APPLICATION OF THE
EXTENDED KALMAN FILTERING TECHNIQUE
TO SHIP MANEUVERING ANALYSIS

by

JOHN G. LUNDBLAD

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE
DEGREE OF BACHELOR OF
SCIENCE
and for the
DEGREE OF MASTER OF
SCIENCE
IN OCEAN ENGINEERING
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

December, 1974
(i.e. February, 1975)

Signature of the Author:

Department of Ocean Engineering
December, 1974

Certified by:

Prof. Martin A. Abkowitz
Thesis Supervisor

Accepted by:

Prof. J. Harvey Evans
Department of Ocean Engineering
Graduate Committee

APPLICATION OF THE EXTENDED KALMAN FILTERING TECHNIQUE TO SHIP MANEUVERING ANALYSIS

by

JOHN G. LUNDBLAD

Submitted to the Department of Ocean Engineering in December, 1974, in partial fulfillment of the requirements for the degree of Bachelor of Science and for the degree of Master of Science in Ocean Engineering.

ABSTRACT

This thesis dealt with the application of a particular technique in systems identification, the Kalman statistical filter, to maneuvering analyses, determining the value of the hydrodynamic coefficients to the general equations of motion. A computer program was developed for use in this identification process. The system that the identification was applied to was the general class of surface vessels. The Mariner-class hull form was singled out for extensive analysis because of the availability of accepted values for the coefficients of these ships in the literature.

The identification process was conducted over a variety of experimental conditions. The results indicate a capability for the program to identify the desired coefficients with reasonable accuracy - within five percent of the accepted true values for the individual coefficients.

It was found that the best type of maneuver was one which generates a continuously varying input of the vessel's motion paramenters, such as the sinusoidal maneuver. Additionally, the process was shown to be able to operate on noisy data containing a large amount of scatter. The new coefficient estimates can be refiltered on additional passes by the process over the same noisy data and thereby re-evaluated and updated to a new estimate. The results of this updating seems to depend upon the accuracy of the estimates obtained from the previous pass over the noisy data.

Thesis Supervisor: Martin A. Abkowitz
Title: Professor of Naval Architecture

ACKNOWLEDGEMENT

I would like to express my thanks to Prof. Martin A. Abkowitz for his supervision and encouragement over the sometimes discouraging course of this study. It is sometimes necessary to have a shoulder to cry on when working with the computer.

I would also like to thank my none-too-nimble fingers for lasting until the end of this paper. I alone must take the blame for any mistakes in text and/or figures.

The financial support for the research came from the United States Maritime Administration - project MARAD 3-36291, M.I.T. OSP 81239. Additional support came from a Sloan Research Summer Traineeship for the summer of 1974.

All computations were done at the M.I.T. Information Processing Center (IPC).

December 18, 1974
Cambridge, Massachusetts

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (CONTINUED)

LIST OF TABLES

LIST OF TABLES (CONTINUED)

NOTATION

Lower case letters represent scalar quantities: x

Lower case leters, underlined, represent vectors: $\underline{x}$

Vector dot product: $\underline{x}^T\underline{x}$

Upper case letters represent matrices: E

The superscript T represents the matrix transpose: E^T

A bar over either a scalar or vector represents the mean value of that quantity: $\bar{x}$

Chapter I

INTRODUCTION

The naval architect must be able to predict the various motions of an ocean vehicle in order to design a vessel which can meet the required aspects of operability under which the ship will function. Without this knowledge, little can be said of the ship's capabilities with any certainty until the system is actually built. An accurate model of the vessel is thus of primary importance for design purposes.

The dynamics of an ocean vehicle can be described theoretically in terms of a general set of motion equations. The utility of this set of equations which can accurately predict the motions of a ship should be readily apparent.

The equations of motion are derived in a number of ways throughout the literature. That method which implements the vector calculus is presented later in this work. The equations' structures are such that they may be applied to many diverse systems, with the judicious choice of coefficients to the equations setting their structure to the particular system at hand. The specification of these coefficients sets the model to the system and is the problem area toward which this work is oriented.

Unfortunately, the exact numerical values of these hydrodynamic coefficients are difficult to attain. Hydrostatic and hydrodynamic theory permits specification of only a few of the parameters. Through potential theory, the acceleration derivatives can be calculated with reasonable accuracy, though they are of minor importance in terms of the general equations.

The coefficients associated with the criteria for dynamical stability in straight line motion,

$$Y_v(N_r-mx_Gu)-N_v(Y_r-mu)>0,$$

namely the velocity derivatives, are unattainable to sufficient accuracy for displacement hulls through present theory. For these and many other cases, one presently must resort to captive model tests in the towing tank. The consequence of this is the introduction of scaling effects inherent in the modeling of ship systems to the correct Froude number and, by necessity, the neglecting of Reynold's number.

There are two principle means of running model tests at present. One uses the rotating-arm mechanism. The other more popular method incorporates the planar motion mechanism. For both methods, the forces and moments exerted on the model hull forms are measured by dynamometers as the model is put through various constrained maneuvers. These forces and moments are then plotted as a function of the motion variables. The slope through the equilibrium condition, usually the origin, of this function then gives the relevant force or moment derivative. Quite obviously, this is not as accurate a procedure as one would desire because of scale effects and the difficulty of attaining certain of the non-linear coefficients.

A possible alternative to this traditional approach is derived from modern control theory. Systems identification consists of a set of theories and their applications, capable of assigning the most suitable numerical values to the variables and coefficients of the equations describing the state of the system. These equations of state consist of the motion equations as well as functions representing the measured motion responses, both assigned levels of uncertainty in their structure and recording capabilities.

One of the methods used in systems identification is statistical filtering. By taking advantage of the estimated uncertainties, or noise, as well as recorded trajectories of the ship motions, the statistical filter is capable of choosing values for these coefficients which minimize the error between the recorded and calculated state values. The specific technique of filtering used in this work is the Kalman filter, an optimum linear filter which was extended to handle non-linear systems.

The main body of this thesis consists of two parts. In Chapters II and III, the theory and equations describing both the system and the identification technique are given.

The equations of motion describing the state of the system are developed, as well as an optimum linear filter and it's non-linear extention for use in the identification.

The second part of this thesis, contained in Chapters IV,V and VI, applies the theory to a practical problem - the identification of the hydrodynamic coefficients of a Mariner-class surface ship. This type of vessel was chosen primarily for consistency with previous studies in the area. Additionally, the coefficients for this class vessel are well documented in the literature and permit a realistic appraisal of the identification results given in Chapter V. Conclusions and future considerations are stated in Chapter VI.

A listing of the general program developed for this study, as well as a description of its usage, are given in the Appendix. Also included are various and sundry items useful in this work and hopefully for any continuation of these studies.

Chapter II

SYSTEMS IDENTIFICATION

2.1 Parametric Identification

Inherent in the understanding of any dynamic system is the ability to model that system accurately through a series of differential equations. The general identification and specification of any system requires that the general structure of the system as given by this mathematical model be known, although the particular values of the parameters in the model need not be specified. For a system in this form, classical identification techniques can be employed determining the particular parametric values. This is referred to as parametric identification. The mathematical equations usually involve what are termed the state variables of the system and their derivatives, along with various constant coefficients to these variables. The coefficients set the model for the particular system or conditions under consideration. It is the values of these coefficients which need to be determined.

In ocean vehicle dynamics, the coefficients primarily relate to the hydrodynamic forces and moments exerted upon the body in response to arbitrary disturbances from equilibrium. These coefficients, which take the form of first, second and higher order force and moment derivatives, may be either completely unknown or reasonably estimated to within a degree of uncertainty. Part of the uncertainty arises from the methods involved in their estimation - model tests and full scale trials. Precise response trajectories of the vehicle motions are difficult to attain. Typically, for systems of this sort the deterministic, or precise, model is waivered for a simpler indeterminate model, where minor higher order terms as well as indeterminate noisy additions to the responses are incorporated into a single noise variable. This concept will be further developed later in this work.

2.2 System Representation

The systems analyst works is a realm defined by state variables and state-space representations of dynamic systems. A set of state variables are simply those variables which, along with a set of initial conditions, can be used to completely describe the dynamic state of a system - past, present and future. For a static system, this definition is trivial. However, since one usually deals with dynamic systems whose state is ever changing with time, the ability to so model that system is crucial to it's identification.

Often the term primary state variable is used in the literature. Its usage is somewhat arbitrary, though frequently it refers to the set of velocity parameters.[8] For this study, the primary state variables will be defined as that subset of the state variables used in the identification procedure - the measured parameters of the vehicle motions. These may include orientation as will as motion variables.

The state-space representation of the dynamic system is that set of equations incorporating the state variables which forms the model of the system.

$$\dot{\underline{x}} = \underline{f}(\underline{x},\underline{u},t)^{*}$$

$$\underline{z} = \underline{h}(\underline{x},\underline{u},t)$$

$$\underline{x}(t_o) = \underline{x}_o$$

Here the state variables, $\underline{x}$, and the input variables, $\underline{u}$, are used in the motion function, $\underline{f}$, giving the time rate of change of the state variable and the measurement function, $\underline{h}$, giving the measured output, $\underline{z}$.

Frequently, the actual structure of the system is known, except for a set of parameters or coefficients, $\underline{p}$.

[*] See Notation - the bar under a lower case letter indicates a vector quantity.

As stated earlier, this structure can be simplified by neglecting extraneous higher order terms. A modification to the equation structure involving a single uncertainty term, $\underline{w}$, compensates for this adjustment. Similarly, any uncertainty in the measurement function can be included in another uncertainty term, $\underline{v}$. The state-space representation then takes the somewhat more complex though useful form,

$$\dot{\underline{x}} = \underline{f}(\underline{x},\underline{u},\underline{p},\underline{w},t)$$

$$\underline{z} = \underline{h}(\underline{x},\underline{u},\underline{p},\underline{v},t)$$

$$\underline{x}(t_o) = \underline{x}_o$$

A significant simplification in both the structure of the above representation and eventually in the computation incorporating the model can be made through the following assumptions about the system under consideration:[16]

(i) The mathematical model and the measurement function are time invariant.

(ii) The system structural uncertainty and measurement noise are linear, and add directly to the equation of state.

(iii) The output measurements are linear functions of the state of the system, and are structurally independent.

(iv) The model coefficients are constants of the system while under observation and hence are the objective of the identification.

The first assumption is the most crucial since the loss of the assumption implies that any identification on the system is valid only over the period of observation and thus cannot be extended to the general case. The time invariant assumption, however, can be lifted if the relationship between the structure and time is known. For a quasi-static structure which is only slowly time-varying, the time invariant assumption can be made, though with some caution.

The dynamic state-space representation under these as assumptions is thereby reduced to,

$$\dot{\underline{x}} = \underline{f}(\underline{x},\underline{u},\underline{p}) + \underline{w} \qquad (2.1)$$

$$\underline{z} = H\underline{x} + \underline{v} \qquad (2.2)$$

For cases such as in this study, where the measured output is assumed to be in direct correspondence with the state of the system, the measurement function is simply the identity matrix.

2.3 Identification Methodology

Many different techniques exist for applying the system representation, eqs. (2.1) and (2.2), to the parametric identification problem. The literature abounds with procedures, many primarily oriented toward specific identification problems.(7),(10),(15) The trick then becomes the matching of the more adept procedure to the situation at hand.

2.3.1 Iterative Procedures

One of the more general methods applied to ship maneuvering, investigated by Brinati,(3) was the model reference technique, an iterative process. This procedure is one of the more conceptually simple identification techniques in current use. It can best be described as a brute force interpolation. The mathematical model is set except for one or two of the coefficients which are varied uniformly in an attempt to find those values which minimize the error function between the model and the actual data. This method was shown to work well. However, the limitations under which it must operate - limited noise levels and a minimal number of observable coefficients per trial, seem to limit its utility in extensive design applications.

2.3.2 Statistical Filtering

An alternative approach and one which has received a great deal of attention in the past decade is that of statistical filtering. Part of this popularity and utility comes from the fact that it provides an optimum use of all available information about the system. This includes statistical estimates of both the noise in the system and its state.

Much of the initial theoretical work on statistical estimation and filtering was performed by Weiner,[18] in the 1930's. It's applicability to systems analysis was developed by Kalman [11],[12] in the 1960's. He showed that an optimum linear filter, based on the covariance matrix of the state estimation errors can lead to a minimum error in the final estimate of the state of the system.

There are two major disadvantages to the Kalman filtering technique, neither of which had a very serious effect upon this study.

(i) The filter has a linear derivation and therefore is valid only for linear systems.

(ii) It requires a reasonable, but not necessarily accurate, estimate of the system and noise parameters before their identification may proceed.

For ocean vehicle systems, the second problem is inconsequential. Reasonable estimates can be made from vehicle coefficients which are presently in the literature, or have been attained from model tests of the class of vehicle desired by traditional methods. For other types of systems, where this estimation problem might become significant, on-line identification techniques are being developed (14) which can work in conjunction with the Kalman filter, but which initially need no precise estimate of the state or noise characteristics.

The restriction to linear systems is also of little concern for those cases where ship maneuvering can be limited to small linear disturbances. This permits the use of simplified linear models available in the literature.(6)

For the general case, however, the restriction of linear modeling is not acceptable. The methods used by Brock(4) and described here lift that restriction and permit the extension of the Kalman filter to the non-linear case - a development which may not be theoretically strong but which works quite well all the same.

2.4 The Kalman Filtering Technique

The Kalman filter was the identification technique used in this work. It is a statistical filter for use in the presence of uncorrelated white noise. Through use of the Kalman filter, the identification problem is reduced to a state estimation of the dynamic system. The filter development follows for a linear system which in turn is followed by it's logical extension to the non-linear case.

2.4.1 Derivation of the Optimum Linear Filter

Earlier in this chapter, the equations for the state-space representation of a dynamic system, eqs. (2.1) and (2.2), were developed,

$$\dot{\underline{x}} = \underline{f}(\underline{x},\underline{u},\underline{p}) + \underline{w}$$

$$\underline{z} = H\underline{x} + \underline{v}$$

where a linear relationship between state and output has been assumed. For simplification in that which follows, the noise factors, $\underline{w}$ and $\underline{v}$, will be discarded for the time being. The dynamic system representation is therefore reduced to,

$$\dot{\underline{x}} = \underline{f}(\underline{x},\underline{u},\underline{p}) \qquad (2.3)$$

$$\underline{z} = H\underline{x} \qquad (2.4)$$

Given a system defined by these state and measured output functions, one desires to estimate the true state of the system at some time t. If numerous measurements of the system are taken, a realistic assumption for most physical systems is that the values attained will approximate a Gaussian distribution. Therefore, the best estimate of the state of the system, $\hat{\underline{x}}$, will be that which approaches the mean, $\bar{\underline{x}}$, of the system,

$$\hat{\underline{x}} = \bar{\underline{x}} = \int_{-\infty}^{\infty} \underline{x}P(\underline{x}|\underline{z})d\underline{x}$$

Any error in this estimate can be defined by

$$\underline{e} = \hat{\underline{x}} - \underline{x}$$

and the covariance matrix of these errors by

$$\begin{aligned} E &= \overline{(\hat{\underline{x}}-\underline{x})(\hat{\underline{x}}-\underline{x})}^{T} \\ &= \overline{\underline{e}\ \underline{e}}^{T} \qquad (2.5) \end{aligned}$$

One of the characteristics of a Gaussian or normal distribution is the fact that the mean of $\underline{x}$ specifies the maximum of its probability density function (PDF),

$$P(\bar{\underline{x}}) = \max\left[P(\underline{x})\right]$$

Therefore, a proper method for determining the optimal estimate of $\underline{x}$ is one which would determine that $\underline{x}$ which maximizes it's PDF. The standard form of the Gaussian PDF for a random variable, y, is given by

$$P(y) = \frac{1}{\sqrt{2\pi\,\sigma}}\; e^{-(y_o-\bar{y})^2/2\sigma^2} \qquad (-\infty \leq y \leq \infty\ )$$

This can be extended to describe a system of n state variables as

$$P(\underline{x}) = \frac{1}{(2\pi)^{n/2}\ E^{1/2}}\; e^{-(\hat{\underline{x}}-\underline{x})(\hat{\underline{x}}-\underline{x})^T/2E}$$

where E is the variance, defined as the square of the standard deviation, σ^2. The problem is then one of maximizing $P(\underline{x})$, under the constraint imposed by the measured output

$$\underline{z} = H\underline{x}$$

Since $\log\left[P(\underline{x})\right]$ attains a maximum for the same value of $\underline{x}$ as $P(\underline{x})$, the problem can be rewritten, using Lagrangian multipliers, as the maximization of $F(\underline{x})$, where

$$\begin{aligned} F(\underline{x}) &= \log\left[P(\underline{x})\right] + \underline{\lambda}^T\ (\underline{z}-H\underline{x}) \\ &= \log\left[\frac{1}{(2\pi)^{n/2}\ E^{1/2}}\right] - (\hat{\underline{x}}-\underline{x})(\hat{\underline{x}}-\underline{x})^T/2E + \underline{\lambda}^T\ (\underline{z}-H\underline{x}) \end{aligned}$$

The variation of $F(\underline{x})$ with $\underline{x}$ is given by

$$\frac{dF(\underline{x})}{d\underline{x}} = (\hat{\underline{x}}-\underline{x})^T E^{-1} - \underline{\lambda}^T H$$

Maximization implies

$$\frac{dF(\underline{x})}{d\underline{x}} = 0$$

or,

$$(\hat{\underline{x}}-\underline{x})^T E^{-1} = \underline{\lambda}^T H$$

Taking the transpose of both sides yields

$$(\hat{\underline{x}}-\underline{x})(E^{-1})^T = \underline{\lambda} H^T$$

but from symmetry,

$$(\hat{\underline{x}}-\underline{x}) = \underline{\lambda} EH^T$$

$$\underline{x} = \hat{\underline{x}} - \underline{\lambda} EH^T \tag{2.6}$$

From the measurement function,

$$\underline{z} = H\underline{x}$$
$$= H(\hat{\underline{x}} - \underline{\lambda}\ EH^T)$$

or,

$$\underline{\lambda} = (H\hat{\underline{x}} - \underline{z})/HEH^T \qquad (2.7)$$

Substitution of eq. (2.7) into eq. (2.6) gives

$$\underline{x} = \hat{\underline{x}} + \left[(\underline{z}-H\hat{\underline{x}})/HEH^T\right]\ EH^T$$
$$= \hat{\underline{x}} + EH^T\left[HEH^T\right]^{-1}(\underline{z}-H\hat{\underline{x}}) \qquad (2.8)$$

This then is that value of $\underline{x}$ which maximizes the PDF for the function and, by definition, is the optimum estimate of the state of the system at time t.

It can be shown that if one includes the measurement noise, $\underline{v}$, in the eq.(2.4), having its specified character, then the more general form of the state estimate is

$$\hat{\underline{x}}' = \hat{\underline{x}} + EH^T\left[HEH^T + R\right]^{-1}(\underline{z}-H\hat{\underline{x}}) \qquad (2.9)$$

where,

$$R = \overline{(\hat{\underline{v}}-\underline{v})(\hat{\underline{v}}-\underline{v})^T} \qquad (2.10)$$

In order to determine the new covariance matrix for this optimal estimate, $\underline{\hat{x}}'$, one need only substract $\underline{\hat{x}}$ from eq.(2.9), arriving at a value for $\underline{e}$. From the relation

$$E = \overline{\underline{e}\ \underline{e}^T}$$

one arrives at the updated matrix

$$E' = E - EH^T (HEH^T + R)^{-1} HE \qquad (2.11)$$

Eqs.(2.9) and (2.11) can be somewhat simplified and possibly more easily understood by defining the new variable K, the gain.

$$K = EH^T (HEH^T + R)^{-1} \qquad (2.12)$$

Then, eqs.(2.9) and (2.11) reduce to

$$\underline{\hat{x}}' = \underline{\hat{x}} + K(\underline{z} - H\underline{\hat{x}}) \qquad (2.13)$$

$$E' = E - KHE \qquad (2.14)$$

At this point, the process noise in the state equation can be re-introduced into eq.(2.3) as given in eq.(2.1).

$$\dot{\underline{x}} = \underline{f}(\underline{x}, \underline{u}, \underline{p}) + \underline{w}$$

The optimum estimate for $\dot{\underline{x}}$ will then take the form

$$\hat{\dot{\underline{x}}} = \underline{f}(\hat{\underline{x}},\underline{u},\underline{p}) \tag{2.15}$$

since the process noise is defined as being of zero-mean. This equation can be rewritten as

$$\hat{\dot{\underline{x}}} = B\hat{\underline{x}} \tag{2.16}$$

where B is a matrix of coefficients acting upon the state variables.

$$B = \frac{\partial \underline{f}(\hat{\underline{x}},\underline{u},\underline{p})}{\partial \underline{x}} \tag{2.17}$$

The error in the state estimate is seen to be

$$\begin{aligned} \dot{\underline{e}} &= \hat{\dot{\underline{x}}} - \dot{\underline{x}} \\ &= B\hat{\underline{x}} - (B\underline{x} + \underline{w}) \end{aligned}$$

The time derivative of the error covariance matrix, $\dot{E}$, is then written as

$$\begin{aligned} E &= \frac{d}{dt}\left(\overline{\underline{e}\,\underline{e}^T}\right) \\ &= \dot{\underline{e}}\,\underline{e}^T + \underline{e}\,\dot{\underline{e}}^T \end{aligned}$$

or finally,

$$\dot{E} = BE + EB^T + (\overline{\underline{w}\ \underline{w}^T})$$

The process noise covariance matrix, Q, is defined as

$$Q = \overline{\underline{w}\ \underline{w}^T} \qquad (2.18)$$

The time rate of change of the error covariance matrix can therefore be converted to the form

$$\dot{E} = BE + EB^T + Q \qquad (2.19)$$

This then is the controlling equation in the variation of the covariance matrix in conjunction with the measurement function over time.

The estimation problem can thus be completely described by eqs. (2.13),(2.14),(2.15) and (2.19). When a measurement of the system is taken, eq. (2.13) determines the optimal estimate, $\hat{\underline{x}}'$, of the state variables at that time. This it does by maximizing the system's PDF based on the previous estimate of the system, $\hat{\underline{x}}$, and the present measured output, $\underline{z}$. The error covariance matrix is similarly determined by eq. (2.14) as a function of the matrix calculated for the previous measurement. Eqs. (2.15) and (2.19) are integrated to update the state and error covariance matrices before the next measurement. These new values of the state and covariance matrices before the next measurement. These new values of the state and covariance matrices are then used to again optimize the system's estimates and the process repeats itself.

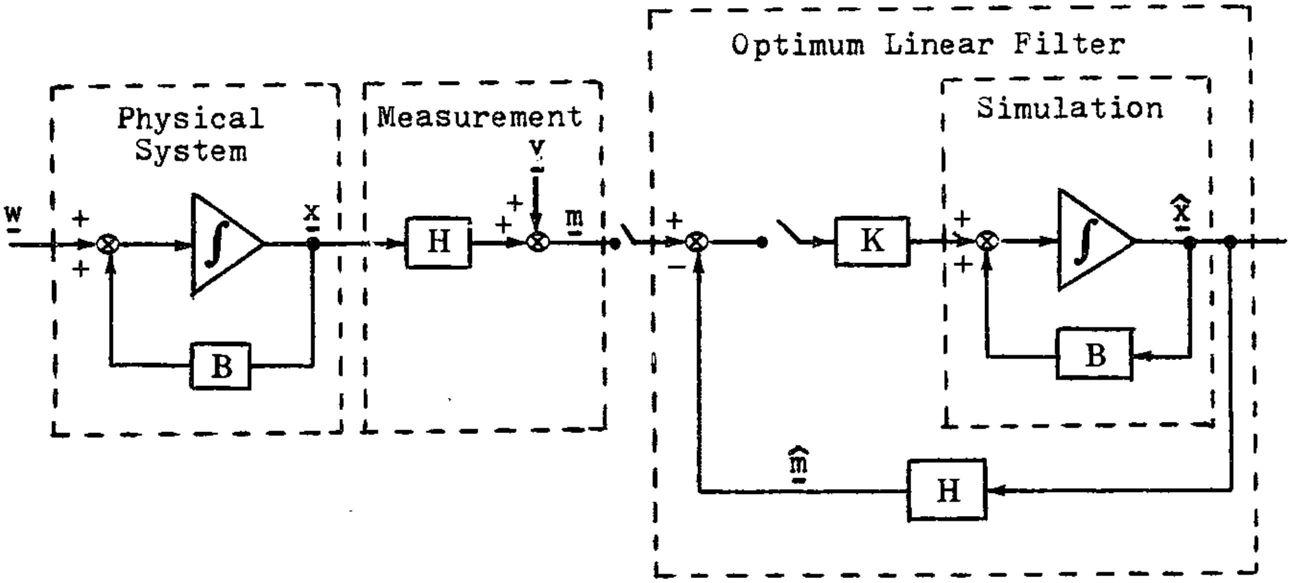


Fig. 2-1. Block Diagram of Optimum Linear Filter[4]

2.4.2 Non-linear Extension

The derivation of the equations relating to the statistical Kalman filter was done for a linear system,

$$\dot{\underline{x}} = \underline{f}(\underline{x},\underline{u},\underline{p},\underline{w})$$

$$\underline{z} = \underline{h}(\underline{x},\underline{u},\underline{p},\underline{v})$$

and indeed, can be applied only to those systems whose dynamics can be considered linear functions of $\underline{x}$, $\underline{u}$, $\underline{p}$ and $\underline{v}$ or $\underline{w}$. However, it is possible to extend these equations to the cases where the dynamics of the system must be described by non-linear functions. This linearization of non-linear equations gives one the versatility of applying the filtering technique to a wider class of physical systems.

Assuming eqs. (2.1) and (2.2) are continuous and differentiable over the region of interest, the state variables can be written as

$$\underline{x} = \underline{x}_o + \delta\underline{x}$$

$$\underline{z} = \underline{z}_o + \delta\underline{z}$$

and eqs. (2.1) and (2.2) can be expanded in a Taylor series about the initial values,

$$\delta\dot{\underline{x}} = \frac{\partial \underline{f}_o}{\partial \underline{x}} \delta\underline{x} + \frac{\partial \underline{f}_o}{\partial \underline{w}} \delta\underline{w} + \cdots$$

$$\delta\underline{z} = \frac{\partial \underline{h}_o}{\partial \underline{x}} \delta\underline{x} + \frac{\partial \underline{h}_o}{\partial \underline{v}} \delta\underline{v} + \cdots$$

The values of $\dot{\underline{x}}$ and $\underline{z}$ can be assumed to be close to the initial values so that $\delta\dot{\underline{x}}$ and $\delta\underline{z}$ can be considered linear functions of $\delta\underline{x}$. This assumes that $\delta\underline{w}$ and $\delta\underline{v}$ are equal to $\underline{w}$ and $\underline{v}$ respectively, which follows from their being uncorrelated.

The linear derivation described earlier can then be used to get the optimum estimate of $\delta\underline{x}$. The equations for the non-linear filter are thus of the same form as those developed for the linear case, with minor redefinitions of the matrices involved.

$$\dot{\hat{\underline{x}}} = \underline{f}(\hat{\underline{x}},\underline{u},\underline{p}) \qquad (2.20)$$

$$\hat{\underline{x}} = \hat{\underline{x}}' + E'H^T (HE'H^T + R_n)^{-1} (\underline{z}-H\hat{\underline{x}}) \qquad (2.21)$$

$$\dot{E} = BE + EB^T + Q_n \qquad (2.22)$$

$$E = E' - E'H^T (HE'H^T + R_n)^{-1} HE' \qquad (2.23)$$

where,

$$B = \frac{\partial \underline{f}}{\partial \underline{x}} (\hat{\underline{x}},\underline{u},\underline{p})$$

$$H = \frac{\partial \underline{h}}{\partial \underline{x}} (\underline{x}, \underline{u}, \underline{p})$$

$$R_n = \frac{\partial \underline{h}}{\partial \underline{v}} \; R \; \frac{\partial \underline{h}}{\partial \underline{v}}^T$$

$$Q_n = \frac{\partial \underline{f}}{\partial \underline{w}} \; Q \; \frac{\partial \underline{f}}{\partial \underline{w}}^T$$

For the assumptions under which eqs. (2.1) and (2.2) were developed, namely additive linear noise and a linear measurement function, the noise covariance matrices, R_n and Q_n, reduce to the form of those found in the linear model, R and Q.

Chapter III

THE SYSTEM

The system under consideration in this work is a Mariner-class vessel operating in unrestricted waters. Under equilibrium conditions, it is assumed to be moving at a constant forward speed. We are interested in determining the effect that various deviations from the equilibrium condition will have upon the motions of the ship. To do so requires a model which accurately portrays the vessel under any and all conditions in which it may be found.

3.1 The Mathematical Model

The best method of simulating a dynamic system is to mathematically recreate it through a series of differential equations which can accurately describe its motions. The mathematical model used in this work, developed by Abkowitz,(1) considers the vessel as a rigid body of constant mass with a stationary center of gravity. Alternative models have been developed in the literature for similar systems, as well as those special cases not included in this model.(8)

A body moving in a fluid medium is considered to be a acted upon by a system of forces and moments. These can be resolved by considering two sets of forces and moments, each of which is equivalent to the other at equilibrium. First, one can consider the body's rigid structure and the forces and moments due to it's mass and the motions of that mass - velocities, accelerations, moments of inertia, et cetera, independent of the body's shape. Secondly, one can consider the forces and moments arising from the medium itself, termed the hydrodynamic forces and moments. These act upon and are initiated by the body's shape - the dynamics of its interaction with the fluid medium. The subsequent motions of the body in the fluid through dynamic equilibrium arise from equating the two systems of forces.

3.1.1 Rigid Body Dynamics in Six Degrees of Freedom

The dynamics of the origonal body structure are ultimately derived from an understanding of Newtonian mechanics.

$$\underline{F} = \frac{d}{dt}\,(\overline{\text{Momentum}}) = X\hat{i} + Y\hat{j} + Z\hat{k}$$

$$\underline{M} = \frac{d}{dt}\,(\overline{\text{Angular Momentum}}) = K\hat{i} + M\hat{j} + N\hat{k}$$

One can consider a rectangular coordinate system with arbitrary origin not necessarily at the center of gravity of the system, but parallel to the principle axes of inertia through the center of gravity. The system can thus be shown in the form of fig. (3-1),

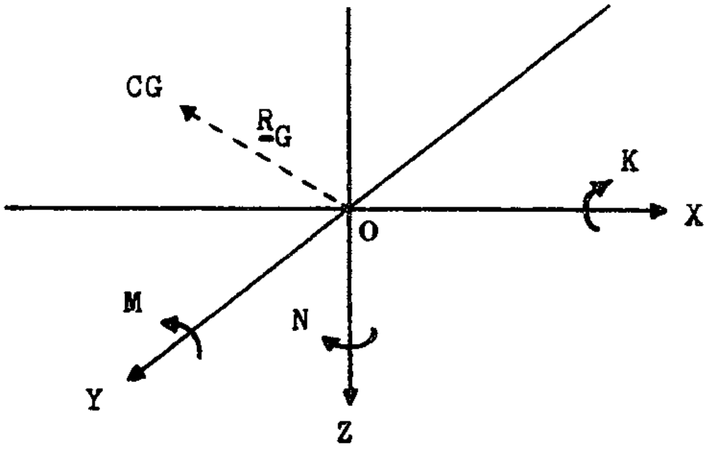


Fig. 3-1. Rectangular Coordinate System

where the relevant forces and moments are as indicated. $\underline{R}_G$ is the vector displacement of the origin from the center of gravity.

Using this notation, the force equation becomes,

$$\underline{F} = m \frac{d}{dt} (\underline{U} + \underline{\Omega} \times \underline{R}_G)$$

under the constant mass assumption, where

$$\underline{\Omega} = p\hat{i} + q\hat{j} + r\hat{k}$$

and is defined as the angular velocity of the center of gravity about the chosen origin. Expansion of the equation yields the force components along the principle axes.

$$X = m\left[\dot{u} + qw - rv - x_G(q^2+r^2) + y_G(pq-\dot{r}) + z_G(pr+\dot{q})\right] \quad (3.1)$$

$$Y = m\left[\dot{v} + ru - pw - y_G(r^2+p^2) + z_G(qr-\dot{p}) + x_G(qp+\dot{r})\right] \quad (3.2)$$

$$Z = m\left[\dot{w} + pv - qu - z_G(p^2+q^2) + x_G(rp-\dot{q}) + y_G(rq+\dot{p})\right] \quad (3.3)$$

In a similar manner, the moment equation can be shown to be equivalent to

$$\underline{M} = \underline{M}_G + \underline{R}_G \times \underline{F}$$

where

$$\underline{M}_G = \frac{d}{dt}\left(I_x p\,\hat{i} + I_y q\,\hat{j} + I_z r\,\hat{k}\right)$$

The moment components about the principle axes are then given by the equations

$$K = I_x\dot{p} + (I_z-I_y)qr + m\left[y_G(\dot{w} + pv - qu) - z_G(\dot{v} + ru - pw) + x_Gy_G(pr-\dot{q}) - x_Gz_G(pq+\dot{r}) + y_Gz_G(r^2-q^2)\right] \quad (3.4)$$

$$M = I_y\dot{q} + (I_x - I_z)rp + m\Big[z_G(\dot{u} + qw - rv) - x_G(\dot{w} + pv - qu)$$
$$+ y_G z_G(qp - \dot{r}) - y_G x_G(qr + \dot{p}) + x_G z_G(p^2 - r^2)\Big] \qquad (3.5)$$

$$N = I_z\dot{r} + (I_y - I_x)pq + m\Big[x_G(\dot{v} + ru - pw) - y_G(\dot{u} + qw - rv)$$
$$+ z_G x_G(rq - \dot{p}) - z_G y_G(rp + \dot{q}) + y_G x_G(q^2 - p^2)\Big] \qquad (3.6)$$

The rigid body structure of the dynamic model can therefore be summarized by eqs. (3.1) through (3.6) for a vessel of constant mass and arbitrary origin of its coordinate system.

3.1.2 The Hydrodynamic Forces and Moments

The dynamic forces and moments acting upon the body in a fluid are functions of the body itself, its motions and the medium through which it passes. For a given body operating in a particular fluid, these functions are dependent only upon the body's movement.

$$\left.\begin{matrix} \underline{F} \\ \underline{M} \end{matrix}\right\} = g(\underline{R}_o, \underline{U}, \dot{\underline{U}}, \underline{\Omega}, \dot{\underline{\Omega}}, \text{effector controls})$$

For this work, it is assumed that the body is operating in unrestricted waters, therefore negating any effect of the orientation parameter, $\underline{R}_o$, on the dynamics. The only effector force and moment contributions will come from the rudder deflection, δ, neglecting higher order terms such as $\dot{\delta}$ and δ.

$$\left.\begin{matrix}\underline{F}\\ \\ \underline{M}\end{matrix}\right\} = g(\underline{U},\ \dot{\underline{U}},\ \underline{\Omega},\ \dot{\underline{\Omega}},\ \delta) \qquad (3.7)$$

The function in eq. (3.7) can be expanded through a Taylor series expansion, assuming the function is continuous and analytic over the region of interest. This assumption is valid under normal operating conditions.

The multi-dimensional expansion is done about a nominal condition, in this case the equilibrium condition of constant forward motion, in terms of the individual components of the functional quantities. Looking at the force equation,

$$\underline{F} = F(u,\ v,\ w,\ p,\ q,\ r,\ \dot{u},\ \dot{v},\ \dot{w},\ \dot{p},\ \dot{q},\ \dot{r},\ \delta)$$

the expansion becomes a lengthy equation of the form

$$\underline{F} = \underline{F}_o + \left(\frac{\partial \underline{F}}{\partial u}\right)_o (\Delta u_o) + \cdots + \frac{1}{2}\left[\left(\frac{\partial^2 \underline{F}}{\partial u^2}\right)_o (\Delta u_o)^2 + \cdots + \left(\frac{\partial^2 \underline{F}}{\partial u\,\partial v}\right)_o (\Delta u_o)(\Delta v_o) + \cdots\right] + \frac{1}{6}\left[\left(\frac{\partial^3 \underline{F}}{\partial u^3}\right)_o (\Delta u_o)^3 + \cdots\right] + \text{higher order terms} \qquad (3.8)$$

It will be seen that many of the terms in eq. (3.8) can be eliminated by employing the proper assumptions.

Two simplifications are now in order. First, standard shorthand notation will be used throught for the force and moment derivatives,

$$\left(\frac{\partial \underline{F}}{\partial x_i}\right)_o = \underline{F}_{x_i}$$

Secondly,

$$(x_i)_o = x_i - (x_i)_o$$

Under the equilibrium conditions of straight ahead motion at constant speed,

$$(x_i)_o = 0$$

and

$$(x_i)_o = x_i$$

for all x_i except u, which does have a non-zero equilibrium value. Therefore, the hydrodynamic forces and moments can be portrayed as

$$\underline{F} = \underline{F}_o + \underline{F}_u(\Delta u) + \cdots + \frac{1}{2}\left[\underline{F}_{uu}(\Delta u)^2 + \cdots + \underline{F}_{uv}(\Delta u)v + \cdots\right] + \frac{1}{6}\left[\underline{F}_{uuu}(\Delta u)^3 + \cdots\right] + \text{higher order terms} \qquad (3.9)$$

3.1.3 Equations of State for a Body Moving in Three Degrees of Freedom

The rigid body structure of the dynamic model for a body of constant mass and arbitrary center of gravity, moving in six degrees of freedom was given in eqs. (3.1) through (3.6). Similarly, the hydrodynamic forces and moments acting upon a body with six degrees of freedom are derived in the form of eq. (3.9) from the Taylor series expansion. With the system under dynamic equilibrium, the hydrodynamic forces and moments are solely responsible for the forces and moments acting on the rigid body, and hence the motions of the body through the fluid. Therefore, the two systems of equations can be equated to determine the resultant motions in six degrees of freedom.

$$\left\{\begin{array}{l}\text{Hydrodynamic and}\\ \text{hydrostatic forces}\\ \text{and moments}\end{array}\right\} = \left\{\begin{array}{l}\text{Inertial}\\ \text{reaction}\\ \text{responses}\end{array}\right\} \qquad (3.10)$$

This general case does not always apply to every system, however. For this study, the body was constrained to only three degrees of freedom - a surface ship operating solely within the horizontal plane. Additionally, it was assumed that these horizontal maneuvers do not excite rolling motions. This assumption applied to the Mariner-class hull form is adequately valid under normal operating conditions. It will also be assumed that y_G is located along the longitudinal plane of symmetry.

Under these conditions,

$$y_G = \phi = \varphi = w = p = q = \dot{w} = \dot{p} = \dot{q} = Z = K = M = 0$$

for any time t. The equations used in eq. (3.10) can therefore be reduced from the general case to that for only three degrees of freedom, with substantial simplification in structure.

The rigid body forces and moments in the horizontal plane, excluding roll, are

$$X = m\,(\dot{u} - rv - x_G r^2) \tag{3.11}$$

$$Y = m\,(\dot{v} + ru + x_G r) \tag{3.12}$$

$$N = I_z \dot{r} + m x_G(\dot{v} + ru) \tag{3.13}$$

while the hydrodynamic forces and moments for the same case are given by the Taylor series expansion of

$$X = X(u, v, r, \dot{u}, \dot{v}, \dot{r}, \delta) \tag{3.14}$$

$$Y = Y(u,\ v,\ r,\ \dot{u},\ \dot{v},\ \dot{r},\ \delta) \qquad (3.15)$$

$$N = N(u,\ v,\ r,\ \dot{u},\ \dot{v},\ \dot{r},\ \delta) \qquad (3.16)$$

Brinati(3) showed that numerous additional terms in the expansion of the hydrodynamic structure could be dropped by additional assumptions. These included cross-coupling between the acceleration and velocity terms, negligible second and higher order terms, and negligible contributions from symmetry considerations. Applying these assumptions, which are quite valid, one arrives at the following form for the hydrodynamic structure.

$$\begin{aligned} X = X_o &+ X_u(\Delta u) + X_{\dot{u}}\dot{u} + \frac{1}{2}\Big[X_{uu}(\Delta u)^2 + X_{vv}v^2 + X_{rr}r^2 \\ &+ X_{\delta\delta}\delta^2\Big] + X_{vr}vr + X_{r\delta}r\delta + X_{v\delta}v\delta + \frac{1}{6}X_{uuu}(\Delta u)^3 \\ &+ \frac{1}{2}\Big[X_{vvu}v^2(\Delta u) + X_{rru}r^2(\Delta u) + X_{\delta\delta u}\delta^2(\Delta u)\Big] \\ &+ X_{vru}vr(\Delta u) + X_{v\delta u}v\delta(\Delta u) + X_{r\delta u}r\delta(\Delta u) \qquad (3.17) \end{aligned}$$

$$\begin{aligned} Y = Y_o &+ Y_v v + Y_\delta \delta + Y_{vu}v(\Delta u) + Y_r r + Y_{ru}r(\Delta u) + Y_{\delta u}\delta(\Delta u) \\ &+ \frac{1}{6}\Big[Y_{vvv}v^3 + Y_{rrr}r^3 + Y_{\delta\delta\delta}\delta^3\Big] + \frac{1}{2}\Big[Y_{vrr}vr^2 \\ &+ Y_{v\delta\delta}v\delta^2 + Y_{vuu}v(\Delta u)^2 + Y_{rvv}rv^2 + Y_{r\delta\delta}r\delta^2 \end{aligned}$$

$$+ Y_{ruu}r(\Delta u)^2 + Y_{\delta vv}\delta v^2 + Y_{\delta rr}\delta r^2 + Y_{\delta uu}\delta(\Delta u)^2\Big]$$

$$+ Y_{vr\delta}vr\delta \qquad (3.18)$$

$$N = N_o + N_v v + N_r r + N_\delta \delta + N_{vu}v(\Delta u) + N_{ru}r(\Delta u)$$

$$+ N_{\delta u}\delta(\Delta u) + \frac{1}{6}\left[N_{vvv}v^3 + N_{rrr}r^3 + N_{\delta\delta\delta}\delta^3\right]$$

$$+ \frac{1}{2}\Big[N_{vrr}vr^2 + N_{v\delta\delta}v\delta^2 + N_{vuu}v(\Delta u)^2 + N_{rvv}rv^2$$

$$+ N_{r\delta\delta}r\delta^2 + N_{ruu}r(\Delta u)^2 + N_{\delta vv}\delta v^2 + N_{\delta rr}\delta r^2$$

$$+ N_{\delta uu}\delta(\Delta u)^2\Big] + N_{vr\delta}vr\delta \qquad (3.19)$$

A further simplification can be made in eqs. (3.17), (3.18) and (3.19), by dropping those terms which individually have negligible effect upon the eventual motions of the ship, without unduly altering the model. This reasoning is somewhat similar to the dropping of the fourth and higher order terms from the expansion. For all cases, dropped terms, if small enough, are in actuality compensated for in the indeterminate model by the uncertainty term. Possibly the most important reason for this simplification is not so much in reducing the equation structure, but rather in what will be shown to be a detrimental effect by these minor terms on the identification process itself. Brinati conducted an examination of these equations and was able to separate a number of minor terms. The results of his work were not verified for this study because of time constraints, but were used in the equation development.

Equating the resulting hydrodynamic structure with that developed for a rigid body under constraint of maneuvering in the horizontal plane, and solving for the acceleration terms, leads to the following set of state equations.

$$\dot{u} = f_1/(m - X_{\dot{u}}) \tag{3.20}$$

$$\dot{v} = \frac{(I_z - N_{\dot{r}})f_2 - (mx_G - Y_{\dot{r}})f_3}{f_4} \tag{3.21}$$

$$\dot{r} = \frac{(m - Y_{\dot{v}})f_3 - (mx_G - N_{\dot{v}})f_2}{f_4} \tag{3.22}$$

where,

$$f_1 = X_o + X_u(\Delta u) + \frac{1}{2} X_{uu}(\Delta u)^2 + \frac{1}{6} X_{uuu}(\Delta u)^3 + \frac{1}{2} X_{vv}v^2 + (\frac{1}{2} X_{rr} + mx_G)r^2 + \frac{1}{2} X_{\delta\delta}\delta^2 + (X_{vr} + m)vr + X_{v\delta}v\delta$$

$$f_2 = Y_o + Y_v v + (Y_r - mu)r + Y_\delta \delta + \frac{1}{6} Y_{\delta\delta\delta}\delta^3 + \frac{1}{2} Y_{rvv}rv^2 + \frac{1}{2} Y_{\delta vv}\delta v^2$$

$$f_3 = N_o + N_v v + (N_r - mx_G u)r + N_\delta \delta + \frac{1}{6} N_{\delta\delta\delta}\delta^3 + \frac{1}{2} N_{rvv} rv^2 + \frac{1}{2} N_{\delta vv}\delta v^2$$

$$f_4 = (m - Y_{\dot{v}})(I_z - N_{\dot{r}}) - (mx_G - N_{\dot{v}})(mx_G - Y_{\dot{r}})$$

Eqs. (3.19), (3.20) and (3.21) then, describe the motions of the vessel in the horizontal plane with three degrees of freedom. Together, they form a set of state variables which, along with the initial conditions of the problem, completely describe the past, present and future motions for any given input. This dynamic model is complete, except for these initial conditions, and forms a sufficient set of equations for the work of this study.

3.2 Sea-Trial Maneuvers

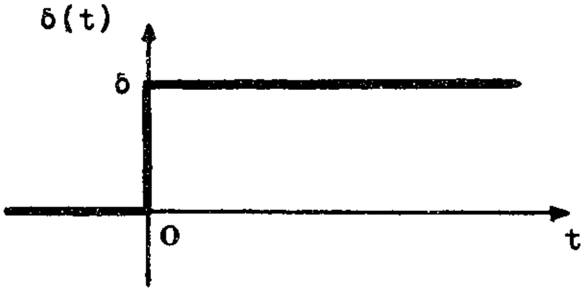
Of primary importance in any maneuvering trial is the proper planning for that trial. Especially in the type of identification process proposed here, it is necessary to know what effect different maneuvers will have on the ability to identify the different hydrodynamic coefficients. It is desirable to know what measurements to make, which motions to record under the various conditions of gradual accelerations, sudden changes in velocity, steady state velocities or the like.

For these reasons, several different maneuvers were used in this study and are described here. As stated earlier, the only control surface considered in this work was the rudder and it's deflections. This was incorporated into the general equations as the variable δ.

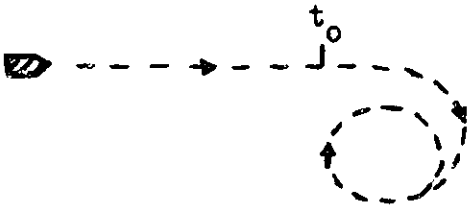
3.2.1 Single-step Rudder Deflection

Formally, the single-step rudder deflection can be de described as a step function of the form

$$\delta(t) = \begin{cases} 0, & t < 0 \\ \delta, & t \geq 0 \end{cases}$$



Graphically, this corresponds to the case where the vessel goes into a constant turn in the steady state.

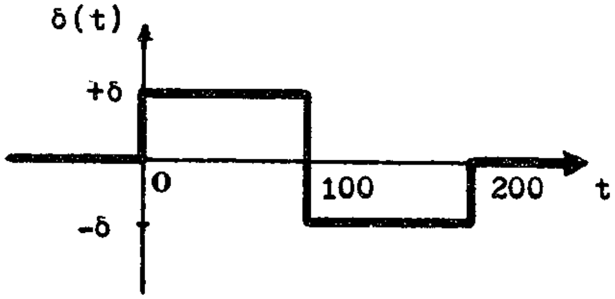


Previous to the rudder deflection, all velocities and accelerations are zero except u. However, u is an unmeasurable motion by conventional methods and for the most part will be neglected. As the effect of the rudder deflection is felt by the vessel, all velocity and acceleration terms become non-zero until the ship reaches it's steady state turning radius. At this point the acceleration terms $\dot{v}$ and $\dot{r}$ have non-zero values for the remainder of the trial. For large rudder deflections, a distinct speed reduction occurs due to the tight turn.$^{(2)}$

3.2.2 Zig-zag Rudder Deflection

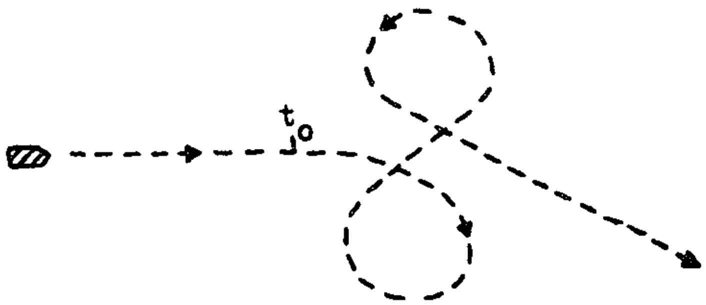
Again, this maneuver can also be formally described by the step function,

$$\delta(t) = \begin{cases} 0, & t < 0,\ t \geq 200 \\ \delta, & 0 \leq t < 100 \\ -\delta, & 100 \leq t < 200 \end{cases}$$



This definition is perhaps not completely realistic to real-life situations at sea since no time-lag is incorporated into it's structure. However, for the purposes of this study, it is an acceptable representation.

Pictorially,

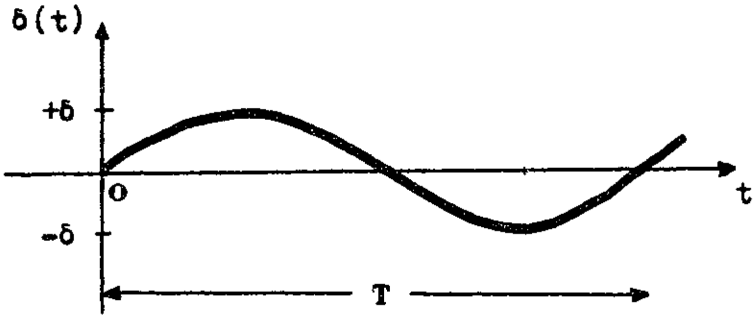


the vessel is seen to go into one steady turn followed by the opposite steady turn and finally achieving the new equilibrium state of constant forward speed, though not necessarily at the original orientation. Essentially there are three steady state conditions during the maneuver. The situation for the motion parameters in the steady state turns is identical to that discussed for the step deflection. For the straight ahead motion at constant speed, both the velocity and acceleration terms are reduced to zero.

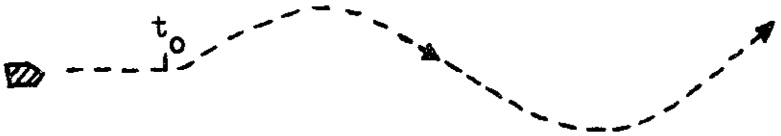
3.2.3 Sinusoidal Rudder Deflections

This deflection is simply a sinusoidal motion of the rudder with a maximum displacement corresponding to δ and with the specified period, T.

$$\delta(t) = \delta \sin \omega t$$



The motions of the ship will follow the rudder deflection in a sinusoidal manner, with a slight time lag.



The important point for this maneuver is that the velocity and the acceleration terms are in a constant state of flux. At no point during the trial, after t_o, is the steady state condition established for any of the motion parameters. This gives the identification process a continually changing system upon which to operate.

It is this maneuver which was used for much of this study.

Chapter IV

APPLICATION OF THE EXTENDED KALMAN FILTER TO THE IDENTIFICATION PROBLEM

4.1 Compatibility Between the System and the Filter

In Chapter II, the concept of parametric identification was developed as a means of system identification applicable to those dynamic systems whose general structure was known, but whose specific parameters or coefficients were unknown. The structure of the indeterminate model was given as

$$\dot{\underline{x}} = \underline{f}(\underline{x}, \underline{u}, \underline{p}) + \underline{w} \tag{4.1}$$

$$\underline{z} = H\underline{x} + \underline{v} \tag{4.2}$$

where the imprecision of the model is represented by the uncertainty terms, $\underline{w}$ and $\underline{v}$.

The utility of statistical filtering as a method for solving the identification problem was shown and the equations for the optimum linear filter, as developed by Kalman, were given. These equations were extended to the non-linear form.

$$\dot{\hat{\underline{x}}} = \underline{f}(\hat{\underline{x}}, \underline{u}, \underline{p}) \tag{4.3}$$

$$\hat{\underline{x}} = \hat{\underline{x}}' + E'H^T\,(HE'H^T + R_n)^{-1}(\underline{z} - H\hat{\underline{x}}) \qquad (4.4)$$

$$\dot{E} = BE + EB^T + Q_n \qquad (4.5)$$

$$E = E' - E'H^T\,(HE'H^T + R_n)^{-1}\,HE \qquad (4.6)$$

The general model for an ocean vehicle was developed in Chapter III for a surface ship moving in the horizontal plane without roll.

$$\dot{\underline{x}} = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} f_1/(m - X_{\dot{u}}) \\ \dfrac{(I_z - N_{\dot{r}})f_2 - (mx_G - Y_{\dot{r}})f_3}{f_4} \\ \dfrac{(m - Y_{\dot{v}})f_3 - (mx_G - N_{\dot{v}})f_2}{f_4} \end{bmatrix} \qquad \begin{matrix} (4.7a) \\ (4.7b) \\ (4.7c) \end{matrix}$$

where f_1, f_2, f_3 and f_4 are as before. This gives the general structure of the system (a vehicle under maneuvering) and identifies that system except for the hydrodynamic coefficients. The initial conditions which specify the dynamic condition of the system are those for which the equations were developed, namely straight ahead motion under constant speed.

By combining these steps, the ability to use the extended Kalman filter to identify the hydrodynamic coefficients of the equations of motion can be attained. First, however, some minor changes must be made in the above development.

The state vector must be extended to the augmented state,

$$x = (x_1, x_2, \cdots, x_n, p_1, p_2, \cdots, p_m)$$

by the inclusion of the unknown coefficients. In this manner the Kalman filtering technique which identifies, or more correctly estimates, the state of the system can be used to identify the coefficients under observation by including them in the state vector.

The input function of the ocean vehicle is known and is frequently a function of time. This removes the time invariant assumption in the structure of $\underline{f}$. However, since the function of time, $\underline{u}(t)$, is known, it can be incorporated into the model and as stated earlier, is an acceptable alteration.

These two redefinitions reduce eq. (4.1) to

$$\dot{\underline{x}} = \underline{f}(\underline{x},t) + \underline{w} = B(t)\underline{x} + \underline{w} \qquad (4.8)$$

which is compatible to that used in the definition of the extended Kalman filter.

Finally, from the assumption of linear additive noise contributions to the equations of state and measurement function, the process and measurement noise covariance matrices, Q_n and R_n, reduce to that of the linear filter.

$$Q_n = Q$$

$$R_n = R$$

From these alterations, the system, a Mariner-class surface vessel in maneuvering, and the identification technique, statistical filtering, may be applied to the problem at hand. This does not necessarily imply that statistical filtering in particular or system identification in general can lead to the complete specification of the system structure. It does mean, however, that one is now in a position to apply the system to the technique and see wether or not an identification capability does indeed exist.

4.2 Noise Generation and Incorporation

Up to this point, very little has been said concerning the noise contributions, $\underline{w}$ and $\underline{v}$, to the system structure and measurement function. In Chapter II, one of the disadvantages of the Kalman filter was stated to be that the statistical characteristics of the uncertainty terms had to be specified. This is true, though the estimation of these characteristics for many cases is relatively straightforward.

The process noise, $\underline{w}$, expresses the uncertainty in the structure of the mathematical model. This arises from the truncation of the Taylor series expansion for the hydrodynamic structure of all terms over third order. It may also incorporate any unknown contributions from the input function, $\underline{u}(t)$, or any spurious deviations from the assumptions used in developing the equations of state. Excitations from the enviornment are also included in the process noise.

The output uncertainly is expressed as a measurement noise, $\underline{v}$. As for the process noise, this term includes all unknown structural aspects of the measurement function, H. For this study, since the measured output is assumed to directly correspond to the actual state of the system, any deviations from this linear correspondence are represented by $\underline{v}$.

The noise contributions are both treated in a similar manner and are felt to be similar, statistically. Both are assumed to be stochastic processes, with uncorrelated, zero-mean, Gaussian white noise. The Gaussian probability density function (PDF) for the uncertainty values is a reasonable assumption for most physical systems. From the central limit theorem of general probability theory, it can be shown that the sum of a large number of independent effects has a Gaussian distribution, regardless of the statistical properties of the small effects individually. The uncertainty terms can therefore be considered Gaussian in nature and treated as random variables for simple incorporation into the model structure.

These assumptions can be summarized as follows.

$$\underline{w} = E[\underline{w}] = 0$$

$$\underline{v} = E[\underline{v}] = 0$$

$$Q = E[\underline{w}\,\underline{w}^T]$$

$$R = E[\underline{v}\,\underline{v}^T]$$

$$E\left[\underline{w}\,\underline{v}^T\right] = 0$$

To be consistent with the work done by others in this same area at MIT,(3),(8),(16) the noise used for the generation of simulated noisy data measurements to be used in the identification will be defined as a percentage noise value. For the generation of noise with the desired Gaussian distribution (see Appendix A - Program Desription), it is necessary to specify both the desired mean and the standard deviation. The mean is assumed to be zero by convention. The standard deviation of the distribution will be defined as that specified percentage of the maximum of the function under consideration. Therefore,

$$\sigma_{\underline{w}} = \sqrt{Q_n}$$

is equal to that percentage of the maximum value of the function $\underline{\dot{x}}$, while

$$\sigma_{\underline{v}} = \sqrt{R_n}$$

is that percentage of the maximum value of the measured output, $\underline{x}$. The maximum values are attained from the trajectory of the deterministic model ($\underline{w} = \underline{v} = 0$) over the interval of observation.

This definition has several unfortunate aspects which must be kept in mind. In particular, it should be apparent that problems in specification will arise for those maneuvers where the function is not uniform in magnitude, but rather peaks for a short period during the trial. For these cases, the noise present will be specified by the percentage of that maximum value, but will be added to function values substantially lower in magnitude over the majority of the period.

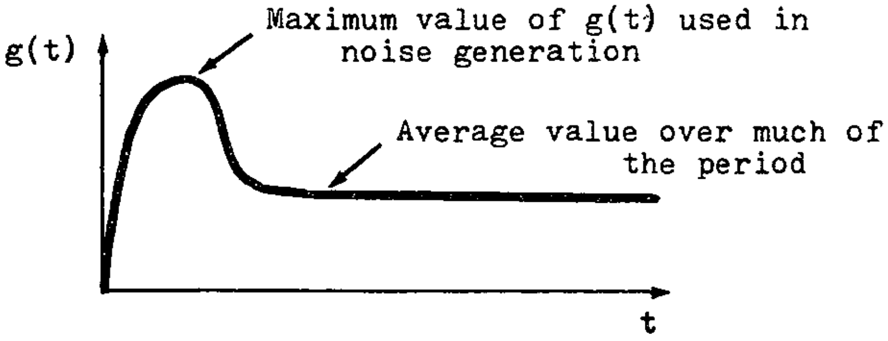

g(t)
Maximum value of g(t) used in noise generation
Average value over much of the period
t

Fig. 4-1. Imprecision in Noise Definition

In these cases, therefore, the actual noise present during the identification process will be noticably larger than that specified, possibly by an order of magnitude or more.

4.3 The Identification Process

Much of the actual implementation of the theory developed up to this point is described in the Appendix. However, a summary of the various steps leading to the results ofthe next chapter should be of value at this time.

The state-space representation of the system was developed in Chapter III and specialized for a surface ship moving with three degrees of freedom. Theoretically, this leads to at least nine primary state variables (x_o, y_o, ψ_o, u, v, r, $\dot{u}$, $\dot{v}$, $\dot{r}$, ...) which could be measured during a particular maneuver. In reality this is not the case. Some of these variables can not be recorded at all during full-scale trials, while others require special devises not normally available on-board ship during maneuvers. Those variables which could be readily recorded by traditional methods are yaw velocity, r, and angle, ψ, along with the sway acceleration $\dot{v}$, actually $(\dot{v} + r u_o)$. A general program dealing with nine primary state variables was developed, but most of the identification studies deal with these three variables - r, ψ, and $\dot{v}$. Indirectly, the sway velocity, v, was also incorporated. Use of an integrating accelerometer on-board ship, while not permitting direct measurement of v, would give an indirect record of the sway velocity which could be used in the identification by the filter. Thus, for this study the state vector is defined as,

$$\underset{=}{x} = \begin{bmatrix} v \\ r \\ \psi \\ \dot{v} \end{bmatrix}$$

There may be a problem arising from the dependence of v upon the measured acceleration, $\dot{v}$, though in this study it did not become readily apparent.

Equations for each of the primary state variables can be derived from eq. (4.7) - $\dot{v}$ directly from the definition and v,r and ψ from the integration of their respective equations.

$$v = \int_{t_1}^{t_2} \dot{v}\, dt$$

$$r = \int_{t_1}^{t_2} \dot{r}\, dt$$

$$\psi = \int_{t_1}^{t_2} r\, dt$$

Using these state variables, the remaining steps in the identification process can be summarized as in Table 4-1.

For this study, the noisy data had to be generated within the program itself before it could be processed.

* * *

STEP 1: Generate the noisy sea-trial data,

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

$$\underline{z} = H\underline{x} + \underline{v}$$

STEP 2: Propagate the estimated state and error covariance matrices over one time step,

$$\dot{\hat{\underline{x}}} = B(t)\hat{\underline{x}}$$

$$\underline{z}_m = H\hat{\underline{x}}$$

$$\dot{E} = BE + EB^T + Q$$

STEP 3: Calculate the Kalman filter gain matrix,

$$K = EH^T (HEH^T + R)^{-1}$$

STEP 4: Update the estimated state and error covariance matrices at the end of the step,

$$\hat{\underline{x}}' = \hat{\underline{x}} + K (\underline{z} - \underline{z}_m)$$

$$E' = E - KHE$$

STEP 5: Set the updated state and error covariance matrices as the new estimates and repeat STEPS 2 through 5 until the end of the identification process.

Table 4-1 Summary of the Computation Steps

Chapter V

RESULTS OF THE IDENTIFICATION PROCESS

The application of the theory developed in Chapters II and III to the problem of identifying the coefficients for a Mariner-class surface vessel was shown in the last chapter. The equations for the extended Kalman filter were given in Table 4-1 as steps of a procedure for their computation. What remains is for the theory to be tested on the system and see if indeed this technique for systems identification is valid under the given conditions.

A program was deloped to do these tasks using the MIT IBM 370/168. It is listed in the Appendix, along with a detailed description of it's use and function. The reader is referred to this section for those details. However, a few brief points are in order at this time.

An attempt was made to keep the program as general as possible, requiring only a change in the input to enact wholesale alterations in the structure of the identification. For the most part, this was accomplished. There remains some card shifting to enable the user to select different measured state variables, but for choices in coefficients identified, trial types and lengths and the like, only a variation in the data deck is necessary.

There are a multitude of different control combinations which may be employed in the identification process. This is simultaneously a blessing and a curse. The results depend on the judicious choice of trial conditions. The identification of a certain coefficient may be attained with good results under one set of conditions, but with totally negative results under different circumstances. Fortunately, there are so many conditions under which a trial may be run that it is possible to mix and match until a certain combination gives the desired results.

This plethora of choices makes a final verdict on the identification difficult. Most of the work done for this project was devoted to developing the computer program. Very little actual analysis could be completed. Therefore, to say that, based upon the sample of results given here, identification is either good or bad is unrealistic. The best set of operating conditions were not examined.

The raison d'être for this chapter is simply to show the possibilities and capabilities of the program, no more. Trends may be observed. Hopefully, these will be of help later in developing a detailed analysis useful in designing full-scale trials. Again, however, it should be emphasized that these results are neither representative nor optimal. They are simply the results for the given set of conditions under which the system operated.

What are these trial conditions which may be systematically altered? A partial list, some of which will later be illustrated, include:

i) Variation in the uncertainty or noise term, measurement and/or process noise

ii) Variation in what the filter is told concerning the amount of noise (noise exaggeration)

iii) Estimates of the coefficient values and the standard deviations of those estimates

iv) The number of coefficients processed at one time, as well as their combination

v) The number and characteristics of the measured primary state variables

vi) The type and magnitude of maneuver

vii) The length of the period of observation

viii) The time increment between observations

ix) Second and third generation identifications

x) Flexibility in using results of sets of maneuvers, each identifying those coefficients for which it is best suited.

Obviously, the choices are many. To best observe the capabilities of this technique, all the above possibilities should be explored systematically. Neither time nor finances permitted doing so for this study. Consequently, it was decided to simply show some results and possibly indicate some trends and/or difficulties in using the program.

An attempt was made to keep all conditions, save one, equal during each trial. In each case, only four coefficients were studied.

$$Y_v \ (6), \ (Y_r - mu) \ (7), \ N_v \ (12), \ (N_r - mx_G u) \ (13)$$

These were chosen because they represent the coefficients used in determining the criteria for dynamic stability in straight line motion (see Chapter I). They are four of the most critical coefficients. Being able to successfully identify these would be a measure of the overall success of this technique.

Additionally, for each case the noise level was kept constant for both the process and measurement noise. When reference is made to 5% noise level, both measurement and process noise are at 5%.

It was felt that identification to within 10% of the true value could be classified as successful. This specification was used in andysing the following results.

5.1 A Typical Identification (Control)

The best results were found to occur when a sinusoidal rudder deflection of 10^o was used in conjunction with the four measured primary state variable - v, r, ψ and $\dot{v}$. This run is an illustration of these results. The conditions under which it ran, namely 5% noise, 376 second trial, no noise exaggeration and four primary state variables, are the controlling cases for the runs which follow. For some however, more than one condition had to be varied. In most runs, the time increment was one second. In this case, the increment had to be increased to two seconds. For smaller increments the filter became unstable.

As can be seen, the results are really quite impressive, with all identifications to within 2% of the accepted true value, except for Y_v at a respectable 6% .

```
*************************************************************
*                                                           *
*  PARAMETRIC IDENTIFICATION - EXTENDED KALMAN FILTER       *
*                                                           *
*************************************************************


        SYSTEM:  MARINER-CLASS SURFACE VESSEL


        MANEUVER:  ZIG-ZAG, WITH SINUSOIDAL RUDDER
                   DEFLECTIONS OF PERIOD 200.0 SECONDS AND
                   MAXIMUM DEFLECTIONS OF 10.0 DEGREES


        NOISE LEVEL:  MEASUREMENT NOISE -   5%

                      PROCESS NOISE -   5%


        EXAGGERATED NOISE FACTOR:    1.0


        TRIAL PERIOD:   376 SECONDS


        TIME STEP:  2.0 SECONDS


        NUMBER OF PRIMARY STATE VARIABLES:   4


        NUMBER OF COEFFICIENTS IDENTIFIED:    4


                   (NON-LINEAR MODEL)
```

Table 5-1a Conditions for the Typical Identification

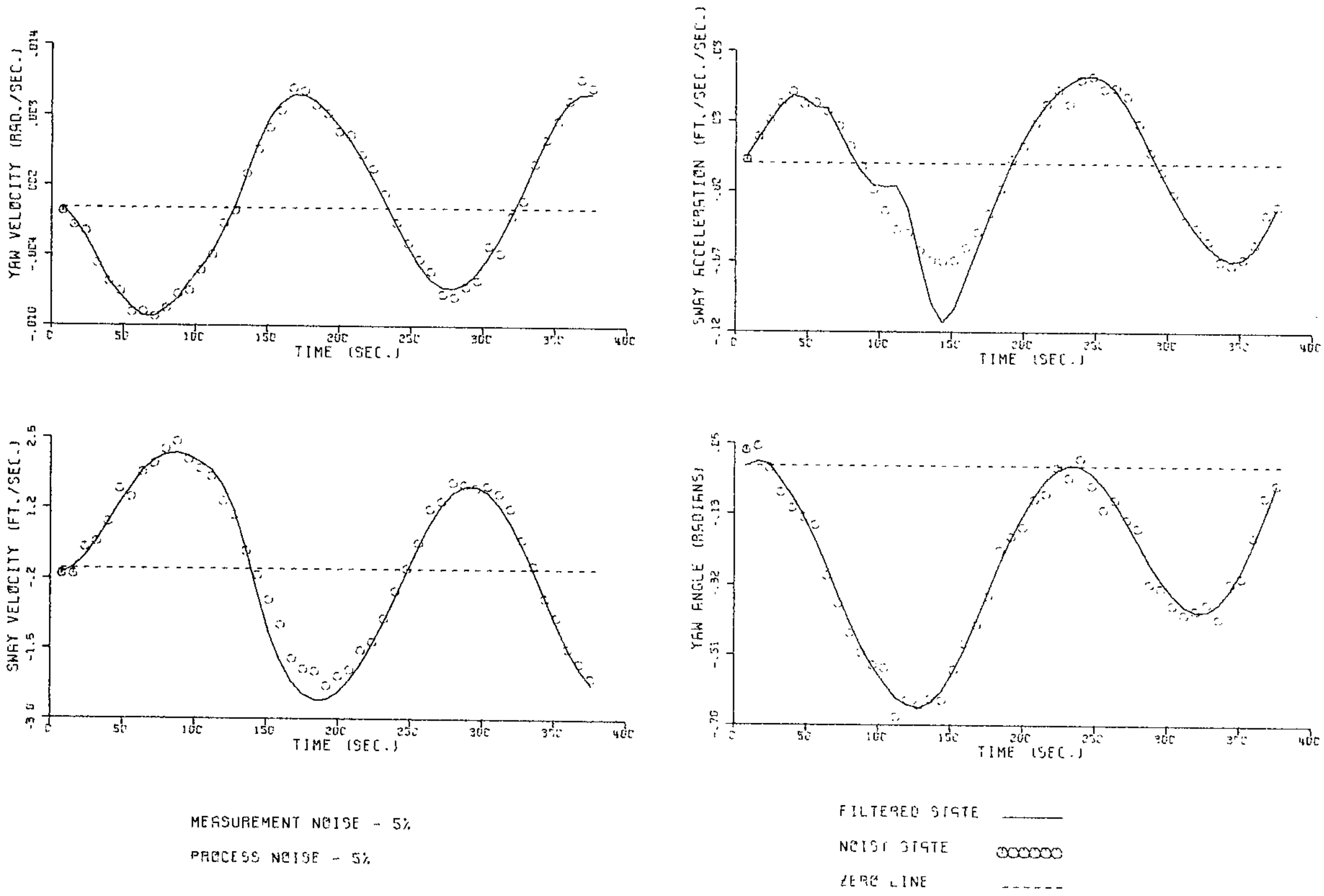


Fig. 5-1a Filtered States from Typical Identification

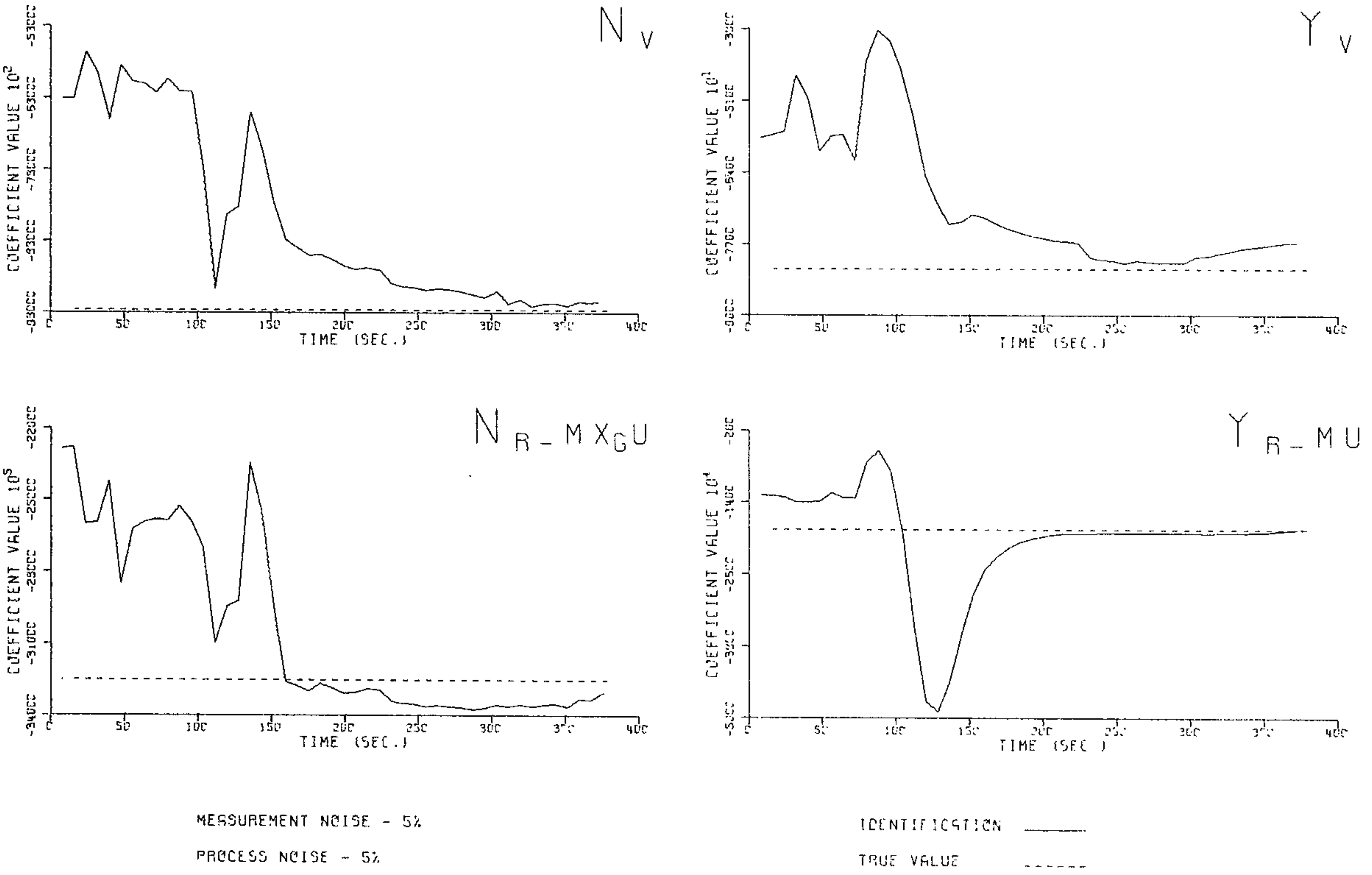


Fig. 5-1b Coefficient Identification for the Typical Case

```
NP =  13      TRUE VALUE  =   -0.32510E+10
SV  =   -0.22752D+10  + OR -    0.97581D+09
FV  =   -0.33036E+10  + OR -    0.67285E+08

IDENTIFICATION WITHIN  1.62% OF THE TRUE VALUE.
```

$(N_r - mx_G u)$

```
NP =  12      TRUE VALUE  =   -0.97735E+07
SV  =   -0.68414D+07  + OR -    0.29321D+07
FV  =   -0.96408E+07  + OR -    0.27363E+06

IDENTIFICATION WITHIN  1.36% OF THE TRUE VALUE.
```

N_v

```
NP =   7      TRUE VALUE  =   -0.18508E+08
SV  =   -0.12955D+08  + OR -    0.55525D+07
FV  =   -0.18497E+08  + OR -    0.26818E+06

IDENTIFICATION WITHIN  0.06% OF THE TRUE VALUE.
```

$(Y_r - mu)$

```
NP =   6      TRUE VALUE  =   -0.81515E+05
SV  =   -0.57060D+05  + OR -    0.24454D+05
FV  =   -0.76791E+05  + OR -    0.13076E+04

IDENTIFICATION WITHIN  5.79% OF THE TRUE VALUE.
```

Y_v

Table 5-1b Coefficient Identification for the Typical Case

5.2 Variation in the Maximum Rudder Deflection

The magnitude of the maximum rudder deflection for the same sinusoidal zig-zag maneuver was increased to a strongly non-linear 35°. The time step had to be increased to two seconds as before, for filter stability. The identification was successful for N_v and $(N_r - mx_G u)$ only. The remaining two coefficients were not determined. It should be noted that the identification process zeroed in on a value for each coefficient, even though for Y_v and $(Y_r - mu)$ that value was incorrect. This consequently was shown as undeserved confidence in the values as shown by the final standard deviations.

The motion trajectories are seen to be well defined after filtering. This, in conjunction with the poor identification of Y_v and $(Y_r - mu)$ implies that these two coefficients do not overly affect the ship's motions. The result of this is the inability of the filter to operate successfully under these conditions.

```
*****************************************************************
*                                                               *
*  PARAMETRIC IDENTIFICATION - EXTENDED KALMAN FILTER           *
*                                                               *
*****************************************************************


        SYSTEM:  MARINER-CLASS SURFACE VESSEL


        MANEUVER:  ZIG-ZAG, WITH SINUSOIDAL RUDDER
                   DEFLECTIONS OF PERIOD 200.0 SECONDS AND
                   MAXIMUM DEFLECTIONS OF 35.0 DEGREES


        NOISE LEVEL:  MEASUREMENT NOISE -   5%

                      PROCESS NOISE -   5%


        EXAGGERATED NOISE FACTOR:    1.0


        TRIAL PERIOD:   376 SECONDS


        TIME STEP:  2.0 SECONDS


        NUMBER OF PRIMARY STATE VARIABLES:   4


        NUMBER OF COEFFICIENTS IDENTIFIED:    4


                  (NON-LINEAR MODEL)
```

Table 5-2a Conditions for the Variation in Rudder Deflection

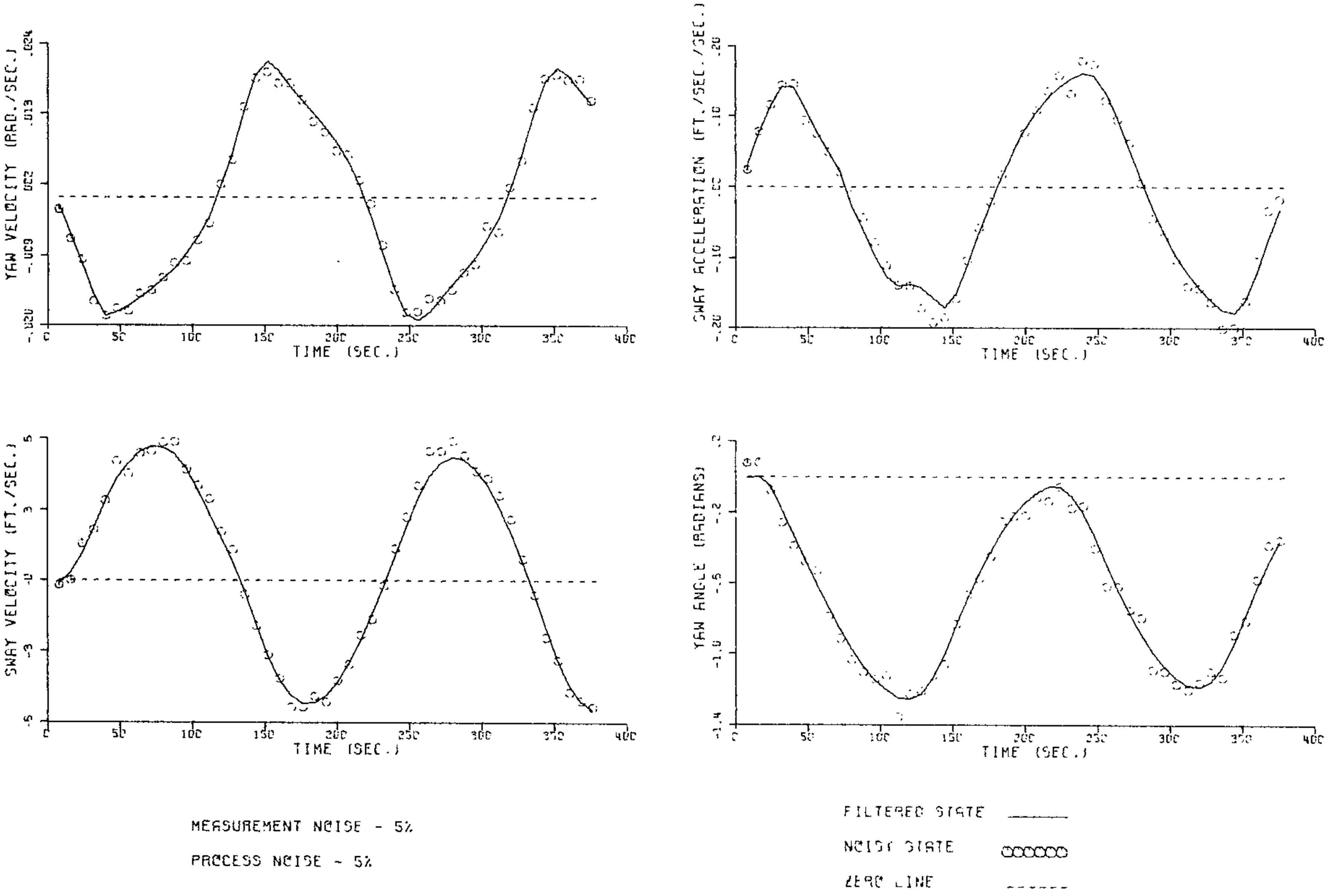


Fig. 5-2a Filtered States - Variation in Maximum Rudder Deflections

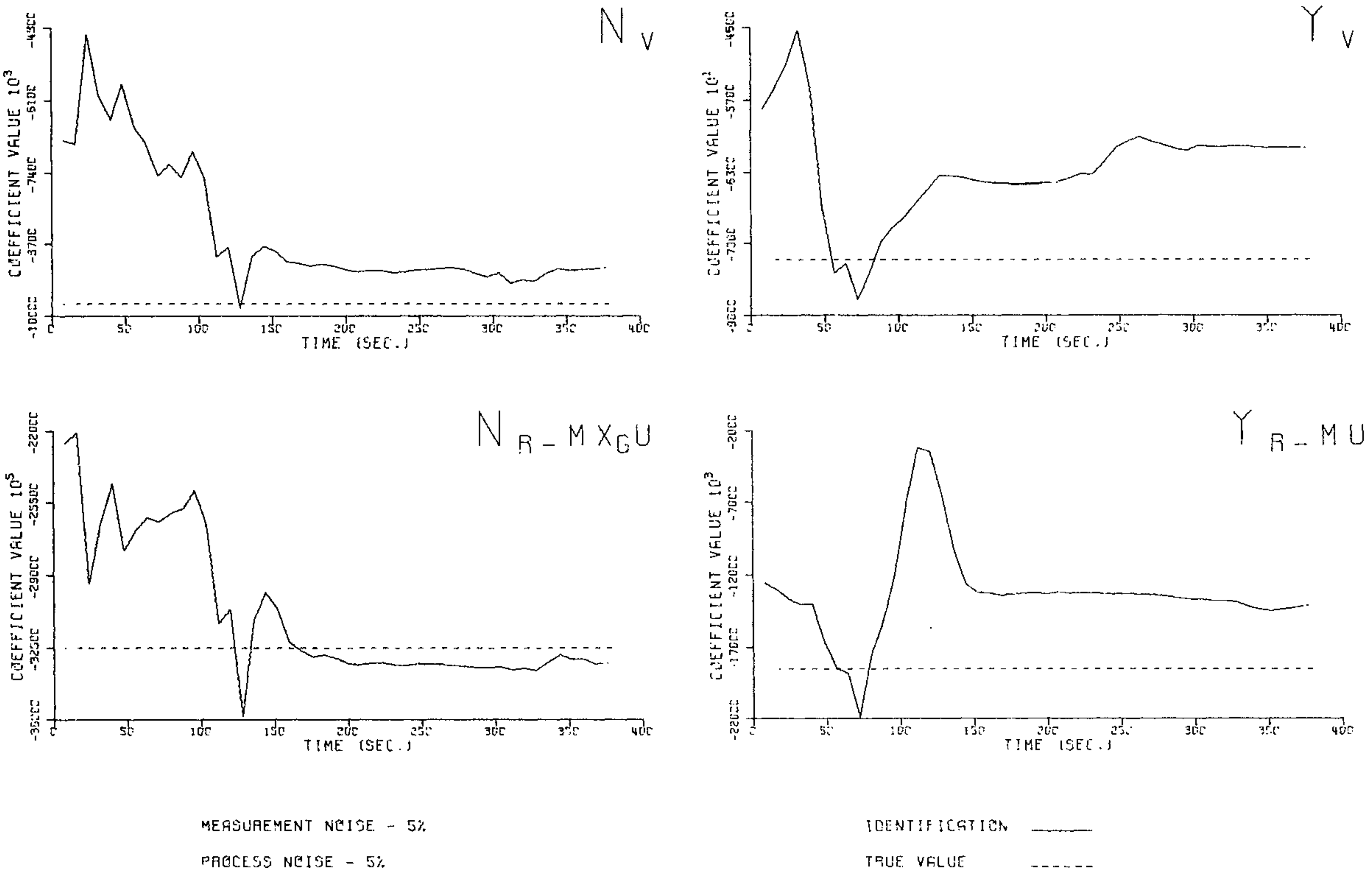


Fig. 5-2b Coefficients - Variation in the Maximum Rudder Deflection

```
NP =  13      TRUE  VALUE   =   -0.32510E+10
SV   =    -0.22752D+10   + OR -     0.97581D+09        (Nr - mxGu)
FV   =    -0.33229E+10   + OR -     0.60134E+08

IDENTIFICATION WITHIN  2.21% OF THE TRUE VALUE.
```

```
NP =  12      TRUE  VALUE   =   -0.97735E+07
SV   =    -0.68414D+07   + OR -     0.29321D+07        Nv
FV   =    -0.91301E+07   + OR -     0.22005E+06

IDENTIFICATION WITHIN  6.58% OF THE TRUE VALUE.
```

```
NP =   7      TRUE  VALUE   =   -0.18508E+08
SV   =    -0.12955D+08   + OR -     0.55525D+07        (Yr - mu)
FV   =    -0.14113E+08   + OR -     0.23044E+06

IDENTIFICATION WITHIN 23.75% OF THE TRUE VALUE.
```

```
NP =   6      TRUE  VALUE   =   -0.81515E+05
SV   =    -0.57060D+05   + OR -     0.24454D+05        Yv
FV   =    -0.64299E+05   + OR -     0.92124E+03

IDENTIFICATION WITHIN 21.12% OF THE TRUE VALUE.
```

Table 5-2b Coefficient Identification for the Variation in Maximum Rudder Deflection

5.3 Variation in the Trial Length

One of the important aspects of any maneuver is the length of the trial over which observations are taken. This case is an investigation of that variable condition. For this run, the two second time increment was continued for a 752 second period. Twice as many observations and therefore twice as many revaluations were made as before. The results are essentially the same as before for N_v and $(N_r - mx_G u)$, but are substantially worse for the remaining coefficients.

This trial is basically two trials, one after the other. It probably could be considered similar to a second generation identification. After 376 seconds, the filter works on the new estimates with the newly derived error covariance matrix. The trial does not change, being a sinusoidal function of time. However, looking at the values at $t = 376$, they do not appear to correspond with those given in the previous trial over 376 seconds under the same conditions.

The lengthening of the trial is felt to be more appropriate to those maneuvers such as the step zig-zag trial where the maneuver over the second half of the trial is different from that of the first half. In this way, two aspects could be studied, the large variation identification followed by the steady state identification. Results under these conditions may be more useful.

```
*********************************************************
*                                                       *
*  PARAMETRIC IDENTIFICATION - EXTENDED KALMAN FILTER   *
*                                                       *
*********************************************************


        SYSTEM:  MARINER-CLASS SURFACE VESSEL


        MANEUVER:  ZIG-ZAG, WITH SINUSOIDAL RUDDER
                   DEFLECTIONS OF PERIOD 200.0 SECONDS AND
                   MAXIMUM DEFLECTIONS OF 10.0 DEGREES


        NOISE LEVEL:  MEASUREMENT NOISE -   5%

                      PROCESS NOISE -   5%


        EXAGGERATED NOISE FACTOR:    1.0


        TRIAL PERIOD:   752 SECONDS


        TIME STEP:  2.0 SECONDS


        NUMBER OF PRIMARY STATE VARIABLES:   4


        NUMBER OF COEFFICIENTS IDENTIFIED:    4


                    (NON-LINEAR MODEL)
```

Table 5-3a Conditions for the Variation in Trial Length

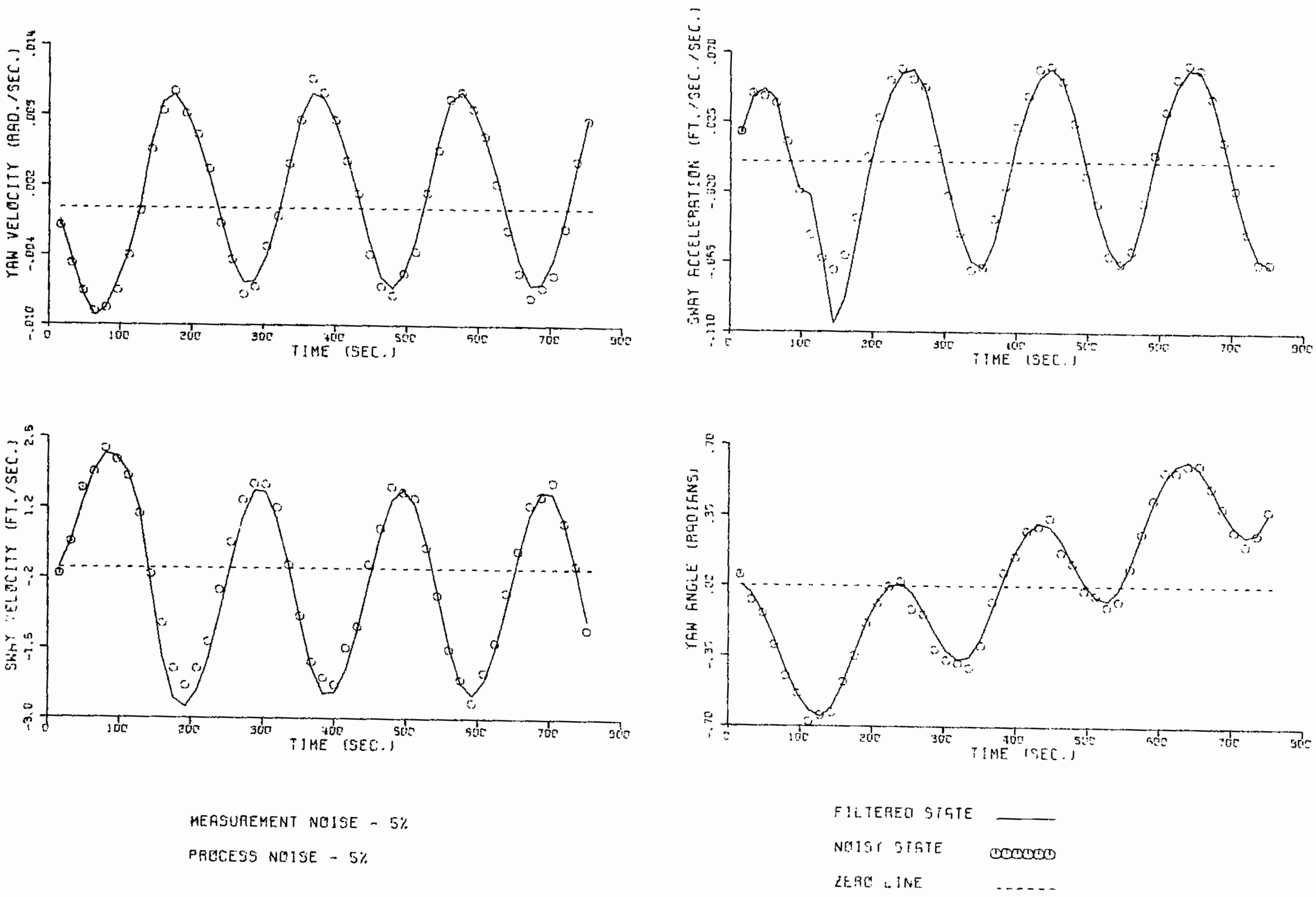


Fig. 5-3a Filtered States - Variation in Trial Length

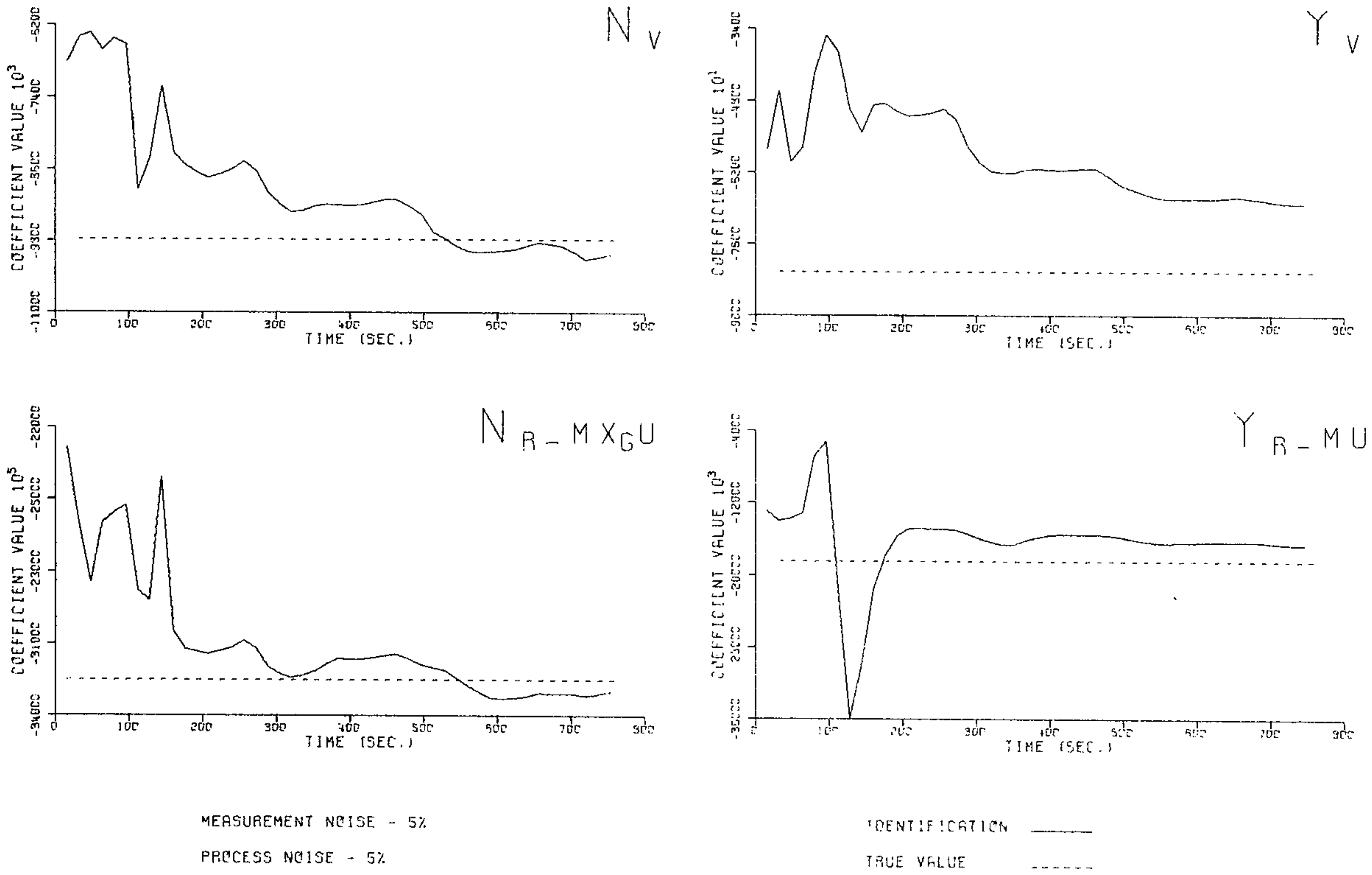


Fig. 5-3b Coefficients - Variation in Trial Length

```
NP =  13      TRUE VALLE   =   -C.3251CE+1C

SV   =    -0.22752C+10   + CR -     C.97581C+C9

FV   =    -C.3259CE+1C   + OR -     C.5C136E+08

ICENTIFICATION WITFIN  1.48% CF TFE TRUE VALUE.
```

$(N_r - mx_G u)$

```
NP =  12      TRUE VALLE   =   -C.97735E+07

SV   =    -0.68414C+07   + CR -     0.29321C+07

FV   =    -C.1CC28E+C8   + OR -     C.21182E+C6

ICENTIFICATION WITHIN  2.60% CF TFE TRUE VALUE.
```

N_v

```
NP =   7      TRUE VALLE   =   -C.185C8E+C8

SV   =    -0.12955C+08   + CR -     0.55525C+07

FV   =    -C.16625E+C8   + OR -     C.25196E+C6

ICENTIFICATION WITHIN 10.18% CF TFE TRUE VALUE.
```

$(Y_r - mu)$

```
NP =   6      TRUE VALLE   =   -0.81515E+C5

SV   =    -0.57060C+05   + OR -     0.24454C+05

FV   =    -C.68C98E+C5   + OR -     C.11C08E+C4

ICENTIFICATION WITHIN 16.46% CF TFE TRUE VALUE.
```

Y_v

Table 5-3b Coefficient Identification for the Variation in Trial Length

5.4 Variation in the Time Increment

It had originally been felt that an increase in accuracy by the filter would occur directly in proportion to the number of observations used in the process. The result illustrated by this case was therefore somewhat surprizing. Only 94 observations on the system were made, with a full four seconds between points. Yet, the results are on the average as good or better than those using twice the number of observations. Two to three percent is excellent and was observed for each coefficient except N_v, which was even better at less than one percent off the accepted true value. Even more encouraging is the appearance of the filtered states for each of the primary state variables. The plot for $\dot{v}$ does not have the characteristic deviation around 100 seconds which was seen on several other trials.

An added benefit from this observation is the substantial savings in computer time and therefore dollars. Only a quarter of the calculations need be made as from the one second trials.

```
*********************************************
*                                           *
*  PARAMETRIC IDENTIFICATION - EXTENDED KALMAN FILTER  *
*                                           *
*********************************************


        SYSTEM:  MARINER-CLASS SURFACE VESSEL


        MANEUVER:  ZIG-ZAG, WITH SINUSOICAL RUDDER
                   DEFLECTIONS OF PERICD 200.0 SECONDS AND
                   MAXIMUM DEFLECTIONS OF 10.0 DEGREES


        NOISE LEVEL:  MEASUREMENT NOISE -   5%

                      PROCESS NCISE -   5%


        EXAGGERATED NOISE FACTCR:  1.0


        TRIAL PERIOD:   376 SECCNDS


        TIME STEP:  4.0 SECCNCS


        NUMBER OF PRIMARY STATE VARIABLES:   4


        NUMBER OF COEFFICIENTS IDENTIFIED:    4


                 (NCN-LINEAR MCDEL)
```

Table 5-4a Conditions for the Variation in Time Increment

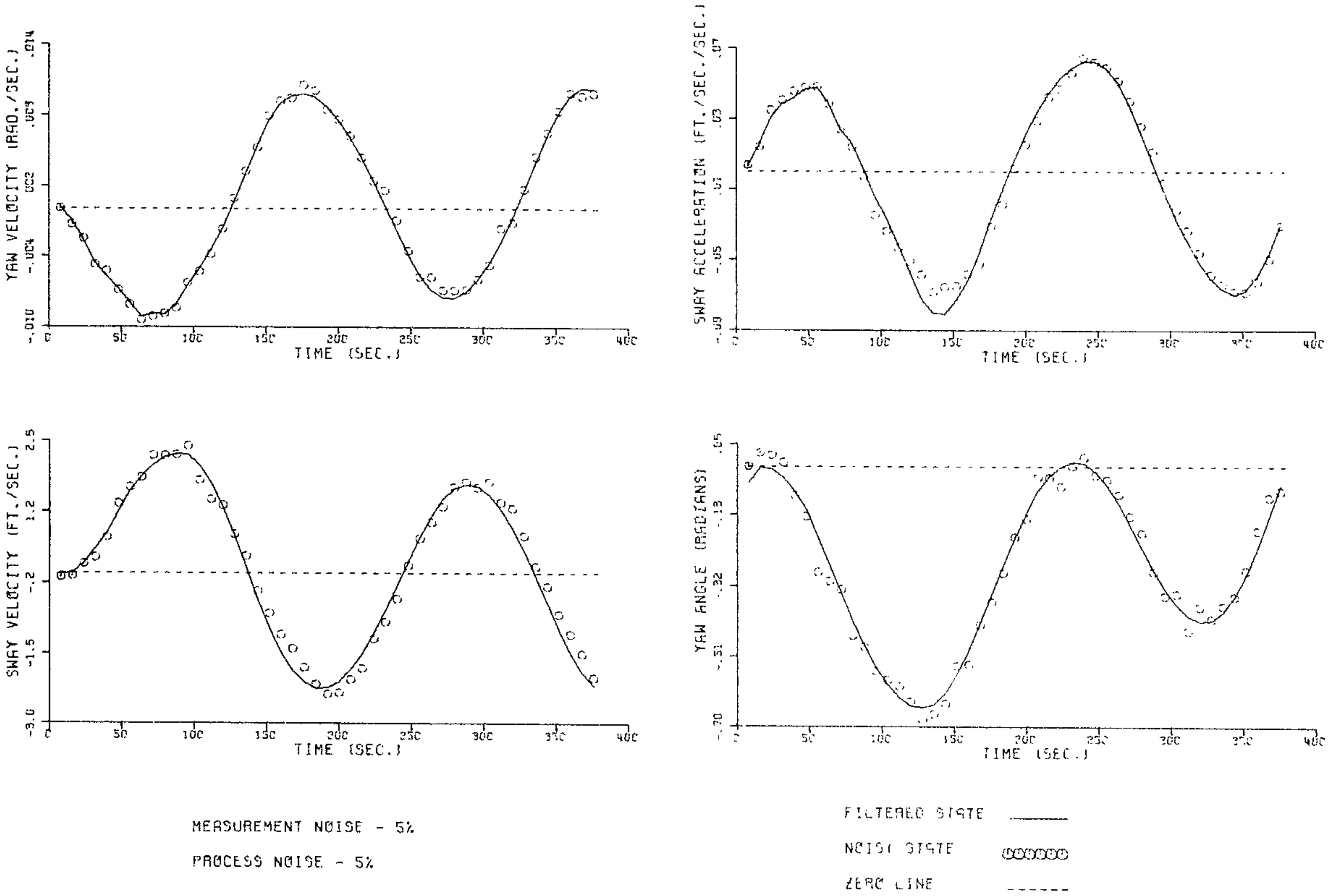


Fig. 5-4b Filtered States - Variation in Time Increment

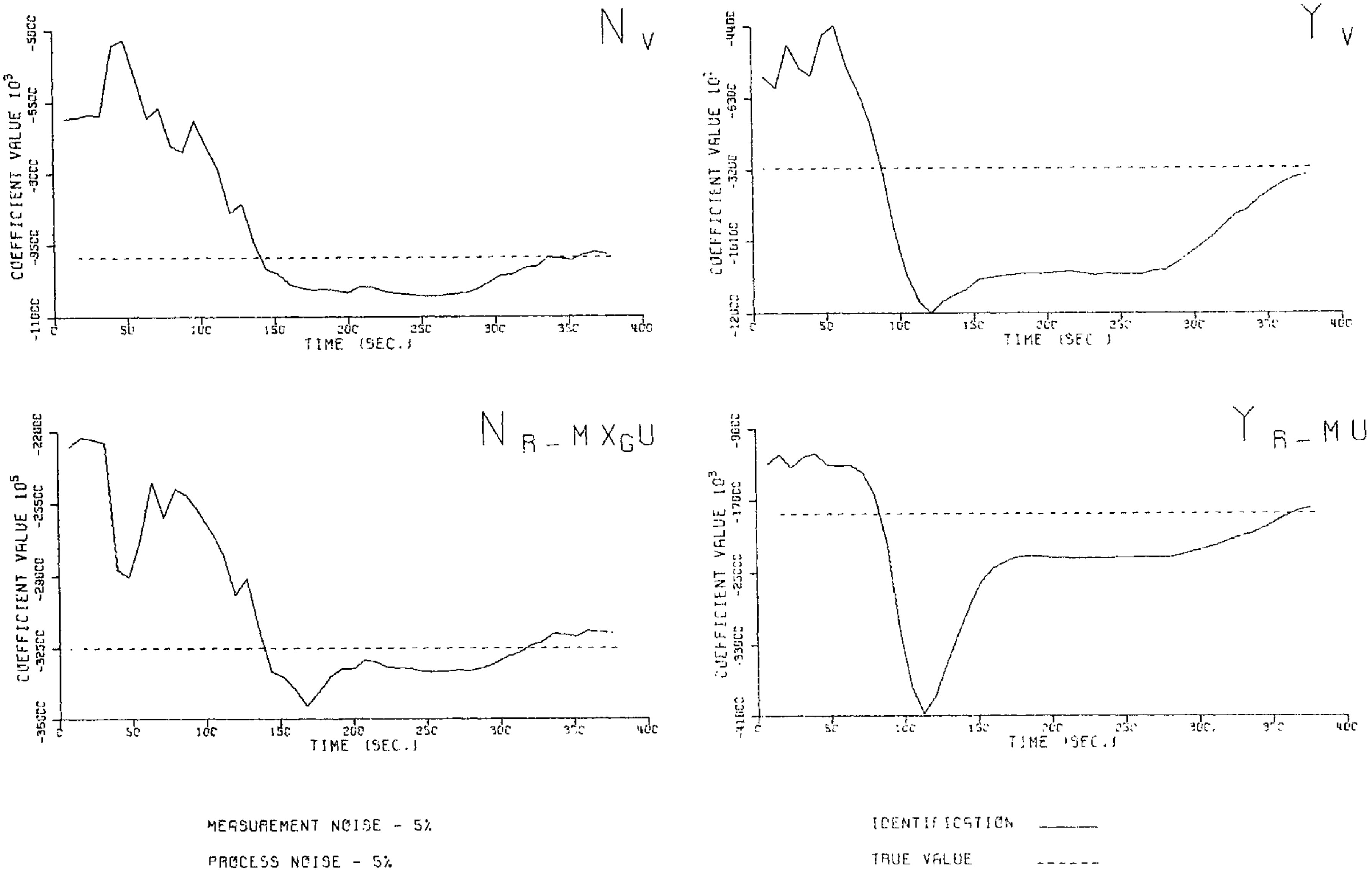


Fig. 5-4b Coefficients - Variation in Time Increment

```
NP =  13     TRUE VALUE  =   -0.32510E+10
SV  =   -0.227520+10   + OR -    0.975810+09
FV  =   -0.31749E+10   + OR -    0.97594E+08

IDENTIFICATION WITHIN  2.34% OF THE TRUE VALUE.
```

$(N_r - mx_G u)$

```
NP =  12     TRUE VALUE  =   -0.97735E+07
SV  =   -0.684140+07   + OR -    0.293210+07
FV  =   -0.96934E+07   + OR -    0.42230E+06

IDENTIFICATION WITHIN  0.82% OF THE TRUE VALUE.
```

N_v

```
NP =   7     TRUE VALUE  =   -0.18508E+08
SV  =   -0.129550+08   + OR -    0.555250+07
FV  =   -0.17908E+08   + OR -    0.62208E+06

IDENTIFICATION WITHIN  3.24% OF THE TRUE VALUE.
```

$(Y_r - mu)$

```
NP =   6     TRUE VALUE  =   -0.81515E+05
SV  =   -0.570600+05   + OR -    0.244540+05
FV  =   -0.83280E+05   + OR -    0.28920E+04

IDENTIFICATION WITHIN  2.17% OF THE TRUE VALUE.
```

Y_v

Table 5-4b Coefficient Identification for the Variation in Time Increment

5.5 Variation in the Number of Observed Primary State Variables

Occasionally, it may be necessary to alter the number of measured motion parameters. It may not be possible to use the integrating accelerometer, or possibly only r and ψ can be measured under the operating conditions of the trial. For this run it was assumed that $\dot{v}$ could not be attained. The measured variables are therefore decreased to three in number - v, r and ψ.

As expected, the results are not as accurate as those obtained when there was more information fed into the filter. Still, all coefficients are in the range of being classified as identified. Except for (Y_r - mu), the difference in accuracy is approximately a factor of two. For (Y_r - mu) the identification was substantially diminished for no other reason than the fact that in the original case, the coefficient was fully identified.

```
*********************************************************
*                                                       *
*  PARAMETRIC IDENTIFICATION - EXTENDED KALMAN FILTER   *
*                                                       *
*********************************************************


        SYSTEM:  MARINER-CLASS SURFACE VESSEL


        MANEUVER:  ZIG-ZAG, WITH SINUSOIDAL RUDDER
                   DEFLECTIONS OF PERIOD 200.0 SECONDS AND
                   MAXIMUM DEFLECTIONS OF 10.0 DEGREES


        NOISE LEVEL:  MEASUREMENT NOISE -   5%

                      PROCESS NOISE -   5%


        EXAGGERATED NOISE FACTOR:  1.0


        TRIAL PERIOD:   376 SECONDS


        TIME STEP:  1.0 SECONDS


        NUMBER OF PRIMARY STATE VARIABLES:   3


        NUMBER OF COEFFICIENTS IDENTIFIED:    4


                  (NON-LINEAR MODEL)
```

Table 5-5a Conditions for the Variation in the Number of Measured Primary State Variables

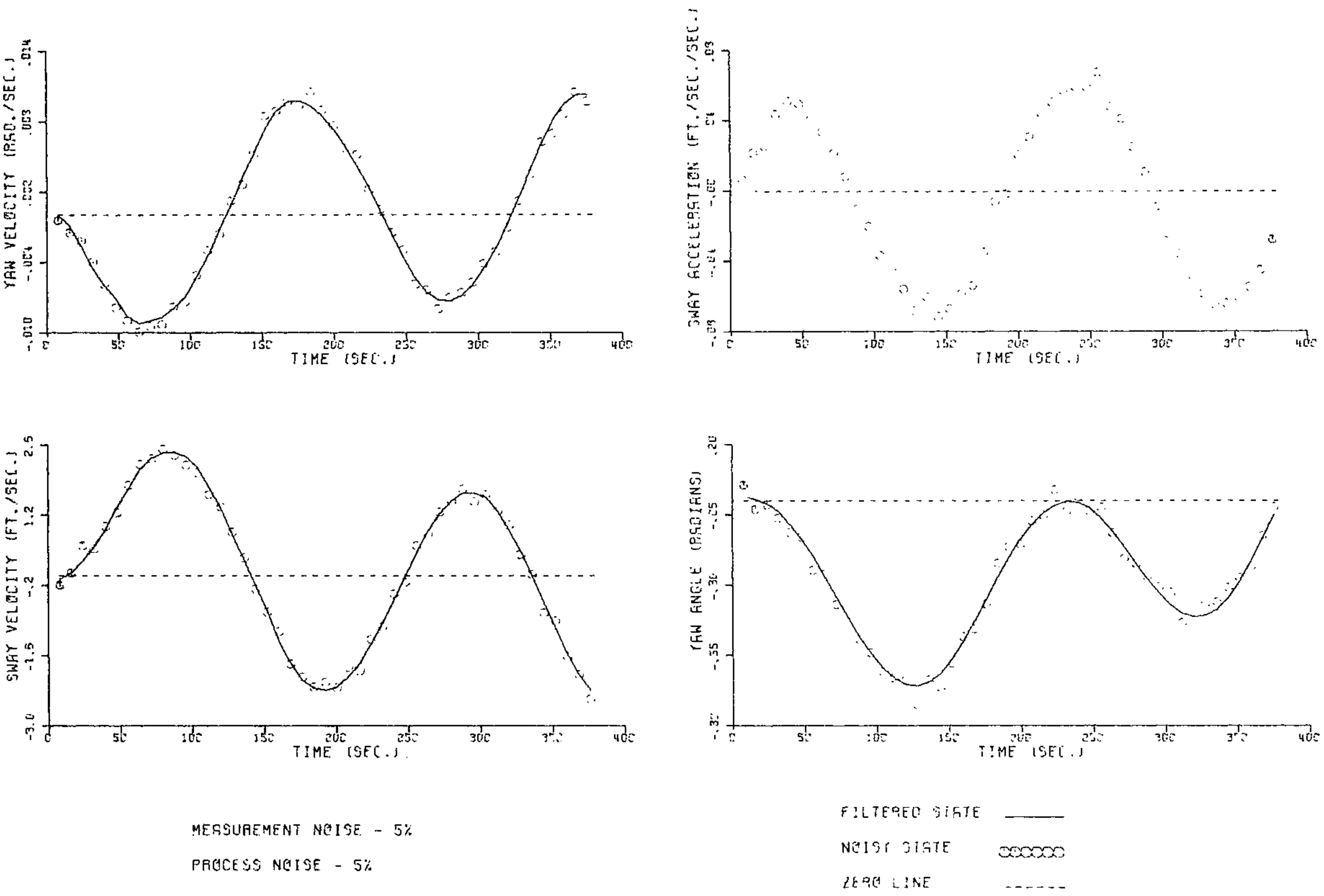


Fig. 5-5a Filtered States - Variation in the Number of Measured State Variables

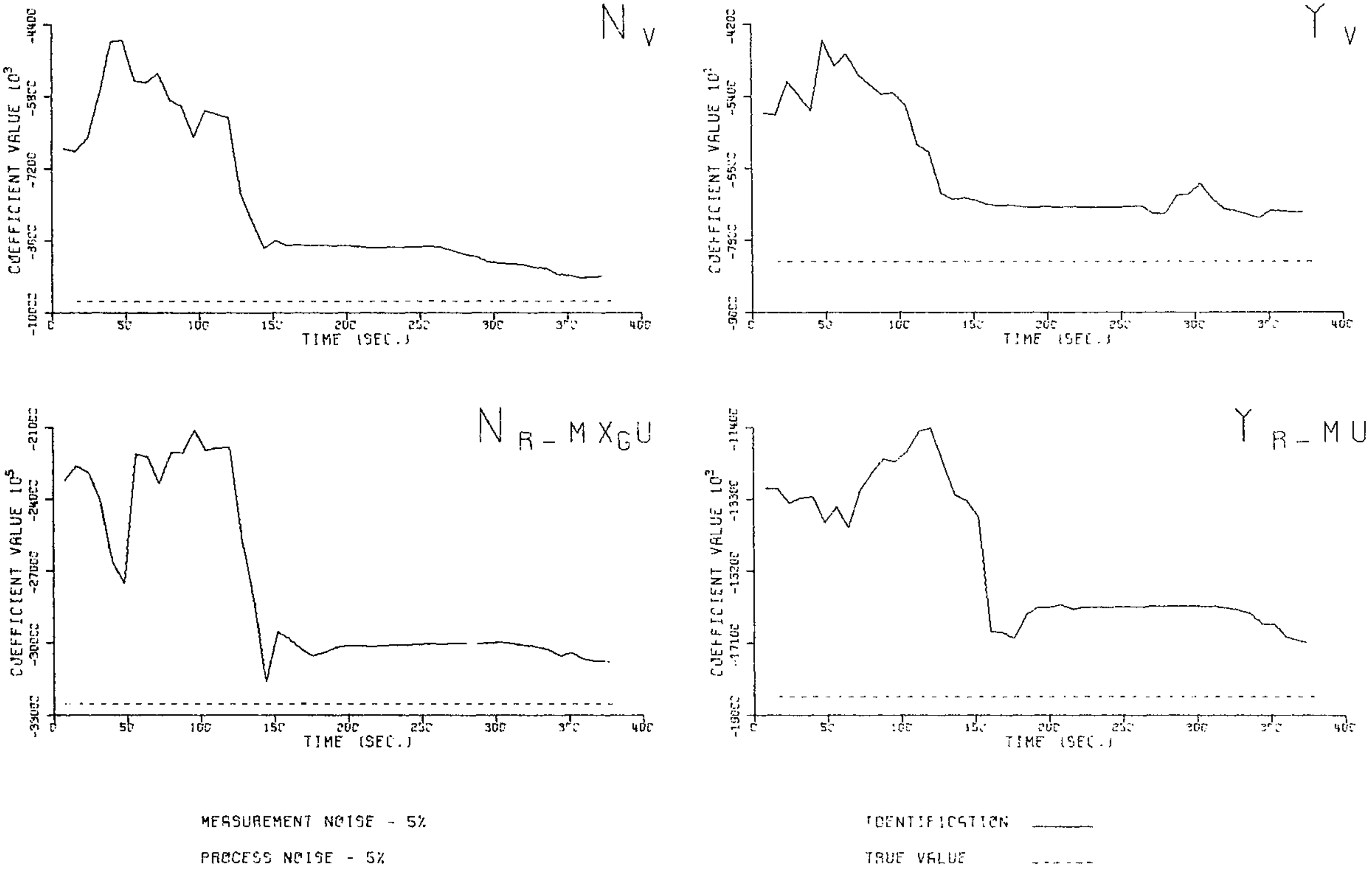


Fig. 5-5b Coefficients - Variation in the Number of Measured State Variables

```
NP =  13      TRUE VALUE   =   -0.32510E+10

SV   =    -0.22752D+10   + OR -     0.97581D+09      (N_r - mx_G u)

FV   =    -0.30754E+10   + OR -     0.42554E+08

IDENTIFICATION WITHIN  5.40% OF THE TRUE VALUE.
```

```
NP =  12      TRUE VALUE   =   -0.97735E+07

SV   =    -0.68414D+07   + OR -     0.29321D+07      Y_v

FV   =    -0.92879E+07   + OR -     0.21235E+06

IDENTIFICATION WITHIN  4.97% OF THE TRUE VALUE.
```

```
NP =   7      TRUE VALUE   =   -0.18508E+08

SV   =    -0.12955D+08   + OR -     0.55525D+07      (Y_r - mu)

FV   =    -0.17107E+08   + OR -     0.33004E+06

IDENTIFICATION WITHIN  7.57% OF THE TRUE VALUE.
```

```
NP =   6      TRUE VALUE   =   -0.81515E+05

SV   =    -0.57060D+05   + OR -     0.24454D+05      N_v

FV   =    -0.73271E+05   + OR -     0.19199E+04

IDENTIFICATION WITHIN 10.11% OF THE TRUE VALUE.
```

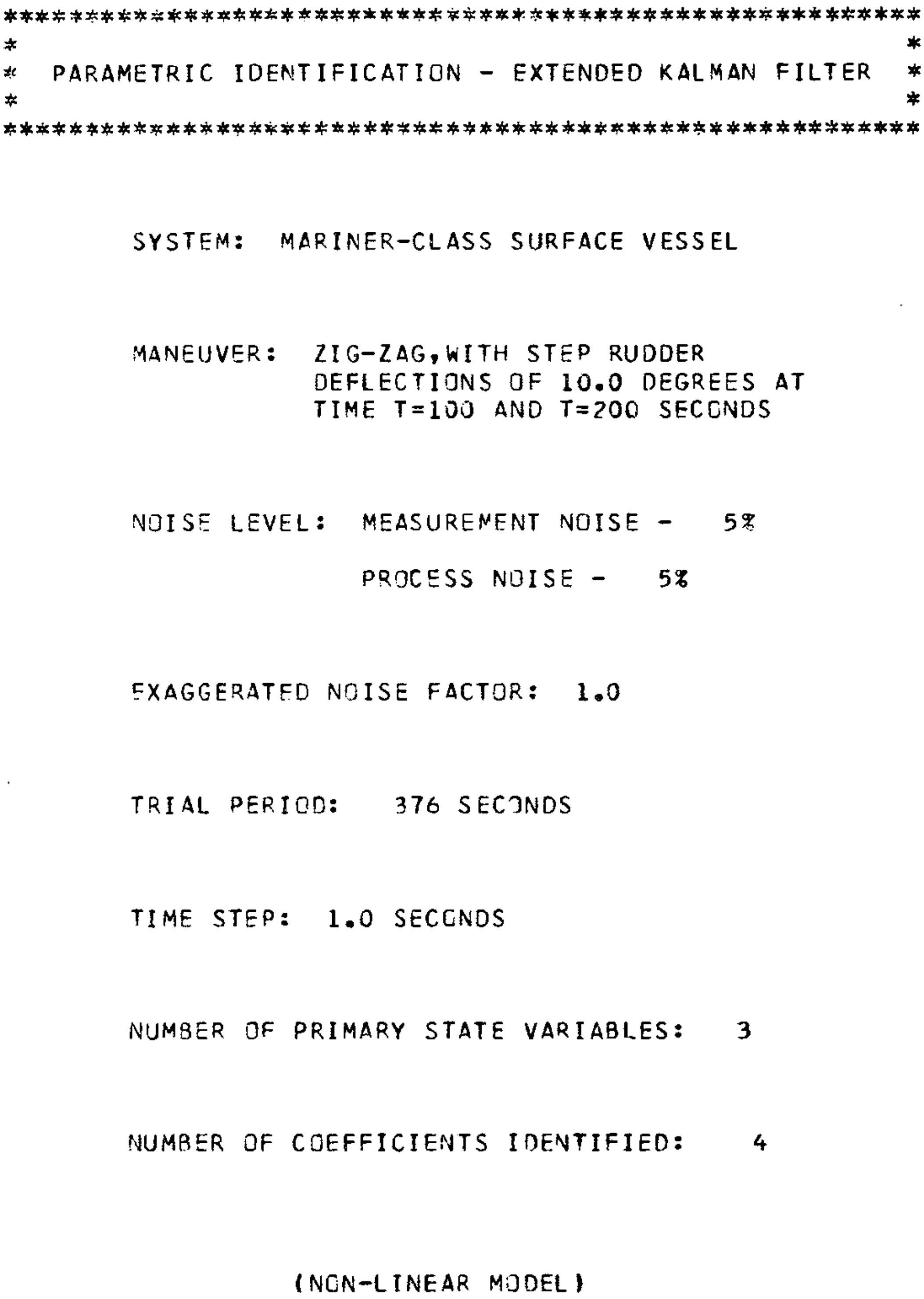
Table 5-5b Coefficient Identification for the Variation in the Number of Measured State Variables

5.6 Variation in the Maneuver

As discussed earlier, one is not limited to the sinusoidal zig-zag maneuver in the identification trial. Any number of different maneuvers can be developed, many already in current use. Two are shown here. These are the single-step rudder rudder deflection and the zig-zag maneuver with step rudder deflections. A discussion of the characteristics of each trial is given in Chapter IV. As indicated, $\dot{v}$ cannot be one of the measured variables for the case of the step zig-zag deflections. For this reason it was decided to continue the policy of the last case, inputing values of v, r and ψ only to the filtering process. The results are best compared to the previous case where the sinusoidal maneuver is used, again measuring but three motion parameters.

The zig-zag step deflection is seen to give results s similar to the sinusoidal case. The value of $(Y_r - mu)$ improved by several percentage points, while that of N_v was much worse. The filter again seems to have settled down after 200 seconds to a particular value. A high degree of confidence in that value is shown, even though it is not as accurate as might be expected. It should be noted that after 200 seconds the maneuver settles down to a steady state and no longer inputs a variation in the motion to the filter.

The case of the single step deflection is disappointing. There was no identification at all. In the case of the coefficient (Y_r - mu), the final value was even worse than the initial estimate. The identification process for N_v and Y_v settles down to a final value very quickly, after just 75 seconds. Again, it is at this point that the velocity has reached it's steady state value. There is no further variation in v over the rest of the period.



```
*************************************************
*                                               *
*  PARAMETRIC IDENTIFICATION - EXTENDED KALMAN FILTER  *
*                                               *
*************************************************


       SYSTEM:  MARINER-CLASS SURFACE VESSEL


       MANEUVER:  ZIG-ZAG,WITH STEP RUDDER
                  DEFLECTIONS OF 10.0 DEGREES AT
                  TIME T=100 AND T=200 SECONDS


       NOISE LEVEL:  MEASUREMENT NOISE -   5%

                     PROCESS NOISE -   5%


       EXAGGERATED NOISE FACTOR:  1.0


       TRIAL PERIOD:   376 SECONDS


       TIME STEP:  1.0 SECONDS


       NUMBER OF PRIMARY STATE VARIABLES:   3


       NUMBER OF COEFFICIENTS IDENTIFIED:    4


                    (NON-LINEAR MODEL)
```

Table 5-6a Conditions for the Variation in Maneuver (Zig-zag Step Rudder Deflection)

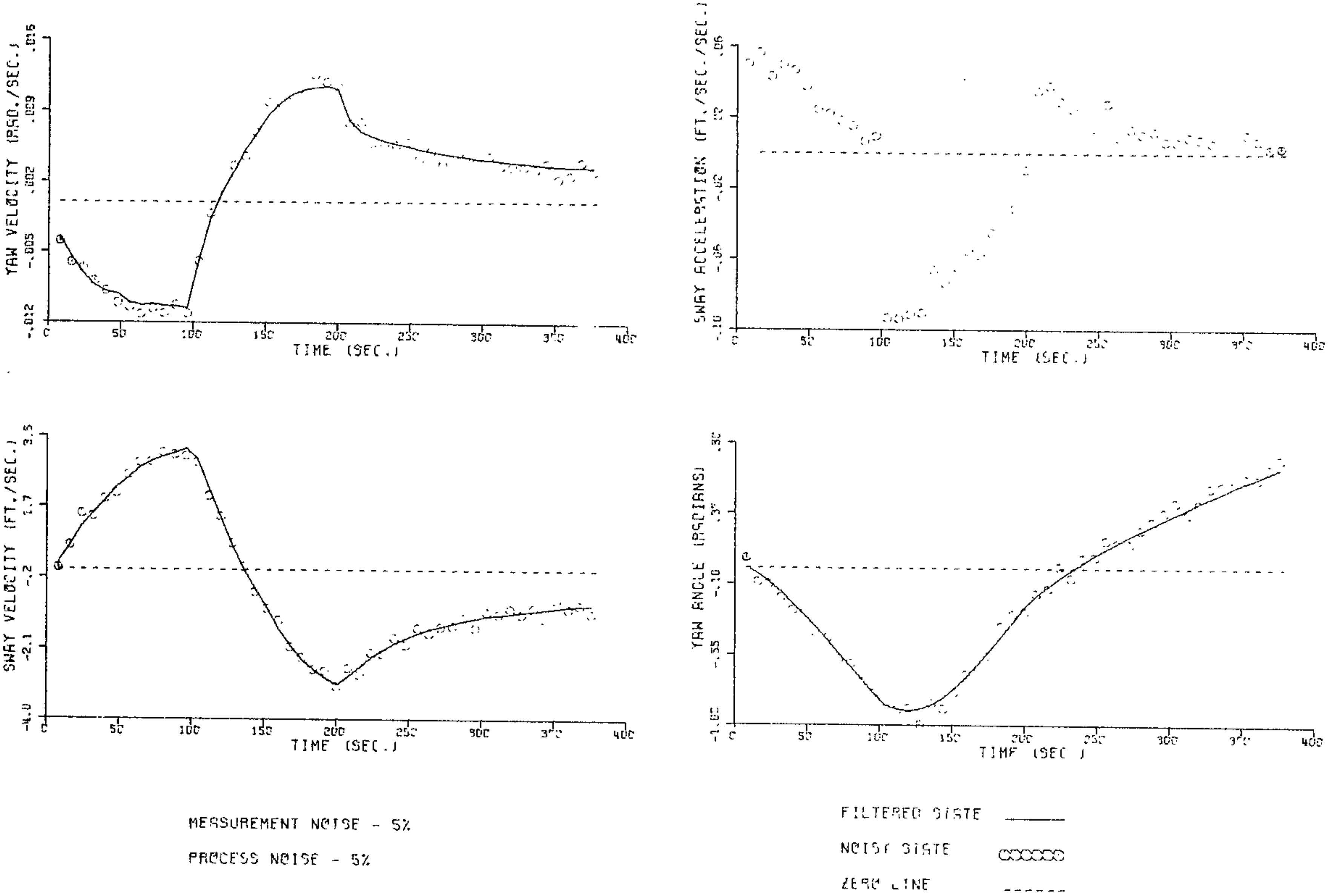


Fig. 5-6a Filtered States - Variation in Maneuver (Zig-zag Step Rudder Deflection)

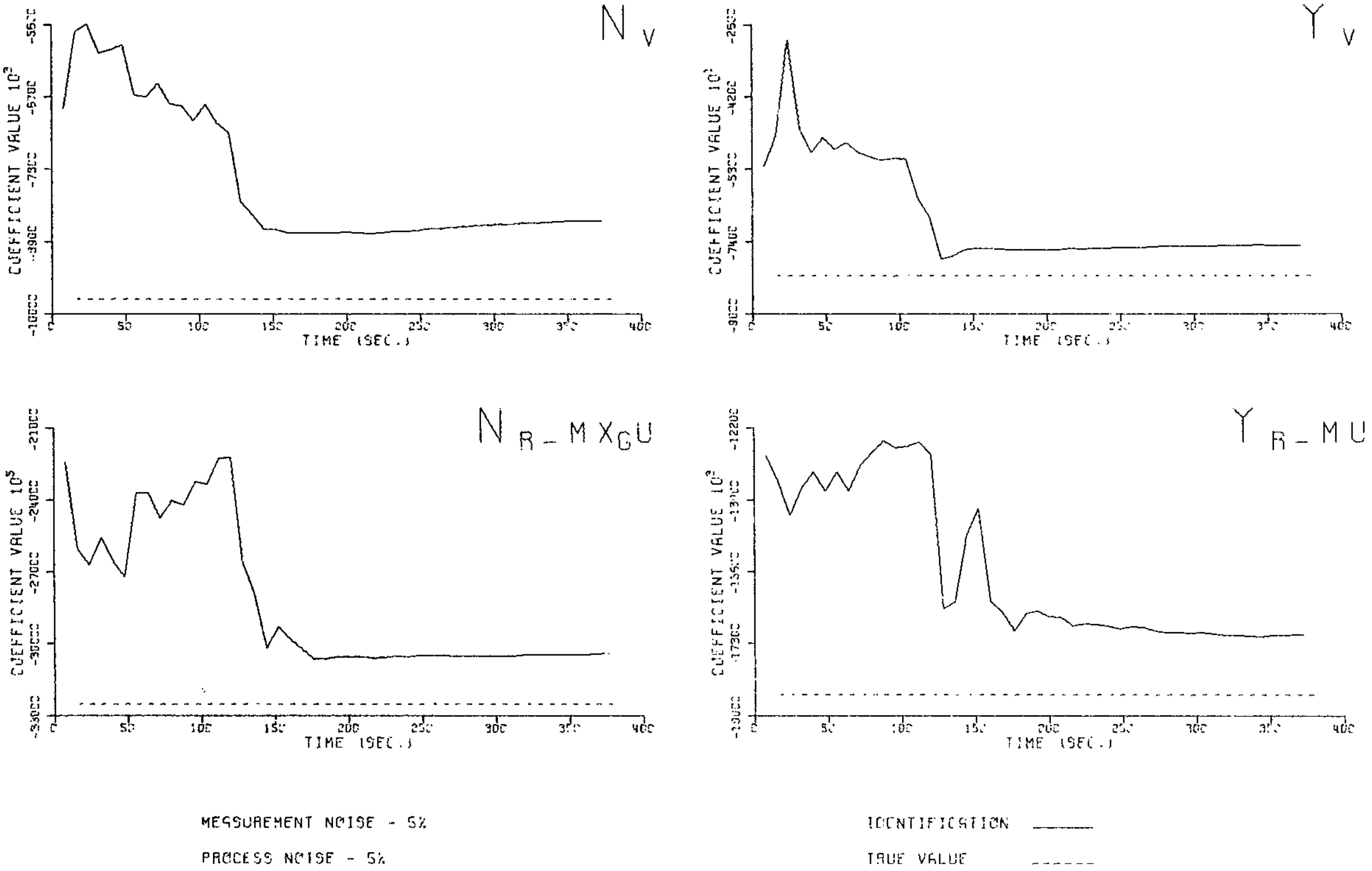


Fig. 5-6b Coefficients - Variation in Maneuver (Zig-zag Step Rudder Deflection)

```
NP =  13      TRUE VALUE  =   -0.32510E+10
SV  =   -0.22752D+10  + OR -    0.97581D+09
FV  =   -0.30418E+10  + OR -    0.55280E+08

IDENTIFICATION WITHIN  6.43% OF THE TRUE VALUE.
```

$(N_r - mx_G u)$

```
NP =  12      TRUE VALUE  =   -0.97735E+07
SV  =   -0.68414D+07  + OR -    0.29321D+07
FV  =   -0.85771E+07  + OR -    0.34697E+06

IDENTIFICATION WITHIN 12.24% OF THE TRUE VALUE.
```

N_v

```
NP =   7      TRUE VALUE  =   -0.18508E+08
SV  =   -0.12955D+08  + OR -    0.55525D+07
FV  =   -0.17090E+08  + OR -    0.32648E+06

IDENTIFICATION WITHIN  7.66% OF THE TRUE VALUE.
```

$(Y_r - mu)$

```
NP =   6      TRUE VALUE  =   -0.81515E+05
SV  =   -0.57060D+05  + OR -    0.24454D+05
FV  =   -0.74825E+05  + OR -    0.39778E+04

IDENTIFICATION WITHIN  8.21% OF THE TRUE VALUE.
```

Y_v

Table 5-6b Coefficient Identification for the Variation in Maneuver (Zig-Zag Step Rudder Deflection)

```
********************************************************
*                                                      *
*  PARAMETRIC IDENTIFICATION - EXTENDED KALMAN FILTER  *
*                                                      *
********************************************************


        SYSTEM:  MARINER-CLASS SURFACE VESSEL


        MANEUVER:  STEP RUDDER DEFLECTION AT T=0
                   MAXIMUM DEFLECTION OF 10.0 DEGREES


        NOISE LEVEL:  MEASUREMENT NOISE -   5%

                      PROCESS NOISE -   5%


        EXAGGERATED NOISE FACTOR:  1.0


        TRIAL PERIOD:   376 SECONDS


        TIME STEP:  1.0 SECONDS


        NUMBER OF PRIMARY STATE VARIABLES:   3


        NUMBER OF COEFFICIENTS IDENTIFIED:    4


                 (NON-LINEAR MODEL)
```

Table 5-6c Conditions for the Variation in Maneuver (Single-Step Rudder Deflection)

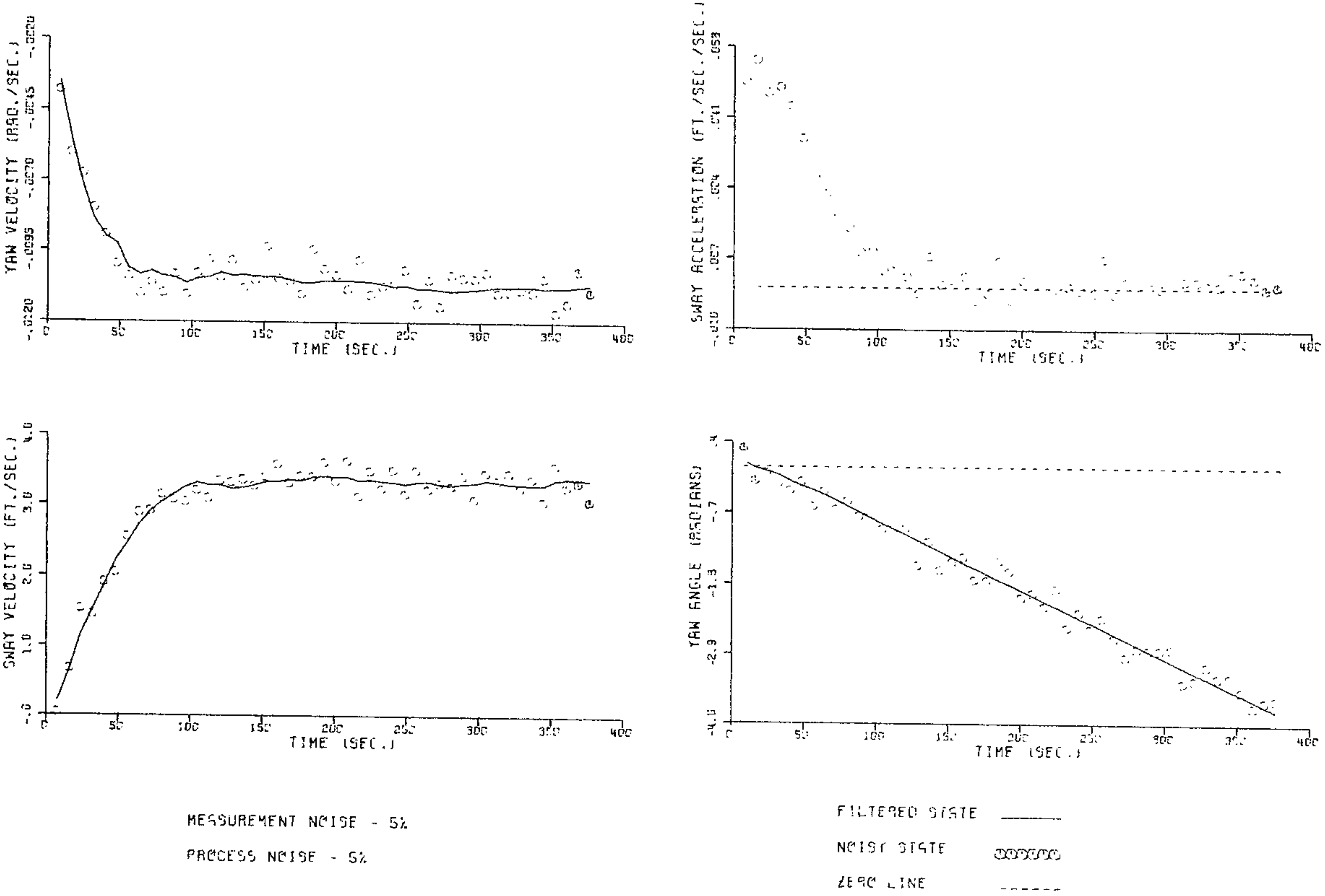


Fig. 5-6c Filtered States - Variation in Maneuver
(Single-Step Rudder Deflection)

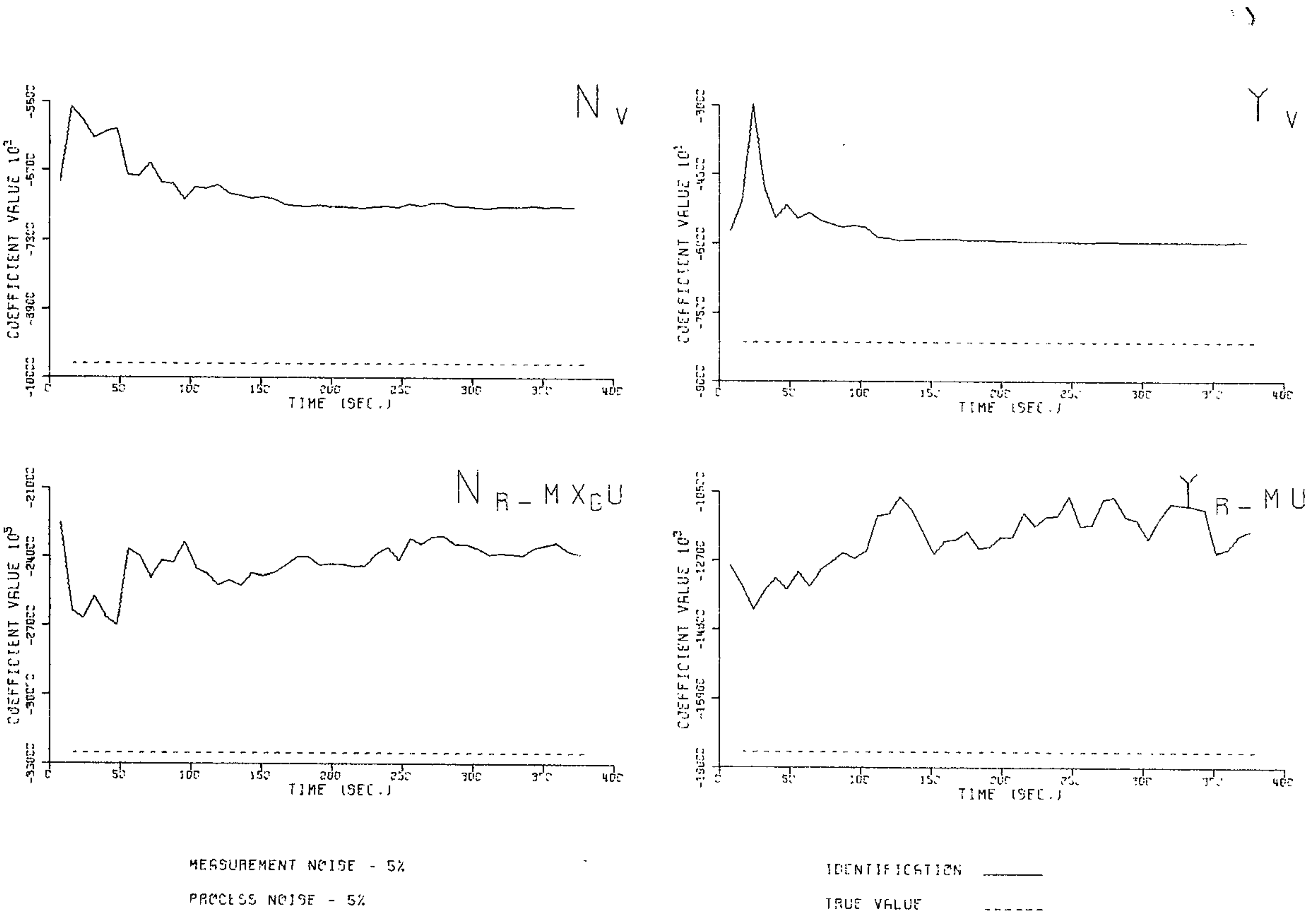


Fig. 5-6d Coefficients - Variation in Maneuver
(Single-Step Rudder Deflection)

```
NP =  13      TRUE VALUE   =   -0.32510E+10
SV   =   -0.22752D+10   + OR -    0.97581D+09
FV   =   -0.23889E+10   + OR -    0.66537E+09

IDENTIFICATION WITHIN 26.52% OF THE TRUE VALUE.
```

$(N_r - mx_Gu)$

```
NP =  12      TRUE VALUE   =   -0.97735E+07
SV   =   -0.68414D+07   + OR -    0.29321D+07
FV   =   -0.72667E+07   + OR -    0.19934E+07

IDENTIFICATION WITHIN 25.65% OF THE TRUE VALUE.
```

N_v

```
NP =   7      TRUE VALUE   =   -0.18508E+08
SV   =   -0.12955D+08   + OR -    0.55525D+07
FV   =   -0.11789E+08   + OR -    0.47030E+07

IDENTIFICATION WITHIN 36.30% OF THE TRUE VALUE.
```

$(Y_r - mu)$

```
NP =   6      TRUE VALUE   =   -0.81515E+05
SV   =   -0.57060D+05   + OR -    0.24454D+05
FV   =   -0.59769E+05   + OR -    0.99413E+04

IDENTIFICATION WITHIN 26.68% OF THE TRUE VALUE.
```

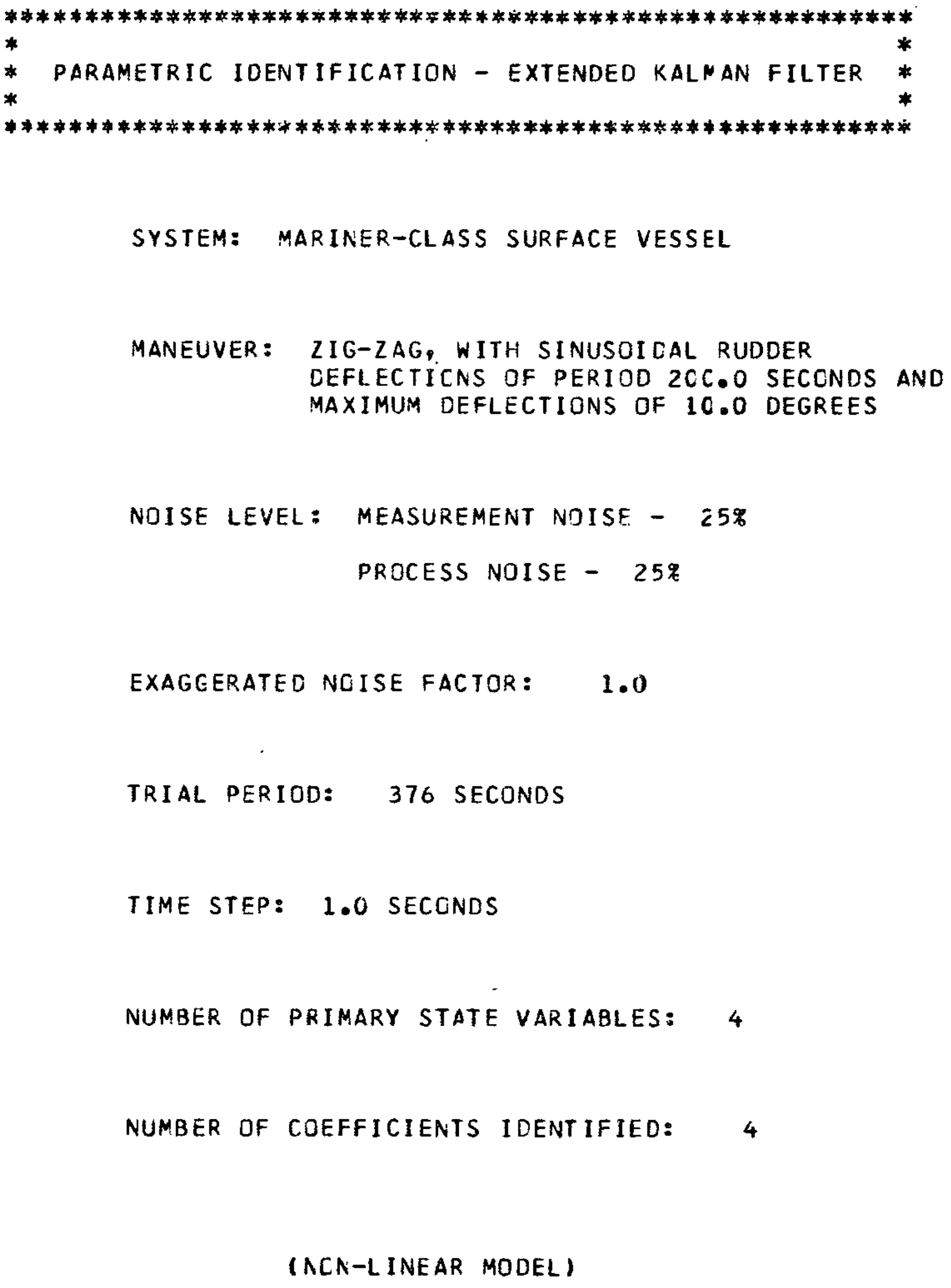
Y_v

Table 5-6d Coefficient Identification for the Variation in Maneuver
(Single-Step Rudder Deflection)

5.7 Variation in the Noise Level

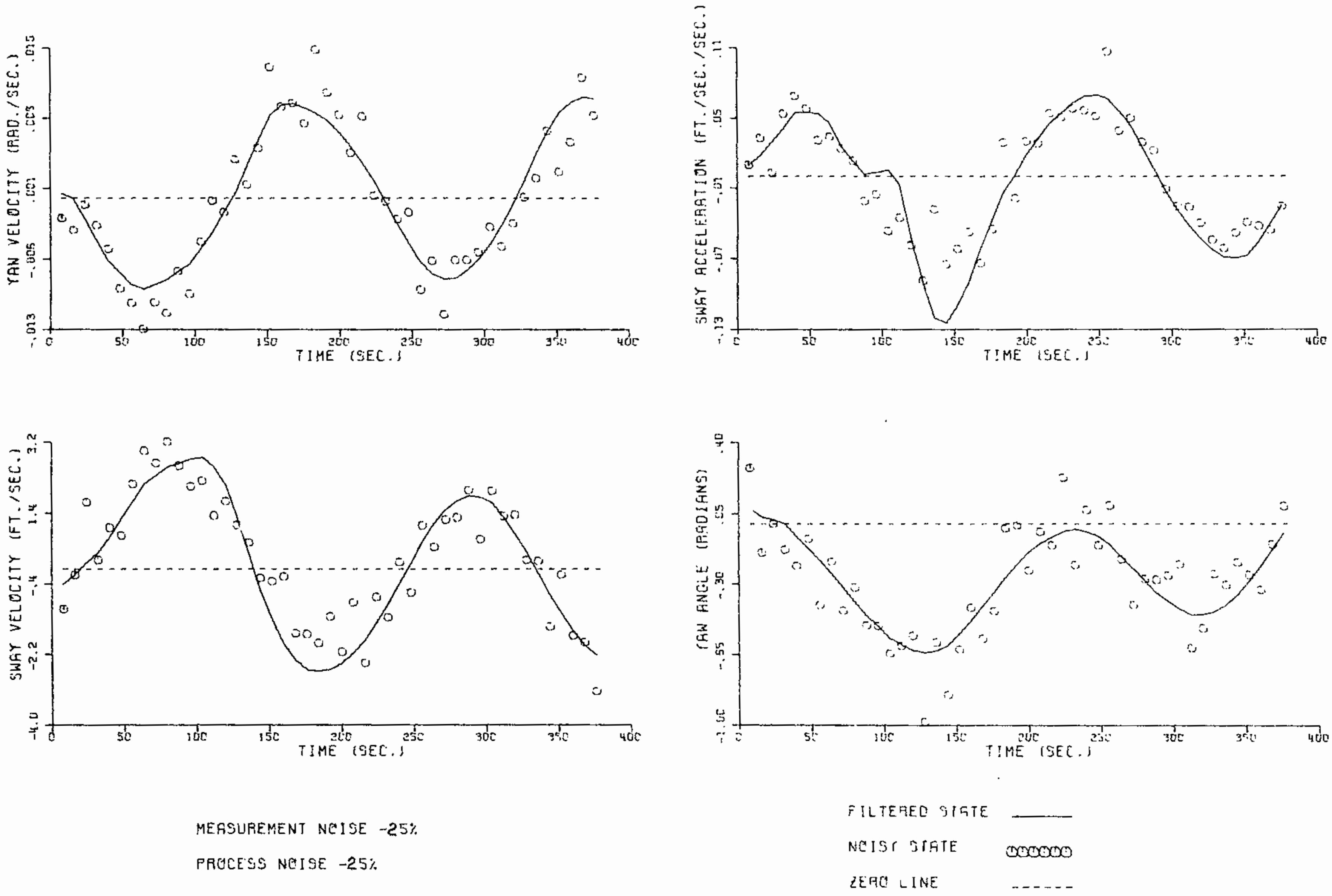
This run was made to see how the filter would react to large quantities of noise (25%) in the measured input. For the most part, these results are encouraging. All coefficients were identified except one, N_v, and the identification of Y_v and (Y_r - mu) is as good as that determined under low noise conditions. Seeing the amount of scatter in the measured data, it is remarkable that any accuracy, much less good identification, can take place. More noise was not run since that did not seem realistic; lesser quantities should exhibit the same results.

It will be seen that these values may be improved upon by resubmitting the final estimate to the filter and reprocessing the information.



```
*******************************************************
*                                                     *
*  PARAMETRIC IDENTIFICATION - EXTENDED KALMAN FILTER *
*                                                     *
*******************************************************


        SYSTEM:  MARINER-CLASS SURFACE VESSEL


        MANEUVER:  ZIG-ZAG, WITH SINUSOIDAL RUDDER
                   DEFLECTICNS OF PERIOD 2CC.0 SECCNDS AND
                   MAXIMUM DEFLECTIONS OF 1C.0 DEGREES


        NOISE LEVEL:  MEASUREMENT NOISE -  25%

                      PROCESS NOISE -  25%


        EXAGGERATED NCISE FACTOR:    1.0


        TRIAL PERIOD:   376 SECONDS


        TIME STEP:  1.0 SECCNDS


        NUMBER OF PRIMARY STATE VARIABLES:   4


        NUMBER OF COEFFICIENTS IDENTIFIED:    4


                   (NCN-LINEAR MODEL)
```

Table 5-7a Conditions for the Variation in Noise Level



MEASUREMENT NOISE -25%

PROCESS NOISE -25%

Fig. 5-7a Filtered States - Variation in Noise Level

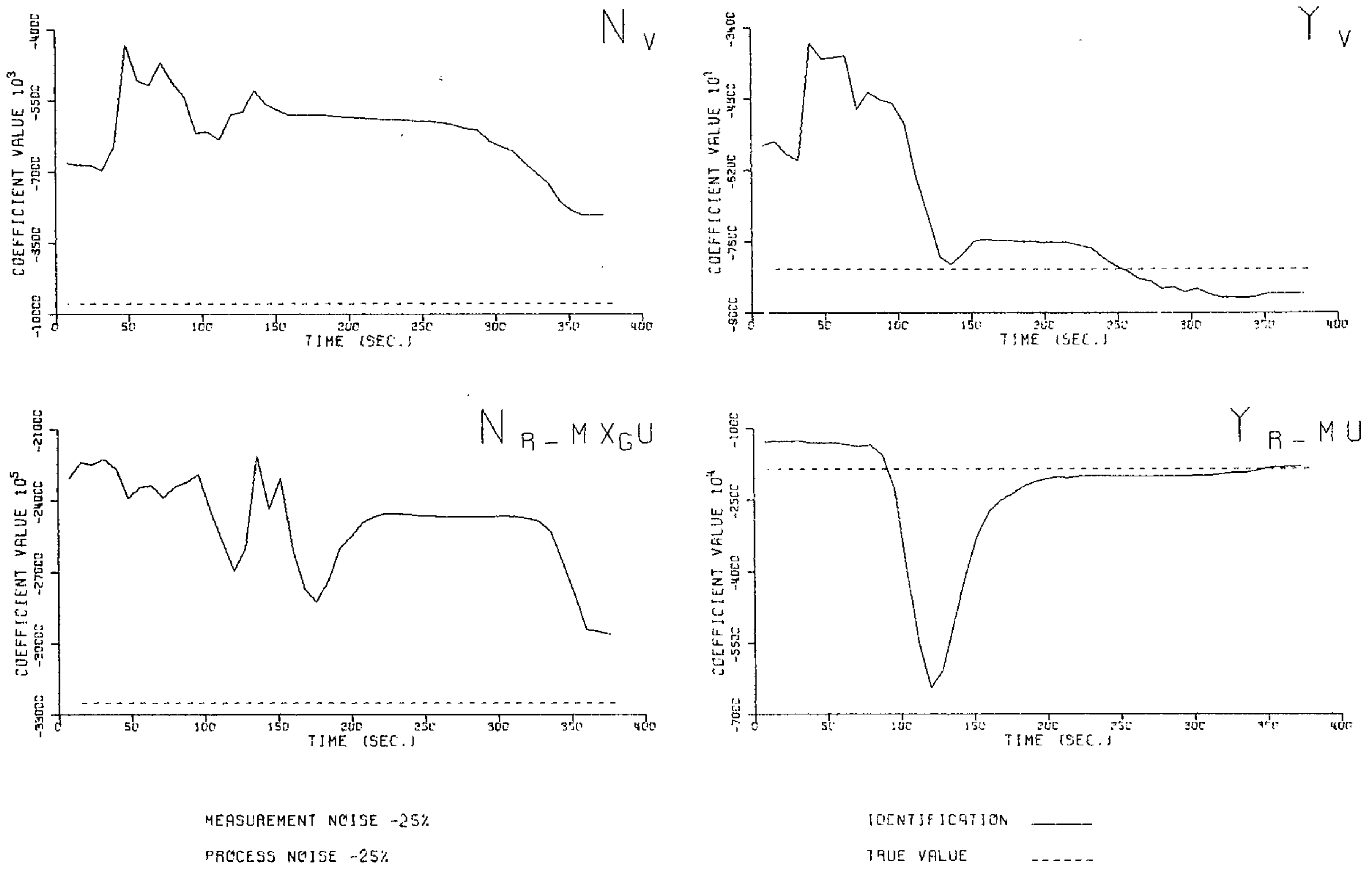


Fig. 5-7b Coefficients - Variation in Noise Level

```
NP =  13      TRUE VALUE   =   -0.32510E+10
SV   =    -0.22752D+10   + OR -     0.97581D+09
FV   =    -0.29581E+10   + OR -     0.17636E+09

IDENTIFICATION WITHIN  9.01% OF THE TRUE VALUE.
```

$(N_r - mx_Gu)$

```
NP =  12      TRUE VALUE   =   -0.97735E+07
SV   =    -0.68414D+07   + OR -     0.29321D+07
FV   =    -0.79114E+07   + OR -     0.71233E+06

IDENTIFICATION WITHIN 19.05% OF THE TRUE VALUE.
```

N_v

```
NP =   7      TRUE VALUE   =   -0.18508E+08
SV   =    -0.12955D+08   + OR -     0.55525D+07
FV   =    -0.18142E+08   + OR -     0.71847E+06

IDENTIFICATION WITHIN  1.98% OF THE TRUE VALUE.
```

$(Y_r - mu)$

```
NP =   6      TRUE VALUE   =   -0.81515E+05
SV   =    -0.57060D+05   + OR -     0.24454D+05
FV   =    -0.86228E+05   + OR -     0.31796E+04

IDENTIFICATION WITHIN  5.78% OF THE TRUE VALUE.
```

Y_v

Table 5-7b Coefficient Identification for the Variation in Noise Level

5.8 2^{ed} Generation Identification

One of the hoped-for capabilities of this system was that, even if the identification over the original pass was not as good as expected, the data could be resubmitted and processed again using the new estimates.

This run was the test of that hypothesis. The data obtained from the 25% noise run was resubmitted. Each coefficient except one showed an increase in accuracy. The remaining coeficient, (Y_r - mu), had been identified to within two percent on the previous pass. It should be noted that Y_v is essentially equivalent, percentage-wise, after the second pass to the first value. Apparently, the improved identification will take place only if first pass yields results more than five percent off the true value. If the identification is within five percent, the filter will become unstable and inaccuracy results. At any rate, this does indicate that the coefficient values can be re-evaluated. As seen here, after two passes the results from a 25% noise level can be made to within five or ten percent of their true values. This should be valuable for future considerations.

```
*************************************************************
*                                                           *
*  PARAMETRIC IDENTIFICATION - EXTENDED KALMAN FILTER       *
*                                                           *
*************************************************************


        SYSTEM:  MARINER-CLASS SURFACE VESSEL


        MANEUVER:  ZIG-ZAG, WITH SINUSOIDAL RUDDER
                   DEFLECTIONS OF PERIOD 200.0 SECONDS AND
                   MAXIMUM DEFLECTIONS OF 10.0 DEGREES


        NOISE LEVEL:  MEASUREMENT NOISE -  25%

                      PROCESS NOISE -  25%


        EXAGGERATED NOISE FACTOR:    1.0


        TRIAL PERIOD:   376 SECONDS


        TIME STEP:  1.0 SECONDS


        NUMBER OF PRIMARY STATE VARIABLES:   4


        NUMBER OF COEFFICIENTS IDENTIFIED:    4


                  (NON-LINEAR MODEL)
```

Table 5-8a Conditions for the Second-Generation Identification

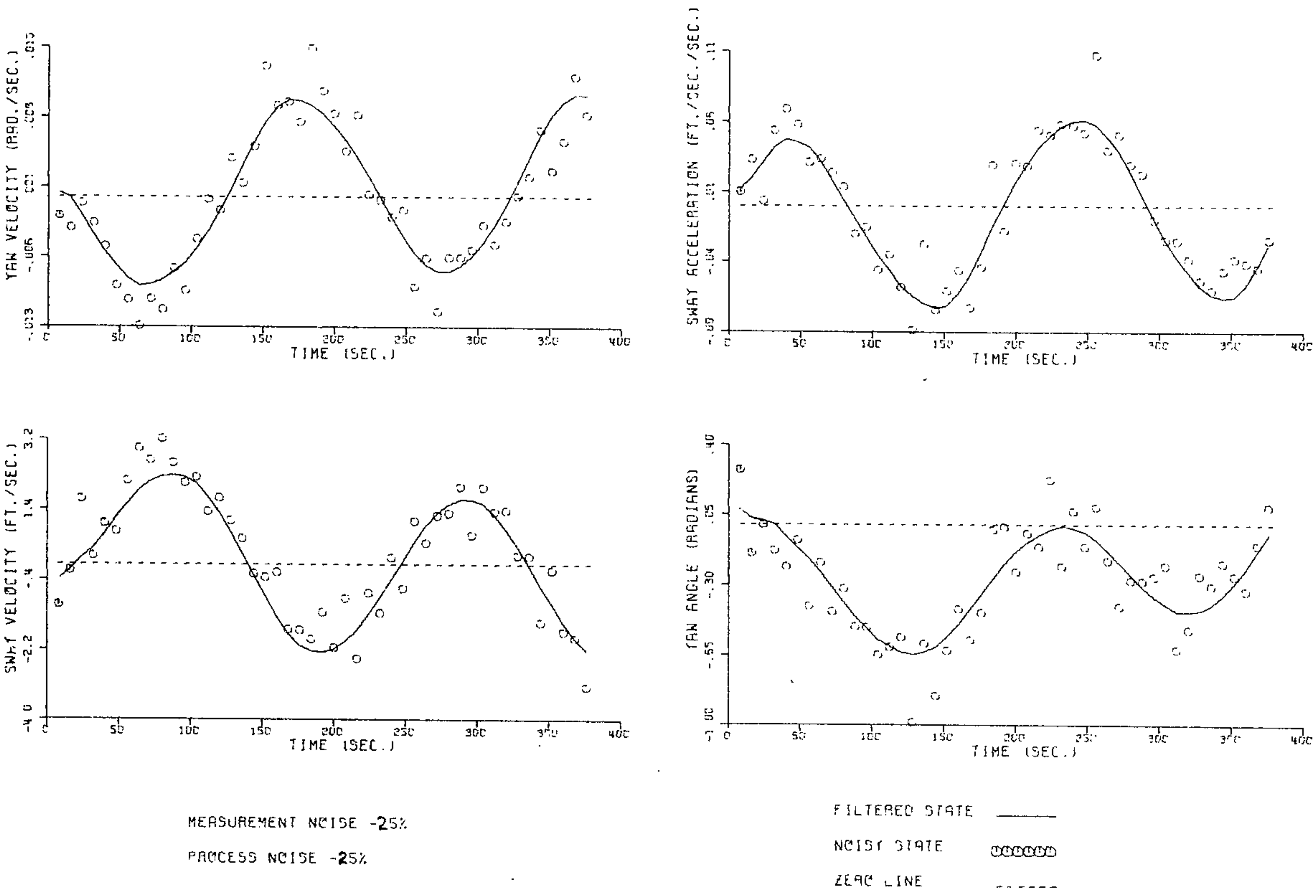


Fig. 5-8a Filtered States - Second-Generation Identification

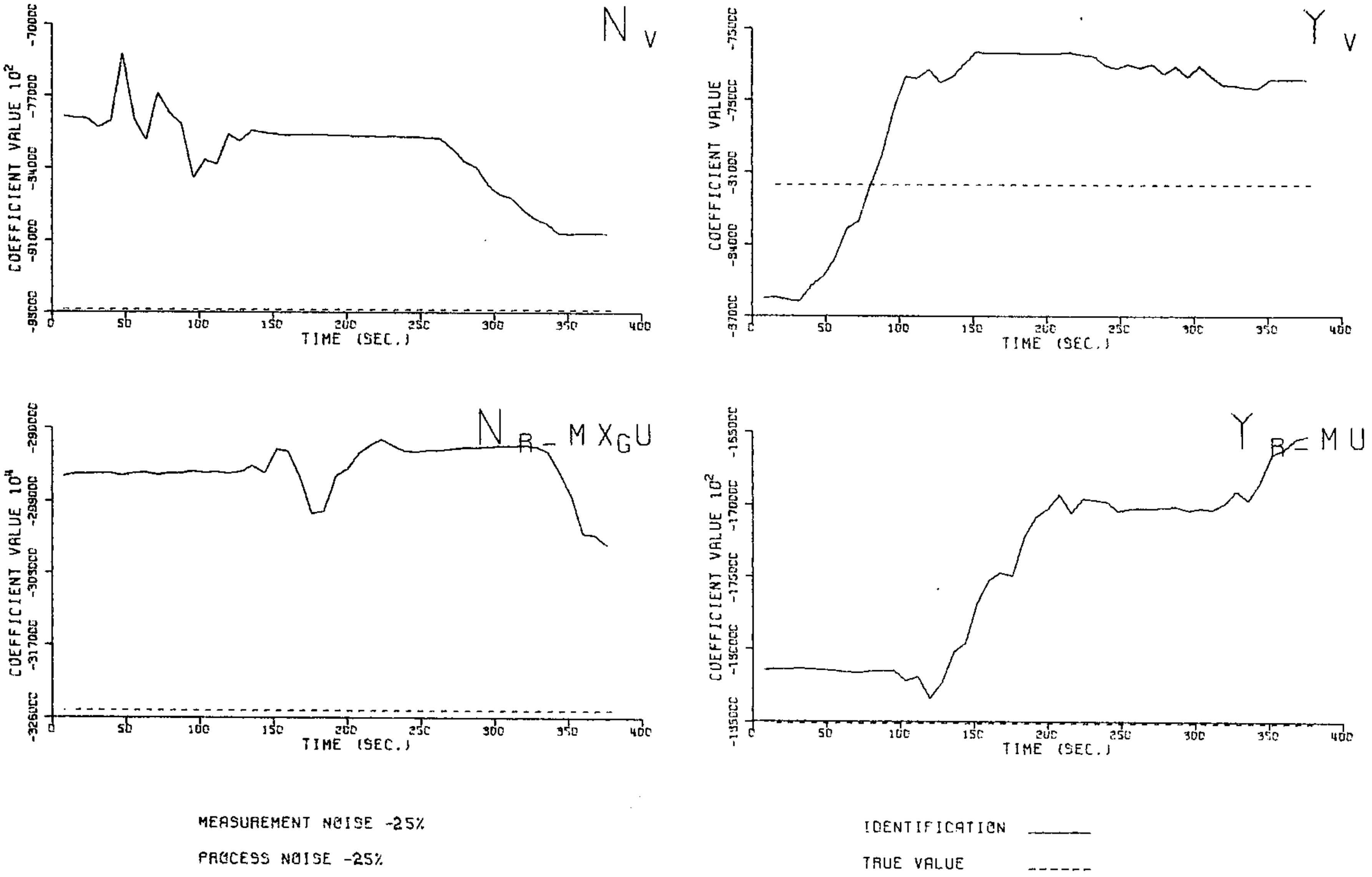


Fig. 5-8b Coefficients - Second-Generation Identification

```
NP =  13      TRUE VALUE   =   -0.32510E+10
SV  =    -0.29581D+10   + OR -     0.17636D+09
FV  =    -0.30434E+10   + OR -     0.10937E+09

IDENTIFICATION WITHIN  6.39% OF THE TRUE VALUE.
```

$(N_r - mx_G u)$

```
NP =  12      TRUE VALUE   =   -0.97735E+07
SV  =    -0.79114D+07   + OR -     0.19000D+07
FV  =    -0.90214E+07   + OR -     0.50969E+06

IDENTIFICATION WITHIN  7.70% OF THE TRUE VALUE.
```

N_v

```
NP =   7      TRUE VALUE   =   -0.18508E+08
SV  =    -0.18142D+08   + OR -     0.71847D+06
FV  =    -0.16524E+08   + OR -     0.48872E+06

IDENTIFICATION WITHIN 10.72% OF THE TRUE VALUE.
```

$(Y_r - mu)$

```
NP =   6      TRUE VALUE   =   -0.81515E+05
SV  =    -0.86228D+05   + OR -     0.31796D+04
FV  =    -0.77134E+05   + OR -     0.22458E+04

IDENTIFICATION WITHIN  5.37% OF THE TRUE VALUE.
```

Y_v

Table 5-8b Coefficient Identification for the Second-Generation Identification

5.9 Noise Exaggeration

Similar to introducing large noise values in the input data is the exaggeration of that level of noise present, be it high or low. This type of variation probably should be reserved for special situations. With the case given here, other cases were run using normal maneuvers with a one second time step and no exaggeration. As stated earlier, the filter became unstable for these runs and no identification was possible. However, it is seen here that by telling the filter that the noise level was higher than it actually was (25 times), some results were obtained. Granted, they are not very good, but compared to the case without exaggeration, namely no results, they are a substantial improvement.

A similar situation occurred when the noise level was very low (1%). By telling the filter that more noise is present than is actually the case, it remains stable and the identification process proceeds to completion.

Having a factor of 25 is probably excessive. A smaller factor of five or so should do the job just as adequately while not affecting the realism of the situation.

```
*********************************************
*                                           *
*  PARAMETRIC IDENTIFICATION - EXTENDEC KALMAN FILTER  *
*                                           *
*********************************************


      SYSTEM:  MARINER-CLASS SURFACE VESSEL


      MANEUVER:  ZIG-ZAG, WITH SINUSOICAL RUDDER
                 DEFLECTIONS OF PERIOD 200.0 SECONDS AND
                 MAXIMUM DEFLECTIONS OF 10.0 DEGREES


      NOISE LEVEL:  MEASUREMENT NOISE -   5%

                    PROCESS NOISE -   5%


      EXAGGERATED NOISE FACTOR:  25.0


      TRIAL PERIOD:   376 SECCNDS


      TIME STEP:  1.0 SECCNDS


      NUMBER OF PRIMARY STATE VARIABLES:   4


      NUMBER OF COEFFICIENTS IDENTIFIED:    4


              (NCN-LINEAR MCDEL)
```

Table 5-9a Conditions for the Variation in Noise Exaggeration

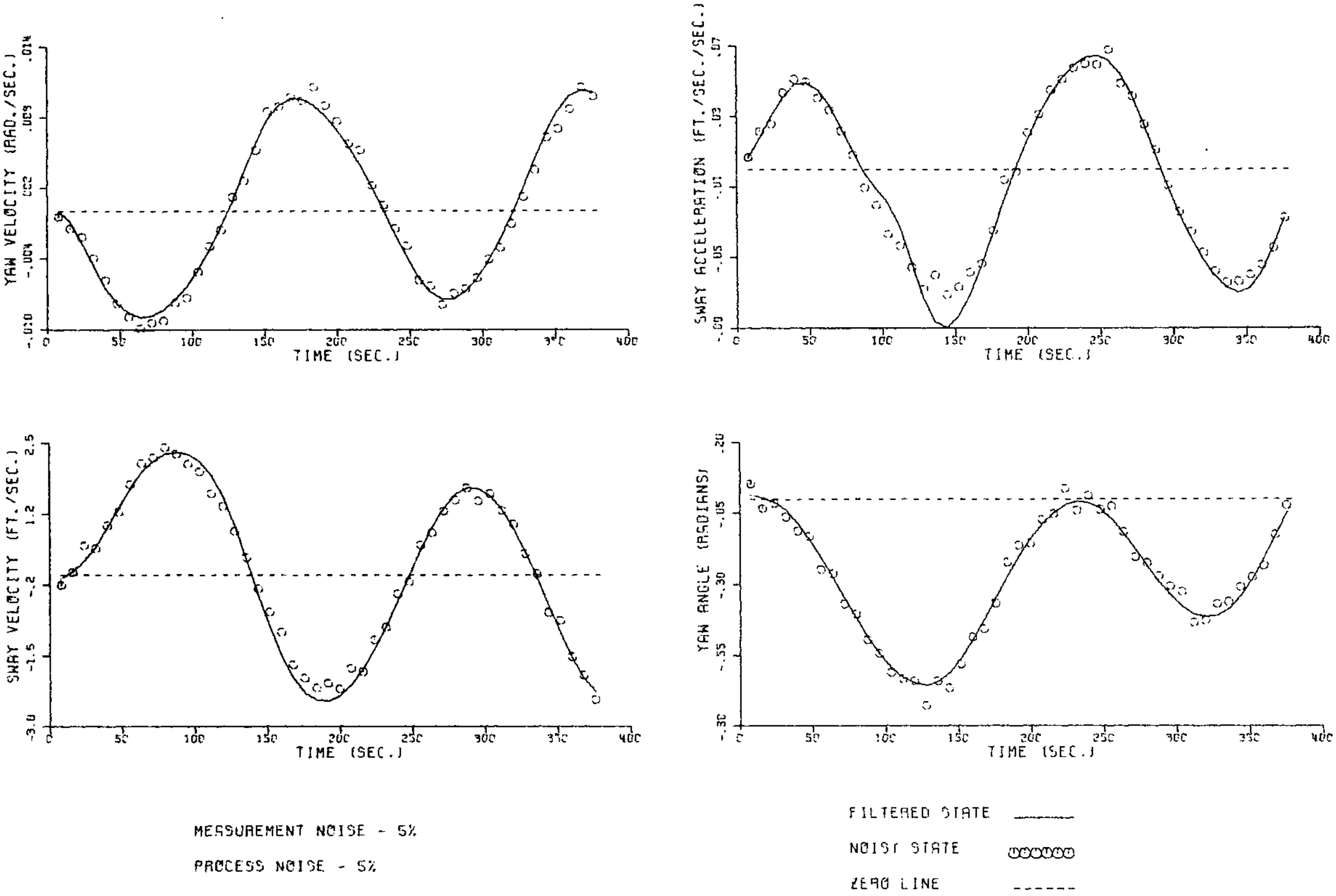


Fig. 5-9a Filtered States - Variation in Noise Exaggeration

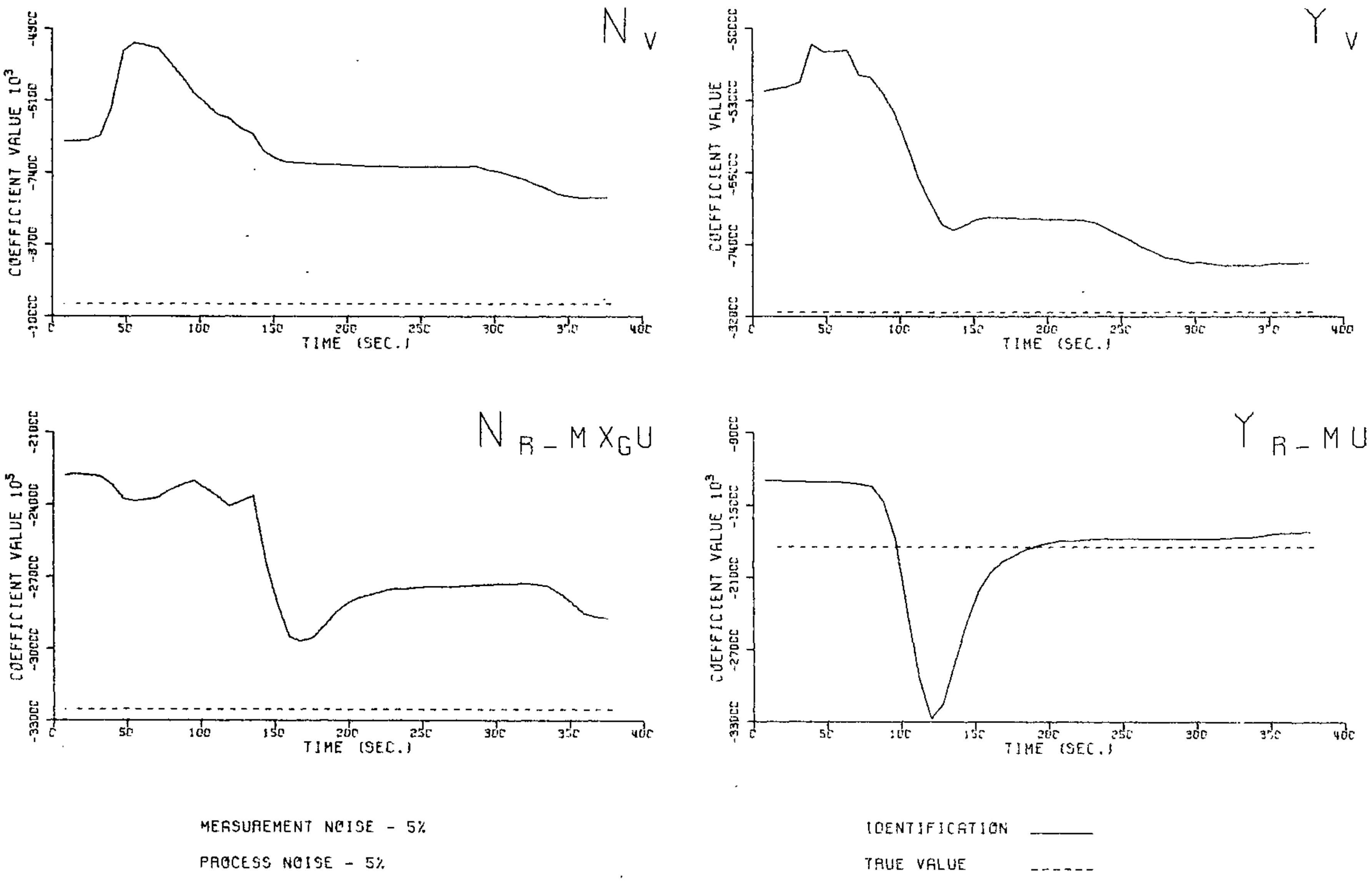


Fig. 5-9b Coefficients - Variation in Noise Exaggeration

```
NP =  13     TRUE VALUE  =   -0.32510E+10
SV  =   -0.22752D+10  + OR -    0.97581D+09
FV  =   -0.28711E+10  + OR -    0.15690E+09

IDENTIFICATION WITHIN 11.69% OF THE TRUE VALUE.
```

$(N_r - mx_G u)$

```
NP =  12     TRUE VALUE  =   -0.97735E+07
SV  =   -0.68414D+07  + OR -    0.29321D+07
FV  =   -0.78529E+07  + OR -    0.71381E+06

IDENTIFICATION WITHIN 19.65% OF THE TRUE VALUE.
```

N_v

```
NP =   7     TRUE VALUE  =   -C.18508E+C8
SV  =   -0.12955D+08  + OR -    0.55525D+07
FV  =   -0.17238E+08  + OR -    0.62796E+06

IDENTIFICATION WITHIN  6.86% OF THE TRUE VALUE.
```

$(Y_r - mu)$

```
NP =   6     TRUE VALUE  =   -0.81515E+05
SV  =   -0.57060D+05  + OR -    0.24454D+05
FV  =   -0.76117E+05  + OR -    0.32170E+04

IDENTIFICATION WITHIN  6.62% OF THE TRUE VALUE.
```

Y_v

Table 5-9b Coefficient Identification for the Variation in Noise Exaggeration

Chapter VI

SUMMARY AND CONCLUSIONS

This study was concerned with the application of a particular technique in systems identification to ship maneuvering analysis and ultimately to ship design.

The technique used here was an extension of the Kalman statistical filter. Statistical filtering is a powerful method of systems identification which provides estimates on the state of the system based on both statistical and physical properties of that system. Kalman developed an optimum linear filter to be used in the identification of linear systems. Brock showed how this linear derivation could be extended to the non-linear case. It was this non-linear extension of Kalman's filter that was used in this work.

The specific system under consideration was a surface vessel operating in the horizontal plane without roll, with particular application to the Mariner-class hull form. The model used for describing systems of this type was developed by Abkowitz as an extension of Newton's Laws of Motion using the vector calculus, and an expansion of the hydrodynamic forces present into a Taylor series. This resulted in a set of partial differential equations describing each motion component of the ship. The general structure of these equations is specified for all systems; the coefficients to these equations are determined by the type of system under observation.

The problem statement for this thesis concerned the identification of these constant coefficients, given the general structure of the equations describing the motions of the system, and noisy recorded responses for the system under maneuvering.

Traditional methods exist for identifying the hydrodynamic coefficients. Primarily these consist of either specification from hydrostatic and hydrodynamic theory, or estimation from model tests done in the towing tank. Theory is capable of calculating only a few of the coefficients. Model tests, because of scale effects inherent to this type of testing, are incapable of accurately determining all coefficient values, particularly the non-linear terms.

By applying the systems identification methodology described here to the Mariner-class vessel, very good results were obtained for the identification of the hydrodynamic coefficients. For numerous cases, under a variety of conditions, results to within at least 10 % of the accepted values were obtained. By carefully specifying the conditions under which the maneuver was run and the filter activated, the estimates to the coefficient values were within 1% - 2% of their true values.

The conditions to be used in a maneuver and the subsequent filtering of the noisy data have a very strong effect upon the final results in the identification. Thus, in this study it was shown that essentially no identification resulted from running the simulated trial with a single-step rudder deflection. However, by changing to a sinusoidal rudder deflection, keeping all other conditions constant resulted in excellent accuracy for the identification.

A discussion of the pertinent results appears in Chapter V. However, several of the more important observations shall be repeated.

For the limited analysis given here, it was found that the sinusoidal maneuver gave the consistently better results. This verifies the result suggested by Hayes, who felt that the best period of the sinusoidal maneuver was one which approximated the "natural frequency" of the system.

It was noted that the filter is able to operate with reasonable success under large (25%) amounts of noise in the data. This is encouraging as it indicates the strength of the method operating under adverse experimental conditions.

The ability to repeat the filtering process on previously updated estimates, with an increase in accuracy, was shown. In conjunction with the filter's ability to operate on very noisy data, improving the noisy estimates, this presents a very powerful tool for the naval architect analyst. Once having the noisy data on tape, he is able to reprocess the data numerous times, under a variety of different conditions. By so doing, it becomes a simple matter of observing that value which is most frequently and with the highest degree of confidence attained, and thus most probably the best estimate for that coefficient.

The work done on this project centered about the development of the computer program listed in the Appendix, applying the Kalman filtering technique to an arbitrary system. The program was designed to be general enough to handle many different situations. No known bugs are in the identification program as given here. Some difficulty was encountered in operating on eight coefficients at one time in preparing the final runs for this paper. However, it was felt that this was caused more by the choice of operating conditions and measured state variables than from any fault in the algorithm. Previous results, using u,v, and r, had shown the program to be capable of operating with reasonable accuracy on any number of chosen coefficients.

Which way now? There remains much to do on this project. Initially, all the equations used here should be checked and verified. A repetition of Brinati's preliminary analysis on which coefficients to include in the model should also be done. For this study, these results were accepted as they were and not checked.

Once the program's structure has been verified, there remains a detailed analysis of the different conditions under which a trial should be run, determining what set of conditions best suits the identification desired. Additionally, some efforts should be spent to incorporate roll into the general motion equations, giving the model a complete generality for most vessels operating in the horizontal plane.

Finally, an identification should be run on real data, not the simulated type used here. This would mean either full-scale sea-trial data or model test results. A more accurate picture of this system's true capabilities would surface under realistic conditions such as these. The method has been developed and shown to work with reasonably good results on simulated noisy data. It remains to see how well it can operate in various real situations, either in design or operation.

Appendix A

PROGRAM DESCRIPTION

MAIN

Remarks:

MAIN is the primary calling routine of the identification program. It serves three purposes. Initially, it inputs all necessary data and specifies the format and type of most variables used in the program. A detailed listing and description of the input data can be found in Appendix C.

Secondly, MAIN organizes the necessary data for later use, initializing the variables and generating noisy sea-trial responses. The necessary data is then fed into the filter and processed.

In the end, MAIN specifies the output modes of the filtered results, both graphically and tabular.

Notable Variables (MAIN):

LP - Actual number of elements within the extended state vector

SP - Maximum number of elements possible within the extended state vector, assuming identification of all coefficients

TS(N) - Time values for all measurements

IST(NO) - Starting values of the primary state variables

ICV(NO) - Estimated covariance of the starting values of the primary state variables

ZV(N), ZR(N), ZPS(N), ZVD(N) - Measured noisy output of the primary state variables generated in subroutine RKL

VP(94), RP(94), PSP(94), VDP(94) - Filtered primary state trajectories, for plotting

VEIV(94), VEIR(94), VPS(94), VVD(94) - Noisy primary state trajectories, for plotting

PP1(94), FP2(94), ••• - Stored arrays of the coefficient values as a function of time during the identification process, for plotting

Subroutines and Function Subprograms Required:

RKL, SETUP, FILTER, PLOTM, SHOMO, SHOCO, ABS

SUBROUTINE SETUP

Remarks:

Subroutine SETUP is used to assign initial values to most of the variables and matrices used in the filtering routine. It also defines the noise covariance matrices, Q and R, while introducing the exaggerated noise factors, if applicable. This allows one to tell the filter that the amount of noise in the measured data is higher than is actually the case. The filter reacts accordingly and does not have as high a degree of confidence in it's values. This prevents the filter from zeroing in on a value for the coefficients as quickly as it might. Especially for those cases with very low noise magnitudes, this provides stability to the filter and more valid results.

Areas of Interest:

(0174 - 0184) - The noise covariance values for the different state variables are defined as the square of the desired standard deviation of the additive noise. These covariances are multiplied by the relavent exaggeration factors.

Notable Variables (SETUP):

A(36) - True values of the hydrodynamic coefficients

A1(36) - Estimated values of the hydrodynamic coefficients

ASD(36) - Estimated standard deviation of the estimated coefficient values

PMS(2LP) - Standard deviation and mean of the noise distribution for the individual state variables

IST(NO) - Starting values of the primary state variables

ICV(NO) - Estimated covariance of the starting values of the primary state variables

XHT(SP) - Extended state vector

XBAR(SP) - Transfer vector for the extended state vector

EHT(SP,SP) - Error covariance matrix

HZ(NO,SP) - Measurement function

Q(LP,LP) - Process noise covariance matrix

R(LP,LP) - Measurement noise covariance matrix

PA1, PA2, ••• - True values of the coefficients

LP1, LP2, ••• - Integer designation of the coefficients to be identified

PW - Exaggeration factor, measurement noise

QW - Exaggeration factor, process noise

Subroutines and Function Subprograms Required:

none

SUBROUTINE RKL

Remarks:

Subroutine RKL generates the noisy sea-trial motion trajectories to be filtered in the identification process. The motion equations were developed in the form

$$\dot{\underline{x}} = \underline{f}(\underline{x}, t) + \underline{w}$$

The form of $\underline{x}$ can be determined by integration of $\dot{\underline{x}}$ over time. The method used in this work was the Runge-Kutta 4th order technique,(9) a method similar in many respects to Simpson's Rule.

$$\underline{x}_{k+1} \simeq \underline{x}_k + \frac{1}{6} (\underline{b}_1 + 2\underline{b}_2 + 2\underline{b}_3 + \underline{b}_4)$$

where,

$$\underline{b}_1 = \Delta t * \underline{f}(\underline{x}_k, t_k)$$

$$\underline{b}_2 = \Delta t * \underline{f}(\underline{x}_k + \frac{1}{2} \underline{b}_1, t_k + \frac{1}{2} \Delta t)$$

$$\underline{b}_3 = \Delta t * \underline{f}(\underline{x}_k + \frac{1}{2}\underline{b}_2, t_k + \frac{1}{2}\Delta t)$$

$$\underline{b}_4 = \Delta t * \underline{f}(\underline{x}_k + \underline{b}_3, t_k + \Delta t)$$

The integration of $\dot{\underline{x}}$ can thus be broken into four distinct phases for each time step. New additive process noise values are generated for each phase and added to the function which is then used in the following phase. Noiseless values are calculated for each phase to be used in calculating the time derivatives of the state variables for the next phase. This assures that the noise added will not be in excess of that specified in the input. If not, the noise would accumulate over each successive phase. The noise distribution at the end of the trial would then be dependent upon the preceeding values.

After the function has been integrated over each time increment, the measurement noise for that step is generated and added to the state value to be stored as the output value of an imaginary measurement devise. To simulate the use of the integrating accelerometer, the noisy acceleration is integrated by a simple geometric method,

$$\underline{x}_k = \underline{x}_{k+1} + \frac{1}{2}(\dot{\underline{x}}_k + \dot{\underline{x}}_{k+1})\Delta t$$

yielding the velocity. This is then given additional measurement noise.

Areas of Interest:

(0065 - 0068) - Generate process noise for the first phase of the integration over the time step. The generated noise will have a Gaussian distribution and characteristics as specified.

(0083 - 0086) - Calculate the noisy state variable values. These will be averaged over the interval and used as the actual output of the period.

(0181 - 0185) - Generate the measurement noise as a Gaussian distribution, with characteristics as specified.

(0188 - 0195) - Calculate the noisy output of the measurement devise,

$$\underline{z} = \underline{x} + \underline{v}$$

Notable Variables (RKL):

ZV(N), ZR(N), ZPS(N), ZVD(N) - Measured noisy output of the primary state variables

P(2LP) - Standard deviation and mean of the noise distribution for the individual state variables

IN(LP) - Random odd integer values used in generating random numbers by RANDU

U(t) - Rudder deflection at time t, $\delta(t)$

DUD(t) - Time rate of change of the rudder deflection at time t, $\dot{\delta}(t)$

DV - Incremental sway acceleration, $\dot{v}$

DR - Incremental yaw acceleration, $\dot{r}$

YV_, YR_, YPS_, YVD_ - Intermediate noisy state variables

YV_N, YR_N, YPS_N, YVD_N - Intermediate noisy state variables

WL - Process noise, $\underline{w}$

VL - Measurement noise, $\underline{v}$

N - Number of increments (measurements, time steps) over the entire period of observation

H - Time increment

NO - Number of primary state variables

<u>Subroutines and Function Subprograms Required</u>:

WNO, U, DUD, FNLV, FNLR

FUNCTION U

Remarks:

Function U specifies the rudder deflections as a function of time, depending on the type of maneuver used in the trial.

$$U(t) = \delta(t)$$

Areas of Interest:

(0016 - 0017) - Step rudder deflection,

(0021 - 0028) - Zig-zag rudder deflection ,

(0032 - 0034) - Sinusoidal rudder deflection,

?

Notable Variables (U):

JJ - Integer specifying the type of maneuver:

1 - Step rudder deflection

2 - Zig-zag maneuver

3 - Sinusoidal maneuver

DI - Magnitude of the maximum rudder deflection in any maneuver over the trial period

T - Time of observation

TL - Half-period of the sinusoidal deflection

Subroutines and Function Subprograms Required:

SIN

FUNCTION DUD

Remarks:

Function DUD calculates the time rate of change of the rudder deflection.

$$DUD(t) = \dot{\delta}(t)$$

This is primarily used in those situations where the extended state vector includes acceleration variables. For these cases, where subroutine EFNT1 calculates the matrix,

$$B(LP,LP) = \frac{\partial \underline{f}(\underline{x},t)}{\partial \underline{x}}$$

it is necessary to calculate the time rate of change for these accelerations. It is here that the additional term, $\dot{\delta}(t)$, appears. (see EFNT1) It should be apparent that $\delta(t)$ need be continuous over time, and therefore only rudder deflections such as

$$\delta(t) = \sin \omega t$$

can be implemented. Thus, the step rudder deflection (zigzag) with it's discontinuous function of δ can not be used in conjunction with the identification of any accelerations.

Subroutines and Function Subprograms Required:

COS

SUBROUTINE WNO

Remarks:

Subroutine WNO transfers the specified mean and standard deviation for a desired noise level into subroutine GAUSS.

It then returns the generated random Gaussian white noise to the calling routine.

Notable Variables (WNO):

IN(LP) - Odd random integer values for use by RANDU in generating a set of random numbers

AM - Desired mean noise level

S - Desired standard deviation of the Gaussian noise

P(2LP) - Mean and standard deviation for the noise levels of those primary state variables used in the extended state vector

W - Generated Gaussian white noise returned to the calling routine

Subroutines and Function Subprograms Required:

GAUSS

SUBROUTINE GAUSS

Remarks:

Subroutine GAUSS is basically the same as that offered by the IBM Scientific Subroutine Package. It computes a normally distributed random number with the given mean and standard deviation. Details on it's theory can be obtained from the literature.(19)

Subroutines and Function Subprograms Required:

RANDU

SUBROUTINE RANDU

Remarks:

Subroutine RANDU computes a uniformily distributed random number between 0.0 and 1.0 for use by subroutine GAUSS. It also is part of the IBM Scientific Subroutine Package.

Subroutines and Function Subprograms Required:

none

FUNCTION FNLV and FUNCTION FNLR

Remarks:

Functions FNLV and FNLR calculate the sway and yaw accelerations for the system at any time t.

$$\begin{bmatrix} \dot{v} \\ \dot{r} \end{bmatrix} = \underline{f}(\underline{x}, t)$$

Notable Variables (FNLV, FNLR):

XV - Sway velocity, v

XR - Yaw velocity, r

U - Rudder deflection, δ

A(36) - Hydrodynamic coefficients of the motion equations

Subroutines and Function Subprograms Required:

none

SUBROUTINE SHOMO

Remarks:

Subroutine SHOMO is one of the options available for showing the output of the identification data. It will call certain CALCOMP routines, both standard and MIT supplied, to plot the motion trajectories, noisy and filtered, as a function of time. It also plots a zero line and prints a key for the various plots. It is not internally adjustable for variations in the input data, nor is it usable on all systems. For this reason it will not be detailed. By setting one of it's parameters, K, to one, the option may be by-passed. It is strongly recommended that this routine be used only for final reports, theses, etc., when it's increased accuracy and readability will compensate for the added cost. Otherwise, it may mean your head. The run time to plot four state variables is approximately four minutes at a cost of $10.00 per hour. The other option, PLOTM, costs approximately $0.05 per plot and is perfectly adequate for production runs when only trends in the identification need be shown.

SUBROUTINE SHOCO

Remarks:

Subroutine SHOCO is the CALCOMP option available to produce plots of the identification of the individual coefficients as a function of time. It also plots the true value of the coefficient, as well as a key to the various plots produced. As with subroutine SHOMO, by setting K to one, the option can be by-passed. The same remarks given to SHOMO apply to SHOCO and need not be repeated.

SUBROUTINE PLOTM

Remarks:

Subroutine PLOTM is one of two options available for graphically portraying the results of the identification.

It is substantially less expensive than the alternative option given in SHOMO and SHOCO. However, the accuracy is limited and is best suited to show trends in the identification or motion trajectories.

The results are plotted on a size (47x51) which is compatible to thesis use. The plot scale variables are formated as F5.2, thus some care must be taken to conform to these specifications. Difficulty will not be encountered when the trial length is 188, 376 or 752 seconds as used in this work.

This routine was modified from the IBM Scientific Subroutine PLOT.(19)

Notable Variables (PLOTM):

NO - Numerical label for the graph, appearing at the head of the plot and consisting of no more than three digits

A - Matrix to be plotted, in single column form. The first N elements form the base vector, while the remaining (M-1)N elements specify the (M-1) cross vectors, where (M-1) $\leq$ 9 representing a maximum of nine plots on one graph

N - Number of rows in matrix A, N $\leq$ 47

M - Number of columns in matrix A, M $\leq$ 10

NS - Code for sorting the base variable data into ascending order,

0 - Data already sorted

1 - Sorting necessary

NE - Code for by-passing PLOTM and using an alternative routine,

1 - PLOTM to be used

2 - CALCOMP to be used

3 - Both routines used simultaneously

Subroutines and Function Subprograms Required:

none

SUBROUTINE FILTER

Remarks:

Subroutine FILTER is the statistical filtering routine of this program. The flow of the routine is identical to that described in Chapter IV. The steps may be summarized as follows.

1. Propagate the estimated state and error covariance matrices,

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

$$\underline{\hat{x}} = \int_{t_1}^{t_2} \underline{\dot{\hat{x}}}\ dt$$

$$\dot{E} = BE + EB^T + Q$$

$$E = \int_{t_1}^{t_2} \dot{E}\ dt$$

2. Calculate the gain for the Kalman filter

$$K = EH^T\ (HEH^T + R)^{-1}$$

3. Update the state and error covariance matrices

$$\underline{z}_m = H\underline{\hat{x}}$$

$$\underline{\hat{x}}' = \underline{\hat{x}} + K(\underline{z} - \underline{z}_m)$$

$$E' = E - KHE$$

The process is repeated for each time increment, with occassional values stored for use later in the plotting routines. The number of values saved corresponds to the number of points to be plotted (47 for this work).

<u>Subroutines and Function Subprograms Required</u>:

PROP, GAIN, UPDT, STORB, U, DUD

SUBROUTINE PROP

Remarks:

Subroutine PROP is used to propagate the estimated state and error covariance matrices over the desired time increment.

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

$$\underline{x} = \int_{t_1}^{t_2} \dot{\underline{x}}\, dt$$

$$E = \int_{t_1}^{t_2} \dot{E}\, dt$$

The time rate of change of the error covariance matrix is called from subroutine EFNT2 where it is calculated in conjunction with subroutine EFNT1. This routine, PROP, is basically the same as RKL, with the integration again done using the Runge-Kutta 4th order technique.

Subroutines and Function Subprograms Required:

FNLV, FNLR, EFNT1, EFNT2

SUBROUTINE EFNT1

<u>Remarks</u>:

Subroutine EFNT1 calculates the matrix,

$$B(LP,LP) = \frac{\partial \underline{f}}{\partial \underline{x}} (\underline{x},t)$$

$$= \frac{\partial \dot{\underline{x}}}{\partial \underline{x}}$$

which is then used by subroutine EFNT2 to calculate the time rate of change of the error covariance matrix, $\dot{E}$. The first subscript indicates the row and the relevant equation, $\underline{f}(\underline{x},t)$, to be differentiated. The second subscript refers to the column and the element of the extended state vector with which the differentiation is taken. The notation can be defined as,

$$\begin{bmatrix} K1 \\ K2 \\ K3 \\ K4 \\ K5 \\ K6 \\ K7 \\ K8 \\ K9 \end{bmatrix} = \begin{bmatrix} f_1(\underline{x},t) \\ f_2(\underline{x},t) \\ f_3(\underline{x},t) \\ f_4(\underline{x},t) \\ f_5(\underline{x},t) \\ f_6(\underline{x},t) \\ f_7(\underline{x},t) \\ f_8(\underline{x},t) \\ f_9(\underline{x},t) \end{bmatrix} = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{r} \\ x_o \\ y_o \\ \psi_o \\ \ddot{u} \\ \ddot{v} \\ \ddot{r} \end{bmatrix}$$

where,

$$\begin{Bmatrix} f_1 \\ f_2 \\ f_3 \end{Bmatrix} = \begin{Bmatrix} \text{equations developed in} \\ \text{Chapter III} \end{Bmatrix}$$

$$\begin{Bmatrix} f_4 \\ f_5 \\ f_6 \end{Bmatrix} = \begin{Bmatrix} u \cos \psi - v \sin \psi \\ u \sin \psi + v \cos \psi \\ r \end{Bmatrix}$$

$$\begin{Bmatrix} f_7 \\ f_8 \\ f_9 \end{Bmatrix} = \begin{Bmatrix} \frac{\partial \underline{f}_1}{\partial t} \\ \frac{\partial \underline{f}_2}{\partial t} \\ \frac{\partial \underline{f}_3}{\partial t} \end{Bmatrix}$$

The elements of the extended state vector depend upon which state variables and coefficients are included in the identification.

Areas of Interest:

(0034 - 0056) - The common parts found in many of the derivatives are computed once and assigned to separate variables.

(0039 - 0042) - The position of the elements of the primary state variables are designated, using the notation given above. For example, K2 = 1 and K3 = 2 implies that the first two elements of the primary state vector are v and r. Therefore, the derivatives of functions K2 ($\dot{v}$) and K3 ($\dot{r}$) occupy the first two rows of the B matrix.

(0061 - 0067) - The derivatives of the functions of the state variables are taken with respect to those state variables, and comprise the initial (NOxNO) submatrix of B.

(0079 - 0209) - The derivatives of the functions of the state variables are taken with respect to the coefficients to be identified. These coefficients are found in the (NO+1) to (NO+MP) elements of the extended state vector, and thus the corresponding columns of the B matrix.

<u>example</u> -

let K3 = 2, I = LP2, LP2 = 7

then,

$$B(K3,I) = \frac{\partial K3}{\partial A(LP2)}$$

$$= \frac{\partial \dot{r}}{\partial A(7)}$$

Subroutines and Function Subprograms Required:

none

SUBROUTINE EFNT2

Remarks:

Subroutine EFNT2 is used in conjunction with subroutine EFNT1 and calculates the time rate of change of the error covariance matrix.

$$\dot{E} = BE + EB^T + Q$$

Subroutine PROP then takes the result and propagates it over the time interval, arriving at a new value of the error covariance matrix, E.

Notable Variables (EFNT2):

EH(SP,SP) - Error covariance matrix

B(LP,LP) - Matrix of the partial derivatives of the extended state vector from subroutine EFNT1

Q(LP,LP) - Process noise covariance matrix

E1(LP,LP) - Time rate of change of the error covariance matrix, $\dot{E}$

Subroutines and Function Subprograms Required:

TRNSPS, MAMP1S, MAMP2S, MAADDS

SUBROUTINE GAIN

Remarks:

Subroutine GAIN is the key routine of the identification process since it calculates the gain of the Kalman filter.

$$K = EH^T (HEH^T + R)^{-1}$$

The gain is then used in subroutine UPDT to update the estimates for both the state and error covariance matrices.

Notable Variables (GAIN):

EB(LP,LP) - Error covariance matrix

H(NO,SP) - Measurement function

K(LP,NO) - Kalman filter gain matrix

C(NO^2) - One-dimensional array containing the elements of the two-dimensional matrix ($HEH^T + R$), for inversion by subroutine MINV

Subroutines and Function Subprograms Required:

TRNSPS, MAMP1S, MAADDS, MINV

SUBROUTINE UPDT

Remarks:

After the estimated state and error covariance matrices have been propagated over the time step and the filter gain has been calculated, subroutine UPDT is called to update the estimates to their values at the end of the increment.

$$\underline{z}_m = H\underline{x}$$

$$\underline{x}' = \underline{x} + K(\underline{z} - \underline{z}_m)$$

$$E' = E - KHE$$

Notable Variables (UPDT):

H(NO,SP) - Measurement function

EB(LP,LP) - Error covariance matrix at the beginning of the time increment

EH(SP,SP) - Updated error covariance matrix

XB(SP) - Estimated extended state vector at the beginning of the time increment

XH(SP) - Updated extended state vector at the end of the time increment

Z(NO) - Value of the primary state variables during the noisy sea-trial

EL(LP) - Measured value of the primary state variables, taken from the estimated state vector operated upon by the measurement function

ES(LP) - Difference between the actual and estimated values of the measured state variables

Subroutines and Function Subprograms Required:

MAMP1S,,MAMP2S, MASUBS, DABS

SUBROUTINE STORB

Remarks:

Subroutine STORB is used to store selected values of the error covariance and state matrices at regular intervals during the identification process. These values are coupled to the time of the reading during the trial and plotted by either PLOTM, or SHOMO and SHOCO. The standard deviations are derived from the diagonal elements of the covariance matrix at the end of the trial. These are used as a representation of the confidence level in the final results.

Notable Variables (STORB):

I - Index of the stored measurement

L - Actual index of the time matrix for the stored measurement

KS - Number of measurements between stored values

PP1(94), PP2(94), ••• - One-dimensional array of the identification of the individual coefficients. The first 47 elements are the times for the corresponding stored measurements, while the remaining 47 elements are the measured values.

EP1, EP2, ••• - Standard deviation of the final value assigned to each coefficient

EE(SP) - Values of the diagonal elements of the covariance matrix

Subroutines and Function Subprograms Required:

DSQRT, DABS

SUBROUTINE TRNSPS

Remarks:

Subroutine TRNSPS is capable of taking the transpose of a matrix A and placing it in a matrix B, leaving matrix A unchanged.

$$A^T \longrightarrow B$$

Matrix A has absolute dimensions IA x JA, while matrix B has absolute dimensions IB x JB. The submatrix of A to be transposed has dimensions of MA x NA. The transposed submatrix in B has dimensions of NA x MA.

This routine is part of the WATFIV library.(20)

Subroutines and Function Subprograms Required:

none

SUBROUTINE MAMP1S and SUBROUTINE MAMP2S

Remarks:

Subroutines MAMP1S and MAMP2S are used to multiply two matrices, A and B, with the product replacing one of the matrices depending upon which routine is called, leaving the other unchanged.

$$A * B \longrightarrow A \qquad \text{(MAMP1S)}$$

$$A * B \longrightarrow B \qquad \text{(MAMP2S)}$$

The absolute dimensions of A are IA x JA, while those of matrix B are IB x JB. The actual multiplication involves the submatrix of A with dimensions MA x NAMB and the submatrix of B with dimensions NAMB x NB. W is a work vector at least as large as MA in MAMP2S or NB in MAMP1S. (WATFIV library)

Subroutines and Function Subprograms Required:

none

SUBROUTINE MAADDS

Remarks:

Subroutine MAADDS adds the elements of matrix A to matrix B, replacing A by the resultant sum and leaving B unchanged.

$$A + B \longrightarrow A$$

The actual size of the added submatrices is MA x NA. The absolute dimensions of A and B are IA x JA and IB x JB respectively. (WATFIV library)

Subroutines and Function Subprograms Required:

none

SUBROUTINE MASUBS

Remarks:

Subroutine MASUBS subtracts matrix B from matrix A, placing the difference in A and leaving B unchanged.

$$A - B \longrightarrow A$$

The absolute dimensions of matrices A and B are IA x JA and IB x JB respectively. The actual size of the subtracted submatrix is MA x NA. (WATFIV library)

Subroutines and Function Subprograms Required:

none

SUBROUTINE MINV

Remarks:

This routine uses the standard Gauss-Jordan method for inversion of matrices. Matrix A is inverted and replaced by it's inverse.

$$A^{-1} \longrightarrow A$$

Calculation of the resulting determinant, D, indicates wether or no the resultant matrix is singular (D = 0). Matrix A is a general matrix of order N. D is the resultant determinant, while L and M are work vectors of length N.

Subroutine MINV is part of the IBM Scientific Subroutine Package.(19)

Subroutines and Function Subprograms Required:

none

Appendix B

THE PROGRAM

"Lasciate ogni speranza voi ch'entrate ... "

Inscription over the entrance to Hell,
in Dante's _Inferno_

```
      INTEGER SP
      DIMENSION VP(94),RP(94),PSP(94),VDP(94)
      DIMENSION PP1(94),PP2(94),PP3(94),PP4(94),PP5(94),PP6(94),PP7(94)
      DIMENSION PP8(94),PP9(94),PP10(94),PP11(94),PP12(94),PP13(94)
      DIMENSION PP14(94),PP15(94),PP16(94)
      DIMENSION PP17(94),PP18(94),PP19(94),PP20(94),PP21(94),PP22(94)
      DIMENSION PP23(94),PP24(94),PP25(94),PP26(94),PP27(94),PP28(94)
      DIMENSION PP29(94),PP30(94),PP31(94),PP32(94),PP33(94),PP34(94)
      DIMENSION PP35(94),PP36(94)
      DIMENSION VEIV(94),VEIR(94),VPS(94),VVD(94)
      DIMENSION INX(8)
      DIMENSION IC(4),IR(4)
      DIMENSION EE(40)
      DIMENSION ZERO(47)
      DIMENSION AV(47),AR(47),APS(47),AVD(47)
      DIMENSION AVP(47),ARP(47),APSP(47),AVDP(47)
      DIMENSION P1(47),P2(47),P3(47),P4(47)
      DIMENSION P5(47),P6(47),P7(47),P8(47)
      DIMENSION TITLE(4,9),YLAB1(9),YLAB2(9),YLAB3(9),YLAB4(9)
      DOUBLE PRECISION H,DI,PW,QW,TL,TI
      DOUBLE PRECISION A(36),AI(36),ASD(36)
      DOUBLE PRECISION PMS(16)
      DOUBLE PRECISION W(8)
      DOUBLE PRECISION WN(4),VN(4)
      DOUBLE PRECISION V(376)
      DOUBLE PRECISION ZV(376),ZR(376),ZPS(376),ZVD(376)
      DOUBLE PRECISION Z(4)
      DOUBLE PRECISION TS(376),US(377),DUS(377)
      DOUBLE PRECISION K(8,4),Q(8,8),R(8,8),B(8,8),EBAR(8,8)
      DOUBLE PRECISION EHT(40,40),XHT(40),XBAR(40),HZ(4,40)
      DOUBLE PRECISION XVI,XRI,XPSI,XVDI
      DOUBLE PRECISION VST,RST,PST,VDST
      DOUBLE PRECISION VCV,RCV,PCV,VDCV
      DOUBLE PRECISION EJ(8,8),E1(8,8),E2(8,8),E3(8,8),E4(8,8)
      DOUBLE PRECISION E5(8,8),EN(8,8),Q1(8,8),BN(8,8),FT(8,8)
      DOUBLE PRECISION EM(8,8),H1(4,8),T(8,4),H2(4,8),H3(8,8)
      DOUBLE PRECISION EL(8),ES(8)
      DOUBLE PRECISION IST(4),ICV(4)
      DOUBLE PRECISION C(16)
      COMMON /OUTP5/ PP17,PP18,PP19,PP20,PP21,PP22,PP23,PP24,PP25,PP26
      COMMON /OUTP6/ PP27,PP28,PP29,PP30,PP31,PP32,PP33,PP34,PP35,PP36
      COMMON /OUTP1/ VP,RP,PSP,VDP,PP1,PP2,PP3,PP4,PP5,PP6,PP7,PP8
      COMMON /OUTP2/ PP9,PP10,PP11,PP12,PP13,PP14,PP15,PP16
      COMMON /PRM1/ PA1,PA2,PA3,PA4,PA5,PA6,PA7,PA8,PA9,PA10,PA11,PA12
      COMMON /OUTP3/ EV,ER,EPS,EVD,EP1,EP2,EP3,EP4,EP5,EP6,EP7,EP8
      COMMON /OUTP7/ EP17,EP18,EP19,EP20,EP21,EP22,EP23,EP24,EP25,EP26
      COMMON /OUTP8/ EP27,EP28,EP29,EP30,EP31,EP32,EP33,EP34,EP35,EP36
      COMMON /OUTP4/ EP9,EP10,EP11,EP12,EP13,EP14,EP15,EP16
      COMMON /PRAM3/ LP17,LP18,LP19,LP20,LP21,LP22,LP23,LP24,LP25,LP26
      COMMON /PRAM4/ LP27,LP28,LP29,LP30,LP31,LP32,LP33,LP34,LP35,LP36
      COMMON /PRM2/ PA13,PA14,PA15,PA16,PA17,PA18,PA20,PA21,PA22,PA23
      COMMON /PRM3/ PA24,PA25,PA26,PA27,PA28,PA29,PA30,PA31,PA32,PA33
      COMMON /PRAM1/ LP1,LP2,LP3,LP4,LP5,LP6,LP7,LP8
      COMMON /PRAM2/ LP9,LP10,LP11,LP12,LP13,LP14,LP15,LP16
      COMMON /EXAG/ PW;QW
      COMMON /INPUT/ DI,TL,JJ
      COMMON /PKI/G
      COMMON /PRM4/ PA34,PA35,PA36
C
C *********
C
      CALL PLOTS(IDUM,IDUM,11)
      CALL FACTOR(0.5)
      CALL PLOT(0.0,2.0,-3)
    1 CONTINUE
      KI = 5
      KO = 6
C
C     INPUT DATA
C
      READ (KI,10) (A(I),I = 1,36)
      IF (A(1).EQ.0.0) GO TO 909
      READ (KI,10) (AI(I),I = 1,36)
      READ (KI,10) (ASD(I),I = 1,36)
   10 FORMAT (6D13.4)
      READ (KI,11) (PMS(J),J = 1,16)
   11 FORMAT (6D13.4)
      READ (KI,12) (INX(I),I = 1,8)
   12 FORMAT (6I6)
      READ (KI,13) G
   13 FORMAT (F10.6)
      READ (KI,14) LP1,LP2,LP3,LP4,LP5,LP6,LP7,LP8,LP9
   14 FORMAT (9I5)
      READ (KI,15) LP10,LP11,LP12,LP13,LP14,LP15,LP16,LP17,LP18
   15 FORMAT (9I5)
      READ (KI,16) LP19,LP20,LP21,LP22,LP23,LP24,LP25,LP26,LP27
   16 FORMAT (9I5)
      READ (KI,17) LP28,LP29,LP30,LP31,LP32,LP33,LP34,LP35,LP36
   17 FORMAT (9I5)
      READ (KI,19) VST,RST,PST,VDST
   19 FORMAT (4F10.5)
      READ (KI,20) VCV,RCV,PCV,VDCV
   20 FORMAT (4F10.3)
      READ (KI,21) KS,N,H
   21 FORMAT(2I4,F10.2)
      READ (KI,22) NM,NP
   22 FORMAT (2I5)
      READ (KI,23) PW,QW
   23 FORMAT (2D10.2)
      READ (KI,26) MP,NO
   26 FORMAT (2I4)
      READ (KI,27) DI,TL,JJ
   27 FORMAT (2F10.3,I5)
      READ (KI,28) NE
   28 FORMAT (I5)
      DO 790 I = 1,4
  790 READ (KI,90) (TITLE(I,J),J = 1,9)
   90 FORMAT (9A4)
```

```
      DO 791 J = 1,9
      YLAB1(J) = TITLE(1,J)
      YLAB2(J) = TITLE(2,J)
      YLAB3(J) = TITLE(3,J)
      YLAB4(J) = TITLE(4,J)
  791 CONTINUE
C
C     INITIAL CONDITIONS
C
      TI = 0.D0
      XVI = 0.D0
      XRI = 0.D0
      XPSI = 0.D0
      XVDI = 0.D0
      JON = N*H
      NN = N
C
C     GENERATE NOISY SEA TRIAL DATA FOR USE IN FILTERING
C
      CALL RKL(H,TI,XVI,XRI,XPSI,XVDI,N,A,ZV,ZR,ZPS,ZVD,US,DUS,TS,INX,
     1PMS,WN,VN,NO,V)
      NPL = 47
      DO 99 I = 1,NPL
      L = KS*I
      NL1 = I+NPL
      VEIV(I) =TS(L)
      VEIR(I) =TS(L)
      VPS(I) = TS(L)
      VVD(I) = TS(L)
      AV(I) = ZV(L)
      AR(I) = ZR(L)
      APS(I) = ZPS(L)
      AVD(I) = ZVD(L)
      VEIV(NL1) = ZV(L)
      VEIR(NL1) = ZR(L)
      VPS(NL1) = ZPS(L)
      VVD(NL1) = ZVD(L)
   99 CONTINUE
C
C     PLOT THE NOISY SEA TRIAL DATA AND INITIALIZE THE STARTING VALUES
C     OF THE PRIMARY STATES
C
      CALL PLOTM(0,VEIV,NPL,2,0,NE)
      NO = 1
      IST(NO) = VST
      ICV(NO) = VCV
      CALL PLOTM(0,VEIR,NPL,2,0,NE)
      NO = NO+1
      IST(NO) = RST
      ICV(NO) = RCV
      CALL PLOTM(0,VPS,NPL,2,0,NE)
      NO = NO+1
      IST(NO) = PST
      ICV(NO) = PCV
      CALL PLOTM(0,VVD,NPL,2,0,NE)
      NO = NO+1
      IST(NO) = VDST
      ICV(NO) = VDCV
      LP = MP+NO
      SP = NO+36
C
C     INITIALIZE THE MATRICES USED IN THE IDENTIFICATION
C
      CALL SETUP(Q,R,LP,NO,SP,EHT,IST,ICV,A,AI,ASD,PMS,XHT,XBAR,HZ,ZERO)
C
C     PERFORM THE PARAMETRIC IDENTIFICATION UPON THE SPECIFIED
C     COEFFICIENTS USING THE INITIAL CONDITIONS AND THE GENERATED
C     NOISY SEA TRIAL DATA
C
      CALL FILTER(EJ,E1,E2,E3,E4,E5,EN,EM,EL,ES,Q1,BN,FT,T,H1,H2,H3,LP,
     1KS,EBAR,B,K,Q,R,EE,NO,Z,SP,ZV,ZR,ZPS,ZVD,US,TS,XHT,XBAR,HZ,A,C,EHT
     2,IC,IR,W,H,TI,N,DUS)
```

```
      KK = 47
      KP = 94
      NS = 0
      N = 1
      M = 2
      DO 401 KMN = 1,47
      NMK = KMN+47
      AVP(KMN) = VP(NMK)
      ARP(KMN) = RP(NMK)
      APSP(KMN) = PSP(NMK)
      AVDP(KMN) = VDP(NMK)
      P1(KMN) = PP1(NMK)
      P2(KMN) = PP2(NMK)
      P3(KMN) = PP3(NMK)
      P4(KMN) = PP4(NMK)
      P5(KMN) = PP5(NMK)
      P6(KMN) = PP6(NMK)
      P7(KMN) = PP7(NMK)
      P8(KMN) = PP8(NMK)
  401 CONTINUE
C
C     PLOT THE FILTERED PRIMARY STATE VARIABLES USING THE IDENTIFIED
C     VALUES OF THE HYDRODYNAMIC COEFFICIENTS
C
      CALL PLOTM(N,VP,KK,M,NS,NE)
      CALL SHOMO(VEIV,AV,VEIV,ZERO,VEIV,AVP,-11.0,5.6,-3,NE,1,YLAB1)
      N = N+1
      CALL PLOTM(N,RP,KK,M,NS,NE)
      CALL SHOMO(VEIR,AR,VEIR,ZERO,VEIR,ARP,-1.5,-5.6,-3,NE,2,YLAB2)
      N = N+1
      CALL PLOTM(N,PSP,KK,M,NS,NE)
      CALL SHOMO(VPS,APS,VPS,ZERO,VPS,APSP,-11.0,5.6,-3,NE,3,YLAB3)
      N = N+1
      CALL PLOTM(N,VDP,KK,M,NS,NE)
      CALL SHOMO(VVD,AVD,VVD,ZERO,VVD,AVDP,5.0,-5.6,-3,NE,4,YLAB4)
      IF (MP.EQ.0) GO TO 560
```

```
C
C     PLOT THE IDENTIFICATION OF THE SPECIFIED COEFFICIENTS AS A
C     FUNCTION OF TIME DURING THE FILTERING PROCESS
C
      GO TO (101,102,103,104,105,106,107,108,109,110,111,112,113,114,
     1115,116,117,118,119,120,121,122,123,124,125,126,127,128,129,130,
     2131,132,133,134,135,136),MP
  136 N = LP36
      CALL PLOTM(N,PP36,KK,M,NS,NE)
  135 N = LP35
      CALL PLOTM(N,PP35,KK,M,NS,NE)
  134 N = LP34
      CALL PLOTM(N,PP34,KK,M,NS,NE)
  133 N = LP33
      CALL PLOTM(N,PP33,KK,M,NS,NE)
  132 N = LP32
      CALL PLOTM(N,PP32,KK,M,NS,NE)
  131 N = LP31
      CALL PLOTM(N,PP31,KK,M,NS,NE)
  130 N = LP30
      CALL PLOTM(N,PP30,KK,M,NS,NE)
  129 N = LP29
      CALL PLOTM(N,PP29,KK,M,NS,NE)
  128 N = LP28
      CALL PLOTM(N,PP28,KK,M,NS,NE)
  127 N = LP27
      CALL PLOTM(N,PP27,KK,M,NS,NE)
  126 N = LP26
      CALL PLOTM(N,PP26,KK,M,NS,NE)
  125 N = LP25
      CALL PLOTM(N,PP25,KK,M,NS,NE)
  124 N = LP24
      CALL PLOTM(N,PP24,KK,M,NS,NE)
  123 N = LP23
      CALL PLOTM(N,PP23,KK,M,NS,NE)
  122 N = LP22
```

```
      CALL PLOTM(N,PP22,KK,M,NS,NE)
  121 N = LP21
      CALL PLOTM(N,PP21,KK,M,NS,NE)
  120 N = LP20
      CALL PLOTM(N,PP20,KK,M,NS,NE)
  119 N = LP19
      CALL PLOTM(N,PP19,KK,M,NS,NE)
  118 N = LP18
      CALL PLOTM(N,PP18,KK,M,NS,NE)
  117 N = LP17
      CALL PLOTM(N,PP17,KK,M,NS,NE)
  116 N = LP16
      CALL PLOTM(N,PP16,KK,M,NS,NE)
  115 N = LP15
      CALL PLOTM(N,PP15,KK,M,NS,NE)
  114 N = LP14
      CALL PLOTM(N,PP14,KK,M,NS,NE)
  113 N = LP13
      CALL PLOTM(N,PP13,KK,M,NS,NE)
  112 N = LP12
      CALL PLOTM(N,PP12,KK,M,NS,NE)
  111 N = LP11
      CALL PLOTM(N,PP11,KK,M,NS,NE)
  110 N = LP10
      CALL PLOTM(N,PP10,KK,M,NS,NE)
  109 N = LP9
      CALL PLOTM(N,PP9,KK,M,NS,NE)
  108 N = LP8
      CALL PLOTM(N,PP8,KK,M,NS,NE)
      CALL SHOCO(PP8,P8,PP8,PA8,0.0,5.6,-3,NE,1,8)
  107 N = LP7
      CALL PLOTM(N,PP7,KK,M,NS,NE)
      CALL SHOCO(PP7,P7,PP7,PA7,9.5,-5.6,-3,NE,2,7)
  106 N = LP6
      CALL PLOTM(N,PP6,KK,M,NS,NE)
      CALL SHOCO(PP6,P6,PP6,PA6,0.0,5.6,-3,NE,3,6)
```

```
 105 N = LP5
     CALL PLOTM(N,PP5,KK,M,NS,NE)
     CALL SHOCO(PP5,P5,PP5,PA5,15.0,-5.6,-3,NE,4,5)
 104 N = LP4
     CALL PLOTM(N,PP4,KK,M,NS,NE)
     CALL SHOCO(PP4,P4,PP4,PA4,0.0,5.6,-3,NE,1,4)
 103 N = LP3
     CALL PLOTM(N,PP3,KK,M,NS,NE)
     CALL SHOCO(PP3,P3,PP3,PA3,9.5,-5.6,-3,NE,2,3)
 102 N = LP2
     CALL PLOTM(N,PP2,KK,M,NS,NE)
     CALL SHOCO(PP2,P2,PP2,PA2,0.0,5.6,-3,NE,3,2)
 101 N = LP1
     CALL PLOTM(N,PP1,KK,M,NS,NE)
     CALL SHOCO(PP1,P1,PP1,PA1,15.0,-7.9,-3,NE,4,1)
C
C    TABULATE THE PRELIMINARY INFORMATION AND THE RESULTS OF THE
C    IDENTIFICATION ANALYSIS
C
     N = 0
     WRITE (KO,557)
     WRITE (KO,550)
     TL = 2.*TL
     GO TO (140,150,160),JJ
 140 WRITE (KO,547) DI
     GO TO 70
 150 WRITE (KO,548) DI
     GO TO 70
 160 WRITE (KO,549) TL,DI
  70 CONTINUE
     WRITE (KO,553) NM,NP
     WRITE (KO,561) PW
     WRITE (KO,552) JON
     WRITE (KO,566) H
     WRITE (KO,562) NO
     WRITE (KO,563) MP
```

```
      WRITE (KO,558)
      WRITE (KO,559)
      GO TO (201,202,203,204,205,206,207,208,209,210,211,212,213,214,
     1215,216,217,218,219,220,221,222,223,224,225,226,227,228,229,230,
     2231,232,233,234,235,236),MP
  236 WRITE (KO,556) LP36,PA36,AI(LP36),ASD(LP36),PP36(KP),EP36
                T1 = ABS(((PA36-PP36(KP))/PA36)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 235
      N = 0
      WRITE (KO,559)
  235 WRITE (KO,556) LP35,PA35,AI(LP35),ASD(LP35),PP35(KP),EP35
                T1 = ABS(((PA35-PP35(KP))/PA35)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 234
      N = 0
      WRITE (KO,559)
  234 WRITE (KO,556) LP34,PA34,AI(LP34),ASD(LP34),PP34(KP),EP34
                T1 = ABS(((PA34-PP34(KP))/PA34)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 233
      N = 0
      WRITE (KO,559)
  233 WRITE (KO,556) LP33,PA33,AI(LP33),ASD(LP33),PP33(KP),EP33
                T1 = ABS(((PA33-PP33(KP))/PA33)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 232
      N = 0
      WRITE (KO,559)
  232 WRITE (KO,556) LP32,PA32,AI(LP32),ASD(LP32),PP32(KP),EP32
                T1 = ABS(((PA32-PP32(KP))/PA32)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 231
      N = 0
      WRITE (KO,559)
  231 WRITE (KO,556) LP31,PA31,AI(LP31),ASD(LP31!,PP31(KP),EP31
                T1 = ABS(((PA31-PP31(KP))/PA31)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 230
      N = 0
      WRITE (KO,559)
  230 WRITE (KO,556) LP30,PA30,AI(LP30),ASD(LP30),PP30(KP),EP30
                T1 = ABS(((PA30-PP30(KP))/PA30)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 229
      N = 0
      WRITE (KO,559)
  229 WRITE (KO,556) LP29,PA29,AI(LP29),ASD(LP29),PP29(KP),EP29
                T1 = ABS(((PA29-PP29(KP))/PA29)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 228
      N = 0
      WRITE (KO,559)
  228 WRITE (KO,556) LP28,PA28,AI(LP28),ASD(LP28),PP28(KP),EP28
                T1 = ABS(((PA28-PP28(KP))/PA28)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 227
      N = 0
      WRITE (KO,559)
  227 WRITE (KO,556) LP27,PA27,AI(LP27),ASD(LP27),PP27(KP),EP27
                T1 = ABS(((PA27-PP27(KP))/PA27)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 226
      N = 0
      WRITE (KO,559)
  226 WRITE (KO,556) LP26,PA26,AI(LP26),ASD(LP26),PP26(KP),EP26
           T1 = ABS(((PA26-PP26(KP))/PA26)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 225
      N = 0
      WRITE (KO,559)
  225 WRITE (KO,556) LP25,PA25,AI(LP25),ASD(LP25),PP25(KP),EP25
           T1 = ABS(((PA25-PP25(KP))/PA25)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 224
      N = 0
      WRITE (KO,559)
  224 WRITE (KO,556) LP24,PA24,AI(LP24),ASD(LP24),PP24(KP),EP24
           T1 = ABS(((PA24-PP24(KP))/PA24)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 223
      N = 0
      WRITE (KO,559)
  223 WRITE (KO,556) LP23,PA23,AI(LP23),ASD(LP23),PP23(KP),EP23
           T1 = ABS(((PA23-PP23(KP))/PA23)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 222
      N = 0
      WRITE (KO,559)
  222 WRITE (KO,556) LP22,PA22,AI(LP22),ASD(LP22),PP22(KP),EP22
           T1 = ABS(((PA22-PP22(KP))/PA22)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 221
      N = 0
      WRITE (KO,559)
  221 WRITE (KO,556) LP21,PA21,AI(LP21),ASD(LP21),PP21(KP),EP21
                T1 = ABS(((PA21-PP21(KP))/PA21)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 220
      N = 0
      WRITE (KO,559)
  220 WRITE (KO,556) LP20,PA20,AI(LP20),ASD(LP20),PP20(KP),EP20
                T1 = ABS(((PA20-PP20(KP))/PA20)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 219
      N = 0
      WRITE (KO,559)
  219 WRITE (KO,556) LP19,PA19,AI(LP19),ASD(LP19),PP19(KP),EP19
                T1 = ABS(((PA19-PP19(KS))/PA19)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 218
      N = 0
      WRITE (KO,559)
  218 WRITE (KO,556) LP18,PA18,AI(LP18),ASD(LP18),PP18(KP),EP18
                T1 = ABS(((PA18-PP18(KP))/PA18)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 217
      N = 0
      WRITE (KO,559)
  217 WRITE (KO,556) LP17,PA17,AI(LP17),ASD(LP17),PP17(KP),EP17
                T1 = ABS(((PA17-PP17(KP))/PA17)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 216
      N = 0
      WRITE (KO,559)
  216 WRITE (KO,556) LP16,PA16,AI(LP16),ASD(LP16),PP16(KP),EP16
                T1 = ABS(((PA16-PP16(KP))/PA16)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 215
      N = 0
      WRITE (KO,559)
  215 WRITE (KO,556) LP15,PA15,AI(LP15),ASD(LP15),PP15(KP),EP15
                T1 = ABS(((PA15-PP15(KP))/PA15)*100.)
      WRITE (KC,555) T1
      N = N+1
      IF (N.LT.4) GO TO 214
      N = 0
      WRITE (KO,559)
  214 WRITE (KO,556) LP14,PA14,AI(LP14),ASD(LP14),PP14(KP),EP14
                T1 = ABS(((PA14-PP14(KP))/PA14)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 213
      N = 0
      WRITE (KO,559)
  213 WRITE (KO,556) LP13,PA13,AI(LP13),ASD(LP13),PP13(KP),EP13
                T1 = ABS(((PA13-PP13(KP))/PA13)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 212
      N = 0
      WRITE (KO,559)
  212 WRITE (KO,556) LP12,PA12,AI(LP12),ASD(LP12),PP12(KP),EP12
                T1 = ABS(((PA12-PP12(KP))/PA12)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 211
      N = 0
      WRITE (KO,559)
```

```
211 WRITE (KO,556) LP11,PA11,AI(LP11),ASD(LP11),PP11(KP),EP11
              T1 = ABS(((PA11-PP11(KP))/PA11)*100.)
    WRITE (KO,555) T1
    N = N+1
    IF (N.LT.4) GO TO 210
    N = 0
    WRITE (KO,559)
210 WRITE (KO,556) LP10,PA10,AI(LP10),ASD(LP10),PP10(KP),EP10
              T1 = ABS(((PA10-PP10(KP))/PA10)*100.)
    WRITE (KO,555) T1
    N = N+1
    IF (N.LT.4) GO TO 209
    N = 0
    WRITE (KO,559)
209 WRITE (KO,556) LP9,PA9,AI(LP9),ASD(LP9),PP9(KP),EP9
              T1 = ABS(((PA9-PP9(KP))/PA9)*100.)
    WRITE (KO,555) T1
    N = N+1
    IF (N.LT.4) GO TO 208
    N = 0
    WRITE (KO,559)
208 WRITE (KO,556) LP8,PA8,AI(LP8),ASD(LP8),PP8(KP),EP8
              T1 = ABS(((PA8-PP8(KP))/PA8)*100.)
    WRITE (KO,555) T1
    N = N+1
    IF (N.LT.4) GO TO 207
    N = 0
    WRITE (KO,559)
207 WRITE (KO,556) LP7,PA7,AI(LP7),ASD(LP7),PP7(KP),EP7
              T1 = ABS(((PA7-PP7(KP))/PA7)*100.)
    WRITE (KO,555) T1
    N = N+1
    IF (N.LT.4) GO TO 206
    N = 0
    WRITE (KO,559)
206 WRITE (KO,556) LP6,PA6,AI(LP6),ASD(LP6),PP6(KP),EP6
```

```
                T1 = ABS(((PA6-PP6(KP))/PA6)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 205
      N = 0
      WRITE (KO,559)
  205 WRITE (KO,556) LP5,PA5,AI(LP5),ASD(LP5),PP5(KP),EP5
                T1 = ABS(((PA5-PP5(KP))/PA5)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 204
      N = 0
      WRITE (KO,559)
  204 WRITE (KO,556) LP4,PA4,AI(LP4),ASD(LP4),PP4(KP),EP4
                T1 = ABS(((PA4-PP4(KP))/PA4)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 203
      N = 0
      WRITE (KO,559)
  203 WRITE (KO,556) LP3,PA3,AI(LP3),ASD(LP3),PP3(KP),EP3
                T1 = ABS(((PA3-PP3(KP))/PA3)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 202
      N = 0
      WRITE (KO,559)
  202 WRITE (KO,556) LP2,PA2,AI(LP2),ASD(LP2),PP2(KP),EP2
                T1 = ABS(((PA2-PP2(KP))/PA2)*100.)
      WRITE (KO,555) T1
      N = N+1
      IF (N.LT.4) GO TO 201
      N = 0
      WRITE (KO,559)
  201 WRITE (KO,556) LP1,PA1,AI(LP1),ASD(LP1),PP1(KP),EP1
                T1 = ABS(((PA1-PP1(KP))/PA1)*100.)
      WRITE (KO,555) T1
  547 FORMAT (///10X,'MANEUVER:  STEP RUDDER DEFLECTION AT T=0',/21X,'MA
     1XIMUM DEFLECTION OF ',F4.1,' DEGREES')
  548 FORMAT (///10X,'MANEUVER:  ZIG-ZAG,WITH STEP RUDDER',/21X,'DEFLECT
     1IONS OF ',F4.1,' DEGREES AT',/21X,'TIME T=100 AND T=200 SECONDS')
  549 FORMAT (///10X,'MANEUVER:  ZIG-ZAG, WITH SINUSOIDAL RUDDER',/20X,'
     1 DEFLECTIONS OF PERIOD',F6.1,' SECONDS AND',/21X,'MAXIMUM DEFLECTI
     2ONS OF',F5.1,' DEGREES')
  550 FORMAT(///10X,'SYSTEM:  MARINER-CLASS SURFACE VESSEL')
  552 FORMAT (///10X,'TRIAL PERIOD:  ',I4,' SECONDS')
  553 FORMAT (///10X,'NOISE LEVEL:  MEASUREMENT NOISE - ',I3,'%',//24X,'
     1PROCESS NOISE - ',I3,'%')
  555 FORMAT (//8X,'IDENTIFICATION WITHIN ',F5.2,'% OF THE TRUE VALUE.
     1')
  556 FORMAT(/////8X,'NP = ',I3,5X,'TRUE VALUE  =',2X,E13.5//8X,'SV  =',
     12X,E13.5,'  + OR -  ',E13.5//8X,'FV  =',2X,E13.5,'  + OR -  ',E13.
     25)
  557 FORMAT (1H1,////2X,'**********************************************
     1**********',/2X,'*
     2      *',/2X,'*  PARAMETRIC IDENTIFICATION - EXTENDED KALMAN FILTE
     3R  *',/2X,'*
     4*',/2X,'********************************************************')
  558 FORMAT (////20X,'(NON-LINEAR MODEL)')
  559 FORMAT ('1')
  566 FORMAT (///10X,'TIME STEP:  ',F3.1,' SECONDS')
  562 FORMAT (///10X,'NUMBER OF PRIMARY STATE VARIABLES:  ',I2)
  561 FORMAT (///10X,'EXAGGERATED NOISE FACTOR:  ',F5.1)
  563 FORMAT (///10X,'NUMBER OF COEFFICIENTS IDENTIFIED:  ',I3)
  560 CONTINUE
      GO TO 1
  909 CONTINUE
      CALL ENDPLT(7.0,0.0,999)
      END
      SUBROUTINE SETUP(Q,R,LP,NO,SP,EHT,IST,ICV,A,AI,ASD,PMS,XHT,XBAR,
     1HZ,ZERO)
C
C     SUBROUTINE SETUP ASSIGNS THE INITIAL VALUES TO MOST OF THE
C     MATRICES USED IN THE IDENTIFICATION
C
      INTEGER SP
      DIMENSION ZERO(1)
      DOUBLE PRECISION IST(1),ICV(1),HZ(NO,SP)
      DOUBLE PRECISION Q(LP,LP),R(LP,LP),EHT(SP,SP),XHT(1),XBAR(1)
      DOUBLE PRECISION A(1),AI(1),ASD(1),PMS(1)
      DOUBLE PRECISION PW,QW
      COMMON /PRM1/ PA1,PA2,PA3,PA4,PA5,PA6,PA7,PA8,PA9,PA10,PA11,PA12
      COMMON /PRM2/ PA13,PA14,PA15,PA16,PA17,PA18,PA20,PA21,PA22,PA23
      COMMON /PRM3/ PA24,PA25,PA26,PA27,PA28,PA29,PA30,PA31,PA32,PA33
      COMMON /PRAM3/ LP17,LP18,LP19,LP20,LP21,LP22,LP23,LP24,LP25,LP26
      COMMON /PRAM4/ LP27,LP28,LP29,LP30,LP31,LP32,LP33,LP34,LP35,LP36
      COMMON /PRAM2/ LP9,LP10,LP11,LP12,LP13,LP14,LP15,LP16
      COMMON /PRAM1/ LP1,LP2,LP3,LP4,LP5,LP6,LP7,LP8
      COMMON /EXAG/ PW,QW
      COMMON /PKI/ G
      COMMON /PRM4/ PA34,PA35,PA36
C
C *********
C
C     STORE THE TRUE VALUES OF THE COEFFICIENTS TO BE IDENTIFIED
C
      PA1 = A(LP1)
      PA2 = A(LP2)
      PA3 = A(LP3)
      PA4 = A(LP4)
      PA5 = A(LP5)
      PA6 = A(LP6)
      PA7 = A(LP7)
      PA8 = A(LP8)
      PA9 = A(LP9)
      PA10 = A(LP10)
      PA11 = A(LP11)
      PA12 = A(LP12)
      PA13 = A(LP13)
      PA14 = A(LP14)
      PA15 = A(LP15)
      PA16 = A(LP16)
      PA17 = A(LP17)
      PA18 = A(LP18)
      PA19 = A(LP19)
      PA20 = A(LP20)
      PA21 = A(LP21)
      PA22 = A(LP22)
      PA23 = A(LP23)
      PA24 = A(LP24)
      PA25 = A(LP25)
      PA26 = A(LP26)
      PA27 = A(LP27)
      PA28 = A(LP28)
      PA29 = A(LP29)
      PA30 = A(LP30)
      PA31 = A(LP31)
      PA32 = A(LP32)
      PA33 = A(LP33)
      PA34 = A(LP34)
      PA35 = A(LP35)
      PA36 = A(LP36)
C
C     ASSIGN INITIAL STATE AND COEFFICIENT ESTIMATION VALUES TO THE
C     EXTENDED STATE VECTOR
C
      DO 20 I = 1,NO
      XHT(I) = IST(I)
   20 CONTINUE
      XHT(NO+1) = AI(LP1)
      XHT(NO+2) = AI(LP2)
      XHT(NO+3) = AI(LP3)
      XHT(NO+4) = AI(LP4)
      XHT(NO+5) = AI(LP5)
      XHT(NO+6) = AI(LP6)
      XHT(NO+7) = AI(LP7)
      XHT(NO+8) = AI(LP8)
      XHT(NO+9) = AI(LP9)
      XHT(NO+10) = AI(LP10)
      XHT(NO+11) = AI(LP11)
      XHT(NO+12) = AI(LP12)
      XHT(NO+13) = AI(LP13)
      XHT(NO+14) = AI(LP14)
      XHT(NO+15) = AI(LP15)
      XHT(NO+16) = AI(LP16)
      XHT(NO+17) = AI(LP17)
      XHT(NO+18) = AI(LP18)
      XHT(NO+19) = AI(LP19)
      XHT(NO+20) = AI(LP20)
      XHT(NO+21) = AI(LP21)
      XHT(NO+22) = AI(LP22)
      XHT(NO+23) = AI(LP23)
      XHT(NO+24) = AI(LP24)
      XHT(NO+25) = AI(LP25)
      XHT(NO+26) = AI(LP26)
      XHT(NO+27) = AI(LP27)
      XHT(NO+28) = AI(LP28)
      XHT(NO+29) = AI(LP29)
      XHT(NO+30) = AI(LP30)
      XHT(NO+31) = AI(LP31)
      XHT(NO+32) = AI(LP32)
      XHT(NO+33) = AI(LP33)
      XHT(NO+34) = AI(LP34)
      XHT(NO+35) = AI(LP35)
      XHT(NO+36) = AI(LP36)
C
C     INITIALIZE THE MEASUREMENT FUNCTION
```

```
C
      DO 4 N1 = 1,NO
      DO 5 M1 = 1,SP
      HZ(N1,M1) = 0.D0
    5 CONTINUE
    4 CONTINUE
      DO 3 N1 = 1,NO
      HZ(N1,N1) = 1.D0
    3 CONTINUE
      DO 91 N = 1,SP
      XBAR(N) = 0.D0
   91 CONTINUE
      DO 300 II = 1,47
      ZERO(II) = 0.0
  300 CONTINUE
C
C     INITIALIZE THE ERROR COVARIANCE MATRIX
C
      DO 7 N1 = 1,SP
      DO 6 Mi = 1,SP
      EHT(N1,M1) = 0.D0
    6 CONTINUE
    7 CONTINUE
      DO 21 I = 1,NO
      EHT(I,I) = ICV(I)
   21 CONTINUE
      EHT(NO+1,NO+1) = ASD(LP1)**2
      EHT(NO+2,NO+2) = ASD(LP2)**2
      EHT(NO+3,NO+3) = ASD(LP3)**2
      EHT(NO+4,NO+4) = ASD(LP4)**2
      EHT(NO+5,NO+5) = ASD(LP5)**2
      EHT(NO+6,NO+6) = ASD(LP6)**2
      EHT(NO+7,NO+7) = ASD(LP7)**2
      EHT(NO+8,NO+8) = ASD(LP8)**2
      EHT(NO+9,NO+9) = ASD(LP9)**2
      EHT(NO+10,NO+10) = ASD(LP10)**2
      EHT(NO+11,NO+11) = ASD(LP11)**2
      EHT(NO+12,NO+12) = ASD(LP12)**2
      EHT(NO+13,NO+13) = ASD(LP13)**2
      EHT(NO+14,NO+14) = ASD(LP14)**2
      EHT(NO+15,NO+15) = ASD(LP15)**2
      EHT(NO+16,NO+16) = ASD(LP16)**2
      EHT(NO+17,NO+17) = ASD(LP17)**2
      EHT(NO+18,NO+18) = ASD(LP18)**2
      EHT(NO+19,NO+19) = ASD(LP19)**2
      EHT(NO+20,NO+20) = ASD(LP20)**2
      EHT(NO+21,NO+21) = ASD(LP21)**2
      EHT(NO+22,NO+22) = ASD(LP22)**2
      EHT(NO+23,NO+23) = ASD(LP23)**2
      EHT(NO+24,NO+24) = ASD(LP24)**2
      EHT(NO+25,NO+25) = ASD(LP25)**2
      EHT(NO+26,NO+26) = ASD(LP26)**2
      EHT(NO+27,NO+27) = ASD(LP27)**2
      EHT(NO+28,NO+28) = ASD(LP28)**2
      EHT(NO+29,NO+29) = ASD(LP29)**2
      EHT(NO+30,NO+30) = ASD(LP30)**2
      EHT(NO+31,NO+31) = ASD(LP31)**2
      EHT(NO+32,NO+32) = ASD(LP32)**2
      EHT(NO+33,NO+33) = ASD(LP33)**2
      EHT(NO+34,NO+34) = ASD(LP34)**2
      EHT(NO+35,NO+35) = ASD(LP35)**2
      EHT(NO+36,NO+36) = ASD(LP36)**2
C
C     SET EXAGERATED NOISE PARAMETERS
C
      DO 1 N1 = 1,LP
      DO 2 M1 = 1,LP
      Q(N1,M1) = 0.D0
      R(N1,M1) = 0.D0
    2 CONTINUE
    1 CONTINUE
      DO 55 IR = 1,NO
      IA = IR+2*NO
      IV = IA+NO
      Q(IR,IR) = QW*PMS(IA)**2
      R(IR,IR) = PW*PMS(IV)**2
   55 CONTINUE
      RETURN
      END
      SUBROUTINE RKL(H,TI,XVI,XRI,XPSI,XVDI,N,A,ZV,ZR,ZPS,ZVD,US,DUS,TS,
     1IN,P,WN,VN,NO,V)
C
C     SUBROUTINE RKL GENERATES THE NOISY SEA TRIAL DATA USED FOR THE
C     IDENTIFICATION BY THE KALMAN FILTER...THE EQUATIONS OF MOTION ARE
C     USED TO GENERATE MOTION TRAJECTORIES WITH A SPECIFIED LEVEL OF
C     NOISE...USING THE RUNGE-KUTTA 4TH ORDER INTEGRATION TECHNIQUE,
C     THE VALUES OF THE STATE VARIABLES AS A FUNCTION OF TIME ARE
C     RETURNED TO THE CALLING PROGRAM
C
      INTEGER SP
      DIMENSION IN(1)
      DOUBLE PRECISION U,UD,DUD,US(1),DUS(1)
      DOUBLE PRECISION H,HM,T,TI,TM,TN,TS(1)
      DOUBLE PRECISION WL,VL,WN(1),VN(1)
      DOUBLE PRECISION V(376)
      DOUBLE PRECISION P(1),A(1)
      DOUBLE PRECISION ZV(1),ZR(1),ZPS(1),ZVD(1)
      DOUBLE PRECISION FNLV,FNLR
      DOUBLE PRECISION DV,DR
      DOUBLE PRECISION ZZ1,ZZ2
      DOUBLE PRECISION XVI,XRI,XPSI,XVDI
      DOUBLE PRECISION XV,XR,XPS,XVD
      DOUBLE PRECISION YV1,YV2,YV3,YV4
      DOUBLE PRECISION YR1,YR2,YR3,YR4
      DOUBLE PRECISION YPS1,YPS2,YPS3,YPS4
      DOUBLE PRECISION YVD1,YVD2,YVD3,YVD4
      DOUBLE PRECISION XVN,XRN,XPSN,XVDN
      DOUBLE PRECISION YV1N,YV2N,YV3N,YV4N
      DOUBLE PRECISION YR1N,YR2N,YR3N,YR4N
      DOUBLE PRECISION YPS1N,YPS2N,YPS3N,YPS4N
      DOUBLE PRECISION YVD1N,YVD2N,YVD3N,YVD4N
      DOUBLE PRECISION DSIN,DCOS
      COMM]N /PKI/ G
C
C *********
```

```
C
C     INITIALIZE THE STATE VARIABLES AND THE TIME INCREMENT USED IN THE
C     INTEGRATION
C
      T = TI
      XV = XVI
      XR = XRI
      XPS = XPSI
      XVD = XVDI
      XVN = XVI
      XRN = XRI
      XPSN = XPSI
      XVDN = XVDI
      HM = H/2.
      NVAR = NO
C
C     START THE INTEGRATION LOOP, ONE CIRCUIT PER TIME STEP
C
      DO 300 IJ = 1,N
      TM = T+HM
      TN = T+H
      UD = U(T)
      US(IJ) = UD
      DUS(IJ) = DUD(T)
C
C     GENERATE THE GAUSSIAN PROCESS NOISE FOR EACH STATE VARIABLE IN
C     THE INITIAL PHASE OF THE TIME STEP
C
      DO 333 IVAR = 1,NVAR
      CALL WNO(IN,P,IVAR,NVAR,WL)
      WN(IVAR) = WL
  333 CONTINUE
C
C     CALCULATE THE NOISELESS STATE VARIABLES AT THE START OF THE TIME
C     INCREMENT, AT TIME T
C
```

```
      DV = FNLV(XV,XR,UD,A)
      DR = FNLR(XV,XR,UD,A)
      YV1 = H*DV
      YR1 = H*DR
      YPS1 = H*XR
      YVD1 = DV
C
C     CALCULATE THE NOISY STATE VARIABLES AT TIME T FOR GENERATING THE
C     ACTUAL NOISY DATA FOR THIS TIME INCREMENT
C
      YV1N = H*(DV+G*WN(1))
      YR1N = H*(DR+G*WN(2))
      YPS1N = H*(XR+G*WN(3))
      YVD1N = DV+G*WN(4)
      ZZ1 = XV+0.5*YV1
      ZZ2 = XR+0.5*YR1
      UD = U(TM)
C
C     GENERATE NEW PROCESS NOISE FOR THE SECOND PHASE OF THE INTEGRATION
C
      DO 335 IVAR = 1,NVAR
      CALL WNO(IN,P,IVAR,NVAR,WL)
      WN(IVAR) = WL
  335 CONTINUE
C
C     DO THE NOISELESS STATE CALCULATIONS AT TIME TM
C
      DV = FNLV(ZZ1,ZZ2,UD,A)
      DR = FNLR(ZZ1,ZZ2,UD,A)
      YV2 = H*DV
      YR2 = H*DR
      YPS2 = H*ZZ2
      YVD2 = DV
C
C     DO THE NOISY STATE CALCULATIONS AT TIME TM
C
```

```
      YV2N = H*(DV+G*WN(1))
      YR2N = H*(DR+G*WN(2))
      YPS2N = H*(ZZ2+G*WN(3))
      YVD2N = DV+G*WN(4)
      ZZ1 = XV+0.5*YV2
      ZZ2 = XR+0.5*YR2
C
C     GENERATE PROCESS NOISE FOR THE THIRD PHASE OF THE INTEGRATION
C
      DO 336 IVAR = 1,NVAR
      CALL WNO(IN,P,IVAR,NVAR,WL)
      WN(IVAR) = WL
  336 CONTINUE
C
C     DO THE NOISELESS STATE CALCULATIONS AT TIME TM
C
      DV = FNLV(ZZ1,ZZ2,UD,A)
      DR = FNLR(ZZ1,ZZ2,UD,A)
      YV3 = H*DV
      YR3 = H*DR
      YPS3 = H*ZZ2
      YVD3 = DV
C
C     DO THE NOISY STATE CALCULATIONS AT TIME TM
C
      YV3N = H*(DV+G*WN(1))
      YR3N = H*(DR+G*WN(2))
      YPS3N = H*(ZZ2+G*WN(3))
      YVD3N = DV+G*WN(4)
      ZZ1 = XV+YV3
      ZZ2 = XR+YR3
      UD = U(TN)
C
C     GENERATE PROCESS NOISE VALUES FOR THE FINAL PHASE OF THE
C     INTEGRATION AT THE END OF THE TIME INCREMENT
C
```

```
      DO 337 IVAR = 1,NVAR
      CALL WNO(IN,P,IVAR,NVAR,WL)
      WN(IVAR) = WL
  337 CONTINUE
C
C     DO THE NOISELESS STATE CALCULATIONS FOR TIME TN
C
      DV = FNLV(ZZ1,ZZ2,UD,A)
      DR = FNLR(ZZ1,ZZ2,UD,A)
      YV4 = H*DV
      YR4 = H*DR
      YPS4 = H*ZZ2
      YVD4 = DV
C
C     DO THE NOISY STATE CALCULATIONS AT TIME TN
C
      YV4N = H*(DV+G*WN(1))
      YR4N = H*(DR+G*WN(2))
      YPS4N = H*(ZZ2+G*WN(3))
      YVD4N = DV+G*WN(4)
C
C     FIND THE VALUE OF THE STATE VARIABLES OVER THE INCREMENT AND ADD
C     TO THE CUMULATIVE TOTAL OVER ALL TIME
C
      XV = XV+1./6.*(YV1+2.*YV2+2.*YV3+YV4)
      XR = XR+1./6.*(YR1+2.*YR2+2.*YR3+YR4)
      XPS = XPS+1./6.*(YPS1+2.*YPS2+2.*YPS3+YPS4)
      XVD = 1./6.*(YVD1+2.*YVD2+2.*YVD3+YVD4)
      XVN = XVN+1./6.*(YV1N+2.*YV2N+2.*YV3N+YV4N)
      XRN = XRN+1./6.*(YR1N+2.*YR2N+2.*YR3N+YR4N)
      XPSN = XPSN+1./6.*(YPS1N+2.*YPS2N+2.*YPS3N+YPS4N)
      XVDN = 1./6.*(YVD1N+2.*YVD2N+2.*YVD3N+YVD4N)
C
C     GENERATE THE GAUSSIAN MEASUREMENT NOISE FOR EACH MEASURED STATE
C     VARIABLE TO BE OUTPUT
C
```

```
      DO 334 IR = 1,NVAR
      IVAR = IR+NVAR
      CALL WNO(IN,P,IVAR,NVAR,WL)
      VN(IR) = WL
  334 CONTINUE
C
C     DETERMINE THE MEASURED NOISY OUTPUT OF THE SYSTEM
C
      V(IJ) = VN(1)
      VL = VN(2)
      ZR(IJ) = XRN+VL
      VL = VN(3)
      ZPS(IJ) = XPSN+VL
      VL = VN(4)
      ZVD(IJ) = XVDN+VL
      T = T+H
      TS(IJ) = T
  300 CONTINUE
      US(N+1) = UD
      DUS(N+1) = DUD(TN)
C
C     INTEGRATE THE MEASURED NOISY ACCELERATION TO GENERATE A NOISY
C     SWAY VELOCITY OUTPUT
C
      ZV(1) = 0.5*H*(XVDI+ZVD(1))
      DO 301 IK = 2,N
      ZV(IK) = ZV(IK-1)+0.5*H*(ZVD(IK)+ZVD(IK-1))
  301 CONTINUE
      DO 302 KI = 1,N
      ZV(KI) = ZV(KI)+V(KI)
  302 CONTINUE
      RETURN
      END
      DOUBLE PRECISION FUNCTION U(T)
C
C     FUNCTION U(T) GENERATES RUDDER DEFLECTIONS FOR SPECIFIC MANEUVERS
C     AS A FUNCTION OF TIME
C
      DOUBLE PRECISION D,DI,T,TL,DSIN,PER
      COMMON /INPUT/ DI,TL,JJ
C
C *********
C
      D = DI/57.296
      GO TO (10,20,30),JJ
C
C     STEP RUDDER DEFLECTION
C
   10 U = D
      RETURN
C
C     ZIG-ZAG RUDDER DEFLECTION
C
   20 IF (T-100.) 3,4,4
    3 U = D
      RETURN
    4 IF (T-200.) 5,6,6
    5 U = -D
      RETURN
    6 U = 0.D0
      RETURN
C
C     SINUSOIDAL RUDDER DEFLECTION
C
   30 PER = T/TL*3.14159
      U = D*DSIN(PER)
      RETURN
      END
      DOUBLE PRECISION FUNCTION DUD(T)
C
C     FUNCTION DUD(T) CALCULATES THE TIME RATE OF CHANGE IN THE RUDDER
C     DEFLECTION FOR SPECIFIC MANEUVERS
C
      DOUBLE PRECISION D,DI,T,TL,DCOS,PER
      COMMON /INPUT/ DI,TL,JJ
C
C *********
C
      D = DI/57.296
      GO TO (10,20,30),JJ
C
C     STEP RUDDER DEFLECTION
C
   10 DUD = 0.D0
      RETURN
C
C     ZIG-ZAG RUDDER DEFLECTICN
C
   20 DUD = 0.D0
      RETURN
C
C     SINUSOIDAL RUDDER DEFLECTION
C
   30 PER = T/TL*3.14159
      DUD = D*DCOS(PER)
      RETURN
      END
      SUBROUTINE WNO(IN,P,IVAR,NVAR,W)
C
C     SUBROUTINE WNO GENERATES GAUSSIAN WHITE NOISE FROM THE SPECIFIED
C     STATISTICAL PROPERTIES CF THE DESIRED NOISE LEVELS
C
      DIMENSION IN(1)
      DOUBLE PRECISION P(1),AM,S,W
C
C *********
C
      IX = IN(IVAR)
      LW = IVAR+2*NVAR
C
C     DESIRED MEAN
C
      AM = P(IVAR)
C
C     DESIRED STANDARD DEVIATION
C
      S = P(LW)
      CALL GAUSS(IX,S,AM,W)
      IN(IVAR) = IX
      RETURN
      END
      SUBROUTINE GAUSS(IX,S,AM,W)
C
C     SUBROUTINE GAUSS COMPUTES A NORMALLY DISTRIBUTED RANDOM NUMBER
C     WITH A GIVEN MEAN AND STANDARD DEVIATION
C
      DOUBLE PRECISION S,AM,W,A
C
C *********
C
      A = 0.D0
      DO 50 I = 1,12
      CALL RANDU(IX,IY,Y)
      IX = IY
   50 A = A+Y
      W = (A-6.D0)*S+AM
      RETURN
      END
      SUBROUTINE RANDU(IX,IY,YFL)
C
C     SUBROUTINE RANDU GENERATES UNIFORMLY RANDOM NUMBERS FOR USE IN
C     SUBROUTINE GAUSS
C
      IY = IX*65539
      IF (IY) 5,6,6
    5 IY = IY+2147483647+1
    6 YFL = IY
      YFL = YFL*0.4656613E-9
      RETURN
      END
      DOUBLE PRECISION FUNCTION FNLV(XV,XR,U,A)
C
C     FUNCTION FNLV CALCULATES THE VALUE OF THE TIME DERIVATIVE OF THE
C     SWAY VELOCITY
C
      DOUBLE PRECISION XV,XR,U,A(1),F2,F3,F4
C
C *********
C
      F2 = A(9)+A(6)*XV+A(7)*XR+A(8)*U+A(26)*U**3 +A(27)*XR*XV**2 +A(28)
     1*U*XV**2
      F3 = A(15)+A(12)*XV+A(13)*XR+A(14)*U+A(31)*U**3 +A(32)*XR*XV**2 +A
     1(33)*U*XV**2
      F4 = 1./(A(4)*A(11)-A(10)*A(5))
      FNLV = F4*(A(11)*F2-A(5)*F3)
      RETURN
      END
      DOUBLE PRECISION FUNCTION FNLR(XV,XR,U,A)
C
C     FUNCTION FNLR CALCULATES THE VALUE OF THE TIME DERIVATIVE OF THE
C     YAW VELOCITY
C
      DOUBLE PRECISION XV,XR,U,A(1),F2,F3,F4
C
C *********
C
      F2 = A(9)+A(6)*XV+A(7)*XR+A(8)*U+A(26)*U**3 +A(27)*XR*XV**2 +A(28)
     1*U*XV**2
      F3 = A(15)+A(12)*XV+A(13)*XR+A(14)*U+A(31)*U**3 +A(32)*XR*XV**2 +A
     1(33)*U*XV**2
      F4 = 1./(A(4)*A(11)-A(10)*A(5))
      FNLR = F4*(A(4)*F3-A(10)*F2)
      RETURN
      END
      SUBROUTINE SHOMO(U1,V1,U2,V2,U3,V3,XPAGE,YPAGE,IPEN,K,M,YLABEL)
C
C     SUBROUTINE SHOMO CAN BE USED TO PORTRAY THE MOTIONS OF THE VESSEL
C     USING THE CALCOMP PLOTTING ROUTINE...IT CAN BE USED IN CONJUNCTION
C     WITH OR SEPERATE FROM THE ALTERNATE ROUTINE PLOTM.
C
      DIMENSION U1(1),V1(1),U2(1),V2(1),U3(1),V3(1)
      DIMENSION YLABEL(1)
      IF (K.EQ.1) GO TO 306
      IF (M.GT.1) GO TO 307
C
C     PRINT A KEY FOR PLOT IDENTIFICATION.
C
      CALL SYMBOL(2.0,0.75,0.14,'MEASUREMENT NOISE - 5% ',0.0,23)
      CALL SYMBOL(2.0,0.25,0.14,'PROCESS NOISE - 5% ',0.0,19)
      X = 13.24
      CALL SYMBOL(11.0,1.0,0.14,'FILTERED STATE  ',0.0,16)
      CALL SYMBOL(X,1.0,0.14,15,0.0,-1)
      X = X+0.15
      DO 301 I = 1,5
      CALL SYMBOL(X,1.0,0.14,15,0.0,-2)
      X = X+0.15
  301 CONTINUE
      X = 13.24
      CALL SYMBOL(11.0,0.5,0.14,'NOISY STATE     ',0.0,16)
      DO 302 I = 1,6
      CALL SYMBOL(X,0.5,0.14,1,0.0,-1)
      X = X+0.15
  302 CONTINUE
      X = 13.24
      CALL SYMBOL(11.0,0.0,0.14,'ZERO LINE       ',0.0,16)
      DO 303 I = 1,6
      CALL SYMBOL(X,0.0,0.14,15,0.0,-1)
      X = X+0.15
  303 CONTINUE
      CALL PLOT(0.0,2.3,-3)
```

```
 307 CONTINUE
C
C     PLOT BOTH THE FILTERED AND NOISY MOTIONS AS A FUNCTION OF TIME.
C
      CALL MINMAX(V1,47,AMIN,AMAX)
      IF (AMIN.LE.0.0.AND.AMAX.GE.0.0) GO TO 304
      CALL PICTUR(8.0,4.0,'TIME (SEC.)',11,YLABEL,33,U1,V1,-47,0.10,
     11,U3,V3,47,0.0,1)
      GO TO 305
 304 CALL PICTUR(8.0,4.0,'TIME (SEC.)',11,YLABEL,33,U1,V1,-47,0.10,
     11,U2,V2,-47,0.10,15,U3,V3,47,0.0,1)
 305 CALL PLOT(XPAGE,YPAGE,IPEN)
 306 CONTINUE
      RETURN
      END
      SUBROUTINE SHOCO(U1,V1,U2,V2I,XPAGE,YPAGE,IPEN,K,M,N)
C
C     SUBROUTINE SHOCO CAN BE USED TO PORTRAY THE COEFFICIENT
C     IDENTIFICATION AS A FUNCTION OF TIME USING THE CALCOMP PLOTTER.
C
      DIMENSION U1(1),V1(1),U2(1),V2(47)
      IF (K.EQ.1) GO TO 307
      DO 300 I = 1,47
      V2(I) = V2I
  300 CONTINUE
      IF (M.GT.1) GO TO 333
      CALL PLOT(0.0,-2.3,-3)
      CALL SYMBOL(2.0,0.75,0.14,'MEASUREMENT NOISE - 5% ',0.0,23)
      CALL SYMBOL(2.0,0.25,0.14,'PROCESS NOISE - 5% ',0.0,19)
      X = 13.24
      CALL SYMBOL(11.0,.75,0.14,'IDENTIFICATION  ',0.0,16)
      CALL SYMBOL(X,.75,0.14,15,0.0,-1)
      X = X+0.15
      DO 301 I = 1,5
      CALL SYMBOL(X,.75,0.14,15,0.0,-2)
      X = X+0.15
  301 CONTINUE
      X = 13.24
      CALL SYMBOL(11.0,.25,0.14,'TRUE VALUE      ',0.0,16)
      DO 302 I = 1,6
      CALL SYMBOL(X,.25,0.14,15,0.0,-1)
      X = X+0.15
  302 CONTINUE
      CALL PLOT(0.0,2.3,-3)
  333 CONTINUE
      CALL PICTUR(8.0,4.0,'TIME (SEC.)',11,'COEFFICIENT VALUE ',18,U1,V1
     1,47,0.0,1,U2,V2,-47,0.10,15)
      CALL PLOT(-11.0,0.3,-3)
      GO TO (304,306,303,305),N
  303 CALL SYMBOL(7.5,3.5,0.48,85,0.0,-1)
```

```
      CALL SYMBOL(8.05,3.4,0.24,101,0.0,-1)
      GO TO 307
  304 CALL SYMBOL(7.5,3.5,0.48,104,0.0,-1)
      CALL SYMBOL(8.05,3.4,0.24,101,0.0,-1)
      GO TO 307
  305 CALL PLOT(-0.70,0.0,-3)
      CALL SYMBOL(6.5,3.5,0.48,85,0.0,-1)
      CALL SYMBOL(7.05,3.4,0.24,89,0.0,-1)
      CALL SYMBOL(7.36,3.5,0.28,15,0.0,-1)
      CALL SYMBOL(7.74,3.5,0.28,84,0.0,-1)
      CALL SYMBOL(8.12,3.5,0.28,103,0.0,-1)
      CALL SYMBOL(8.38,3.4,0.24,71,0.0,-1)
      CALL SYMBOL(8.64,3.5,0.28,100,0.0,-1)
      CALL PLOT(0.70,0.0,-3)
      GO TO 307
  306 CALL SYMBOL(6.5,3.5,0.48,104,0.0,-1)
      CALL SYMBOL(7.05,3.4,0.24,89,0.0,-1)
      CALL SYMBOL(7.36,3.5,0.28,15,0.0,-1)
      CALL SYMBOL(7.74,3.5,0.28,84,0.0,-1)
      CALL SYMBOL(8.12,3.5,0.28,100,0.0,-1)
      GO TO 307
  308 CALL SYMBOL(6.5,3.5,0.28,84,0.0,-1)
      CALL SYMBOL(6.85,3.5,0.28,15,0.0,-1)
      CALL SYMBOL(7.20,3.5,0.48,104,0.0,-1)
      CALL SYMBOL(7.75,3.4,0.24,101,0.0,-1)
      GO TO 307
  309 CALL PLOT(-0.3,0.0,-3)
      CALL SYMBOL(6.5,3.5,0.28,84,0.0,-1)
      CALL SYMBOL(6.85,3.5,0.28,103,0.0,-1)
      CALL SYMBOL(7.11,3.4,0.24,71,0.0,-1)
      CALL SYMBOL(7.40,3.5,0.28,15,0.0,-1)
      CALL SYMBOL(7.75,3.5,0.48,104,0.0,-1)
      CALL SYMBOL(8.30,3.4,0.24,89,0.0,-1)
      CALL PLOT(0.30,0.0,-3)
      GO TO 307
  310 CALL PLOT(-0.3,0.0,-3)
```

```
      CALL SYMBOL(6.5,3.5,0.48,73,0.0,-1)
      CALL SYMBOL(7.05,3.4,0.24,105,0.0,-1)
      CALL SYMBOL(7.36,3.5,0.28,15,0.0,-1)
      CALL SYMBOL(7.74,3.5,0.48,85,0.0,-1)
      CALL SYMBOL(8.29,3.4,0.24,89,0.0,-1)
      CALL PLOT(0.3,0.0,-3)
      GO TO 307
  311 CALL SYMBOL(7.5,3.5,0.48,85,0.0,-1)
      CALL SYMBOL(8.05,3.4,0.24,43,0.0,-1)
  307 CONTINUE
      CALL PLOT(0.0,-0.3,-3)
      CALL PLOT(XPAGE,YPAGE,IPEN)
      RETURN
      END
      SUBROUTINE PLOTM(NO,A,N,M,NS,NE)
C
C     *******************************************************
C
C        SUBROUTINE PLOT
C
C        PURPOSE
C           PLOT SEVERAL CROSS VARIABLES Y VERSUS A BASE
C           VARIABLE X IN A FORMAT SUITABLE FOR THESIS USE
C
C        USAGE
C           CALL PLOT(NO,A,N,M,NS)
C
C        DESCRIPTION OF PARAMETERS
C           NO - PLOT NUMBER OF .LTE. 3 DIGITS
C           A  - MATRIX OF DATA TO BE PLOTTED. MUST BE IN
C                STANDARD SINGLE COLUMN FORM. FIRST COLUMN
C                REPRESENTS BASE VARIABLE AND SUCCESSIVE
C                COLUMNS ARE THE CROSS VARIABLES (MAXIMUM IS
C                NINE).
C           N  - NUMBER OF ROWS IN MATRIX A. N MUST BE
C                .LTE. 47
C           M  - NUMBER OF COLUMNS IN MATRIX A. M MUST BE
C                .LTE. 10
C           NS - CODE FOR SORTING THE BASE VARIABLE DATA IN
C                ASCENDING ORDER
C                  0  SORTING IS NOT NECESSARY(ALREADY IN
C                     ASCENDING ORDER)
C                  1  SORTING IS NECESSARY
C     *******************************************************
C
      DIMENSION OUT(51),IANG(9),YPR(6),YPT(3),A(1)
      INTEGER*2 OUT,IANG,BLANK
      IF (NE.EQ.2) GO TO 99
      DATA IANG /'1 ','2 ','3 ','4 ','5 ','6 ','7 ','8 ','9 '/
      DATA BLANK /'  '/
```

```
C     FORMAT STATEMENTS FOR THESIS USE
    1 FORMAT(1H1,27X,7H  PLOT ,I8)
    2 FORMAT (1H ,E10.3,'*',51A1,'*')
    3 FORMAT(1H ,10X,':',51X,':')
    4 FORMAT (1H ,18X,' *....* INCREMENT IS  ',E15.7)
    5 FORMAT (1H ,8X,E15.7,5X,E15.7,5X,E15.7)
    7 FORMAT(1H ,11X,36H*....*....*....*....*....*....*....*,
     115H....*....*....*)
    8 FORMAT (1H ,3X,E9.2,1X,E9.2,1X,E9.2,1X,E9.2,1X,E9.2,1X,E9.2)
    9 FORMAT(1H )
      NL=47
      NTH=51
      NLL=NL
      IF(NS)16,16,10
C
C     SORTING ROUTINE
C
   10 DO 15 I=1,N
      DO 14 J=I,N
      IF(A(I)-A(J))14,14,11
   11 L=I-N
      LL=J-N
      DO 12 K=1,M
      L=L+N
      LL=LL+N
      F=A(L)
      A(L)=A(LL)
   12 A(LL)=F
   14 CONTINUE
   15 CONTINUE
   16 CONTINUE
C
C     FIND BASE AND CROSS VARIABLE SCALES
C
      XSCAL=(A(N)-A(1))/(FLOAT(NLL-1))
      M1=N+1
      YMAX = -1.E37
      YMIN = 1.E37
      M2=M*N
      DO 40 J=M1,M2
      IF (A(J) .GT. YMAX) YMAX=A(J)
      IF (A(J) .LT. YMIN) YMIN=A(J)
   40 CONTINUE
      YSCAL=(YMAX-YMIN)/50.0
      IF (YSCAL.EQ.0.) YSCAL=1.E-37
      YPR(1) = YMIN
      DO 90 KN = 1,4
      YPR(KN+1)=YPR(KN)+YSCAL*10.0
   90 CONTINUE
      YPR(6)=YMAX
      YPT(1)=YMIN
      YSTAR=YSCAL*5.0
      YPT(2)=YMIN+YSCAL*25.0
      YPT(3)=YMAX
C
C     PRINT HEADING AND CROSS VARIABLE SCALE
C
      WRITE(6,1)NO
      WRITE(6,4)YSTAR
      WRITE(6,5)(YPT(IP),IP = 1,3)
        WRITE(6,8)(YPR(IP),IP=1,6)
      WRITE(6,7)
C
C     FIND BASE VARIABLE PRINT POSITION
C
      XB=A(1)
      L=1
      MY=M-1
      I=1
      XEPS=XSCAL/FLOAT(2*(NLL-1))
   45 F = FLOAT(I-1)
      XPR = XB+F*XSCAL
      XDIF=A(L)-XPR-XEPS
      IF(XDIF)50,50,70
C
C     FIND CROSS VARIABLES
C
   50 DO 55 IX=1,NTH
      OUT(IX)=BLANK
   55 CONTINUE
      DO 60 J=1,MY
      LL=L+J*N
      JP = ((A(LL)-YMIN)/YSCAL)+1.0
      OUT(JP)=IANG(J)
   60 CONTINUE
C
C     PRINT LINE AND CLEAR, OR SKIP
C
      WRITE(6,2)XPR,(OUT(IZ),IZ=1,NTH)
      L=L+1
      GO TO 80
   70 WRITE(6,3)
   80 I=I+1
      IF(I-NLL)45,84,86
   84 XPR=A(N)
      GO TO 50
C
C     PRINT BOTTOM AND CROSS VARIABLE SCALE
C
   86 WRITE(6,7)
      WRITE(6,8)(YPR(IP),IP=1,6)
      WRITE(6,9)
   99 CONTINUE
      RETURN
      END
      SUBROUTINE FILTER(EJ,E1,E2,E3,E4,E5,EN,EM,EL,ES,Q1,BN,FT,T,H1,H2,
     1H3,LP,KS,EBAR,B,K,Q,R,EE,NO,Z,SP,ZV,ZR,ZPS,ZVD,US,TS,XHT,XBAR,HZ,
     2A,C,EHT,IC,IR,W,H,TI,N,DUS)
C
C     SUBROUTINE FILTER IS THE MAIN FILTERING ROUTINE FOR THE PROGRAM
C
      INTEGER SP
      DIMENSION VP(94),RP(94),PSP(94),VDP(94)
      DIMENSION PP1(94),PP2(94),PP3(94),PP4(94),PP5(94),PP6(94),PP7(94)
      DIMENSION PP8(94),PP9(94),PP10(94),PP11(94),PP12(94),PP13(94)
      DIMENSION PP14(94),PP15(94),PP16(94)
      DIMENSION PP17(94),PP18(94),PP19(94),PP20(94),PP21(94),PP22(94)
      DIMENSION PP23(94),PP24(94),PP25(94),PP26(94),PP27(94),PP28(94)
      DIMENSION PP29(94),PP30(94),PP31(94),PP32(94),PP33(94),PP34(94)
      DIMENSION PP35(94),PP36(94)
      DIMENSION IC(1),IR(1)
      DIMENSION EE(1)
      DOUBLE PRECISION DUS(1),USDV,DUD,UZD
      DOUBLE PRECISION W(1)
      DOUBLE PRECISION Z(1),ZV(1),ZR(1),ZPS(1),ZVD(1)
      DOUBLE PRECISION K(LP,NO),Q(LP,LP),R(LP,LP),B(LP,LP),EBAR(LP,LP)
      DOUBLE PRECISION EHT(SP,SP),XHT(1),HZ(NO,SP),A(1),XBAR(1)
      DOUBLE PRECISION DI,TI,USV,UZ,U,DT,H
      DOUBLE PRECISION FNLV,FNLR
      DOUBLE PRECISION E1(LP,LP),E2(LP,LP),E3(LP,LP),E4(LP,LP),E5(LP,LP)
      DOUBLE PRECISION EJ(LP,LP),EN(LP,LP),Q1(LP,LP),BN(LP,LP',FT(LP,LP)
      DOUBLE PRECISION EM(LP,LP),H1(NO,LP),T(LP,NO),EL(1),ES(1)
      DOUBLE PRECISION H2(NO,LP),H3(LP,LP)
      DOUBLE PRECISION TS(1),US(1)
      DOUBLE PRECISION C(1)
      COMMON /OUTP1/ VP,RP,PSP,VDP,PP1,PP2,PP3,PP4,PP5,PP6,PP7,PP8
      COMMON /OUTP2/ PP9,PP10,PP11,PP12,PP13,PP14,PP15,PP16
      COMMON /OUTP5/ PP17,PP18,PP19,PP20,PP21,PP22,PP23,PP24,PP25,PP26
      COMMON /OUTP6/ PP27,PP28,PP29,PP30,PP31,PP32,PP33,PP34,PP35,PP36
      COMMON /OUTP3/ EV,ER,EPS,EVD,EP1,EP2,EP3,EP4,EP5,EP6,EP7,EP8
      COMMON /OUTP4/ EP9,EP10,EP11,EP12,EP13,EP14,EP15,EP16
      COMMON /OUTP7/ EP17,EP18,EP19,EP20,EP21,EP22,EP23,EP24,EP25,EP26
      COMMON /OUTP8/ EP27,EP28,EP29,EP30,EP31,EP32,EP33,EP34,EP35,EP36
      COMMON /PRAM3/ LP17,LP18,LP19,LP20,LP21,LP22,LP23,LP24,LP25,LP26
      COMMON /PRAM4/ LP27,LP28,LP29,LP30,LP31,LP32,LP33,LP34,LP35,LP36
      COMMON /PRAM1/ LP1,LP2,LP3,LP4,LP5,LP6,LP7,LP8
      COMMON /PRAM2/ LP9,LP10,LP11,LP12,LP13,LP14,LP15,LP16
      COMMON /INPUT/ DI,TL,JJ
      COMMON /PKI/G
C
C *********
C
      KB = 2
      MH = 1
      KFIM = 47
C
C     PROCESS THE SEA TRIAL DATA
C
      USV = US(1)
      USDV = DUS(1)
      UZ = U(TI)
      UZD = DUD(TI)
      US(1) = UZ
      DUS(1) = UZD
      DT = H
C
C     PROPAGATE THE STATE AND ERROR COVARIANCE MATRICES FROM THE
C     INITIAL CONDITIONS
C
      CALL PROP(DT,US,A,Q,1,EJ,E1,E2,E3,E4,E5,B,EN,Q1,BN,FT,EBAR,LP,NO,
     1SP,EHT,XHT,XBAR,W,DUS)
C
C     CALCULATE THE INITIAL GAIN FOR THE KALMAN FILTER
C
      CALL GAIN(HZ,R,EBAR,K,EM,H1,T,LP,NO,SP,W,H2,C,IC,IR)
C
C     UPDATE THE STATE AND ERROR COVARIANCE MATRICES FROM THEIR
```

```
C     INITIAL VALUES
C
      CALL UPDT(Z,ZV,ZR,ZPS,ZVD,HZ,1,EBAR,K,EL,ES,H2,H3,LP,NO,
     1XHT,XBAR,EHT,W,SP)
      US(1) = USV
      DUS(1) = USDV
      JLI = 2
C
C     BEGIN ITERATIONS FOR FILTERING
C
      DO 104 IM = KB,N
      NH = IM-1
      JLI = JLI+1
      LL = JLI-1
C
C     DETERMINE THE INCREMENTAL TIME STEP
C
      DT = TS(IM)-TS(NH)
C
C     PROPAGATE THE STATE AND ERROR COVARIANCE MATRICES FOR A TIME DT
C
      CALL PROP(DT,US,A,Q,IM,EJ,E1,E2,E3,E4,E5,B,EN,Q1,BN,FT,EBAR,LP,
     1NO,SP,EHT,XHT,XBAR,W,DUS)
C
C     COMPUTE THE KALMAN FILTER GAIN
C
      CALL GAIN(HZ,R,EBAR,K,EM,H1,T,LP,NO,SP,W,H2,C,IC,IR)
C
C     UPDATE THE STATE AND ERROR COVARIANCE MATRICES
C
      CALL UPDT(Z,ZV,ZR,ZPS,ZVD,HZ,IM,EBAR,K,EL,ES,H2,H3,LP,NO,
     1XHT,XBAR,EHT,W,SP)
      IF(LL.LT.KS) GO TO 377
C
C     STORE THOSE VALUES OF THE STATE AND ERROR COVARIANCE MATRICES
C     SELECTED FOR PLOTTING
```

```
C
      CALL STORB(TS,MH,KFIM,KS,NO,EE,XHT,EHT,SP)
      MH = MH+1
      JLI = 1
  377 CONTINUE
  104 CONTINUE
      RETURN
      END
      SUBROUTINE PROP(H,US,A,Q,I,EJ,E1,E2,E3,E4,E5,B,EN,Q1,BN,FT,EBAR,LP
     1,NO,SP,EHT,XHT,XBAR,W,DUS)
C
C     SUBROUTINE PROP PROPAGATES THE STATE AND ERROR COVARIANCE MATRICES
C     FOR EACH TIME INCREMENT OF THE ITERATION
C
      INTEGER SP
      DOUBLE PRECISION W(1)
      DOUBLE PRECISION EHT(SP,SP),XHT(1),XBAR(1)
      DOUBLE PRECISION E1(LP,LP),E2(LP,LP),E3(LP,LP),E4(LP,LP),E5(LP,LP)
      DOUBLE PRECISION EJ(LP,LP),EBAR(LP,LP),A(1),Q(LP,LP)
      DOUBLE PRECISION US(1),DUS(1),UV,UD
      DOUBLE PRECISION H,HM
      DOUBLE PRECISION XV,XR,XPS,XVD,XRD
      DOUBLE PRECISION YV1,YV2,YV3,YV4
      DOUBLE PRECISION YR1,YR2,YR3,YR4
      DOUBLE PRECISION YPS1,YPS2,YPS3,YPS4
      DOUBLE PRECISION YVD1,YVD2,YVD3,YVD4
      DOUBLE PRECISION YRD1,YRD2,YRD3,YRD4
      DOUBLE PRECISION ZZ1,ZZ2,ZZ3,ZZ4
      DOUBLE PRECISION FNLV,FNLR
      DOUBLE PRECISION B(LP,LP),EN(LP,LP),Q1(LP,LP),BN(LP,LP),FT(LP,LP)
      DOUBLE PRECISION DSIN,DCOS
      COMMON /PRAM3/ LP17,LP18,LP19,LP20,LP21,LP22,LP23,LP24,LP25,LP26
      COMMON /PRAM4/ LP27,LP28,LP29,LP30,LP31,LP32,LP33,LP34,LP35,LP36
      COMMON /PRAM2/ LP9,LP10,LP11,LP12,LP13,LP14,LP15,LP16
      COMMON /PRAM1/ LP1,LP2,LP3,LP4,LP5,LP6,LP7,LP8
C
C *********
C
C     INTEGRATE THE STATE VALUES OVER THE TIME INCREMENT USING THE
C     RUNGE-KUTTA 4TH ORDER TECHNIQUE OF INTEGRATION
C
      MP = LP-NO
      UV = US(I)
      UD = DUS(I)
```

```
C
C     INITIALIZE THE STATES AND THE COEFFICIENTS OF INTEREST TO THOSE
C     VALUES ASSIGNED INITIALLY OR CALCULATED IN THE PREVIOUS INCREMENT
C
      XV = XHT(1)
      XR = XHT(2)
      XPS = XHT(3)
      XVD = XHT(4)
      XRD = FNLR(XV,XR,UV,A)
      IF (MP.EQ.0) GO TO 500
      GO TO (101,102,103,104,105,106,107,108,109,110,111,112,113,114,
     1115,116,117,118,119,120,121,122,123,124,125,126,127,128,129,130,
     2131,132,133,134,135,136),MP
  136 A(LP36) = XHT(SP)
  135 A(LP35) = XHT(SP-1)
  134 A(LP34) = XHT(SP-2)
  133 A(LP33) = XHT(SP-3)
  132 A(LP32) = XHT(SP-4)
  131 A(LP31) = XHT(SP-5)
  130 A(LP30) = XHT(SP-6)
  129 A(LP29) = XHT(SP-7)
  128 A(LP28) = XHT(SP-8)
  127 A(LP27) = XHT(SP-9)
  126 A(LP26) = XHT(SP-10)
  125 A(LP25) = XHT(SP-11)
  124 A(LP24) = XHT(SP-12)
  123 A(LP23) = XHT(SP-13)
  122 A(LP22) = XHT(SP-14)
  121 A(LP21) = XHT(SP-15)
  120 A(LP20) = XHT(SP-16)
  119 A(LP19) = XHT(SP-17)
  118 A(LP18) = XHT(SP-18)
  117 A(LP17) = XHT(SP-19)
  116 A(LP16) = XHT(SP-20)
  115 A(LP15) = XHT(SP-21)
  114 A(LP14) = XHT(SP-22)
  113 A(LP13) = XHT(SP-23)
  112 A(LP12) = XHT(SP-24)
  111 A(LP11) = XHT(SP-25)
  110 A(LP10) = XHT(SP-26)
  109 A(LP9) = XHT(SP-27)
  108 A(LP8) = XHT(SP-28)
  107 A(LP7) = XHT(SP-29)
  106 A(LP6) = XHT(SP-30)
  105 A(LP5) = XHT(SP-31)
  104 A(LP4) = XHT(SP-32)
  103 A(LP3) = XHT(SP-33)
  102 A(LP2) = XHT(SP-34)
  101 A(LP1) = XHT(SP-35)
  500 CONTINUE
      DO 1 N = 1,LP
      DO 2 M = 1,LP
      EJ(N,M) = EHT(N,M)
    2 CONTINUE
    1 CONTINUE
C
C     CALCULATE THE STATE VALUES AND THE TIME RATE OF CHANGE OF THE
C     ERROR COVARIANCE MATRIX AT THE START OF THE INCREMENT
C
      HM= H/2.
      YVD1 = FNLV(XV,XR,UV,A)
      YRD1 = FNLR(XV,XR,UV,A)
      YV1 = H*YVD1
      YR1 = H*YRD1
      YPS1 = H*XR
      CALL EFNT1(A,UV,B,LP,SP,XV,XR,NO,XVD,XRD,UD,H)
      CALL EFNT2(E1,Q,EN,Q1,BN,FT,B,LP,NO,SP,W,EHT)
      ZZ1 = XV+0.5*YV1
      ZZ2 = XR+0.5*YR1
      ZZ3 = XPS+0.5*YPS1
      ZZ4 = YVD1
      XHT(1) = ZZ1
      XHT(2) = ZZ2
      XHT(3) = ZZ3
      XHT(4) = ZZ4
      DO 3 N = 1,LP
      DO 4 M = 1,LP
      E2(N,M) = H*E1(N,M)
      EHT(N,M) = EJ(N,M)+HM*E1(N,M)
    4 CONTINUE
    3 CONTINUE
C
C     DO THE R-K CALCULATIONS AT THE MIDDLE OF THE INCREMENT
C
      UV = (US(I)+US(I+1))/2.
      UD = (DUS(I)+DUS(I+1))/2.
      YVD2 = FNLV(ZZ1,ZZ2,UV,A)
      YRD2 = FNLR(ZZ1,ZZ2,UV,A)
      YV2 = H*YVD2
      YR2 = H*YRD2
      YPS2 = H*ZZ2
      CALL EFNT1(A,UV,B,LP,SP,ZZ1,ZZ2,NO,YVD1, RD1,UD,H)
      CALL EFNT2(E1,Q,EN,Q1,BN,FT,B,LP,NO,SP,W,EHT)
      ZZ1 = XV+0.5*YV2
      ZZ2 = XR+0.5*YR2
      ZZ3 = XPS+0.5*YPS2
      ZZ4 = YVD2
      XHT(1) = ZZ1
      XHT(2) = ZZ2
      XHT(3) = ZZ3
      XHT(4) = ZZ4
      DO 5 N = 1,LP
      DC 6 M = 1,LP
      E3(N,M) = HM*E1(N,M)
      EHT(N,M) = EJ(N,M)+HM*E1(N,M)
    6 CONTINUE
    5 CONTINUE
C
```

```
C     REPEAT THE R-K CALCULATIONS FOR THE MIDDLE OF THE INCREMENT
C
      YVD3 = FNLV(ZZ1,ZZ2,UV,A)
      YRD3 = FNLR(ZZ1,ZZ2,UV,A)
      YV3 = H*YVD3
      YR3 = H*YRD3
      YPS3 = H*ZZ2
      CALL EFNT1(A,UV,B,LP,SP,ZZ1,ZZ2,NO,YVD2,YRD2,UD,H)
      CALL EFNT2(E1,Q,EN,Q1,BN,FT,B,LP,NO,SP,W,EHT)
      ZZ1 = XV+YV3
      ZZ2 = XR+YR3
      ZZ3 = XPS+YPS3
      ZZ4 = YVD3
      XHT(1) = ZZ1
      XHT(2) = ZZ2
      XHT(3) = ZZ3
      XHT(4) = ZZ4
      DO 7 N = 1,LP
      DO 8 M = 1,LP
      E4(N,M) = HM*E1(N,M)
      EHT(N,M) = EJ(N,M)+H*E1(N,M)
    8 CONTINUE
    7 CONTINUE
C
C     DO THE R-K CALCULATIONS FOR THE END OF THE INCREMENT
C
      UV = US(I+1)
      UD = DUS(I+1)
      YVD4 = FNLV(ZZ1,ZZ2,UV,A)
      YRD4 = FNLR(ZZ1,ZZ2,UV,A)
      YV4 = H*YVD4
      YR4 = H*YRD4
      YPS4 = H*ZZ2
      CALL EFNT1(A,UV,B,LP,SP,ZZ1,ZZ2,NO,YVD3,YRD3,UD,H)
      CALL EFNT2(E1,Q,EN,Q1,BN,FT,B,LP,NO,SP,W,EHT)
      DO 9 N = 1,LP
      DO 10 M = 1,LP
      E5(N,M) = H*E1(N,M)
   10 CONTINUE
    9 CONTINUE
C
C     FROM THE STATE VALUES CALCULATED OVER THE TIME INCREMENT,
C     DETERMINE THE NEW STATE VALUES PROPAGATED FROM T TO T+DT
C
      XBAR(1) = XV+1./6.*(YV1+2.*YV2+2.*YV3+YV4)
      XBAR(2) = XR+1./6.*(YR1+2.*YR2+2.*YR3+YR4)
      XBAR(3) = XPS+1./6.*(YPS1+2.*YPS2+2.*YPS3+YPS4)
      XBAR(4) = 1./6.*(YVD1+2.*YVD2+2.*YVD3+YVD4)
      IF (MP.EQ.0) GO TO 501
      N1 = NO+1
      DO 200 N = N1,LP
      XBAR(N) = XHT(N)
  200 CONTINUE
  501 CONTINUE
C
C     PROPAGATE THE ERROR COVARIANCE MATRIX
C
      DO 11 N = 1,LP
      DO 12 M = 1,LP
      EBAR(N,M) = EJ(N,M)+1./6.*(E2(N,M)+2.*E3(N,M)+2.*E4(N,M)+E5(N,M))
   12 CONTINUE
   11 CONTINUE
      RETURN
      END
      SUBROUTINE EFNT1(A,U,B,LP,SP,XV,XR,NO,XVD,XRD,UD,H)
C
C     SUBROUTINE EFNT1 CALCULATES THE MATRIX B...THE PARTIAL DERIVATIVES
C     OF THE MOTION EQUATIONS WITH RESPECT TO THE VARIOUS ELEMENTS OF
C     THE EXTENDED STATE VECTCR
C
      INTEGER SP
      DOUBLE PRECISION H
      DOUBLE PRECISION UD
      DOUBLE PRECISION XV,XR
      DOUBLE PRECISION XVD,XRD
      DOUBLE PRECISION B(LP,LP),A(1),X(1),C2,C5,C6,D1,D2,D3,D4,U
      DOUBLE PRECISION C7,C8,C9,C10,D5,D6,D7,D8
      DOUBLE PRECISION DCOS,DSIN
      COMMON /PRAM3/ LP17,LP18,LP19,LP20,LP21,LP22,LP23,LP24,LP25,LP26
      COMMON /PRAM4/ LP27,LP28,LP29,LP30,LP31,LP32,LP33,LP34,LP35,LP36
      COMMON /PRAM1/ LP1,LP2,LP3,LP4,LP5,LP6,LP7,LP8
      COMMON /PRAM2/ LP9,LP10,LP11,LP12,LP13,LP14,LP15,LP16
C
C *********
C
      MP = LP-NO
C
C     INITIALIZE THE MATRIX TC ZERO
C
      DO 1 N = 1,LP
      DO 2 M = 1,LP
      B(N,M) = 0.D0
    2 CONTINUE
    1 CONTINUE
C
C     CALCULATE THOSE ELEMENTS OF THE MATRIX WHICH ARE NON-ZERO
C
      C2 = 1./(A(4)*A(11)-A(5)*A(10))
      C5 = A(9)+A(6)*XV  +A(7)*XR  +A(8)*U+A(26)*U**3 +A(27)*XR  *XV  **
     12 +A(28)*U*XV  **2
```

```
      C6 = A(15)+A(12)*XV  +A(13)*XR  +A(14)*U+A(31)*U**3 +A(32)*XR  *XV
     1  **2 +A(33)*U*XV  **2
      K2 = 1
      K3 = 2
      K6 = 3
      K8 = 4
      D1 = A(6)+A(27)*XR  *XV  *2.+2.*A(28)*XV  *U
      D2 = A(12)+2.*A(32)*XR  *XV  +2.*A(33)*XV  *U
      D3 = A(7)+A(27)*XV  **2
      D4 = A(13)+A(32)*XV  **2
      C7 = 2.*A(27)*(XV  *XRD+XR  *XVD)+2.*A(28)*U*XVD+2.*A(28)*XV  *UD
      C8 = 2.*A(32)*(XV  *XRD+XR  *XVD)+2.*A(33)*U*XVD+2.*A(33)*XV  *UD
      C9 = A(6)*XVD+A(7)*XRD+A(27)*(XV  **2*XRD+2.*XR  *XVD*XV  )+2.*A(2
     1 8)*U*XV  *XVD+A(8)*UD+3.*A(26)*UD*U**2+A(28)*XV  **2*UD
      C10 = A(12)*XVD+A(13)*XRD+A(32)*(XRD*XV  **2+2.*XR  *XV  *XVD)+2.*
     1 A(33)*U*XV  *XVD+A(14)*UD+3.*A(31)*U**2*UD+A(33)*XV  **2*UD
      D5 = 2.*A(27)*XV  *XVD
      D6 = 2.*A(32)*XV  *XVD
      D7 = A(11)*C9-A(5)*C10
      D8 = A(4)*C10-A(10)*C9
C
C     CALCULATE THOSE ELEMENTS CORRESPONDING TO THE PARTIAL DERIVATIVES
C     WITH RESPECT TO THE STATE VARIABLES OF THE EXTENDED STATE VECTOR
C
      B(K2,K2) = C2*(A(11)*D1-A(5)*D2)
      B(K2,K3) = C2*(A(11)*D3-A(5)*D4)
      B(K3,K2) = C2*(A(4)*D2-A(10)*D1)
      B(K3,K3) = C2*(A(4)*D4-A(10)*D3)
      B(K8,K2) = C2*(A(11)*C7-A(5)*C8)
      B(K8,K3) = C2*(A(11)*D5-A(5)*D6)
      B(K8,K8) = C2*(A(11)*D1-A(5)*D2)
      NPA = LP1
      I = NO+1
      N = 1
   10 CONTINUE
C
```

```
C     CALCULATE THE REMAINING ELEMENTS WHICH CORRESPOND TO THE PARTIAL
C     DERIVATIVES WITH RESPECT TO THE IDENTIFIED COEFFICIENTS OF THE
C     EXTENDED STATE VECTOR
C
      GO TO(11,12,13,14,15,16,17,18,19,20,21,22,23,24,55,56,57,28,29,30,
     131,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46),NPA
   11 CONTINUE
      GO TO 25
   12 CONTINUE
      GO TO 25
   13 CONTINUE
      GO TO 25
   14 CONTINUE
      B(K2,I) = -C2**2 *A(11)*(A(11)*C5-A(5)*C6)
      B(K3,I) =-C2**2 *A(11)*(A(4)*C6-A(10)*C5)+C2*C6
      B(K8,I) = C2**2*A(11)*D7
      GO TO 25
   15 CONTINUE
      B(K2,I) = C2**2 *A(10)*(A(11)*C5-A(5)*C6)-C2*C6
      B(K3,I) = C2**2 *A(10)*(A(4)*C6-A(10)*C5)
      B(K8,I) = -C2**2*(A(10)*D7-C2*C10)
      GO TO 25
   16 CONTINUE
      B(K2,I) =  C2*A(11)*XV
      B(K3,I) = -C2*A(10)*XV
      B(K8,I) = C2*A(11)*XVD
      GO TO 25
   17 CONTINUE
      B(K2,I) = C2*A(11)*XR
      B(K3,I) = -C2*A(10)*XR
      B(K8,I) = C2*A(11)*XRD
      GO TO 25
   18 CONTINUE
      B(K2,I) = C2*A(11)*U
      B(K3,I) = -C2*A(10)*U
      B(K8,I) = A(11)*UD*C2
      GO TO 25
   19 CONTINUE
      B(K2,I) = C2*A(11)
      B(K3,I) = -C2*A(10)
      GO TO 25
   20 CONTINUE
      B(K2,I) = C2**2 *A(5)*(A(11)*C5-A(5)*C6)
      B(K3,I) = C2**2 *A(5)*(A(4)*C6-A(10)*C5)-C2*C5
      B(K8,I) = -C2**2*A(5)*D7
      GO TO 25
   21 CONTINUE
      B(K2,I) = -C2**2*A(4)*(A(11)*C5-A(5)*C6)+C2*C5
      B(K3,I) = -C2**2 *A(4)*(A(4)*C6-A(10)*C5)
      B(K8,I) = C2**2*(A(4)*D7-C2*C9)
      GO TO 25
   22 CONTINUE
      B(K2,I) = -C2*A(5)*XV
      B(K3,I) = C2*A(4)*XV
      B(K8,I) = -C2*A(5)*XVD
      GO TO 25
   23 CONTINUE
      B(K2,I) = -C2*A(5)*XR
      B(K3,I) = C2*A(4)*XR
      B(K8,I) = -C2*A(5)*XRD
      GO TO 25
   24 CONTINUE
      B(K2,I) = -C2*A(5)*U
      B(K3,I) = C2*A(4)*U
      B(K8,I) = -A(5)*UD*C2
      GO TO 25
   55 CONTINUE
      B(K2,I) = -C2*A(5)
      B(K3,I) = C2*A(4)
      GO TO 25
   56 CONTINUE
      GO TO 25
```

```
 57 CONTINUE
    GO TO 25
 28 CONTINUE
    GO TO 25
 29 CONTINUE
    GO TO 25
 30 CONTINUE
    GO TO 25
 31 CONTINUE
    GO TO 25
 32 CONTINUE
    GO TO 25
 33 CONTINUE
    GO TO 25
 34 CONTINUE
    GO TO 25
 35 CONTINUE
    B(K3,I) = -C2*A(10)*XV**3
    B(K8,I) =  C2*A(11)*3*XV**2*XVD
    GO TO 25
 36 CONTINUE
    B(K2,I) = C2*U**3 *A(11)
    B(K3,I) = -C2*U**3 *A(10)
    B(K8,I) = A(11)*3.*U**2*UD*C2
    GO TO 25
 37 CONTINUE
    B(K2,I) = C2*X(3)*X(2)**2 *A(11)
    B(K3,I) = -C2*A(10)*XR*XV**2
    GO TO 25
 38 CONTINUE
    B(K2,I) = C2*U*X(2)**2 *A(11)
    B(K3,I) =-C2*U*XV  **2 *A(10)
    B(K8,I) = A(11)*(2.*U*XV  *XVD+XV  **2*UD)*C2
    GO TO 25
 39 CONTINUE
    B(K3,I) = -C2*A(10)*XV*U**2
```

```
      B(K8,I) =  C2*A(11)*(XV*2*U*UD+U**2*XVD)
      GO TO 25
   40 CONTINUE
      B(K3,I) = C2*A(4)*XV**3
      B(K8,I) = -C2*A(5)*3*XV**2*XVD
      GO TO 25
   41 CONTINUE
      B(K2,I) = -C2*U**3 *A(5)
      B(K3,I) = C2*U**3 *A(4)
      B(K8,I) = -A(5)*3.*U**2*UD*C2
      GO TO 25
   42 CONTINUE
      B(K2,I) = -C2*X(3)*X(2)**2 *A(5)
      B(K3,I) = C2*XR   *XV   **2 *A(4)
      B(K8,I) = -C2*A(5)*(XRD*XV   **2+2.*XR   *XV   *XVD)
      GO TO 25
   43 CONTINUE
      B(K2,I) = -C2*U*X(2)**2 *A(5)
      B(K3,I) = C2*U*XV   **2 *A(4)
      B(K8,I) = -A(5)*(2.*U*XV   *XVD+XV   **2*UD)*C2
      GO TO 25
   44 CONTINUE
      B(K3,I) = C2*A(4)*XV*U**2
      B(K8,I) = -C2*A(5)*(XV*2*U*UD+U**2*XVD)
      GO TO 25
   45 CONTINUE
      GO TO 25
   46 CONTINUE
   25 CONTINUE
      N = N+1
      IF (N.GT.MP) GO TO 26
      GO TO (101,102,103,104,105,106,107,108,109,110,111,112,113,114,
     1115,116,117,118,119,120,121,122,123,124,125,126,127,128,129,130,
     2131,132,133,134,135,136),N
      GO TO 27
  101 NPA = LP1
```

```
      GO TO 27
  102 NPA = LP2
      GO TO 27
  103 NPA = LP3
      GO TO 27
  104 NPA = LP4
      GO TO 27
  105 NPA = LP5
      GO TO 27
  106 NPA = LP6
      GO TO 27
  107 NPA = LP7
      GO TO 27
  108 NPA = LP8
      GO TO 27
  109 NPA = LP9
      GO TO 27
  110 NPA = LP10
      GO TO 27
  111 NPA = LP11
      GO TO 27
  112 NPA = LP12
      GO TO 27
  113 NPA = LP13
      GO TO 27
  114 NPA = LP14
      GO TO 27
  115 NPA = LP15
      GO TO 27
  116 NPA = LP16
      GO TO 27
  117 NPA = LP17
      GO TO 27
  118 NPA = LP18
      GO TO 27
  119 NPA = LP19
```

```
      GO TO 27
  120 NPA = LP20
      GO TO 27
  121 NPA = LP21
      GO TO 27
  122 NPA = LP22
      GO TO 27
  123 NPA = LP23
      GO TO 27
  124 NPA = LP24
      GO TO 27
  125 NPA = LP25
      GO TO 27
  126 NPA = LP26
      GO TO 27
  127 NPA = LP27
      GO TO 27
  128 NPA = LP28
      GO TO 27
  129 NPA = LP29
      GO TO 27
  130 NPA = LP30
      GO TO 27
  131 NPA = LP31
      GO TO 27
  132 NPA = LP32
      GO TO 27
  133 NPA = LP33
      GO TO 27
  134 NPA = LP34
      GO TO 27
  135 NPA = LP35
      GO TO 27
  136 NPA = LP36
   27 I = N+NO
      GO TO 10
```

```
26 CONTINUE
   RETURN
   END
```

```
      SUBROUTINE EFNT2(E1,Q,EN,Q1,BN,FT,B,LP,NO,SP,W,EH)
C
C     SUBROUTINE EFNT2 CALCULATES THE TIME RATE OF CHANGE OF THE ERROR
C     COVARIANCE MATRIX AND RETURNS THE VALUE TO SUBROUTINE PROP TO BE
C     INTEGRATED OVER THE TIME INCREMENT
C
      INTEGER SP
      DOUBLE PRECISION EH(SP,SP),W(1),E1(LP,LP),EN(LP,LP),Q(LP,LP)
      DOUBLE PRECISION Q1(LP,LP),B(LP,LP),BN(LP,LP),FT(LP,LP)
C
C *********
C
      DO 1 N = 1,LP
      DO 2 M = 1,LP
      Q1(N,M) = Q(N,M)
      BN(N,M) = B(N,M)
      EN(N,M) = EH(N,M)
    2 CONTINUE
    1 CONTINUE
      CALL TRNSPS(BN,LP,LP,LP,LP,FT,LP,LP)
      CALL MAMP1S(BN,LP,LP,EN,LP,LP,LP,LP,LP,W,LP)
      CALL MAMP2S(EN,LP,LP,FT,LP,LP,LP,LP,LP,W,LP)
      CALL MAADDS(BN,LP,LP,LP,LP,FT,LP,LP)
      CALL MAADDS(BN,LP,LP,LP,LP,Q1,LP,LP)
      DO 3 N = 1,LP
      DO 4 M = 1,LP
      E1(N,M) = BN(N,M)
    4 CONTINUE
    3 CONTINUE
      RETURN
      END
      SUBROUTINE GAIN(H,R,EB,K,EM,H1,T,LP,NO,SP,W,H2,C,IC,IR)
C
C     SUBROUTINE GAIN DETERMINES THE EXTENDED KALMAN FILTER GAIN USED IN
C     UPDATING THE STATE AND ERROR COVARIANCE ESTIMATES
C
      INTEGER SP
      DIMENSION IR(1),IC(1)
      DOUBLE PRECISION H(NO,SP),W(1),H2(NO,NO),DET
      DOUBLE PRECISION R(LP,LP),EB(LP,LP),K(LP,NO),EM(LP,LP),H1(NO,LP)
      DOUBLE PRECISION T(LP,NO),C(1)
C
C *********
C
      DO 1 N = 1,LP
      DO 2 M = 1,LP
      EM(N,M) = EB(N,M)
    2 CONTINUE
    1 CONTINUE
      DO 3 N = 1,NO
      DO 4 M = 1,LP
      H1(N,M) = H(N,M)
    4 CONTINUE
    3 CONTINUE
      CALL TRNSPS(H1,NO,LP,NO,LP,T,LP,NO)
      CALL MAMP1S(EM,LP,LP,T,LP,NO,LP,LP,NO,W,NO)
      CALL MAMP1S(H1,NO,LP,EM,LP,LP,NO,LP,NO,W,NO)
      CALL MAADDS(H1,NO,LP,NC,NO,R,LP,LP)
      DO 8 N = 1,NO
      DO 9 M = 1,NO
      H2(N,M) = H1(N,M)
      KK = (N-1)+(M-1)*NO+1
      C(KK) = H2(N,M)
    9 CONTINUE
    8 CONTINUE
      CALL MINV(C,NO,DET,IR,IC)
C
```

```
C     VERIFY THAT THE DETERMINATE OF THE INVERTED MATRIX IS NOT ZERO
C
      IF (DET.NE.0.D0) GO TO 5
      WRITE (6,100) DET
  100 FORMAT(1H1,//5X,'DETERMINATE = ',F20.10)
    5 CONTINUE
      DO 10 N = 1,NO
      DO 11 M = 1,NO
      KK = (N-1)+(M-1)*NO+1
      H2(N,M) = C(KK)
   11 CONTINUE
   10 CONTINUE
      CALL MAMP1S(EM,LP,LP,H2,NO,NO,LP,NO,NO,W,NO)
      DO 6 N = 1,LP
      DO 7 M = 1,NO
      K(N,M) = EM(N,M)
    7 CONTINUE
    6 CONTINUE
      RETURN
      END
      SUBROUTINE UPDT(Z,ZV,ZR,ZPS,ZVD,H,IM,EB,K,EL,ES,H2,H3,LP,NO,
     1XH,XB,EH,W,SP)
C
C     SUBROUTINE UPDT IS USED TO UPDATE THE STATE AND ERROR COVARIANCE
C     MATRICES TO THEIR VALUE AT THE END OF THE SPECIFIED TIME INCREMENT
C
      INTEGER SP
      DOUBLE PRECISION Z(1),ZV(1),ZR(1),ZPS(1),ZVD(1)
      DOUBLE PRECISION EB(LP,LP),XH(1),EH(SP,SP),EL(1),H2(NO,LP)
      DOUBLE PRECISION H3(LP,LP),W(1),ES(1),DABS
      DOUBLE PRECISION H(NO,SP),XB(1),K(LP,NO)
C
C *********
C
      Z(1) = ZV(IM)
      Z(2) = ZR(IM)
      Z(3) = ZPS(IM)
      Z(4) = ZVD(IM)
      DO 1 I = 1,LP
      EL(I) = XB(I)
    1 CONTINUE
      DO 2 N = 1,NO
      DO 3 M = 1,LP
      H2(N,M) = H(N,M)
      H3(N,M) = H(N,M)
    3 CONTINUE
    2 CONTINUE
      CALL MAMP2S(H2,NO,LP,EL,LP,1,NO,LP,1,W,LP)
C
C     FIND THE DIFFERENCE BETWEEN THE CALCULATED STATE VALUE AND THAT
C     FROM THE NOISY SEA TRIAL
C
      DO 6 I = 1,NO
      ES(I) = Z(I)-EL(I)
    6 CONTINUE
C
```

```
C     CALCULATE THE INCREMENTAL CHANGE IN STATE
C
      CALL MAMP2S(K,LP,NO,ES,NO,1,LP,NO,1,W,LP)
C
C     UPDATE THE ELEMENTS OF THE EXTENDED STATE VECTOR
C
      DO 7 I = 1,LP
      XH(I) = XB(I)+ES(I)
    7 CONTINUE
C
C     CALCULATE THE INCREMENTAL CHANGE IN THE ERROR COVARIANCE MATRIX
C
      CALL MAMP2S(K,LP,NO,H3,LP,LP,LP,NO,LP,W,LP)
      CALL MAMP1S(H3,LP,LP,EB,LP,LP,LP,LP,LP,W,LP)
C
C     UPDATE THE ERROR COVARIANCE MATRIX
C
      CALL MASUBS(EB,LP,LP,LP,LP,H3,LP,LP)
      DO 8 N = 1,LP
      DO 9 M = 1,LP
      EH(N,M) = DABS(EB(N,M))
    9 CONTINUE
    8 CONTINUE
      RETURN
      END
      SUBROUTINE STORB(T,MH,K,KS,NO,EE,XH,EH,SP)
C
C     SUBROUTINE STORB STORES SELECTED VALUES OF THE STATE AND ERROR
C     COVARIANCE MATRICES AT REGULAR INTERVALS OVER TIME FOR USE IN
C     THE PLOTTING ROUTINE
C
      INTEGER SP
      DIMENSION VP(94),RP(94),PSP(94),VDP(94)
      DIMENSION PP1(94),PP2(94),PP3(94),PP4(94),PP5(94),PP6(94),PP7(94)
      DIMENSION PP8(94),PP9(94),PP10(94),PP11(94),PP12(94),PP13(94)
      DIMENSION PP14(94),PP15(94),PP16(94)
      DIMENSION PP17(94),PP18(94),PP19(94),PP20(94),PP21(94),PP22(94)
      DIMENSION PP23(94),PP24(94),PP25(94),PP26(94),PP27(94),PP28(94)
      DIMENSION PP29(94),PP30(94),PP31(94),PP32(94),PP33(94),PP34(94)
      DIMENSION PP35(94),PP36(94)
      DIMENSION EE(1)
      DOUBLE PRECISION XH(1),EH(SP,SP)
      DOUBLE PRECISION T(1),D,DSQRT,DABS
      COMMON /OUTP1/ VP,RP,PSP,VDP,PP1,PP2,PP3,PP4,PP5,PP6,PP7,PP8
      COMMON /OUTP5/ PP17,PP18,PP19,PP20,PP21,PP22,PP23,PP24,PP25,PP26
      COMMON /OUTP6/ PP27,PP28,PP29,PP30,PP31,PP32,PP33,PP34,PP35,PP36
      COMMON /OUTP2/ PP9,PP10,PP11,PP12,PP13,PP14,PP15,PP16
      COMMON /OUTP3/ EV,ER,EPS,EVD,EP1,EP2,EP3,EP4,EP5,EP6,EP7,EP8
      COMMON /OUTP7/ EP17,EP18,EP19,EP20,EP21,EP22,EP23,EP24,EP25,EP26
      COMMON /OUTP8/ EP27,EP28,EP29,EP30,EP31,EP32,EP33,EP34,EP35,EP36
      COMMON /OUTP4/ EP9,EP10,EP11,EP12,EP13,EP14,EP15,EP16
C
C *********
C
      I = MH
      L = KS*I
C
C     STORE THE TIME VALUES OF EACH OBSERVATION
C
      D = T(L)
      VP(I) = D
```

```
RP(I) = D
PSP(I) = D
VDP(I) = D
PP1(I) = D
PP2(I) = D
PP3(I) = D
PP4(I) = D
PP5(I) = D
PP6(I) = D
PP7(I) = D
PP8(I) = D
PP9(I) = D
PP10(I) = D
PP11(I) = D
PP12(I) = D
PP13(I) = D
PP14(I) = D
PP15(I) = D
PP16(I) = D
PP17(I) = D
PP18(I) = D
PP19(I) = D
PP20(I) = D
PP21(I) = D
PP22(I) = D
PP23(I) = D
PP24(I) = D
PP25(I) = D
PP25(I) = D
PP26(I) = D
PP27(I) = D
PP28(I) = D
PP29(I) = D
PP30(I) = D
PP31(I) = D
PP32(I) = D
```

```
      PP33(I) = D
      PP34(I) = D
      PP35(I) = D
      PP36(I) = D
C
C     STORE THE MEASURED VALUES OF THE ELEMENTS OF THE EXTENDED STATE
C     VECTOR
C
      N = I+K
      VP(N) = XH(1)
      RP(N) = XH(2)
      PSP(N) = XH(3)
      VDP(N) = XH(4)
      PP1(N) = XH(NO+1)
      PP2(N) = XH(NO+2)
      PP3(N) = XH(NO+3)
      PP4(N) = XH(NO+4)
      PP5(N) = XH(NO+5)
      PP6(N) = XH(NO+6)
      PP7(N) = XH(NO+7)
      PP8(N) = XH(NO+8)
      PP9(N) = XH(NO+9)
      PP10(N) = XH(NO+10)
      PP11(N) = XH(NO+11)
      PP12(N) = XH(NO+12)
      PP13(N) = XH(NO+13)
      PP14(N) = XH(NO+14)
      PP15(N) = XH(NO+15)
      PP16(N) = XH(NO+16)
      PP17(N) = XH(NO+17)
      PP18(N) = XH(NO+18)
      PP19(N) = XH(NO+19)
      PP20(N) = XH(NO+20)
      PP21(N) = XH(NO+21)
      PP22(N) = XH(NO+22)
      PP23(N) = XH(NO+23)
      PP24(N) = XH(NO+24)
      PP25(N) = XH(NO+25)
      PP26(N) = XH(NO+26)
      PP27(N) = XH(NO+27)
      PP28(N) = XH(NO+28)
      PP29(N) = XH(NO+29)
      PP30(N) = XH(NO+30)
      PP31(N) = XH(NO+31)
      PP32(N) = XH(NO+32)
      PP33(N) = XH(NO+33)
      PP34(N) = XH(NO+34)
      PP35(N) = XH(NO+35)
      PP36(N) = XH(NO+36)
      IF (I.LT.K) GO TO 100
C
C     STORE THE STANDARD DEVIATIONS FOR EACH ELEMENT OF THE EXTENDED
C     STATE VECTOR AT THE END OF THE IDENTIFICATION PROCESS
C
      DO 1 M = 1,SP
      EE(M) = DSQRT(DABS(EH(M,M)))
    1 CONTINUE
      EV = EE(1)
      ER = EE(2)
      EPS = EE(3)
      EVD = EE(4)
      EP1 = EE(NO+1)
      EP2 = EE(NO+2)
      EP3 = EE(NO+3)
      EP4 = EE(NO+4)
      EP5 = EE(NO+5)
      EP6 = EE(NO+6)
      EP7 = EE(NO+7)
      EP8 = EE(NO+8)
      EP9 = EE(NO+9)
      EP10 = EE(NO+10)
      EP11 = EE(NO+11)
      EP12 = EE(NO+12)
      EP13 = EE(NO+13)
      EP14 = EE(NO+14)
      EP15 = EE(NO+15)
      EP16 = EE(NO+16)
      EP17 = EE(NO+17)
      EP18 = EE(NO+18)
      EP19 = EE(NO+19)
      EP20 = EE(NO+20)
      EP21 = EE(NO+21)
      EP22 = EE(NO+22)
      EP23 = EE(NO+23)
      EP24 = EE(NO+24)
      EP25 = EE(NO+25)
      EP26 = EE(NO+26)
      EP27 = EE(NO+27)
      EP28 = EE(NO+28)
      EP29 = EE(NO+29)
      EP30 = EE(NO+30)
      EP31 = EE(NO+31)
      EP32 = EE(NO+32)
      EP33 = EE(NO+33)
      EP34 = EE(NO+34)
      EP35 = EE(NO+35)
      EP36 = EE(NO+36)
  100 CONTINUE
      RETURN
      END
      SUBROUTINE TRNSPS(A,IA,JA,MA,NA,B,IB,JB)
C
C     SUBROUTINE TRNSPS TAKES THE TRANSPOSE OF MATRIX A AND STORES IT
C     IN MATRIX B
C
      DOUBLE PRECISION A(IA,JA),B(IB,JB)
C
C *********
C
      K = MA
      IF(NA.LT.MA) K = NA
      DO 2 I = 1,K
      DO 2 J = I,K
      B(I,J) = A(J,I)
    2 B(J,I) = A(I,J)
      IF(MA-NA) 3,4,5
    3 L = MA+1
      DO 6 K = L,NA
      DO 6 I = 1,MA
    6 B(K,I) = A(I,K)
    4 RETURN
    5 L = NA+1
      DO 7 I = 1,NA
      DO 7 J = L,MA
    7 B(I,J) = A(J,I)
      RETURN
      END
      SUBROUTINE MAMP1S(A,IA,JA,B,IB,JB,MA,NAMB,NB,W,IW)
C
C     SUBROUTINE MAMP1S MULTIPLIES MATRIX A BY MATRIX B AND STORES THE
C     PRODUCT IN MATRIX A
C
      DOUBLE PRECISION A(IA,JA),B(IB,JB),W(IW),WJ
C
C *********
C
      DO 2 I = 1,MA
      DO 1 J = 1,NB
      WJ = 0.D0
      DO 11 K = 1,NAMB
   11 WJ = WJ+A(I,K)*B(K,J)
    1 W(J) = WJ
      DO 2 K = 1,NB
    2 A(I,K) = W(K)
      RETURN
      END
      SUBROUTINE MAMP2S(A,IA,JA,B,IB,JB,MA,NAMB,NB,W,IW)
C
C     SUBROUTINE MAMP2S MULTIPLIES MATRIX A BY MATRIX B AND STORES THE
C     PRODUCT IN MATRIX B
C
      DOUBLE PRECISION A(IA,JA),B(IB,JB),W(IW),WI
C
C *********
C
      DO 2 J = 1,NB
      DO 1 I = 1,MA
      WI = 0.D0
      DO 11 K = 1,NAMB
   11 WI = WI+A(I,K)*B(K,J)
    1 W(I) = WI
      DO 2 I = 1,MA
    2 B(I,J) = W(I)
      RETURN
      END
      SUBROUTINE MAADDS(A,IA,JA,MA,NA,B,IB,JB)
C
C     SUBROUTINE MAADDS ADDS MATRIX A TO MATRIX B AND STORES THE SUM IN
C     MATRIX A
C
      DOUBLE PRECISION A(IA,JA),B(IB,JB)
C
C *********
C
      DO 1 J = 1,NA
      DO 1 I = 1,MA
    1 A(I,J) = A(I,J)+B(I,J)
      RETURN
      END
      SUBROUTINE MASUBS(A,IA,JA,MA,NA,B,IB,JB)
C
C     SUBROUTINE MASUBS SUBTRACTS MATRIX B FROM MATRIX A WITH THE
C     RESULT STORED IN MATRIX A
C
      DOUBLE PRECISION A(IA,JA),B(IB,JB)
C
C *********
C
      DO 1 J = 1,NA
      DO 1 I = 1,MA
    1 A(I,J) = A(I,J)-B(I,J)
      RETURN
      END
      SUBROUTINE MINV(A,N,D,L,M)
C
C     SUBROUTINE MINV INVERTS THE MATRIX A AND PLACES THE RESULT
C     IN LOCATION A
C
      DIMENSION A(1),L(1),M(1)
      DOUBLE PRECISION A,D,BIGA,HOLD,DABS
C
C *********
C
C        SEARCH FOR THE LARGEST ELEMENT
C
      D = 1.D0
      NK = -N
      DO 80 K = 1,N
      NK = NK+N
      L(K) = K
      M(K) = K
      KK = NK+K
      BIGA = A(KK)
      DO 20 J = K,N
      IZ = N*(J-1)
      DO 20 I = K,N
      IJ = IZ+I
   10 IF (DABS(BIGA)-DABS(A(IJ))) 15,20,20
   15 BIGA = A(IJ)
      L(K) = I
      M(K) = J
   20 CONTINUE
C
C        INTERCHANGE ROWS
C
      J = L(K)
      IF (J-K) 35,35,25
   25 KI = K-N
      DO 30 I = 1,N
      KI = KI+N
      HOLD = -A(KI)
      JI = KI-K+J
      A(KI) = A(JI)
   30 A(JI) = HOLD
C
C         INTERCHANGE COLUMNS
C
   35 I = M(K)
      IF (I-K) 45,45,38
   38 JP = N*(I-1)
      DO 40 J = 1,N
      JK = NK+J
      JI = JP+J
      HOLD = -A(JK)
      A(JK) = A(JI)
   40 A(JI) = HOLD
C
C         DIVIDE COLUMN BY MINUS PIVOT(VALUE OF PIVOT ELEMENT
C         IS CONTAINED IN BIGA)
C
   45 IF (BIGA) 48,46,48
   46 D = 0.D0
      RETURN
   48 DO 55 I = 1,N
      IF (I-K) 50,55,50
   50 IK = NK+I
      A(IK) = A(IK)/(-BIGA)
   55 CONTINUE
C
C         REDUCE MATRIX
C
      DO 65 I = 1,N
      IK = NK+I
      HOLD = A(IK)
      IJ = I-N
      DO 65 J = 1,N
      IJ = IJ+N
      IF (I-K) 60,65,60
   60 IF (J-K) 62,65,62
   62 KJ = IJ-I+K
      A(IJ) = HOLD*A(KJ)+A(IJ)
   65 CONTINUE
C
C         DIVIDE ROW BY PIVOT
C
      KJ = K-N
      DO 75 J = 1,N
      KJ = KJ+N
      IF (J-K) 70,75,70
   70 A(KJ) = A(KJ)/BIGA
   75 CONTINUE
C
C         PRODUCT OF PIVOTS
C
      D = D*BIGA
C
C         REPLACE PIVOT BY RECIPROCAL
C
      A(KK) = 1.0/BIGA
   80 CONTINUE
C
C         FINAL ROW AND COLUMN INTERCHANGE
C
      K = N
  100 K = (K-1)
      IF (K) 150,150,105
  105 I = L(K)
      IF (I-K) 120,120,108
  108 JQ = N*(K-1)
      JR = N*(I-1)
      DO 110 J = 1,N
      JK = JQ+J
      HOLD = A(JK)
      JI = JR+J
      A(JK) = -A(JI)
  110 A(JI) = HOLD
  120 J = M(K)
      IF (J-K) 100,100,125
  125 KI = K-N
      DO 130 I = 1,N
      KI = KI+N
      HOLD = A(KI)
      JI = KI-K+J
      A(KI) = -A(JI)
  130 A(JI) = HOLD
      GO TO 100
  150 RETURN
      END
```

Appendix C

INPUT DATA DESCRIPTION

Card #1 - Card #6: (A(I), I = 1,36) FORMAT (6D13.4)

A(I) - True values of all coefficients to the motion equations, taken from the literature

Card #7 - Card #12: (AI(I), I = 1,36) FORMAT (6D13.4)

AI(I) - Initial estimates of all coefficients to the motion equations, approximately 30% of the accepted true values

Card #13 - Card #18: (ASD(I), I = 1,36) FORMAT (6D13.4)

ASD(I) - The standard deviations to the estimates of all coefficients to the motion equations

Card #19 - Card #21: (PMS(J), J = 1,16) FORMAT (6D13.4)

PMS(1) - PMS(4) - Mean process noise values for the designated state variables

PMS(5) - PMS(8) - Mean measurement noise values for the designated state variables

PMS(9) - PMS(12) - Desired standard deviations of the process noise distributions

PMS(13) - PMS(16) - Desired standard deviations of the measurement noise distributions

Card #22 - Card #23: (INX(I), I = 1,8) FORMAT (6I6)

INX(I) - Odd integer values for use in the Gaussian noise generation, random number generator

Card #24: G FORMAT (F10.6)

G - Identity factor (1) for the process noise vector $\underline{w}$

Card #25 - Card #28: LP1 - LP36 FORMAT (9I5)

LP_ - Coefficients to be identified, in increasing numerical order ... remaining positions to be assigned other arbitrary non-zero values (i.e. 1)

Card #29: VST, RST, PST, VDST FORMAT (4F10.5)

VST - Initial sway velocity

RST - Initial yaw velocity

PST - Initial yaw angle

VDST - Initial yaw acceleration

Card #30: VCV, RCV, PCV, VDCV FORMAT (4F10.3)

VCV - Estimated covariance of the initial sway velocity

RCV - Estimated covariance of the initial yaw velocity

PCV - Estimated covariance of the initial yaw angle

VDCV - Estimated covariance of the initial sway acceleration

Card #31: KS, N, H FORMAT (2I4,F10.2)

KS - Number of measured points between plotted points

N - Actual number of measurements over the trial period

H - Time increment per measurement

note: Trial period = N * H

N/KS = number of plotted points = 47

Card #32: NM, NP FORMAT (2I5)

NM - Percentage measurement noise

NP - Percentage process noise

Card #33: PW, QW FORMAT (2D10.2)

PW - Exaggeration factor for the process noise

QW - Exaggeration factor for the measurement noise

Card #34: MP, NO FORMAT (2I4)

MP - Number of coefficients to be identified

NO - Number of measured primary state variables used

Card #35: DI, TL, JJ FORMAT (2F10.3,I5)

DI - Maximum rudder deflection in degrees

TL - Half-period of the sinusoidal maneuver

JJ - Type of maneuver desired for the identification

1 = Single-step rudder deflection

2 = Step zig-zag rudder deflection

3 = Sinusoidal rudder deflection

Card #36: NE FORMAT (I5)

NE - Type of plotting desired for output of results

1 = Use PLOTM plotting routine only

2 = Use CALCOMP plotting routines only

3 = Use both plotting options simultaneously

Card #37 - Card #40: ((TITLE(I,J), J = 1,9), I = 1,4) FORMAT (9A4)

TITLE(I,J) - Character strings used to label CALCOMP plots of the primary state variables

HYDRODYNAMIC COEFFICIENTS(17),(3)

(Mariner-class Hull Form)

Coefficient:	Label:	Dimensionalized Value:
$(m-X_{\dot{u}})$	A(1)	12.3068 E5
X_u	A(2)	-0.8429 E4
$1/2\ X_{uu}$	A(16)	0.1248 E3
$1/6\ X_{uuu}$	A(17)	-0.0113 E2
$(1/2\ X_{rr}+mx_G)$	A(19)	13.9243 E6
$1/2\ X_{\delta\delta}$	A(20)	-1.6859 E5
$(X_{vr}+m)$	A(21)	11.6915 E6
$X_{v\delta}$	A(22)	0.6547 E4
$1/2\ X_{vv}$	A(18)	-0.2492 E4
$(m-Y_{\dot{v}})$	A(4)	22.6504 E5
$(mx_G-Y_{\dot{r}})$	A(5)	-66.5270 E5
Y_v	A(6)	-8.1515 E4
$1/6\ Y_{vvv}$	A(25)	-0.8853 E3
(Y_r-mu)	A(7)	-18.5084 E6
Y_δ	A(8)	4.9423 E5
$1/6\ Y_{\delta\delta\delta}$	A(26)	-1.6006 E5
$1/2\ Y_{rvv}$	A(27)	8.8863 E4
$1/2\ Y_{\delta vv}$	A(28)	0.3308 E4
$1/2\ Y_{v\delta\delta}$	A(29)	-0.2669 E3
Y_o	A(9)	-0.6404 E4

Coefficient:	Label:	Dimensionalized Value:
$(mx_G - N_{\dot{v}})$	A(10)	-1.7560 E7
$(I_z - N_{\dot{r}})$	A(11)	33.8608 E9
N_v	A(12)	-9.7735 E6
1/6 N_{vvv}	A(30)	0.0947 E4
$(N_r - mx_G u)$	A(13)	-32.5103 E8
N_δ	A(14)	-1.3033 E8
1/6 $N_{\delta\delta\delta}$	A(31)	4.2256 E7
1/2 N_{rvv}	A(32)	-1.6753 E8
1/2 $N_{\delta vv}$	A(33)	-0.7164 E6
1/2 $N_{v\delta\delta}$	A(34)	0.4636 E6
N_o	A(15)	2.6293 E6

Remarks:

This list is comprised only of those coefficients used in the identification program. All others are assumed zero.

The dimensionalized values were obtained from the non-dimensional form by assuming the following values:

$$\rho = 1.9905 \text{ lbf-sec}^2/\text{ft}^4$$
$$L = 528.01 \text{ ft}$$
$$u = 25.317 \text{ ft/sec}$$

```
C     SAMPLE DATA DECK ...                                                             0001
C                                                                                      0002
C        THIS DECK WAS USED IN THE IDENTIFICATION SCHEME OF                            0003
C        SECTION 5.2 - VARIATION IN TRIAL LENGTH.                                      0004
C                                                                                      0005
C                                                                                      0006
     12.3068D5    -8.4290D3          0D0    22.6504D5   -66.5270D5    -8.1515D4        0007
    -18.5084D6     4.9423D5    -6.4040D3    -1.7560D7    33.8608D9    -9.7735D6        0008
    -32.5103D8    -1.3033D8     2.6293D6     0.1248D3    -0.0113D2    -0.2492D4        0009
     13.9243D6    -1.6859D5    11.6915D5     0.6547D4          0D0          0D0        0010
     -0.8853D3    -1.6006D5     8.8863D4     0.3308D4    -0.2669D3     0.9470D5        0011
      4.2256D7    -1.6753D8    -0.7164D6     0.4636D6          0D0          0D0        0012
     15.000 D5   -11.000 D3          0D0    29.450 D5   -46.569 D5   -57.060 D3        0013
    -12.955 D6    64.000 D4   -44.830 D2   -12.292 D6    43.900 D9   -68.414 D5        0014
    -22.752 D8   -91.236 D6     2.000 D6     0.900 D2    -0.080 D1    -0.190 D4        0015
     10.000 D6    -1.300 D5     9.000 D5     0.500 D4          0D0          0D0        0016
     -0.700 D3    -1.200 D5     7.700 D4     0.250 D4    -0.200 D3     0.750 D5        0017
      3.500 D7    -1.300 D8    -0.600 D6     0.350 D6          0D0          0D0        0018
      2.700 D5     2.600 D3          0D0    67.951 D4    19.958 D5    24.454 D3        0019
     55.525 D5    14.827 D4    19.213 D2    52.680 D5    10.158 D9    29.321 D5        0020
     97.581 D7    39.101 D6          0D0     0.348 D3     0.033 D1     0.592 D3        0021
      3.924 D6     0.386 D5     2.692 D5     0.155 D4          0D0          0D0        0022
      0.185 D3     0.400 D5     1.186 D4     0.080 D4     0.066 D3     0.197 D5        0023
      0.725 D7     0.375 D8     0.116 D6     0.113 D6          0D0          0D0        0024
     0.0    D0    0.0    D0    0.0    D0    0.0    D0    0.0    D0    0.0    D0        0025
      0.0000D0     0.0000D0     0.0035D0    0.0015D-2     0.0005D0     0.0001D0        0026
      0.1235D0     0.0005D0     0.0330D0     0.0035D0                                  0027
      1    11   101  1001  1101  1011                                                  0028
      1    11   101  1001  1101  1011                                                  0029
  1.0                                                                                  0030
     6    7   12   13    1    1    1    1    1                                         0031
     1    1    1    1    1    1    1    1    1                                         0032
     1    1    1    1    1    1    1    1    1                                         0033
     1    1    1    1    1    1    1    1    1                                         0034
   0.0       0.0       0.0       0.0                                                   0035
     0.5       0.5       0.5       0.5                                                 0036
   8 376       2.0                                                                                       0037
    5    5                                                                                               0038
    1.0000     1.0000                                                                                    0039
   4   4                                                                                                 0040
    10.0      100.0       3                                                                              0041
    3                                                                                                    0042
    SWAY VELOCITY (FT./SEC.)                                                                             0043
    YAW VELOCITY (RAD./SEC.)                                                                             0044
       YAW ANGLE (RADIANS)                                                                               0045
SWAY ACCELERATION (FT./SEC./SEC.)                                                                        0046
```

REFERENCES

1. Abkowitz, M. A., Stability and Motion Control of Ocean Vehicles, M.I.T. Press, Cambridge, Massachusetts, 1969.

2. ――――――, Discussion on paper - "Maneuvering of Large Tankers", W. B. vanBerlekom and T. A. Goddard, SNAME Transactions, Vol. 80, 1972.

3. Brinati, H. L., "System Identification Applied to Maneuvering Trials," M.I.T. Engineers Thesis, Department of Ocean Engineering, 1973.

4. Brock, L. D., "Application of Statistical Estimation to Navigation Systems," M.I.T. Ph.D Thesis, Department of Aeronautics and Astronautics, 1965.

5. Bryson, A. E., and Ho, Y. C., Applied Optimal Control Optimization, Estimation and Control, Blaisdell Publishing Co., 1969.

6. Comstock, J. P., ed., Principles of Naval Architecture, SNAME, 1967.

7. Galiana, F. D., "A Review of Basic Principles and of Available Techniques in System Identification," M.I.T. Report No. 20, Power Systems Engineering Group, 1969.

8. Hayes, M. N., "Parametric Identification of Non-linear Stochastic Systems Applied to Ocean Vehicle Dynamics," M.I.T. Ph.D Thesis, Department of Ocean Engineering, 1971.

9. Hildebrand, F. B., Advanced Calculus for Applications, Prentice-Hall,Inc., 1962.

10. Ho, Y. C., and Lee, R. C. K., "Identification of Linear Dynamic Systems," Information and Control, Vol.8, Feb. 1965.

11. Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," Journal of Basic Engineering, March, 1960.

12. ————, "Linear Stochastic Filtering Theory - Reappraisal and Outlook," Symposium on System Theory, Brooklyn Polytechnique Institute, April, 1965.

13. Kaplan, P., et.al., "The Application of System Identification to the Dynamics of Naval Craft," 9th Symposium on Naval Hydrodynamics, 1972.

14. Mehra, R. K.,"Identification of Stochastic Linear Systems Using Kalman Filter Representation," AIAA Journal, Vol. 9, No. 1, January, 1971.

15. Netavoli, A. N., and deFigueiredo, R. J. P., "On the Identification of Non-linear Dynamical Systems," IEEE Transactions, Automatic Control, Vol. AC-14, No. 5, Feb. 1971.

16. Reis, J. M. D. B., "Identification of Ship Model Motion Parameters," M.I.T. Engineer Thesis, Department of Ocean Engineering, 1971.

17. Strom-Tejsen, J., "A Digital Computer Technique for Prediction of Standard Maneuvers of Surface Ships," Report 2130, David Taylor Model Basin, 1965.

18. Weiner, N., Extrapolation, Interpolation and Smoothing of Stationary Time Series, M.I.T. Press, Cambridge, Massachusetts, 1949.

19. I.B.M. Scientific Subroutine Package, Version III, "Programmers Manual," Program 360A-CM-03X, 5th Edition, August, 1970.

20. WATFIV Library Manual, University of Waterloo.