spread ($0.3\hat{\sigma}_y^2$)			Spread ($0.5\hat{\sigma}_y^2$)		
Size	5	9	13	5	9	13	5	9	13
1280	-0.6217	-0.1601	-0.0880	-0.4339	0.3208	0.2916	-0.7093	0.2682	0.1755
2560	-0.5140	-0.0612	0.0098	-0.2838	0.2471	0.1653	-0.3807	0.2222	0.0501
5120	-0.4333	0.0269	0.1083	-0.0655	0.2943	0.0956	-0.0780	0.2795	0.0097

Table 66. MMAE Standard Deviation of Mean Estimation Errors ($M/M/1$ Case 2)

Data	Spread ($0.1\hat{\sigma}_y^2$)			Spread ($0.3\hat{\sigma}_y^2$)			Spread ($0.5\hat{\sigma}_y^2$)		
Size	5	9	13	5	9	13	5	9	13
1280	1.0834	1.0352	1.1537	1.6010	1.4411	1.3428	1.5160	1.3676	1.2529
2560	0.9289	0.8345	0.8988	1.1833	0.9934	1.1409	1.1984	0.9649	1.0658
5120	0.8260	0.5782	0.6909	0.6297	0.6710	0.6821	0.6321	0.6651	0.5620

and total spread. Not surprisingly, the best estimate of the mean corresponds to the case with the highest actual coverage. This table also points out that, if the filters are too closely spaced, especially for small sample sizes, the mean is consistently underestimated by MMAE. Table 66 indicates that, as the MMAE mean estimation errors improved, the standard deviation of the errors was lowered.

The first $M/M/1$ MMAE trial indicated that there was a significant potential for achieving good coverage rates using the MMAE approach with evenly-spaced filters. From the results of the first trial an indication of what spacing and number of filters worked best was obtained, and several more trials were run. The results of a representative case are presented and discussed below.

This trial used 9, 11, and 13 filters with spreads of 0.8, 1.0, and 1.2 times $\hat{\sigma}_y^2$. Table 67 indicates the actual coverage rates have greatly improved, when compared to the baseline MMAE case, for all sample sizes.

Tables 68 and 69 present the average half widths and their associated standard devation for this $M/M/1$ MMAE trial. These half widths and standard deviations indicate that the MMAE approach is constructing confidence intervals with half widths and standard devations similar to those constructed by the widely applied techniques used in the baseline case. These results are extremely encouraging, considering the fact that these MMAE results are simply from a "guessed"

Table 67. MMAE Actual Coverage Rates ($M/M/1$ Case 2)

Data	Spread ($1.2\hat{\sigma}_y^2$)			Spread ($1.4\hat{\sigma}_y^2$)			Spread ($1.6\hat{\sigma}_y^2$)		
	9	11	13	9	11	13	9	11	13
1280	0.6640	0.7120	0.7160	0.6760	0.7480	0.7560	0.6760	0.7480	0.7560
2560	0.7640	0.7560	0.7560	0.7400	0.7720	0.7800	0.7400	0.7720	0.7800
5120	0.7960	0.8200	0.8560	0.8120	0.8400	0.8640	0.8120	0.8400	0.8640

Table 68. MMAE Average Half Width ($M/M/1$ Case 2)

Data	Spread ($1.2\hat{\sigma}_y^2$)			Spread ($1.4\hat{\sigma}_y^2$)			Spread ($1.6\hat{\sigma}_y^2$)		
	9	11	13	9	11	13	9	11	13
1280	1.1725	1.4143	1.3773	1.2782	1.4998	1.4064	1.2783	1.4997	1.4062
2560	1.0562	1.1093	1.1180	0.9816	1.0917	1.0409	0.9816	1.0915	1.0407
5120	0.7168	0.7959	0.7888	0.7186	0.7309	0.7347	0.7190	0.7310	0.7347

combination of filters and spacing and may not be the best possible MMAE results. As expected from the increased coverage rates, Tables 70 and 71 indicate that the MMAE mean estimation error is better than in the baseline case as is its associated standard deviation.

Table 69. MMAE Standard Deviation of Half Widths ($M/M/1$ Case 2)

Data	Spread ($1.2\hat{\sigma}_y^2$)			Spread ($1.4\hat{\sigma}_y^2$)			Spread ($1.6\hat{\sigma}_y^2$)		
	9	11	13	9	11	13	9	11	13
1280	0.6629	1.0112	0.9673	0.7248	1.0386	0.9439	0.7244	1.0383	0.9434
2560	0.6330	0.7608	0.8289	0.5430	0.7120	0.6645	0.5427	0.7119	0.6638
5120	0.4006	0.5978	0.4334	0.4311	0.3166	0.3076	0.4331	0.3170	0.3075

Table 70. MMAE Average Mean Estimation Errors ($M/M/1$ Case 2)

Data	Spread ($1.2\hat{\sigma}_y^2$)			Spread ($1.4\hat{\sigma}_y^2$)			Spread ($1.6\hat{\sigma}_y^2$)		
	9	11	13	9	11	13	9	11	13
1280	-0.5462	0.1647	0.1329	-0.4432	-0.0311	-0.2189	-0.4425	-0.0307	-0.2184
2560	-0.2232	0.2040	0.1871	-0.2994	-0.0634	-0.1586	-0.2994	-0.0638	-0.1588
5120	-0.0729	0.1226	0.1127	-0.0650	0.0066	-0.0025	-0.0648	0.0066	-0.0026

Table 71. MMAE Standard Deviation of Mean Estimation Errors ($M/M/1$ Case 2)

Data	Spread ($1.2\hat{\sigma}_y^2$)			Spread ($1.4\hat{\sigma}_y^2$)			Spread ($1.6\hat{\sigma}_y^2$)		
	9	11	13	9	11	13	9	11	13
1280	1.4064	1.4781	1.3760	1.4698	1.3623	1.0349	1.4695	1.3617	1.0342
2560	0.8606	1.1794	1.1047	1.0206	1.1036	0.8986	1.0207	1.1040	0.8994
5120	0.6236	0.7370	0.7146	0.5202	0.5350	0.4911	0.5188	0.5348	0.4912

Normally-Spaced Filters. As discussed in Chapter III, the usual approach to constructing confidence intervals (and the one used for all the MMAE intervals so far) involves the use of a multiplier. The multiplier is based on the assumed probability distribution of the parameter. That multiplier is often a t-critical value. All of the MMAE confidence intervals presented to this point were constructed with the use of a t-critical multiplier. The use of a t-critical value assumes that the underlying distribution of the sample is normally distributed. In the MMAE technique, the sample is the set of filter means and their associated final probabilities. This sample may or may not follow a normal distribution. Therefore, it was decided to analyze the final MMAE filter probabilities.

The figures that follow show the final MMAE filter probabilities for a sample size of 5120 (since this sample size usually provided the best coverage) for the three MMAE trials discussed in the previous section. There are three curves displayed on each graph; each curve corresponds to one of the three choices for number of filters used. For each case there are three figures, one for each spread value. By looking at these figures, and comparing the ones which provided good versus "not so good" coverage, some unique insights into the MMAE approach can be seen. However, each of these curves represents the final probabilities of one single run (one representation of the stochastic process) and not the aggregate final probabilities over all runs. Thus, these curves may not be indicative of the average final probabilities, but hopefully insights could be made as to a good spacing of the filters.

Figures 2-4 are from the AR(2) Case 1, MMAE results obtained in the previous section. Figure 2 shows the final probabilities for 11, 21, and 31 filters placed evenly apart with a total spread of $\pm 2\hat{\sigma}_{\bar{y}}$ from the MMAE estimate of the mean. Table 57 indicates that the use of 11 filters provided the highest coverage for this spread. The figure shows that the curve representing 11 filters does appear to look normally shaped. The other two curves appear to be skewed. It is interesting to note that no single filter received a final probability greater that approximately 0.16.

Figure 3 shows the final probabilities for 11, 21, and 31 filters placed so as to cover a range of $\pm 5\hat{\sigma}_{\bar{y}}$ from the average value of the simulation observations. Table 57 indicates that, for this spread, all three number of filter selections provided nominal coverage. The use of both 11 and 31 filters provided extremely high coverage The figure illustrates that, by setting the low and high filters at approximately 49.9 and 50.2 respectively, as was done when the spread was set to $\pm 2\hat{\sigma}_{\bar{y}}$

Figure 2. MMAE Final Filter Probabilities (AR(2) Case 1, spread = $\pm 2\hat{\sigma}_{\tilde{y}}$)

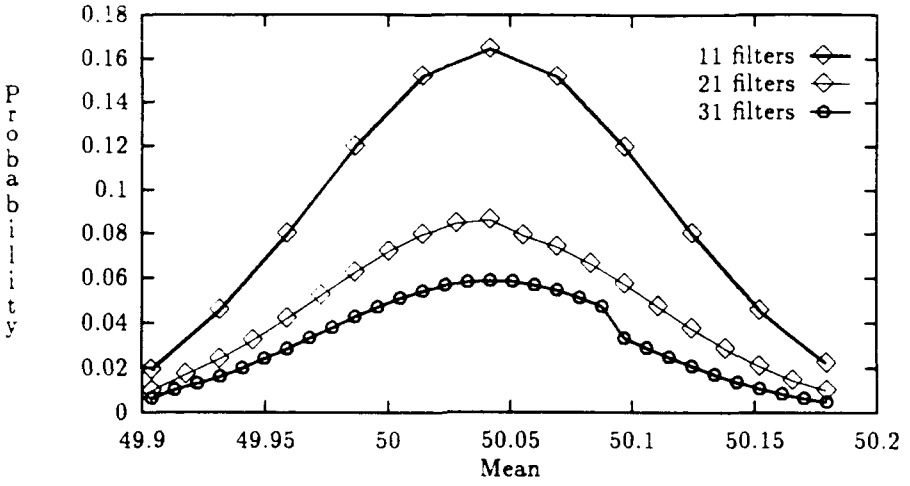
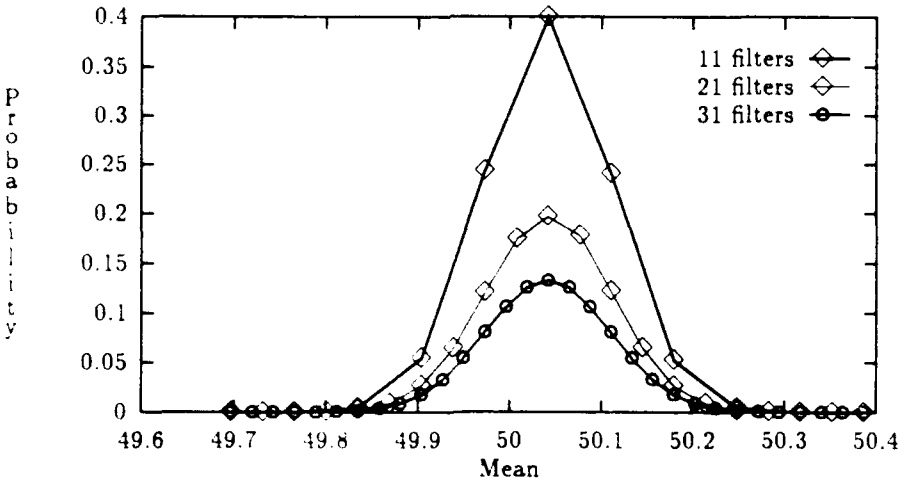


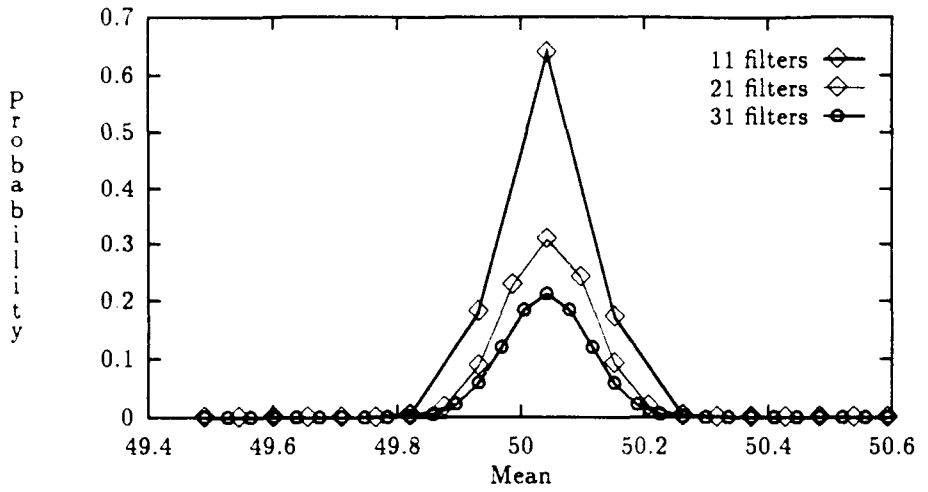
Figure 3. MMAE Final Filter Probabilities (AR(2) Case 1,spread = $\pm 5\hat{\sigma}_{\tilde{y}}$)



in Figure 2, the tails of the distribution are cut off and an artificial upper bound is imposed on the variance estimate. This phenomenon was discussed in Chapter IV.

Figure 3 illustrates that, when the spread is set too large, many of the filters receive a final

Figure 4. MMAE Final Filter Probabilities (AR(2) Case 1, spread = $\pm 8\hat{\sigma}_{\tilde{y}}$)



probability of nearly zero and offer no contribution to the MMAE estimates of the mean and its associated variance.

Figures 5–7 are from the first MMAE trial of $M/M/1$ Case 2, results obtained in the previous section. Figure 5 shows the final probabilities for 5, 9, and 13 filters spaced evenly apart with a total spread of $\pm 0.1\hat{\sigma}_{\tilde{y}}^2$ from the MMAE estimate of the mean. Table 62 indicates that none of these combinations of filters and spacing provided near nominal coverage. The use of 13 filters provided the highest coverage for this spread. The figure shows that the filter probabilities do not appear normally spaced, and the tails may be cut off.

Figure 6 shows the final probabilities for the same number of filters as above with a total spread of $\pm 0.3\hat{\sigma}_{\tilde{y}}^2$ from the MMAE estimate of the mean. This figure clearly illustrates, as does Figure 7, that the use of 5 filters for these spread values is not enough filters. The final probability for the center filter is almost one in both cases. This will certainly cause a loss of information in estimating the variance.

Figures 8–10 are from the second $M/M/1$ MMAE trial case presented in the previous section. Table 67 presents the actual coverage rates for this case, and it indicates that many of the actual coverage rates were near nominal for this case. In all three cases, 13 filters provided the best

Figure 5. MMAE Final Filter Probabilities ($M/M/1$ Case 2, Spread = $\pm 0.1\hat{\sigma}_y^2$)

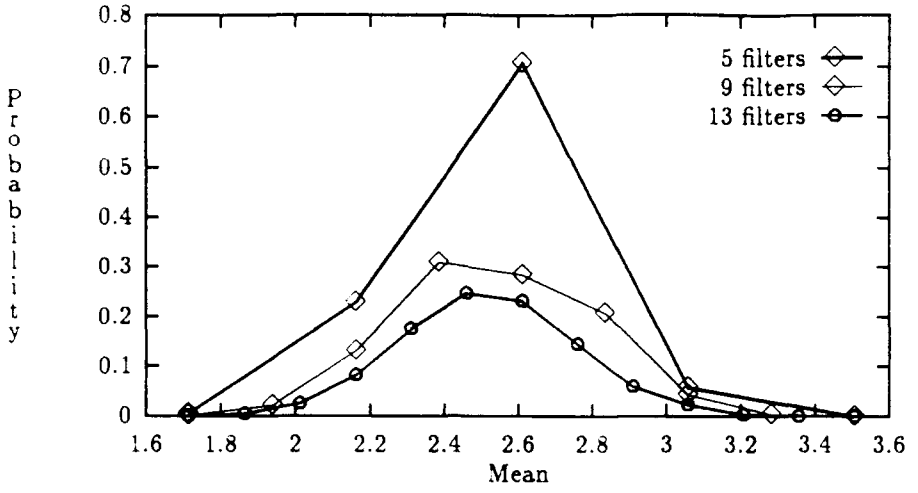
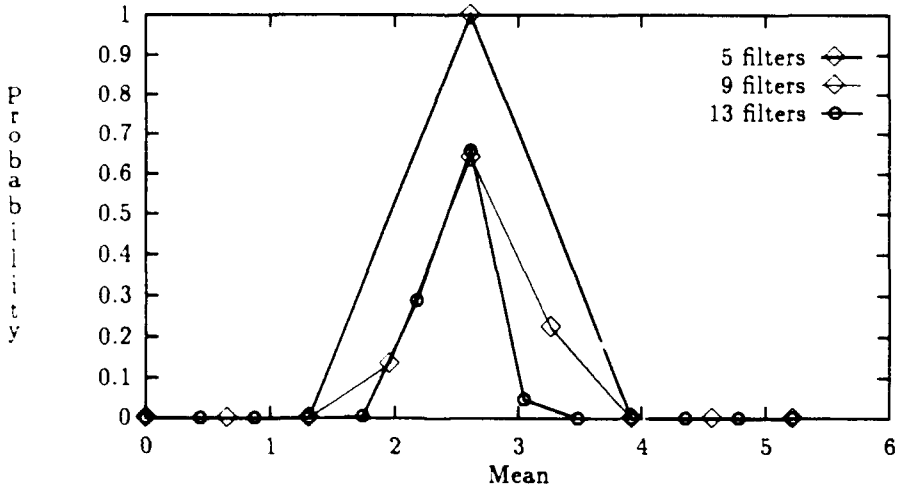


Figure 6. MMAE Final Filter Probabilities ($M/M/1$ Case 2, Spread = $\pm 0.3\hat{\sigma}_y^2$)



coverage. All three of these figures illustrate that the MMAE method appears to work best when one filter has a large (above 0.6) final probability, and some of the surrounding (those nearest the one with the large probability) filters end up with smaller, but significant probabilities.

Figure 7. MMAE Final Filter Probabilities ($M/M/1$ Case 2, Spread = $\pm 0.5\hat{\sigma}_y^2$)

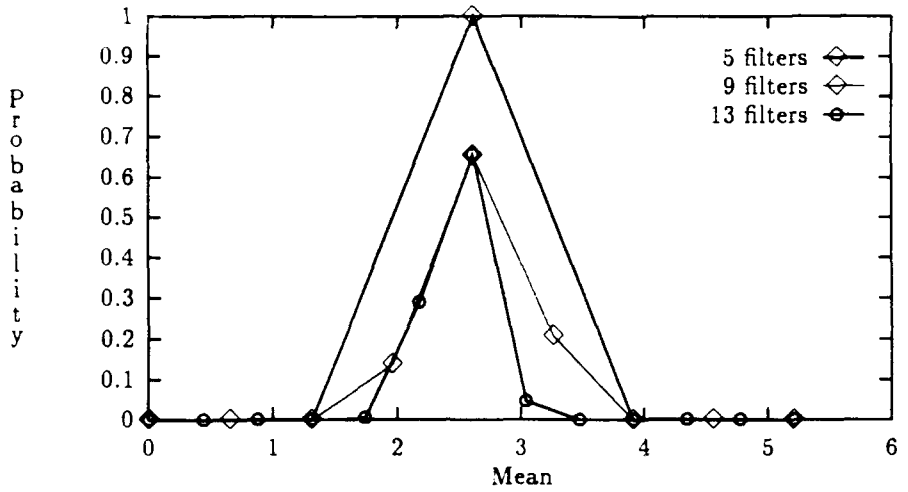
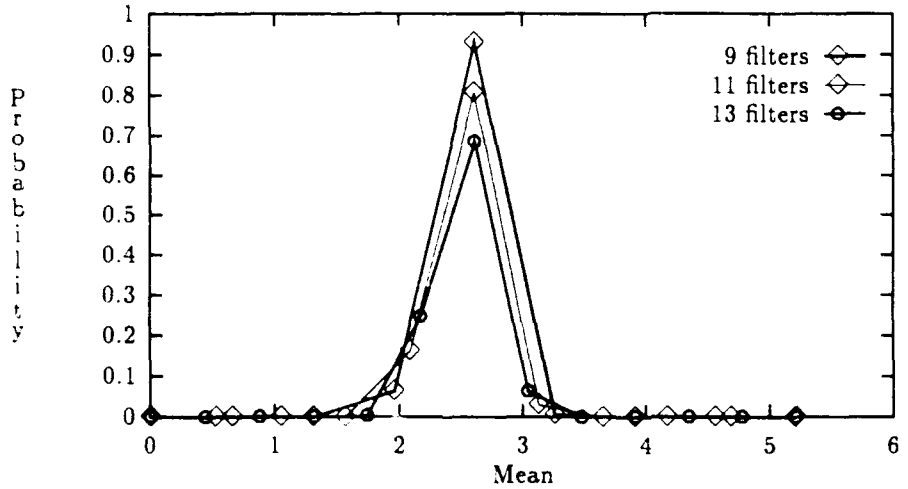


Figure 8. MMAE Final Filter Probabilities ($M/M/1$ Case 2, Spread = $\pm 1.2\hat{\sigma}_y^2$)



The curves in Figures 8–10 appeared to be converging to a normal distribution as the spread and number of filters increased. However, with evenly-spaced filters, many of the end filters were ending up with a probability of nearly zero. In the above cases, the filters were evenly spaced in

Figure 9. MMAE Final Filter Probabilities ($M/M/1$ Case 2, Spread = $\pm 1.4\hat{\sigma}_y^2$)

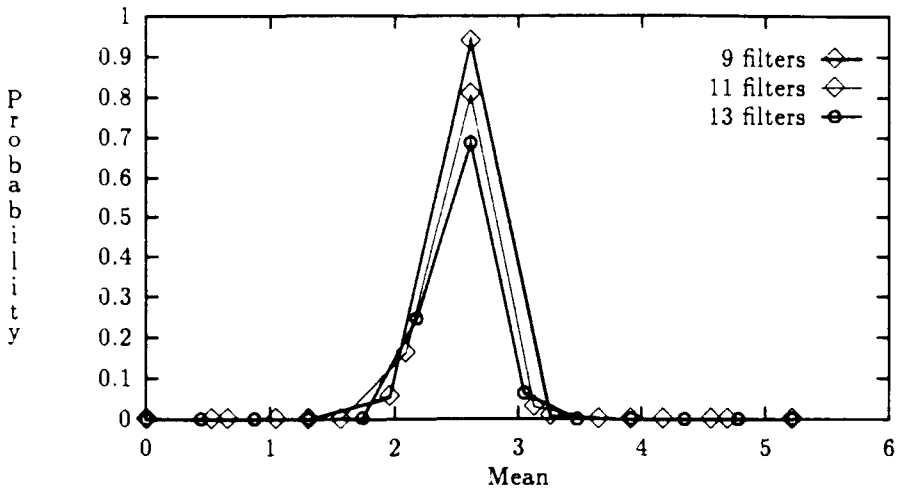
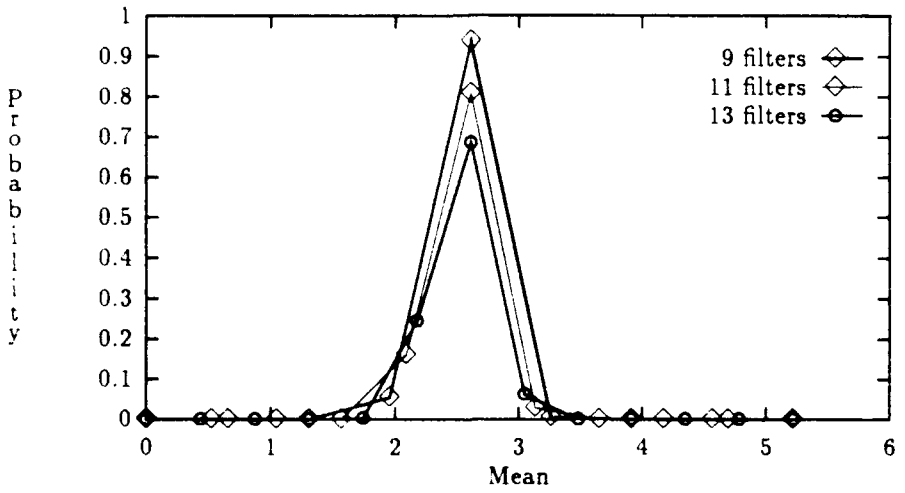
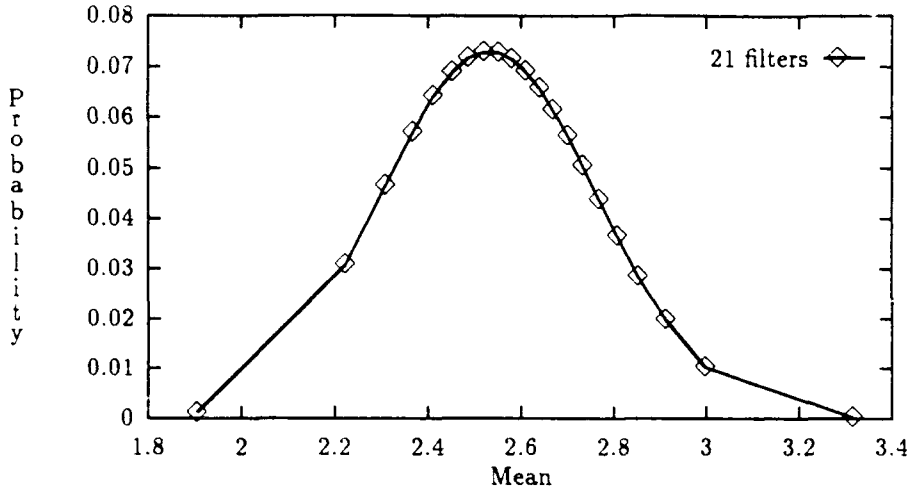


Figure 10. MMAE Final Filter Probabilities ($M/M/1$ Case 2, Spread = $\pm 1.6\hat{\sigma}_y^2$)



an attempt to obtain final probabilities that appeared normally distributed. One other approach is to space the filters at uneven amounts, using an inverse cumulative distribution function for the t distribution to determine the appropriate spacing. The final probabilities should appear normal

Figure 11. MMAE Final Filter Probabilities ($M/M/1$ Case 2, spread = normal)



and a t-critical value would be appropriate.

Figure 11 shows the final probabilities for 21 filters normally spaced for a sample size of 5120, using data from $M/M/1$ Case 2. The end filters were placed at $\pm 3\hat{\sigma}_{\bar{y}}$ from the average of the simulation observations. The curve appears to follow a normal distribution, although the difference between filter parameters is very small. It is interesting to note that the filters are not evenly divided on either side of the curve's peak value. This might indicate that the MMAE filter probabilities are still in the process of stabilizing and the current MMAE estimate of the mean will be low; the theoretical mean value for this case is 3.2. If the MMAE filter probabilities are still stabilizing, it either means that we still have transient data, or that the MMAE technique simply requires a large sample size.

After reviewing Figure 11, it was decided to run 1000 replications to analyze the results. Table 72 indicates that, with these selections for total spread and number of filters, the normal spacing technique does not work well. The coverage rates reported in the table are all well below nominal. The half widths appear to be underestimated, as does the mean value of the process. However, these results are only for one combination of total spread and number of filters, and better results might be obained with other combinations.

Table 72. Results From Normal Spacing of Filters

Data Size	Coverage Rates	Average Half Width	Standard Deviation of Half Widths	Mean Estimation Error	Standard Deviation of Error
1280	0.5760	0.8004	0.4834	-0.2624	2.2292
2560	0.5880	0.6652	0.5179	-0.2709	0.9143
5120	0.5750	0.4692	0.3015	-0.2475	0.6513

Table 73. MMAE PI Actual Coverage Rates ($M/M/1$ Case 2)

Data	Spread ($1.2\hat{\sigma}_y^2$)			Spread ($1.4\hat{\sigma}_y^2$)			Spread ($1.6\hat{\sigma}_y^2$)		
	12	13	15	12	13	15	12	13	15
1280	0.7400	0.7760	0.7560	0.7640	0.7880	0.7920	0.7720	0.7880	0.7920
2560	0.8520	0.8680	0.8360	0.8360	0.8720	0.8560	0.8360	0.8760	0.8600
5120	0.9360	0.9520	0.9240	0.9440	0.9520	0.9280	0.9440	0.9520	0.9280

Probabilistic Approach. One other approach for using an MMAE model was discussed in Chapter IV. This approach involved running a simulation output process through a bank of Kalman filters and then using the final MMAE probabilities as a discrete approximation of the underlying distribution. By backing off five percent from each end of the bank one will obtain a "probabilistic interval" containing 90 percent of the final MMAE probabilities. This method will probably require more filters than the original MMAE method to reduce interpolation between the discrete filters which will serve as end points.

The intervals constructed using this technique will not be symmetric. Choobineh and Ballard reported good results for a technique that also produced asymmetric confidence intervals [6].

An initial trial of this method was run using 12, 13, and 15 filters spaced evenly apart with a total spread of $\pm$ 1.2, 1.4, and 1.6 times $\hat{\sigma}_y^2$. Table 73 shows the actual coverage rates for this MMAE PI method. The coverage rates are very high, especially for large sample size, for $M/M/1$ data with a traffic intensity of 0.8.

Table 74 shows the average half widths for the first MMAE PI trial. As expected, the high coverage rates are direct results of very large interval widths. However, Tables 73 and 74 indicate that this method has the potential to construct meaningful confidence intervals. After running several more cases some excellent results were obtained. One representative case is discussed in the following paragraphs.

Tables 75–80 present the results of the MMAE PI technique using 18, 21, and 24 filters evenly spaced at $\pm$ 1.2, 1.4, and 1.6 times $\hat{\sigma}_y^2$. Table 75 indicates that for a sample size of 5120 nominal coverage was obtained in almost every case.

Table 74. MMAE PI Average Half Width ($M/M/1$ Case 2)

Data	Spread ($1.2\hat{\sigma}_y^2$)			Spread ($1.4\hat{\sigma}_y^2$)			Spread ($1.6\hat{\sigma}_y^2$)		
	12	13	15	12	13	15	12	13	15
1280	1.3516	1.4940	1.4583	1.4668	1.5024	1.4781	1.4655	1.5056	1.4790
2560	1.2435	1.2399	1.2415	1.1961	1.2046	1.1758	1.1916	1.2092	1.1761
5120	1.0044	1.0282	0.9740	1.0087	1.0167	0.9453	1.0087	1.0167	0.9454

Table 75. MMAE PI Actual Coverage Rates ($M/M/1$ Case 2)

Data	Spread ($1.2\hat{\sigma}_y^2$)			Spread ($1.4\hat{\sigma}_y^2$)			Spread ($1.6\hat{\sigma}_y^2$)		
	18	21	24	18	21	24	18	21	24
1280	0.7040	0.7240	0.7280	0.7200	0.7640	0.7640	0.7240	0.7640	0.7640
2560	0.8360	0.8160	0.8040	0.8080	0.8400	0.8280	0.8080	0.8400	0.8280
5120	0.9000	0.8880	0.8920	0.9000	0.9080	0.8960	0.9000	0.9080	0.8960

The average half widths are found in Table 76. This table indicates that, for large sample size, this method results in intervals with tight half widths.

Table 77 lists the standard deviation of the half widths for this technique. Again, for large sample sizes, the results are excellent. The combination of results in Tables 75–77 clearly indicate that there is great potential in this method. For example, in the baseline Monte Carlo analysis, NOBM with 5 batches and a sample size of 5120 resulted in the highest coverage. This NOBM coverage was 0.881, with an average half width of 0.898, and a standard deviation of 0.525. The MMAE probabilistic approach offers better results for all three of these performance criteria. The MMAE PI technique with 24 filters evenly spaced around the center filter with a total spread of $1.6\hat{\sigma}_y^2$ offered a higher coverage of 0.896, with a smaller average half width of 0.827, and a lower

Table 76. MMAE PI Average Half Width ($M/M/1$ Case 2)

Data	Spread ($1.2\hat{\sigma}_y^2$)			Spread ($1.4\hat{\sigma}_y^2$)			Spread ($1.6\hat{\sigma}_y^2$)		
	18	21	24	18	21	24	18	21	24
1280	1.2242	1.3775	1.3607	1.3246	1.4462	1.3779	1.3228	1.4467	1.3784
2560	1.1287	1.1479	1.1207	1.0676	1.1344	1.0745	1.0664	1.1343	1.0750
5120	0.8971	0.8997	0.8588	0.9034	0.8606	0.8287	0.9004	0.8567	0.8274

Table 77. MMAE PI Standard Deviation of Half Widths ($M/M/1$ Case 2)

Data	Spread ($1.2\hat{\sigma}_y^2$)			Spread ($1.4\hat{\sigma}_y^2$)			Spread ($1.6\hat{\sigma}_y^2$)		
	18	21	24	18	21	24	18	21	24
1280	0.6222	0.8644	0.8606	0.7031	0.9097	0.8373	0.7043	0.9099	0.8370
2560	0.4950	0.6531	0.6921	0.4756	0.6591	0.5842	0.4755	0.6592	0.5857
5120	0.3013	0.4445	0.3372	0.3186	0.3154	0.2821	0.3098	0.3046	0.2801

Table 78. MMAE PI Average Upper Bound ($M/M/1$ Case 2)

	Data	Spread ($1.2\hat{\sigma}_y^2$)			Spread ($1.4\hat{\sigma}_y^2$)			Spread ($1.6\hat{\sigma}_y^2$)		
		18	21	24	18	21	24	18	21	24
Mean	1280	3.9638	4.7427	4.7044	4.1747	4.5541	4.3947	4.1753	4.5549	4.3960
	2560	4.1588	4.5251	4.5180	4.0382	4.2340	4.1362	4.0372	4.2343	4.1375
	5120	4.0610	4.2098	4.1646	4.0574	4.0619	4.0280	4.0578	4.0623	4.0283
St. Dev.	1280	1.7264	1.8771	1.8471	1.7955	1.9232	1.7299	1.7934	1.9214	1.7282
	2560	1.1240	1.3820	1.4014	1.2936	1.4338	1.2968	1.2950	1.4337	1.2982
	5120	0.7258	0.9863	0.9349	0.7018	0.7521	0.7192	0.7021	0.7523	0.7192

standard deviation of 0.280.

Tables 78, 79, 80 present the MMAE PI average upper bound, lower bound, and mean value, respectively, along with the associated standard deviations. These tables indicate the asymmetric nature of the intervals produced with this technique.

Table 79. MMAE PI Average Lower Bound ($M/M/1$ Case 2)

	Data	Spread ($1.2\hat{\sigma}_y^2$)			Spread ($1.4\hat{\sigma}_y^2$)			Spread ($1.6\hat{\sigma}_y^2$)		
		18	21	24	18	21	24	18	21	24
Mean	1280	1.5153	1.9876	1.9831	1.5254	1.6617	1.6389	1.5297	1.6614	1.6392
	2560	1.9013	2.2292	2.2766	1.9031	1.9653	1.9872	1.9044	1.9657	1.9874
	5120	2.2668	2.4104	2.4471	2.2506	2.3407	2.3705	2.2570	2.3489	2.3734
St. Dev.	1280	1.1018	1.3053	1.2614	0.9811	0.6948	0.6172	0.9688	0.6932	0.6166
	2560	0.6646	1.0495	1.0944	0.6950	0.6201	0.6274	0.6950	0.6202	0.6276
	5120	0.4517	0.5397	0.5257	0.4351	0.3646	0.3460	0.4141	0.3335	0.3409

Table 80. MMAE PI Average Estimate of the Mean Value ($M/M/1$ Case 2)

	Data	Spread ($1.2\hat{\sigma}_y^2$)			Spread ($1.4\hat{\sigma}_y^2$)			Spread ($1.6\hat{\sigma}_y^2$)		
		18	21	24	18	21	24	18	21	24
Mean	1280	2.7128	3.3608	3.3441	2.8276	3.0790	2.9890	2.8265	3.0797	2.9900
	2560	3.0349	3.3891	3.3930	2.9420	3.0826	3.0456	2.9419	3.0829	3.0457
	5120	3.1471	3.3312	3.3143	3.1603	3.2098	3.2016	3.1606	3.2102	3.2015
St. Dev.	1280	1.3762	1.4404	1.3841	1.3465	1.1793	1.0408	1.3433	1.1751	1.0375
	2560	0.8076	1.1336	1.0938	0.9937	0.9572	0.8934	0.9937	0.9547	0.8915
	5120	0.5905	0.7383	0.7099	0.4930	0.5106	0.4898	0.4913	0.5123	0.4895

Summary

This chapter has shown results that clearly indicate a great potential for confidence interval construction techniques based on the use of the Kalman filters. In particular, the MMAE PI technique, when used with the large sample size, was shown to outperform all of the baseline techniques. All of the Kalman filter techniques produced confidence intervals with, when compared to the currently used techniques, low standard deviations of half widths.

The estimation routine was able to identify the presence of measurement noise in the AR(2) cases when it was present. However, it underestimated the magnitude of the measurement noise's variance. For $M/M/1$ data the estimation routine placed all of the noise in the Q_d term and thus assigned a large value to K_1.

The following chapter summarizes the results presented in this chapter. In addition, the next chapter offers several conclusions that can be drawn from these results and discusses various areas for follow-on research.

VI. Conclusions and Recommendations

The first objective of this effort was to explore the use of the Kalman filter and the information it provides in constructing confidence intervals for steady-state parameters of discrete-event simulations. The second objective was to compare these methods to commonly used techniques for confidence interval construction. Both of these objectives have been met and many areas for future research have been identified. This chapter has three major areas of discussion. First, the results from the previous chapter are summarized. Next, conclusions are drawn from these results. Finally, recommendations are made for potential follow-on research topics.

Summary of the Results

Before discussing the final conclusions of this research effort, a summary of the results is presented. The best cases from the MMAE investigation in the previous chapter were run for 1000 replications. This reduces the estimation accuracy and allows for better comparisons between the MMAE techniques and the baseline Monte Carlo analysis. The actual coverage rates for the baseline techniques using $M/M/1$ Case 2 data are found in Table 81. The average half widths and standard deviations are found in Table 83 and Table 85 respectively.

Table 82 presents the actual coverage rates for three Kalman filter techniques. SKF is the technique that estimates the variance from a single Kalman filter. After obtaining the variance, a confidence interval is constructed, using Equation (2), the estimated variance and the average value of all the observations.

MMAE used 13 evenly-spaced filters with a total spread of $\pm$ 1.6 times $\hat{\sigma}_y^2$. The filters were centered on the average value of all the observations. A confidence interval is obtained, using Equation (2), from the MMAE estimate of the mean, Equation (16), and its associated variance, Equation (17).

MMAE PI uses the probabilistic approach of finding the interval that has 90 percent of the final MMAE filter probabilities. MMAE PI used 24 evenly-spaced filters centered on the average value of the observations with a total spread of $\pm$ 1.6 times $\hat{\sigma}_y^2$. The average half widths for these three techniques are found in Table 84 and their associated standard deviations are found in Table 86.

Table 81. Actual Coverage Rates (Summary of Accepted Techniques)

Data	NOBM			OBM			STDS			AUTO
Size	5	10	20	5	10	20	5	10	20	
1280	0.815	0.784	0.730	0.801	0.781	0.735	0.747	0.672	0.531	0.792
2560	0.857	0.843	0.814	0.852	0.826	0.802	0.820	0.765	0.674	0.830
5120	0.881	0.868	0.844	0.878	0.869	0.850	0.843	0.827	0.762	0.845

Note: With nominal rate of 0.9, estimation accuracy is $\approx \pm 0.016$ for 1000 runs.

Table 82. Actual Coverage Rates (Summary of Kalman Filter Techniques)

Data Size	SKF	MMAE	MMAE PI
1280	0.777	0.732	0.757
2560	0.824	0.800	0.842
5120	0.836	0.855	0.898

Note: With nominal rate of 0.9, estimation accuracy is $\approx \pm 0.016$ for 1000 runs.

Tables 81 and 82 indicate that, for a large sample size, the MMAE PI technique provides the highest actual coverage rate of any of the baseline or Kalman filter techniques. At a sample size of 2560, the MMAE PI technique offers coverage rates similar to those provided by the better baseline techniques (e.g. NOBM and OBM). At the smallest sample size, 1260, the coverage rates provided by MMAE PI were significantly below those provided by NOBM and OBM with large batches. Of the three Kalman filter techniques, the SKF technique provided the best coverage rate for the small sample size. However, as sample size increases the SKF technique did not provide coverage rates that were competitive with the accepted techniques. The MMAE technique provided coverage rates that were lower than MMAE PI for all sample sizes. However, the coverage rates were competitive when compared to the accepted techniques.

Table 83. Average Half Widths (Summary of Accepted Techniques)

Data	NOBM			OBM			STDS			AUTO
Size	5	10	20	5	10	20	5	10	20	
1280	1.520	1.219	1.009	1.378	1.177	0.992	1.187	0.888	0.595	1.455
2560	1.201	1.018	0.884	1.096	0.971	0.866	1.056	0.823	0.621	0.990
5120	0.898	0.776	0.713	0.830	0.755	0.700	0.806	0.710	0.576	0.700

Table 84. Average Half Widths (Summary of Kalman Filter Techniques)

Data Size	SKF	MMAE	MMAE PI
1280	1.239	1.285	1.301
2560	0.930	1.000	1.061
5120	0.671	0.722	0.822

Table 85. Standard Deviation of Half Widths (Summary of Accepted Techniques)

Data	NOBM			OBM			STDS			AUTO
Size	5	10	20	5	10	20	5	10	20	
1280	1.165	0.755	0.508	0.959	0.701	0.492	0.730	0.374	0.167	2.246
2560	0.807	0.592	0.429	0.648	0.526	0.405	0.677	0.398	0.193	0.804
5120	0.525	0.391	0.314	0.435	0.357	0.297	0.471	0.342	0.206	0.365

Tables 83 and 84 indicate that the Kalman filter techniques offer small average half widths for the levels of coverage they provide. In particular, with a sample size of 5120, MMAE PI had an average half width of 0.822. This half width is smaller than the average half width (0.898) provided by the baseline technique with the highest coverage (NOBM with 5 batches). Tables 85 and 86 indicate that, for the levels of coverage provided by the Kalman filter techniques, not only are the average half widths smaller, as seen in Table 84, but their standard deviations are also smaller.

Conclusions

The results summarized in the previous section demonstrate that the techniques based on the Kalman filter offer novel and efficient ways to construct meaningful confidence intervals for steady-state parameters of discrete-event simulations.

Table 86. Standard Deviation of Half Widths (Summary of Kalman Filter Techniques)

Data Size	SKF	MMAE	MMAE PI
1280	1.100	0.933	0.877
2560	0.622	0.656	0.607
5120	0.318	0.346	0.333

The probabilistic approach (MMAE PI) appears to provide the best results for large sample sizes. Its performance rapidly degrades as sample size goes down. This might be due to the presence of some remaining transient data in the system. On the other hand, a $M/M/1$ model is extremely complex and the MMAE bank of filters might simply require a large amount of data to stabilize.

The MMAE approach that uses evenly-spaced filters to obtain an estimate of the mean, Equation (16), and its associated variance, Equation (17), and then uses a t-critical value and Equation (2) to construct a confidence interval, also provided promising results. However, the approach that used normally-spaced filters provided low actual coverage rates. Only one case was run using normally-spaced filters and the total spread value may not have been high enough. This would have caused an underestimation of the means variance and the intervals half width, resulting in low actual coverage rates.

The SKF technique provided reasonable coverages when considering its computational ease (the data is only processed through one estimated filter). However, the variance estimate used in this method is based on a modified version of an approximation, Equation (32), used by Fishman [10] in his autoregressive method, and this autoregressive method provided higher actual coverage rates for all sample sizes. This observation may be indicative of an underlying problem in the estimation routines used to obtain the "best" Kalman filter. The results in Chapter IV indicated that the methodology used to estimate the Kalman filter parameters was consistently underestimating the measurement noise in the AR(2) cases where measurement noise was induced. It was hoped that the measurement noise would act as a "lack of fit" term when the $M/M/1$ queue was represented by an AR(2) model. If, in the $M/M/1$ case, the value of R were higher, the value of K_1 would have been lower and perhaps better estimates of the variance would have been obtained.

The MMAE and SKF approach both made use of a t-critical value. The use of a t-critical value is appropriate only when the underlying distribution of the sample is normally distributed. It is not clear if the sampling distribution used in the MMAE technique is normal. The use of the incorrect multiplier may have caused erroneous results.

In summary, for the trials run in this effort, the MMAE PI technique is clearly the best. It provides actual coverage rates near or equal to nominal, while keeping the average half width and the standard deviation of the half widths extremely low. None of the baseline cases or other

Kalman filter techniques performed as well as MMAE PI for large sample sizes.

Recommendations

In addition to providing excellent results, this research has revealed several areas that offer a great potential for future research efforts. The following paragraphs discuss some of these areas.

The Kalman filter technique based on the use of one estimated filter provides, in addition to a reasonable confidence interval, an estimate of the system's underlying variance. None of the baseline techniques offer this information. This variance information could be exploited in future research endeavors.

The results obtained by the MMAE and MMAE PI method were excellent. However, these results may not be the best results obtainable with the Kalman filter methods. As discussed in Chapter III and seen in the results from Chapter IV, the number of filters used, the spacing between the filters, and the total spread of the filters are three critical parameters affecting MMAE performance. The results presented in this effort were based on various combinations of these parameters that were selected based on the hueristics discussed. A logical follow-on would be to perform some type of statistical analysis to determine how these issues, and their interactions, affect the performance measures of the confidence intervals. In particular, various response surface methodology techniques (e.g., experimental design, and factor analysis) could be utilized for this purpose.

It would also make sense to plot the final MMAE filter probabilities averaged over the entire set of runs. These plots might give a better indication of why some arrangements performed better than others, and whether or not the use of a t-critical value is appropriate.

An $M/M/1$ queue is characterized by an interarrival rate and a service rate. Both of these generate exponentially distributed interarrival and service times, whereas the AR(2) model is based on normally distributed shocks. This apparent conflict may have been the underlying cause of the underestimation of R and hence the overestimation of K_1. Perhaps a different dynamics model could be found to alleviate this problem.

As discussed in the conclusions section above, the MMAE methods work best with large sample sizes. It would be interesting to increase the sample size even further and see if the results

continue to improve.

The MMAE filter's final probabilities may take longer to stabilize if they are subjected to transient data. The length of the transient in $M/M/1$ queues can not be analytically determined. One possible explanation for the MMAE's better performance with larger sample sizes is that not enough data was truncated. It would be interesting to use both untruncated data and data with a greater amount of truncation than used in this research (5000 observations). The results of this analysis could indicate if the MMAE filter probabilities simply require a large sample size, or if there was transient data still remaining in the data used in this effort and the large sample size helped to "wash out" the remaining transient.

The set of simulation observations used to estimate the various Kalman filters was the same set of observations used to generate the residuals. This might have induced a bias problem. It would be interesting to run the simulation for an even longer sample size, then split the data into two sets, one for estimating the Kalman filter parameters, and one for actually processing through the Kalman filter to generate the residuals (and thus the variance estimate).

A final recommendation would be to run paired t tests of the difference between the various actual coverage rates and average half widths. These tests would indicate if the differences are statistically significant. Choobineh and Ballard [6] also recommend looking at paired t tests on the actual coverage rates per unit length of the half width. This t test might give a better impression of whether one method is statistically different from another, although it does not take into account the standard deviation of the half widths.

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SUBROUTINE KFMAIN(DS)

      INCLUDE '[RHOWARD.LATVER]SLAMCOM.FOR'
      INCLUDE '[RHOWARD.LATVER]COMVAR.FOR'

C
C  THIS SUBROUTINE COMPUTES CONFIDENCE INTERVALS USING BOTH
C  FISHMAN'S VARIANCE APPROXIMATION (SKF) METHOD AND THE MMAE
C  METHOD USING EVENLY SPACED FILTERS.
C
C        LOCAL VARIABLES & DEFINITIONS
C
C  FVAR - ESTIMATE OF VARIANCE BASED ON FISHMANS APPROXIMATION
C  FSDEV - CORRESPONDIND STANDARD DEVIATION
C  FDOF - CORRESPONDING DEGREES OF FREEDOM
C  MVAR - ESTIMATE OF VARIANCE BASED ON MMAE
C  MSDEV - CORRESPONDING STANDARD DEVIATION
C  RESID - SUM OF SQUARED RESIDUALS FROM THE KALMAN FILTER
C  A - K.F. CALCULATED COVARIANCE OF RESIDUAL SEQUENCE
C  SPREAD - THE AMOUNT SUBTRACTED AND ADDED TO THE AVERAGE VALUE OF X
C           TO DETERMINE THE MINIMUM AND MAXIMUM VALUES FOR THE MEAN
C           IN THE MMAE BANK OF KALMAN FILTERS
C  MEAN(NFILT) - PROPOSED MEAN VALUES FOR EACH OF THE FILTERS
C  INCREM - THE INCREMENT (SPACING) BETWEEN ADJACENT FILTERS
C  P(NFILT) - THE CONDITIONAL PROBABILITIES OF EACH FILTER BEING THE
C             FILTER WITH THE CORRECT MEAN VALUE
C  RN(NFILT) - THE RESIDUAL VALUE FOR EACH FILTER
C  CDF(NFILT) -  THE PROBABILITY OF SEEING A GIVEN RESIDUAL IN A FILTER
C  N1PLUS - STATE ESTIMATE AFTER PROPAGATION BUT BEFORE MEASUREMENT
C           UPDATE
C  N2PLUS - STATE ESTIMATE AFTER PROPAGATION BUT BEFORE MEASUREMENT
C           UPDATE
C  N1MINUS - STATE ESTIMATE AFTER PROPAGATION AND MEASUREMENT UPDATE
C  N2MINUS - STATE ESTIMATE AFTER PROPAGATION AND MEASUREMENT UPDATE
C  NUM(NFILT) - CDF(NFILT*P(NFILT)
C  DENOM - SUMMATION OF ALL NUM(NFILT)
C  I,W  - USED AS COUNTER INDICES IN DO LOOPS
C

      DOUBLE PRECISION RESID,A,FSDEV,FDOF,FVAR,MVAR,MSDEV
      DOUBLE PRECISION QD,R,SPREAD
      DOUBLE PRECISION INCREM,TA,P(NFILT)
      DOUBLE PRECISION MEAN(NFILT),N1MINUS(NFILT),N2MINUS(NFILT),
     *N1PLUS(NFILT),N2PLUS(NFILT)
      DOUBLE PRECISION ZN(NFILT),RN(NFILT) CDF(NFILT),NUM(NFILT),DENOM
      INTEGER I,W,DS
```

```
C
C  FIRST CALCULATE VALUES OF RHO1 AND RHO2 FOR THE DATA
C
      CALL ESTPAR
C
C  FIND THE VALUES OF K1,K2,PHI1, AND PHI2 THAT MINIMIZE THE RESIDUAL
C  SUM OF SQAURES FOR THE KALMAN FILTER
C
      CALL SEARCH
C
C  MAKE SURE THE CORRECT VALUES ARE STORED FOR PHI1 AND PHI2 AND K2
C  AND RESID  (THE CURRENT VALUES MAY BE ONES FROM A NONOPTIMAL
C  POINT IN THE SEARCH ROUTINE, WE ARE ONLY GUARANTEED THAT K1 IS
C  THE CORRECT OPTIMAL VALUE)
C
      CALL ESTPHI(K1)
      CALL KFRESID(MX,K1,K2,RESID)
C
C  USE FISHMANS APPROXIMATION TO ESTIMATE THE VARIANCE AND DEGREES
C  OF FREEDOM BASED ON THIS OPTIMAL KALMAN FILTER
C
      CALL GETPQR(RESID,QD,R,A)
C
C  KEEP TRACK OF STATISTICS FOR OUTPUT REPORT
C
      SA(DS) = SA(DS) + A
      SQD(DS) = SQD(DS) + QD
      SR(DS) = SR(DS) + R
      SK1(DS) = SK1(DS) + K1
      SK2(DS) = SK2(DS) + K2
      SPHI1(DS) = SPHI1(DS) + PHI1
      SPHI2(DS) = SPHI2(DS) + PHI2
      SA2(DS) = SA2(DS) + A*A
      SQD2(DS) = SQD2(DS) + QD*QD
      SR2(DS) = SR2(DS) + R*R
      SK12(DS) = SK12(DS) + K1*K1
      SK22(DS) = SK22(DS) + K2*K2
      SPHI12(DS) = SPHI12(DS) + PHI1*PHI1
      SPHI22(DS) = SPHI22(DS) + PHI2*PHI2
```

```
      FVAR = QD/(NDAT*(1.-PHI1-PHI2)**2) + R/NDAT
      FDOF = (NDAT*(1.-PHI1-PHI2))/(4.+4.*PHI2) - 1.
      FSDEV = DSQRT(FVAR)
      SPREAD = SPVAL * FSDEV

C
C  USE MMAE TO OBTAIN AN ESTIMATE OF THE MEAN VALUE AND ITS
C  ASSOCIATED VARIANCE
C

C
C  FIRST, INITIALIZE SOME PARAMETERS TO ZERO AND SET THE MEAN VALUE
C  TO BE USED IN EACH OF THE FILTERS.
C

      DENOM = 0.0
28    MEAN(1) = MX - SPREAD
      IF (MEAN(1).LT.0.0) THEN
          SPREAD = SPREAD - 0.01
          GOTO 28
      ENDIF
      INCREM = SPREAD/((DBLE(NFILT)-1.0)/2.0)
      DO 10 I=2,NFILT
         MEAN(I) = MEAN(1) + (INCREM*(I-1))
10    CONTINUE

C
C  SET THE A PRIORI PROBABILITY INFORMATION FOR EACH OF THE FILTERS.
C  THE PROGRAM IS SET UP TO GIVE THE INITIAL PROBABILITIES A UNIFORM
C  DISTRIBUTION OR A TRIANGULAR DISTRIBUTION.  IF YOU WANT A
C  TRIANGULAR DISTRIBUTION PUT A 'C' (COMMENT MARK) IN THE FIRST
C  POSITION ON THE LINE THAT SAYS GOTO 121
C

      DO 120 I = 1,NFILT
         P(I) = 1./DBLE(NFILT)
120   CONTINUE

C     GOTO 121

      P(1) = 1./121.
      P(2) = 2./121.
      P(3) = 3./121.
      P(4) = 4./121.
      P(5) = 5./121.
      P(6) = 6./121.
      P(7) = 7./121.
      P(8) = 8./121.
      P(9) = 9./121.
      P(10) = 10./121.
      P(11) = 11./121.
```

```
      P(12) = 10./121.
      P(13) = 9./121.
      P(14) = 8./121.
      P(15) = 7./121.
      P(16) = 6./121.
      P(17) = 5./121.
      P(18) = 4./121.
      P(19) = 3./121.
      P(20) = 2./121.
      P(21) = 1./121.

121   CONTINUE


C
C  PROPAGATE AND UPDATE EACH OF THE FILTERS THROUGH THE ENTIRE SET OF
C  DATA ON EACH PASS UPDATE THE CONDITIONAL PROBABILITY OF EACH FILTER
C  BEING THE CORRECT ONE
C

      N1PLUS(1) = 0
      N2PLUS(1) = 0
      DO 20 W = 1,NDAT
         DO 30 I = 1,NFILT
            N1MINUS(I) = PHI1 * N1PLUS(I) + PHI2 * N2PLUS(I)
            N2MINUS(I) = N1PLUS(I)
            ZN(I) = X(W) - MEAN(I)
            RN(I) = ZN(I) - N1MINUS(I)
            N1PLUS(I) = N1MINUS(I) + K1*RN(I)
            N2PLUS(I) = N2MINUS(I) + K2*RN(I)
            CDF(I) = (1./SQRT(2.*3.1415926*A))*EXP(-0.5*
     *                  (RN(I)**2)*(1./A))
            IF (CDF(I).LT.1D-10) CDF(I) = 1D-10
            NUM(I) = CDF(I)*P(I)
            DENOM = DENOM + NUM(I)
30       CONTINUE
         DO 40 I = 1,NFILT
            P(I) = NUM(I)/DENOM
            IF (P(I).LT.1D-20) P(I) = 1D-20
40       CONTINUE
         DENOM = 0.0
20    CONTINUE

C
C   OBTAIN THE MMAE ESTIMATE FOR THE MEAN VALUE AND IT'S VARIANCE
C

      MMX = 0.
      MVAR = 0.
      DO 50 I = 1,NFILT
         MMX = MMX + MEAN(I)*P(I)
```

```
50    CONTINUE
      DO 60 I = 1,NFILT
         MVAR = MVAR + ((MEAN(I)-MMX)*(MEAN(I)-MMX)*P(I))
60    CONTINUE
      MSDEV = DSQRT(MVAR)


C
C  THESE COUNTER LINES KEEP TRACK OF THE TWO ESTIMATES OF THE MEAN
C  (MX AND MMX) SQUARED DEVIATIONS FROM THE TRUE VALUE.
C

         SMXD(DS) = SMXD(DS)+(MX - TRUVAL)
         SMMXD(DS) = SMMXD(DS)+(MMX - TRUVAL)
         SMXD2(DS) = SMXD2(DS)+(MX-TRUVAL)*(MX-TRUVAL)
         SMMXD2(DS) = SMMXD2(DS)+(MMX-TRUVAL)*(MMX-TRUVAL)

C
C  CALL SUBROUTINE CI TO CALCULATE CONFIDENCE INTERVALS
C
C

      CALL CI(FSDEV,5,DS,1)
      CALL CI(MSDEV,6,DS,1)

      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE CI(STDEV,J,DS,BS)

      INCLUDE '[RHOWARD.LATVER]COMVAR.FOR'
      INCLUDE '[RHOWARD.LATVER]SLAMCOM.FOR'

C  THIS SUBROUTINE CALCULATES A CONFIDENCE INTERVAL AND KEEPS TRACK
C  OF THE NECESSARY STATISTICS FOR THE MONTE CARLO ANALYSIS
C
C        LOCAL VARIABLES & DEFINITIONS
C
C  STDEV - THE STANDARD DEVIATION
C  CONF -  (1 - ALPHA/2) E.G. FOR A 90% CONFIDENCE INTERVAL CONF = 0.95
C  J - THE METHOD NUMBER
C       5 = K.F. USING FISHMANS APPROXIMATION FOR VARIANCE
C       6 = K.F. USING MMAE VARIANCE
C       4 = FISHMAN'S AUTOREGRESSIVE TECHNIQUE
C       1 = NON-OVERLAPPING BATCH MEANS
C       2 = OVERLAPPING BATCH MEANS
C       3 = STANDARDIZED TIME SERIES
C  HW - THE INTERVALS HALF-WIDTH
C  LB - THE INTERVALS LOWER BOUND
```

```
C  UB - THE INTERVALS UPPER BOUND
C  BS = BATCH SIZE DESIGNATOR (SEE SLAMCOM FOR EXPLANATION)
C  DS = DATA SIZE DESIGNATOR (SEE SLAMCOM FOR EXPLANATION)

      INTEGER J,DS,BS
      DOUBLE PRECISION STDEV,HW,LB,UB,MEANVAL

C
C  SET MEANVAL = MX OR MMX DEPENDING ON WHICH ESTIMATE OF THE MEAN YOU WANT
C

      IF (J.EQ.6) THEN
         MEANVAL = MMX
      ELSE
         MEANVAL = MX
      ENDIF

      HW = TCRIT(J,BS)*STDEV
      LB = MEANVAL - HW
      UB = MEANVAL + HW
      IF (LB.LE.truval.AND.UB.GE.truval COVER(J,DS,BS)=COVER(J,DS,BS)+1.
      SUMW(J,DS,BS) = SUMW(J,DS,BS) + HW
      SUMW2(J,DS,BS) = SUMW2(J,DS,BS) + HW*HW

      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE ESTPAR

      INCLUDE '[RHOWARD.LATVER]COMVAR.FOR'
      INCLUDE '[RHOWARD.LATVER]SLAMCOM.FOR'

C
C  THIS SUBROUTINE ESTIMATES THE NECESSARY PARAMETERS (PHI1 AND PHI2)
C  FOR PROCESSING THE DATA THROUGH A KALMAN FILTER.
C
C        LOCAL VARIABLES & DEFINITIONS
C
C  NUM1 - FIRST ORDER LAG COVARIANCE
C  NUM2 - SECOND ORDER LAG COVARIANCE
C  SUMSQR - SUM OF SQUARED DEVIATIONS OF OBSERVATIONS FROM MX
C  L - USED AS COUNTER INDEX IN DO LOOP
C

      DOUBLE PRECISION NUM1,NUM2,SUMSQR
      INTEGER L

      NUM1 = 0.
      NUM2 = 0.
```

```
      SUMSQR = (X(1)-MX)**2
      DO 10 L = 2,NDAT
         NUM1 = NUM1 + (X(L)-MX)*(X(L-1)-MX)
         SUMSQR = SUMSQR + (X(L)-MX)**2
         IF (L.EQ.2) GOTO 10
         NUM2 = NUM2 + (X(L)-MX)*(X(L-2)-MX)
10    CONTINUE

      RHO1 = NUM1/SUMSQR
      RHO2 = NUM2/SUMSQR

      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE ESTPHI(V)

      INCLUDE '[RHOWARD.LATVER]COMVAR.FOR'
      INCLUDE '[RHOWARD.LATVER]SLAMCOM.FOR'

C
C  THIS SUBROUTINE CALCULATES THE VALUES OF PHI1 AND PHI2 FOR THE
C  KALMAN FILTER GIVEN A SPECIFIC VALUE OF K1 AS ITS ARGUMENT (V)
C
C        LOCAL VARIABLES AND DEFINITIONS
C
C  V - GIVEN VALUE OF K1 IN THE KALMAN FILTER
C

      DOUBLE PRECISION V
      PHI2 = (RHO2*V - RHO1**2)/(V**2 - RHO1**2)
      PHI1 = (RHO1 - RHO1*PHI2)/V

C
C  IF THE CALCULATED VALUES OF PHI1 AND PHI2 WOULD MAKE THIS AN
C  UNSTABLE AR(2) SET THE VALUE OF PHI1 TO 999.
C

      IF (ABS(PHI2).GE.1.OR.PHI1+PHI2.GE.1.OR.PHI2-PHI1.GE.
     *1) PHI1 = 999.

      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE SEARCH

      INCLUDE '[RHOWARD.LATVER]COMVAR.FOR'
      INCLUDE '[RHOWARD.LATVER]SLAMCOM.FOR'
```

```
C
C  THIS SUBROUTINE USES THE METHOD OF GOLDEN SECTIONS TO SEARCH FOR THE
C  VALUE OF K1 THAT MINIMIZES THE RESIDUAL SUM OF SQUARED ERRORS IN A
C  KALMAN FILTER BASED ON AN AR(2) MODEL WITH ESTIMATED VALUES OF
C  PHI1 AND PHI2
C
C        LOCAL VARIABLES & DEFINITIONS
C
C  TOL - THE TOLERANCE RANGE FOR ENDING THE SEARCH ROUTINE
C  Q - CURRENT LOWER BOUNDARY FOR SEARCH
C  R - CURRENT UPPER BOUNDARY FOR SEARCH
C  DELTA - GOLDEN RATION FOR DETERMINING POINTS
C  PT1 - GOLDEN SECTION SEARCH POINT NEARER Q
C  PT2 - GOLDEN SECTION SEARCH POINT NEARER R
C  COUNT - INDEX TO INSURE SEARCH DOESN'T ENTER AN ENDLESS LOOP
C  V1,V2,V3,V4 - SUM OF SQUARED RESIDUALS USING POINTS Q,PT1,PT2, AND
C                R RESPECTIVELY
C

      DOUBLE PRECISION TOL,CHECK,DELTA,QP,RP,V1,V2,V3,V4,PT1,PT2
      INTEGER COUNT
      PARAMETER (DELTA=0.618)
      PARAMETER (TOL = 0.002)
      EXTERNAL CHECK
      DOUBLE PRECISION ETLP

C
C  INITIALIZE SEARCH SPACE FOR POSSIBLE VALUES OF K
C

      QP=0.00001
      RP=1.0
      COUNT = 0

C
C  DETERMINE FIRST TWO POINTS
C

      PT1 = RP - DELTA*(RP-QP)
      PT2 = QP + DELTA*(RP-QP)

C
C  INITIALIZE VALUES FOR FIRST FOUR POINTS
C

      V1 = CHECK(QP)
      V2 = CHECK(PT1)
      V3 = CHECK(PT2)
      V4 = CHECK(RP)
```

```
C
C  CHECK POINTS & ELIMINATE SECTION & SHUFFLE POINTS/VALUES
C

10     COUNT = COUNT + 1
       ETLP = RP-QP
       IF ((V1.LT.V3.AND.V1.LT.V4).OR.(V2.LT.V3.AND.V2.LT.V4)) THEN
         RP = PT2
         PT2 = PT1
         PT1 = RP - DELTA * (RP-QP)
         V4 = V3
         V3 = V2
         V2 = CHECK(PT1)
       ELSE IF ((V3.LT.V1.AND.V3.LT.V2).OR.(V4.LT.V1.AND.V4.LT.V2)) THEN
         QP = PT1
         PT1 = PT2
         PT2 = QP + DELTA*(RP-QP)
         V1 = V2
         V2 = V3
         V3 = CHECK(PT2)
       ENDIF

C
C  SEE IF STOPPING RULE IS MET
C

       IF (ETLP.EQ.RP-QP) GOTO 15
       IF (COUNT.GT.50) GOTO 20
       IF ((RP-QP).GT.TOL) GOTO 10

C
C  STOPPING RULE MET SO ASSIGN VALUE FOR K1
C

15     K1 = (RP+QP)/2.0

20     RETURN
       END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

       DOUBLE PRECISION FUNCTION CHECK(V)

       INCLUDE '[RHOWARD.LATVER]COMVAR.FOR'
       INCLUDE '[RHOWARD.LATVER]SLAMCOM.FOR'

C
C  THIS FUNCTION EVALUATES THE SUM OF SQUARED RESIDUALS FOR A FILTER
C  WITH A K1 VALUE OF V
C
```

```
      DOUBLE PRECISION V

C
C  FIRST FIND THE VALUES OF PHI1 AND PHI2 FOR THIS PARTICULAR VALUE
C  OF K1
C

      CALL ESTPHI(V)

C
C  IF ESTPHI ASSIGNED A VALUE OF 999. TO PHI1 THEN K1 IS NOT AN
C  ALLOWABLE VALUE SO SET THE FUNCTION TO AN EXTREMELY LARGE NUMBER.

      IF (PHI1.EQ.999.) THEN
         CHECK = 99999999999999999999.
         GOTO 10
      ENDIF

C
C  IF ESTPHI FOUND ADMISSIBLE VALUES FOR PHI1 AND PHI2 FIND THE VALUE
C  OF THE SUM OF SQUARTED RESIDUALS
C

      CALL KFRESID(MX,V,K2,RESID)
      CHECK = RESID

10    RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE KFRESID(TMEAN,TK1,TK2,RESID)

      INCLUDE '[RHOWARD.LATVER]COMVAR.FOR'
      INCLUDE '[RHOWARD.LATVER]SLAMCOM.FOR'

C
C  THIS SUBROUTINE USES A KALMAN FILTER TO PROPOGATE AND UPDATE STATE
C  ESTIMATES AND KEEPS TRACK OF THE SUM OF SQUARED RESIDUALS FOR THE FILTER
C
C        LOCAL VARIABLES AND DEFINITIONS
C
C  TMEAN - MEAN VALUE OF THE PROCESS TO BE USED IN FILTER
C  TK1 - VALUE OF K1 TO BE USED IN FILTER
C  TK2 - VALUE OF K2 TO BE USED IN FILTER
C  N1PLUS - STATE ESTIMATE AFTER PROPAGATION BUT BEFORE MEASUREMENT UPDATE
C  N2PLUS - STATE ESTIMATE AFTER PROPAGATION BUT BEFORE MEASUREMENT UPDATE
C  N1MINUS - STATE ESTIMATE AFTER PROPAGATION AND MEASUREMENT UPDATE
C  N2MINUS - STATE ESTIMATE AFTER PROPAGATION AND MEASUREMENT UPDATE
C  RN - RESIDUAL VALUE
C  ZN - MEASUREMENT 9OBSERVATION FROM PROCESS MMINUS ITS MEAN)
```

```
C  RESID - RUNNING SUM OF SQUARED RESIDUALS
C  W - COUNTER INDEX FOR DO LOOP
C  PMEAN - OPTIONAL VARIABLE TO CALCULATE AVERAGE STATE ESTIMATE
C

      DOUBLE PRECISION TK1,TK2,TMEAN,N1PLUS,N2PLUS,PMEAN
      DOUBLE PRECISION N1MINUS,N2MINUS,RN,ZN,RESID
      INTEGER W

C
C  INITIALIZE VALUES AND CALCULATE K2
C

C      pmean = 0.
      N1PLUS = 0.
      N2PLUS = 0.
      RESID = 0.
      TK2 = ((PHI1*TK1) - (PHI1*(TK1**2)))/(1-PHI2 + (PHI2*TK1))

C
C  RUN DATA THROUGH KALMAN FILTER
C

      DO 10 W = 1,NDAT,1
         N1MINUS = PHI1 * N1PLUS + PHI2 * N2PLUS
         N2MINUS = N1PLUS
         ZN = X(W) - TMEAN
         RN = ZN - N1MINUS
         N1PLUS = N1MINUS + TK1*RN
         N2PLUS = N2MINUS + TK2*RN
         RESID = RESID + RN*RN
10    CONTINUE

C
C  IF USING PMEAN CALCULATE ITS VALUE
C

      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE GETPQR(RESID,QD,R,A)

      INCLUDE '[RHOWARD.LATVER]COMVAR.FOR'
      INCLUDE '[RHOWARD.LATVER]SLAMCOM.FOR'

C
C  THIS SUBROUTINE BACKS OUT THE VALUES OF QD AND R GIVEN K1,K2, AND RESID
C  IT ALSO CALCULATES THE VALUE OF A FOR USE IN MMAE
C
```

```
C        LOCAL VARIABLES & DEFINITIONS
C
C  RESID - VALUE OF SUM OF SQUARED RESIDUALS FROM THE K.F.
C  QD - CALCULATED VALUE FOR COVARIANCE OF PROCESS
C  R - CALCULATED VALUE FOR COVARIANCE OF MEASUREMENT NOISE
C  A - VALUE OF COVARIANCE OF RESIDUAL SEQUENCE
C  P11M,PI2M,P22M - ELEMENTS OF THE K.F. P(-) COVARIANCE MATRIX
C                   P(-) MEANS BEFORE MEASUREMEMT UPDATE
C  P11P,P12P,P22P - ELEMENTS OF THE K.F. P(+) COVARIANCE MATRIX
C                   P(+) MEANS BEFORE MEASUREMENT UPDATE
C

      DOUBLE PRECISION QD,R,P11M,P12M,P22M,P11P,P12P,P22P,A,RESID

      P11M = (K1*RESID)/NDAT
      R = RESID/NDAT - P11M
      A = P11M + R
      P12M = K2*A
      P11P = P11M - ((P11M**2)/A)
      P22M = P11P
      P12P = P12M - ((P11M*P12M)/A)
      P22P = P22M - ((P12M**2)/A)

      QD = P11M - ((PHI1*((PHI1*P11P)+(PHI2*P12P)))+(PHI2*((PHI1
     **P12P)+(PHI2*P22P))))

      RETURN
      END


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE MAINB(DS)

      INCLUDE '[RHOWARD.LATVER]COMVAR.FOR'
      INCLUDE '[RHOWARD.LATVER]SLAMCOM.FOR'

C
C  THIS SUBROUTINE COMPUTES CONFIDENCE INTERVALS USING THE FOLLOWING
C  THREE METHODS THAT REQUIRE THE SELECTION OF A BATCH SIZE TO RUN
C
C  METHOD #       METHOD NAME
C  --------       ---------------------------
C     1           NON-OVERLAPPING BATCH MEANS
C     2           OVERLAPPING BATCH MEANS
C     3           STANDARDIZED TIME SERIES
C
C        LOCAL VARIABLES & DEFINITIONS
C
C  BATSIZ - BATCH SIZE
C  NBATCH - TOTAL NUMBER OF BATCHES (MUST BE LESS THAN 100)
```

```
C  (!NOTE! THIS RELATION MUST HOLD TRUE :: NDAT = NBATCH * BATSIZ)
C  (IF USER WANTS MORE THAN 100 BATCHES REDIMENSION ARRAYS IN
C  NOBM,OBM, AND STDS)
C

      INTEGER J,DS

      DO 10 J = 1,3
         NBATCH = BAMT(J)
         BATSIZ = DSIZE(DS)/NBATCH

      CALL NOBM(DS,J)
      CALL OBM(DS,J)
      CALL STDS(DS,J)
10    CONTINUE

      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE NOBM(DS,BS)

      INCLUDE '[RHOWARD.LATVER]SLAMCOM.FOR'

C
C   NON-OVERLAPPING BATCH MEANS CODE FOR CONSTRUCTING CONFIDENCE INTERVALS
C
C        LOCAL VARIABLES & DEFINITIONS
C
C  N,M - COUNTER INDICES FOR DO LOOPS
C  XBAR(100) - VECTOR OF BATCH MEANS
C  NVAR - NON-OVERLAPPING BATCH MEANS VARIANCE
C  NSDEV - CORRESPONDING STANDARD DEVIATION
C  SUMX - RUNNING SUM OF OBSERVATIONS IN A BATCH
C  SUMVAR - RUNNING SUM OF SQAURED DEVIATIONS OF BATCH MEANS FROM
C           GRAND MEAN
C

      INTEGER N,M,DS,BS
      DOUBLE PRECISION XBAR(20),NVAR,NSDEV,SUMX,SUMVAR
C
C  INITIALIZE VALUES
C

      SUMVAR = 0.0

C
C   CALCULATE BATCH MEANS
C
```

```
      DO 20 M = 1,NBATCH
         SUMX = 0.0
         DO 30 N = (M-1)*BATSIZ+1,M*BATSIZ
            SUMX=SUMX+X(N)
30       CONTINUE
         XBAR(M) = SUMX/BATSIZ
20    CONTINUE

C
C   CALCULATE ESTIMATE OF VARIANCE AND HALF-WIDTH
C

      DO 40 M = 1,NBATCH
         SUMVAR = SUMVAR + ((XBAR(M) - MX)**2)
40    CONTINUE
      NVAR = SUMVAR/((NBATCH-1.0)*NBATCH)
      NSDEV = DSQRT(NVAR)

C
C   CALCULATE CONFIDENCE INTERVAL
C

      CALL CI(NSDEV,1,DS,BS)

      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE OBM(DS,BS)

      INCLUDE '[RHOWARD.LATVER]SLAMCOM.FOR'

C
C  THIS SUBROUTINE CONSTRUCTS A CONFIDENCE INTERVAL BASED ON THE
C  METHOD OF OVERLAPPING BATCH MEANS
C
C        LOCAL VARIABLES & DEFINITIONS
C
C  XBAR(10000) - VECTOR OF BATCH MEANS
C  N,M - COUNTER INDICES FOR DO LOOPS
C  SUMX - RUNNING SUM OF OBSERVATIONS IN A BATCH
C  SUMVAR - RUNNING SUM OF SQUARED DEVIATIONS OF BATCH MEANS FROM THE
C           GRAND MEAN
C  OVAR - OVERLAPPING BATCH MEANS VARIANCE
C  OSDEV - CORRESPONDING VARIANCE
C  B - DUMMY VARIABLE USED IN COMPUTING ODOF
C

      DOUBLE PRECISION XBAR(5000),NBTCH
      INTEGER N,M,DS,BS
```

```
      DOUBLE PRECISION OVAR,OSDEV,SUMX,SUMVAR

C
C  INITIALIZE VALUES
C

      NBTCH=NDAT-BATSIZ+1.0
      SUMVAR = 0.0

C
C  CALCULATE OVERLAPPING BATCHED MEANS
C

         DO 20 M = 1,NBTCH
            SUMX = 0.0
            DO 30 N = M,BATSIZ+M-1
               SUMX = SUMX + X(N)
30          CONTINUE
            XBAR(M) = SUMX/BATSIZ
20       CONTINUE

C
C  CALCULATE ESTIMATE OF VARIANCE AND DEGREES OF FREEDOM
C

         DO 40 M = 1,NBTCH
            SUMVAR = SUMVAR + ((XBAR(M) - MX)**2)
40       CONTINUE
         OVAR = BATSIZ*SUMVAR/((NBTCH-1)*NBTCH)
         OSDEV = DSQRT(OVAR)

C
C  CALCULATE CONFIDENCE INTERVAL
C

      CALL CI(OSDEV,2,DS,BS)

      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE STDS(DS,BS)

      INCLUDE '[RHOWARD.LATVER]SLAMCOM.FOR'

C
C   STANDARDIZED TIME-SERIES CODE FOR CONSTRUCTING CONFIDENCE INTERVALS
C
C        LOCAL VARIABLES & DEFINITIONS
C
```

```
C  CUMA
C  CUMS
C  SUMX
C  XBAR(100)
C  SVAR
C  SSDEV
C  SDOF
C  XAV(100,100)
C  S1(100,100)
C  A(100)
C

      DOUBLE PRECISION CUMA,CUMS
      INTEGER I,N,M,K,J,DS,BS
      DOUBLE PRECISION XBAR(20),SUMX,SUMVAR
      DOUBLE PRECISION SVAR,SSDEV,SDOF
      DOUBLE PRECISION XAV(20,1024), S1(20,1024),A(20)

C
C  INITIALIZE VALUES
C

      SUMVAR = 0.0
      CUMA = 0.0
      CUMS = 0.0
      SVAR= 0.0

C
C   CALCULATE BATCH MEANS
C

      DO 20 M = 1,NBATCH
         SUMX = 0.0
         DO 30 N = (M-1)*BATSIZ+1,M*BATSIZ
            SUMX=SUMX+X(N)
30       CONTINUE
         XBAR(M) = SUMX/BATSIZ
20    CONTINUE

C
C  CALCULATE THE CUMULATIVE AVERAGE OF THE FIRST J OBS IN THE I'TH BATCH
C  AND CENTER THE TIME SERIES (CALCULATE S VALUES)
C

      DO 40 I = 1,NBATCH
         DO 50 J = 1,BATSIZ
            DO 60 K = 1+((I-1)*BATSIZ),J+((I-1)*BATSIZ)
               CUMS = CUMS+X(K)
60          CONTINUE
            XAV(I,J)=(1./DBLE(J))*CUMS
            CUMS = 0 0
```

```
            S1(I,J)=J*(XBAR(I)-XAV(I,J))
            CUMA = CUMA + S1(I,J)
50        CONTINUE
          A(I) = CUMA
          SVAR = SVAR + A(I)**2
          CUMA=0.0
40      CONTINUE

C
C   CALCULATE ESTIMATE OF STANDARD DEVIATION AND DEGREES OF FREEDOM
C

        SSDEV = SQRT(12.) * DSQRT(SVAR/NDAT/NDAT/(BATSIZ**2-1.))
        SDOF = NBATCH

C
C   CALCULATE CONFIDENCE INTERVAL
C

        CALL CI(SSDEV,3,DS,BS)

        RETURN
        END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

        SUBROUTINE MAINNB(DS)

        INCLUDE '[RHOWARD.LATVER]SLAMCOM.FOR'

C
C  THIS SUBROUTINE CALCULATES CONFIDENCE INTERVALS USING THOSE
C  TECHNIQUES THAT DO NOT REQUIRE THE SELECTION OF A BATCH SIZE.
C
C  METHOD #        METHOD NAME
C  --------        ---------------------------
C     4            AUTOREGRESSIVE
C

C
C  BEGIN CALCULATTING CI'S
C

        INTEGER DS

        CALL AUTOR(DS)

        RETURN
        END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
      SUBROUTINE AUTOR(DS)

C
C  AUTOREGRESSIVE CODE FOR CONSTRUCTING CONFIDENCE INTERVALS
C
C  THIS CODE IS A MODIFIED VERSION OF A CODE WRITTEN BY GEORGE FISHMAN.
C
C  THE ORIGINAL CODE CAN BE FOUND IN:
C                 "Principles of Discrete Event Simulation"
C                  John Wiley & Sons, New York, NY
C                  Copywrite 1978
C
C  THIS CODE HAS BEEN MODIFIED TO CONSTRUCT CONFIDENCE INTERVALS FOR A MONTE
C  CARLO EVALUATION AND TO USE DOUBLE PRECISION VARIABLES
C
C  VARIABLE DEFINITIONS
C

      INCLUDE '[RHOWARD.LATVER]SLAMCOM.FOR'

      INTEGER I,NDX,IORDER,DS
      DOUBLE PRECISION N
      INTEGER NBND/2/,MAXNDX/51/
      INTEGER JR(2,2)
      DOUBLE PRECISION R(52),S(52),VXBAR(52),DFV(52),STAT(2,52)
      DOUBLE PRECISION B(1500),BND(2),ALPHA,ASDEV,ADOF
      BND(1) = .05
      BND(2) = .01
      ALPHA = 0.100

      CALL ARMTHD(BND,NBND,R,S,B,XBAR,VXBAR,DFV,
     1JR,ALPHA,STAT,NDX,MAXNDX)
      IF (NDX.GE.MAXNDX.OR.NDX.LE.0) GO TO 28
      IORDER = NDX - 1
C      WRITE (1,*) IORDER, '  IORDER'
      ASDEV = DSQRT(VXBAR(IORDER+1))
      ADOF = DFV(IORDER+1)
      TCRIT(4,1) = STUDTP(ADOF,CONF)
      CALL CI(ASDEV,4,DS,1)
      GO TO 36

28    PRINT *,'ERROR - - CAN NOT FIT AN AR MODEL!!'

36    RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE ARMTHD(BND,NBND,R,S,B,XBAR,VXBAR,DFV,JR,
     1ALPHA,STAT,NDX,MAXNDX)
```

```
C
C  THIS CODE IS A MODIFIED VERSION OF A CODE WRITTEN BY GEORGE FISHMAN.
C
C  THE ORIGINAL CODE CAN BE FOUND IN:
C                 "Principles of Discrete Event Simulation"
C                  John Wiley & Sons, New York, NY
C                  Copywrite 1978
C

      INCLUDE '[RHOWARD.LATVER]SLAMCOM.FOR'

      DOUBLE PRECISION AVG,VAR,EPS,COV,SUMV,SUMWL,SUMS,SUML
      DOUBLE PRECISION R(1),S(1),B(1),VXBAR(1),RHO(1),DFV(1)
      DOUBLE PRECISION STAT(2,1),BND(1)
      INTEGER FIVEK/5000/
      INTEGER NBND,NDX,MAXNDX,JR(2,1)
      DOUBLE PRECISION N
      NDX=-1
      N = NDAT
      IF(N.LE.2) RETURN
      AVG=MX
      VAR=0.0D+00
      DO 2 I=1,N
         VAR=VAR+(X(I)-AVG)**2
2     CONTINUE
      VAR=VAR/N
      XBAR=AVG
      R(1)=VAR
      IF (MAXNDX.LE.3) RETURN
      NM1 = INT(N)-1
      IF (N.GE.MAXNDX) NM1=MAXNDX-1
      DO 4 LAG=1,NM1
         COV=0.0D+00
         NMLAG=INT(N)-LAG
         DO 3 J=1,NMLAG
            COV=COV+(X(J)-AVG)*(X(J+LAG)-AVG)
  3      CONTINUE
         COV=COV/NMLAG
         LAGP1=LAG+1
         R(LAGP1)=COV
4     CONTINUE
      R(LAGP1+1)=0.0
      B(1)=1.0
      S(1)=R(1)
      B(2)=1.0
      SUMS=-R(2)/R(1)
      B(3)=SUMS
      SUMS=SUMS*R(2)
      SUMV=R(1)+SUMS
      SUMWL=R(3)+SUMS
```

```
      S(2)=SUMV
      INDX=3
      NM1P1=NM1+1
      DO 15 I=3,NM1P1
         IM1=I-1
         INDX=INDX+1
         B(INDX)=1.0
         SUMS=-SUMWL/SUMV
         SUMV=VAR
         SUMWL=R(I+1)
         DO 18 J=2,IM1
            INDX=INDX+1
            COV=B(INDX-IM1)+SUMS*B(INDX-2*(J-1))
            SUMV=SUMV+COV*R(J)
            SUMWL=SUMWL+COV*R(I-J+2)
            B(INDX)=COV
18       CONTINUE
         INDX=INDX+1
         B(INDX)=SUMS
         SUMV=SUMV+SUMS*R(I)
         SUMWL=SUMWL+SUMS*R(2)
         S(I)=SUMV
15    CONTINUE
      SMAX=SUMV
      DF=NM1
      NDX=NM1+1
      STAT(1,1)=N*(1.0-SMAX/S(1))
      STAT(2,1)=CHISQP(DF,1.0-ALPHA)
      IF(STAT(1,1).LE.STAT(2,1))NDX=1
      INDX=1
      SUMS=B(INDX)
      VXBAR(1)=S(1)/(SUMS*SUMS*N)
      DFV(1)=N-1
      DO 21 JRK=2,NM1
         DF=NM1+1-JRK
         STAT(1,JRK)=N*(1.0-SMAX/S(JRK))
         STAT(2,JRK)=CHISQP(DF,1.0-ALPHA)
         IF(STAT(1,JRK).LE.STAT(2,JRK).AND.NDX.GT.JRK) NDX=JRK
         SUMS=0.0D+00
         SUML=0.0D+00
         DO 13 J=1,JRK
            INDX=INDX+1
            SUML=SUML+(JRK-1-2*(J-1))*B(INDX)
            SUMS=SUMS+B(INDX)
13       CONTINUE
         DFV(JRK)=N*SUMS/(2.0D+00*SUML)
         VXBAR(JRK)=S(JRK)/(SUMS*SUMS*N)
21    CONTINUE
      IF (NDX.GT.NM1) NDX=NM1
      S(LAGP1+1)=S(NDX)
      IF (NBND.LT.1) RETURN
```

```
      DO 223 LRM = 1,2
         DO 120 J=1,NDX
            S(J)=(2-LRM)*R(J)/VAR+(1-LRM)
120      CONTINUE
         K=1
         EPS=BND(K)
         JRK=NDX-1
         INDX=(NDX+1)*NDX/2+1-JRK
         DO 22 J=1,FIVEK
            SUMV=0.0D+00
            INDX=INDX+JRK
            DO 23 I=1,JRK
               INDX=INDX-1
               SUMV=SUMV-B(INDX)*S(I)
               S(I)=S(I+1)
23          CONTINUE
            S(JRK)=SUMV
122         IF (DABS(SUMV).GT.EPS) GO TO 22
            JR(LRM,K)=J
            IF (K.GE.NBND) GO TO 223
            K= K+1
            EPS=BND(K)
            GO TO 122
22       CONTINUE
         DO 222 KK=K,NBND
            JR(LRM,KK)=FIVEK+1
222       CONTINUE
223   CONTINUE
      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


      FUNCTION STDZ(P)

C
C  THIS CODE IS A MODIFIED VERSION OF A CODE WRITTEN BY GEORGE FISHMAN.
C
C  THE ORIGINAL CODE CAN BE FOUND IN:
C                "Principles of Discrete Event Simulation"
C                 John Wiley & Sons, New York, NY
C                 Copywrite 1978
C

      DOUBLE PRECISION C(3),D(3),P,Q,T,XP,NUM,DEN
      DATA C/2.515517,.8082853,.010328/,D/1.432788,.189269,.001308/
      Q=P
      IF (P.GT..5) Q=1.0-Q
      T=SQRT(-2.0*DLOG(Q))
      NUM=(C(3)*T+C(2))*T+C(1)
```

```
      DEN=((D(3)*T+D(2))*T+D(1))*T+1.0
      XP=T-NUM/DEN
      IF (P.LT..5) XP=-XP
      STDZ=XP
      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      FUNCTION CHISQP(DF,P)


C
C  THIS CODE IS A MODIFIED VERSION OF A CODE WRITTEN BY GEORGE FISHMAN.
C
C  THE ORIGINAL CODE CAN BE FOUND IN:
C                 "Principles of Discrete Event Simulation"
C                  John Wiley & Sons, New York, NY
C                  Copywrite 1978
C

      DOUBLE PRECISION Y,DF,Z,YSQ,SQDF,SQHALF,H(7)
      DATA OLDZ/-10.0/,OLDDF,OLDP/-1.0,-1.0/
      CHISQP=0.0
      IF(P.EQ.OLDP) GO TO 1
      OLDP=P
      Z=STDZ(P)
1     CONTINUE
      IF (DF.LE.0) RETURN
      CHISQP=CHISQ
      IF (OLDDF.EQ.DF.AND.OLDZ.EQ.Z) RETURN
      CHISQ=0.0
      CHISQP=CHISQ
      SQDF=SQRT(DF)
      OLDDF=SQRT(DF)
      OLDDF=DF
      IF (Z.EQ.OLDZ) GO TO 2
      OLDZ=Z
      Y=Z
      SQHALF=SQRT(.5)
      YSQ=Y*Y
      H(1)=Y/SQHALF
      H(2)=2.0*(YSQ-1.0)/3.0
      H(3)=(YSQ-7.0)*Y*SQHALF/9.0
      H(4)=-((6.0*YSQ+14.0)*YSQ-32.0)/405.0
      H(5)=((9.0*YSQ+256.0)*YSQ-433.0)*Y*SQHALF/4860.
      H(6)=(((12.*YSQ-243.0)*YSQ-923.0)*YSQ+1472.0)/25515.0
      H(7)=-(((3753.0*YSQ+4353.0)*YSQ-289517.C)*YSQ-289717.0)*Y*
     1SQHALF/9185400.0
2     CONTINUE
      CHISQ=H(7)
```

```
      DO 20 IBACK=1,6
      I=7-IBACK
      CHISQ=CHISQ/SQDF+H(I)
20    CONTINUE
      CHISQ=(CHISQ/SQDF+1.0)*DF
      CHISQP=CHISQ
      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      DOUBLE PRECISION FUNCTION STUDTP(DF,P)


C
C  THIS CODE IS A MODIFIED VERSION OF A CODE WRITTEN BY GEORGE FISHMAN.
C
C  THE ORIGINAL CODE CAN BE FOUND IN:
c                 "Principles of Discrete Event Simulation
C                  John Wiley & Sons, New York, NY
C                  Copywrite 1978
C

      DOUBLE PRECISION Z,ZZ,DF,OLDZ,OLDDF,OLDP,H(4)
      DATA OLDP,OLDDF/-1.0,-1.0/,OLDZ/-10.0/
      STUDTP=0.0
      IF(P.EQ.OLDP) GO TO 4
      OLDP=P
      Z=STDZ(P)
      IF (Z.LE.0.0) Z = -Z
      GO TO 4
4     CONTINUE
      IF (DF.LE.0.0) RETURN
      STUDTP=STUDT
      IF(OLDDF.EQ.DF.AND.OLDZ.EQ.Z) RETURN
      OLDDF=DF
      STUDT=0.0
      STUDTP=STUDT
      IF(Z.GE.0) GO TO 3
      P=1.0-P
      Z=-Z
3     CONTINUE
      IF(OLDZ.EQ.Z) GO TO 2
      OLDZ=Z
      ZZ=Z*Z
      H(1)=(ZZ+1.0)*Z/4.0
      H(2)=((5.0*ZZ+16.0)*ZZ+3.0)*Z/96.0
      H(3)=(((3.0*ZZ+19.0)*ZZ+17.0)*ZZ-15.0)*Z/384.0
      H(4)=((((79.0*ZZ+776.0)*ZZ+1482.0)*ZZ-1920.0)*ZZ-945.0)*Z/92160.0
2     CONTINUE
      ZZ=0.0
```

```
      DO 1 I=1,4
      J=5-I
      ZZ=(ZZ+H(J))/DF
1     CONTINUE
      STUDT=ZZ+Z
      IF(P.LT.0.5) STUDT=-STUDT
      STUDTP=STUDT
      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

C                             SLAMCOM
C
C  THIS IS A FILE OF COMMON VARIABLES INCLUDED IN EACH PROGRAM THROUGH
C  AN INCLUDE COMMAND.  THE VARIABLES ARE:
C
C  MX - THE CALCULATED MEAN VALUE OF THE X OBSERVATIONS
C  MMX - THE MMAE ESTIMATED MEAN VALUE OF THE X OBSERVATIONS
C  NRUNS - THE NUMBER OF RUNS FOR THE MONTE CARLO ANALYSIS
C  TRUVAL - THE THEORETICAL MEAN VALUE OF THE OBSERVATIONS
C  NENT - THE NUMBER OF ENTITIES GENERATED BY THE SLAM CODE
C         (INCLUDING THOSE TRUNCATED)
C  X - A VECTOR OF THE OBSERVATIONS
C  CONF - DETERMINES DESIRED CONFIDENCE LEVEL (1 - ALPHA/2) = CONF SO
C         FOR A 90% C.I. CONF = 0.95
C  NBATCH - NUMBER OF BATCHES FOR THOSE METHODS REQUIRING BATCHING
C  NDAT - THE AMOUNT OF DATA IN THIS PARTICULAR RUN
C  BATSIZ - THE SIZE OF THE BATCHES
C  NBATCH -  THE NUMBER OF OBSERVATIONS PER BATCH
C  DSIZE(3) - THREE POSSIBEL VALUES FOR DATA SIZE
C  BAMT(3) - THREE POSSIBLE NUMBER OF BATCHES
C  SPVAL - A NUMBER THAT DETERMINES HOW MANY STANDARD DEVIATIONS (AS
C          ESTIMATED BY FISHMANS APPROX) AWAY FROM THE MIDDLE FILTER
C          YOU PLACE THE OUTSIDE FILTERS IN MMAE
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     FOR THE FOLLOWING ARRAYS I,J, AND K REPRESENT INDICES TO SHOW
C     THE FOLLOWING:
C            I = METHOD NUMBER USED (1 TO 6)
C                WHERE 1 = NON-OVERLAPPING BATCH MEANS
C                      2 = OVERLAPPING BATCH MEANS
C                      3 = STANDARDIZED TIME SERIES
C                      4 = FISHMAN'S AUTOREGRESSIVE METHOD
C                      5 = K.F. USING FISHMANS APPROX FOR VARIANCE
C                      6 = MMAE K.F.
C            J = AMOUNT OF DATA USED (1 TO 3)
C                WHERE 1 = 1280 OBSERVATIONS
C                      2 = 2560 OBSERVATIONS
```

```
C                     3 = 5120 OBSERVATIONS
C            K = BATCH SIZE (1 TO 3)
C                WHERE 1 = J/5
C                      2 = J/10
C                      3 = J/20
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  COVER(I,J,K) - CONTAINS THE NUMBER OF TIMES EACH CONFIDENCE
C                 INTERVAL CONTAINS THE ACTUAL VALUE
C  SUMW(I,J,K) - CONTAINS THE RUNNING SUM OF EACH INTERVALS
C                HALF WIDTH
C  SUMW2(I,J,K) - CONTAINS THE RUNNING SUM OF EACH INTERVALS
C                 HALF WIDTHS SQUARED
C  WIDTH(I,J,K) - AVERAGE HALF-WIDTH
C  STAND(I,J,K) - STANDARD DEVIATION OF HALF-WIDTHS
C  TCRIT(I,J,K) - APPROPRIATE VALUE OF T CRITICAL FOR EACH METHOD
C  MA(J),VA(J),SA(J),SA2(J) - MEAN OF A, VARIANCE OF A, RUNNING SUM
C                             OF A VALUES, RUNNING SUM OF SQUARED A VALUES.
C                             WHERE A = K.F. ESTIMATE OF RESIDUALS SEQUENCE'S
C                             VARIANCE
C  MQD(J),VQD(J),SQD(J),SQD2(J) - SAME AS ABOVE WHERE QD IS THE K.F.
C                                 ESTIMATE OF THE VARIANCE OF THE
C                                 NOISE IN THE PROCESS
C  MR(J),VR(J),SR(J),SR2(J) - SAME AS ABOVE WHERE R IS THE K.F.
C                             ESTIMATE OF THE VARIANCE OF THE MEASUREMENT NOISE
C  MK1(J),VK1(J),SK1(J),SK12(J) - SAME AS ABOVE WHERE K1 IS THE FIRST
C                                 ELEMENT OF THE K.F. GAIN VECTOR K
C  MK2(J),VK2(J),SK2(J),SK22(J) - SAME AS ABOVE WHERE K2 IS THE
C                                 SECOND ELEMENT OF THE K.F. GAIN VECTOR K
C  MPHI1(J),VPHI1(J),SPHI1(J0,SPHI12(J) - SAME AS ABOVE WHERE PHI1 IS
C                                         THE K.F. ESTIMATE OF PHI1
C  MPHI2(J),VPHI2(J),SPHI2(J),SPHI22(J) - SAME AS ABOVE WHERE PHI2 IS
C                                         THE K.F. ESTIMATE OF PHI2
C  MMXD(J),VMXD(J),SMXD(J),SMXD2(J) - SAME AS ABOVE WHERE MXD IS THE
C                                     DEVIATIONS OF MX (THE AVERAGE
C                                     VALUE OF THE X OBSERVATIONS)
C                                     FROM THE TRUE VALUE OF THE MEAN
C  MMMXD(J),VMMXD(J),SMMXD(J),SMMXD2(J) - SAME AS ABOVE WHERE MMXD IS THE
C                                         DEVIATIONS OF MMX (MMAE'S
C                                         ESTIMATE OF THE MEAN OF X)
C                                         FROM THE TRUE VALUE OF THE MEAN
C


C
C  SET PARAMETERS THAT DO NOT CHANGE EVER!!!!
C

      DOUBLE PRECISION NRUNS
      DOUBLE PRECISION CONF
```

```
      INTEGER NFILT
      DOUBLE PRECISION SPVAL
      PARAMETER (SPVAL = 5.)
      PARAMETER (NFILT = 21)
      PARAMETER (CONF = 0.95)
      PARAMETER (NRUNS = 1000.)

      DOUBLE PRECISION DSIZE(3)
      DATA DSIZE / 1280,2560,5120 /
      DOUBLE PRECISION BAMT(3)
      DATA BAMT / 5,10,20 /

C
C  DECLARE COMMON VARIABLES
C

      DOUBLE PRECISION X(5120)
      DOUBLE PRECISION MX
      DOUBLE PRECISION MMX
      DOUBLE PRECISION NDAT
      DOUBLE PRECISION NBATCH
      DOUBLE PRECISION BATSIZ
      DOUBLE PRECISION TCRIT(6,3)
      DOUBLE PRECISION TRUVAL
      INTEGER NENT
      DOUBLE PRECISION COVER(6,3,3),SUMW(6,3,3),SUMW2(6,3,3)
     *,WIDTH(6,3,3),STAND(6,3,3)
     *,SA(3),SA2(3),SQD(3),SQD2(3),SR(3),SR2(3),SK1(3)
     *,SPHI1(3),SPHI12(3),SPHI2(3),SPHI22(3)
     *,SK12(3),SK2(3),SK22(3)
     *,MA(3),VA(3),MR(3),VR(3),MQD(3),VQD(3),MK1(3),VK1(3)
     *,MK2(3),VK2(3),MPHI1(3),VPHI1(3),MPHI2(3),VPHI2(3)
     *,MMXD(3),MMMXD(3),VMXD(3),VMMXD(3),SMXD(3),SMMXD(3)
     *,SMXD2(3),SMMXD2(3)

      COMMON/SLAMCOM/X,MX,MMX,TCRIT,TRUVAL,NENT,COVER,SUMW,SUMW2,
     *SA,SA2,SQD,SQD2,SR,SR2,SK1,SK12,SK2,SK22,SPHI1,SPHI12,SPHI2,
     *SPHI22,MA,VA,MQD,VQD,MR,VR,MK1,VK1,MK2,VK2,
     *MPHI1,VPHI1,MPHI2,VPHI2,WIDTH,STAND,NDAT,BATSIZ,NBATCH,MMXD,MMMXD,
     *VMXD,VMMXD,SMXD,SMMXD,SMXD2,SMMXD2

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


C
C                       COMVAR
C
C
C  COMMON BLOCK OF GLOBAL VARIABLES
C  THESE VARIABLES ARE USED IN THE SUBROUTINES DEALING WITH THE KALMAN
C  FILTER TECHNIQUES FOR CONFIDENCE INTERVAL CONSTRUCTION
```

```
C
C                VARIABLES & DEFINITIONS
C
C
C  K1 - THE VALUE OF K1 IN THE KF
C  K2 - THE VALUE OF K2 IN THE KF
C  PHI1 - THE ESTIMATED VALUE OF PHI1
C  PHI2 - THE ESTIMATED VALUE OF PHI2
C  RHO1 - THE CALCULATED VALUE OF RHO1
C  RHO2 - THE CALCULATED VALUE OF RHO2
C

      DOUBLE PRECISION K1,K2,PHI1,PHI2,RHO1,RHO2,RHO3
      COMMON /COMVAR/ K1,K2,PHI1,PHI2,RHO1,RHO2,RHO3
```

Appendix B. *Computer Routines for AR(2) Data Generation*

```
      PROGRAM ARGEN

      INCLUDE '[RHOWARD.LATVER]COMVAR.FOR'
      INCLUDE '[RHOWARD.LATVER]SLAMCOM.FOR'

C
C  THIS PROGRAM GENERATES OBSERVATIONS FROM AN AR(2) PROCESS
C  (I.E. X(I) =  KSI + AR1*X(I-1) + AR2*X(I-2) + EPSILON)
C  THE PROGRAM ALSO ALLOWS FOR THE INTRODUCTION OF MEASUREMENT NOISE
C  INTO THE PROCESS
C
C        LOCAL VARIABLES & DEFINITIONS
C
C  I,P - COUNTER INDICES FOR DO LOOPS
C  AR1 - THE VALUE OF THE FIRST AUTOREGRESSIVE COEFFICIENT
C  AR2 - THE VALUE OF THE SECOND AUTOREGRESSIVE COEFFICIENT
C  VAREPS - THE VARIENCE OF EPSILON
C  KSI - THE VALUE OF KSI (A CONSTANT)
C  VARMN - THE VARIANCE OF MEASUREMENT NOISE
C  MEAN -  THE THEORETICAL MEAN VALUE OF THE PROCESS
C  TEMP1 - A DUMMY VARIABLE FOR X(I-1)
C  TEMP2 - A DUMMY VARIABLE FOR X(I-2)
C  SHOCK 1 - NORMAL (0,1) PSEUDORANDOM VARIATE
C  SHOCK 2 - NORMAL (0,1) PSEUDORANDOM VARIATE
C  WIDTH(NMETH) - AVERAGE HALF-WIDTH FOR EACH METHOD
C  STAND(NMETH) - STANDARD DEVIATION OF THE HALF-WIDTHS FOR EACH METHOD
C  SMX,SMX2,NX,MX,VX,SX -  USED TO KEEP STATS ON X VARIABLES
C  SMU,SMU2,NU,MU,VU,SU -  USED TO KEEP STATS ON UNIFORM VARIATES
C  SMN,SMN2,NN,MN,VN,SN -  USED TO KEEP STATS ON NORMAL VARIATES
C

      INTEGER I,P
      DOUBLE PRECISION B
      DOUBLE PRECISION AR1, AR2, KSI, VAREPS, MEAN, TEMP1, TEMP2
      DOUBLE PRECISION SHOCK1, SHOCK2, VARMN
      PARAMETER (VAREPS= 1.)
      PARAMETER (VARMN = 1.)
      DOUBLE PRECISION SMX,SMX2,SMU,SMU2,SMN,SMN2,NX,NU,NN
      DATA SMX,SMX2,SMU,SMU2,SMN,SMN2,NX,NU,NN/ 9*0.0 /
      DOUBLE PRECISION VX,MU,VU,SU,MN,VN,SN
      DOUBLE PRECISION STUDTP
      EXTERNAL STUDTP

      OPEN(UNIT=1,FILE='[RHOWARD.LATVER]AR22.new',STATUS='NEW')
      OPEN(UNIT=12,FILE='[RHOWARD.LATVER]CA2.TDF',STATUS='NEW')
      OPEN(UNIT=13,FILE='[RHOWARD.LATVER]HA2.TDF',STATUS='NEW')
      OPEN(UNIT=14,FILE='[RHOWARD.LATVER]SA2.TDF',STATUS='NEW')
      OPEN(UNIT=15,FILE='[RHOWARD.LATVER]KA2.TDF',STATUS='NEW')
      OPEN(UNIT=8,FILE='[RHOWARD.LATVER]MA2.TDF',STATUS='NEW')
```

```
      OPEN(UNIT=17,FILE='[RHOWARD.LATVER]VA2.TDF',STATUS='NEW')


C
C  BY OPENING THESE FILES AND WRITING THE APPROPRIATE VALUES TO THEM
C  THROUGHOUT THE PROGRAM ONE CAN RECORD THE X OBSERVATIONS THE
C  NORMAL(0,1) PSEUDORANDOM VARIATES AND THE UNIFORM(0,1) PSEUDORANDOM
C  VARIATES.  THIS IS USEFUL FOR DEBUGGING PURPOSES AND TO INSURE THE
C  PSEUDEORANDOM NUMBER GENERATOR WORKS SUFFICIENTLY WELL
C

C     OPEN (UNIT=12,FILE='UNI.DAT',STATUS='UNKNOWN')
C     OPEN (UNIT=13,FILE='NORM.DAT',STATUS='UNKNOWN')
C     OPEN (UNIT=14,FILE='AR2.DAT',STATUS='UNKNOWN')

C
C  SET CONSTANT VALUES AND DETERMINE THEORETICAL MEAN
C

      AR1 = 0.5
      AR2 = 0.3
      KSI = 10.
      MEAN = KSI/(1. - AR1 - AR2)
      TRUVAL = MEAN


C
C  USE THE MEAN FOR THE FIRST TWO DUMMY VARIABLES
C

      TEMP1 = MEAN
      TEMP2 = MEAN

C
C  INITIALIZE MONTE CARLO ANALYSIS VARIABLES
C

      CALL INITIAL

C
C  INITIALIZE VALUES FOR EACH RUN
C

      DO 777 P = 1,NRUNS
      NX = 0.0
      SMX = 0.0
      SMX2 = 0.0

C
C  PRODUCE OBSERVATIONS FOR A SINGLE RUN
C
```

```
      DO 100 I = 1,5119,2
         CALL NORM(SHOCK1,SHOCK2,SMU,SMU2,NU,SMN,SMN2,NN)
         X(I) = KSI + AR1*TEMP1 + AR2*TEMP2 + SHOCK1*VAREPS
         X(I+1) = KSI + AR1*X(I) + AR2*TEMP1 + SHOCK2*VAREPS

C
C  ADD MEASUREMENT NOISE
C

         CALL NORM(SHOCK1,SHOCK2,SMU,SMU2,NU,SMN,SMN2,NN)
         X(I) = X(I) + SHOCK1*VARMN
         X(I+1) = X(I+1) + SHOCK2*VARMN

C      WRITE (14,*) X(I)
C      WRITE (14,*) X(I+1)

C
C  KEEP TRACK OF RUNNING SUMS TO CALCULATE AVERAGE VALUE AND STANDARD
C  DEVIATION OF OBSERVATIONS
C

         SMX = SMX + X(I)+ X(I+1)
         SMX2 = SMX2 + X(I)*X(I) + X(I+1)*X(I+1)
         NX = NX + 2

C
C  ASSIGN VALUES TO TEMPORARY VARIABLES
C  (!NOTE! SUBTRACT OFF MEASUREMENT NOISE)
C

         TEMP1 = X(I+1) - SHOCK2*VARMN
         TEMP2 = X(I) - SHOCK1*VARMN

100   CONTINUE

C
C  CALCULATE STANDARD DEVIATION AND MEAN VALUE OF X OBSERVATIONS,
C  UNIFORM PSEUDEORANDOM VARIATES AND NORMAL PSEUDO RANDOM VARIATES
C

      CALL STDEV(SMX,SMX2,NX,MX,VX,SX)
      CALL STDEV(SMU,SMU2,NU,MU,VU,SU)
      CALL STDEV(SMN,SMN2,NN,MN,VN,SN)

C      PRINT *,' '
C       PRINT *,'X -    ', 'NUM = ', NX, 'MEAN = ',MX,'VAR = ',VX
C       PRINT *,'U -    ', 'NUM = ', NU, 'MEAN = ',MU,'VAR = ',VU
C       PRINT *,'N -    ', 'NUM = ', NN, 'MEAN = ',MN,'VAR = ',VN
C      PRINT *,' '
```

```
C
C  CALL ROUTINES FOR CONFIDENCE INTERVAL CONSTRUCTION
C

      CALL CONTROL

777   CONTINUE

C
C  PRODUCE OUTPUT REPORT
C

      CALL FIGURE
      CALL OUTPUT(AR1,AR2,VAREPS,VARMN)

      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE NORM(N1,N2,SMU,SMU2,NU,SMN,SMN2,NN)

C
C   SUBROUTINE NORM(N1,N2)
C
C   THIS SUBROUTINE GENERATES TWO NORMAL R.V.'S USING THE BOX-MULLER
C   TECHNIQUE.  IT CALLS THE INTERNAL FORTRAN FUNCTION RAN WITH A
C   SEED VALUE OF ISEED ON THE FIRST CALL AFTER THAT THE FUNCTION MAINTAINS
C   A NEW VALUE FOR SEED.  RAN RETURNS A UNIFORM ZERO,ONE VARIATE.
C
C        LOCAL VARIABLES & DEFINITIONS
C
C  U1,U2 - TWO PSEUDORANDOM UNIFORM(0,1) VARIATES
C  N1,N2 - TWO PSEUDORANDOM NORMAL(0) VARIATES
C  V1,V2,W,Y - FOUR DUMMY VARIABLES USED TO OBTAIN THE NORMAL VARIATES
C  IX - INITIAL SEED VALUE
C  NU,SMU,SMU2 - NUMBER OF UNIFORM VARIATES, SUM OF THE UNIFORM
C                VARIATES, AND SUM OF THE SQUARED UNIFORM VARIATES
C  NN,SMN,SMN2 - NUMBER OF UNIFORM VARIATES, SUM OF THE NORMAL
C                VARIATES, AND SUM OF THE SQUARED NORMAL VARIATES
C

      DOUBLE PRECISION U1,U2
      DOUBLE PRECISION N1,N2
      DOUBLE PRECISION V1,V2,W,Y
      INTEGER IX
      DATA IX / 2637775 /
      DOUBLE PRECISION SMU,SMU2,NU,SMN,SMN2,NN

      INTRINSIC RAN

10    U1 = RAN(IX)
```

```
      U2 = RAN(IX)

C      WRITE (12,*) U1
C      WRITE (12,*) U2

      SMU = SMU + U1 + U2
      SMU2 = SMU2 + U1*U1 + U2*U2
      NU = NU + 2

      V1 = 2*U1-1
      V2 = 2*U2-1
      W = V1*V1 + V2*V2

      IF (W.GT.1) GOTO 10
      Y = DSQRT((-2*DLOG(W))/W)
      N1 = V1*Y
      N2 = V2*Y

C      WRITE (13,*) N1
C      WRITE (13,*) N2

      SMN = SMN + N1 + N2
      SMN2 = SMN2 + N1*N1 + N2*N2
      NN = NN + 2

      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE STDEV(SUMX,SUMX2,N,MEAN,VAR,SDEV)

C
C  THIS SUBROUTINE CALCULATES THE AVERAGE VALUE, VARIANCE, AND
C  STANDARD DEVIATION OF A SET OF NUMBERS GIVEN THE SUM OF THE
C  OBSERVATIONS, THE SUM OF THE SQUARED OBSERVATIONS, AND THE
C  NUMBER OF OBSERVATIONS
C
C        LOCAL VARIABLES & DEFINITIONS
C
C  SUMX - SUM OF THE OBSERVATIONS
C  SUMX2 - SUM OF THE SQUARED OBSERVATIONS
C  N - NUMBER OF OBSERVATIONS
C  MEAN - AVERAGE VALUE OF THE OBSERVATION
C  VAR - VARIANCE OF THE OBSERVATIONS
C  SDEV - STANDARD DEVIATION OF THE OBSERVATIONS
C

      DOUBLE PRECISION SUMX,SUMX2,N,MEAN,VAR,SDEV

      MEAN = SUMX/N
```

```
VAR = SUMX2/N - MEAN*MEAN
SDEV = DSQRT(VAR)

RETURN
END
```

Appendix C. *Computer Routines for M/M/1 Data Generation*

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C  THIS IS THE MAIN PROGRAM USING SLAM II TO MODEL AN M/M/1 QUEUE AND
C  RECORD OBSERVATIONS ON EACH ENTITIES WAITING TIME IN QUEUE.
C

      PROGRAM MAIN
      DIMENSION NSET(10000)
      INCLUDE 'SLAM$DIR:PARAM.INC'

      COMMON/SCOM1/ATRIB(MATRB), DD(MEQT), DDL(MEQT), DTNOW, II, MFA,
     1MSTOP,NCLNR, NCRDR, NPRNT, NNRUN, NNSET, NTAPE, SS(MEQT),
     2SSL(MEQT),TNEXT, TNOW, XX(MMXXV)

      INCLUDE '[RHOWARD.LATVER]SLAMCOM.FOR'

      COMMON QSET(10000)
      EQUIVALENCE (NSET(1),QSET(1))
      EQUIVALENCE (XX(1),BUSY)

C
C  OPEN FILES FOR DATA OUTPUT REPORTS
C

      OPEN(UNIT=12,FILE='CM8W.TDF',STATUS='NEW')
      OPEN(UNIT=13,FILE='HM8W.TDF',STATUS='NEW')
      OPEN(UNIT=14,FILE='SM8W.TDF',STATUS='NEW')
      OPEN(UNIT=15,FILE='KM8W.TDF',STATUS='NEW')
      OPEN(UNIT=8,FILE='MM8W.TDF',STATUS='NEW')
      OPEN(UNIT=17,FILE='VM8W.TDF',STATUS='NEW')

C
C  SET SLAM II PARAMETERS
C

      NNSET=10000
      NCRDR=5
      NPRNT=6
      NTAPE=7
      NPLOT=2

C
C  BEFORE FIRST RUN, INITIALIZE PARAMETERS
C

      CALL INITIAL
```

```
C
C  CALL SLAM II PROGRAM
C

      CALL SLAM
C
C  AFTER SLAM HAS COMPLETED ALL RUNS CALL ROUTINE TO CALCULATE STATISTICS
C

      CALL FIGURE

C
C  AFTER STATISTICS ARE CALCULATED CALL OUTPUT TO PRODUCE OUTPUT FILES
C

      CALL OUTPUT

      STOP
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


      SUBROUTINE EVENT(I)

C
C  SUBROUTINE THAT ENTERS CUSTOMERS INTO SERVICE AND RECORDS WAITING TIME WHEN
C  SERVICE IS COMPLETE
C

      INCLUDE 'SLAM$DIR:PARAM.INC'
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)

      INCLUDE '[RHOWARD.LATVER]SLAMCOM.FOR'

C*****
C*****BEGIN SUBROUTINE
C*****

      DIMENSION A(10)
      GO TO (1,2),I

C*****
C*****SEE IF SERVER IS AVAILABLE AT CUSTOMER ARRIVAL
C*****

    1 IF (NNACT(1).EQ.0) GOTO 10
      CALL FILEM(1,ATRIB)
```

```
      RETURN
   10 ATRIB(2)=0
      CALL ENTER(1,ATRIB)
      RETURN

C*****
C*****AFTER COMPLETION OF SERVICE, SEE IF ANOTHER CUSTOMER IS WAITING
C*****AND RECORD WAITING TIME IN QUEUE.
C*****

    2 NENT = NENT + 1
      IF (NNQ(1).EQ.0) GOTO 20
      CALL RMOVE(1,1,A)
      A(2)=TNOW - A(1)
      CALL ENTER(1,A)
   20 IF (NENT.LT.5000) GOTO 30
      X(NENT-5000) = ATRIB(2)

   30 RETURN
      END

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      SUBROUTINE INTLC

C
C  SLAM CALLS THIS SUBROUTINE BEFORE EVERY RUN TO REINITIALIZE PARAMETERS
C

      INCLUDE 'SLAM$DIR:PARAM.INC'
      COMMON/SCOM1/ATRIB(MATRB), DD(MEQT), DDL(MEQT), DTNOW, II, MFA,
     1MSTOP,NCLNR, NCRDR, NPRNT, NNRUN, NNSET, NTAPE, SS(MEQT),
     2SSL(MEQT),TNEXT, TNOW, XX(MMXXV)

      INCLUDE '[RHOWARD.LATVER]SLAMCOM.FOR'

      NENT=0

      RETURN
      END

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      SUBROUTINE OTPUT

C
C  SLAM CALLS THIS SUBROUTINE AT THE END OF EVERY RUN, THIS ROUTINE CALLS
C  THE ROUTINES THAT CALCULATE CONFIDENCE INTERVALS.
C

            INCLUDE 'SLAM$DIR:PARAM.INC'
```

```
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)

      INCLUDE '[RHOWARD.LATVER]SLAMCOM.FOR'

      CALL CONTROL

      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

C
C  THIS IS THE SLAM II CODE FOR WAITING TIME IN QUEUE WITH A TRAFFIC
C  INTENSITY OF 0.8
C

GEN,HOWARD,THESIS,1/15/92,1000,YES,NO,YES/NO,NO,NO,72;
LIMITS,1,5,100;
INITIALIZE;
NETWORK;
      CREATE,EXPON(1.0),0,1,10120;
CRE   EVENT,1,1;
      TERMINATE;
ARR   ENTER,1,1;
      ACTIVITY(1)/1,EXPON(.8);
DON   EVENT,2,1;
      TERMINATE,10120;
      ENDNETWORK;
FIN;
```

```
      SUBROUTINE OUTPUT

C
C  THIS SUBROUTINE WRITES THE RESULTS TO OUTPUT FILES IN A FORMAT
C  THAT LATEX CAN READ IN TO PRODUCE FINAL TABLES
C

      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)

      INCLUDE '[RHOWARD.LATVER]SLAMCOM.FOR'

      INTEGER J

      DO 10 J = 1,3
      WRITE(12,101) INT(DSIZE(J)),
     *             COVER(1,J,1),COVER(1,J,2),COVER(1,J,3),COVER(2,J,1),
     *             COVER(2,J,2),COVER(2,J,3),COVER(3,J,1),COVER(3,J,2),
     *             COVER(3,J,3),COVER(4,J,1),COVER(5,J,1),COVER(6,J,1)
10    CONTINUE

      DO 20 J = 1,3
      WRITE(13,101) INT(DSIZE(J)),
     *             WIDTH(1,J,1),WIDTH(1,J,2),WIDTH(1,J,3),WIDTH(2,J,1),
     *             WIDTH(2,J,2),WIDTH(2,J,3),WIDTH(3,J,1),WIDTH(3,J,2),
     *             WIDTH(3,J,3),WIDTH(4,J,1),WIDTH(5,J,1),WIDTH(6,J,1)
20    CONTINUE

      DO 30 J = 1,3
      WRITE(14,101) INT(DSIZE(J)),
     *             STAND(1,J,1),STAND(1,J,2),STAND(1,J,3),STAND(2,J,1),
     *             STAND(2,J,2),STAND(2,J,3),STAND(3,J,1),STAND(3,J,2),
     *             STAND(3,J,3),STAND(4,J,1),STAND(5,J,1),STAND(6,J,1)
30    CONTINUE

      DO 40 J = 1,3
      WRITE(15,102) INT(DSIZE(J)),MA(J),MQD(J),MR(J),
     *             MK1(J),MK2(J),MPHI1(J),MPHI2(J)
40    CONTINUE

      DO 50 J = 1,3
      WRITE(15,102) INT(DSIZE(J)),
     *             VA(J),VQD(J),VR(J),VK1(J),VK2(J),VPHI1(J),VPHI2(J)
50    CONTINUE

      DO 60 J = 1,3
      WRITE(8,103) INT(DSIZE(J)),MMXD(J),MMMXD(J)
```

```
60    CONTINUE

      DO 70 J = 1,3
      WRITE(17,103) INT(DSIZE(J)),VMXD(J),VMMXD(J)
70    CONTINUE

101   FORMAT (I6, 12('&', F6.3), ' \\')
102   FORMAT (I6, 7('&', F8.4), ' \\')
103   FORMAT (I6, 2('&', F8.4), ' \\')

      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE INITIAL

C
C  THIS SUBROUTINE INITIALIZES VARIABLES USET TO KEEP TRACK OF THE STATISTICS
C

      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)

      INCLUDE '[RHOWARD.LATVER]SLAMCOM.FOR'

      DOUBLE PRECISION SERRATE,ARRRATE

      WRITE(1,*)  'IN PROGRAM INITIAL'


      DO 10 J= 1,3
         DO 20 I = 1,6
            DO 30 K = 1,3
               COVER(I,J,K) = 0.
               SUMW(I,J,K) = 0.
               SUMW2(I,J,K) = 0.
30          CONTINUE
20       CONTINUE
         SA(J) = 0.0
         SA2(J) = 0.0
         SQD(J) = 0.0
         SQD2(J) = 0.0
         SR(J) = 0.0
         SR2(J) = 0.0
         SK1(J) = 0.0
         SK12(J) = 0.0
         SK2(J) = 0.0
         SK22(J) = 0.0
         SPHI1(J) = 0.0
         SPHI12(J) = 0.0
```

```
         SPHI2(J) = 0.0
         SPHI22(J) = 0.0
         SMXD(J) = 0.0
         SMXD2(J) = 0.0
         SMMXD(J) = 0.0
         SMMXD2(J) = 0.0
10     CONTINUE

C
C  SET TCRIT VALUES FOR METHODS THAT DON'T CHANGE  NOTE!!  THESE CRITICAL
C  VALUES NEED TO BE CHANGED ACCORDING TO ALPHA, NUMBER OF BATCHES, DATA
C  SIZE, ETC.
C

      TCRIT(1,1) = 2.132
      TCRIT(1,2) = 1.833
      TCRIT(1,3) = 1.729
      TCRIT(2,1) = 1.943
      TCRIT(2,2) = 1.766
      TCRIT(2,3) = 1.7
      TCRIT(3,1) = 2.015
      TCRIT(3,2) = 1.812
      TCRIT(3,3) = 1.725
      TCRIT(5,1) = 1.645
      TCRIT(6,1) = 1.725

C
C  THE ARRRIVAL RATES AND SERVICE RATES LISTED BELOW ARE RATES PER UNIT
C  TIME.  SO IN THE SLAM CODE YOU SHOULD PUT 1/ARRRATE AND 1/SERRATE IN.
C

      ARRRATE = 1.
      SERRATE = 5./4.
      TRUVAL =  ARRRATE/(SERRATE*(SERRATE-ARRRATE))
      TI = ARRRATE/SERRATE
      RETURN
      END


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE FIGURE

C
C  AFTER THE FINAL RUN OF THE MONTE CARLO ANALYSIS THIS SUBROUTINE CALCULATES
C  THE APPROPRIATE STATISTICS
C

      INCLUDE '[RHOWARD.LATVER]SLAMCOM.FOR'

      INTEGER I,J,K
```

```
      DOUBLE PRECISION TEMP

      DO 10 J = 1,3
         DO 20 I = 1,6
            DO 30 K = 1,3
               COVER(I,J,K) = COVER(I,J,K)/NRUNS
               WIDTH(I,J,K) = SUMW(I,J,K)/NRUNS
               TEMP = SUMW2(I,J,K)/NRUNS - WIDTH(I,J,K)**2
               IF (TEMP.LE.0) THEN
                  STAND(I,J,K) = 0.0
               ELSE
                  STAND(I,J,K) = DSQRT(TEMP)
C                  WRITE (1,*) STAND(I,J,K)
               ENDIF
30          CONTINUE
20       CONTINUE
         MA(J) = SA(J)/NRUNS
         MQD(J) = SQD(J)/NRUNS
         MR(J) = SR(J)/NRUNS
         MK1(J) = SK1(J)/NRUNS
         MK2(J) = SK2(J)/NRUNS
         MPHI1(J) = SPHI1(J)/NRUNS
         MPHI2(J) = SPHI2(J)/NRUNS
         VA(J) = DSQRT(SA2(J)/NRUNS - MA(J)**2)
         VQD(J) = DSQRT(SQD2(J)/NRUNS - MQD(J)**2)
         VR(J) = DSQRT(SR2(J)/NRUNS - MR(J)**2)
         VK1(J) = DSQRT(SK12(J)/NRUNS - MK1(J)**2)
         VK2(J) = DSQRT(SK22(J)/NRUNS - MK2(J)**2)
         VPHI1(J) = DSQRT(SPHI12(J)/NRUNS - MPHI1(J)**2)
         VPHI2(J) = DSQRT(SPHI22(J)/NRUNS - MPHI2(J)**2)
         MMXD(J) = SMXD(J)/NRUNS
         MMMXD(J) = SMMXD(J)/NRUNS
         VMXD(J) = DSQRT(SMXD2(J)/NRUNS - MMXD(J)**2)
         VMMXD(J) = DSQRT(SMMXD2(J)/NRUNS - MMMXD(J)**2)
10    CONTINUE

      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE CONTROL

C
C  THIS SUBROUTINE CHANGES THE SAMPLE SIZE FOR THE MONTE CARLO ANALYSIS
C

      INCLUDE '[RHOWARD.LATVER]SLAMCOM.FOR'

      INTEGER I
```

```
      DO 10 I = 1,3
         NDAT = DSIZE(I)
         CALL GETMX(I)
         CALL KFMAIN(I)
         CALL MAINNB(I)
         CALL MAINB(I)
10    CONTINUE

      RETURN
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      SUBROUTINE GETMX(DS)

C
C  THIS SUBROUTINE CALCULATES THE AVERAGE VALUE OF THE OBSERVATIONS FOR
C  A GIVEN SAMPLE SIZE
C

      INCLUDE '[RHOWARD.LATVER]SLAMCOM.FOR'

      INTEGER DS,I
      DOUBLE PRECISION SUMX

      SUMX = 0.0
      DO 10 I = 1,NDAT
         SUMX = SUMX + X(I)
10    CONTINUE
      MX = SUMX/NDAT

      RETURN
      END
```

Vita

Captain Randall B. Howard was born on 25 November 1965 in Summitt, New Jersey. He graduated from Dunwoody High School, Dunwoody Georgia, in 1982. He then attended Birmingham Southern College, graduating in 1986 with a Bachelor of Science in Mathematics and Chemistry. Upon graduation, he received his commission in the United States Air Force through ROTC. He served as a Fuels Chemist and as Group Leader, Missile Fuels Research and Development for the Wright Research and Development Center at Wright Patterson AFB, OH from October 1986 to August 1990. In August 1990, he entered the Air Force Institute of Technology's School of Engineering, in the Operational Sciences Department.

Permanent address: 5532 Brinson Way
Norcross, GA 30092

Bibliography

1. Abraham, Bovas and Johannes Ledolter. *Statistical Methods for Forecasting*. New York: John Wiley & Sons, 1983.

2. Akaike, Hirotugu. "Markovian Representation of Stochastic Processes and Its Application to the Analysis of Autoregressive Moving Average Processes," *Annals of the Institute of Statistical Mathematics*, *26*(3):363–387 (1974).

3. Bauer, Kenneth W. and others. *Confidence Interval Estimation for Output of Discrete-Event Simulation Using the Kalman filter*. Research Proposal (Submitted to Air Force Office of Scientific Research), Wright-Paterson AFB, OH: School of Engineering, Air Force Institute of Technology (AU), September 1991.

4. Box, George E. P. and Gwilym M. Jenkins. *Time Series Analysis, Forecasting, and Control*. Oakland, CA: Holden-Day Publishers, 1976.

5. Chen, Bor-Chung and Robert G. Sargent. "Confidence Interval Estimation for the Parameter of Stationary Processes," *Management Science*, *36*(2):200–211 (February 1990).

6. Choobineh, F. and J. L. Ballard. "A Method of Confidence Interval Construction for Simulation Output Analysis," *Operations Research Letter*, *8*(5):265–270 (October 1989).

7. Committee on Armed Forces. United States Congress. *Department of Defense Critical Technology Plan*. Technical Report. 1990.

8. Fishman, George S. "The Allocation of Computer Time in Comparing Simulation Experiments," *Operations Research*, *18*(1):280–295 (September 1971).

9. Fishman, George S. "Grouping observations in digital simulation," *Management Science*, *24*(5):510–521 (January 1978).

10. Fishman, George S. *Principles of Discrete Event Simulation*. New York: John Wiley & Sons, 1978.

11. Gallagher, Mark A. *Multivariate Output Analysis of Discrete-Event Simulations using the Kalman Filter*. PhD Dissertation {*Not Yet Published*}, Wright-Patterson AFB, OH: School of Engineering, Air Force Institute of Technology (AU), January 1992.

12. Glynn, Peter W. and Donald L. Iglehart. "Simulation Output Analysis Using Standardized Time Series," *Mathematics of Operations Research*, *15*(1):1–16 (February 1990).

13. Goldsman, David and Lee Schruben. "New Confidence Interval Estimators Using Standardized Time Series," *Management Science*, *36*(3):393–397 (March 1990).

14. Grier, Peter. "Improving Systems with Simulators," *Air Force Magazine*, *73*(8):40–43 (August 1990).

15. Harvey, Andrew C. *Forecasting, Structural Time Series Models and the Kalman Filter*. New York: Cambridge University Press, 1989.

16. Heidelberger, Philip and Peter D. Welch. "A Spectral Method for Confidence Interval Generation and Run Length Control in Simulations," *Communications of the ACM*, *24*(4):233–245 (April 1981).

17. Hillier, Frederick S. and Gerald J. Lieberman. *Introduction to Operations Research*. New York: McGraw-Hill Publishing Company, 1990.

18. Kalman, R. E. "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, *82*:34–45 (1960).

19. Kalman, R. E. and R. S. Bucy. "New Results in Linear Filtering and Prediction Theory," *Journal of Basic Engineering*, *83*:95–108 (1961).

20. Kang, Keebom and David Goldsman. "The Correlation Between Mean and Variance Estimators in Computer Simulation," *IIE Transactions*, *22*(1):15–23 (March 1990).

21. Kang, Keebom and Bruce Schmeiser. "Graphical Methods for Evaluating and Comparing Confidence-Interval Procedures," *Operations Research*, *38*(3):546–552 (March–June 1990).

22. Kelton, W. David and Averill M. Law. "A New Approach for Dealing with the Startup Problem in Discrete Event Simulation," *Naval Research Logistics Quarterly*, *30*:641–658 (1983).

23. Lainiotis, Demetrios G. "Optimal Adaptive Estimation: Structure and Parameter Adaptation," *IEEE Transactions on Automatic control*, *AC-16*(2):160–170 (April 1971).

24. Lainiotis, Demetrios G. "Partitioning: A Unifying Framework for Adaptive Systems, I:Estimation," *Proceedings of the IEEE*, *64*(8):1126–1143 (August 1976).

25. Law, Averill M. "Confidence Intervals in Discrete Event Simulation: A comparison of Replication and Batch Means," *Naval Logistics Research Quarterly*, *23*:667–678 (1977).

26. Law, Averill M. "Statistical Analysis of Simulation Output Data," *Operations Research*, *31*(6):983–1029 (November 1983).

27. Law, Averill M. and John S. Carson. "A Sequential Procedure for Determining the Length of a Steady-State Simulation," *Operations Research*, *27*(5):1011–1025 (September–October 1979).

28. Law, Averill M. and W. David Kelton. "Confidence Intervals for Steady-State Simulation: I," *Operations Research*, *32*(6):1221–1239 (November 1984).

29. Law, Averill M. and W. David Kelton. *Simulation Modeling and Analysis*. New York: McGraw-Hill Inc., 1991.

30. Maybeck, Peter S. *Stochastic Models, Estimation, and Control Volume 1*. New York: Academic Press, Inc., 1979.

31. Maybeck, Peter S. *Stochastic Models, Estimation, and Control Volume 2*. New York: Academic Press, Inc., 1982.

32. Maybeck, Peter S. *Stochastic Models, Estimation, and Control Volume 3*. New York: Academic Press, Inc., 1982.

33. Maybeck, Peter S. "Adaptive Tracking of Dynamic Airborne Vehicles Based on (FLIR) Image Plane Intensity Data." In *Proceedings of the Third Bad Honnef Conference on Stochastic Differential Systems*, pages 284–305, Berlin: Springer-Verlag, 1986.

34. Maybeck, Peter S. and Karl P. Hentz. "Investigation of Moving-Bank Multiple Model Adaptive Algorithms," *AIAA Journal of Guidance, Control, and Dynamics*, *10*(1):90–96 (January–February 1987).

35. Mehra, R.K. "On-line Identification of Linear Dynamic Systems with Applications to Kalman Filtering," *IEEE Transactions on Automatic Control*, *AC-16*(1):753–768 (April 1971).

36. Meinhold, Richard. J. and Nozer. D. Singpurwalla. "Understanding the Kalman Filter," *Journal of the American Statistician*, *37*(3):123–127 (May 1983).

37. Meketon, Marc S. and Bruce Schmeiser. "Overlapping Batch Means: Something for Nothing?." In *Proceedings of the 1984 Winter Simulation Conference*, pages 227–230, Atlanta, GA: Institute of Electrical and Electronics Engineers, 1984.

38. Pritsker, Alan. B. *Introduction to Simulation and SLAM II*. West Lafayette, IN: Systems Publishing Corporation, 1986.

39. Ross, Sheldon M. *Introduction to Probability Models*. New York: Academic Press, Inc., 1989.

40. Schmeiser, Bruce. "Batch Size Effects in the Analysis of Simulation Output," *Operations Research*, *30*(3):556–568 (May–June 1982).

41. Schriber, Thomas J. and Richard W. Andrews. "ARMA-Based Confidence Intervals for Simulation Output Analysis," *American Journal of Mathematical and Management Sciences*, *4*(3–4):345–373 (1984).

42. Schruben, Lee. "Confidence Interval Estimation Using Standardized Time Series," *Operations Research*, *31*(6):1090–1108 (November–December 1983).

43. Schruben, Lee W. "Detecting Initialization Bias in Simulation Output," *Operations Research*, *30*(3):569–590 (May–June 1982).

44. Sheldon, Stuart N. and Peter S. Maybeck. "An Optimizing Design Strategy for Multiple Model Adaptive Estimation and Control," *Proceedings of the 29th IEEE Conference on Decision and Control*, pages 3522–3527 (December 1990).

45. Welch, Peter D. "The Statistical Analysis of Simulation Results." In Lavenberg, S. S., editor, *The Computer Performance Modeling Handbook*, pages 268–328, New York: Academic Press, 1983.

46. Wilson, James R. "Statistical Aspects of Simulation." In Brans, J. P., editor, *Operational Research '84: Proceedings of the Tenth International Conference on Operational Research*, pages 921–937, Amsterdam: Elsevier Science Publishers B.V. (North Holland), 1984.

47. Wilson, James R. and A. Alan B. Pritsker. "A Survey of research on the Simulation Startup Problem," *Simulation*, *31*(2):55–58 (August 1978).