ADAPTIVE FILTERING AND SMOOTHING IN DECONVOLUTION FOR SEISMIC PROCESSES

By

MELVILLE RONALD D'MELLO

Bachelor of Electrical Engineering
Aligarh Muslim University
Aligarh, Uttar Pradesh, India
1972

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
MASTER OF SCIENCE
December, 1974

ADAPTIVE FILTERING AND SMOOTHING IN DECONVOLUTION

FOR SEISMIC PROCESSES

Thesis Approved:

Craig S. Sims
Thesis Adviser

Edward L. Shreve

Rao Yarlagadda

D. D. Durham
Dean of the Graduate College

PREFACE

This study is concerned with the application of time domain techniques, specifically a form of Kalman Filter theory to the deconvolution problem in geophysical exploration processes. Primarily it is required to estimate the impulse response from the seismic trace.

I wish to express my thanks to Professor Craig S. Sims, my thesis adviser, for his invaluable guidance throughout this study. He has my sincere appreciation for devoting many hours of valuable time in discussing the problem.

Appreciation is also expressed to the other committee members, Dr. Rao Yarlagadda and Dr. Edward Shreve for their assistance in the preparation of the final manuscript. I would also like to thank Dr. Jerry Ware of CONOCO and Dr. Sven Treitel of AMOCO for their invaluable discussions during the various phases of this study.

In addition, appreciation is extended to my colleagues Mr. Y. W. Park and Dr. V. K. Bhandari for stimulating discussions on the subject. I would also like to acknowledge AMOCO, CONOCO and CITGO for their financial support. A special note of thanks to Mrs. Dixie Jennings is in order, for her excellent typing of the final copy.

Finally, I would like to express my gratitude to my parents, brother and sister for their understanding, encouragement, support and many sacrifices.

TABLE OF CONTENTS

LIST OF FIGURES

Figure | Page

Figure Page

CHAPTER I

INTRODUCTION

1.1 The Problem to be Considered

A basic problem of estimating a signal from a noisy set of data occurs in seismic exploration and in other areas of science such as communication and control theory.

A dynamite blast detonated near the surface of the earth or at a shallow depth in water gives rise to a sharp seismic disturbance. This initial pulse will be followed closely in time by the reflection of the source pulse from the surface of the earth. In general, there will also be near-surface multiple reflections, called reverberations, which are caused by reflections between a shallow strong reflector and the surface of the earth. For example, in marine shooting, the water-earth interface provides a strong change in acoustic impedance yielding a strong reflector and setting up multiple reflections in the water layer. In the case of land prospecting, the first hundred feet of the earth is generally a low velocity layer compared to the first, more rigid rock encountered. This top soil layer is referred to as the weathered layer and is the cause of near surface reverberations. Therefore the input wavelet which propagates into the seismic section of interest is the result of the input pulse from the source plus all of the trailing near surface reverberations.

As this rather ringy, input wavelet propagates into the seismic section of interest, it will be reflected back to the surface of the earth whenever a change in acoustical impedance is encountered, that is, if the geology changes from, say, sand to shale. The amplitude of the returned wavelet depends on the reflection coefficient at the change in geology, which in turn depends on the elastic properties of the rocks involved. These reflected events are detected on the surface of the earth by geophones placed at predetermined distances and recorded as a function of time, the record being called a seismogram. As a result of this simple description of the seismic reflection process, the seismogram can be modeled as the result of a weighted, delayed sum of this resulting input wavelet. Such a weighted delayed sum is a convolution of a resulting input wavelet with the impulse response of the seismic section of interest. The impulse response consists of all primary reflections plus all multiple reflections which occur between the many different layers of geology at depth. The object of deconvolution then is to remove the effect of this ringy, input wavelet and thereby get a better estimate of the impulse response of the seismic section of interest.

This problem is a data processing one where either prediction, filtering and/or smoothing can be applied depending on the time of interest. Techniques are available for treating prediction, filtering and smoothing problems in both the time domain and the frequency domain and attempts have been made in this direction. Wiener filter theory, based on time invariant systems, have been implemented in the time domain and is currently in widespread use. In this work, another time domain approach, specifically a modified Kalman filter technique which assumes that a random process can be modeled as the output of a linear system driven by white noise is pursued. By using a state variable formulation which is a time domain realization of the conventional transfer function formulation, the problem of deconvolution is formulated within a modern time domain format. Though both, Wiener filtering as well as Kalman filtering can be used in multichannel data acquisition problems (1), it is in the solution of time varying problems that the Kalman filter theory holds greatest promise.

1.2 The Approach

Given the autocorrelation function and the mean of a random process, a geophysical model is built which has this process as the output of a linear system excited by a white noise input. To do this, some basic assumptions are made. These are: (1) The wavelet is minimum phase time function. This means that the wavelet is a one sided transient with its energy concentrated around the zero time; and (2) the geology is unpredictable, which means that the sharp knife-like impulses are mutually uncorrelated. This leads to an interesting observation, namely, the autocorrelation function of the seismic trace is equal to the autocorelation function of the wavelet, as the uncorrelated elements of the impulses average out.

By using the above, a model has already been derived (14). However, in most deconvolution problems, the parameters describing the system are not completely defined. As for example, in a geophysical model, the velocity of the wavelet is a function of depth, while the reflection or refraction or transmission that results depends on the nature of the seismic layer it confronts. If these parameters randomly change and they are not measurable, then one has to rely on statistical data for modeling the system.

The geophysical process which has been described is assumed to be modeled by a set of state variable candidate models to account for the uncertainty. With the passage of time, as more data is obtained, the undefined parameters hopefully are learned and the states estimated, accomplishing the desired deconvolution. Such techniques are called system identification and adaptive filtering and are usually based on the assumption that the right model is one of the candidate models. Since these data processing results are not required to be computed on-line, another operation commonly known as smoothing can be used to achieve a refined estimate of the states. This operation makes use of the filtered estimates and the complete data set to improve the estimate of the states. However, existing smoothing algorithms are not adequate to solve the problem posed here. Therefore, an adaptive smoothing algorithm scheme is derived, taking into consideration the different candidate models as in the filtering part. The above sequence of operations, namely adaptive filtering and adaptive smoothing are simulated on the digital computer to test their performance in the task of deconvolution. While theoretically the estimate obtained is the optimal estimate under Gaussian conditions, the model for the seismic reflection process is assumed to have poisson inputs and the estimates are thus suboptimal. For this reason, simulation is especially important.

1.3 Objectives and Findings

There are three primary purposes of this work. The first is to apply recently developed adaptive time domain techniques to geophysical models and verify that deconvolution is achieved. The second objective is to develop new adaptive smoothing techniques to improve the accuracy of the estimate. The third aim is to modify a previous mathematical model of the seismic process under consideration, so that it is more realistic. This is done by using an input which is poisson distributed in time with random amplitudes.

The first objective is accomplished directly by applying an adaptive filtering scheme, referred to as "Estimation Under Uncertainty". This is a modification of Kalman filtering to account for model uncertainty. The results show that after enough observations have been taken, system identification seems to be achieved only in certain cases and is therefore not very reliable, although deconvolution is achieved with reasonable accuracy. There seems to be no way of determining how many observations should be taken before the operation is to be terminated and it is quite possible that a wrong model may be identified if enough data is not taken. The second objective is seen to be an extension of the adaptive filtering case and the results provide a marked improvement in the estimation of the state. The third is achieved by using the fact that the first two moments of a random process sometimes gives ample statistical knowledge of a linear system for simulation purposes. The results show that with poisson inputs, the estimates are quite accurate.

The work is application oriented and is not intended as a rigorous mathematical treatment of stochastic optimal estimation theory. The works which are most directly related to this investigation are discussed in Chapter II.

1.4 Organization

The remainder of this study is concerned with accomplishing the objectives and demonstrating the results mentioned in the previous section. In Chapter II, the adaptive filter algorithms are reviewed and the adaptive smoothing algorithms formulated. The geophysical model as suggested by Bayless and Brigham (14) is described and improved upon.

The simulation of the model and the algorithms for different poisson inputs are investigated in Chapter III under various uncertainty conditions and circumstances. The filtered and smoothed results are shown and compared. Chapter IV discusses the effects of improper modeling, in which the true model is not one of the candidate models. An iterative narrowing in procedure to remove the uncertainties in the model is also discussed. Chapter V contains a summary and conclusions of the results obtained in this work. Suggestions for further research are also included in this chapter.

CHAPTER II

ADAPTIVE FILTERING, SMOOTHING AND DECONVOLUTION

2.1 Background

Of particular interest in the area of seismic exploration, has been the need for restoring a signal to its original value by eliminating all undesired noise and distortion effects. This task is sometimes referred to as Deconvolution or Inverse Filtering.

Wiener (2) several years ago investigated problems of least-squares-estimation for stochastic processes and developed what has become known as Wiener Filter Theory. Later Kalman (3,4) investigated dynamical-state-estimation problems by specifying not the autocorrelation of a signal process, but a "model" for it as a linear dynamical system driven by white noise. The resulting filter is called the Kalman Filter.

Robinson (5,6), Rice (7), Kunetz (8), Robinson and Treitel (9), Clarke (10), Peakcock and Treitel (11), and many others have approached the deconvolution problem by using Wiener's method. Recently Treitel (12) has introduced the complex Wiener approach. Ulrych (13) has investigated another technique called homomorphic filtering. While Wiener's theory has been applied to time invariant systems, the solution of the resulting integral equation is usually difficult. This is especially true when extended to time varying cases.

Bayless and Brigham (14) and Crump (15) treated the deconvolution problem by using continuous and discrete Kalman filter techniques, respectively. The use of state space methods to treat physical and mathematical models lends itself easily to digital computer simulations and can solve many time varying problems as well.

It is the Bayless and Brigham paper (14) that forms the basis of this work. Hence it is appropriate here to consider this work in some detail.

2.1.1 Deconvolution in Seismic Exploration

Deconvolution is a technique to remove distortions to the signal in the course of its path through the seismic media, geophones, amplifiers, and recorders. In a sense, it is the filtering of the source that causes distortion to the signal and hence, inverse filtering must be applied to undo the effects of this undesired filtering. The schematic figure for treating the deconvolution problem is shown in Figure 1.

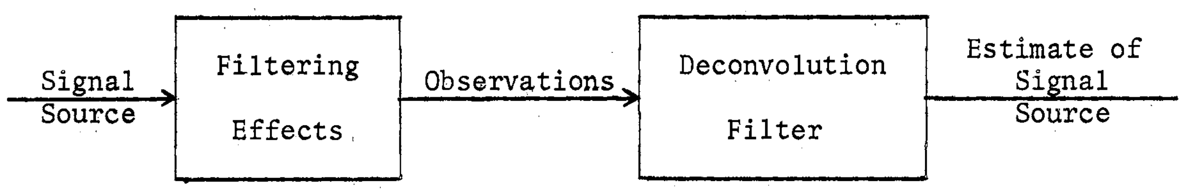

Figure 1. Basic Deconvolution Problem

Bayless and Brigham in their paper considered a basic wavelet model to be a minimum phase function.

To explain the complicated nature of the seismogram, Ricker (16) proposed the Ricker wavelet model. It is a time function and is the response of the earth to a sharp seismic disturbance. Hence the basic wavelet model is

$$h_2(t) = e^{-at} \text{ Sin } bt \qquad t \geq 0 \tag{2.1}$$

The seismic disturbance which creates the wavelet is an impulse, approximated by

$$h_1(t) = e^{-ct} \qquad t \geq 0 \tag{2.2}$$

and the input u(t) is a sequence of random poisson distributed impulses of the form

$$u(t) = \sum_{i=-\infty}^{\infty} \delta(t-t_i) - Q \tag{2.3}$$

where $E\{u(t)u(\tau)\} = Q\delta(t-\tau)$ and t_i is a random poisson variable with average time of occurrence, Q. It is desired to estimate the arrival times of these impulses, u(t). Because the Kalman filter estimates a state, the input u(t) is passed through an impulsive reflection generator (2.2). Its output $x_1(t)$ is the desired state to be estimated. Incidentally $x_1(t)$ is in the form of sharply decaying exponential waves, whose time of occurrence is required to be estimated. The simulation was done using an analog computer, as the plant dynamics and observation model were continuous in time. The schematic diagram of the Bayless and Brigham model is shown in Figure 2.

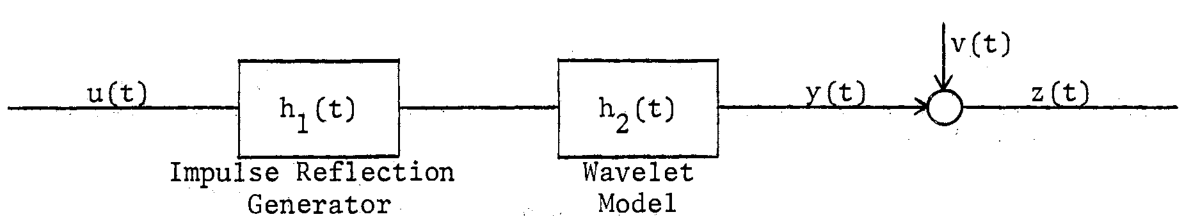

Figure 2. Seismic Reflection Process Model

For the system described above, in the impulsive generation model, there is not enough impulse or energy to drive the wavelet model. Therefore the observation cannot even register the presence of an impulse. Hence the reflection generator assumed by Bayless and Brigham is modified in this development, and is assumed to be

$$h_1(t) = ce^{-ct} \qquad t \geq 0 \tag{2.4}$$

because

$$\begin{aligned} \lim_{t \to 0} h_1(t) &= \lim_{t \to 0} ce^{-ct} \\ &= c \end{aligned} \tag{2.5}$$

For an impulse, the following must be satisfied;

$$\int_{t-\varepsilon}^{t+\varepsilon} \delta(t)dt = 1 \tag{2.6}$$

and for the above

$$\int_0^\infty ce^{-ct}\, dt = 1 \tag{2.7}$$

is satisfied.

Hence $h_1(t)$ represents an impulse and theoretically as t approaches zero, the impulse magnitude should increase to an infinite value. This is shown graphically in Figure 3. The modified model is indicated in Figures 4 and 5.

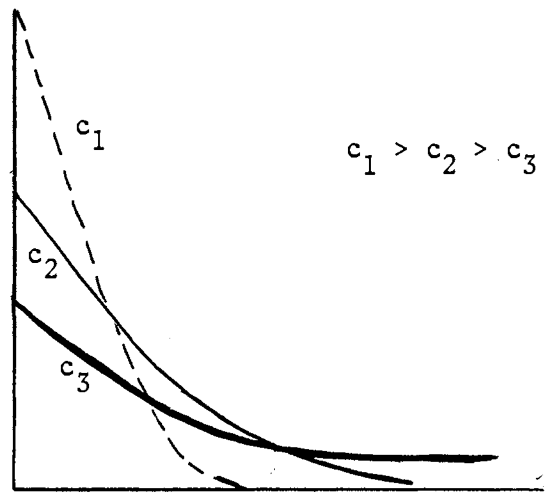

Figure 3. Approximation of an Impulse

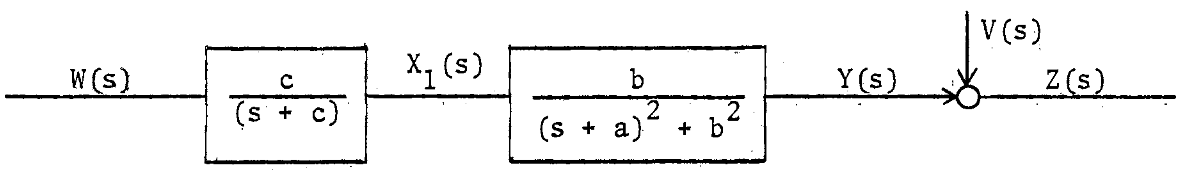

Figure 4. Modified Bayless and Brigham Model (Frequency Domain)

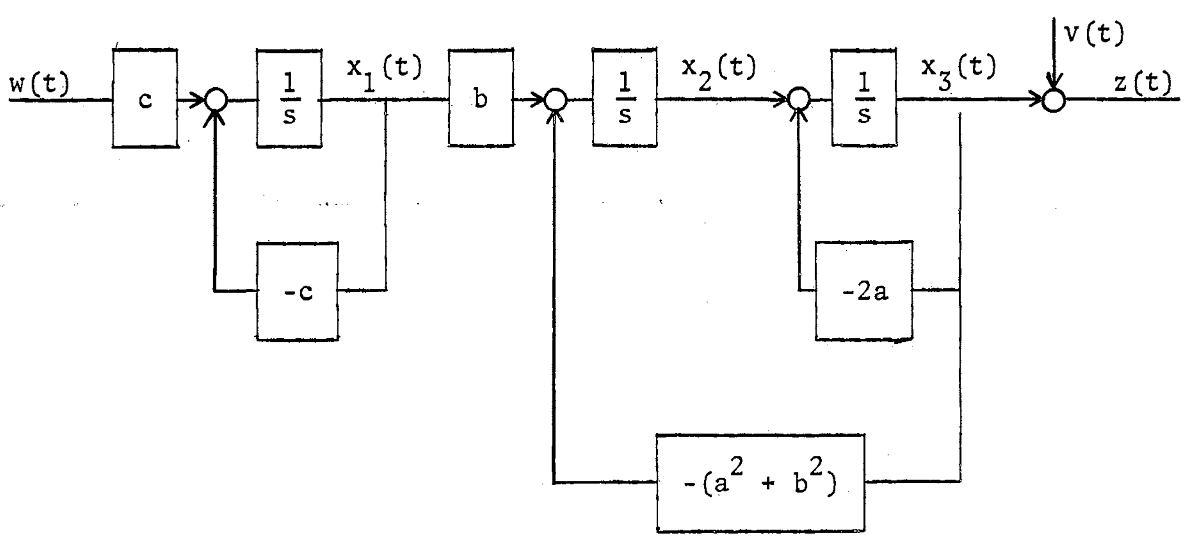

Figure 5. Modified Bayless and Brigham Model (State Space Form)

The dynamic equations describing the system are:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} -c & 0 & 0 \\ b & 0 & -(a^2+b^2) \\ 0 & 1 & -2a \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} c \\ 0 \\ 0 \end{bmatrix} w(t) \quad (2.8)$$

where w(t) is a poisson distributed input with zero mean and variance equal to Q and the observation model is

$$z(t) = [0 \quad 0 \quad 1] \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + v(t) \qquad (2.9)$$

where v(t) is zero mean Gaussian white noise with variance parameter R.

One of the problems with the Bayless and Brigham model is that it does not deal with random amplitude poisson distributed impulses. They suggest but do not investigate a form of input defined by

$$u(t) = \sum_{i=-\infty}^{\infty} m_i \, \delta(t-t_i) - E\left[\sum_{i=-\infty}^{\infty} m_i \, \delta(t-t_i) \right] \qquad (2.10)$$

where m_i's are random variables and t_i's are the poisson distributed random variables.

2.1.2 Comment

The topics considered here make the application to the deconvolution problem more general. The chief features to be considered in the remainder of the work are: (1) Since model parameters are in most cases not known, adaptive schemes can be applied. (2) Adaptive smoothing techniques can be incorporated as deconvolution is usually an off-line computational process; and (3) random amplitude poisson distributed impulses as defined in (2.10) can be implemented to make the model more realistic.

These features are used to help identify the correct model and then improve the estimate by applying the smoothing techniques. Thus even with the model not known correctly, deconvolution can still be achieved. Before applying these techniques to the modified Bayless and Brigham model, it is appropriate to give some consideration to the basic estimation under uncertainty scheme.

2.1.3 Estimation Under Uncertainty

In applications of Kalman filtering to seismic data processing, Bayless and Brigham have assumed the system parameters in the dynamic model to be well defined. In practice it is unlikely that the system parameters would be known. Hence it seems appropriate enough to apply adaptive filtering to "learn" the system parameters. Using the methods first derived by Lainiotis (17) and modified by Lee and Sims (18), an attempt is made to apply adaptive filtering to seismic problems. Since the approach taken by Lee and Sims has an influence on this thesis, the method is summarized below.

Given a set of candidate models, one of which is true, let θ_i index the ith model. The system dynamics are specified as

$$\theta_i : \quad \dot{x}_i(t) = F_i(t)x_i(t) + G_i(t)w_i(t) \qquad t \geq t_0 \tag{2.11}$$

where $i = 1,2,\cdots,N$ and N is finite; $x_i(t)$ is an n-dimensional vector representing the state of the system for the ith model; $w_i(t)$ is a qth order disturbance whose elements are zero mean white noise; $F_i(t)$ is a nxn matrix for the ith model; and $G_i(t)$ is a nxq matrix for the ith model.

The output of the ith model is a linear transformation of the state

$$y_i(t) = H_i(t)x_i(t) \tag{2.12}$$

where $H_i(t)$ is an mxn matrix for the ith model, and $y_i(t)$ is an m-dimensional output vector for the ith model. In general, $F_i(t)$, $G_i(t)$, and $H_i(t)$ are functions of time and subject to uncertainty.

The observation model is discrete and can be expressed as

$$z(k) = y_i(t_k) + v(k) \qquad k = 1,2,\cdots,N \tag{2.13}$$

and depends on which model is active at a given time, where v(k) is a m-vector measurement noise for zero mean, discrete Gaussian process; z(k) is an m-vector observation at time t_k.

It is assumed that the measurement noise and the plant noise under each hypothesis, θ_i, are independent Gaussian white noise sequences with zero mean and variances

$$E\{w_i(t)w_i^T(\tau)\} = Q_i(t)\delta(t-\tau) \tag{2.14}$$

$$E\{v_i(k)v_i^T(j)\} = R_i(k)\delta_{kj} \quad . \tag{2.15}$$

The expected value of the initial condition for the state is

$$E\{x_i(t_0)\} = \overline{x}_i(t_0) \tag{2.16}$$

and the initial condition for the covariance of error is

$$\begin{aligned} Var_{x_i}(t_0) = V_{x_i}(t_0) &= E\{\tilde{x}_i(t_0)\tilde{x}_i^T(t_0)\} \\ &= E\{[x_i(t_0) - \hat{x}(t_0)][x_i(t_0) - \hat{x}_i(t_0)]^T\} \end{aligned} \tag{2.17}$$

where $V_{x_i}(t_0)$ is an nxn matrix, and $Q_i(t)$, $R_i(t)$, $x_i(t_0)$ and $V_{x_i}(t_0)$ are all subject to uncertainty.

It is also assumed that $x_i(t_0)$ is independent of the noise sequence $\{w_i(t)\}$ and $\{v_i(k)\}$. Also an a priori probability, $p_r(\theta_i)$, is assumed for each candidate model. The set of measurements available up to stage k is denoted as $Z_k = \{z(1), z(2), \cdots, z(k)\}$.

The best estimate is determined by the conditional probability density $p(x(t)|Z_k)$. The conditional mean is

$$\hat{x}(t) = E\{x(t)|Z_k\} \qquad \text{for} \qquad t_k < t < t_{k+1}$$

and from the fundamental theorem of expectation,

$$\hat{x}(t) = \int_{-\infty}^{\infty} x(t)p(x(t)|Z_k)dx(t) \tag{2.18}$$

But the conditional density can be described as

$$p(x(t)|Z_k) = \sum p_r(\theta_i|Z_k)p(x(t)|Z_k,\theta_i) \tag{2.19}$$

and hence

$$\begin{aligned} \hat{x}(t) &= \int_{-\infty}^{\infty} x(t) \sum p_r(\theta_i|Z_k)p(x(t)|Z_k,\theta_i)dx(t) \\ &= \sum_{i=1}^{N} p_r(\theta_i|Z_k) \int_{-\infty}^{\infty} x(t)p(x(t)|Z_k,\theta_i)dx(t) \\ &= \sum_{i=1}^{N} p_r(\theta_i|Z_k)\hat{x}_i(t|Z_k) \end{aligned} \tag{2.20}$$

where $p_r(\theta_i|Z_k)$ is the a posteriori probability that the ith model is active at time t, and $\hat{x}_i(t|Z_k)$ is the conditional mean estimate of the ith candidate model expressed as

$$\hat{x}_i(t|Z_k) = E\{x_i(t)|Z_k\} = E\{x(t)|\theta_i,Z_k\}$$

$$= \int_{-\infty}^{\infty} x(t)p(x(t)|Z_k,\theta_i)dx(t) \quad . \qquad (2.21)$$

Hence the estimate of the state reduces to finding the sum of the products of the estimates and a posteriori probabilities of each candidate model active at time t, given the observation set Z_k. The solution to the estimation problem can be obtained in a recursive manner consisting of three parts.

(1) Predictor. In between observations, the estimator acts as a predictor. The conditional mean estimate of the state for each model is

$$\dot{\hat{x}}_i(t|Z_k) = F_i(t)\hat{x}_i(t|Z_k) \qquad (2.22)$$

and the covariance of the error of each candidate model is

$$\dot{V}_{x_i}(t|Z_k) = F_i(t)V_{x_i}(t|Z_k) + V_{x_i}(t|Z_k)F_i^T(t) + G_i(t)Q_i(t)G_i^T(t)$$

$$\text{for } i = 1,2,\cdots,N \qquad (2.23)$$

where $\hat{x}_i(t|Z_k)$ is defined in Equation 2.21 and

$$V_{x_i}(t|Z_k) = \mathrm{Var}\{x_i(t)|Z_k\}$$

$$= E\{[x_i(t) - \hat{x}_i(t|Z_k)][x_i(t) - \hat{x}_i(t|Z_k)]^T\} \quad .$$

The initial conditions for the state and covariance of error are indicated by Equations 2.16 and 2.17.

(2) Corrector. At time t_{k+1}, a measurement z_{k+1} is taken and the

updated mean and covariance are expressed as

$$\hat{x}_i(t_{k+1}|Z_{k+1}) = \hat{x}_i(t_{k+1}|Z_k) + K_i(t_{k+1})[z(k+1) - H_i\hat{x}_i(t_{k+1}|Z_k)] \tag{2.25}$$

and

$$V_{x_i}(t_{k+1}|Z_{k+1}) = V_{x_i}(t_{k+1}|Z_k) - K_i(t_{k+1})H_iV_{x_i}(t_{k+1}|Z_k)$$
$$= (I - K_iH_i)V_{x_i}(t_{k+1}|Z_k)(I - K_iH_i)^T + K_iR_iK_i^T$$
$$i = 1,2,\cdots N \tag{2.26}$$

where

$$K_i(t_{k+1}) = V_{x_i}(t_{k+1}|Z_k)H_i^T[H_iV_{x_i}(t_{k+1}|Z_k)H_i^T + R_i]^{-1} \tag{2.27}$$

is called the Kalman gain.

(3) Identifier. When a new measurement z_{k+1} is obtained, the a posteriori probability $p_r(\theta_i|Z_k)$ is updated. This a posteriori probability $p_r(\theta_i|Z_{k+1})$ provides a measure of certainty of whether the model is the true one and is expressed as

$$p_r(\theta_i|Z_{k+1}) = \left[1 + \sum_{\substack{j=1 \\ j\neq i}}^{N} L_{ji}\frac{p_r(\theta_j|Z_k)}{p_r(\theta_i|Z_k)}\right]^{-1} \tag{2.28}$$

where L_{ji} defined as the likelihood ratio is

$$L_{ji} = \left|\frac{H_iV_{x_i}H_i^T + R_i}{H_jV_{x_j}H_j^T + R_j}\right|^{\frac{1}{2}} \exp\{-\frac{1}{2}[\cdot]\} \tag{2.29}$$

and

$$[\cdot] = [z(k+1) - H_j\hat{x}_j(t_{k+1}|Z_k)]^T[H_jV_{x_j}H_j^T + R_j]^{-1}[z(k+1) - H_j\hat{x}_j(t_{k+1}|Z_k)]$$

$$- [z(k+1) - H_i\hat{x}_i(t_{k+1}|Z_k)]^T[H_iV_{x_i}H_i^T + R_i]^{-1}[z(k+1) - H_i\hat{x}_i(t_{k+1}|Z_k)] \quad .$$

When the true model is included as a candidate model, then the a posteriori probability of that model will converge to one with a sufficiently large number of observations, while the probability of the other candidate models approach zero.

This means that more weight is given to the correct model than to the others as the probability of the true model approaches one. In the event that the correct model has its a posteriori probability exactly equal to one, the adaptivity is removed as there is no more uncertainty involved and ordinary Kalman filter theory can be applied (19). The above algorithm serves a useful purpose in relation to the Bayless and Brigham paper in that if the model parameters are not known exactly, then a number of candidate models can be selected. As more observations are obtained, the weighting for each candidate model changes and in the long run the correct model should have the highest a posteriori probability and hence have its estimate weighted the most. This can be considered in effect as making the uncertain parameters slowly be known and to approach that of the correct model. The theory suggests that the highest a posteriori probability model be taken as the right one, because given an infinite number of observations, the right model should have its probability reach one.

While the above algorithm applies to the case of Gaussian inputs, the model proposed should have poisson distributed random noise inputs.

Hence the estimates are only the best linear estimates, conditioned on the individual candiates. Only an approximation to the conditional mean is obtained.

As the computation is not required in real time, it is plausible that smoothing techniques if applied could give a more accurate estimate. Hence it is appropriate to look into the process of smoothing.

2.1.4 Fixed Interval Smoothing

In smoothing, the estimate of the state is required at time t, given the noisy measurement data over the interval that includes the time t. While there are many methods of smoothing, fixed interval smoothing will be considered here.

One of the first developments in smoothing was by Bryson and Frazier (20) who used the calculus of variations approach by treating the above as an optimization problem. Later Rauch (21) and Meditch (22) published different treatments of smoothing. Then Fraser (23) and Mehra (24) developed a new form of smoothing, combining two filters, the forward and the backward Kalman filter. Mehra and Bryson (25) and Bryson and Henrikson (26) extended the work to colored noise for continuous discrete processes, respectively. Recently Kailath and Frost (27) have applied the innovations approach to least squares estimation in smoothing.

Mathematically, the smoothing problem can be stated as, given the observation set $Z(t_2) = \{Z(\tau),\ t_0 \leq \tau < t_2\}$, find the estimate $\hat{x}(t_1|t_2)$ where $t_1 < t_2$.

For the problem under consideration, the optimal linear fixed interval smoothing algorithms cannot be used. Because of model uncertainty, an adaptive technique of smoothing is developed. As more data is available, smoothing provides a better estimate than filtering.

2.2 Development of the Smoothing Algorithm

The system dynamics and observation model are the same as considered in Section 2.1.3, with θ_i representing the ith model. As a result of adaptive filtering, the a posteriori probability $p_r(\theta_i|Z_N)$ as well as the filtered estimates of each candidate model $\hat{x}_i(t|Z_k)$ at each time interval t are known.

Consider the conditional mean estimate at time t based on the entire observation set Z_N.

$$E\{x(t)|Z_N)\} = E_{\theta_i}\{E\{x(t)|Z_N,\theta_i\}\} \tag{2.30}$$

This can be written as the weighted sum of the smoothed estimates, conditioned on the ith model being correct with the weighting indicated by the final a posteriori probability as stored in the filtered algorithm.

$$E\{x(t)|Z_N\} = \sum_{i=1}^{N} p_r(\theta_i|Z_N)\hat{x}_i(t|Z_N) \tag{2.31}$$

where $Z_N = \{z(1),z(2),\cdots,z(N)\}$ = the entire data set.

$\hat{x}_i(t|Z_N)$ is the smoothed estimate of the ith candidate model at time t.

However, it is known in the minimum mean square sense, that the best linear estimate is also the conditional mean estimate (27). Hence

$$\hat{x}(t|Z_N) = E\{x(t)|Z_N\}$$

$$= \sum_{i=1}^{N} p_r(\theta_i|Z_N)\hat{x}_i(t|Z_N) \tag{2.32}$$

where $\hat{x}(t|Z_N)$ is the best smoothed estimate of the system states and is an n dimensional vector.

It is seen that the smoothed estimate is an extension of the filtering case except now the weighting for each model is considered a constant for each model and is the value at the last data point z(N).

The solution of $\hat{x}_i(t|Z_N)$ is obtained as follows (19). The smoothed estimate at time t is found from the equation

$$\dot{\hat{x}}_i(t|Z_N) = F_i(t)\hat{x}_i(t|Z_N) + S_i(t)[\hat{x}_i(t|Z_N) - \hat{x}_i(t)] \qquad (2.33)$$

$$\text{for } t_0 \leq t < t_N$$

where $\hat{x}_i(t)$ is the optimal filtered estimate of the ith model at time t and $S_i(t)$ is an nxn smoothing filter gain matrix for the ith model.

The terminal condition

$$\hat{x}_i(t_N|Z_N) = \hat{x}_i(t_N)$$

implies that at the final time t_N, the smoothed estimate equals the filtered estimate.

The smoothing filter gain $S_i(t)$ for the ith model is

$$S_i(t) = G_i(t)Q_i(t)G_i^T(t)V_{x_i}^{-1}(t|Z_k) \qquad (2.34)$$

where $V_{x_i}^{-1}(t|Z_k)$ is the inverse covariance of the error as obtained in the filter algorithm for the ith model at time t.

2.3 Application of Adaptive Processing to Deconvolution

In this section, the application of adaptive filtering and smoothing to the seismic processing problem of Bayless and Brigham is described.

Consider the parameter 'a' in the F matrix of the Bayless and

Brigham model to be unknown, and also let the variance of plant noise Q be unknown. Let a_i be the value of 'a' in the F matrix in the ith candidate model and let Q_i be the value of Q in the ith model. Hence the model can be described as

$$\begin{bmatrix} \dot{x}_{1i} \\ \dot{x}_{2i} \\ \dot{x}_{3i} \end{bmatrix} = \begin{bmatrix} -c & 0 & 0 \\ b & 0 & -(a_i^2+b^2) \\ 0 & 1 & -2a_i \end{bmatrix} \begin{bmatrix} x_{1i} \\ x_{2i} \\ x_{3i} \end{bmatrix} + \begin{bmatrix} c \\ 0 \\ 0 \end{bmatrix} w \tag{2.35}$$

where x_i is the state when the ith model is active.

The observation is discrete and expressed as

$$z_k = [0 \quad 0 \quad 1] \begin{bmatrix} x_{1i} \\ x_{2i} \\ x_{3i} \end{bmatrix} + v_k \tag{2.36}$$

where v_k is random white noise with zero mean and covariance

$$E\{v(k)v(j)\} = R\delta_{kj} \tag{2.37}$$

and w(t) is defined as

$$w(t) = \sum_{k=1}^{\infty} \delta(t-t_k) - Q_i \tag{2.38}$$

which is a poisson distribution with zero mean and covariance

$$E\{w(t)w(\tau)\} = Q_i\delta(t-\tau)$$

2.3.1 Filter Equations

(1) Predictor. To be used when no observations are available. The state predictor equation is:

$$\begin{bmatrix} \dot{\hat{x}}_{1i} \\ \dot{\hat{x}}_{2i} \\ \dot{\hat{x}}_{3i} \end{bmatrix} = \begin{bmatrix} -c & 0 & 0 \\ b & 0 & -(a_i^2+b^2) \\ 0 & 1 & -2a_i \end{bmatrix} \begin{bmatrix} \hat{x}_{1i} \\ \hat{x}_{2i} \\ \hat{x}_{3i} \end{bmatrix} \quad (2.39)$$

The variance equation is:

$$\begin{bmatrix} \dot{V}_{11i} & \dot{V}_{12i} & \dot{V}_{13i} \\ \dot{V}_{12i} & \dot{V}_{22i} & \dot{V}_{23i} \\ \dot{V}_{13i} & \dot{V}_{23i} & \dot{V}_{33i} \end{bmatrix} = \begin{bmatrix} -c & 0 & 0 \\ b & 0 & -(a_i^2+b^2) \\ 0 & 1 & -2a_i \end{bmatrix} \begin{bmatrix} V_{11i} & V_{12i} & V_{13i} \\ V_{12i} & V_{22i} & V_{23i} \\ V_{13i} & V_{23i} & V_{33i} \end{bmatrix}$$

$$+ \begin{bmatrix} V_{11i} & V_{12i} & V_{13i} \\ V_{12i} & V_{22i} & V_{23i} \\ V_{13i} & V_{23i} & V_{33i} \end{bmatrix} \begin{bmatrix} -c & 0 & 0 \\ b & 0 & -(a_i^2+b^2) \\ 0 & 1 & -2a_i \end{bmatrix}^T + \begin{bmatrix} c \\ 0 \\ 0 \end{bmatrix} Q_i [c \quad 0 \quad 0] \quad (2.40)$$

The initial conditions for the state equations and the variance equations are appropriately given.

(2) Corrector. When the observation z_{k+1} is obtained, the Kalman gain at $t = t_{k+1}$ is

$$\begin{bmatrix} K_{1i} \\ K_{2i} \\ K_{3i} \end{bmatrix} = \begin{bmatrix} V_{11i} & V_{12i} & V_{13i} \\ V_{12i} & V_{22i} & V_{23i} \\ V_{13i} & V_{23i} & V_{33i} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \left\{ [0\ 0\ 1] \begin{bmatrix} V_{11i} & V_{12i} & V_{13i} \\ V_{12i} & V_{22i} & V_{23i} \\ V_{13i} & V_{23i} & V_{33i} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + R \right\}^{-1} \tag{2.41}$$

The updated estimate at t_{k+1} given the observation z_{k+1} is

$$\begin{bmatrix} \hat{x}_{1i} \\ \hat{x}_{2i} \\ \hat{x}_{3i} \end{bmatrix} = \begin{bmatrix} \hat{x}_{1i} \\ \hat{x}_{2i} \\ \hat{x}_{3i} \end{bmatrix}_* + \begin{bmatrix} K_{1i} \\ K_{2i} \\ K_{3i} \end{bmatrix} \left[z(k+1) - [0\quad 0\quad 1] \begin{bmatrix} \hat{x}_{1i} \\ \hat{x}_{2i} \\ \hat{x}_{3i} \end{bmatrix}_* \right] \tag{2.42}$$

where * denotes the predicted estimate at t_{k+1}. The updated variance is

$$\begin{bmatrix} V_{11i} & V_{12i} & V_{13i} \\ V_{12i} & V_{22i} & V_{23i} \\ V_{13i} & V_{23i} & V_{33i} \end{bmatrix}$$

$$= \left\{ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} K_{1i} \\ K_{2i} \\ K_{3i} \end{bmatrix} [0\quad 0\quad 1] \right\} \begin{bmatrix} V_{11i} & V_{12i} & V_{13i} \\ V_{12i} & V_{22i} & V_{23i} \\ V_{13i} & V_{23i} & V_{33i} \end{bmatrix}_* \left\{ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right.$$

$$\left. - \begin{bmatrix} K_{1i} \\ K_{2i} \\ K_{3i} \end{bmatrix} [0\quad 0\quad 1] \right\}^T + \begin{bmatrix} K_{1i} \\ K_{2i} \\ K_{3i} \end{bmatrix} R [K_{1i}\quad K_{2i}\quad K_{3i}] \tag{2.43}$$

where * denotes the predicted variance at t_{k+1}.

(3) Identifier. The a priori probabilities are assumed at the initial time and each candidate model is assigned an a priori probability. The a posteriori probability is found out as indicated in Equations (2.28) and (2.29) when a new observation is obtained.

Finally the estimates of the states of the model are

$$\begin{bmatrix} \hat{x}_1(t) \\ \hat{x}_2(t) \\ \hat{x}_3(t) \end{bmatrix} = \sum_{i=1}^{N} p_r(\theta_i) \begin{bmatrix} \hat{x}_{1i}(t|Z_{k+1}) \\ \hat{x}_{2i}(t|Z_{k+1}) \\ \hat{x}_{3i}(t|Z_{k+1}) \end{bmatrix} . \qquad (2.44)$$

It should be noted that the Kalman gain in the corrector part as well as the variance of the error in the predictor and corrector parts can be precalculated as they are independent of observations.

2.3.2 Smoothing Equations

From the filtering equations above, all the estimates of each candidate model from the initial time t_0 to the final time t_N as well as $V_{x_i}(t|Z_k)$ for each model should be stored.

Smoothing can be considered as a process working backwards starting at the final time t_N and arriving at the initial time t_0.

The equations for adaptive smoothing with terminal conditions

$$\hat{x}_i(t_N|Z_N) = \hat{x}(t_N)$$

are

$$\begin{bmatrix} \dot{\hat{x}}_{1i} \\ \dot{\hat{x}}_{2i} \\ \dot{\hat{x}}_{3i} \end{bmatrix} = \begin{bmatrix} -c & 0 & 0 \\ b & 0 & -(a_i^2+b^2) \\ 0 & 1 & -2a_i \end{bmatrix} \begin{bmatrix} \hat{x}_{1i} \\ \hat{x}_{2i} \\ \hat{x}_{3i} \end{bmatrix}$$

$$+ \begin{bmatrix} S_{11i} & S_{12i} & S_{13i} \\ S_{12i} & S_{22i} & S_{23i} \\ S_{13i} & S_{23i} & S_{33i} \end{bmatrix} \begin{bmatrix} \hat{x}_{1i}(t|Z_N) - \hat{x}_{1i}(t) \\ \hat{x}_{2i}(t|Z_N) - \hat{x}_{2i}(t) \\ \hat{x}_{3i}(t|Z_N) - \hat{x}_{3i}(t) \end{bmatrix} .$$

The smoothing gain $S_i(t)$ is

$$\begin{bmatrix} S_{11i} & S_{12i} & S_{13i} \\ S_{12i} & S_{22i} & S_{23i} \\ S_{13i} & S_{23i} & S_{33i} \end{bmatrix} = \begin{bmatrix} c \\ 0 \\ 0 \end{bmatrix} Q_i [c \quad 0 \quad 0] V_{x_i}^{-1}(t|Z_N) \quad (2.46)$$

where $V_{x_i}^{-1}(t|Z_N)$ is the inverse of the error covariance of the ith model at time t.

Finally the smoothed estimate is

$$\begin{bmatrix} \hat{x}_1(t|Z_N) \\ \hat{x}_2(t|Z_N) \\ \hat{x}_3(t|Z_N) \end{bmatrix} = \sum_{i=1}^{N} p_r(\theta_i|Z_N) \begin{bmatrix} \hat{x}_{1i}(t|Z_N) \\ \hat{x}_{2i}(t|Z_N) \\ \hat{x}_{3i}(t|Z_N) \end{bmatrix} \quad (2.47)$$

where $p_r(\theta_i|Z_N)$ is the final value of the a posteriori probabilities found out from the filter part for the ith model.

In this chapter the adaptive filtering and smoothing equations for the Bayless and Brigham model have been given. To investigate the validity of the algorithms and to judge its performance when applied to seismic problems, computer simulations are carried out. The next chapter discusses these simulations.

CHAPTER III

SIMULATION RESULTS

3.1 Introduction

Simulation may be defined as a technique for conducting experiments on an analog, digital or hybrid computer, which involves certain types of mathematical and logical models that describe the behavior of the system over a period of time.

With computer simulation, insight can be gained into complex systems, formulation and testing of theories, and the prediction of the behavior of the systems in the future.

For the model of Bayless and Brigham, computer simulations are carried out to see whether the adaptive algorithms described previously, are adequate for the task of deconvolution and to evaluate their performance under a variety of circumstances. In particular, simulation is important due to the fact that the theory derived in Chapter II is not developed for poisson inputs. Hence one cannot predict without experimentation the performance of the algorithm.

3.2 The Method of Simulation

The total simulation program is rather involved, incorporating many facets, such as the generation of both poisson and Gaussian noise and the numerical integration of many equations. The simulation program (illustrated by the flow chart in Figure 6) is listed in the appendix, and some

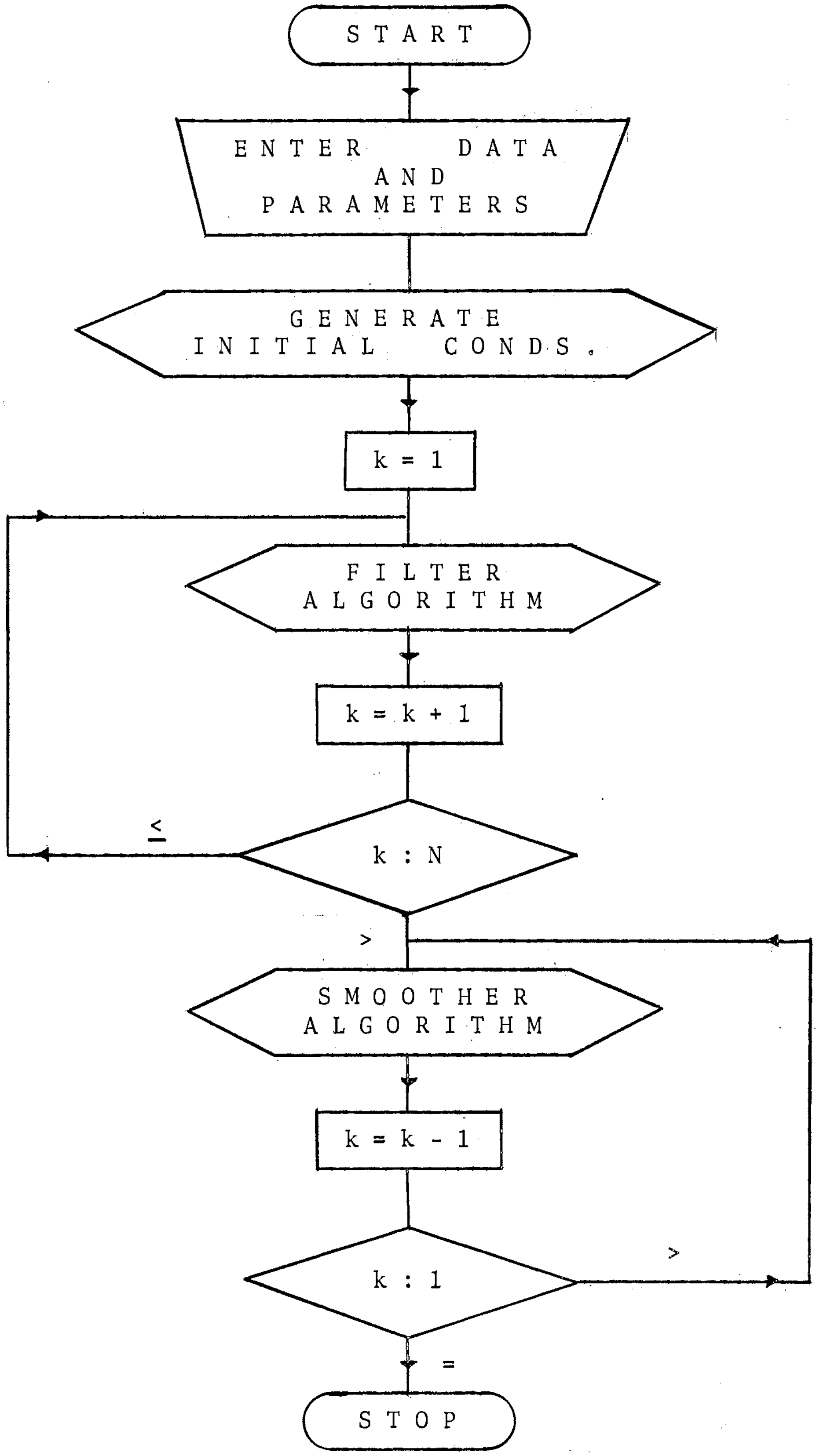

Figure 6. The Flow Chart for Simulation of the Model

of the important aspects of the program as well as experimental results are presented in the sequel.

3.2.1 Integration Method

The integration method used here is referred to as the fourth order Runge-Kutta Method (28). It is based on Simpson's Rule for finding the area under the given curve. There are, however, a number of such methods having minor variations and most computer libraries contain one or more as general integration procedures. The method is summarized below.

If a first order differential equation is

$$\dot{y} = f(x,y) \tag{3.1}$$

where $\dot{y}$ is the derivative of y with respect to x and

$$y(x_n) = y_n \tag{3.2}$$

then one computes

$$\begin{aligned} k_1 &= hf(x_n, y_n) \\ k_2 &= hf(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}) \\ k_3 &= hf(x_n + \frac{h}{2}, y_n = \frac{k_3}{2}) \\ k_4 &= hf(x_n + h, y_n + k_3) \end{aligned} \tag{3.3}$$

where h is the integration step size. The next value of y is evaluated according to

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{3.4}$$

The Runge-Kutta methods are self starting, have easy changes in step size and are particularly straight forward to apply on digital computers. However, they require a number of evaluations of the slopes f(x,y) at each step, equal to the order of the method.

An important consideration for this method of integration is the choice of step size 'h'. If the step size is too large, the result will be inaccurate, but if it is too small, an excessive amount of computer time is required. There are various ways of estimating 'h'. One rule of the thumb is to select h as 1/10 the smallest eigenvalue of the linearized system. Another scheme is simply run the program with two choices of step size of say .01 and .001 and compare. If the results are identical, then the larger step size is chosen.

3.2.2 Random Noise Generation

In the simulation, three kinds of random noise inputs are considered. In one case the equal amplitude random poisson distribution is the input to the system. In the second case the random amplitude poisson (RAP) distribution is the input. Gaussian additive white noise corrupts the output to give the observation for the model. In this section the generation of these noise disturbances are discussed.

(1) Equal Amplitude Poisson Distribution (EAP). A zero mean poisson distributed input of equal amplitude is represented by

$$u(t) = \sum_{i=-\infty}^{\infty} \delta(t-t_i) - Q \qquad (3.5)$$

where t_i's are the random poisson variables, and Q is the average number of occurrences per second. Let $y_1, y_2, \cdots, y_n$ be random independent variables that have an exponential density function

$$p_Y(y) = Qe^{-QY} \qquad y > 0$$
$$= 0 \qquad \text{otherwise} \qquad . \tag{3.6}$$

Starting at an arbitrary time t = 0, assume

$$
\begin{aligned}
t_1 &= y_1 \\
t_2 &= y_1 + y_2 \\
t_3 &= y_1 + y_2 + y_3 \\
&\vdots \\
t_n &= y_1 + y_2 + \cdots + y_n \qquad .
\end{aligned} \tag{3.7}
$$

Then it can be proved that the random variables t_i are poisson in nature (29).

To realize an exponential distribution function from a uniform distributed function between (0,1), a nonlinear transformation is required. This transformation is represented in the block diagram of Figure 7, where the random variable X has the density function

$$p_X(x) = 1 \qquad 0 \leq x < 1$$
$$= 0 \qquad \text{otherwise} \tag{3.8}$$

and the random variable Y has the density function

$$p_Y(y) = Qe^{-Qy} \qquad y \geq 0$$
$$= 0 \qquad \text{otherwise} \qquad . \tag{3.9}$$

It is required to find the transformation h = g(x).

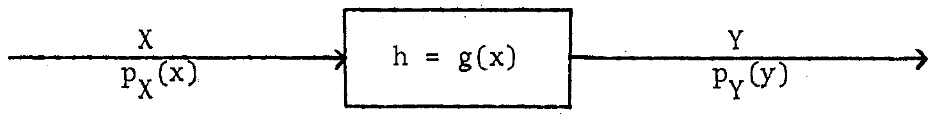

Figure 7. Transformation of a Random Variable

If X and Y are related on a one-to-one basis, then equating the probabilities gives

$$P(X \leq x) = P(Y \leq y) \tag{3.10}$$

or

$$\int_0^x dx = \int_0^y Qe^{-Qy}\, dy$$

or

$$x = (1 - e^{-Qy})$$

or

$$y = \frac{-\ln(1 - x)}{Q} = \frac{-\ln(x)}{Q} \tag{3.11}$$

where Q is a parameter of the exponential process.

Hence, by this nonlinear transformation, for various values of X, a sequence of random variables Y are generated. These random variables are independent and possess the exponential density function. The poisson distributed inputs can then be obtained by Equation (3.7).

The statistics of the white noise process are

$$E\{u(t)\} = 0 \quad \text{and} \quad E\{u(t)u(\tau)\} = Q\delta(t-\tau) \quad . \tag{3.12}$$

Therefore, given the first and second moments of this random process, the equal amplitude poisson distribution can be implemented.

(2) Random Amplitude Poisson Distribution (RAP). The RAP distribution input can be expressed as

$$u(t) = \sum_{i=-\infty}^{\infty} m_i \delta(t-t_i) - E\left[\sum_{i=-\infty}^{\infty} m_i \delta(t-t_i)\right] \tag{3.13}$$

where m_i is a random distribution of mean m which reflects the intensity of the impulses.

Of the statistics of random processes, the first and second moments are most useful. In fact first and second order moments provide necessary and sufficient information for problems involving linear systems and/or Gaussian random processes. As in the previous case, efforts will be centered on deriving the first and second moments to obtain a statistical knowledge of impulse processes.

The first moment is (30)

$$\begin{aligned} E\{u(t)\} &= E[\sum m_i \delta(t-t_i)] \\ &= \sum E[m_i]E[\delta(t-t_i)] \\ &= mQ \end{aligned} \tag{3.14}$$

where $E\{m_i\} = m$ and Q is the mean for the poisson process.

The second moment is

$$\begin{aligned} E\{u(t)u(\tau)\} &= \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} E[m_i m_j^T]E[\delta(t-t_i-\tau)\delta(t-t_j)] \\ &= Q\rho(0)\delta(t-t_i) + Q \sum_{n=1}^{\infty} \rho(n)f_n(t-\tau) \end{aligned} \tag{3.15}$$

where $\rho(i) = E\{m_k m_{k+i}^T\}$; f_n is the probability density function associated with n consecutive intervals of poisson process t_i; and Q is the variance of the poisson process.

If in the above, an assumption is made that the random variable m_i is zero mean and independent, then considerable simplification results.

In fact if m_i is uniformly distributed between -1 and 1, then its first and second moments are 0 and 1/3, respectively, so that the first and second moments of the RAP distribution input are

$$E\{u(t)\} = 0$$

and

$$E\{u(t)u(\tau)\} = \frac{Q}{3}\,\delta(t-\tau) \quad . \qquad (3.16)$$

A little modification is required in the equal amplitude poisson distribution described in the previous section to get the RAP distribution. When the pulses are initiated onto the system in a poisson manner, the amplitude is made to vary according to the various values of the random variable m_i, with the above statistics. It is seen that the previous section on the equal amplitude poisson distribution problem is a particular case of the more general RAP distribution problem. The m_i's are no more random, but deterministic with a value of 1.

(3) Gaussian Distribution. There are many applications in simulation and analysis of dynamical systems which require large amounts of pseudo random numbers. There has been a great interest in recent years in the generation of pseudo random numbers on a digital computer. Since these numbers are generated by deterministic means, the term pseudo random is applied to the generated numbers. Chambers (31) and many others, treat the subject of uniform pseudo random sequences.

The two most popular methods of generating uniform random numbers are the multiplicative method and the mixed congruential method. The first method can be described by the recurrence formula

$$X_{i+1} = AX_i \text{ (modulo M)} \quad .$$

The second is described by

$$X_{i+1} = AX_i + C \text{ (modulo M)} \tag{3.18}$$

where A, M and C are constants usually chosen to yield a long period and other desirable statistical properties of the sequence. On division of AX_i or $AX_i + C$ by M and taking the remainder, the next random number is obtained. As each random number is obtained, it can be divided by M to be normalized to the unit interval. The numbers so obtained will approximate a uniform distribution very closely.

Only the multiplicative method will be considered in this section. The generator described in Equation (3.17) may be easily implemented on any digital computer. The multiplier A is always chosen as $8K\pm3$ where K is a positive integer. This is done to insure a full period of M/4. Also the starting number X_0 must be an odd integer to obtain a full period. There are no other requirements except that A must be chosen to yield good statistical properties in the generated numbers.

Brown and Rowland (32) have obtained satisfactory statistical properties from a pseudo random generator with A as 19971, $M = 2^{20}$, and X_0 as 31571. These generated numbers are uniformly distributed on (0,1). These can be converted into a zero mean, unity variance, Gaussian distribution by the exact closed form relation developed by Box and Muller (33). These are

$$Z_1 = (-2 \ln X_1)^{1/2} \text{ Cos } 2\pi X_2 \tag{3.19}$$

$$Z_2 = (-2 \ln X_1)^{1/2} \text{ Sin } 2\pi X_2 \tag{3.20}$$

where X_1 and X_2 are uniformly distributed random variables and Z_1 and Z_2 are Gaussian random variables.

3.2.3 Parameter Selection

The Bayless and Brigham model being simulated has the same parameters chosen as is indicated in Section 2.1.1, except where differences are allowed to account for model uncertainty. However a question arises as to the value of the input impulse to the model.

The impulsive reflection generator described by Equation (2.4) has at the instant the pulse is initiated onto the system, an output of value c. It is seen that the input impulse amplitude is a function of the type of integration done and the step size chosen. As the value of c for the model has been chosen to be 1000, this means a small step size must be selected for the iterations to work. The choice was made to be .0001. Hence given the step size and the method of integration (fourth order Runge-Kutta), it is seen that the input impulse magnitude required is approximately 10^4. Also the initial conditions for the states are at zero and the initial variance conditions are chosen to be zero except V_{11} the variance of state x_1, the output of the reflection generator. This value is taken to be 1000.

It is thought that the points discussed in this section are those which required some explanation. The remainder of the program involves the mechanics of implementing the flow chart.

3.3 Comments

A number of experiments were conducted. These experiments are motivated by the following questions.

(1) To what degree is system identification achieved?

(2) How significant a role does adaptive smoothing play in the estimation of the state?

(3) How do the results of adaptive filtering under uncertainty compare with ordinary Kalman filtering for a known model?

(4) How does adaptive smoothing compare with smoothing with certain knowledge of the model?

(5) What are the effects of the level of measurement noise on the performance of the estimator?

3.4 Case With Equal Amplitude Poisson Input

The modified Bayless and Brigham model is used with the parameter 'a' in the F matrix in Equation (2.8) uncertain and simulated on the computer. The poisson input is generated as explained in the previous section as is the Gaussian white noise process. The integration method is the Runge-Kutta fourth order method as explained previously. In the simulation presented below, the integration step size is 0.1 msec. and the observation is taken every 0.5 msec.

3.4.1 Effects of Uncertainty in the Model

In this experiment four values of 'a' are possible for the candidate models. They are 'a' = 150, 100, 50, and 10 while the other parameters are fixed at b = 100π, c = 1000 and Q = 500. The candidate models under consideration are:

$$\theta_1 : \dot{\underline{X}}(t) = \begin{bmatrix} -1000 & 0 & 0 \\ 100\pi & 0 & -(150^2+100^2\pi^2) \\ 0 & 1 & -300 \end{bmatrix} \underline{X}(t) + \begin{bmatrix} 1000 \\ 0 \\ 0 \end{bmatrix} w(t)$$

$$\theta_2 : \dot{\underline{X}}(t) = \begin{bmatrix} -1000 & 0 & 0 \\ -100\pi & 0 & -(100^2+100^2\pi^2) \\ 0 & 1 & -200 \end{bmatrix} \underline{X}(t) + \begin{bmatrix} 1000 \\ 0 \\ 0 \end{bmatrix} w(t)$$

$$\theta_3 : \dot{\underline{X}}(t) = \begin{bmatrix} -1000 & 0 & 0 \\ -100\pi & 0 & -(50^2+100^2\pi^2) \\ 0 & 1 & -100 \end{bmatrix} \underline{X}(t) + \begin{bmatrix} 1000 \\ 0 \\ 0 \end{bmatrix} w(t)$$

$$\theta_4 : \dot{\underline{X}}(t) = \begin{bmatrix} -1000 & 0 & 0 \\ -100\pi & 0 & -(10^2+100^2\pi^2) \\ 0 & 1 & -20 \end{bmatrix} \underline{X}(t) + \begin{bmatrix} 1000 \\ 0 \\ 0 \end{bmatrix} w(t).$$

The model with 'a' as 50 or θ_3 is true. The output and the measurement are assumed as

$$y(t) = [0 \quad 0 \quad 1]\underline{X}(t)$$

and

$$z(k) = y(t_k) + v(k) .$$

The covariance of the plant is

$$E\{w(t)w(\tau)\} = Q\delta(t-\tau)$$

where Q is 500 and w(t) is poisson distributed. The covariance of the measurement noise is

$$E\{v(k)v^T(j)\} = R\delta_{kj}$$

where R is 10^{-5} and v(k) is zero mean Gaussian white noise. The initial conditions $\overline{x}_i(t_0)$ and $V_{x_i}(t_0)$ are

$$\overline{X}_i(t_0) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad ; \quad V_{x_i}(t_0) = \begin{bmatrix} 1000 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} .$$

The a priori probability assigned to each model is

$$p_r(\theta_i) = 0.25 \qquad \text{for all } i .$$

Result 1. The desired state to be estimated is x_1, the spiked output from the reflection generator. A typical single run is shown in Figure 8, where the estimated and the actual values are plotted. The associated identification capability is plotted in Figure 9. It is seen that adaptive filtering provides estimates of the impulses after a noticeable lag. This is because of the discrete observation model. An event that occurs between observations is not noticed until the next observation. Also the identification of θ_3 as the true model is indicated after a number of samples, by the greater a posteriori probability $p_r(\theta_3|Z_k)$.

Result 2. The effect of adaptive smoothing is seen on the estimate of x_1, as its true and smoothed values are plotted in Figure 10. Also the observation set from which this estimate x_1 is made is shown in Figure 11. To bring out the comparison between the adaptive smoothing and filtering operations, a plot of true, filtered and smoothed values for the first fifty observations is shown in Figure 12. It is clearly seen that the process of adaptive smoothing removes the lag in the

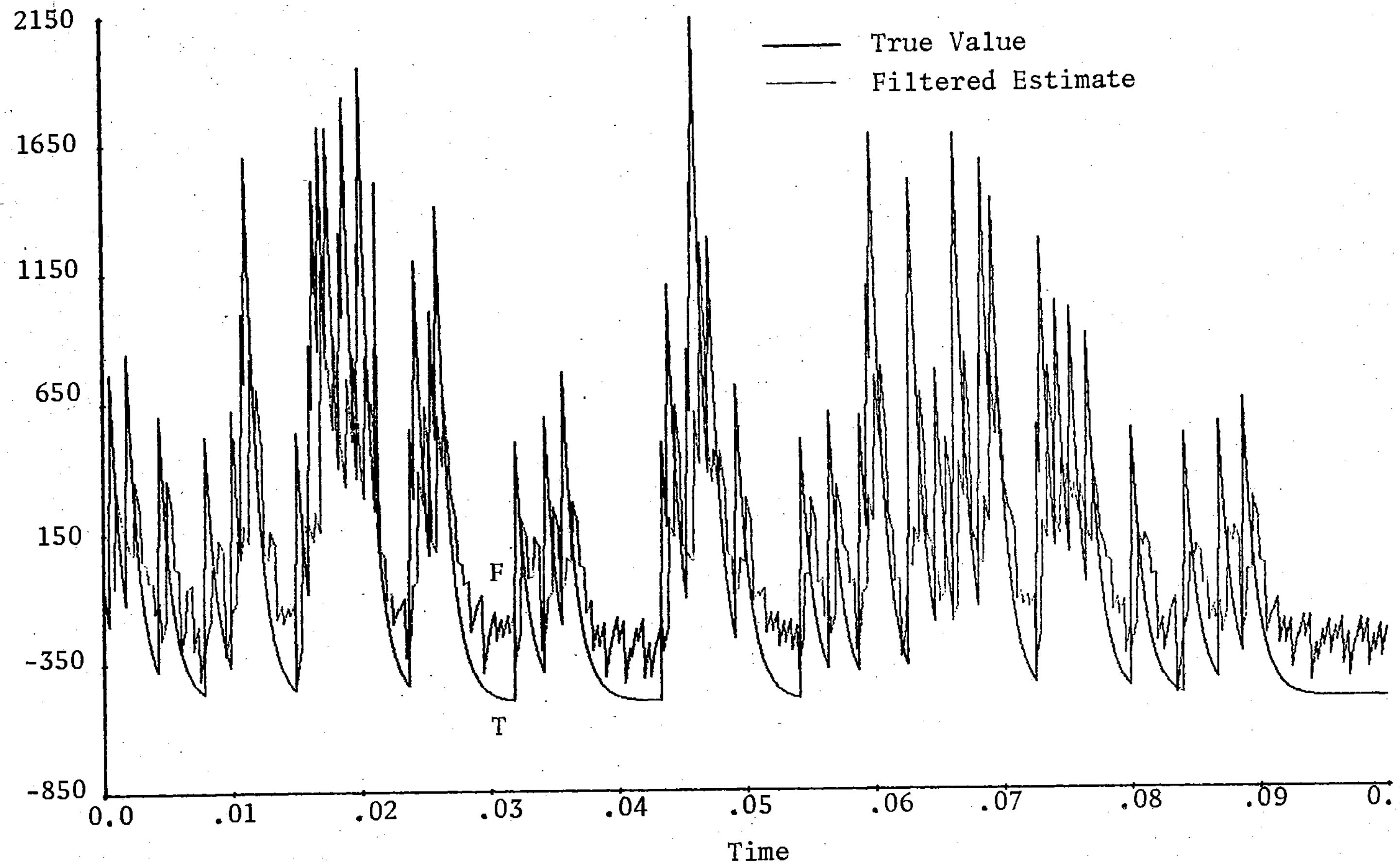

Figure 8. True and Filtered Estimate of State x_1 for EAP Input With Parameter 'a' Uncertain

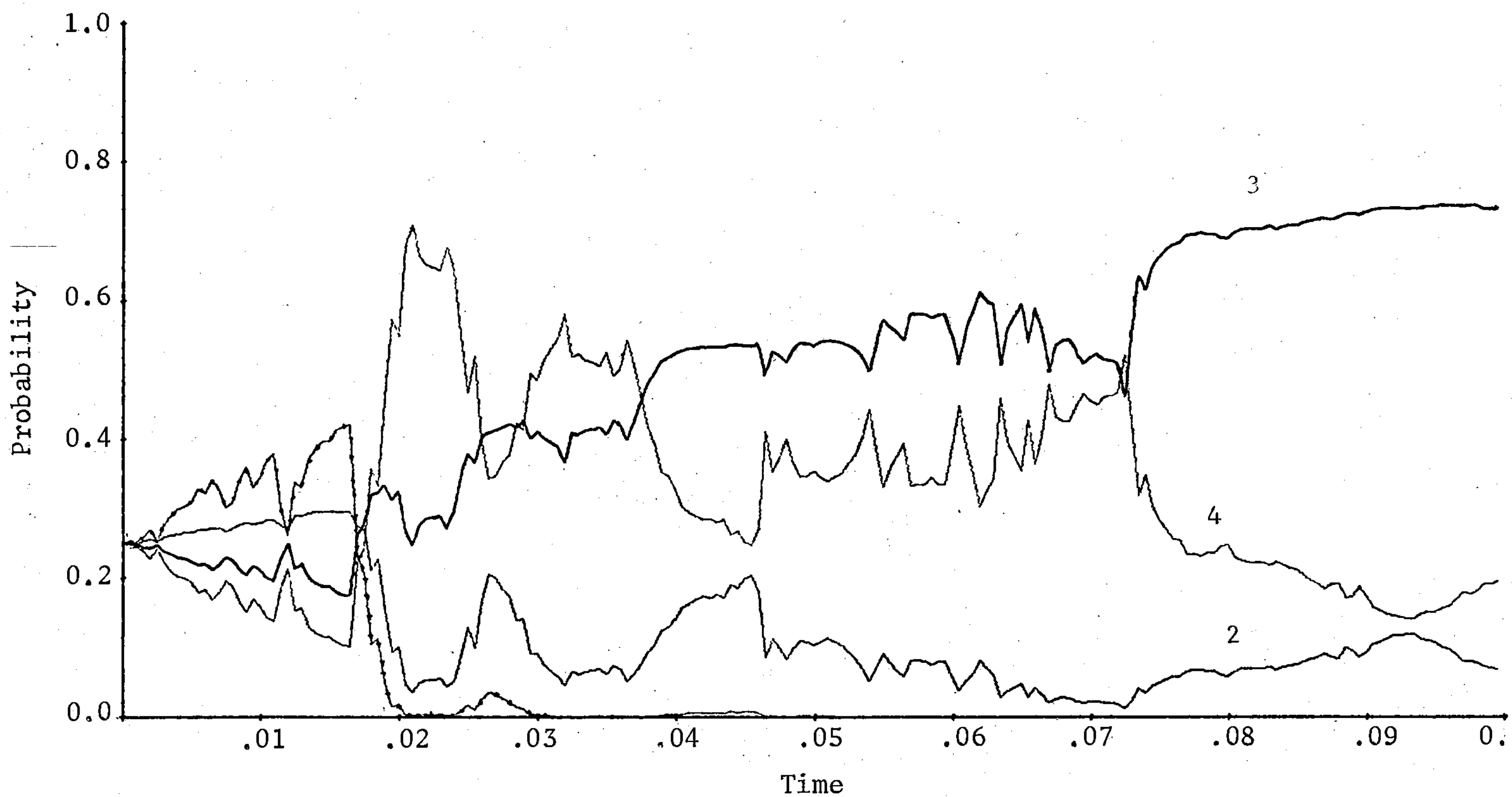

Figure 9. The a Posteriori Probability for the Four Candidate Models With 'a' Uncertain

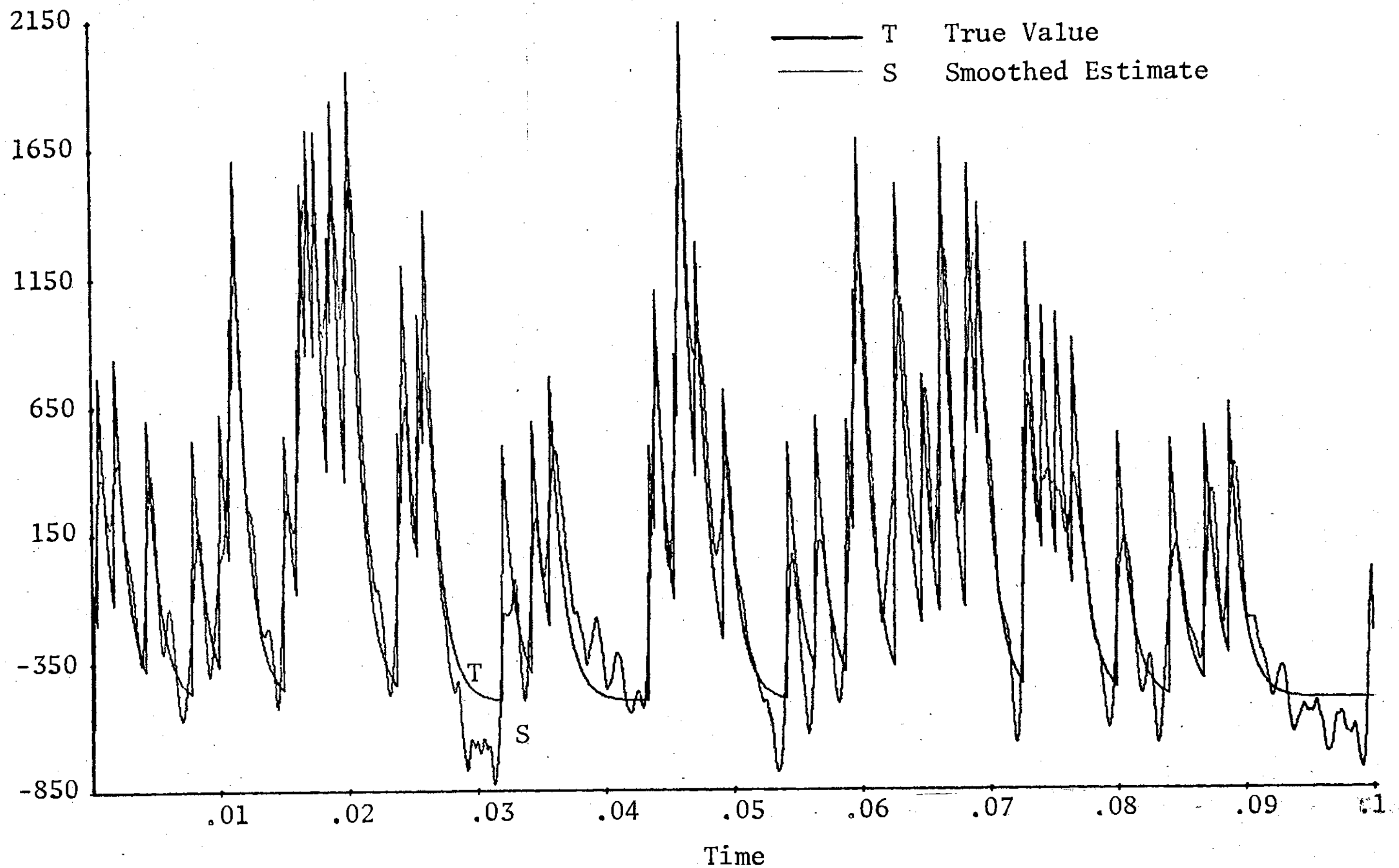

Figure 10. True and Smoothed Estimate of State x_1 for EAP Input With Parameter 'a' Uncertain

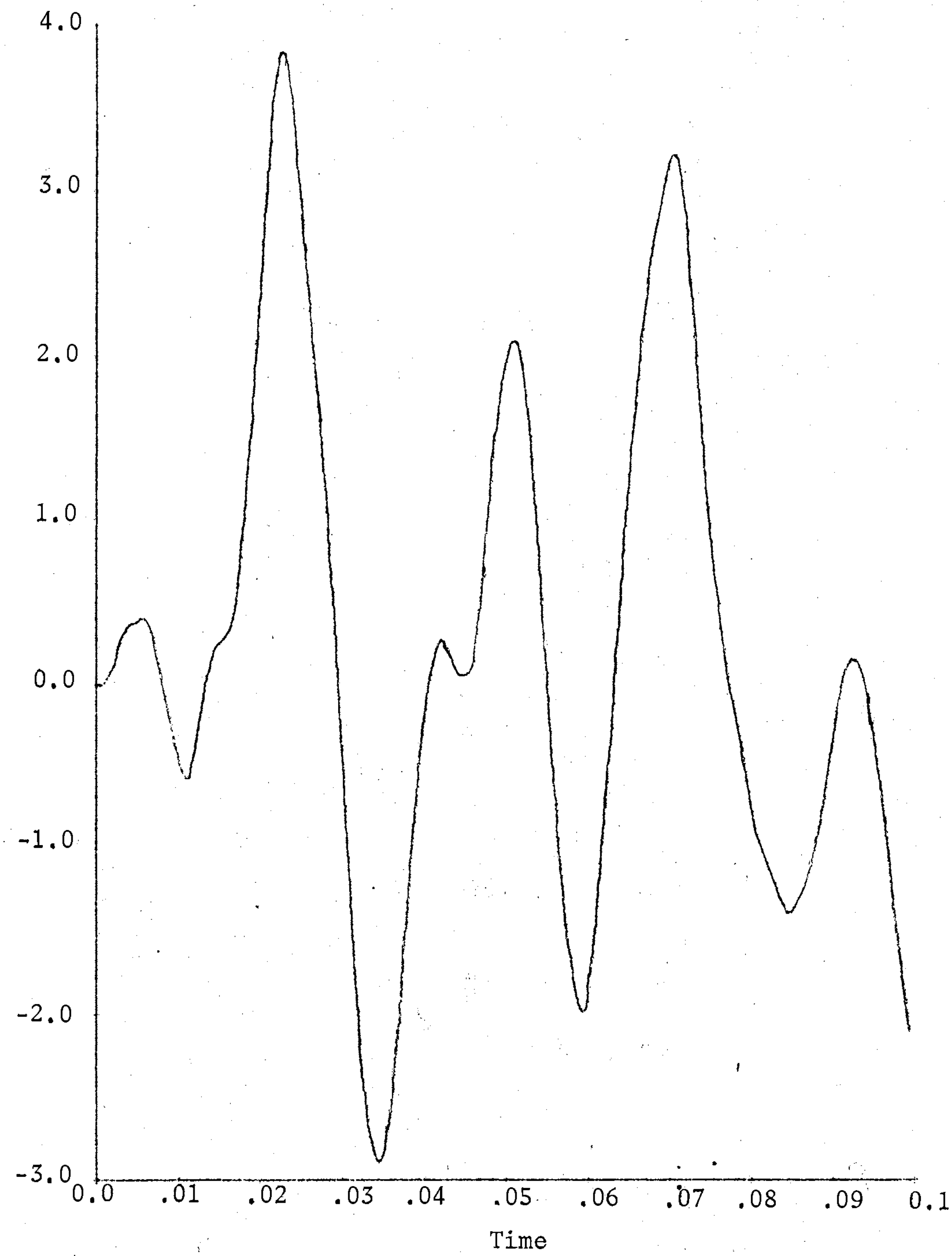

Figure 11. The Observation set for Equal Amplitude Poisson Input

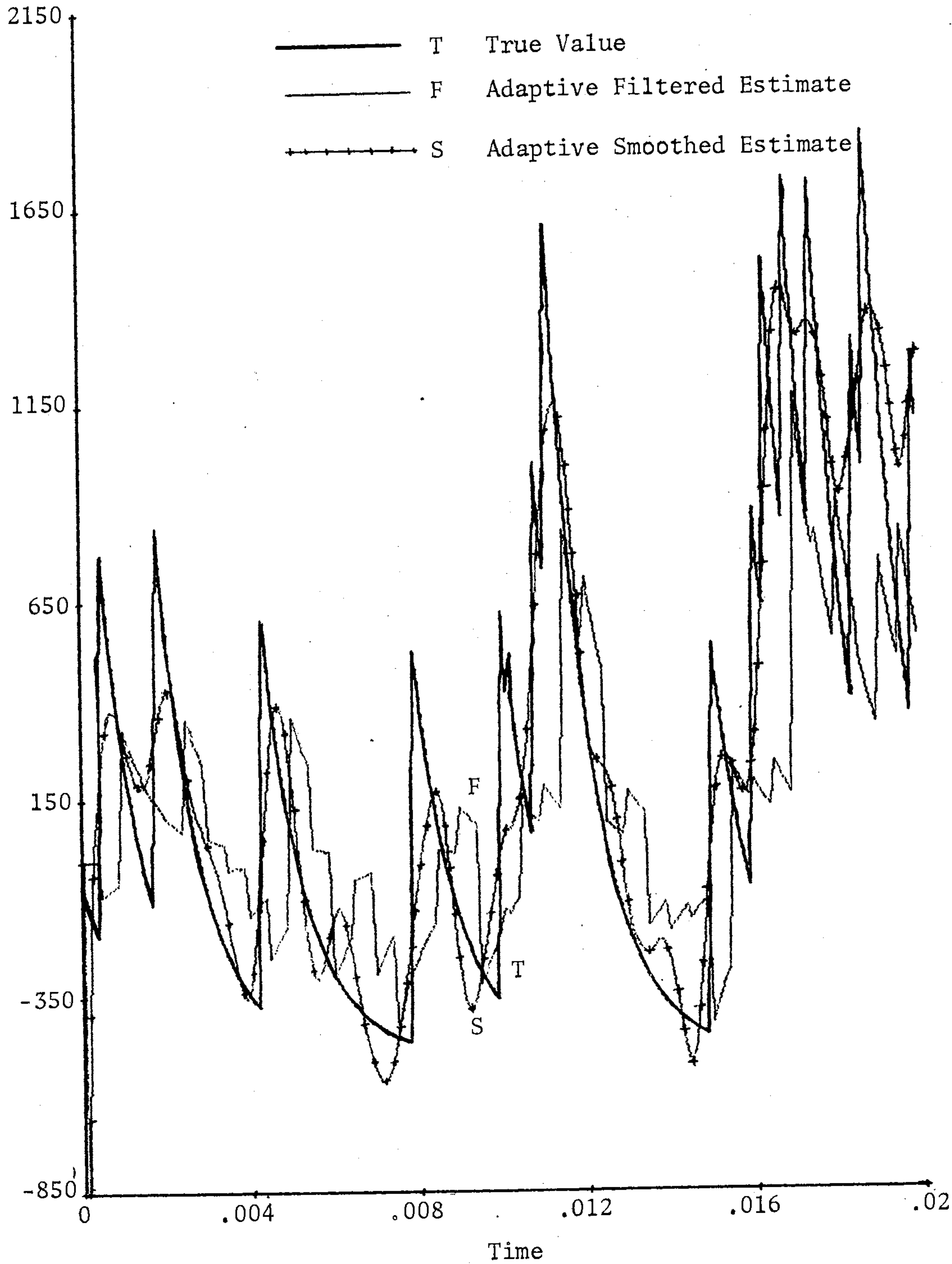

Figure 12. True, Filtered and Smoothed Estimate for EAP Input With 'a' Uncertain

estimate as observed in the filtered results and hence makes for accurate results.

Result 3. To demonstrate the effects of adaptive filtering when compared with ordinary Kalman filtering for a known model, the true value, adaptive estimate, and Kalman estimate are plotted in Figure 13. Even with uncertainty in the model, the adaptive filtered estimate is almost the same as the Kalman filtered estimate. This result is significant, since it shows that one can accomplish accurate deconvolution even with model uncertainty.

Result 4. It is important to judge the effects of adaptive smoothing as compared with nonadaptive smoothing, or smoothing under uncertainty. Figure 14 is a plot of the true value, the adaptive, and nonadaptive smoothed estimates of x_1 for the first fifty observations. There seems to be very little difference in the results of the smoothing methods and model uncertainty does not impair the estimate of the state x_1.

Result 5. Another experiment conducted is based on the assumption that there is uncertainty in the frequency of occurrence, Q, of impulses for the poisson input, instead of uncertainty in the model. The model is fixed with the parameters of the model given as $a = 50$, $b = 100\pi$, $c = 1000$, while Q has four possible values: 900, 700, 500, and 100. The third candidate model indexed by θ_3 is the actual model, and has a value of $Q = 500$. Each model is given the same initial conditions as in the previous experiments and the same a priori probabilities. A single run is made and the actual and filtered values of x_1 is plotted in Figure 15. Again a lag is noticed in the filtered estimate as was seen in the case of uncertainty in the parameter 'a'. Also the identification capability

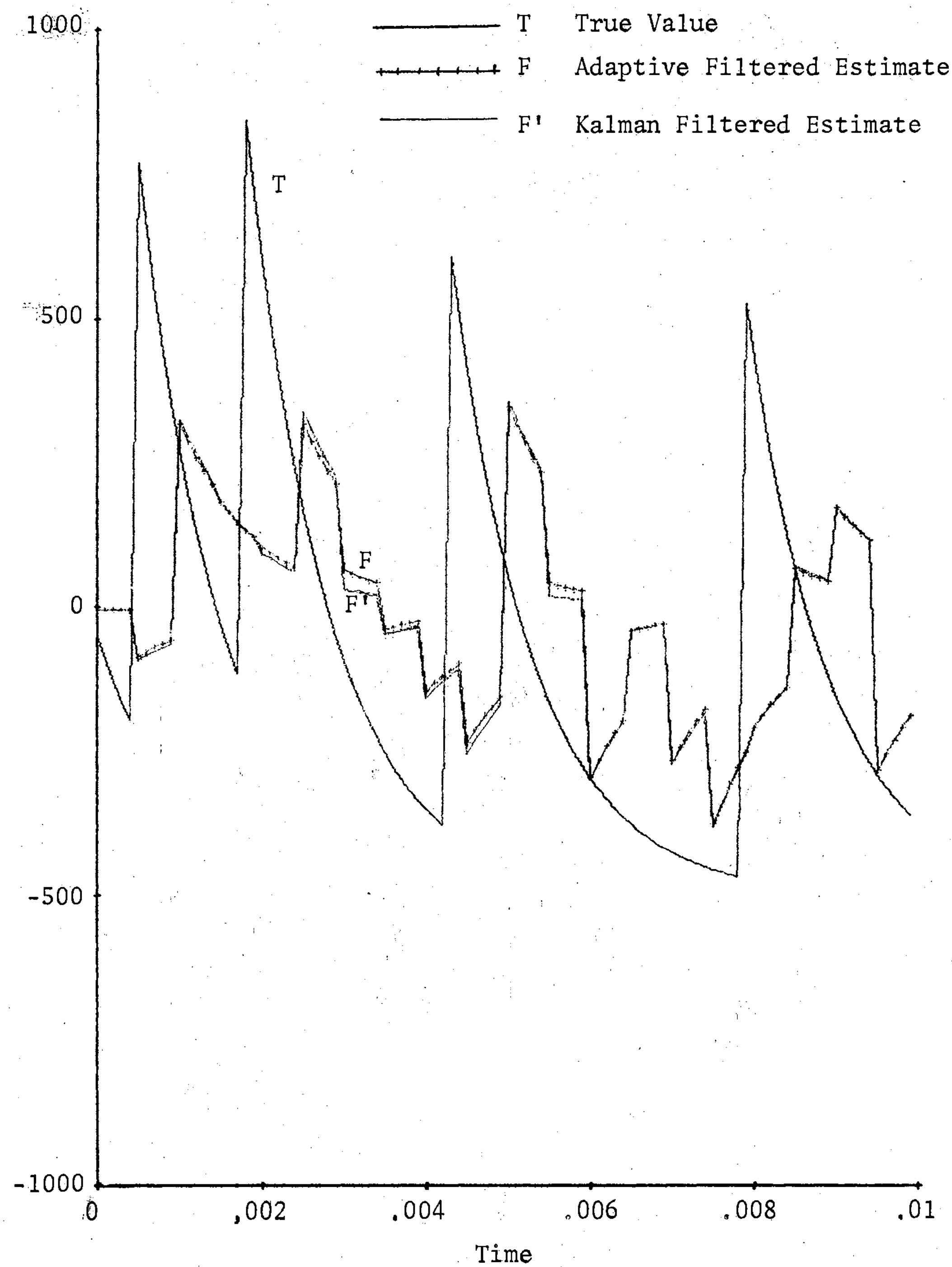

Figure 13. True, Adaptive Filtered and Kalman Filtered Estimate for EAP Input With 'a' Uncertain

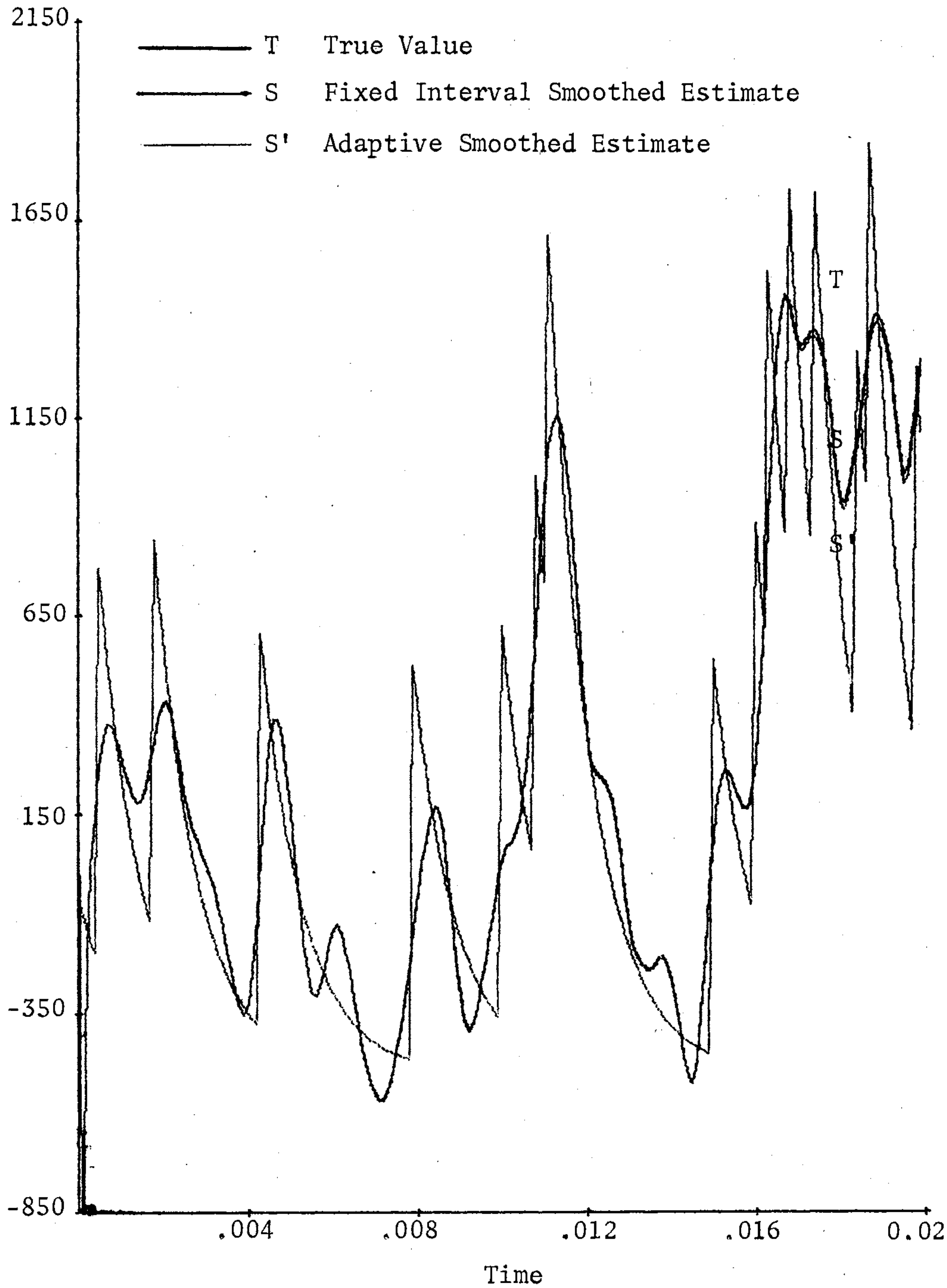

Figure 14. True, Adaptive Smoothed and Fixed Interval Smoothed Estimate for EAP Input With 'a' Uncertain

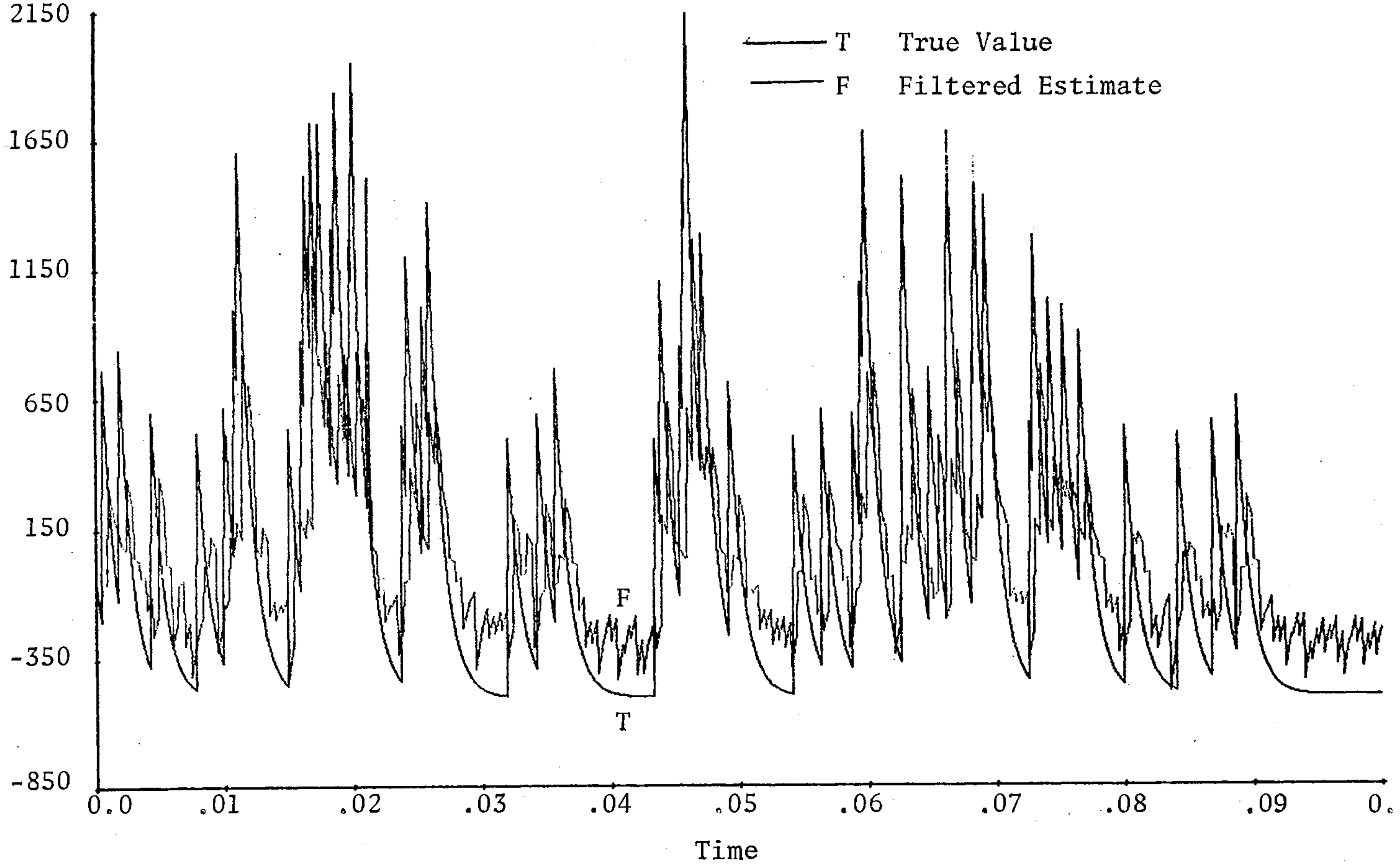

Figure 15. True and Filtered Estimate of State x_1 for EAP Input With 'Q' Uncertain

is seen in Figure 16 as the correct model reaches the highest a posteriori probability. It is noticed, however, that the number of samples required for the system identification to take place is larger. It is thought that the degree of uncertainty in the system due to the unknown Q has a second order effect and consequently, a larger amount of data needs to be taken. The shape of some of the probability curves resemble a saw-tooth curve and if the observation is curtailed at say .08 secs. the wrong model will have a higher a posteriori probability. Hence it can be inferred that system identification is not very reliable. However, even when the posterior probabilities are misleading, accurate deconvolution is accomplished.

Result 6. Adaptive smoothing is carried out for the above case. Figure 17 shows a plot of true and smoothed values of the state x_1. In order to compare the adaptive smoothing and the adaptive filtering estimates, a plot of all three, the true, filtered, and smoothed values is shown in Figure 18 for the first fifty observations. As is evidenced, there is a definite improvement due to the process of adaptive smoothing in obtaining the estimate of x_1.

3.4.2 Effects of Measurement Noise

In the experiments conducted so far, the measurement noise has been assumed to be very low ($R = 10^{-5}$). However, in practical situations, the level of noise is larger. It is of importance, therefore, to examine the effect of noise on the system performance. Hence, in the following experiment, the level of measurement noise is raised by a factor of 100 to $R = 10^{-3}$ and is compared with $R = 10^{-5}$ where R is the variance of the noise.

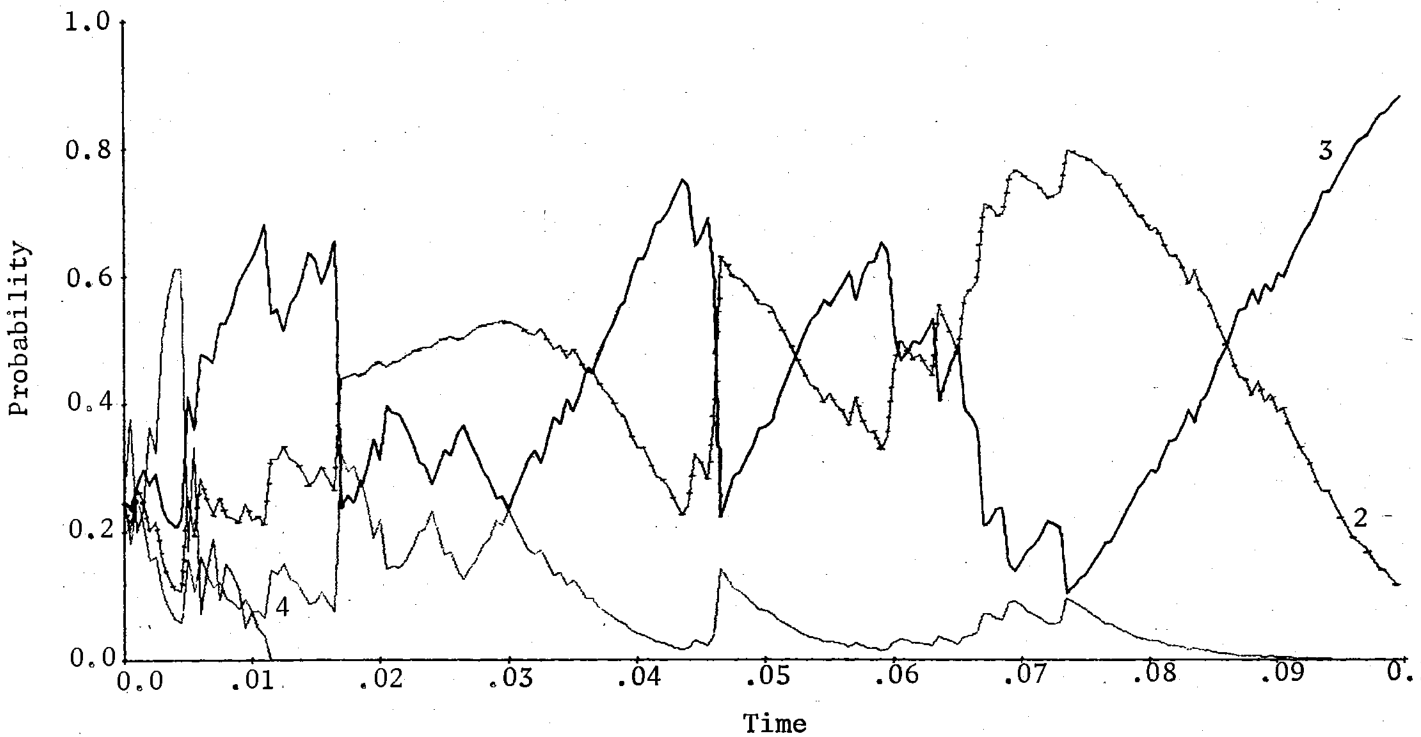

Figure 16. The a Posteriori Probability for the Four Candidate Models With 'Q' Uncertain

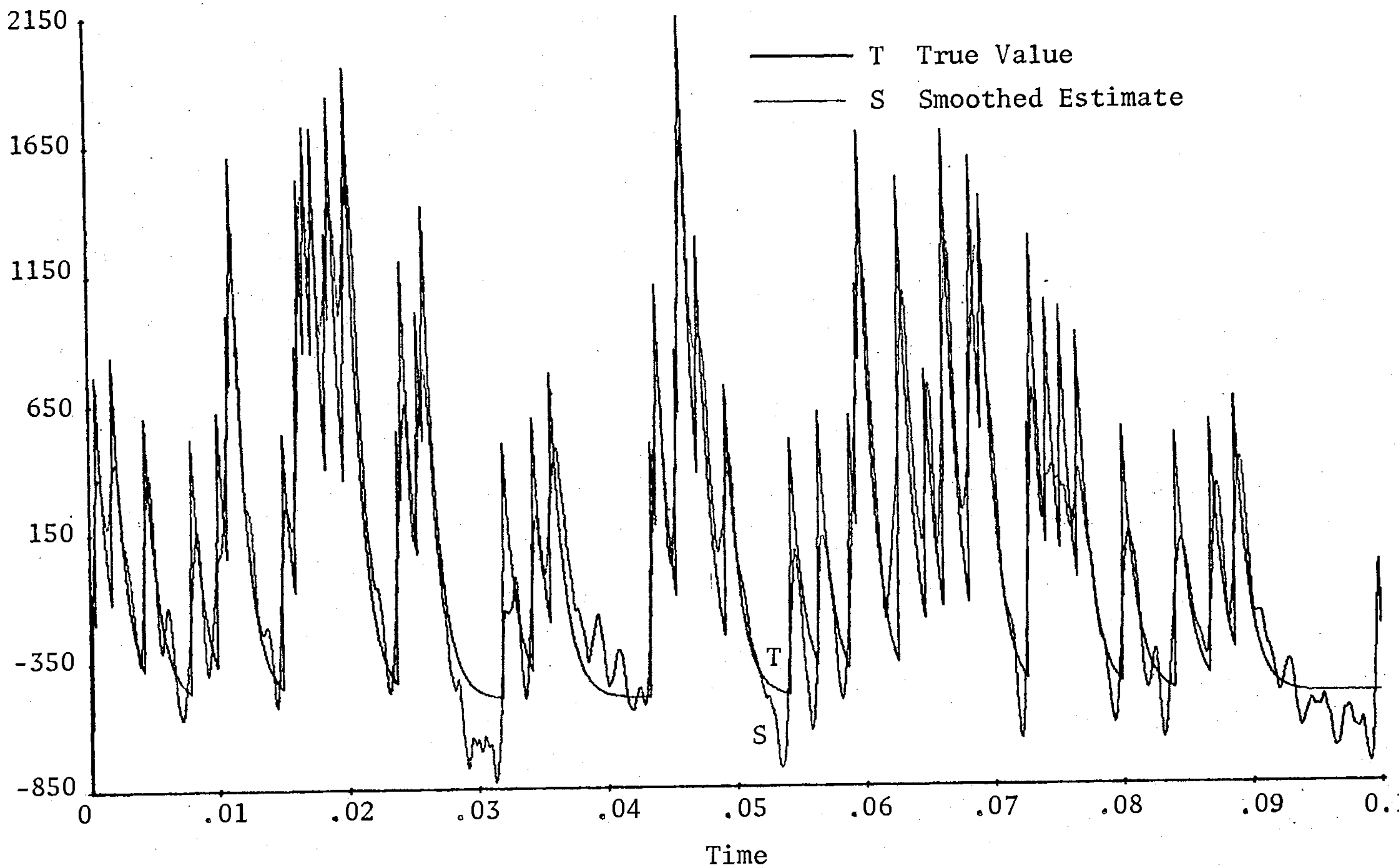

Figure 17. True and Smoothed Estimate of State x_1 for EAP Input With 'Q' Uncertain

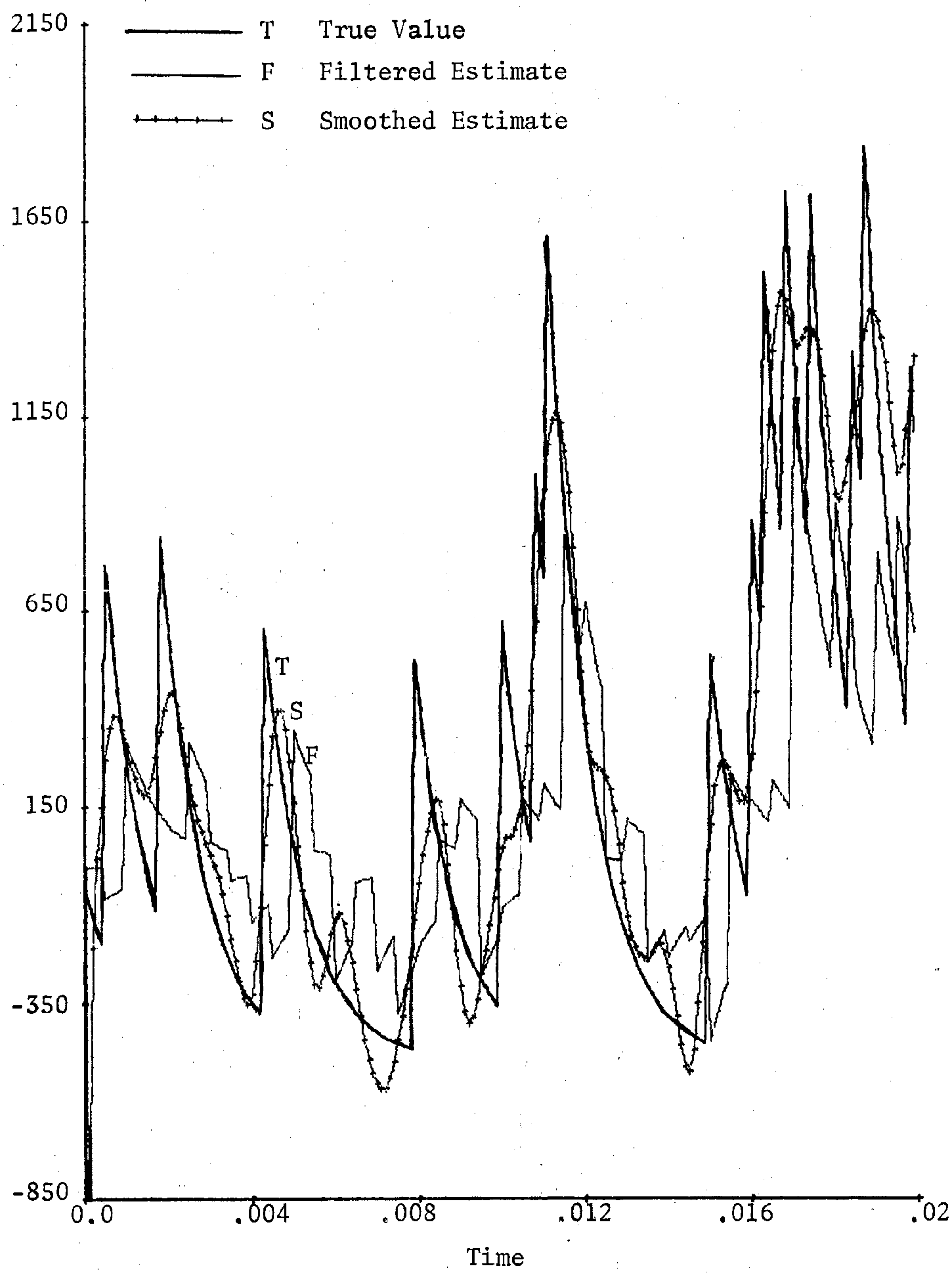

Figure 18. True, Filtered and Smoothed Estimate for EAP Input With 'Q' Uncertain

The system has the uncertain parameter 'a' in the model and has four possible values specified as 150, 100, 50 and 10. The other parameters are fixed as before.

Result 1. A plot has been shown in Figure 19 of the true value and the filtered estimate of x_1 with two different levels of noise. It is seen that with larger noise the estimate is poorer than with the case of less noise. In other words, the ratio of signal to noise is made smaller and the effect is seen to degrade the estimate. If the noise level is further increased, there is a point where the signal to ratio is such that the noise prevails and the filter cannot do accurate deconvolution. Figure 20 shows the observation that is processed by the filters to obtain the desired estimates. It is seen that the noise is high enough to be noticeable when $R = 10^{-3}$.

Result 2. Adaptive smoothing is also conducted with two different levels of measurement noise. As can be observed in Figure 21, the adaptive smoothing process for the larger level of measurement noise does not do as well as when the level of noise is smaller.

3.5 Case With Random Amplitude Poisson Input

For the model to be more realistic, it is necessary that the input be RAP distributed. The method of generation for this distribution has already been described in Section 3.2.2. The method of integration and the method of obtaining Gaussian measurement noise remain the same. The observations are at time intervals of .5 msecs.

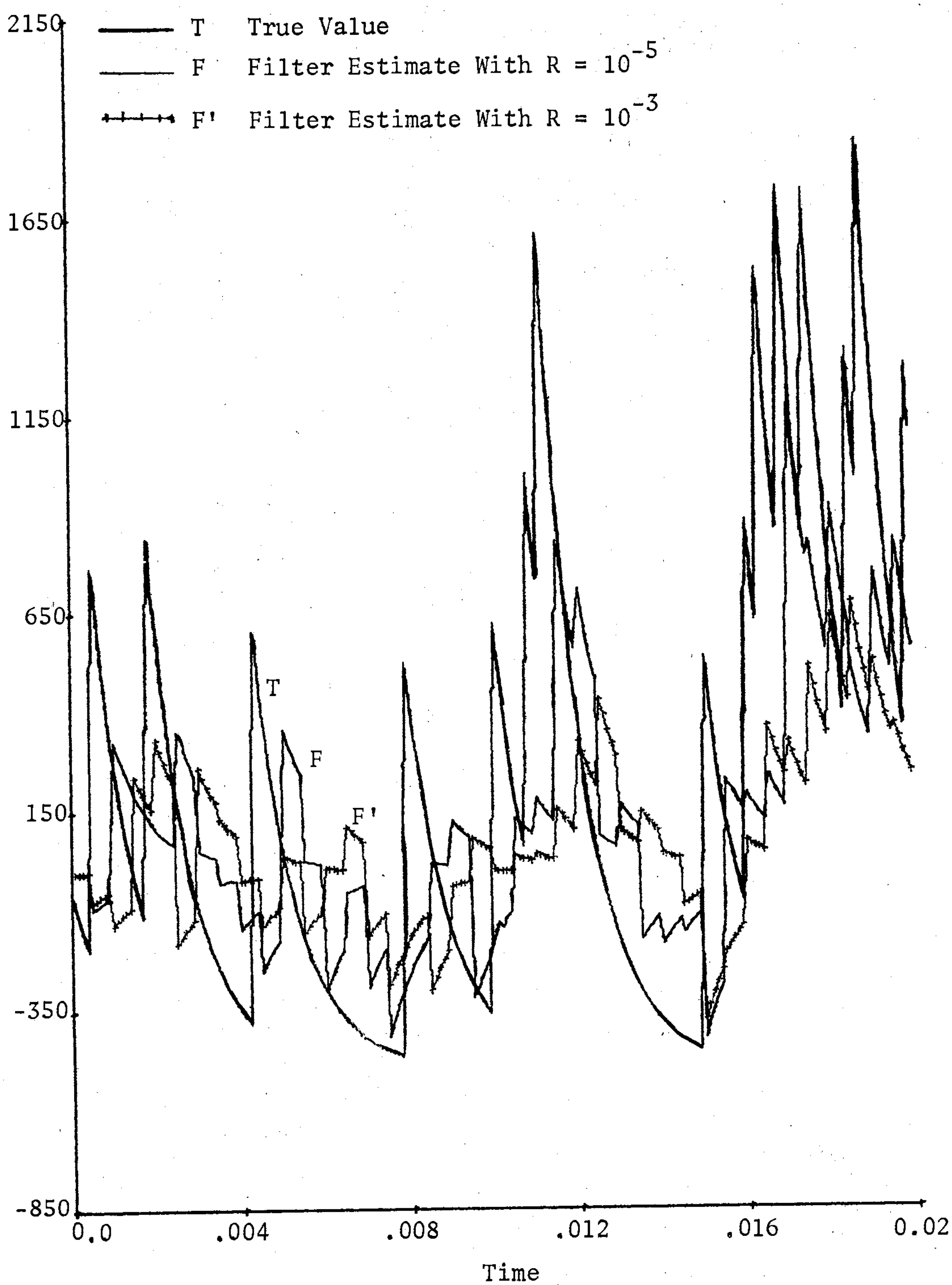

Figure 19. True and Filtered Estimate for EAP Input With two Different Measurement Noise

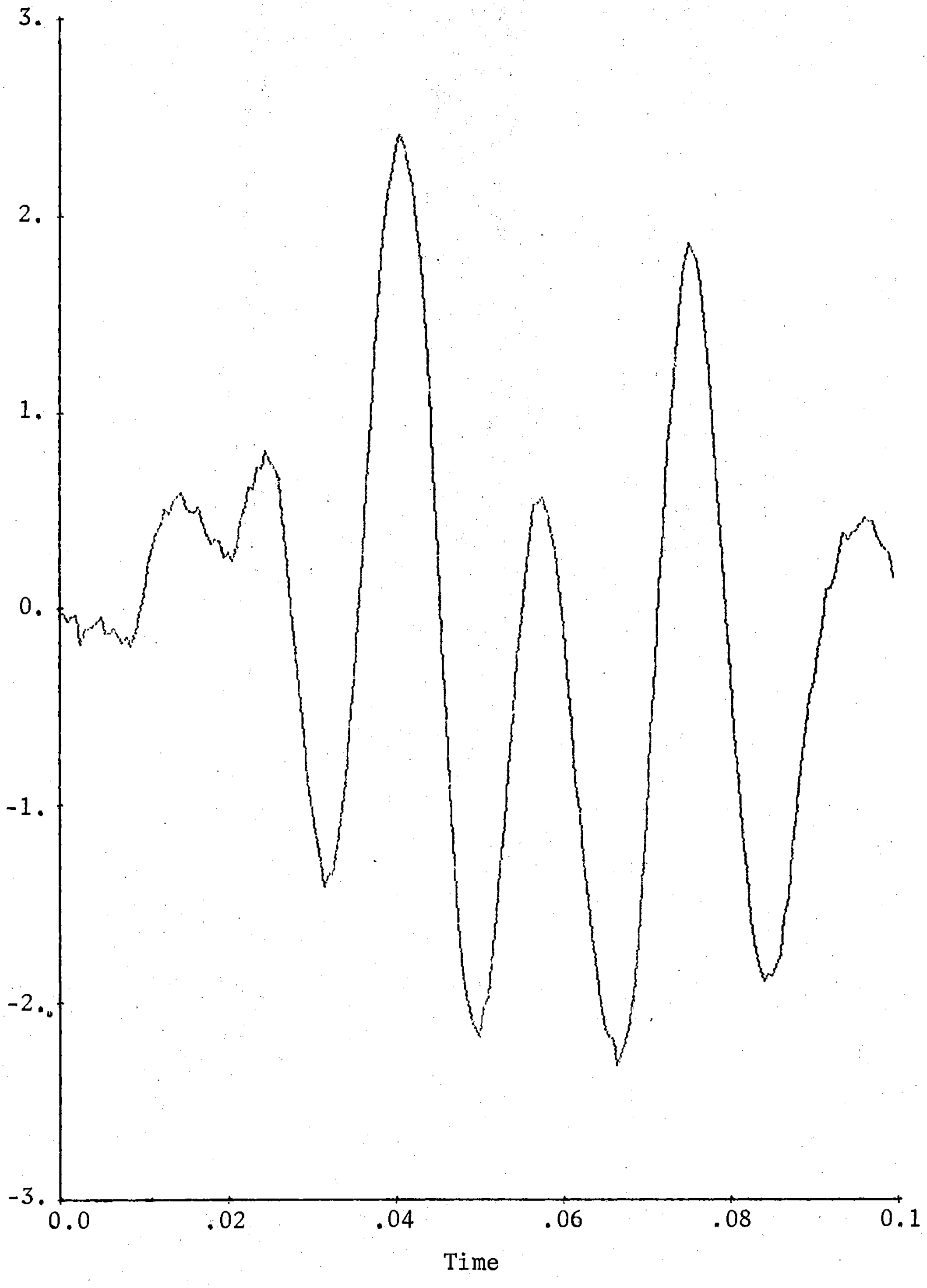

Figure 20. The Observation set for EAP Input and Measurement Noise Variance $R = 10^{-3}$

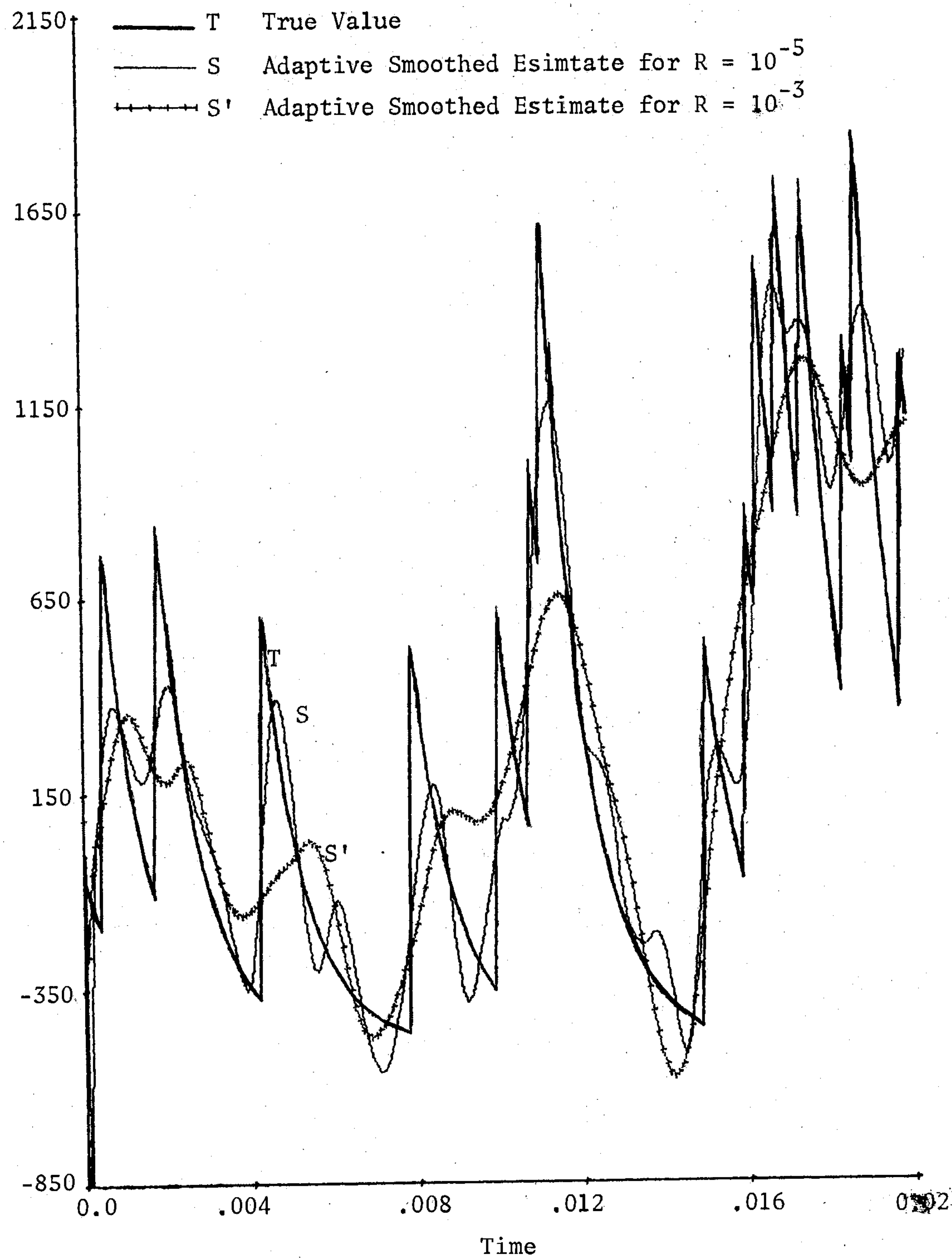

Figure 21. True and Smoothed Estimate for EAP Input With two Different Measurement Noise

3.5.1 Effects of Uncertainty in the Model

The same experiments are conducted on the modified Bayless and Brigham model as in Section 3.4 with the parameter 'a' having possible values of 150, 100, 50 and 10, while the other parameters are fixed as before. The same initial conditions are used and each model is given an equal a priori probability.

Result 1. A single stage run is made and the filtered results compared with the actual values in Figure 22. It is seen, however, that to achieve a satisfactory system identification, the number of observations must be greatly increased. This is shown in Figure 23. It is only in the last few hundred iterations of the 400 observations taken, that the correct model did achieve a respectable probability. In spite of the lack of reliability of system identification, the estimated values obtained are quite satisfactory. As seen in the previous section, the estimated values lag behind the true state x_1, due to the discrete observation.

Result 2. Adaptive smoothing is implemented next and the results plotted along with the true values of the state x_1 in Figure 24. Since the results are cluttered up, a magnified version of the first 200 iterations is plotted in Figure 26, showing the true, filtered and smoothed estimates. As in the previous experiments, it is observed that the smoothing results show a remarkable improvement in the estimates, removing much of the lag. Figure 25 shows a plot of the observation set from which these estimates have been made.

Result 3. For the case of an RAP distributed input, an experiment is conducted where the Q of the model, namely the frequency of occurrence of the impulses in the input are uncertain and the other model parameters

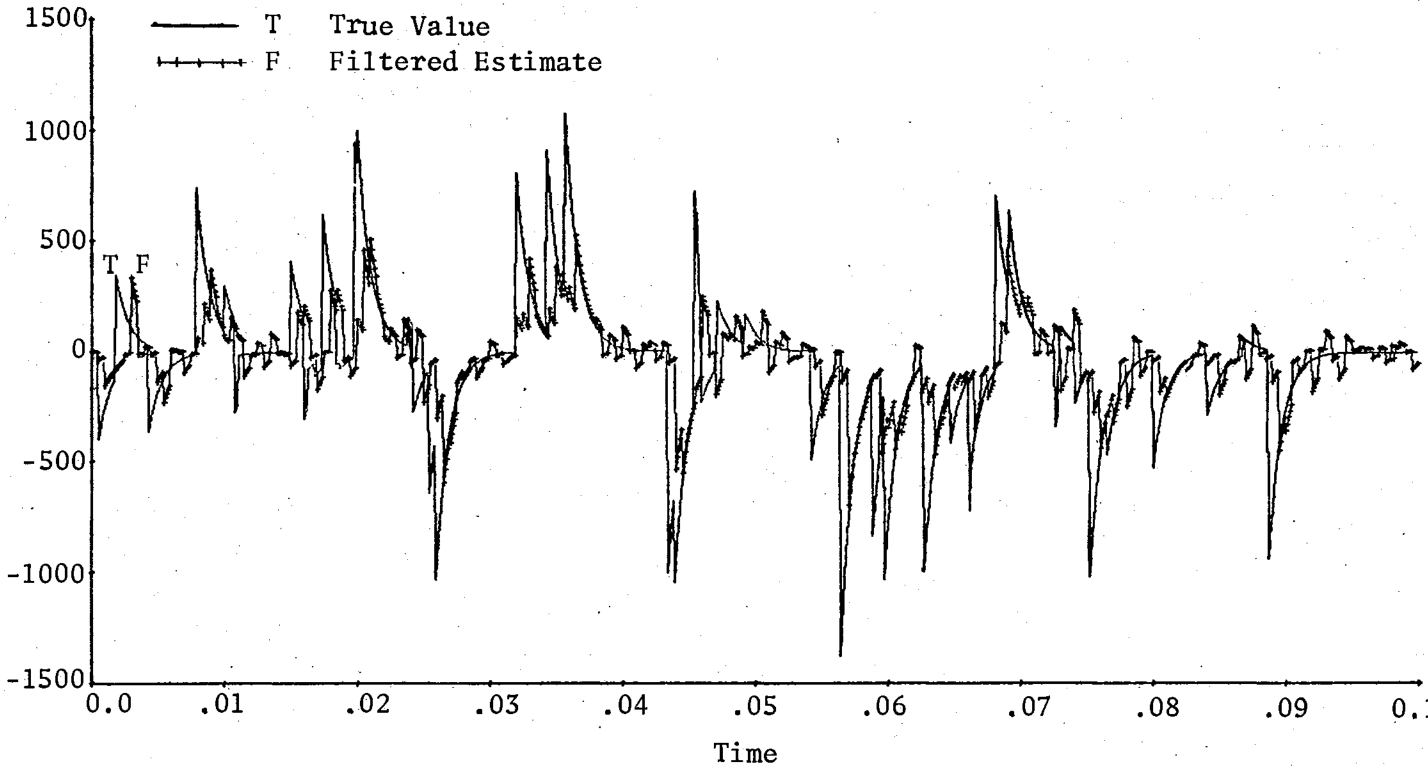

Figure 22. True and Filtered Estimate of State x_1 for RAP Input With 'a' Uncertain

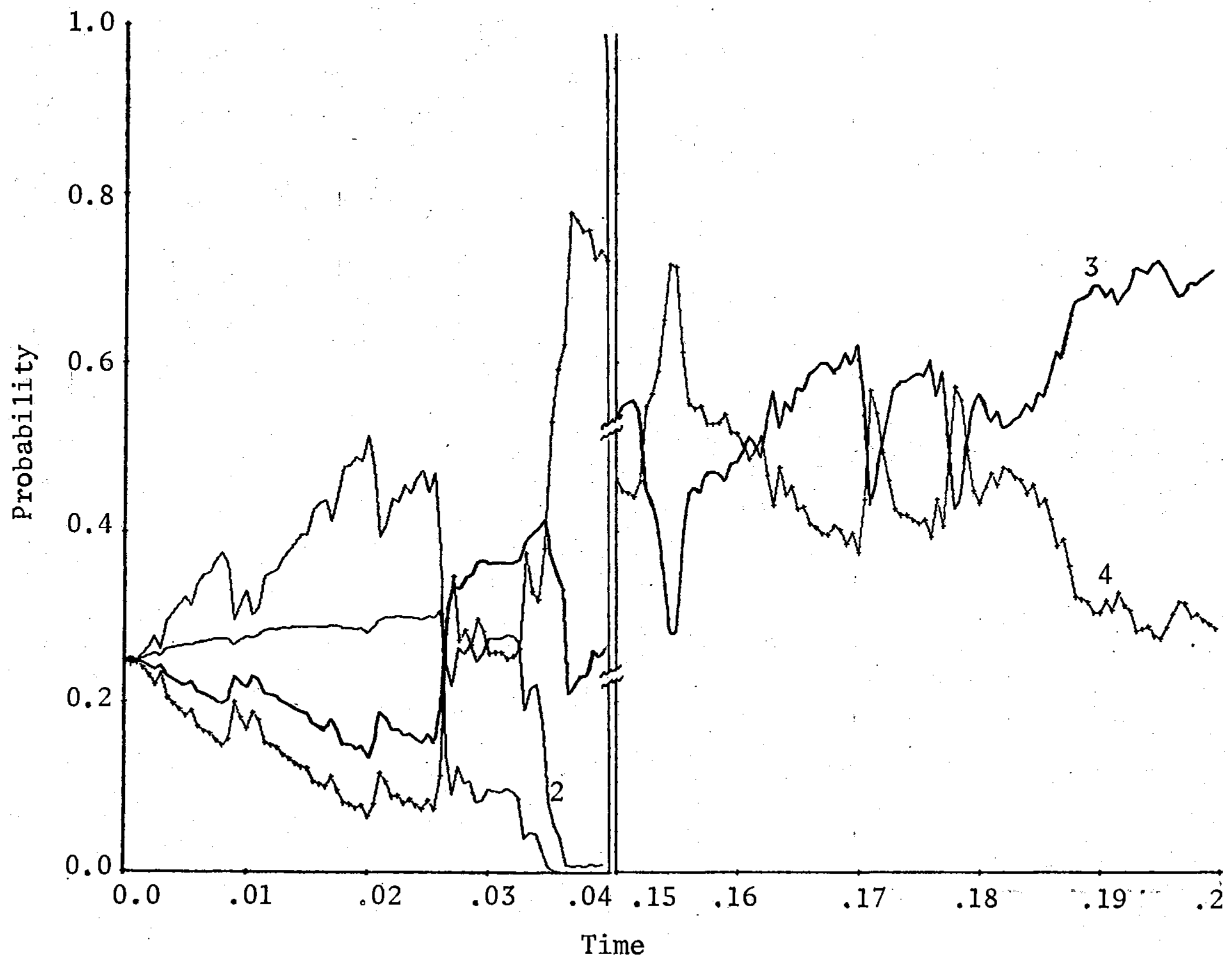

Figure 23. The a Posteriori Probability for the Four Candidate Models With 'a' Uncertain

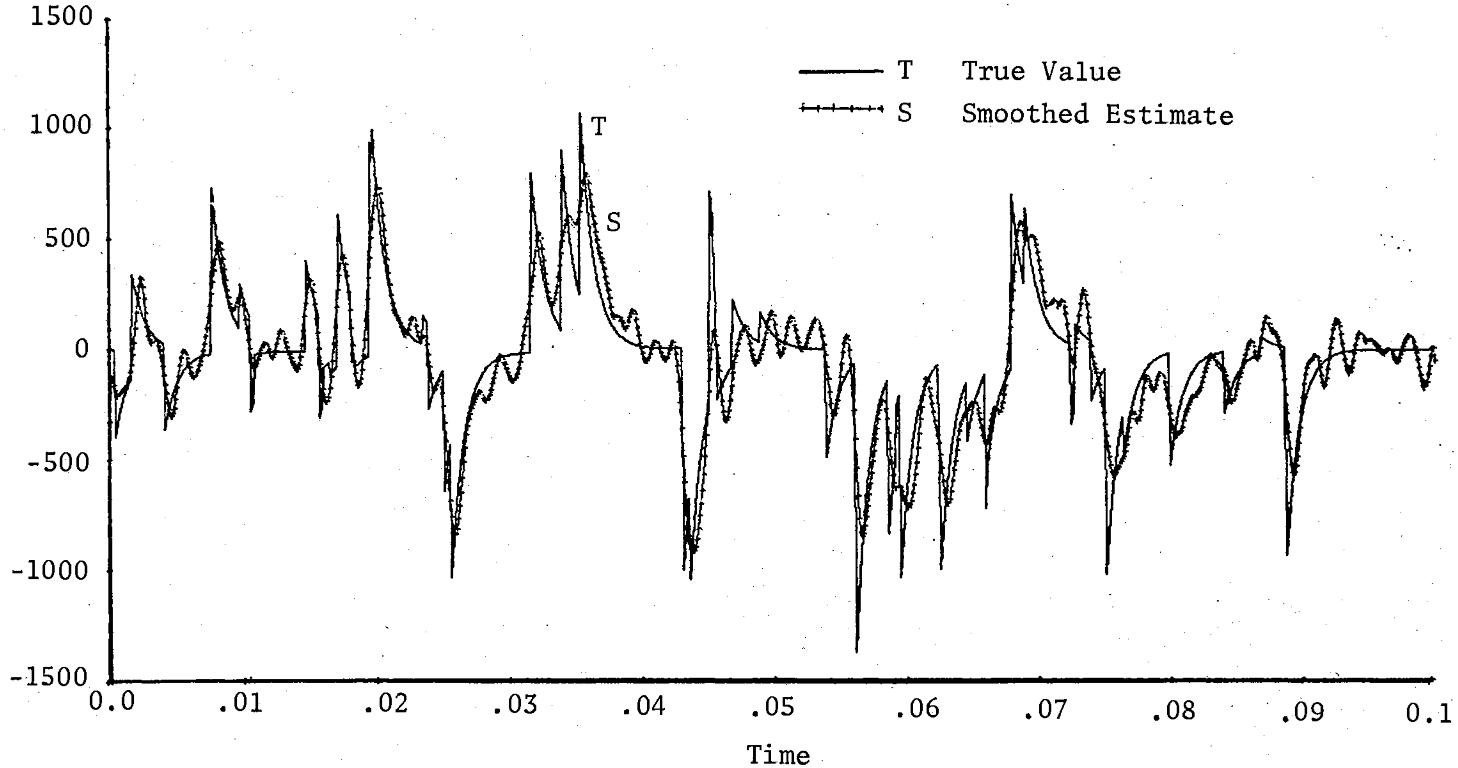

Figure 24. True and Smoothed Estimate of State x_1 for RAP Input With 'a' Uncertain

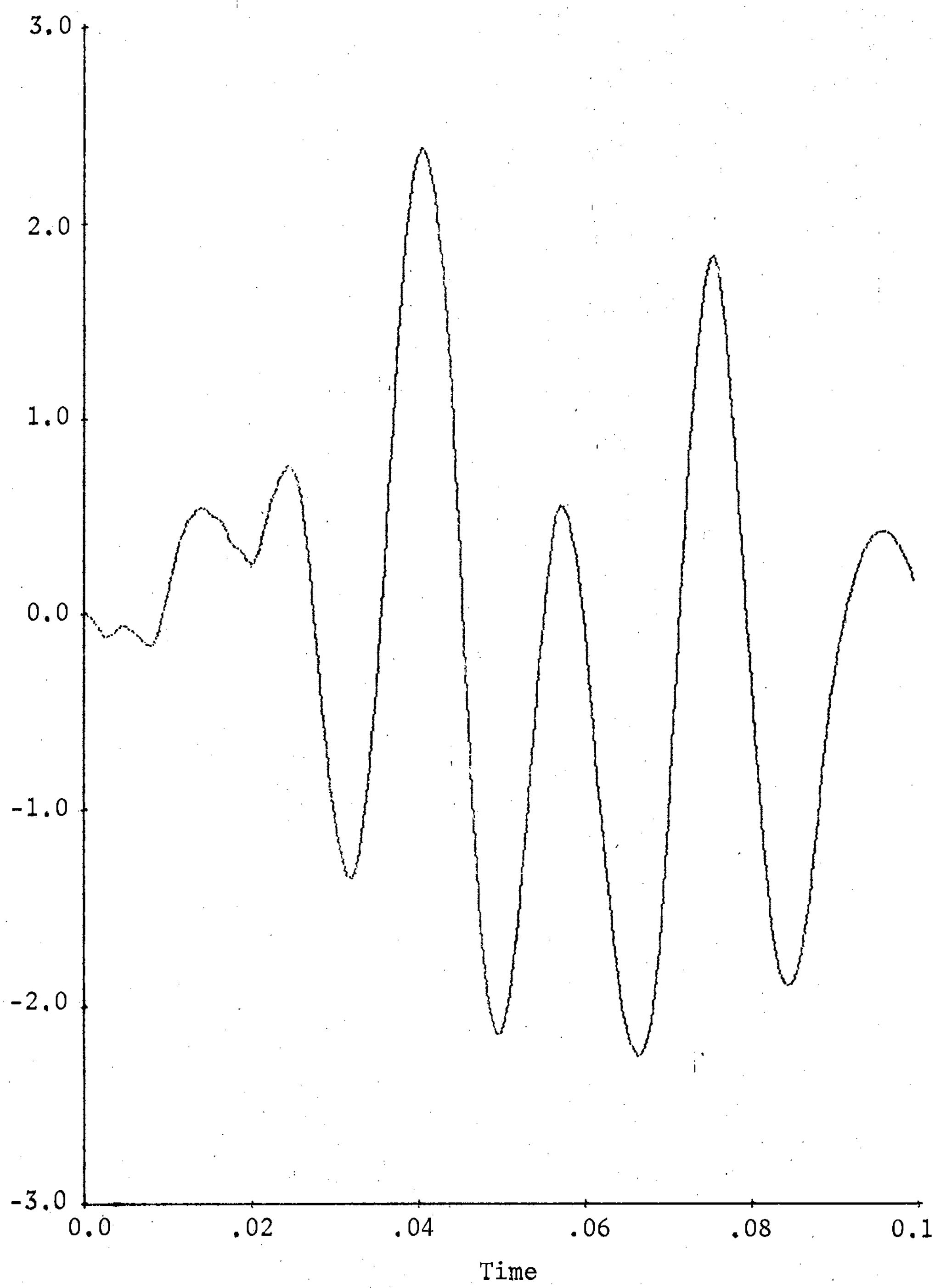

Figure 25. The Observation set for Random Amplitude Poisson Input

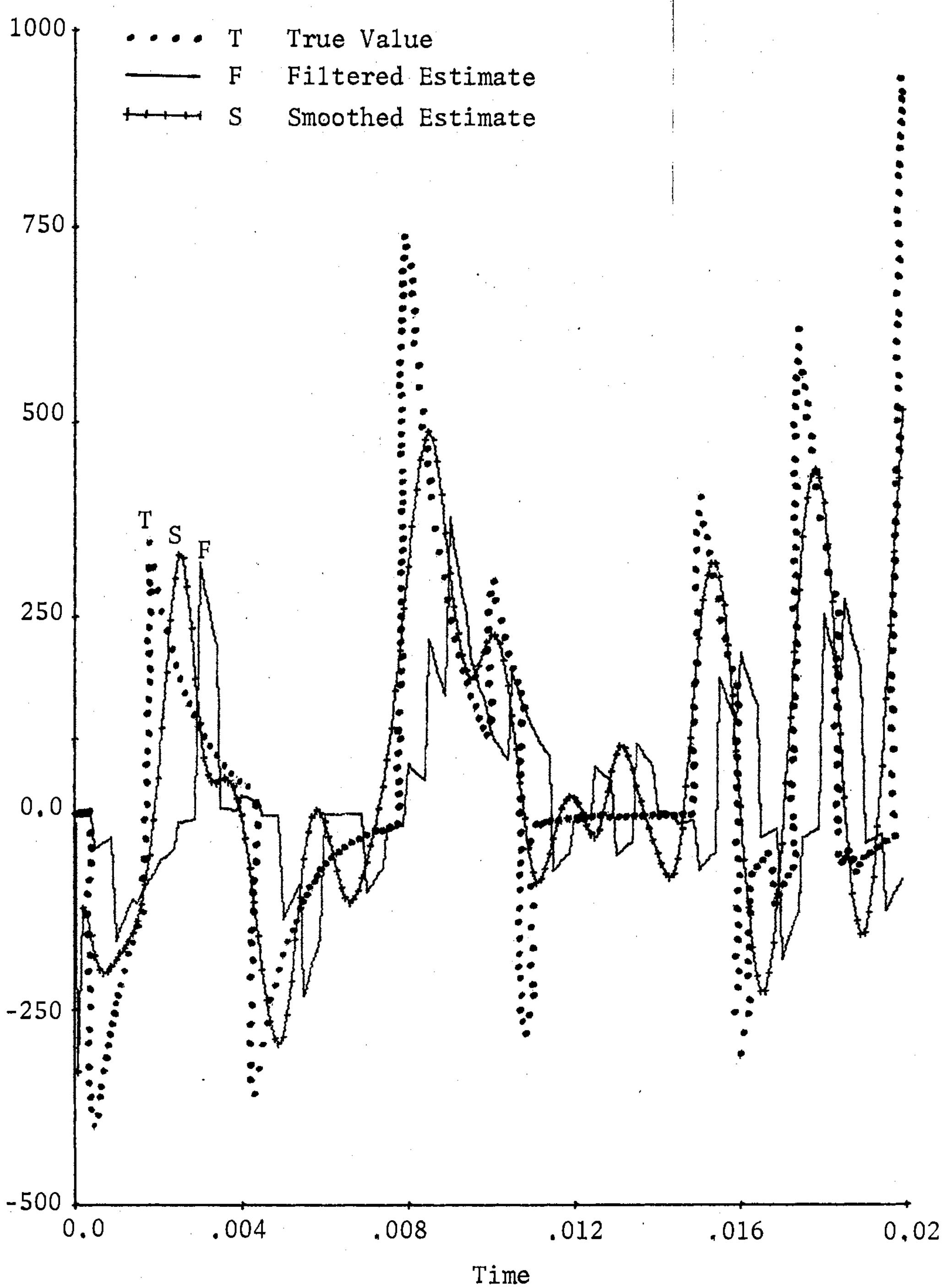

Figure 26. True, Filtered and Smoothed Estimate for RAP Input With 'a' Uncertain

are known at values specified by Bayless and Brigham. In other words, the four candidate models have values of Q as 900, 700, 500 and 100, the third model representing the true one. Figure 27 is a plot of the true value of x_1 when compared with the adaptive filtered estimate. There is a definite lag in the estimated values as compared with the true values. The system identification capability is seen in the plot shown in Figure 28.

Result 4. Adaptive smoothing when applied to the above model with uncertain Q gives results as plotted in Figure 29. A closer observation of the effect of adaptive smoothing as compared with filtering can be seen in Figure 30 for the first 200 iterations. The process of adaptive smoothing reduces the lag in the filtered results to a great extent.

Result 5. To compare Kalman filtering under certainty with adaptive filtering, an experiment is conducted with the model being fixed and the Q of the input being uncertain as described above. In Figure 31, plots of the true value, the Kalman estimate, and the adaptive filtered estimate of x_1 are shown. It is seen that the difference in the estimates is minimal and hence model uncertainty does not significantly degrade the results. This seems important because even though the model is uncertain, the estimation scheme does nearly as well as in the case where the model is completely known.

3.5.2 Effects of Measurement Noise

It is important to see the effect of measurement noise on adaptive filtering and smoothing given a RAP input. As in Section 3.4.2, the two levels of measurement noise are chosen to be $R = 10^{-3}$ and $R = 10^{-5}$ where R is the variance parameter of noise. The experiment is conducted with

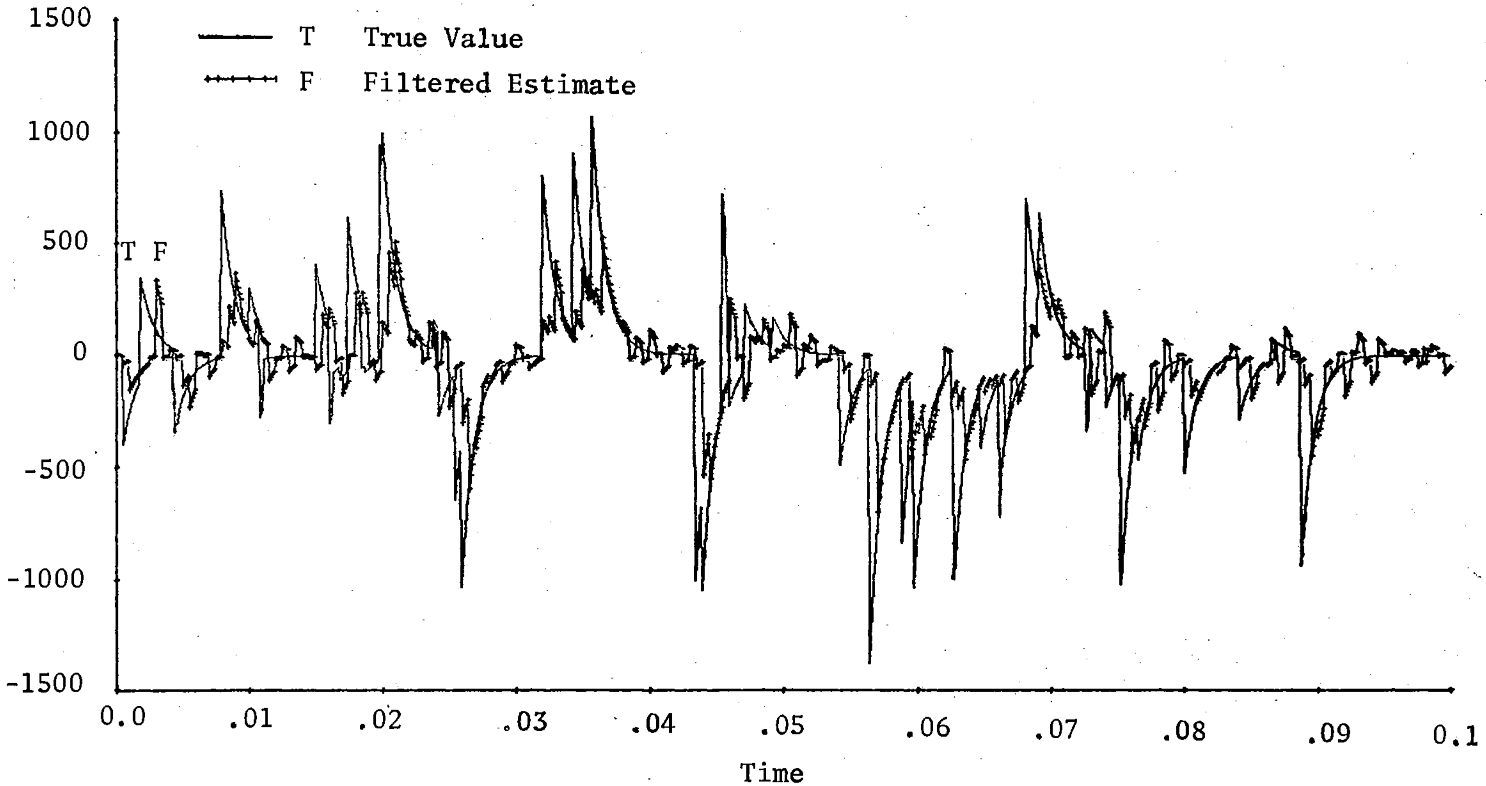

Figure 27. True and Filtered Estimate of State x_1 for RAP Input With 'Q' Uncertain

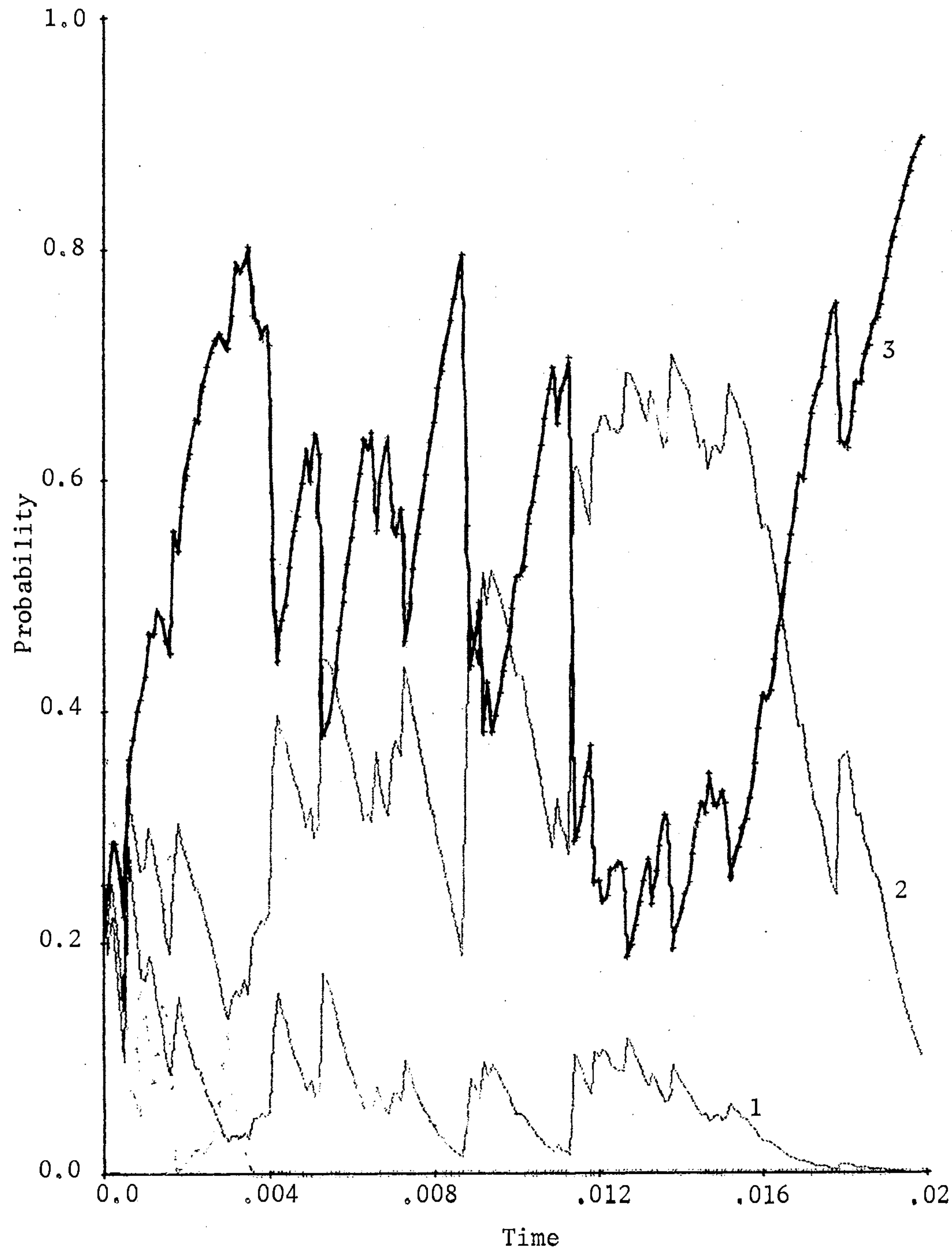

Figure 28. The a Posteriori Probability for the Four Candidate Models With 'Q' Uncertain

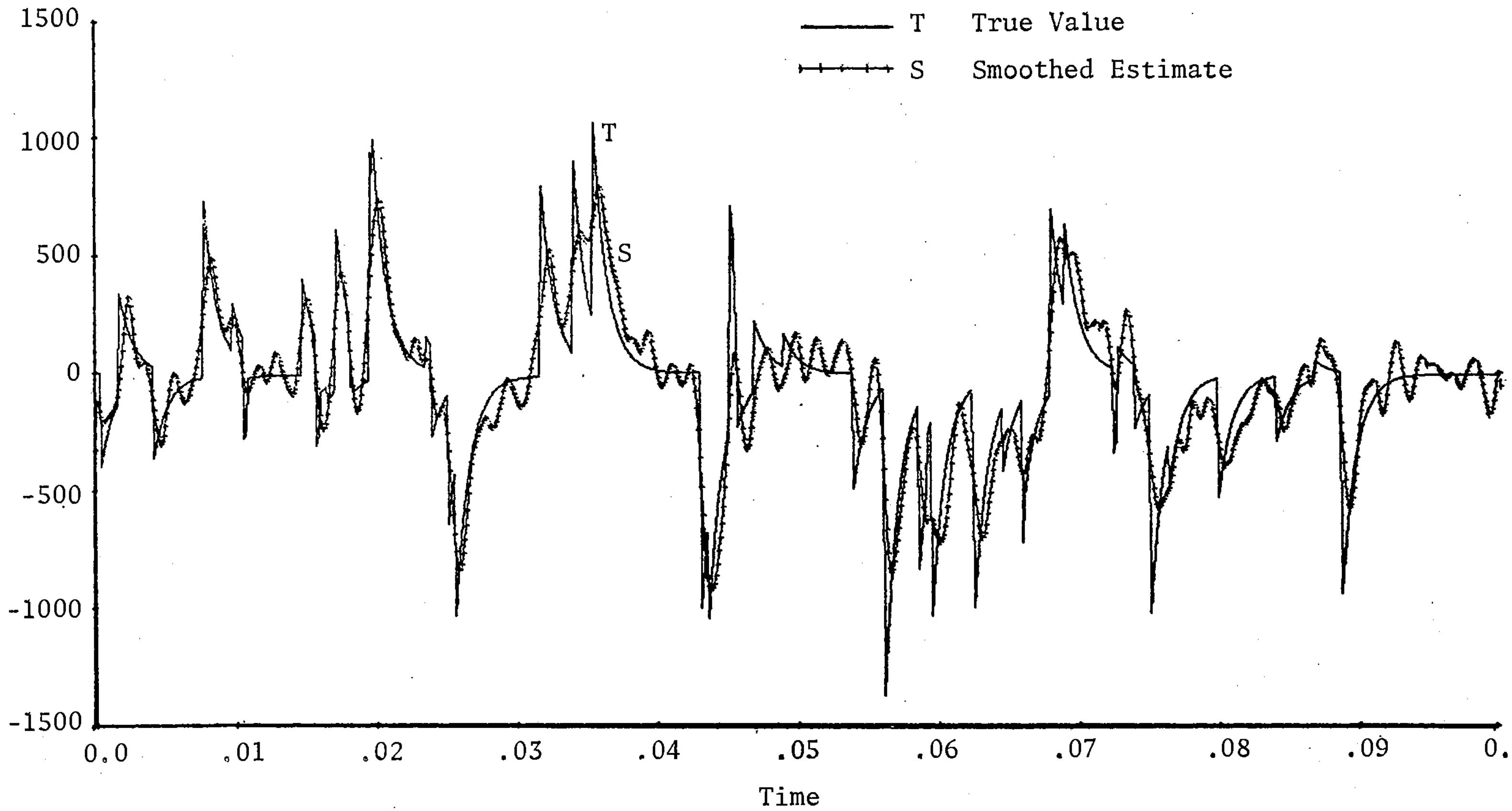

Figure 29. True and Smoothed Estimate of State x_1 for RAP Input With 'Q' Uncertain

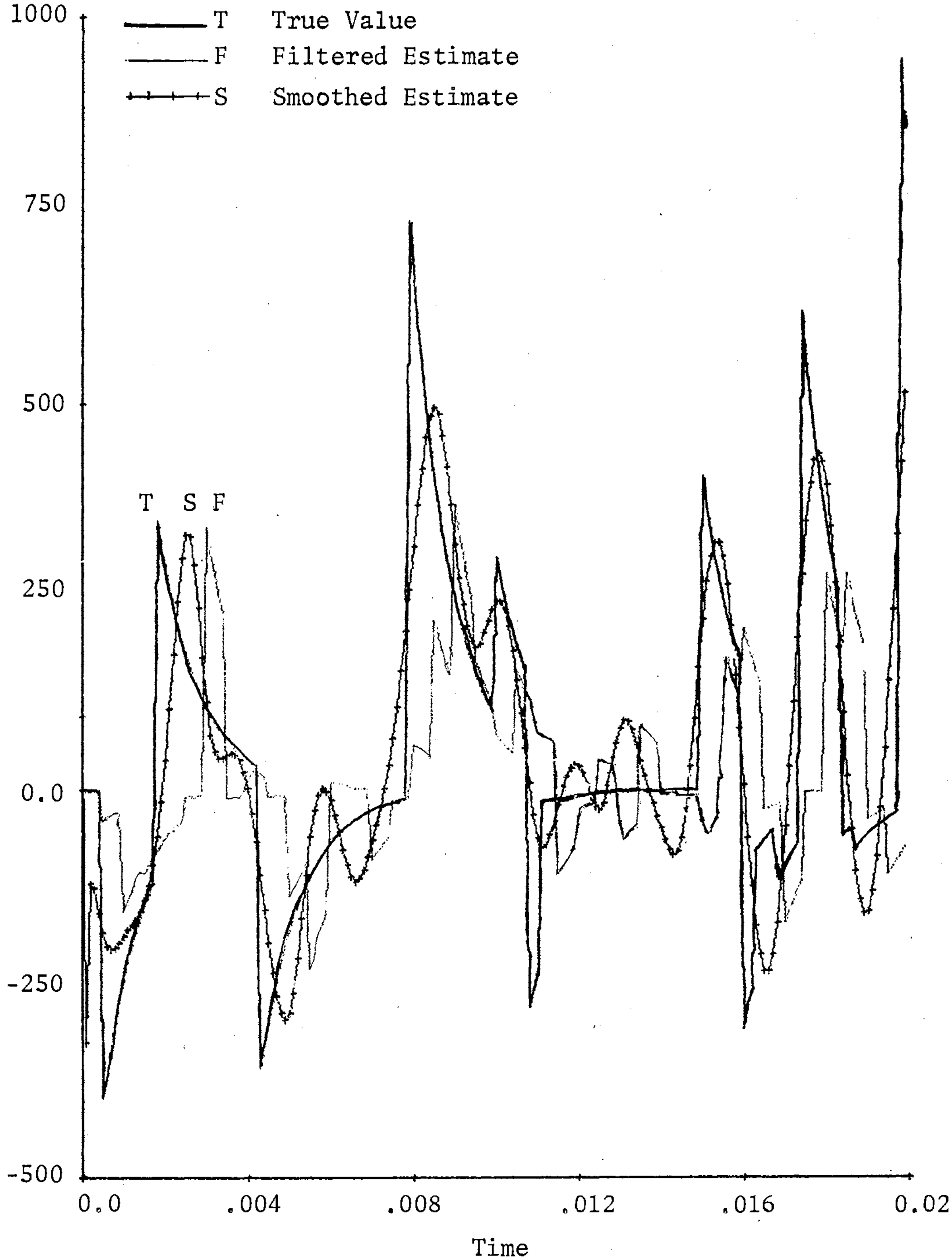

Figure 30. True, Filtered and Smoothed Estimate for RAP Input With 'Q' Uncertain

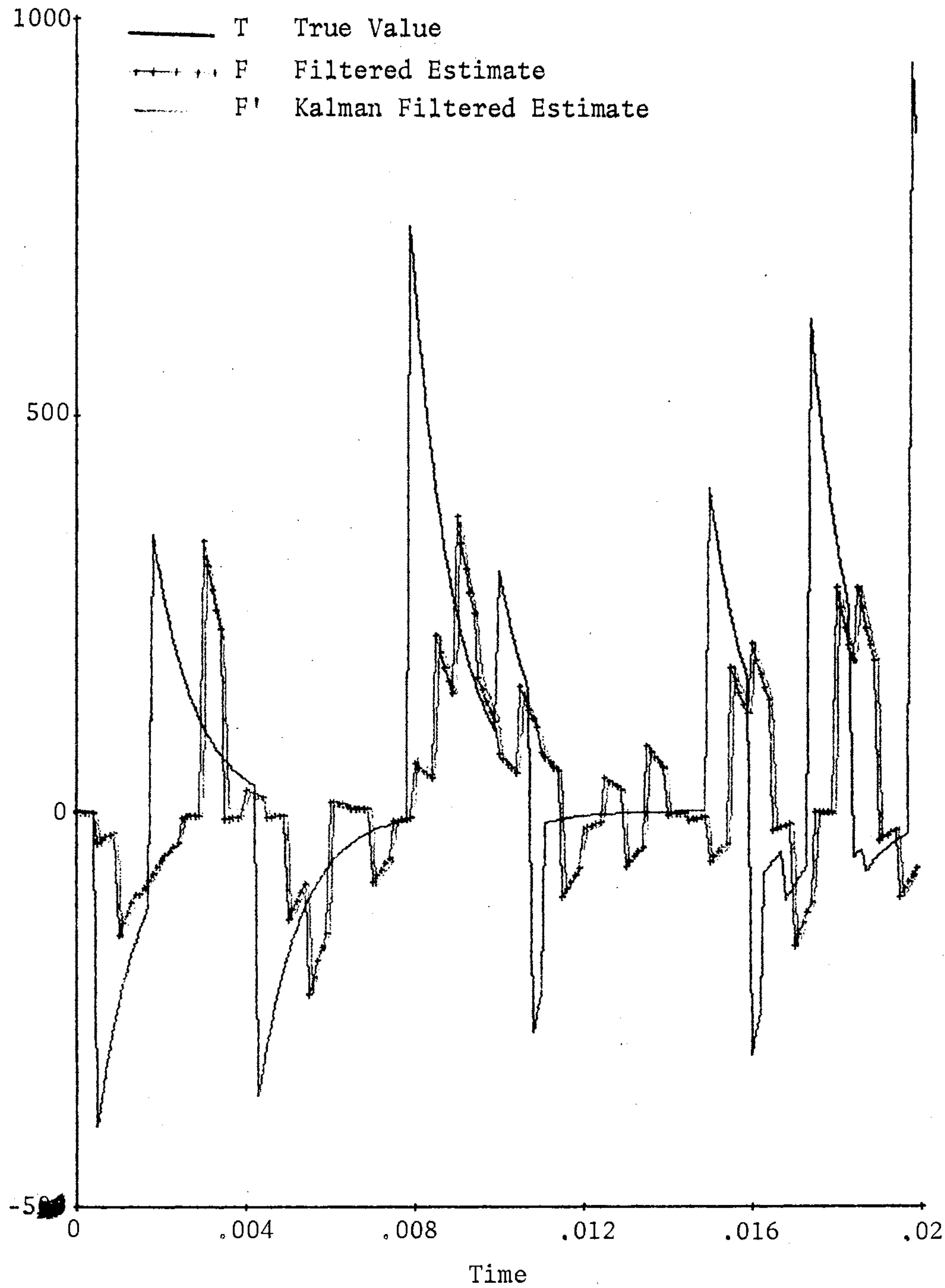

Figure 31. True, Adaptive Filtered and Kalman Filtered Estimate for RAP Input With 'Q' Uncertain

the Q of the plant noise having four possible values: 900, 700, 500 and 100. The other parameters are fixed, and known with certainty.

Result 1. A single run for the two levels of measurement noise is made and adaptive filtering applied. In Figure 32 the effects of the larger measurement noise are seen. As the measurement noise is increased, a certain amount of degradation sets in and the estimate is worse than with the case of less measurement noise. It should be noted that if the noise level is raised excessively, the signal to noise ratio may be so poor that the operation of filtering is useless.

Result 2. Adaptive smoothing for the two levels of measurement noise is conducted and the results shown in Figure 33. As seen earlier, the effect of the higher measurement noise level is to cause a deterioration in the estimate when compared with the small measurement noise having $R = 10^{-5}$. However, for this level of measurement noise, estimation of the state x_1 is still satisfactory, though the magnitude of the estimates has dropped due to smaller Kalman gains K as a result of the measurement noise increase.

3.6 Summary

A geophysical model has been simulated for the case of equal amplitude poisson inputs and RAP inputs. It has been seen that though the theory has been developed for Gaussian inputs, the estimation scheme worked well for the above inputs. Deconvolution was achieved for the case of model uncertainty and the estimation was not hampered due to the fact that the parameters of the model were unknown. Adaptive smoothing was very effective in improving the estimate of the state. System identification, however, was not very reliable though adequate estimation

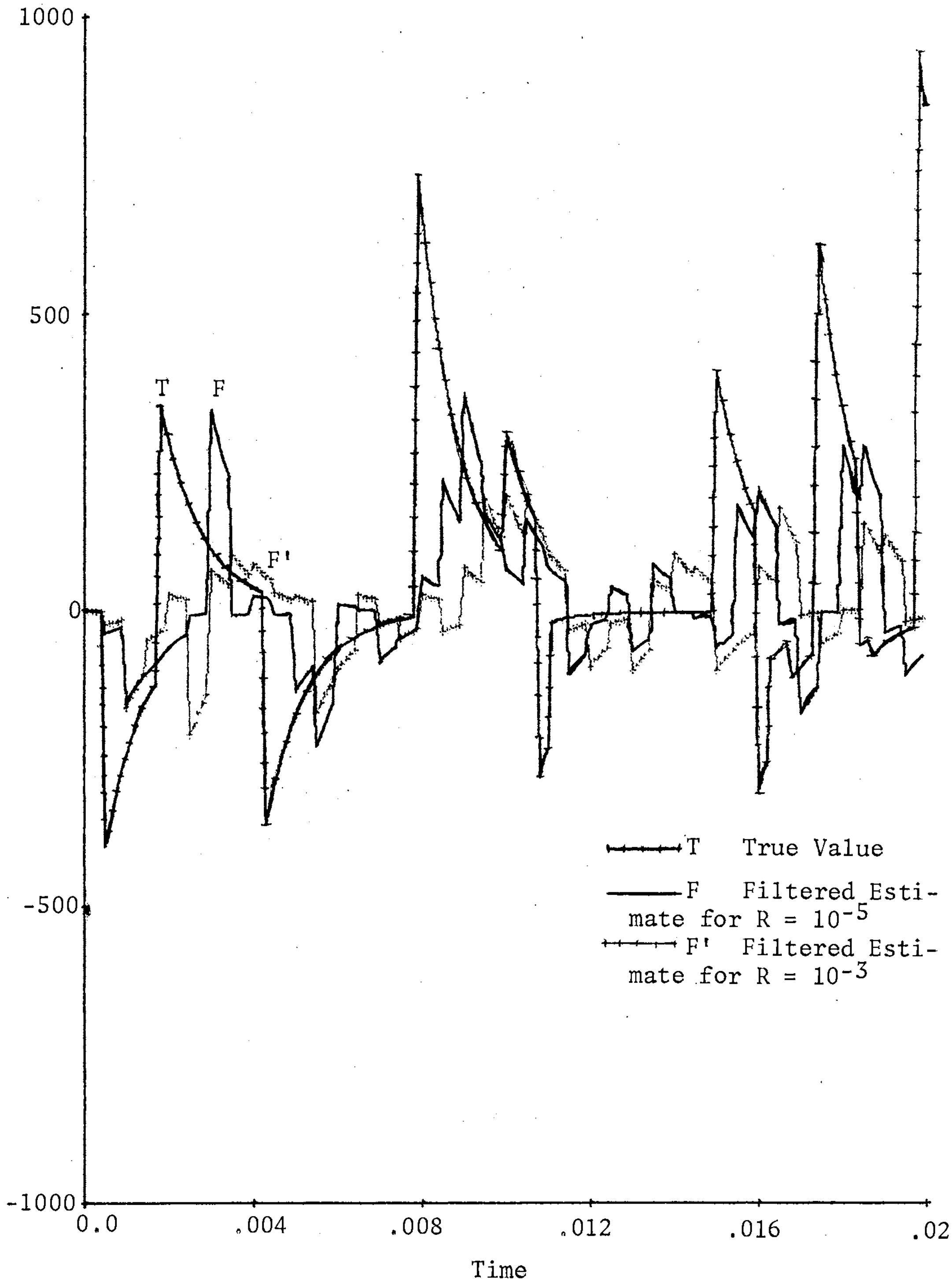

Figure 32. True and Filtered Estimate for RAP Input With two Different Measurement Noise

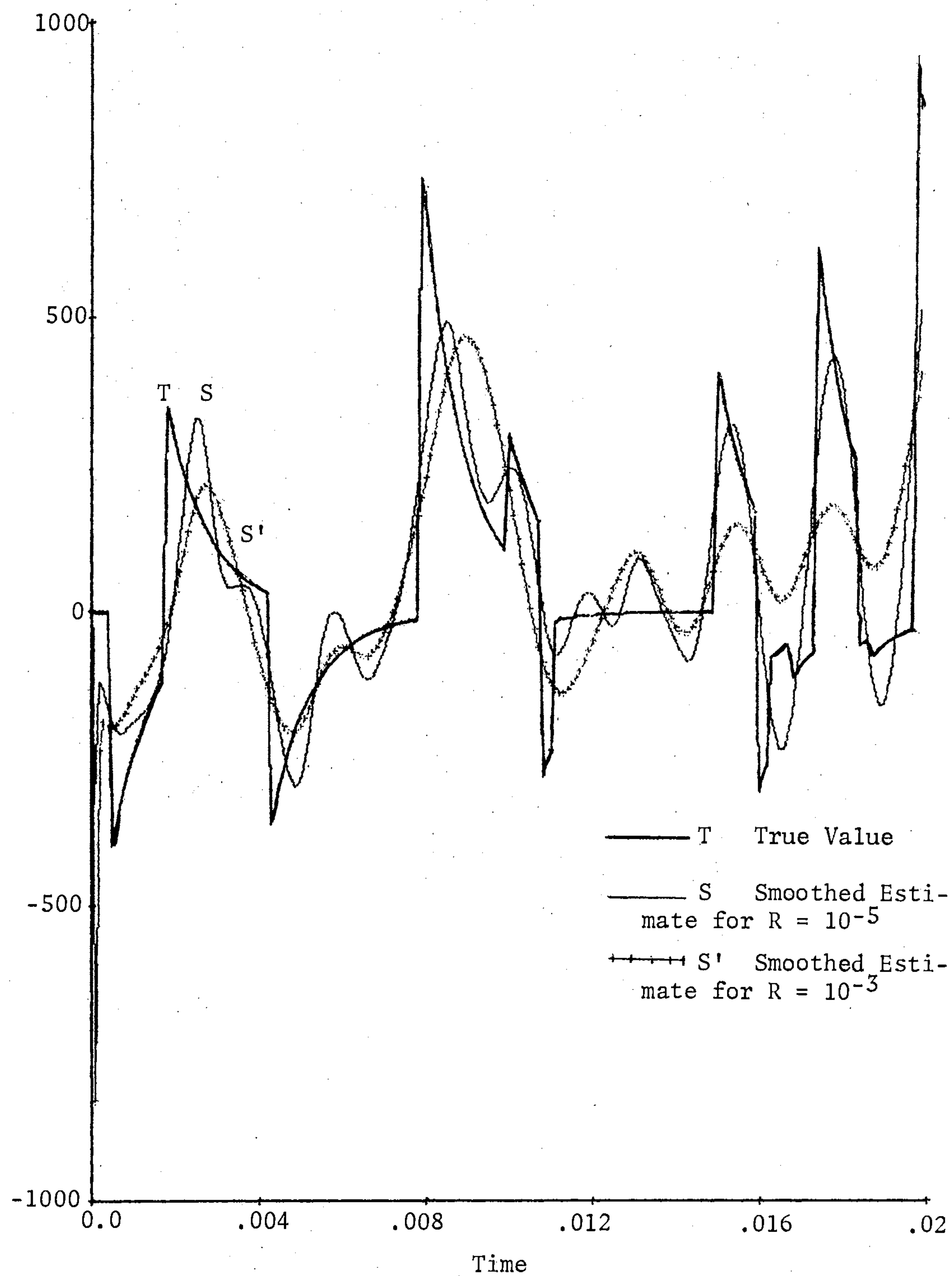

Figure 33. True and Smoothed Estimate for RAP Input With two Different Measurement Noise

was always achieved. Even for higher levels of measurement noise, deconvolution seemed to work, although the signal to noise ratio had to be reasonable so as to make the filtering effective. It is also possible to obtain from the same algorithm described herein, the predictive operator x_3, or the basic wavelet shape so that its inverse operator can be found out and the impulse response obtained.

It may so happen that the correct model is not one of the candidate models because it is unknown. The next chapter discusses the effects this has on the estimation scheme proposed.

CHAPTER IV

EVALUATION

4.1 Introduction

The idea of adaptive filtering as described in the previous chapters uses the underlying assumption that the true model must be one of the candidate models. Ordinarily in practice, the parameters in the model are unknown and one has to rely on judgement in selecting the candidates. It must be expected that the parameter set chosen for the candidate models may not include the true one. It is interesting then, to observe the effects this will have on the system. The motivating questions for the following experiments are: (1) Will the estimation scheme identify the model closest to the true one? (2) If it does, will there be any degradation in the estimate?

4.2 Effects of Improper Modeling

The basic Bayless and Brigham model has been taken and two candidate models chosen with parameters 'a' as 60. and 30. The true value of 'a' in the model is known to be 50. Each is assigned an a priori probability of 0.5 and the same initial conditions are specified as in the previous experiments.

The candidate model equations can be written as

$$\theta_1 : \dot{\underline{X}} = \begin{bmatrix} -1000 & 0 & 0 \\ -100\pi & 0 & -(60^2+100^2\pi^2) \\ 0 & 1 & -120 \end{bmatrix} \underline{X} + \begin{bmatrix} 1000 \\ 0 \\ 0 \end{bmatrix} w \qquad (4.1)$$

$$\theta_2 : \dot{\underline{X}} = \begin{bmatrix} -1000 & 0 & 0 \\ -100\pi & 0 & -(30^2+100^2\pi^2) \\ 0 & 1 & -60 \end{bmatrix} \underline{X} + \begin{bmatrix} 1000 \\ 0 \\ 0 \end{bmatrix} w \qquad (4.2)$$

The observation model is

$$Z = [0 \quad 0 \quad 1]\ \underline{X} + v \qquad (4.3)$$

where v is a zero mean white noise with variance 10^{-5}.

Result 1. A single run is made with the above two candidate models assumed. Figure 34 shows the identification capability of the two models. It is seen that as more data are obtained the model with the unknown parameter closest to the true value approaches a probability of one. At the end of 2000 iterations the model with 'a' as 60. has a probability of .83 while 'a' as 30. has a probability of .17.

Result 2. A plot of the true, filtered and smoothed estimates of the state x_1 has been shown in Figure 35 for 200 iterations. As evidenced, smoothing gives a more accurate estimate and inspite of improper modeling, deconvolution is achieved. This is quite significant because even though the model may not have exact parameters, estimation can still succeed.

When adaptive filtering is applied and the correct model is not included as a candidate model, then the model that most closely

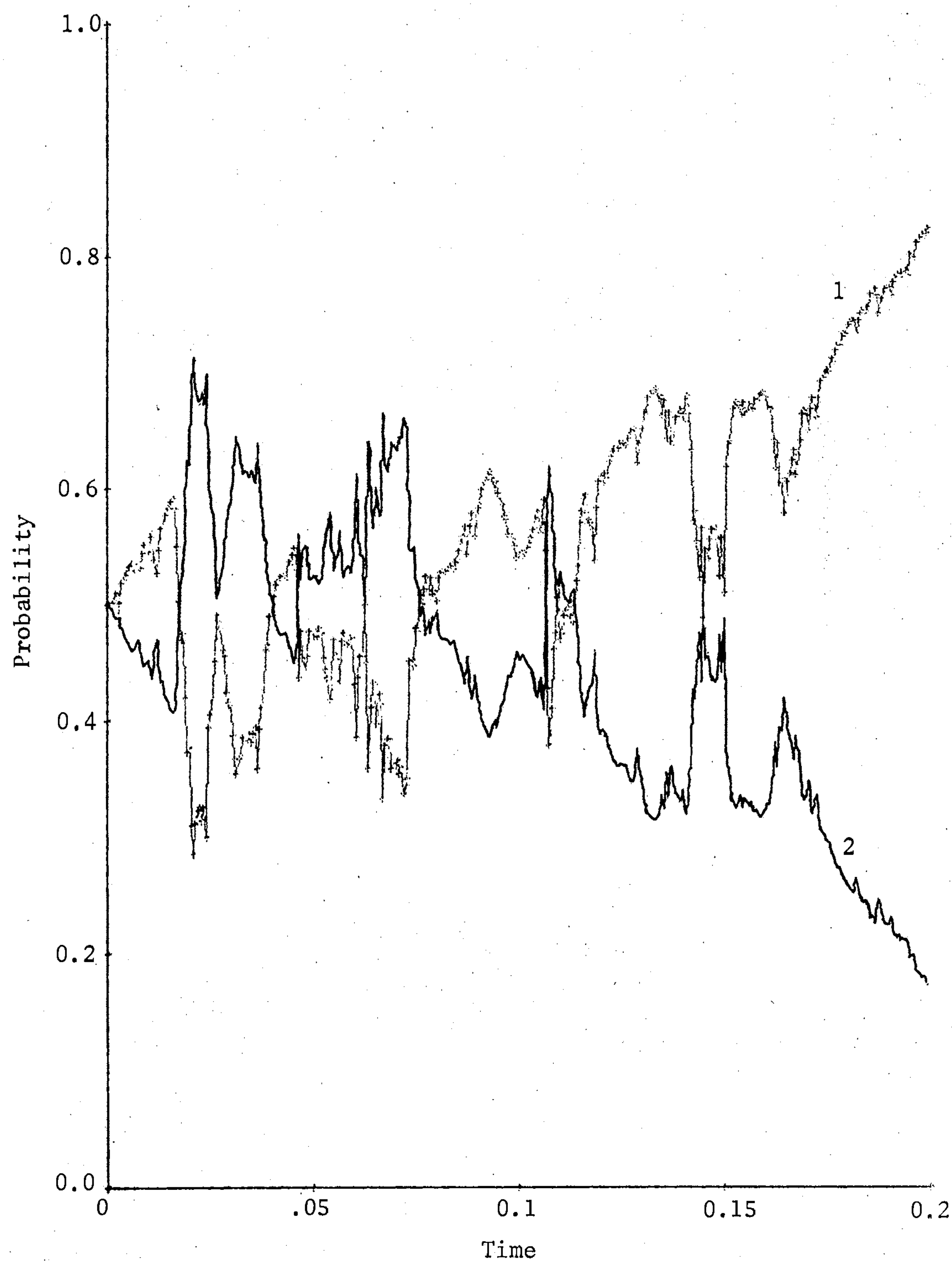

Figure 34. The a Posteriori Probabilities for the two Incorrect Models With EAP Input

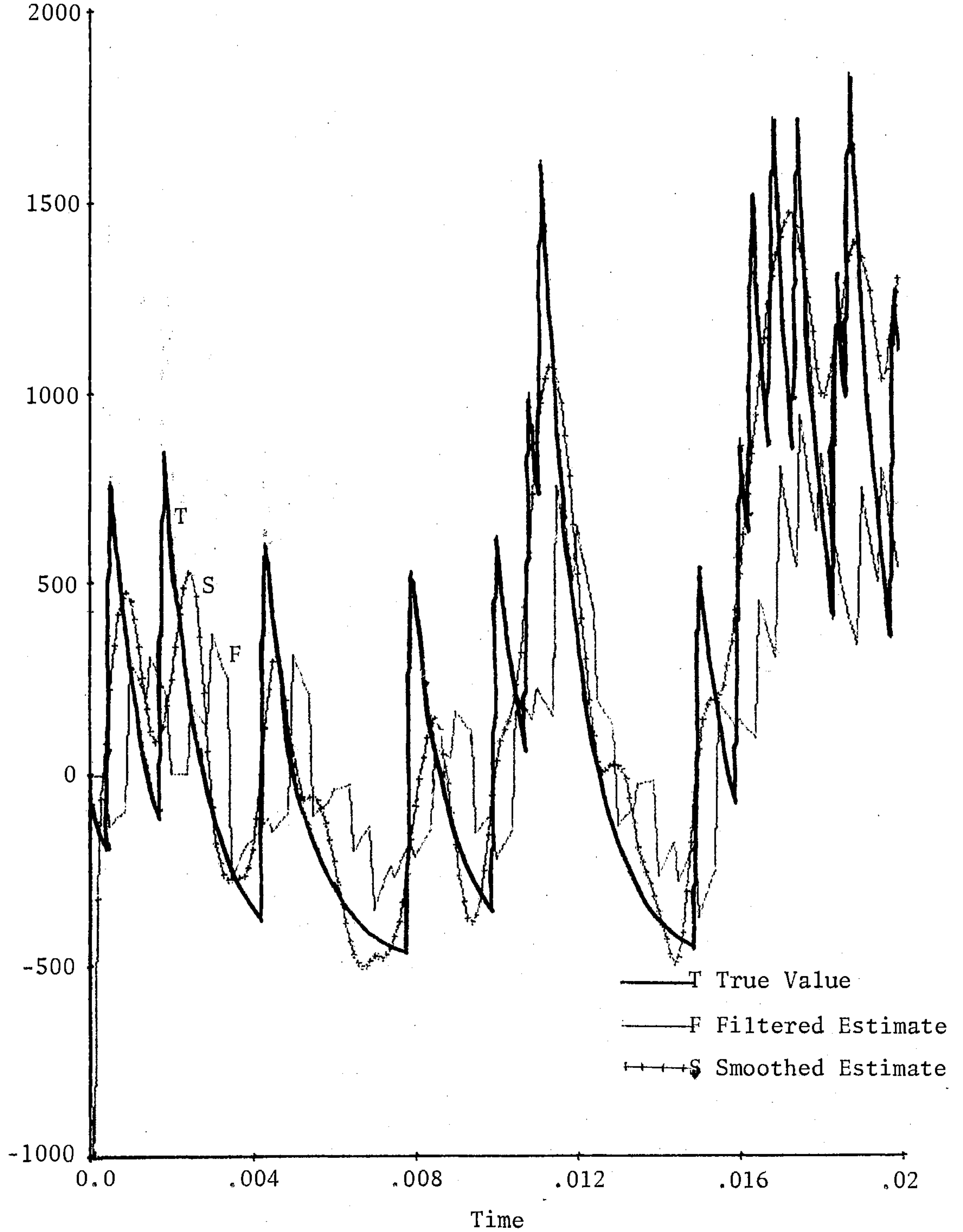

Figure 35. True, Filtered and Smoothed Estimate of State x_1 for the two Incorrect Models With EAP Input

represents the true model hopefully achieves the highest a posteriori probability. In effect, it is possible to narrow down on the unknown parameters from a number of possible values to those centered around the parameters of the model with the highest a posteriori probability. This is discussed in the next section.

4.3 Iterative Use

In conducting the last experiment, it has been seen that the candidate model with 'a' as 60. had a higher a posteriori probability than the candidate model with 'a' as 30.0. Hence, it can be inferred that the correct calue of 'a' is nearer to a value of 60.0. One can repeat the process with candidate models centered about 'a' as 60.0.

In the following experiment, two values of 'a' are selected. These are 50. and 70., respectively.

The candidate model equations are

$$\theta_1 : \dot{\underline{X}} = \begin{bmatrix} -1000 & 0 & 0 \\ -100\pi & 0 & -(50^2+100^2\pi^2) \\ 0 & 1 & -100 \end{bmatrix} \underline{X} + \begin{bmatrix} 1000 \\ 0 \\ 0 \end{bmatrix} w \qquad (4.4)$$

$$\theta_2 : \dot{\underline{X}} = \begin{bmatrix} -1000 & 0 & 0 \\ -100\pi & 0 & -(70^2+100^2\pi^2) \\ 0 & 1 & -140 \end{bmatrix} \underline{X} + \begin{bmatrix} 1000 \\ 0 \\ 0 \end{bmatrix} w \qquad (4.5)$$

with the same observation model and initial conditions as before.

Result 1. For a single run, the system identification capability has been observed in Figure 36. As can be seen, the candidate model with 'a' as 50. reached a higher probability when the observation set was curtailed and the experiment stopped. On running the above two models for a longer period, it was found that the a posteriori probability of the correct model dropped off to a very low value. However, on observing the estimation of the state and comparing with the true value, it was found that good estimation was still achieved. Hence one may conclude that the system identification cannot be relied upon though the deconvolution capability seems reliable.

It has been proved, that for the case of Gaussian inputs, the true model will eventually converge to an a posteriori probability of one given enough data (18). However, for the geophysical process being modeled, the inputs are a sequence of random poisson distributed impulses and convergence has not been proved for this case. One reason for the convergence not being obtained in this experiment may be attributed to a certain sequence of noise that was implemented to generate the observations. The fact that the inputs are non-Gaussian seems to be a possible reason.

To show the reliability of the filter, a Monte Carlo analysis should be conducted for different noise sequences. Then a statistical evaluation for these various noisy observations could be made. However, due to the inherent expense of Monte Carlo simulation, these experiments were not conducted.

4.4 Summary

It has been observed that even with some of the parameters in the model unknown, by a method of reduction and iteration, the parameter

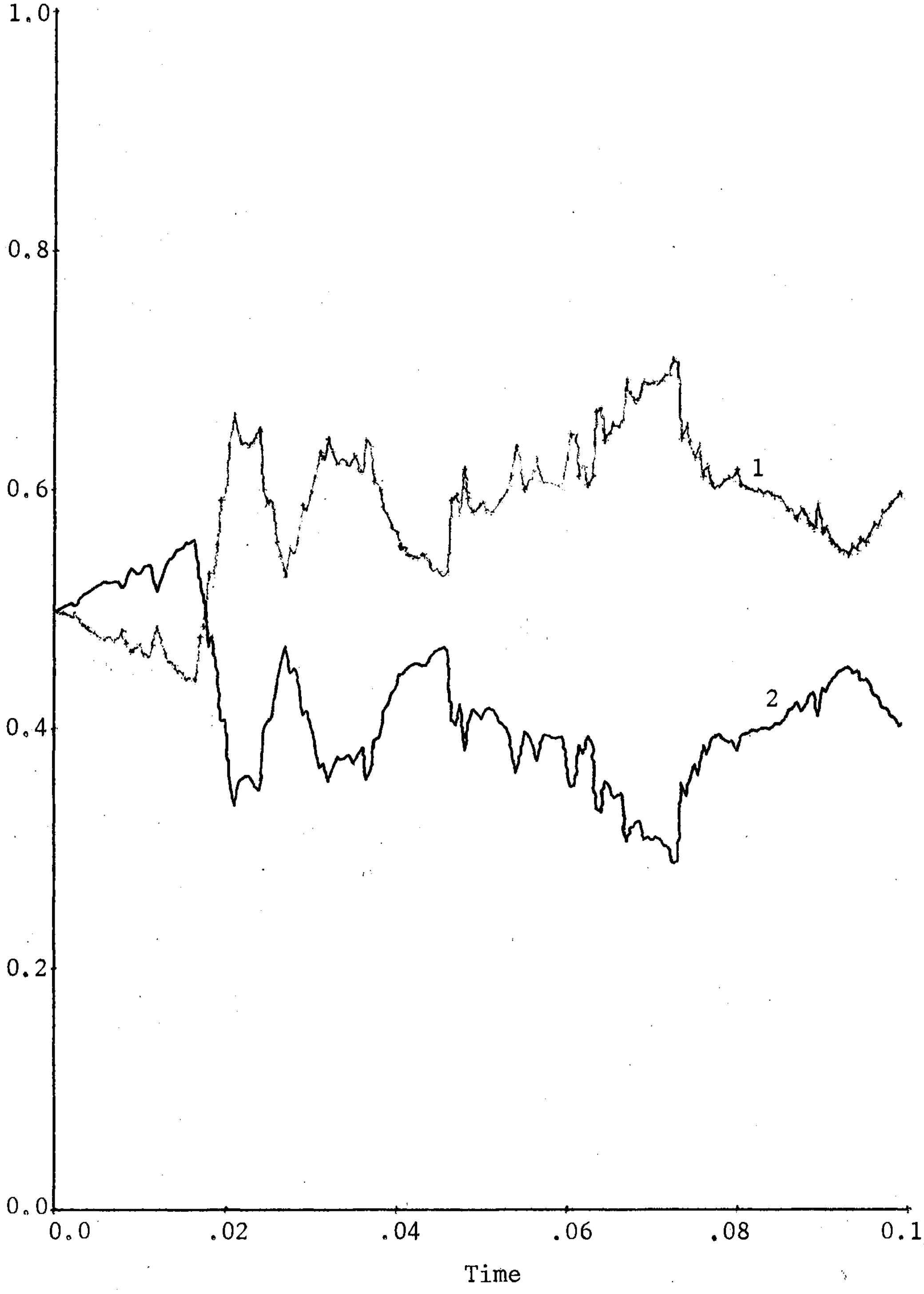

Figure 36. The a Posteriori Probabilities for the two Models With EAP Input Having 'a' as 50. and 70.

values can be identified, though not reliably. Estimation of the desired state is achieved, and the algorithm seems a promising way to accomplish deconvolution.

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

5.1 Summary

In this work, a new technique for solving the deconvolution problem for seismic processes in the time domain has been postulated. The technique proposed is a modified version of Kalman filter theory which requires a state space formulation, wherein a model is built to represent the seismic reflection process. While acknowledging the fact that this model is only a simplified version of the rather complex seismic problem, nevertheless, it gives an indication as to whether the technique being pursued is feasible.

Some of the model parameters have been assumed unknown as often happens in practical seismic problems. It has been shown that the adaptive filtering scheme proposed in this work accomplishes accurate deconvolution in spite of model uncertainty.

This data processing problem is basically a post experimental analysis type problem and therefore, on line processing is not required. A smoothing operation was derived to improve results. The smoothing and filtering techniques presented here are derived on the basis of inputs that are Gaussian white noise. Since the inputs to the geophysical model are more accurately modeled as poisson in nature, digital simulations were conducted to check whether the algorithms proposed are adequate for the task of deconvolution.

The algorithms that have been presented have an advantage over those presently used in that they are valid for both time invariant systems and time varying systems. A number of experiments have been conducted to demonstrate the performance of the methods presented here under a variety of conditions for deconvolution. These include a variation in measurement noise intensity, uncertainty in the plant noise and the model parameters, comparison of adaptive and non-adaptive filtering and smoothing and the effects of incorrect modeling. In all cases the deconvolution capabilities of the algorithm seemed adequate under reasonable signal to noise levels. The algorithm did not learn the true values of the unknown parameters with any degree of reliability, as was previously hoped for. This may be attributed to the use of poisson inputs.

5.2 Suggestions for Further Research

In this research, emphasis has been on demonstrating the applicability of the Kalman filter theory under various circumstances and uncertainties in the model, rather than on the theoretical aspects of modeling. Research in obtaining a suitable dynamic structure to represent the seismic reflection process should be rewarding. The choice of the proper number of states to adequately model a given process is not an easy one. While there is a definite need to assume a more random wavelet rather than a fixed configuration taken in this work, the order of the system dynamics should not be increased to such an extent as to make it computationally inefficient.

The method of obtaining a suitable model to represent the seismic problem seems to involve taking a closer look at the physics of the seismic reflection process. This involves the propagation of waves in an elastic media and the reflection and transmission coefficients that arise as well as the angle of incidence at the point of impact with the seismic layer.

The method of deriving an appropriate model by taking a state space model consistent with the covariance information is also possible, but the state of the art is not developed sufficiently enough for the time variant case.

When a suitable state space model has been developed, which involves constant parameters (time invariant systems), then there are recursive algorithms that avoid solving the Riccati equations, thereby obtaining significant computational advantages. These procedures involve the so-called Chandrasekhar type algorithms (34).

In the work done, only synthetic data has been used. A comparison with other methods of solving the deconvolution problem such as minimum phase filtering, homomorphic filtering and maximum entropy methods is suggested using the synthetic data.

Finally real field data could be used and the versatility of the filtering and smoothing operations developed here verified and compared with the existing techniques mentioned above.

5.3 Conclusions

The problem of deconvolution using the Kalman filter theory with uncertain modeling knowledge has been proposed and solved.

It has been seen that even for cases of model uncertainty, that is, even when the parameters in the model are not clearly defined that the filtering gives good estimates. However the effect of smoothing, proposed as a process to follow the filtering operation has been shown to refine the estimates to a great extent.

It is significant to point out that the results were not degraded excessively due to model uncertainty in both adaptive filtering and adaptive smoothing operations.

System identification seems to be unreliable and there is no way of determining exactly when to curtail the use of data in filtering. However, in spite of this, the estimates obtained did verify that the deconvolution problem can be solved using the suggested method.

Also, as a matter of interest, the wavelet input could also be estimated from the filtering and smoothing algorithms. This wavelet is called the predictive operator in minimum phase filtering from which the inverse operator is designed and implemented.

Thus the objectives mentioned in the beginning of this work have been met. Further work in the deconvolution problem using this approach should be rewarding.

SELECTED BIBLIOGRAPHY

(1) Wiggins, R. A., and E. A. Robinson. "Recursive Solution to the Multichannel Filtering Problem." Journal of Geophysical Research, Vol. 70, No. 8 (1965), 1885-1891.

(2) Wiener, N. Extrapolation, Interpolation and Smoothing of Stationary Time Series. New York: John Wiley and Sons, Inc., 1970.

(3) Kalman, R. E., and R. S. Bucy. "New Results in Linear Filtering and Prediction Theory." ASME Transactions: Journal of Basic Engg., Vol. 83D (1961), 95-108.

(4) Kalman, R. E. "A New Approach to Linear Filtering and Prediction Problems." ASME Transactions: Journal of Basic Engg., Vol. 82D (1960), 35-45.

(5) Robinson, E. A. "Predictive Decomposition of Seismic Traces." Geophysics, Vol. 22 (1957), 767-778.

(6) Robinson, E. A. "Mathematical Development of Discrete Filters for the Detection of Nuclear Explosions." Journal of Geophysical Research, Vol. 68, No. 19 (1963), 5559-5567.

(7) Rice, R. B. "Inverse Convolution Filters." Geophysics, Vol. 27 (1962), 4-12.

(8) Kunetz, G. "Essai d' analyse de traces seimiques." Geophysical Prospecting, Vol. 9 (1962), 317-341.

(9) Robinson, E. A., and S. Treitel. "Principles of Digital Wiener Filtering." Geophysical Prospecting, Vol. 15 (1967), 311-333.

(10) Clarke, G. K. C. "Time Varying Deconvolution Filters." Geophysics, Vol. 33 (1968), 936-944.

(11) Peacock, K. L., and S. Treitel. "Predictive Deconvolution:Theory and Practice." Geophysics, Vol. 34 (1969), 155-169.

(12) Treitel, S. "The Complex Weiner Filter." Geophysics, Vol. 39 (1974), 169-173.

(13) Ulrych, T. J. "Application of Homomorphic Filtering to Seismology." Geophysics, Vol. 36 (1971), 650-660.

(14) Bayless, J. W., and E. D. Brigham. "Application of Kalman Filtering to Continuous Signal Restoration." Geophysics, Vol. 35 (1970), 2-23.

(15) Norman, N. D. "A Kalman Filter Approach to the Deconvolution of Seismic Signals." Geophysics, Vol. 39 (1974), 1-13.

(16) Ricker, N. "The Form and Nature of Seismic Waves and the Structure of Seismograms." Geophysics, Vol. 5 (1940), 348-366.

(17) Hilborn, C. G., and Lainiotis, D. G. "Optimal Estimation in Presence of Unknown Parameters." IEEE Transactions on Systems Science and Cybernetics, Vol. CCS-5 (1969), 109-115.

(18) Lee, A. Y., and C. S. Sims. "Adaptive Estimation and Stochastic Control for Uncertain Models." International Journal of Control, Vol. 19 (1974), 625-640.

(19) Meditch, J. S. Stochastic Optimal Linear Estimation and Control. New York: McGraw Hill Inc., 1969.

(20) Bryson, A. E., Jr., and M. Frazier. "Smoothing for Linear and Non-Linear Dynamic Systems." Proc. on Optimal System Synthesis Conf., Sept. 11-13, 1962, Wright Patterson and APB, Ohio.

(21) Rauch, H. E. "Solution to the Linear Smoothing Problem." IEEE Transactions on Automatic Control, Vol. AC-8 (1963), 371-372.

(22) Meditch, J. S. "On Optimal Linear Smoothing Theory." Information and Control, Vol. 10 (1961), 598-615.

(23) Fraser, D. C. "On the Application of Optimal Linear Smoothing Techniques to Linear and Non Linear Dynamic Systems." (Unpub. Ph.D. thesis, M.I.T., January, 1967).

(24) Mehra, R. K. "On Optimal and Suboptimal Linear Smoothing." Proc. National Electronics Conf., (1968), 119-124.

(25) Mehra, R. K., and A. E. Bryson, Jr. "Linear Smoothing Using Measurements Containing Noise With an Application to Inertial Navigation." IEEE Transactions on Automatic Control, Vol. AC-13 (1968), 496-503.

(26) Bryson, A. E., Jr., and L. J. Henrikson. "Estimation Using Sampled Data Containing Sequentially Correlated Noise." Tech. Report, Harvard Univ. Div. of Applied Engg. and Physics, Cambridge, Mass., (1967).

(27) Kailath, T., and P. Frost. "An Innovations Approach to Least Squares Estimation-Part 11 Linear Smoothing in Additive Noise." IEEE Transactions on Automatic Control, Vol. AC-13 (1968), 655-660.

(28) Hamming, R. W. Numerical Methods for Scientists and Engineers. New York: McGraw Hill Inc., 1962.

(29) Papoulis, A. Probability, Random Variables and Stochastic Processes. New York: McGraw Hill Inc., 1965.

(30) Beutler, L. J., and O. A. Z. Leneman. "The Spectral Analysis of Impulse Processes." Information and Control, Vol. 12 (1968), 236-258.

(31) Chambers, R. P. "Random Number Generation." IEEE Spectrum, Vol. 4 (1967).

(32) Brown, R. J., and J. R. Rowland. "Autocorrelation Significance in Digital Pseudo Random Number Generation." Internal Tech. Report, School of Electrical Engg., Georgia Institute of Technology, Atlanta, Georgia, (1970), 1-20.

(33) Box, G. E. P., and M. E. Muller. "A Note on the Generation of Normal Deviates." Annals of Mathematical Statistics, Vol. 28 (1958), 610-611.

(34) Kailath, T. "Some New Algorithms for Recursive Estimation in Constant Linear Systems." IEEE Transaction on Information Theory, Vol. IT-19 (1973), 750-760.

APPENDIX

COMPUTER PROGRAM FOR SIMULATING THE MODIFIED BAYLESS AND BRIGHAM MODEL USING ADAPTIVE KALMAN FILTERS

```
C*
C*
C*    *************************
C*    THIS PROGRAM FINDS THE IMPULSE RESPONSE OF THE EARTH  XHAT(1,1)
C*    AFTER GENERATING A SEISMIC TRACE. THIS IS FOR AN EQUAL
C*    AMPLITUDE POISSON INPUT. FOUR MODELS ARE CHOSEN AND THE
C*    PARAMETERS IN THE MODEL ARE FIXED EXCEPT ALPHA WHICH HAS VALUES OF
C*    150, 100, 50, 10.
C*    THE ITERATION IS DONE EVERY 0.1 MSECS. WHILE THE OBSERVATION IS
C*    TAKEN EVERY 0.5 MSECS.
C*    COVARIANCE OF THE PLANT =500.
C*    COVARIANCE OF THE MEASUREMENT NOISE R=10**-5
C*    X IS TRUE STATE.
C*    XHAT IS FILTERED STATE ESTIMATE.
C*    XSAT IS SMOOTHED STATE ESTIMATE.
C*    VAR IS THE COVARIANCE OF THE ERROR.
C*    FOR THE CASE OF RANDOM AMPLITUDE POISSON INPUT , CHANGE
C*    PULSE1=10**4*YG WHERE YG IS A RANDOM VARIABLE HAVING THE FIRST
C*    TWO MOMENTS AS 0 AND 1/3. THE COVARIANCE OF THE PLANT NOISE IS
C*    CHANGED TO Q/3.
C*    THE SUBROUTINES USED ARE RANDU, RMGN, MODEL, RKUT, ADAPT, CHANGE
C*    SMOOTH.
C*    *************************
C*
C*
       DIMENSION TIME(200),TIME1(200),TIMM(200),XHAT(10,1),XHAT1(10,1),
      CXHAT2(10,1),XHAT3(10,1),XHAT4(10,1),ZOBS(1000),VAR1(10,10),
      CVAR2(10,10),VAR3(10,10),VAR4(10,10),YF(1000),VAR(10,1),X(10,10),
      CXD(10,10),ARR(10),BNNOV(10),APROB(10),ALR(10,10),ALKR(10),
      CYHAT1(10,1),YHAT2(10,1),YHAT3(10,1),YHAT4(10,1),COVAR1(10,10),
      CCOVAR2(10,10),COVAR3(10,10),COVAR4(10,10),Z(1000),XSAT(10,1),
      CXDSAT(10,1),X1(1000),X2(1000),X3(1000),XX1(1000),XX2(1000),
      CXX3(1000),XXX1(1000),XXX2(1000),XXX3(1000),XSAT1(10,1),
      CAA1(1000),AA2(1000),AA3(1000),BB1(1000),BB2(1000),BB3(1000),
      CCC1(1000),CC2(1000),CC3(1000),DD1(1000),DD2(1000),DD3(1000),
      CTT1(1000),TT2(1000),TT3(1000),TT4(1000),TT5(1000),TT6(1000),
      CUU1(1000),UU2(1000),UU3(1000),UU4(1000),UU5(1000),UU6(1000),
      CVV1(1000),VV2(1000),VV3(1000),VV4(1000),VV5(1000),VV6(1000),
      CWW1(1000),WW2(1000),WW3(1000),WW4(1000),WW5(1000),WW6(1000),
      CXDSAT1(10,1),XSAT2(10,1),XDSAT2(10,1),XSAT3(10,1),XDSAT3(10,1),
      CXSAT4(10,1),XDSAT4(10,1)
       COMMON/VARIES/VTRUE(3,3)
       COMMON/PARAM/NV,NX,NY,RRR,ANNOV,E
       COMMON/MEL/H(1,3),HT(3,1),R,ZZ,AJ(10,10)
       COMMON/COEFF/Q,ALPHA
       COMMON/KKKKK/IOBS
       COMMON/PPPP/PULSE1
       COMMON/KALMAN/AKGAIN(10,1)
       COMMON/SMEETH/SMGAIN(3,3),N,AMGAIN(3,3)
       E=0.0001
       YMEAN=0.0
       RR=1.0/10**5
       SIG2=SQRT(RR)
       R=RR
       NY=3
       NX=3
       NV=3
       NV1=NV+1
       NX1=NX+1
       NY1=NY+1
       IN=0
       IOBS=5
       ALPHA=50.0
```

```
      TIME(1)=0.0
      M=1
      MX=8143
1     CALL RANDU (MX,MY,YFL)
      MX=MY
      Q=500.
      QQ=1.0/Q
      ZOBS(M)=-QQ*ALOG(1.-YFL)
      TIME(M+1)=ZOBS(M)+TIME(M)
      TDIFF=TIME(M+1)-TIME(M)
      IF (TDIFF.LE.0.0001) GO TO 1
      IF (TIME(M).GE.0.1) GO TO 2
      TIME1(M)=TIME(M)
      M=M+1
      GO TO 1
 2    CONTINUE
C
C  CHANGE THE RANDOM FUNCTION TO A POISSON DISTRIBUTED FUNCTION.
      DO 3 I=1,4
3     X(I,1)=0.0
      DO 5 I=1,4
5     APROB(I)=0.25
      DO 10 I=1,4
      XHAT1(I,1)=0.0
      XHAT2(I,1)=0.0
      XHAT3(I,1)=0.0
10     XHAT4(I,1)=0.0
      H(1,1)=0.0
      H(1,2)=0.0
      H(1,3)=1.0
      DO 14 I=1,3
14    HT(I,1)=H(1,I)
      DO 15 I=1,3
      DO 15 J=1,3
      VAR1(I,J)=0.0
      VAR2(I,J)=0.0
      VAR3(I,J)=0.0
15     VAR4(I,J)=0.0
      VAR1(1,1)=1000.
      VAR2(1,1)=1000.
      VAR3(1,1)=1000.
      VAR4(1,1)=1000.
      KK=200
      IX=31571
      DUM=0.1
      MM=2
      DO 20 I=1,3
      DO 20 J=1,3
      AJ(I,J)=0.0
20    AJ(I,I)=1.0
      CALL RMGN (YMEAN,SIG2,IX,DUM,KK,YF)
      DO 70 N=1,1000
      TIMM(MM)=ABS(TIME1(MM)-N*E)
      IF (TIMM(MM).LE.0.00005) GO TO 21
      GO TO 22
21    PULSE1=10**4
      MM=MM+1
      GO TO 25
22    PULSE1=-500.
C  PULSE SHOULD APPEAR ONLY AT THE TIME OF ITERATION,CORRESPONDING TO
C   TO THE INTEGRATION GOING ON.THIS TIME HAS ALREADY BEEN CALCULATED
C   PREVIOUSLY FOR A FIXED Q.
25    CONTINUE
      Q=500.
      ALPHA=50.0
       DO 35 J=1,4
```

```
      CALL MODEL (X,XD)
35    CALL RKUT (J,E,NX,X,XD,X(4,1))
      IF (IOBS.EQ.5) GO TO 40
      GO TO 45
40    IN=IN+1
      Z(IN)=X(3,1)+YF(IN)
      ZZ=Z(IN)
      IOBS=1
      GO TO 50
 45   IOBS=IOBS+1
50    CONTINUE
      ALPHA=150.0
      CALL ADAPT (XHAT1,VAR1,YHAT1,COVAR1)
      ARR(1)=RRR
      BNNOV(1)=ANNOV
      TT1(N)=VTRUE(1,1)
      TT2(N)=VTRUE(1,2)
      TT3(N)=VTRUE(1,3)
      TT4(N)=VTRUE(2,2)
      TT5(N)=VTRUE(2,3)
      TT6(N)=VTRUE(3,3)
C
C  ALPHA IS GIVEN A VALUE OF 150. AND THE INVERSE COVARIANCE OF THE
C  ERROR IS STORED IN THE TT ARRAYS.
C
      ALPHA=100.0
      CALL ADAPT (XHAT2,VAR2,YHAT2,COVAR2)
      ARR(2)=RRR
      BNNOV(2)=ANNOV
      UU1(N)=VTRUE(1,1)
      UU2(N)=VTRUE(1,2)
      UU3(N)=VTRUE(1,3)
      UU4(N)=VTRUE(2,2)
      UU5(N)=VTRUE(2,3)
      UU6(N)=VTRUE(3,3)
C
C   ALPHA IS GIVEN A VALUE OF 100. AND THE INVERSE COVARIANCE OF THE
C  ERROR IS STORED IN THE UU ARRAYS.
C
      ALPHA=50.0
      CALL ADAPT (XHAT3,VAR3,YHAT3,COVAR3)
      ARR(3)=RRR
      BNNOV(3)=ANNOV
      VV1(N)=VTRUE(1,1)
      VV2(N)=VTRUE(1,2)
      VV3(N)=VTRUE(1,3)
      VV4(N)=VTRUE(2,2)
      VV5(N)=VTRUE(2,3)
      VV6(N)=VTRUE(3,3)
C
C  ALPHA IS GIVEN A VALUE OF 50. AND THE INVERSE COVARIANCE OF THE
C  ERROR IS STORED IN THE VV ARRAYS.
C
      ALPHA=10.0
      CALL ADAPT (XHAT4,VAR4,YHAT4,COVAR4)
      ARR(4)=RRR
      BNNOV(4)=ANNOV
      WW1(N)=VTRUE(1,1)
       WW2(N)=VTRUE(1,2)
      WW3(N)=VTRUE(1,3)
      WW4(N)=VTRUE(2,2)
      WW5(N)=VTRUE(2,3)
      WW6(N)=VTRUE(3,3)
C
C  ALPHA IS GIVEN A VALUE OF 10. AND THE INVERSE COVARIANCE OF THE
C  ERROR IS STORED IN THE WW ARRAYS.
```

```
C
      DO 210 I=1,4
      XHAT1(I,1)=YHAT1(I,1)
      XHAT2(I,1)=YHAT2(I,1)
      XHAT3(I,1)=YHAT3(I,1)
      XHAT4(I,1)=YHAT4(I,1)
210   CONTINUE
C
C  THE ESTIMATES OF THE FOUR MODELS ARE PLACED IN XHAT ARRAYS TO BE
C  USED IN THE NEXT UPDATE.
      DO 250 I=1,3
      DO 250 J=1,3
      VAR1(I,J)=COVAR1(I,J)
      VAR2(I,J)=COVAR2(I,J)
      VAR3(I,J)=COVAR3(I,J)
      VAR4(I,J)=COVAR4(I,J)
250    CONTINUE
C  THE COVARIANCE OF THE ERROR OF THE FOUR MODELS ARE PLACED IN VAR
C  ARRAYS TO BE USED IN THE NEXT UPDATE.
      IF (IOBS.NE.1) GO TO 355
C
C  ONLY UPDATE THE PROBABILITY WHEN THE OBSERVATION IS TAKEN.
C
      DO 320 I=1,4
      DO 320 J=1,4
      IF (I.NE.J) GO TO 310
      GO TO 320
310   ALR(J,I)=(ARR(J)/ARR(I))**0.5*EXP(-0.5*BNNOV(J)**2*ARR(J)+0.5*
     CBNNOV(I)**2*ARR(I))
320    CONTINUE
C
C HAVE OBTAINED THE LIKELIHOOD RATIO.
C
      DO 330 I=1,4
      ALKR(I)=0.0
      DO 330 J=1,4
      IF (I.NE.J) GO TO 335
      GO TO 330
335    ALKR(I)=ALKR(I)+ALR(J,I)*(APROB(J)/APROB(I))
330   CONTINUE
      DO 350 I=1,4
350   APROB(I)=1.0/(1.0+ALKR(I))
C
C HAVE UPDATED THE A POSTERIORI PROBABILITY OF THE CANDIDATE MODELS.
C
355   CONTINUE
      DO 360 I=1,3
      XHAT(I,1)=XHAT1(I,1)*APROB(1)+XHAT2(I,1)*APROB(2)+XHAT3(I,1)*
     CAPROB(3)+XHAT4(I,1)*APROB(4)
360   CONTINUE
      X1(N)=X(1,1)
      X2(N)=X(2,1)
      X3(N)=X(3,1)
366    XX1(N)=XHAT(1,1)
      XX2(N)=XHAT(2,1)
       XX3(N)=XHAT(3,1)
C
C  PLACED THE ESTIMATES IN ARRAYS FOR USE IN PLOTTING AND STORAGE.
C
      AA1(N)=XHAT1(1,1)
      AA2(N)=XHAT1(2,1)
      AA3(N)=XHAT1(3,1)
      BB1(N)=XHAT2(1,1)
      BB2(N)=XHAT2(2,1)
      BB3(N)=XHAT2(3,1)
      CC1(N)=XHAT3(1,1)
```

```
      CC2(N)=XHAT3(2,1)
      CC3(N)=XHAT3(3,1)
      DD1(N)=XHAT4(1,1)
      DD2(N)=XHAT4(2,1)
      DD3(N)=XHAT4(3,1)
70    CONTINUE
      XSAT(4,1)=0.1
      EE=-0.0001
C
C  HAVE TC INTEGRATE BACKWARDS FOR THE SMOORTHING PROCESS.
C
      XSAT1(1,1)=AA1(1000)
      XSAT1(2,1)=AA2(1000)
      XSAT1(3,1)=AA3(1000)
      XSAT2(1,1)=BB1(1000)
      XSAT2(2,1)=BB2(1000)
      XSAT2(3,1)=BB3(1000)
      XSAT3(1,1)=CC1(1000)
      XSAT3(2,1)=CC2(1000)
      XSAT3(3,1)=CC3(1000)
      XSAT4(1,1)=DD1(1000)
      XSAT4(2,1)=DD2(1000)
      XSAT4(3,1)=DD3(1000)
C  THE TERMINAL CONDITION IS SATISFIED.
C
      DO 1070 M=1,999
      N=1000-M
       Q=500.
      ALPHA=150.
      CALL CHANGE (UU1,UU2,UU3,UU4,UU5,UU6)
      DO 1005 I=1,3
      DO 1005 J=1,3
1005  SMGAIN(I,J)=AMGAIN(I,J)
      DO 1010 J=1,4
      CALL SMOOTH (XSAT1,XDSAT1,AA1,AA2,AA3)
1010   CALL RKUT (J,EE,NX,XSAT1,XDSAT1,XSAT(4,1))
C
C  FOR ALPHA=150, THE SMOOTHED ESTIMATE IS OBTAINED.
C
      XSAT(4,1)=XSAT(4,1)-EE
      ALPHA=100.0
      CALL CHANGE (TT1,TT2,TT3,TT4,TT5,TT6)
      DO 1015 I=1,3
      DO 1015 J=1,3
1015   SMGAIN(I,J)=AMGAIN(I,J)
      DO 1020 J=1,4
       CALL SMOOTH (XSAT2,XDSAT2,BB1,BB2,BB3)
1020    CALL RKUT (J,EE,NX,XSAT2,XDSAT2,XSAT(4,1))
C
C  FOR ALPHA=100, THE SMOOTHED ESTIMATE IS OBTAINED.
C
      XSAT(4,1)=XSAT(4,1)-EE
      ALPHA=50.0
      CALL CHANGE (VV1,VV2,VV3,VV4,VV5,VV6)
      DO 1025 I=1,3
      DC 1025 J=1,3
1025  SMGAIN(I,J)=AMGAIN(I,J)
      DO 1030 J=1,4
      CALL SMOOTH (XSAT3,XDSAT3,CC1,CC2,CC3)
1030    CALL RKUT (J,EE,NX,XSAT3,XDSAT3,XSAT(4,1))
C
C  FOR ALPHA=50,THE SMOOTHED ESTIMATE IS OBTAINED.
C
      XSAT(4,1)=XSAT(4,1)-EE
      ALPHA=10.0
      CALL CHANGE (WW1,WW2,WW3,WW4,WW5,WW6)
```

```
      DO 1035 I=1,3
      DO 1035 J=1,3
1035  SMGAIN(I,J)=AMGAIN(I,J)
      DO 1040 J=1,4
      CALL SMOOTH (XSAT4,XDSAT4,DD1,DD2,DD3)
1040  CALL RKUT (J,EE,NX,XSAT4,XDSAT4,XSAT(4,1))
C
C  FOR ALPHA=10,THE SMOOTHED ESTIMATE IS OBTAINED.
C
      DO 1045 I=1,3
      XSAT(I,1)=APROB(1)*XSAT1(I,1)+APROB(2)*XSAT2(I,1)+
     CAPROB(3)*XSAT3(I,1)+APROB(4)*XSAT4(I,1)
1045     CONTINUE
C
C  FOUND THE SMOOTHED ESTIMATES BY THE WEIGHTED SUM OF THE CANDIDATE
C  MODEL ESTIMATES.
C
      XXX1(N)=XSAT(1,1)
      XXX2(N)=XSAT(2,1)
       XXX3(N)=XSAT(3,1)
C
C  THE SMOOTHED VALUES ARE STORED IN ARRAYS FOR FURTHER USE.
C
1070   CONTINUE
      XXX1(1000)=XX1(1000)
C  APPROPRIATE WRITE STATEMENTS MUST BE PUT WHEREVER NECESSARY.
C
      STOP
      END


      SUBROUTINE MODEL (X,XD)                                           MDEL0010
C*                                                                      MDELCCCC
C*    ************************                                          MDELCCCC
C*    THIS PROGRAM GIVES THE STATE SPACE MODEL OF THE GEOPHYSICAL      MDELCCCC
C*    SYSTEM FROM WHICH THE OBSERVATIONS ARE OBTAINED.                  MDELCCCC
C*    X(1,1) IS THE STATE THAT IS DESIRED IN DECONVOLUTION. IT IS THE  MDELCCCC
C*    IMPULSE RESPONSE OF THE EARTH.                                    MDELCCCC
C*    THE PROGRAM IS FOR A MAXIMUM OF TEN FIRST ORDER DIFFERENTIAL     MDELCCCC
C*    EQUATIONS. IF MORE ARE REQUIRED THEN REDIMENSION THE X AND XD.   MDELCCCC
C*    ************************                                          MDELCCCC
C*                                                                      MDELCCCC
       DIMENSION X(10,10),XD(10,10)                                     MDEL0020
      COMMON/COEFF/Q,ALPHA                                              MDEL0030
      COMMON/PPPP/PULSE1                                                MDEL0040
      XD(1,1)=-1000.*X(1,1)+1000.*PULSE1                                MDEL0050
      XD(2,1)=314.7*X(1,1)-10**5*X(3,1)                                 MDEL0060
       XD(3,1)=X(2,1)-2.*ALPHA*X(3,1)                                   MDEL0070
      RETURN                                                            MDEL0080
      END                                                               MDEL0090
```

```
      SUBROUTINE RK4(KUTTA,DT,NX,X,DX,TM)
C
C ...............................................................
C SUBROUTINE RK4 INTEGRATES UPTO 50 FIRST ORDER DIFFERENTIAL EQUATIONS
C  IN THE FORM OF A TEN BY FIVE ARRAY.
C IF MORE ARE REQUIRED THEN WE NEED TO REDIMENSION XA,DXA ARRAYS.
C THIS ROUTINE USES RUNGE KUTTA FOURTH ORDER INTEGRATION TECHNIQUE.
C KUTTA TAKES THE VALUES 1,2,3,4.
C DT IS THE STEP SIZE OF INTEGRATION.
C NX IS THE NUMBER OF EQUATIONS TO BE INTEGRATED.
C X IS THE STATE VECTOR REPRESENTING SET OF FIRST ORDER DIFFERENTIAL
C EQUATIONS TO BE INTEGRATED.
C
C DX IS THE DERIVATIVE OF X.
C TM IS THE CURRENT VALUE OF TIME.THE INDEPENDENT VARIABLE.
C ...............................................................
C
      DIMENSION X(10,5),XA(10,5),DX(10,5),DXA(10,5)
      GO TO (10,30,50,70),KUTTA
10    HDT=0.5*DT
      DO 20 I=1,NX
      DO 20 J=1,I
      XA(I,J)=X(I,J)
      DXA(I,J)=DX(I,J)
20    X(I,J)=X(I,J)+HDT*DX(I,J)
      TM= TM+HDT
      RETURN
30    DO 40 I=1,NX
      DO 40 J=1,I
      DXA(I,J)=DXA(I,J)+(DX(I,J)+DX(I,J))
40    X(I,J)=XA(I,J)+HDT*DX(I,J)
      RETURN
50    DO 60 I=1,NX
      DO 60 J=1,I
      DXA(I,J)=DXA(I,J)+(DX(I,J)+DX(I,J))
60    X(I,J)=XA(I,J)+DT*DX(I,J)
      TM= TM+HDT
      RETURN
70    VDT=DT*0.1666667
      DO 80 I=1,NX
      DO 80 J=1,I
80    X(I,J)=XA(I,J)+VDT*(DXA(I,J)+DX(I,J))
      RETURN
       END
```

```
      SUBROUTINE RMGN(XMEAN,SIG,IX,DUM,K,YFL)                          RMGN0010
C*    **************************                                       RMGNCCCC
C*                                                                     RMGNCCCC
C     SUBROUTINE RMGN................                                  RMGNCCCC
C     GENERATING PSEUDO RANDOM NUMBERS WITH MEAN EQUAL TO XMEAN,       RMGNCCCC
C     AND STANDARD DEVIATION EQUAL TO  SIG.                            RMGNCCCC
C     IA=19971          IX=31571           M=2**20= 1048576            RMGNCCCC
C     Z(K+1)= IA*Z(K)         (MODULO M )                              RMGNCCCC
C     VALUES TO BE SPECIFIED FIRST TIME .............                  RMGNCCCC
C     1. DUM=0.1              2. IX=31571     3. K NUMBER  OF RANDOM NUMBERMGNCCCC
C     YFL IS ARRAY OF RANDOM NUMBERS GENERATED OF DIMENSION K.         RMGNCCCC
C*                                                                     RMGNCCCC
C*    **************************                                       RMGNCCCC
      DIMENSION YFL(1)                                                 RMGN0020
      IA= 19971                                                        RMGN0030
      DO 1 I=1,K                                                       RMGN0040
      IY= IA*IX                                                        RMGN0050
      IYP=IY/1048576                                                   RMGN0060
      IX= IY-IYP*1048576                                               RMGN0070
      AX=IX                                                            RMGN0080
      U=AX/1048576.                                                    RMGN0090
      IF(U)410,410,412                                                 RMGN0100
410   U=-U                                                             RMGN0110
412   CONTINUE                                                         RMGN0120
      IX=IY                                                            RMGN0130
      TZ= SQRT(-2.0*ALOG(DUM))*SIG                                     RMGN0140
      XNORM=     TZ*COS(6.28318*U)+XMEAN                               RMGN0150
      DUM=U                                                            RMGN0160
1     YFL(I)= XNORM                                                    RMGN0170
      RETURN                                                           RMGN0180
      END                                                              RMGN0190
```

```
      SUBROUTINE VARIN (X,XD)                                          VARI0010
C*                                                                     VARICCCC
C*    **************************                                       VARICCCC
C*    THIS PROGRAM IS TO SOLVE THE VARIANCE EQUATION OF THE PREDICTOR  VARICCCC
C*    PART IN ADAPTIVE FILTERING.                                      VARICCCC
C*    ALPHA AND Q ARE VARIABLE.                                        VARICCCC
C*    THE PROGRAM IS FOR HUNDRED FIRST ORDER DIFFERENTIAL EQUATIONS.   VARICCCC
C*    IF MORE ARE REQUIRED THEN REDIMENSION X AND XD.                  VARICCCC
C*    **************************                                       VARICCCC
C*                                                                     VARICCCC
      DIMENSION X(10,10),XD(10,10)                                     VARI0020
      COMMON/COFFF/Q,ALPHA                                             VARI0030
      XD(1,1)=-2000.*X(1,1)+10**6*Q                                    VARI0040
      XD(2,1)=-1000.*X(2,1)+314.7*X(1,1)-10**5*X(3,1)                  VARI0050
      XD(3,1)=-1000.*X(3,1)+X(2,1)-2.*ALPHA*X(3,1)                     VARI0060
      XD(2,2)=629.4*X(2,1)-2.*10**5*X(3,2)                             VARI0070
       XD(3,2)=X(2,2)+314.7*X(3,1)-10**5*X(3,3)-2.*ALPHA*X(3,2)        VARI0080
      XD(3,3)=2.*X(3,2)-4.*ALPHA*X(3,3)                                VARI0090
      RETURN                                                           VARI0100
      END                                                              VARI0110
```

```
      SUBROUTINE RKUT (KUTTA,DT,NX,X,DX,TM)                             RKUT0010
C                                                                       RKUTCCCC
C ......................................................................RKUTCCCC
C SUBROUTINE RKUT INTEGRATES UPTO 50 FIRST ORDER DIFFERENTIAL EQUATIONS RKUTCCCC
C  IN THE FORM OF A COLUMN VECTOR FIFTY BY ONE.                         RKUTCCCC
C IF MORE ARE REQUIRED THEN WE NEED TO REDIMENSION XA,DXA ARRAYS.       RKUTCCCC
C THIS ROUTINE USES RUNGE KUTTA FOURTH ORDER INTEGRATION TECHNIQUE.     RKUTCCCC
C KUTTA TAKES THE VALUES 1,2,3,4.                                       RKUTCCCC
C DT IS THE STEP SIZE OF INTEGRATION.                                   RKUTCCCC
C NX IS THE NUMBER OF EQUATIONS TO BE INTEGRATED.                       RKUTCCCC
C X IS THE STATE VECTOR REPRESENTING SET OF FIRST ORDER DIFFERENTIAL    RKUTCCCC
C EQUATIONS TO BE INTEGRATED.                                           RKUTCCCC
C                                                                       RKUTCCCC
C DX IS THE DERIVATIVE OF X.                                            RKUTCCCC
C TM IS THE CURRENT VALUE OF TIME.THE INDEPENDENT VARIABLE.             RKUTCCCC
C ......................................................................RKUTCCCC
C                                                                       RKUTCCCC
C                                                                       RKUTCCCC
      DIMENSION X(50,1),XA(50,1),DX(50,1),DXA(50,1)                     RKUT0020
      J=1                                                               RKUT0030
      GO TO (10,30,50,70),KUTTA                                         RKUT0040
10    HDT=0.5*DT                                                        RKUT0050
      DO 20 I=1,NX                                                      RKUT0060
      XA(I,J)=X(I,J)                                                    RKUT0070
      DXA(I,J)=DX(I,J)                                                  RKUT0080
20     X(I,J)=X(I,J)+HDT*DX(I,J)                                        RKUT0090
      TM=TM+HDT                                                         RKUT0100
      RETURN                                                            RKUT0110
30    DO 40 I=1,NX                                                      RKUT0120
      DXA(I,J)=DXA(I,J)+(DX(I,J)+DX(I,J))                               RKUT0130
40    X(I,J)=XA(I,J)+HDT*DX(I,J)                                        RKUT0140
      RETURN                                                            RKUT0150
50     DO 60 I=1,NX                                                     RKUT0160
      DXA(I,J)=DXA(I,J)+(DX(I,J)+DX(I,J))                               RKUT0170
60     X(I,J)=XA(I,J)+DT*DX(I,J)                                        RKUT0180
      TM=TM+HDT                                                         RKUT0190
      RETURN                                                            RKUT0200
70    VDT=DT*0.1666667                                                  RKUT0210
      DO 80 I=1,NX                                                      RKUT0220
80    X(I,J)=XA(I,J)+VDT*(DXA(I,J)+DX(I,J))                             RKUT0230
      RETURN                                                            RKUT0240
      END                                                               RKUT0250


       SUBROUTINE PRED (X,XD)                                           PRED0010
C*                                                                      PREDCCCC
C*    *************************                                         PREDCCCC
C*    THIS PROGRAM GIVES THE PREDICTOR STATES FOR THE ADAPTIVE FILTERING PREDCCCC
C*    PROCESS.                                                          PREDCCCC
C*    ALPHA IS VARIABLE.                                                PREDCCCC
C*    THE PROGRAM IS FOR TEN FIRST ORDER DIFFERENTIAL EQUATIONS.        PREDCCCC
C*    IF MORE ARE REQUIRED THEN REDUMENSION X AND XD.                   PREDCCCC
C*                                                                      PREDCCCC
       DIMENSION X(10,10),XD(10,10)                                     PRED0020
      COMMON/COEFF/Q,ALPHA                                              PRED0030
      XD(1,1)=-1000.*X(1,1)                                             PRED0040
      XD(2,1)=314.7*X(1,1)-10**5*X(3,1)                                 PRED0050
      XD(3,1)=X(2,1)-2.*ALPHA*X(3,1)                                    PRED0060
      RETURN                                                            PRED0070
      END                                                               PRED0080
```

```
      SUBROUTINE INV(A,NA,DET,L)                                        INV 0010
C*    *************************                                         INV CCCC
C*                                                                      INV CCCC
C*    THIS SUBROUTINE FINDS THE INVERSE OF A MATRIX.                    INV CCCC
C*    A IS THE INPUT MATRIX TO BE INVERTED . IT IS DESTROYED IN         INV CCCC
C*    COMPUTATION AND REPLACED BY A RESULTANT MATRIX.                   INV CCCC
C*    NA IS THE ORDER OF THE MATRIX.NA(1) IS THE NUMBER OF ROWS,        INV CCCC
C*    AND NA(2) IS THE NUMBER OF COLUMNS.                               INV CCCC
C*    D IS THE RESULTANT DETERMINANT.                                   INV CCCC
C*    L IS A WORK VECTOR OF LENGTH 2N.                                  INV CCCC
C*    THE STANDARD GAUSS-JORDAN METHOD IS USED. THE DETERMINANT IS      INV CCCC
C*    CALCULATED. A DETERMINANT OF ZERO INDICATES THAT THE MATRIX IS    INV CCCC
C*    SINGULAR.                                                         INV CCCC
C*                                                                      INV CCCC
C*    *************************                                         INV CCCC
      DIMENSION A(1), L(1),NA(2)                                        INV 0020
C     SEARCH FOR LARGEST ELEMENT                                        INV CCCC
      DET= 1.                                                           INV 0030
      N=NA(1)                                                           INV 0040
      NSQ=N*N                                                           INV 0050
      NK = - N                                                          INV 0060
      DO 80 K= 1, N                                                     INV 0070
      NK = NK + N                                                       INV 0080
      L(K) = K                                                          INV 0090
      L(N+K)=K                                                          INV 0100
      KK = NK + K                                                       INV 0110
      BIGA = A(KK)                                                      INV 0120
      DO 20 J= K, N                                                     INV 0130
      IZ = N*(J - 1)                                                    INV 0140
      DO    20 I= K, N                                                  INV 0150
      IJ = IZ + I                                                       INV 0160
10    IF (ABS(BIGA)-ABS(A(IJ))) 15,20,20                                INV 0170
   15 BIGA = A(IJ)                                                      INV 0180
      L(K) = I                                                          INV 0190
      L(N+K)=J                                                          INV 0200
   20 CONTINUE                                                          INV 0210
C     INTERCHANGE ROWS                                                  INV CCCC
      J = L(K)                                                          INV 0220
      IF(J - K) 35, 35, 25                                              INV 0230
   25 KI = K - N                                                        INV 0240
      DO 30 I = 1, N                                                    INV 0250
      KI = KI + N                                                       INV 0260
      HOLD = -A(KI)                                                     INV 0270
      JI = KI - K + J                                                   INV 0280
      A(KI) = A(JI)                                                     INV 0290
   30 A(JI) = HOLD                                                      INV 0300
C     INTERCHANGE COLUMNS                                               INV CCCC
   35 I=L(N+K)                                                          INV 0310
      IF (I - K) 45, 45, 38                                             INV 0320
   38 JP = N*(I - 1)                                                    INV 0330
      DO 40 J= 1, N                                                     INV 0340
      JK = NK + J                                                       INV 0350
      JI = JP + J                                                       INV 0360
      HOLD = -A(JK)                                                     INV 0370
      A(JK) = A(JI)                                                     INV 0380
   40 A(JI) = HOLD                                                      INV 0390
C     DIVIDE COLUMN BY MINUS PIVOT(VALUE OF PIVOT ELEMENTS IS CONTAINED INV CCCC
C     IN BIGA)                                                          INV CCCC
   45 IF(BIGA) 48, 46, 48                                               INV 0400
   46 DET = 0.                                                          INV 0410
      KKK=KK-1                                                          INV 0420
      WRITE  (6,1046)  KKK                                              INV 0430
 1046 FORMAT  (' INV ERROR  DETERMINANT OF A=0   RANK OF A=',I4)        INV 0440
```

```
      RETURN                                                            INV 0450
   48 DO 55 I= 1, N                                                     INV 0460
      IF (I - K) 50, 55, 50                                             INV 0470
   50 IK = NK + I                                                       INV 0480
      A(IK) =-A(IK)/(BIGA)                                              INV 0490
   55 CONTINUE                                                          INV 0500
C     REDUCE MATRIX                                                     INV CCCC
      DO 65 I= 1, N                                                     INV 0510
      IK = NK + I                                                       INV 0520
      HOLD = A(IK)                                                      INV 0530
      IJ = I - N                                                        INV 0540
      DO 65 J= 1, N                                                     INV 0550
      IJ = IJ + N                                                       INV 0560
      IF(I  - K ) 60, 65, 60                                            INV 0570
   60 IF(J- K) 62, 65, 62                                               INV 0580
   62 KJ = IJ -I + K                                                    INV 0590
      A(IJ) = HOLD* A(KJ) + A(IJ)                                       INV 0600
   65 CONTINUE                                                          INV 0610
C     DIVIDE ROW BY PIVOT                                               INV CCCC
      KJ = K - N                                                        INV 0620
      DO 75 J= 1, N                                                     INV 0630
      KJ = KJ + N                                                       INV 0640
      IF(J - K) 70, 75, 70                                              INV 0650
   70 A(KJ) = A(KJ)/BIGA                                                INV 0660
   75 CONTINUE                                                          INV 0670
C     PRODUCT OF PIVOTS                                                 INV CCCC
      DET=DET*BIGA                                                      INV 0680
C     REPLACE PIVOT BY RECIPROCAL                                       INV CCCC
      A(KK) = 1./BIGA                                                   INV 0690
   80 CONTINUE                                                          INV 0700
C     FINAL ROW AND COLUMN INTERCHANGE                                  INV CCCC
      K = N                                                             INV 0710
  100 K = K - 1                                                         INV 0720
      IF(K) 150, 150, 105                                               INV 0730
  105 I = L(K)                                                          INV 0740
      IF (I - K ) 120, 120, 108                                         INV 0750
  108 JQ = N*( K - 1)                                                   INV 0760
      JR = N*(I- 1)                                                     INV 0770
      DO 110 J= 1, N                                                    INV 0780
      JK = JQ + J                                                       INV 0790
      HOLD = A(JK)                                                      INV 0800
      JI = JR + J                                                       INV 0810
      A(JK) = - A(JI)                                                   INV 0820
  110 A(JI) = HOLD                                                      INV 0830
  120 J=L(N+K)                                                          INV 0840
      IF(J - K) 100, 100, 125                                           INV 0850
  125 KI = K - N                                                        INV 0860
      DO 130 I= 1, N                                                    INV 0870
      KI = KI + N                                                       INV 0880
      HOLD = A(KI)                                                      INV 0890
      JI = KI - K + J                                                   INV 0900
      A(KI) = - A(JI)                                                   INV 0910
  130 A(JI) = HOLD                                                      INV 0920
      GO TO 100                                                         INV 0930
  150 RETURN                                                            INV 0940
      END                                                               INV 0950
```

```
      SUBROUTINE ADAPT (XHAT,VAR,YHAT,COVAR)                              ADAP0010
C                                                                         ADAPCCCC
C                                                                         ADAPCCCC
C*    ************************                                            ADAPCCCC
C*                                                                        ADAPCCCC
C*    THIS SUBROUTINE SOLVES THE CORRECTOR PART OF ADAPTIVE FILTERING.    ADAPCCCC
C*    IT FINDS OUT THE ESTIMATE AND VARIANCE OF A SYSTEM, GIVEN THE       ADAPCCCC
C*    OBSERVATION, H MATRIX,, ITS TRANSPOSE, THE COVARIANCE OF THE        ADAPCCCC
C*    OBSERVATION NOISE R, THE COVARIANCE OF THE PLANT NOISE Q.IT UPDATESADAPCCCC
C*    THE ESTIMATE EVERY IOBS  VALUES, WHICH IN THIS PROGRAM IS FIVE.     ADAPCCCC
C*    IT IS USED FOR A SYSTEM OF ORDER TEN. IF MORE ARE REQUIRED          ADAPCCCC
C*    THEN REDIMENSION THE PROGRAM.                                       ADAPCCCC
C*    XHAT(4,1) IS THE TIME.                                              ADAPCCCC
C*    VAR(7,1) IS THE TIME.                                               ADAPCCCC
C*    VAR IS THE INITIAL VARIANCE.                                        ADAPCCCC
C*    COVAR IS THE UPDATED VARIANCE.                                      ADAPCCCC
C*    XHAT IS THE OLD ESTIMATE.                                           ADAPCCCC
C*    YHAT IS THE UPDATED ESTIMATED.                                      ADAPCCCC
C*    E IS THE INCREMENT FOR THE INTEGRATION .                            ADAPCCCC
C*    RRR, ANNCV ARE RETAINED FOR USE IN THE MAIN PROGRAM.                ADAPCCCC
C*    THE SUBPROGRAMS USED ARE PRED, RKUT, VARIN, RK4, INV.               ADAPCCCC
C*                                                                        ADAPCCCC
C*    ************************                                            ADAPCCCC
C                                                                         ADAPCCCC
      DIMENSION X(10,10),XDHAT(10,10),VAR(10,10),VARD(10,10),CVAR(10,1),ADAP0020
     CCORR(10,1),XKGAIN(10,1),AKG(10,10),ADENT(10,10),COVAR(10,10),       ADAP0030
     CXHAT(10,1),YHAT(10,1),XD(10,10),BKGAIN(1,10),CKGAIN(1,10),          ADAP0040
     CDKGAIN(10,10),BDENT(10,10),ENTER(10,10),EENTE(10,10),NA(2),L(6)     ADAP0050
      COMMON/VARIES/VTRUE(3,3)                                            ADAP0060
      COMMON/PARAM/NV,NX,NY,RRR,ANNCV,E                                   ADAP0070
      COMMON/COEFF/Q,ALPHA                                                ADAP0080
      COMMON/KKKKK/IOBS                                                   ADAP0090
      COMMON/KALMAN/AKGAIN(10,1)                                          ADAP0095
      COMMON/MEL/H(1,3),HT(3,1),R,ZZ,AJ(10,10)                            ADAP0100
      NA(1)=3                                                             ADAP0110
      NA(2)=3                                                             ADAP0120
      DO 46 J=1,4                                                         ADAP0130
      CALL PRED (XHAT,XDHAT)                                              ADAP0140
46    CALL RKUT (J,E,NY,XHAT,XDHAT,XHAT(4,1))                             ADAP0150
C    SOLVED THE PREDICTOR EQUATONS.                                       ADAPCCCC
C    XHAT(4,1) IS THE TIME                                                ADAPCCCC
       VAR(7,1)=XHAT(4,1)-E                                               ADAP0160
      DO 47 J=1,4                                                         ADAP0170
      CALL VARIN (VAR,VARD)                                               ADAP0180
47     CALL RK4 (J,E,NV,VAR,VARD,VAR(7,1))                                ADAP0190
C   SOLVED THE VARIANCE EQUATIONS.                                        ADAPCCCC
C    VAR(7,1) IS THE TIME.                                                ADAPCCCC
      DO 49 I=1,3                                                         ADAP0200
      DO 49 J=1,3                                                         ADAP0210
      VTRUE(I,J)=VAR(I,J)                                                 ADAP0220
      VTRUE(J,I)=VTRUE(I,J)                                               ADAP0230
49    VAR(I,J)=VAR(J,I)                                                   ADAP0240
      CALL INV (VTRUE,NA,DET,L)                                           ADAP0250
C    FOUND OUT THE INVERSE OF THE ERROR COVARIANCE TO BE USED IN THE      ADAPCCCC
C    SMOOTHING ALGORITHM.                                                 ADAPCCCC
       IF (IOBS.NE.1)GO TO 96                                             ADAP0260
      DO 50 I=1,3                                                         ADAP0270
      CVAR(I,1)=0.0                                                       ADAP0280
      DO 50 K=1,3                                                         ADAP0290
50    CVAR(I,1)=CVAR(I,1)+VAR(I,K)*HT(K,1)                                ADAP0300
      CCVAR=0.0                                                           ADAP0310
      DO 55 K=1,3                                                         ADAP0320
55    CCVAR=CCVAR+H(1,K)*CVAR(K,1)                                        ADAP0330
```

```
      RRR=1.0/(CCVAR+R)                                                ADAP0340
      DO 60 I=1,3                                                      ADAP0350
60     AKGAIN(I,1)=CVAR(I,1)*RRR                                       ADAP0360
C  SOLVED THE KALMAN GAIN EQUATIONS.                                   ADAPCCCC
C  AKGAIN(I,1) ARE THE THREE VALUES OF THE KALMAN GAIN                 ADAPCCCC
      DO 62 I=1,3                                                      ADAP0370
62     BKGAIN(1,I)=AKGAIN(I,1)                                         ADAP0380
      DO 63 I=1,3                                                      ADAP0390
63    CKGAIN(1,I)=R*BKGAIN(1,I)                                        ADAP0400
      DO 64 I=1,3                                                      ADAP0410
      DO 64 J=1,3                                                      ADAP0420
64    DKGAIN(I,J)=AKGAIN(I,1)*CKGAIN(1,J)                              ADAP0430
      HX=0.0                                                           ADAP0440
      DO 65 J=1,3                                                      ADAP0450
65    HX=HX+H(1,J)*XHAT(J,1)                                           ADAP0460
      ANNOV=ZZ-HX                                                      ADAP0470
C  SOLVED FOR  TO GET THE INNOVATIONS.                                 ADAPCCCC
      DO 70 I=1,3                                                      ADAP0480
      CORR(I,1)=AKGAIN(I,1)*ANNOV                                      ADAP0490
70    YHAT(I,1)=XHAT(I,1)+CORR(I,1)                                    ADAP0500
      YHAT(4,1)=XHAT(4,1)                                              ADAP0510
C  SOLVED FOR UPDATED ESTIMATE WHERE CORR IS THE CORRECTOR PART ADDED ONADAPCCCC
      DO 75 I=1,3                                                      ADAP0520
      DO 75 J=1,3                                                      ADAP0530
75    AKG(I,J)=AKGAIN(I,1)*H(1,J)                                      ADAP0540
      DO 80 I=1,3                                                      ADAP0550
      DO 80 J=1,3                                                      ADAP0560
80    ADENT(I,J)=(AJ(I,J)-AKG(I,J))                                    ADAP0570
      DO 81 I=1,3                                                      ADAP0580
      DO 81 J=1,3                                                      ADAP0590
81    BDENT(J,I)=ADENT(I,J)                                            ADAP0600
      DO 82 I=1,3                                                      ADAP0610
      DO 82 J=1,3                                                      ADAP0620
      ENTER(I,J)=0.0                                                   ADAP0630
      DO 82 K=1,3                                                      ADAP0640
82    ENTER(I,J)=ENTER(I,J)+VAR(I,K)*BDENT(K,J)                        ADAP0650
      DO 83 I=1,3                                                      ADAP0660
      DO 83 J=1,3                                                      ADAP0670
      EENTE(I,J)=0.0                                                   ADAP0680
      DO 83 K=1,3                                                      ADAP0690
83    EENTE(I,J)=EENTE(I,J)+ADENT(I,K)*ENTER(K,J)                      ADAP0700
      DO 85 I=1,3                                                      ADAP0710
      DO 85 J=1,3                                                      ADAP0720
85    COVAR(I,J)=EENTE(I,J)+DKGAIN(I,J)                                ADAP0730
C  UPDATED THE VARIANCE EQUATIONS.                                     ADAPCCCC
      GO TO 99                                                         ADAP0740
96    DO 97 I=1,4                                                      ADAP0750
97    YHAT(I,1)=XHAT(I,1)                                              ADAP0760
      RRR=0.0                                                          ADAP0770
      ANNOV=0.0                                                        ADAP0780
      DO 98 I=1,3                                                      ADAP0790
      DO 98 J=1,3                                                      ADAP0800
98    COVAR(I,J)=VAR(I,J)                                              ADAP0810
99    RETURN                                                           ADAP0820
      END                                                              ADAP0830
```

```
      SUBROUTINE CHANGE (SS1,SS2,SS3,SS4,SS5,SS6)
C     *************************
C*    THIS PROGRAM TAKES THE ONE DIMENSIONAL INVERSE COVARIANCE MATRIX
C*    ALREADY PRECOMPUTED IN THE FILTERING PART AND CHANGES IT TO A
C*    TWO DIMENSIONAL ARRAY AND FINDS THE SMOOTHED FILTER GAIN.
C*    SS IS THE INVERSE OF THE ERROR COVARIANCE IN A SINGLE ARRAY. IT
C*    IS CHANGED TO A TWO DIMENSIONAL ARRAY TVVR.
C*    TVVR IS THE INVERSE COVARIANCE OF ERROR .
C*    AMGAIN IS THE SMOOTHED GAIN OBTAINED ,TO BE USED IN THE
C*    SMOOTHED ESTIMATE.
C*    THIS PROGRAM WORKS FOR HUNDRED VALUES OF THE INVERSE COVARIANCE
C*    OF ERROR.
C     *************************
      DIMENSION SS1(1000),SS2(1000),SS3(1000),SS4(1000),SS5(1000),
     CSS6(1000),TVVR(10,10),BQB(10,10)
      COMMON/COEFF/Q,ALPHA
      COMMON/SMEETH/SMGAIN(3,3),N,AMGAIN(3,3)
      TVVR(1,1)=SS1(N)
      TVVR(1,2)=SS2(N)
      TVVR(1,3)=SS3(N)
      TVVR(2,2)=SS4(N)
      TVVR(2,3)=SS5(N)
      TVVR(3,3)=SS6(N)
      DO 1 I=1,3
      DO 1 J=1,3
1     TVVR(J,I)=TVVR(I,J)
      DO 2 I=1,3
      DO 2 J=1,3
2     BQB(I,J)=0.0
      BQB(1,1)=10**6*Q
      DO 3 I=1,3
      DO 3 J=1,3
      AMGAIN(I,J)=0.0
      DO 3 K=1,3
3     AMGAIN(I,J)=AMGAIN(I,J)+BQB(I,K)*TVVR(K,J)
      RETURN
      END
```

```
      SUBROUTINE SMOOTH (XSAT,XDSAT,XX1,XX2,XX3)                        SM000010
C                                                                       SMOOCCCC
C*    ***********************                                           SMOOCCCC
C*                                                                      SMOOCCCC
C*    THIS SUBROUTINE CALCULATES THE ADAPTIVE SMOOTHED ESTIMATES OF THE SMOOCCCC
C*    STATES, GIVEN THE SMOOTHED FILTER GAIN AND THE FILTERED ESTIMATES SMOOCCCC
C*    AT THE TIME OF COMPUTATION.                                       SMOOCCCC
C*    XX1, XX2, XX3 ARE THE FILTERED STATES STORED FROM THE ADAPTIVE    SMOOCCCC
C*    FILTERING ALGORITHM.                                              SMOOCCCC
C*    SMGAIN IS THE SMOOTHED GAIN MATRIX FROM SUBROUTINE CHANGE.        SMOOCCCC
C*    ALPHA IS VARIABLE.                                                SMOOCCCC
C*                                                                      SMOOCCCC
C*    **************************                                        SMOOCCCC
C                                                                       SMOOCCCC
      DIMENSION XSAT(10,1),XDSAT(10,1),AAA(10),SMCORR(10),              SM000020
     CXX1(1000),XX2(1000),XX3(1000)                                     SM000030
      COMMON/SMEETH/SMGAIN(3,3),N,AMGAIN(3,3)                           SM000040
      COMMON/COEFF/Q,ALPHA                                              SM000050
      AAA(1)=XSAT(1,1)-XX1(N)                                           SM000060
      AAA(2)=XSAT(2,1)-XX2(N)                                           SM000070
      AAA(3)=XSAT(3,1)-XX3(N)                                           SM000080
      DO 15 I=1,3                                                       SM000090
      SMCORR(I)=0.0                                                     SM000100
      DO 15 J=1,3                                                       SM000110
      SMCORR(I)=SMCORR(I)+SMGAIN(I,J)*AAA(J)                            SM000120
15    CONTINUE                                                          SM000130
      XDSAT(1,1)=-1000.*XSAT(1,1)+SMCORR(1)                             SM000140
      XDSAT(2,1)=314.7*XSAT(1,1)-10**5*XSAT(3,1)+SMCORR(2)              SM000150
      XDSAT(3,1)=XSAT(2,1)-2.*ALPHA*XSAT(3,1)+SMCORR(3)                 SM000160
      RETURN                                                            SM000170
      END                                                               SM000180
```

VITA

Melville Ronald D'Mello

Candidate for the Degree of

Master of Science

Thesis: ADAPTIVE FILTERING AND SMOOTHING IN DECONVOLUTION FOR SEISMIC PROCESSES

Major Field: Electrical Engineering

Biographical:

Personal Data: Born in Goa, India, February 21, 1949, the son of Mr. and Mrs. Claro D'Mello.

Education: Graduated from St. Joseph's European High School, Bangalore, India in March, 1966; received the Bachelor of Electrical Engineering degree from Aligarh Muslim University, Aligarh, India, in September, 1972; completed requirements for the Master of Science degree at Oklahoma State University, Stillwater, Oklahoma, in December, 1974.

Professional Experience: Engineer Trainee, Kirloskar Electrical Company, Bangalore, India, from May to August, 1968; Research Assistant, School of Electrical Engineering and Center for Systems Science from January to August, 1974.