N91-17088

Attitude Determination and Calibration using a Recursive Maximum Likelihood-Based Adaptive Kalman Filter

D. A. Kelly, A. Fermelia*
G. K. F Lee†

23 May 1990

Abstract

Kalman filter technniques are widely used in the areas of attitude and orbit determination, prediction and calibration. These techniques work well if the system dynamics are well-defined. Problems arise, however, when the system parameters are unknown ahead of time or changing over time.

This paper discusses an adaptive Kalman filter design that utilizes recursive maximum likelihood parameter identification. At the center of this design is the Kalman filter itself, which has the responsibility for attitude determination. At the same time, however, the identification algorithm is continually identifying the system parameters. The approach is applicable to nonlinear as well as linear systems. This adaptive Kalman filter design has much potential for real-time implementation, especially considering the fast clock speeds, cache memory and internal RAM available today.

The recursive maximum likelihood (RML) identification algorithm is discussed in detail, with special attention directed towards its unique matrix formulation. Next, the procedure for using the algorithm is described along with comments on how this algorithm interacts with the Kalman filter.

Finally, a spacecraft attitude determination/calibration example is provided. In the development of the dynamics for this example, the angular velocity of one of the axis is assumed to be constant. In the simulation, however, this velocity varies slowly. The RML identifier is used to continually identify this changing velocity. This AKF-RML method may be used to identify multiple parameters such as sensor biases or external torques.

*Hughes Aircraft Company, Aurora, Colorado
†Mechanical and Aerospace Department, North Carolina State University

1 Introduction

A problem that has received recent attention is the estimation of systems whose dynamics are unknown or changing. In these situations, it is often neccessary to utilize an estimation algorithm that is able to adjust the system dynamics model automatically. Self-adjusting estimation schemes are known as *adaptive filters*. There are many examples of systems with unknown or changing dynamics, several of which have just recently become more of a concern.

Spacecraft attitude and orbit estimation are two such areas. Many current operational spacecraft do most of the attitude estimation, prediction and calibration on the ground using large mainframe computers. Recent literature, [19] and [23] for example, has suggested on-board attitude prediction and calibration using adaptive filtering techniques. Adaptive filtering methods have already been used for some time in spacecraft orbit estimation [27].

Adaptive filter techniques have also been applied in the steering of ships [28] and has been suggested for use in calibrating ocean navigational gyros [21]. Industrial processes research has led to the development of adaptive filter algorithms for monitoring chemical processes. Chen, Wadhwani and Roberts [3], for example, offer an adaptive filter technique for monitoring changes in raw material composition.

Adaptive control methods often utilize adaptive filters. An extensive amount of literature has surfaced within the last fifteen years in the field of adaptive control of robotic manipulators alone. An example of a robotic manipulator adaptive control scheme is discussed in detail in an article by Lee, Kelly and Karim [15].

A relative newcomer to the field of adaptive filtering is large space structures (LSS). Larger spacecraft offer several advantages over smaller spacecraft including longer on-orbit lifespans (thus, fewer launch vehicles are required), on-orbit refueling (for lower orbit spacecraft) and more or larger payloads. As these spacecraft increase in size and complexity, some problems arise. A larger, more complex spacecraft will have lower bending frequencies, more fuel slosh, larger disturbances [11] and the possibility of greater interaction between multiple payloads [10]. Adaptive filtering techniques have been proposed for use in identifying parameters such as vibrational frequencies, damping coefficients, and attitude estimation.

Accurately estimating the states of a system whose dynamics are time-invariant and known is often easily accomplished. There are numerous estimation methods for accomplishing this task, depending on the particular application. New problems are created, however, when the system has unknown or time-varying dynamics. If estimation is attempted for system dynamics that are incorrectly modelled, large errors in the estimated states are likely. Even more problems may arise if control is to be applied based upon the estimated states. It is clear that a state feedback regulator applying control to a system based upon these incorrect states may have problems determining the correct amount of control to apply, possibly causing the system to go unstable.

This paper discusses an adaptive filter technique that utilizes a Kalman filter. Adaptive filters based upon Kalman filters are known as *adaptive Kalman filters* (AKF). This AKF design is developed in the second and third sections, then applied to a satellite attitude determination problem in Section 4.

2 A RML Identification Technique

2.1 Motivation for Development

Using a Kalman filter-based adaptive filter offers several advantages over other adaptive filter techniques. Even though it may be computationally intense depending upon the application, the adaptive Kalman filter design provides the capability to easily provide the system states where and when they may be needed. A large space structure system could be designed so that the states are the structure's attitude (pitch, roll, yaw and their rates). For this example, the AKF could serve as not only the state estimator, but the provider of these estimated states to other payloads as well.

To be adaptive, however, the AKF must be able to accurately identify unknown or time-varying system parameters. The burden of this task falls upon the identifier. Since it may be desireable to use a Kalman filter-based adaptive filter, it seems reasonable that the identifier algorithm chosen should be one that works well with a Kalman filter. This chapter will cover the development of such an identifier.

Maximum likelihood techniques have been used for parameter identification for many years. Lee [18], in his book published in 1964, claims that Fisher [8] first developed identification using maximum likelihood techniques. The concept of recursive maximum likelihood identification using a Newton-Raphson type optimization technique and Kalman filter equations was originally conceived by Stepner and Mehre [25] in 1973. Their algorithm, which was designed to handle nonlinear as well as linear dynamics, was very computationally intensive. In 1986, Fermelia [7] further expanded the concepts, and Sjodin and Fermelia [24] developed working code for a first order dynamics model in 1987. Sjodin and Fermelia's algorithm was much simpler in concept, thus giving it a much better chance for real-time implementation. Kelly [14] and Fermelia extended this work to multiple second order dynamic models in 1989.

2.2 System Model

This dissertation discusses an adaptive Kalman filter design where the system model is defined in state-space form. Consider a system described by

$$\dot{\underline{x}} = F\underline{x} + G\underline{u} + \underline{w} \tag{1}$$

$$\underline{z} = H\underline{x} + \underline{v}. \tag{2}$$

The noise vectors, $\underline{w}$ and $\underline{v}$, are assumed to be normally distributed with a mean of zero.

In order to obtain a discrete-time model of the above system model, it is assumed that the set of discrete-time points $\{0, 1, \ldots, k, k+1\}$ are sufficiently close together that piecewise constant approximations may be made. The solution of $\dot{\underline{x}} = F\underline{x} + G\underline{u}$ for such an interval may be expressed as [2]

$$\underline{x}_{k+1} = \Phi_{k+1,k}\underline{x}_k + \int_k^{k+1} \Phi_{k+1,\tau} G_\tau d\tau \underline{u}_k. \tag{3}$$

2.3 Defining the Performance Index

In this section, a performance index is defined for RML identification. Since the identification technique is a recursive maximum likelihood method, the log likelihood function must first be defined. To identify the system parameters, the log likelihood function must be maximized. This function is maximized by minimizing its negative term. This negative term is chosen as the performance index.

2.3.1 The Log Likelihood Function

The identification method that is developed in this section is a recursive maximum likelihood method. Variables that are vectors rather than scalars are underlined. Variables that are estimated or identified have a hat, such as $\hat{\underline{x}}$. First, define a log likelihood function, $\mathcal{L}(\underline{\theta})$, as

$$\mathcal{L}(\underline{\theta}) = \ln[p(\underline{Z}|\underline{\theta})] \tag{4}$$

where $\underline{\theta}$ is a vector of unknown parameters and $p(\underline{Z}|\underline{\theta})$ is the conditional probability density function of the observations, $\underline{Z}$, given $\underline{\theta}$. The maximum likelihood estimate, $\underline{\theta}$, is the parameter vector that most likely caused the observations [24].

Repeated use of Bayes' Law is used to derive an expression for $\mathcal{L}(\underline{\theta})$. Let $\underline{Z}_{k+1}$ be the set of all observations at time $k+1$, or

$$\underline{Z}_{k+1} = [\, \underline{z}_1 \, \underline{z}_2 \cdots \underline{z}_{k+1} \,]. \tag{5}$$

Now, $p(\underline{Z}_{k+1}|\underline{\theta})$ may be expressed as

$$p(\underline{Z}_{k+1}|\underline{\theta}) = p(\underline{z}_1 \, \underline{z}_2 \cdots \underline{z}_{k+1}|\underline{\theta}). \tag{6}$$

$\mathcal{L}(\underline{\theta})$ may now be written as

$$\mathcal{L}(\underline{\theta}) = \ln \prod_{i=1}^{k+1} p(\underline{z}_i|\underline{Z}_{i-1}, \underline{\theta}). \tag{7}$$

If the noise values, $\underline{w}_k$ *and* $\underline{v}_{k+1}$, are normally distributed, $p(\underline{z}_i|\underline{Z}_{i-1},\underline{\theta})$ is normally distributed. From [24],

$$p(\underline{z}_i|\underline{Z}_{i-1},\underline{\theta}) = [(2\pi)^m det\ B]^{-1/2} exp(-\frac{1}{2}\underline{\nu}_i^T B^{-1} \underline{\nu}_i) \tag{8}$$

where

$$\begin{aligned} m &\stackrel{\text{def}}{=} dim\,[\underline{\nu}] \\ B &\stackrel{\text{def}}{=} E\,(\underline{\nu}\,\underline{\nu}^T) \\ \underline{\nu}_i &\stackrel{\text{def}}{=} \underline{z}_i - \hat{\underline{z}}_i. \end{aligned}$$

Substituting this into the Equation 8 gives

$$\begin{aligned} \mathcal{L}(\underline{\theta}) &= \ln(\prod_{i=1}^{k+1}[(2\pi)^m\ det\ B]^{-1/2} exp(-\frac{1}{2}\,\underline{\nu}_i^T\ B^{-1}\ \underline{\nu}_i)) \\ &= -\frac{1}{2}\,m(k+1)\ \ln(2\pi) \\ &\quad -\sum_{i=1}^{k+1}\frac{1}{2}\,(\,\ln(det\ B)\ +\ \underline{\nu}_i^T\ B^{-1}\ \underline{\nu}_i). \end{aligned} \tag{9}$$

2.3.2 The Performance Index

Now that the log likelihood function has been defined, an expression for the performance index must be found. In Equation 9, $\mathcal{L}(\underline{\theta})$ is maximized if the summation expression is minimized. Consider the following performance index,

$$J_{k+1}(\underline{\theta}) = \sum_{i=1}^{k+1}\frac{1}{2}\,(\,\ln(det\ B)\ +\ \underline{\nu}_i^T\ B^{-1}\ \underline{\nu}_i) \tag{10}$$

to be minimized.

If $J_{k+1}(\underline{\theta})$ is expanded using a Taylor Series expansion, one obtains

$$\begin{aligned} J_{k+1}(\underline{\theta}) &= J_{k+1}(\underline{\theta}^*) + \frac{\partial J_{k+1}(\underline{\theta})}{\partial\underline{\theta}}|_{\underline{\theta}=\underline{\theta}^*}\delta\underline{\theta} \\ &\quad + \frac{1}{2}\frac{\partial^2 J_{k+1}(\underline{\theta})}{\partial\underline{\theta}^2}|_{\underline{\theta}=\underline{\theta}^*}\delta\underline{\theta}^2 + \cdots \end{aligned} \tag{11}$$

where $\underline{\theta}^*$ is the current value of the parameter, $\underline{\theta}$.

If third and higher order partial terms are assumed to be negligible, the above expression can be solved for $\delta\underline{\theta}$, the parameter error. At steady state, it is desired that $J_{k+1}(\underline{\theta}) - J_{k+1}(\underline{\theta}^*) = 0$. What remains of the Taylor Series expansion is

$$0 = \frac{\partial J_{k+1}(\underline{\theta})}{\partial\underline{\theta}}|_{\underline{\theta}=\underline{\theta}^*}\delta\underline{\theta} + \frac{1}{2}\frac{\partial^2 J_{k+1}(\underline{\theta})}{\partial\underline{\theta}^2}|_{\underline{\theta}=\underline{\theta}^*}\delta\underline{\theta}^2. \tag{12}$$

Solving for the parameter update, the following expression is obtained:

$$\delta\underline{\theta} \;=\; -\,2\left[\frac{\partial^2 J_{k+1}(\underline{\theta})}{\partial\underline{\theta}^2}\Big|_{\underline{\theta}=\underline{\theta}^*}\right]^{-1} \;\frac{\partial J_{k+1}(\underline{\theta})}{\partial\underline{\theta}}\Big|_{\underline{\theta}=\underline{\theta}^*}. \tag{13}$$

The parameter update, $\underline{\theta}_{new}$ is found by adding the error in the parameter, $\delta\underline{\theta}$, to the current value of the parameter, $\underline{\theta}_{old}$.

$$\underline{\theta}_{new} \;=\; \underline{\theta}_{old} \;+\; \delta\underline{\theta} \tag{14}$$

Equation 13 is very similar in form to the recursive Newton-Raphson algorithm.

The first and second partials are found by differentiating $J_{k+1}(\underline{\theta})$ from Equation 10. The error covariance matrix, B, is assumed to be constant once steady state has been reached. Thereby, the first term of Equation 10 may be ignored when the partial is taken, giving [1] [24] [25]

$$\frac{\partial J_{k+1}(\underline{\theta})}{\partial\underline{\theta}} \;=\; \sum_{i=1}^{k+1}\frac{\partial\underline{\nu}_i^T}{\partial\underline{\theta}}B^{-1}\underline{\nu}_i \tag{15}$$

$$\frac{\partial^2 J_{k+1}(\underline{\theta})}{\partial\underline{\theta}^2} \;=\; \sum_{i=1}^{k+1}\frac{\partial\underline{\nu}_i^T}{\partial\underline{\theta}}B^{-1}\frac{\partial\underline{\nu}_i}{\partial\underline{\theta}^T}. \tag{16}$$

The error covariance matrix B is defined as [20] [25]

$$B \;=\; \frac{1}{k+1}\sum_{i=1}^{k+1}\underline{\nu}_i\underline{\nu}_i^T. \tag{17}$$

The error covariance matrix B contains a vector, $\underline{\nu}$, multiplied by its transpose, $\underline{\nu}$. Because of this, B will tend to be singular for the first and second iterations. This singularity will cause difficulties in solving for B^{-1}. Depending upon the dynamics of the system, the identifier generally works well if B^{-1} is reinitialized on or about the third iteration. Hence, one solves for B on the first two iterations, but not for B^{-1} , $\frac{\partial J}{\partial\underline{\theta}}$ or $\frac{\partial^2 J}{\partial\underline{\theta}^2}$. Then on the third iteration, begin solving for all identifier values.

2.4 RML Identification Algorithm

The next two sections of this chapter discuss the matrix (multiple identified parameters) form of RML identification. The first section covers the details of the derivation of the matrix form of RML identification. This is followed by a section describing the procedure for matrix RML identification.

2.5 RML Identification Algorithm

The recursive identification parameter update algorithm is described by Equations 13 through 17. Equation 13 is comprised of Equations 15 and 16. The $\frac{\partial \underline{\nu}_i^T}{\partial \underline{\theta}}$ term of these last two equations is now described. Since the following derivations rely heavily upon the Kalman filter equations, Table 1 [9] is presented as an quick reference of the set of Kalman filter equations.

Table 1: Summary of Kalman Filter Equations

	Kalman Filter Equations
Process Model	$\underline{x}_{k+1} = \Phi_{k+1,k}\underline{x}_k + \underline{w}_k \qquad \underline{w}_k \sim N(\underline{0}, Q_k)$
Measurement Model	$\underline{z}_{k+1} = H_{k+1}\underline{x}_{k+1} + \underline{v}_{k+1} \qquad \underline{v}_{k+1} \sim N(\underline{0}, R_k)$
State Estimate Extrapolation	$\hat{\underline{x}}_{k+1/k} = \Phi_{k+1,k}\hat{\underline{x}}_{k/k}$
Error Covariance Extrapolation	$\overline{P}_{k+1/k} = \Phi_{k+1,k}P_{k/k}\Phi^T_{k+1,k} + Q_k$
Error Covariance Update	$P_{k+1/k+1} = [I - K_{k+1}H_{k+1}]\overline{P}_{k+1/k}$
Kalman Gain	$K_{k+1} = \overline{P}_{k+1/k}H^T_{k+1}\left[H_{k+1}\overline{P}_{k+1/k}H^T_{k+1} + R_{k+1}\right]^{-1}$
State Estimate Update	$\hat{\underline{x}}_{k+1/k+1} = \hat{\underline{x}}_{k+1/k} + K_{k+1}\left[\underline{z}_{k+1} - H_{k+1}\hat{\underline{x}}_{k+1/k}\right]$

First, define a variation in $\underline{\nu}_{k+1}$ as the error between the desired innovations, $\underline{\nu}_{k+1}(D)$, and the incorrect innovations, $\underline{\nu}_{k+1}(I)$.

$$\delta\underline{\nu}_{k+1} = \underline{\nu}_{k+1}(D) - \underline{\nu}_{k+1}(I). \tag{18}$$

The desired innovations can be viewed as the innovations expected for a Kalman filter whose dynamics match those of the system being modeled. The incorrect innovations can be viewed as the innovations expected from a Kalman filter whose model dynamics do not match those of the system being modeled.

In order to obtain an expression for $\delta\underline{\nu}_{k+1}$, begin with the Kalman filter expression for the innovations.

$$\underline{\nu}_{k+1} = \underline{z}_{k+1} - \hat{\underline{z}}_{k+1/k}. \tag{19}$$

Then, observing the variation of both sides,

$$\delta\underline{\nu}_{k+1} = \delta(\underline{z}_{k+1} - \hat{\underline{z}}_{k+1/k}). \tag{20}$$

Since there is no error in $\underline{z}_{k+1}$ due to errors in the system dynamics or modeling, $\delta \underline{z}_{k+1} = 0$. Thus,

$$\delta \underline{\nu}_{k+1} = -\delta(\hat{\underline{z}}_{k+1/k}) \tag{21}$$

Assume that the unknown or changing parameters exist only in $\hat{\Phi}_{k+1,k}$, the Kalman filter state transition matrix. Using the Kalman filter equation for $\hat{\underline{z}}_{k+1/k}$ from Table 1, expand to get

$$\delta \underline{\nu}_{k+1} = -\delta \left[H_{k+1} \hat{\Phi}_{k+1,k} \hat{\underline{x}}_{k/k} \right]. \tag{22}$$

The parameters to be identified occur only in $\hat{\Phi}_{k+1,k}$, the state transition matrix. Therefore,

$$\begin{aligned} \delta \underline{\nu}_{k+1} &= -\left[H_{k+1} \delta \hat{\Phi}_{k+1,k} \hat{\underline{x}}_{k/k} + H_{k+1} \hat{\Phi}_{k+1,k} \delta \hat{\underline{x}}_{k/k} \right] \\ &= -\left[H_{k+1} \frac{\partial \hat{\Phi}_{k+1,k}}{\partial \underline{\theta}^T} \hat{\underline{x}}_{k/k} + H_{k+1} \hat{\Phi}_{k+1,k} \frac{\partial \hat{\underline{x}}_{k/k}}{\partial \underline{\theta}^T} \right] \delta \underline{\theta}. \end{aligned} \tag{23}$$

The term $\partial \hat{\Phi}$ may be expressed as

$$\partial \hat{\Phi} = \begin{bmatrix} \partial \phi_{11} & \partial \phi_{12} & \cdots & \partial \phi_{1n} \\ \partial \phi_{21} & \partial \phi_{22} & \cdots & \partial \phi_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \partial \phi_{n1} & \partial \phi_{n2} & \cdots & \partial \phi_{nn} \end{bmatrix} \tag{24}$$

where

$$\begin{aligned} \partial \phi_{ij} &= \frac{\partial \phi_{ij}}{\partial \theta_1} \delta \theta_1 + \frac{\partial \phi_{ij}}{\partial \theta_2} \delta \theta_2 + \cdots + \frac{\partial \phi_{ij}}{\partial \theta_n} \delta \theta_n \\ &= \frac{\partial \phi_{ij}}{\partial \underline{\theta}^T} \delta \underline{\theta}. \end{aligned} \tag{25}$$

Let the terms contained by the brackets in Equation 23 be represented by a matrix A_{k+1}.

$$A_{k+1} = H_{k+1} \frac{\partial \hat{\Phi}_{k+1,k}}{\partial \underline{\theta}^T} \hat{\underline{x}}_{k/k} + H_{k+1} \Phi_{k+1,k} \frac{\partial \hat{\underline{x}}_{k/k}}{\partial \underline{\theta}^T}. \tag{26}$$

Then,

$$\frac{\partial \nu_{k+1}^T}{\partial \underline{\theta}} = -A_{k+1}. \tag{27}$$

The first term of A_{k+1} may be realized by the following algorithm [5]:

$$\begin{aligned} H_{k+1} \delta \hat{\Phi}_{k+1,k} \hat{\underline{x}}_{k/k} &= H_{k+1} \begin{bmatrix} \hat{\underline{x}}_{k/k}^T \frac{\partial \phi_1}{\partial \underline{\theta}^T} \\ \hat{\underline{x}}_{k/k}^T \frac{\partial \phi_2}{\partial \underline{\theta}^T} \\ \vdots \\ \hat{\underline{x}}_{k/k}^T \frac{\partial \phi_n}{\partial \underline{\theta}^T} \end{bmatrix} \delta \underline{\theta} \\ &= H_{k+1} M_{k+1} \delta \underline{\theta} \end{aligned} \tag{28}$$

or

$$H_{k+1}\frac{\partial\hat{\Phi}_{k+1,k}}{\partial\underline{\theta}^T}\hat{\underline{x}}_{k/k} = H_{k+1}M_{k+1} \tag{29}$$

where ϕ_{ij} are elements of $\hat{\Phi}_{k+1,k}$ and

$$\begin{aligned}
\phi_1 &= [\,\phi_{11}\ \phi_{12}\ \cdots\ \phi_{1n}\,]^T \\
\phi_2 &= [\,\phi_{21}\ \phi_{22}\ \cdots\ \phi_{2n}\,]^T \\
&\ \vdots \\
\phi_n &= [\,\phi_{n1}\ \phi_{n2}\ \cdots\ \phi_{nn}\,]^T .
\end{aligned}$$

An expression must now be found for the second term of A_{k+1}. If an expression can be found for $\frac{\partial\hat{\underline{x}}_{k+1/k+1}}{\partial\underline{\theta}^T}$, then $\frac{\partial\hat{\underline{x}}_{k/k}}{\partial\underline{\theta}^T}$ is simply the value of $\frac{\partial\hat{\underline{x}}_{k+1/k+1}}{\partial\underline{\theta}^T}$ from the last iteration. Let a matrix B_{k+1} be defined as

$$B_{k+1} = \frac{\partial\hat{\underline{x}}_{k+1/k+1}}{\partial\underline{\theta}^T}. \tag{30}$$

Then, B_{k+1} from the last iteration is defined as

$$B_k = \frac{\partial\hat{\underline{x}}_{k/k}}{\partial\underline{\theta}^T}. \tag{31}$$

Again, using the Kalman filter equation for $\hat{\underline{x}}_{k+1/k+1}$ from Table 1, the following expression is derived

$$\begin{aligned}
B_{k+1}\delta\underline{\theta} &= \delta\hat{\underline{x}}_{k+1/k+1} \\
&= \delta\left[\hat{\Phi}_{k+1,k}\hat{\underline{x}}_{k/k} + G_{k+1}\underline{\nu}_{k+1}\right] \\
&= \left[\frac{\partial\hat{\Phi}_{k+1,k}}{\partial\underline{\theta}^T}\hat{\underline{x}}_{k/k} + \hat{\Phi}_{k+1,k}\frac{\partial\hat{\underline{x}}_{k/k}}{\partial\underline{\theta}^T}\right. \\
&\qquad \left. + \frac{\partial G_{k+1}}{\partial\underline{\theta}^T}\underline{\nu}_{k+1} + G_{k+1}\frac{\partial\underline{\nu}_{k+1}}{\partial\underline{\theta}^T}\right]\delta\underline{\theta}.
\end{aligned} \tag{32}$$

Thus, B_{k+1} may be written in terms of M_{k+1}, B_k and A_{k+1} as

$$\begin{aligned}
B_{k+1} &= M_{k+1} + \hat{\Phi}_{k+1,k}B_k \\
&\qquad + \frac{\partial G_{k+1}}{\partial\underline{\theta}^T}\underline{\nu}_{k+1} - G_{k+1}A_{k+1}.
\end{aligned} \tag{33}$$

Let a matrix C_{k+1} be defined such that

$$C_{k+1} = \frac{\partial G_{k+1}}{\partial\underline{\theta}^T}\underline{\nu}_{k+1} \tag{34}$$

or

$$C_{k+1}\delta\underline{\theta} = \delta G_{k+1}\underline{\nu}_{k+1}. \tag{35}$$

An expression for $\frac{\partial G_{k+1}}{\partial \underline{\theta}^T}$ is found next. Again, expand using the Kalman filter equation for G_{k+1} from Table 1.

$$\delta G_{k+1} = \left[\frac{\partial P_{k+1/k+1}}{\partial \underline{\theta}^T} H_{k+1}^T R_{k+1}^{-1}\right]\delta\underline{\theta}. \tag{36}$$

Expand $\frac{\partial P_{k+1/k+1}}{\partial \underline{\theta}^T}$ the same way.

$$\delta P_{k+1/k+1} = [I - G_{k+1}H_{k+1}]\,\delta P_{k+1/k} - \delta G_{k+1}\, H_{k+1}P_{k+1/k}. \tag{37}$$

Letting

$$\Gamma_{k+1} = I - G_{k+1}H_{k+1} \tag{38}$$

Equation 37 becomes

$$\delta P_{k+1/k+1} = \left[\Gamma_{k+1}\frac{\partial P_{k+1/k}}{\partial \underline{\theta}^T} - \frac{\partial G_{k+1}}{\partial \underline{\theta}^T}H_{k+1}P_{k+1/k}\right]\delta\underline{\theta}. \tag{39}$$

Thus

$$\frac{\partial P_{k+1/k+1}}{\partial \underline{\theta}^T} = \Gamma_{k+1}\frac{\partial P_{k+1/k}}{\partial \underline{\theta}^T} - \frac{\partial G_{k+1}}{\partial \underline{\theta}^T}H_{k+1}P_{k+1/k}. \tag{40}$$

Now, $\frac{\partial P_{k+1/k+1}}{\partial \underline{\theta}^T}$ may be described as

$$\begin{aligned}\frac{\partial P_{k+1/k+1}}{\partial \underline{\theta}^T} &= \frac{\partial P_{k+1/k}}{\partial \underline{\theta}^T} - \frac{\partial G_{k+1}}{\partial \underline{\theta}^T}H_{k+1}P_{k+1/k} \\ &\quad - G_{k+1}H_{k+1}\frac{\partial P_{k+1/k}}{\partial \underline{\theta}^T}\end{aligned} \tag{41}$$

and substituting $\delta P_{k+1/k+1}$ into Equation 36 yields

$$\begin{aligned}\delta G_{k+1} &= \left[\Gamma_{k+1}\frac{\partial P_{k+1/k}}{\partial \underline{\theta}^T}H_{k+1}^T R_{k+1}^{-1}\right] \\ &\quad \left[I + H_{k+1}P_{k+1/k}H_{k+1}^T R_{k+1}^{-1}\right]^{-1}\delta\underline{\theta}.\end{aligned} \tag{42}$$

Introduce the matrix V_{k+1} where

$$V_{k+1} = H_{k+1}^T R_{k+1}^{-1}\left[I + H_{k+1}P_{k+1/k}H_{k+1}^T R_{k+1}^{-1}\right]^{-1}\underline{\nu}_{k+1}. \tag{43}$$

Then, post multiplying Equation 42 by $\underline{\nu}_{k+1}$, an expression for C_{k+1} is obtained as

$$\begin{aligned}C_{k+1} &= \frac{\partial G_{k+1}}{\partial \underline{\theta}^T}\underline{\nu}_{k+1} \\ &= \Gamma_{k+1}\frac{\partial P_{k+1/k}}{\partial \underline{\theta}^T}V_{k+1}\end{aligned} \tag{44}$$

Letting

$$D_{k+1} = \frac{\partial P_{k+1/k}}{\partial \underline{\theta}^T} V_{k+1} \tag{45}$$

allows C_{k+1} to be expressed in terms of a matrix D_{k+1} .

$$C_{k+1} = \Gamma_{k+1} D_{k+1} \tag{46}$$

Next, an expression for D_{k+1} is found as

$$\begin{aligned} D_{k+1}\delta\underline{\theta} &= \delta P_{k+1/k} V_{k+1} \\ &= \delta\hat{\Phi}_{k+1,k} W_{k+1} + \hat{\Phi}_{k+1,k}\, \delta P_{k/k}\, U_{k+1} \\ &\quad + \hat{\Phi}_{k+1,k} P_{k/k}\, \delta\hat{\Phi}^T_{k+1,k} V_{k+1} \end{aligned} \tag{47}$$

where

$$W_{k+1} = P_{k/k}\hat{\Phi}^T_{k+1,k} V_{k+1} \tag{48}$$

$$U_{k+1} = \hat{\Phi}^T_{k+1,k} V_{k+1}. \tag{49}$$

Using the same technique as in Equation 28, the first term of the D_{k+1} expression may be written as

$$\begin{aligned} \delta\hat{\Phi}_{k+1,k} W_{k+1} &= \begin{bmatrix} W^T_{k+1} \frac{\partial \phi_1}{\partial \underline{\theta}^T} \\ W^T_{k+1} \frac{\partial \phi_2}{\partial \underline{\theta}^T} \\ \vdots \\ W^T_{k+1} \frac{\partial \phi_n}{\partial \underline{\theta}^T} \end{bmatrix} \delta\underline{\theta} \\ &= S_{k+1}\delta\underline{\theta}. \end{aligned} \tag{50}$$

The third term can be written as

$$\begin{aligned} \hat{\Phi}_{k+1,k} P_{k/k}\, \delta\hat{\Phi}^T_{k+1,k} V_{k+1} &= \Phi_{k+1,k} P_{k/k} \begin{bmatrix} V^T_{k+1} \frac{\partial \phi^*_1}{\partial \underline{\theta}^T} \\ V^T_{k+1} \frac{\partial \phi^*_2}{\partial \underline{\theta}^T} \\ \vdots \\ V^T_{k+1} \frac{\partial \phi^*_n}{\partial \underline{\theta}^T} \end{bmatrix} \delta\underline{\theta} \\ &= \hat{\Phi}_{k+1,k} P_{k/k} N_{k+1}\delta\underline{\theta} \end{aligned} \tag{51}$$

where

$$\begin{aligned} \phi^*_1 &= [\,\phi_{11}\ \phi_{21}\ \cdots\ \phi_{n1}\,]^T \\ \phi^*_2 &= [\,\phi_{12}\ \phi_{22}\ \cdots\ \phi_{n2}\,]^T \\ &\ \vdots \\ \phi^*_n &= [\,\phi_{1n}\ \phi_{2n}\ \cdots\ \phi_{nn}\,]^T . \end{aligned}$$

Let the second term of the D_{k+1} expression be written in terms of a matrix E_{k+1},

$$\hat{\Phi}_{k+1,k}\delta P_{k/k}U_{k+1} = \hat{\Phi}_{k+1,k}E_{k+1}\delta\underline{\theta}. \tag{52}$$

Combining all three terms for D_{k+1} gives

$$D_{k+1} = \hat{\Phi}_{k+1,k}E_{k+1} + S_{k+1} + \hat{\Phi}_{k+1,k}P_{k/k}N_{k+1}. \tag{53}$$

An expression for E_{k+1} must now be found. Equation 52 defines E_{k+1} as

$$E_{k+1}\delta\underline{\theta} \;=\; \delta P_{k/k}U_{k+1}. \tag{54}$$

Using the Kalman filter equation for $P_{k/k}$ yields

$$\delta P_{k/k} = \delta P_{k/k-1} - \delta P_{k/k}H_k^T R_k^{-1} H_k P_{k/k-1} - P_{k/k}H_k^T R_k^{-1} H_k \delta P_{k/k-1}. \tag{55}$$

Solving for $\delta P_{k/k}$ and postmultiplying by U_{k+1} yields

$$\delta P_{k/k}U_{k+1} = \Gamma_k \delta P_{k/k-1} Y_{k+1} \tag{56}$$

or

$$E_{k+1}\delta\underline{\theta} \;=\; \Gamma_k \delta P_{k/k-1} Y_{k+1} \tag{57}$$

where

$$\begin{aligned} Y_{k+1} &= T_k U_{k+1} &(58)\\ T_k &= I + H_k^T R_k^{-1} P_{k/k-1}. &(59)\end{aligned}$$

Let a matrix F_{k+1} be defined as

$$F_{k+1}\delta\underline{\theta} \;=\; \delta P_{k/k-1} Y_{k+1} \tag{60}$$

such that E_{k+1} is expressed in terms of F_{k+1}.

$$E_{k+1} \;=\; \Gamma_k F_{k+1} \tag{61}$$

The matrix F_{k+1} is expressed in terms of D_k, the previous value of the D_{k+1} matrix. Since $\delta P_{k/k-1}$ is symmetric,

$$\begin{aligned} F_{k+1}\delta\underline{\theta} &= \delta P_{k/k-1} Y_{k+1}\\ &= V_k^* V_k^T \delta P_{k/k-1} Y_{k+1}\\ &= V_k^* \left(D_k \delta\underline{\theta}\right)^T Y_{k+1} &(62)\end{aligned}$$

where V_k^* is defined as the pseudo-inverse of V_k^T [26], or

$$V_k^* = V_k \left[V_k^T V_k\right]^{-1}. \tag{63}$$

This allows F_{k+1} to be expressed as

$$F_{k+1} = \left[\begin{array}{cccc} V_k^* Y_{k+1}^T d_1 & V_k^* Y_{k+1}^T d_2 & \cdots & V_k^* Y_{k+1}^T d_j \end{array}\right] \tag{64}$$

where d_j is defined as the j^{th} column of D_k.

3 Matrix RML Identification Procedure

The identification algorithm has now been derived for the matrix (multiple parameter) case where only the state transition matrix, $\hat{\Phi}_{k+1,k}$, is a function of the parameters. The procedure for using the matrix form of RML identification is now explained. There are six major steps, with steps five and six having several substeps.

1. Set B_k equal to B_{k+1} of the last identifier iteration. If this is the first iteration, set it equal to some small reasonable number. The user may have to try several different starting values in addition to letting the identifier iterate several times in order to determine a reasonable initial value.
2. Calculate $\frac{\partial \nu_{k+1}^T}{\partial \underline{\theta}}$ using Equation 27.
3. Calculate the error covariance, B, using Equation 17.
4. Calculate the first and second partials of J_{k+1} using Equations 15 and 16.
5. If ready to update,
   - Calculate the parameter error vector, $\delta\underline{\theta}$, using Equation 13.
   - Update the parameter vector, $\underline{\theta}$, using Equation 14.
6. If not ready to update, solve for B_{k+1} to be used as B_k in the next iteration. To do this,
   - Set D_k, T_k, V_k and Γ_k equal to D_{k+1}, T_{k+1}, V_{k+1} and Γ_{k+1}, respectively, from the previous iteration. From the last iteration of the Kalman filter, get the values for $P_{k/k}$, $\hat{\underline{x}}_{k/k}$, $P_{k+1/k}$, G_{k+1}, H_{k+1} and R_{k+1}.
   - Calculate:

     (a) V_k^* from Equation 63
     (b) V_{k+1} from Equation 43
     (c) U_{k+1} from Equation 49
     (d) Y_{k+1} from Equation 58
     (e) F_{k+1} from Equation 64
     (f) E_{k+1} from Equation 61
     (g) N_{k+1} from Equation 51
     (h) W_{k+1} from Equation 48
     (i) S_{k+1} from Equation 50
     (j) D_{k+1} from Equation 53
     (k) Γ_{k+1} from Equation 38
     (l) C_{k+1} from Equation 46
     (m) B_{k+1} from Equation 33

4 Attitude Determination Example

This section discusses an application of AKF-RML to a spacecraft attitude scenario. First, the dynamic equations are developed. Next, the state space model for the dynamic equations is explained. Finally, results are shown using simulated data.

4.1 Dynamic Equations

Assume the attitude of a spacecraft may be described using Euler's equations [12],

$$I_1 \frac{\delta \omega_1}{\delta t} = N_1 + (I_2 - I_3)\,\omega_2\omega_3 \tag{65}$$

$$I_2 \frac{\delta \omega_2}{\delta t} = N_2 + (I_3 - I_1)\,\omega_3\omega_1 \tag{66}$$

$$I_3 \frac{\delta \omega_3}{\delta t} = N_3 + (I_1 - I_2)\,\omega_1\omega_2 \tag{67}$$

where I_i represents the inertia about the ith axis, ω_i is the angular velocity about the ith axis and N_i is the applied torque about the ith axis.

If torque-free motion ($N_i = 0$), symmetry of two of the inertias ($I_1 = I_2 = I_T$) and constant velocity in the third axis ($\omega_3 = n$) is assumed, then the above equations become

$$I_T \frac{\delta \omega_1}{\delta t} = -(I_3 - I_T)\,\omega_2\omega_3 \tag{68}$$

$$I_T \frac{\delta \omega_2}{\delta t} = (I_3 - I_T)\,\omega_3\omega_1 \tag{69}$$

$$I_T \frac{\delta \omega_3}{\delta t} = 0. \tag{70}$$

Differentiating Equation 68 yields

$$I_T \ddot{\omega}_1 = -(I_3 - I_T)\,\dot{\omega}_2\omega_3. \tag{71}$$

Multiplying by I_T and substituting in Equation 69 gives

$$I_T^2 \ddot{\omega}_1 = -(I_3 - I_T)\,\omega_3 I_T \dot{\omega}_2 \tag{72}$$

$$= -(I_3 - I_T)\,\omega_3 (I_3 - I_T)\,\omega_3\omega_1 \tag{73}$$

$$= -(I_3 - I_T)^2\,\omega_3^2\omega_1 \tag{74}$$

or

$$\ddot{\omega}_1 = -\left(\frac{I_3 - I_T}{I_T}\right)^2 \omega_1. \tag{75}$$

Integrating Equation 70 then yields an expression for ω_2 as

$$\omega_2 = \left(\frac{I_3 - I_T}{I_T}\right) \omega_3\omega_1 t \tag{76}$$

where t is the step size.

4.2 State Space Model

Equation 75 is in the form of a simple undamped harmonic oscillator. Therefore, it may be expressed in continuous state space form as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{77}$$

or in discrete state space form as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}_{k+1} = \begin{bmatrix} \cos\omega_n t & \frac{1}{\omega_n}\sin\omega_n t \\ -\omega_n \sin\omega_n t & \cos\omega_n t \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k \tag{78}$$

where

$$\omega_n = \left(\frac{I_3 - I_T}{I_T}\right)\omega_3. \tag{79}$$

4.3 Attitude Determination Results

The example described in Sections 4.1 and 4.2 were implemented on a XT-compatable PC. The true natural frequency, ω_n, was set to -6.0 radians. The natural frequency in the Kalman filter dynamics was assumed to be unknown, and the identifier set up to solve for the true value.

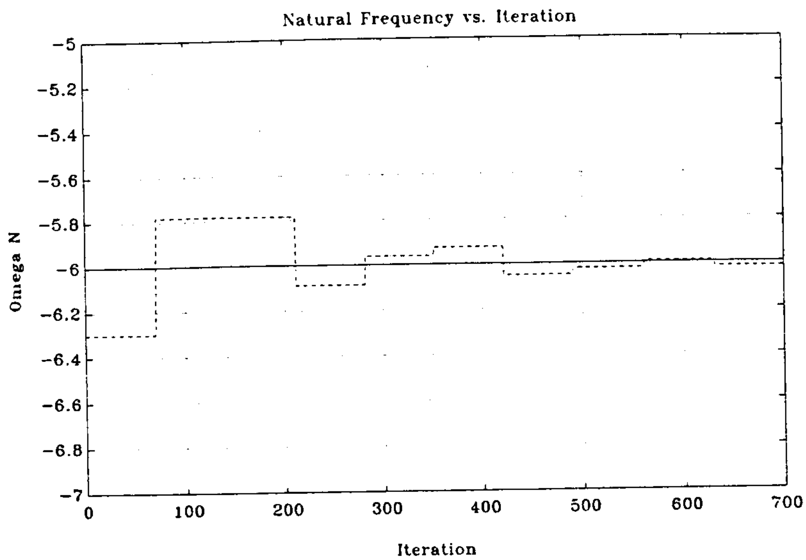
In the first test case, AKF-RML was set up to operate in a pseudo-batch mode. In this mode, the identifier is actually solving for the unknown parameter at each iteration, but the parameter is updated every k iterations. Figure 1(a) shows the results of this test case with k set to 70 iterations and the initial guess on ω_n equal to -6.3 radians.

While this pseudo-batch mode is useful for analysis and to gain insight into the operation of the AKF-RML algorithm, the goal is still to operate recursively. Figure 1(b) shows the results of recursive AKF-RML with the initial guess of ω_n at -6.6 radians.

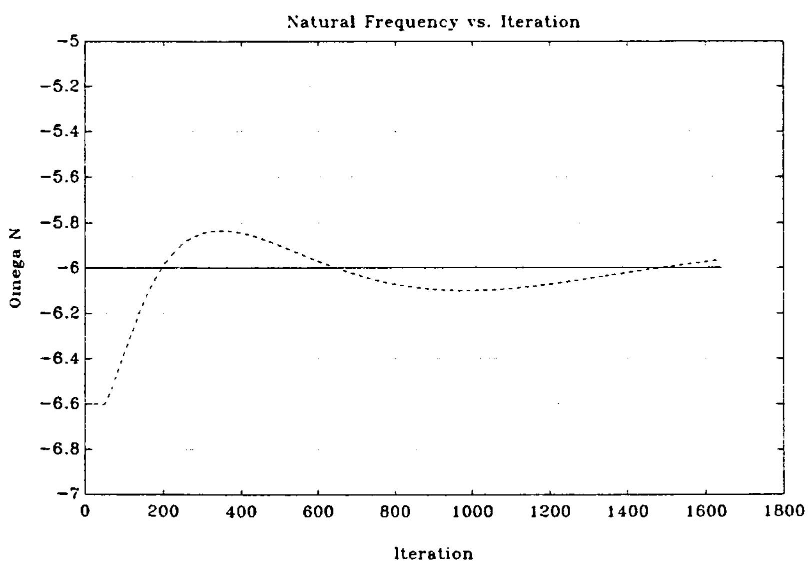
5 Summary

Adaptive filtering is rapidly gaining popularity as a method of estimating systems with unknown or changing dynamics. This paper offers a recursive identification algorithm that is designed to be used in conjunction with a Kalman filter.

In [14], the recursive maximum likelihood identification algorithm is developed and tested. Extensive testing is accomplished using simulated data, beginning with the simple first order, decaying exponential case. This is extended to second order spring-mass dynamics, with excellent results obtained for both of these cases. Next, damping is added to create the well-known spring-mass-damper dynamic case. Very



(a)



(b)

Figure 1: AKF-RML Results in (a) Batch Mode and (b) Recursive Mode

good results are obtained when identifying the natural frequency or damping coefficient individually. Problems arise, however, when both parameters are identified concurrently. This problem presents itself whenever the parameters to be identified are not of the same magnitude, and is solved by the addition of a scaling matrix to the parameter update equation.

The identification algorithm developed in this paper may be computationally intense in some applications. As with all Kalman filter algorithms, the computational needs rise quickly as the number of states and observations increase. The identification algorithm computational needs will rise not only with an increase in the states and observations, but with an increase in the number of identified parameters as well. Another possible disadvantage is the trial and error method that is currently used to find the parameter scale factors. The use of scale factors is actually not a disadvantage against this identification algorithm alone, as similar scaling or weighting matrices are used in many current identification algorithms when identifying parameters of different magnitudes. In addition, maximum likelihood techniques tend to converge poorly when the initial conditions are far away from the true conditions. In most practical applications the users should be able to arrive at reasonable initial conditions.

The advantages of using this adaptive filter design are many. Incorporation of a Kalman filter estimator in AKF-RML allows the designer to choose system states that relate to real-world entities such as position, velocity and acceleration. In the case of aircraft or satellite applications, for example, those states can then be passed on to various payloads for pointing requirements. RML identification is designed specifically to be used in conjunction with the Kalman filter. Therefore, this adaptive Kalman filter design may be implemented for systems whose dynamics are unknown or varying.

In addition, the RML identification algorithm described in this paper is able to handle relatively high levels of noise. The common problem of insufficient excitation prevents parameters from being identified at very low noise levels, but this problem disappears as noise levels are brought up enough to provide sufficient excitation.

Results are shown for a simple spacecraft attitude determination example. Plots are presented for two cases. The first case shows the results of AKF-RML in batch mode where the parameter is updated after a series of iterations. Next, results are provided of AKF-RML in recursive mode. More research needs to done, especially in the area of choosing initial conditions, in order to get AKF-RML to operate efficiently in recursive mode.

6 References

[1] M. Athans and F. C. Schweppe. Gradient Matrices and Matrix Calculations. MIT Tech. Note 1965-53, November 1965.

[2] W. L. Brogan. Modern Control Theory, Second Edition. Prentice-Hall, Inc., Englewood Cliffs, N. J., 1985.

[3] S. Chen, D. S. Wadhwani and P. D. Roberts, 'Heirarchical Integrated System Optimisation (Optimization) and Parameter Estimation using a Microcomputer Based System,' IEEE Proceedings of the 25th Conference on Decision and Control, Athens, Greece, December 1986.

[4] M. Ciletti and J. Tyler, 'Proceedings of the Symposium on Control Theory and Navy Applications,' U.S. Naval Postgraduate School, Monterey, California, July 15-17, 1975.

[5] A. Fermelia. Informal Notes. This realization was derived by A. Fermelia in 1985 and used by B. Sjodin in his Masters thesis (see Sjodin reference in this bibliography).

[6] A. Fermelia and D. A. Kelly, 'Least-Squares Formulation to a Newton-Raphson Parameter Update Algorithm,' manuscript written in December 1989.

[7] A. Fermelia and R. H. Larson, 'Closed Loop Methodology (CLM) Applied to Signal Processing; A Revisitation,' 1986 American Control Conference, 18-20 June, 1986, Volume 2 of 3, pp. 847-851.

[8] R. A. Fisher, 'On an Absolute Criterion for Fitting Frequency Curves,' Mess. Math., vol. 41, p. 155, 1912.

[9] A. Gelb, editor. Applied Optimal Estimation. The MIT Press, Cambridge, Massachusetts, 1986.

[10] G. Goodwin and K. Sin. Adaptive Filtering Prediction and Control. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1984.

[11] B. Govin and B. Claudinon, 'Adaptive Control of Flexible Space Structures,' Proceedings of the Ninth IFAC/ESA Symposium, The Netherlands, pp. 361-371, 5-9 July 1982.

[12] M. H. Kaplan. Modern Spacecraft Dynamics and Control. John Wiley and Sons, New York, New York, 1976.

[13] D. A. Kelly, G. K. F. Lee and A. Fermelia,'A Self-Tuning Regulator Approach to the Control of Large Space Structures,' SIAM Conference on Control in the 90's: Achievements, Opportunities, and Challenges, San Francisco, California, 17-19 May 1989.

[14] D. A. Kelly, 'An Adaptive Kalman Filter Design Using Recursive Maximum Likelihood Identification,' Dissertation, Department of Electrical Engineering, Colorado State University, Colorado, Spring 1990.

[15] G. K. F. Lee and D. A. Kelly, 'A Self-Tuning Regulator Design Using Recursive Maximum Likelihood Identification,' a memo sent to Dr. R. Montgomery, Spacecraft Control Branch, NASA Langley Research Center, Hampton, Virginia, May 1989.

[16] G. K. F. Lee, D. A. Kelly and A. Karim, 'A Discrete-Time Adaptive Controller for Nonlinear Systems Using a Recursive Maximum Likelihood Structure,' ISMM International Symposium on Computer Applications in Design, Simulation and Analysis, New Orleans, Louisiana, 5-7 March 1990.

[17] G. K. F. Lee and D. A. Kelly, 'A Self-Tuning Regulator Algorithm for Discrete-Time Decentralized Systems,' ISMM International Conference on Microcomputer Applications, Los Angeles, California, 14-16 December 1989.

[18] R. Lee. Optimal Estimation, Identification, and Control. Research Monograph No. 28, The M.I.T. Press, Cambridge, Massachusetts, 1964.

[19] E. J. Lefferts, F. L. Markley and M. D. Shuster, 'Kalman Filtering for Spacecraft Attitude Estimation,' AIAA J. Guidance, vol. 5, no. 5, September/October 1982.

[20] R. K. Mehra, 'On-Line Identification of Linear Dynamic Systems With Applications to Kalman Filtering,' IEEE Trans. Auto. Control, April 1970.

[21] C. E. Mueller and G. Stein,'Calibration of High Precision Navigators with Maximum Likelihood Estimation,' Proceedings of the Symposium on Control Theory and Navy Applications, U.S. Naval Postgraduate School, Monterey, California, 15-17 July 1975.

[22] A. Sage and J. Melsa. System Identification. Academic Press, New York, New York, 1971, pp. 66-81.

[23] M. D. Shuster, D. M. Chitre and D. P. Niebur, 'In-Flight Estimation of Spacecraft Attitude Sensor Accuracies and Alignments,' AIAA J. Guidance, vol. 5, no. 4, July/August 1982.

[24] R. Sjodin, 'CLM: A Closed Loop Methodology for the Modeling of Dynamic Systems,' Masters Thesis, Department of Electrical Engineering, University of Colorado, Boulder, Colorado, 1987.

[25] D. E. Stepner and R. K. Mehra, 'Maximum Likelihood Identification and Optimal Input Design for Identifying Aircraft Satbility and Control Derivatives,' NASA Contractor Report CR-2200, NASA, Washington, D. C., March 1973.

[26] Y. Takahashi, M. J. Rabins and D. M. Auslander. Control and Dynamic Systems. Addison-Wesley Publishing Company, Reading, Massachusetts, pp. 765-790, 1970.

[27] 'TRACE66: Trajectory Analysis and Orbit Determination Program,' SAMSO-TR-71-141, volumes 1-10, The Aerospace Corporation, El Segundo, California, 31 May 1974.

[28] 'Identification of Ship Steering Dynamics Using Inertial Sensors,' IFAC Identification and System Parameter Estimation 1985, pp. 175-182, York, UK, 1985.