AFIT/GE/ENG/95D-11

Reducing Lag in Virtual Displays
Using Multiple Model Adaptive Estimation

Thesis

David W. Kyger, Captain, USAF

AFIT/GE/ENG/95D-11

The views expressed in this thesis are those of the author and do not reflect the official position of the Department of Defense or the U.S. Government.

AFIT/GE/ENG/95D-11

Reducing Lag in Virtual Displays Using Multiple Model Adaptive Estimation

THESIS

Presented to the Faculty of the School of Engineering of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical Engineering

David W. Kyger, B.S.
Captain, USAF

December 1995

Approved for public release; distribution is unlimited

Table of Contents

List of Figures

List of Tables

Acknowledgments

It has been my pleasure and honor to work with so many bright, dedicated people. In addition, I was lucky enough to make some good friends along the way. Thanks to all the professors in the Guidance and Control track who challenged me; and thanks to all the G & C students who pulled me along when the cobwebs of nine years away from engineering weighed me down. Without the camaraderie in our group my sense of humor would have been an early casualty of the AFIT battles. I owe a special debt to Jeff Hebert who patiently guided me through the basics of C++. Few people would have been so generous with their time.

Thanks to Dr. Peter Maybeck whose intelligence and enthusiasm were inspirational. He never lost faith in me, even after I discovered I spent a large part of my summer research time perfecting an algorithm to predict the present.

My greatest thanks go to my wife Debbie. After two extended deployments and four years of following me around the world, this may have been the most difficult assignment for her. Even in the worst of times, I knew her love would see us through. She ensured our son Daniel got the love and attention he deserved -- from both of us. Daniel will never know how valuable he was to me. Whenever I began to lose perspective, he reminded me that school was not the most important thing in my life.

List of Figures

List of Tables

Abstract

Multiple Model Adaptive Estimation is an effective method for reducing lag in virtual environment displays. Lag in displays (the time from head motion to the appearance of the proper image on the display) is a significant detriment to realism in virtual envirnoments. Increasing the speed of the computers which control the virtual display is not a final answer. No matter how fast the processors work, there will always be demands to do more. Predicting angular head positions (look-angles) can reduce the lag by allowing the computer to calculate the appropriate scene before it is needed on the display. Single predictors cannot adequately cover the dynamic range of head motion. Using a bank of three elemental filters, lag in head orientation can be significantly reduced when compared to systems with no predictor or the single, nonadaptive Kalman filter predictor proposed by Liang. Predictions lead to small overshoots when the angular velocity of the head reverses direction, but the overshoot lasts for only one frame and is not significantly larger than the nonadaptive predictor.

Reducing Lag in Virtual Displays
Using Multiple Model Adaptive Estimation

1. Introduction

Virtual environments have great potential as training aids. They allow training at low cost and low risk. The effectiveness of this training is, in part, dependent on the realism of the system. One of the major problems that decreases realism is lag [5]. Lag is the time delay between the human input to a virtual environment and the response of the environment [3]. When lag is present, the human operator will adapt his responses to try to compensate for the lag. If the virtual environment is used for teaching skills, the student will learn based on the presence of artificial lag. When the student moves to an actual system, with different lag characteristics, some of the techniques and strategies developed in the virtual environment may not successfully transfer, making the virtual environment training less useful [9]. There have been many efforts to understand and overcome lag [5]. Still, the problem persists.

One approach to solving the problem of lag in visual displays is to predict where the human subject will be looking (look-angle) some short time in the future, matched to the computational lag required for rendering the scene, typically 100 msec to 200 msec for our research computational resources and those envisioned for virtual environment simulators which will be fielded in the near future. This information would allow the computer that generates images to complete all necessary operations and store the appropriate graphics in a buffer before it is actually needed, to render the scene at the predicted look-angle and finish that process by the time the subject's head actually points in that direction. To make an accurate prediction, a designer needs a model of human head motion. A single model cannot describe head motion characteristics because they cover such a broad range, from staring to rapid tracking movements to chaotic motion when a subject is attempting to acquire a new target or threat [9]. This thesis will use multiple models to characterize head motion. Specifically, it will use multiple model adaptive estimation (MMAE) to predict look-angle, adapting to the changing characteristics of head motion (angular velocity or angular acceleration) in a real-world flight control environment. This is intended to address the shortcomings of *nonadaptive* predictors that have been developed in the past [10,14] .

1.1 Assumptions/Restrictions

The following is a list of assumptions made for this research:

1. The specific aim of this research is to predict pilot head orientation 100 msec to 200 msec into the future during flight simulation. The prediction time is driven by two factors. First, research [14] shows lag varies from 85 msec to 160 msec depending on factors such as sample rate and communications schemes with the head-position tracker. Second, the system developed by previous researchers [21, 24] runs at 10 Hz, making integer multiples of 100 millisecond intervals a natural choice.

2. Only head angular orientation will be considered. Translational motion will not be measured or predicted. In flight the dominant motion of a pilot's head is angular.

3. There is no reliable "truth model" [4,9] of head motion. That is, no mathematical model exists that can accurately predict look-angles. This precludes testing a proposed filter design against an accurately *simulated* head motion from such a validated "truth model", and simply differencing estimates and "true" values to assess the errors committed by the filter. Instead, models will be tested against a cross-section of empirical data.

4. Gross tracking motions are made with the head. The eyes are used for fine tracking. This research will not deal with eye motion.

1.2 Background

The Air Force has long been involved in flight simulators. They allow training in situations that are too dangerous to practice in actual flight. They are also cheaper and allow flight maneuvers to be isolated and repeated more quickly. Virtual reality flight simulators are a natural extension of present systems. Their potential advantages include lower cost (and thus more widespread availability than conventional half-dome simulators) and greater flexibility. Presently, their primary drawback is that they are less realistic than the more conventional flight simulators.

The Air Force is heavily involved in trying to improve the realism of virtual environments. The principle motivation for the research is the need for effective, low-cost, safe training aids. As part of this effort, the Air Force is researching ways to reduce or eliminate the effect of lag by predicting look-angles. Overcoming lag will also have impact in areas beyond virtual reality, such as telerobotics, head-slaved weapons and distributed simulation (players in different geographical locations simultaneously participating in a real-time simulation) [2, 25].

Perhaps the most obvious way to reduce lag is to increase the speed of required computations. Hardware improvements alone cannot solve the problem [14]. Sensor response time, transmission time, and computational overhead involved in image management all contribute to lag. As computer speed increases, so will the demand for more realistic graphics and other tasks. Faster computers may reduce lag, but they will not eliminate the problem.

1.3 Summary of Current Knowledge

As previously mentioned, large amounts of research have focused on the lag problem. This summary will cover two categories; causes and effects of lag and techniques for predicting look-angle.

Lag in virtual displays has been blamed for many of the shortcomings of virtual environments. Oman [22] showed that motion sickness and "simulator sickness" both come from sensory conflict: when a subject's expectations for sensory signals do not match actual signals. In 1993 Kozak [13] presented research that showed no learning transfer from a virtual task to a real-world task. Kozak concluded virtual environments must be more realistic for them to be effective training aids. Researchers agree [7,9,13] that lag is a detriment to human performance in a virtual environment.

Any effort to predict look-angles must start with modeling head motion. Immediately, there are significant difficulties. Jagacinski [12] pointed out that when a human is part of a closed loop system, the system can no longer be simply described with a transfer function. McRuer [4,15] proposed a crossover model suggesting the human compensates for other components of a closed-loop system to force the overall system to respond as a first order system. This model assumes the operator of a manual control system will maintain a high open-loop gain so the cross-over frequency remains just below the 180° phase lag frequency and a small phase margin is maintained. Further, the crossover model uses linear describing functions [8] and, therefore, does not account for non-linearities without approximation. This model does not consider operator learning and time varying behavior.

Kleinman [4] proposed an optimal control model for human performance in a closed loop. This model interprets human perception as a noisy measurement, response time as a time delay, and imperfect response as noisy input. It further assumes that the position data presented visually inherently gives an indication of velocity. The basic premise is that the human operator in a tracking task operates as an LQG controller with a particular choice of quadratic weighting matrices and under the constraint of having to incorporate a predictor to compensate for a 0.2 second transport delay between visual input and muscular output. The optimal model requires the inputs, performance criteria, and measurements to be well described and the system being represented must be linearized.

Both the crossover and optimal control models are only accurate under very restrictive conditions. They cannot be used to predict human motion in more general settings than a specific tracking task.

Longinow [15] used a Radial Basis Function network (neural net) to predict head position from 150 msec to 350 msec into the future. He chose the prediction lengths to match expected near-term requirements for predicting look-angles. The best results came from predicting over no more than 200 msec.

Liang [14] developed a Kalman filter estimator to predict head motion. Liang modeled head motion as a first-order Gauss-Markov velocity process. The filter was used to predict the motion of a pendulum. His research showed good results under these highly restrictive test conditions. Friedman [10] used Kalman filtering in a virtual drumset to synchronize the display of an actual drumstick with a virtual drumset. He used a non-adaptive, constant-acceleration model to predict the position of the drumstick. During periods of high acceleration the Kalman filter predictor showed overshoot. Friedman concluded that overshoot was intrinsic to all prediction methods. However, he suggested overshoot could be minimized by using more complex noise and dynamics models. Friedman specifically suggested using MMAE techniques.

O'Connor [21] began the study of MMAE applied to head motion. Russell [24] followed O'Connor's work by designing and creating the structure for applying MMAE to a virtual environment. Russell's emphasis was on the software aspects of the virtual environment and his results on the effectiveness of MMAE for predicting head motion were inconclusive. This research continues his work, concentrating on using MMAE to predict look-angles.

1.4 Objectives

The purpose of this research is to decrease lag effects in virtual displays by predicting angular head motion. Specifically, MMAE techniques will be applied to predict the look-angle 0.1 to 0.2 seconds into the future. This time interval was chosen for two reasons. First, the majority of current research [5] deals with delays in this interval. Second, the virtual environment being used in this research runs at a fixed rate of 10 Hz, thus dictating steps of 0.1 seconds in readily implemented predictors. Several head motion models will be investigated.

First-order Gauss Markov acceleration (FOGMA) model - This model assumes that head motion angular acceleration is well-modeled as a first-order Gauss-Markov process (the output of a first-order lag driven by white Gaussian noise); i.e. that the rate of change in acceleration is a function of the current acceleration and white noise. The FOGMA model is motivated by its successful use in describing airborne targets [6]. Tracking airborne targets is a common task for pilots. Russell's work used FOGMA models [24]. This model has the form:

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \mathbf{a} \\ \dot{\mathbf{a}} &= -\frac{1}{\tau}\mathbf{a} + \mathbf{w} \end{aligned} \tag{1.1}$$

where:

$\mathbf{p}$ = angular position

$\mathbf{v}$ = angular velocity

$\mathbf{a}$ = angular acceleration

$\mathbf{w}$ = zero-mean white Gaussian noise of strength $\mathbf{Q}$

T = time constant of the first-order lag

Notice that as T goes to infinity, the FOGMA model becomes a constant acceleration model (plus white noise for tuning and indicating the model is imperfect). Changing the values of T and **Q** results in very different descriptions of motion.

First order Gauss Markov velocity (FOGMV) model: This model assumes that angular velocity is well-modeled as a first-order Gauss-Markov process; i.e., that the rate of change in velocity is a function of the current velocity and white noise:

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= -\frac{1}{\tau}\mathbf{v} + \mathbf{w} \end{aligned} \tag{1.2}$$

Notice here that as T goes to infinity, this becomes a constant velocity model (plus white noise for tuning and indicating the model is imperfect). This model describes less aggressive head motion. It was used by Liang [14] in the pendulum experiment mentioned earlier. A drawback to this model is that it yields a zero-mean velocity, which may not be very descriptive of head motion.

The final MMAE may use elemental filters from more than one of the types listed above. For example, the FOGMV model may be best for slow motion and the FOGMA best for rapid motion. The MMAE can use these different models in separate elemental filters within its structure to form a single predictor.

1.5 Methodology

Proposed predictors will be tested against measured look-angles. The measured look-angle data is pilot head position as measured by an Ascension Bird [9] magnetic tracker during a variety of fighter missions flown in a visual simulator. The look-angle data was collected at 60 Hz and tracker specifications claim angle information is accurate within 0.1 degrees [8]. At each sample period, the predictor will estimate what the look-angle will be at future sample times. These predictions will be compared to the look-angles actually measured at the future times. These comparisons will show the accuracy of the predictions and will portray problems such as overshoot and oscillation.

Since Liang's filter is the best replicatable predictor available, it will serve as a benchmark. The MMAE must outperform Liang's filter to justify its greater complexity and computational demand. A second benchmark of performance will be the case of no predictor in the loop at all, representative of many current implementations of virtual environment simulators. Thus, MMAE predictor performance will be bracketted by perfect (and unobtainable) prediction and no prediction, with a direct comparison to the best results of a *nonadaptive* predictor (i.e., Liang's).

The MMAE will consist of 3 Kalman filter predictors. The look-angle data from simulator missions is divided into three categories of head motion: slow (staring and fine tracking), moderate (continuous target tracking), and fast (target reacquisition). Each elemental filter will be tuned by adjusting time constants (τ) and $\mathbf{Q}$ values for a specific type of motion; slow, moderate and fast. Once the elemental filters are tuned, they will be combined into an MMAE. The MMAE will be tested against all categories of motion and concatenated segments to display onsets of changing characteristics of head motion.

The MMAE algorithm was first developed by O'Connor [21]. The virtual display was developed by Russell [24] on Silicon Graphics Incorporated (SGI) 4D and Reality Engine workstations using C++ and Performer [26] library graphics support. The virtual display allows researchers to see how different models and tuning affect lag and overshoot. This simplifies the task of identifying potential filters and getting approximate tunings, as well as allowing the assessment of the *visual impact* of ringing and other phenomena (that might not be obvious simply from time histories of errors, or of error statistics).

The visual display is generated in a PT-01 head-mounted display (HMD). The PT-01 uses two 420x230 pixel Color Active Matrix Liquid Displays with a total field of view of approximately 60° [23]. Look-angle is measured by a Polhemus Fastrack magnetic tracker. The tracker has a resolution of 0.1 degrees and a static accuracy of 0.5 degrees [26]. The maximum output rate is 60 samples per second. The Polhemus tracker consists of a magnetic field transmitter and receiver. The receiver is attached to the HMD and senses relative changes in the transmitted magnetic field whenever the HMD moves. This change is translated into a unit vector, in Cartesian coordinates, which points in the direction of HMD orientation. This unit vector is the only measurement available to the MMAE.

1.6 Text Overview

Chapter 2 introduces the theory of the Kalman filter and MMAE. Chapter 3 discusses the dynamics and measurement models in detail and presents the overall tracking algorithms. Chapter 4 presents the results of testing the elemental filters based on the various models, and of MMAE algorithms composed of such elemental filters. Chapter 5 contains conclusions and gives recommendations for further research.

1.7 Summary

This chapter introduced the problem of lag in virtual displays and motivated the use of MMAE to predict head motion. It gave a brief overview of the current research in lag and outlined the direction of this thesis effort.

2. Theory

The purpose of this chapter is to introduce the concepts of Kalman filtering and Multiple Model Adaptive Estimation (MMAE). A general understanding of these concepts is essential to the development of this thesis. Only the motivating features and implementable results of these two estimation techniques are presented here. Full mathematical developments can be found in the references cited [16,17]. The more knowledgeable reader may skip this section.

2.1 The Kalman Filter

The Kalman filter provides the optimal estimate of states of a linear dynamic system when certain conditions are met. The full derivation of the Kalman filter can be found in [16]. This section covers the assumptions and the update and propagation equations of the Kalman filter.

The Kalman filter provides an optimal state estimate for linear systems driven by white, Gaussian noise. The filter uses measurements corrupted by noise, and internal models of the dynamics of the system and the measuring devices, to estimate the system state. Consider a system described by

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{G}(t)\mathbf{w}(t) \qquad (2.1)$$

having available measurements

$$\mathbf{z}(t_i) = \mathbf{H}(t_i)\mathbf{x}(t_i) + \mathbf{v}(t_i) \tag{2.2}$$

where

- $\mathbf{x}$ = state
- $\mathbf{u}$ = input
- $\mathbf{w}$ = zero-mean, white Gaussian noise of strength $\mathbf{Q}$: $E\{\mathbf{w}(t)\mathbf{w}^T(t+\tau)\} = \mathbf{Q}(t)\delta(\tau)$
- $\mathbf{z}$ = noise corrupted measurement
- $\mathbf{v}$ = white Gaussian noise with a covariance of $\mathbf{R}(t_i)$: $E\{\mathbf{v}(t_i)\mathbf{v}^T(t_j)\} = \mathbf{R}(t_i)\,\delta_{ij}$

The equivalent discrete-time system for this system model [17] can be written as

$$\mathbf{x}(t_{i+1}) = \mathbf{\Phi}(t_{i+1},t_i)\mathbf{x}(t_i) + \mathbf{B}_d(t_i)\mathbf{u}(t_i) + \mathbf{G}_d(t_i)\mathbf{w}_d(t_i) \tag{2.3}$$

where

$\mathbf{\Phi}(t_{i+1},t_i)$ = state transition matrix such that

$$\frac{d}{dt}\mathbf{\Phi}(t,t_i) = \mathbf{F}(t)\,\mathbf{\Phi}(t,t_i) \quad \text{and}$$

$$\mathbf{\Phi}(t_i,t_i) = \mathbf{I}$$

$\mathbf{B}_d$ = equivalent discrete-time input matrix associated with $\mathbf{B}$ by

$$\mathbf{B}_d(t_i) = \int_{t_i}^{t_{i+1}} \mathbf{\Phi}(t_{i+1},\tau)\mathbf{B}(\tau)\,d\tau$$

$\mathbf{G}_d$ = equivalent discrete-time noise distribution matrix, assumed to be $\mathbf{I}$ here

$\mathbf{Q}_d$ = equivalent discrete-time covariance of the noise $\mathbf{w}_d$ such that

$$\mathbf{Q}_d(t_i) = \int_{t_i}^{t_{i+1}} \mathbf{\Phi}(t_{i+1},\tau)\mathbf{G}(\tau)\mathbf{Q}(\tau)\mathbf{G}^T(\tau)\mathbf{\Phi}^T(t_{i+1},\tau)\,d\tau$$

For such a system, the state estimate is propagated between measurements by

$$\hat{\mathbf{x}}(t_{i+1}^-) = \mathbf{\Phi}(t_{i+1},t_i)\hat{\mathbf{x}}(t_i^+) + \mathbf{B}_d(t_i)\mathbf{u}(t_i) \tag{2.4}$$

$$\mathbf{P}(t_{i+1}^-) = \mathbf{\Phi}(t_{i+1},t_i)\mathbf{P}(t_i^+)\mathbf{\Phi}^T(t_{i+1},t_i) + \mathbf{G}_d(t_i)\mathbf{Q}_d(t_i)\mathbf{G}_d^T(t_i) \tag{2.5}$$

where

$\hat{\mathbf{x}}$ = state estimate

$\mathbf{P}$ = covariance of $\hat{\mathbf{x}}$

The state estimate is updated with measurements by

$$\mathbf{A}(t_i) = \mathbf{H}(t_i)\mathbf{P}(t_i^-)\mathbf{H}^T(t_i) + \mathbf{R}(t_i) \tag{2.6}$$

$$\mathbf{K}(t_i) = \mathbf{P}(t_i^-)\mathbf{H}^T(t_i)\mathbf{A}^{-1}(t_i) \tag{2.7}$$

$$\hat{\mathbf{x}}(t_i^+) = \hat{\mathbf{x}}(t_i^-) + \mathbf{K}(t_i)\,[\mathbf{z}(t_i) - \mathbf{H}(t_i)\hat{\mathbf{x}}(t_i^-)] \tag{2.8}$$

$$\mathbf{P}(t_i^+) = \mathbf{P}(t_i^-) - \mathbf{K}(t_i)\mathbf{H}(t_i)\mathbf{P}(t_i^-) \tag{2.9}$$

where
H = measurement matrix
R = covariance of the measurement noise
z = measurement

The assumptions of a linear system driven by white, Gaussian noise do not hold for all physical problems. In such cases the output of the Kalman filter isn't optimal and adjustments may be necessary to achieve a sufficiently accurate estimate.

2.2 Multiple Model Adaptive Estimation

The Kalman filter's ability to produce accurate estimates of the true states of a physical system is limited by the extent that its internal dynamics model, measurement model, and assumed noise statistics adequately describe true system characteristics. This implies that the parameters which describe the dynamic system (for example, the coefficients of the system differential equations) must be correctly evaluated and embedded in the filter model. Often, however, some of these parameters are unknown to the designer and/or changing in time. One method of dealing with this situation is through Multiple Model Adaptive Estimation (MMAE) [18].

As in the development presented in [18], consider the vector **a** containing all the uncertain and/or dynamic parameters in a given system model. Note that, for this problem, uncertain parameters are confined to Φ and $\mathbf{Q}_d$ Let this system model be described by the first-order, linear, stochastic differential equation of the form:

$$\mathbf{x}(t_{i+1}) = \mathbf{\Phi}(\mathbf{a})\mathbf{x}(t_i) + \mathbf{B}_d(\mathbf{a})\mathbf{u}(t_i) + \mathbf{G}_d(\mathbf{a})\mathbf{w}_d(t_i) \tag{2.10}$$

with physically realizable, noise-corrupted, sampled-data measurements described by:

$$\mathbf{z}(t_i) = \mathbf{H}(\mathbf{a})\mathbf{x}(t_i) + \mathbf{v}(t_i) \tag{2.11}$$

where:

$\mathbf{x}(t_i)$ = n-dimensional system state vector

$\mathbf{u}(t_i)$ = deterministic control vector

$\mathbf{w}_d(t_i)$ = white gaussian vector dynamics driving noise of covariance $\mathbf{Q}_d(a)$ (assumed to be stationary)

$\mathbf{z}(t_i)$ = m-dimensional measurement vector

$\mathbf{v}(t_i)$ = discrete-time white gaussian measurement noise of covariance $\mathbf{R}(a)$ (assumed to be stationary)

$\mathbf{\Phi}(\mathbf{a})$ = state transition matrix (assumed to be time-invariant)

$\mathbf{B}_d(\mathbf{a})$ = input matrix (time invariant)

$\mathbf{G}_d(\mathbf{a})$ = noise distribution matrix (time invariant)

$\mathbf{H}(\mathbf{a})$ = measurement matrix (time invariant)

Since **a** may assume a continuous range of values over the space of allowable parameters, it is necessary, for tractability, to discretize a into a set of J finite vector values: $\mathbf{a}_1$, $\mathbf{a}_2$, ... $\mathbf{a}_J$. A Multiple Model Adaptive Estimator consists of J independent Kalman filters, in which the j-th filter is based on, and tuned to, the specific parameter set aj. Figure 2.1 shows a MMAE. These J filters form a bank of elemental filters which are processed in parallel. Each elemental filter produces its own estimate of the true state, denoted $\hat{\mathbf{x}}_j(t_i)$,

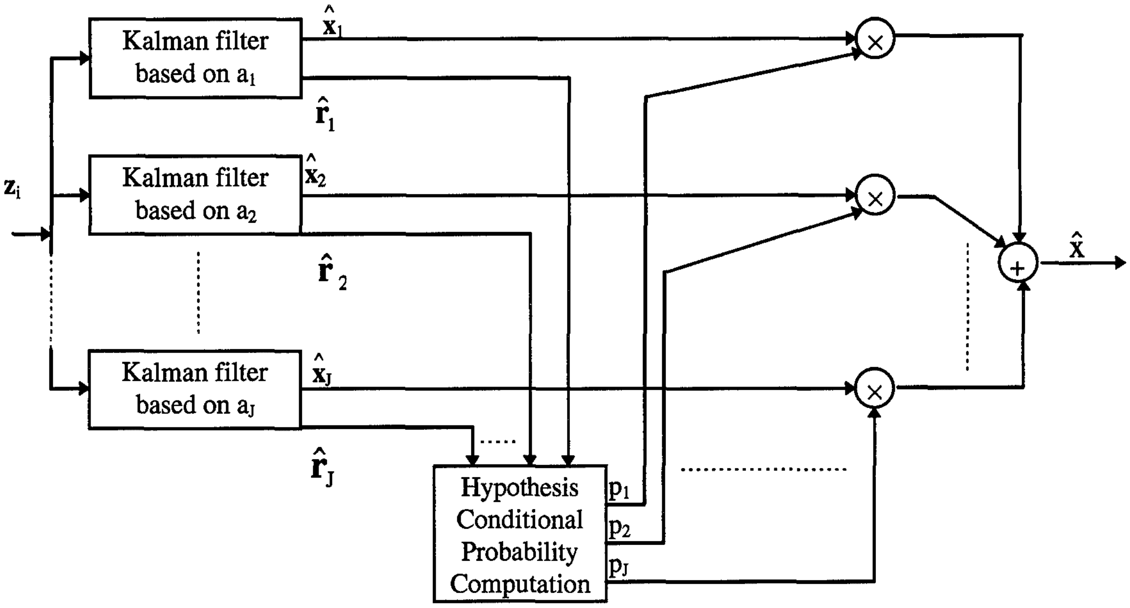

Kalman filter based on a_1
$\hat{x}_1$
$\hat{r}_1$
z_i
Kalman filter based on a_2
$\hat{x}_2$
$\hat{r}_2$
Kalman filter based on a_J
$\hat{x}_J$
$\hat{r}_J$
Hypothesis Conditional Probability Computation
p_1
p_2
p_J
×
+
$\hat{x}$

Figure 2.1. Multiple Model Adaptive Estimation Algorithm

for the j-th hypothesized value of **a**. The residuals of all J elemental filters are then used to calculate the probability that **a** assumes the value $\mathbf{a}_j$ at time t_i, for $j = 1, 2, \ldots J$.

This probability is called the "hypothesis conditional probability" and is denoted as $p_j(t_i)$. In essence, $p_j(t_i)$ represents the probability of the j-th filter's system model being correct at time t_i. Its computation will be shown subsequently.

The overall state estimation of MMAE, $\hat{\mathbf{x}}_{mmae}(t_i)$, is the probabilistically weighted average of the estimates of the elemental filters; that is [18]:

$$\hat{\mathbf{x}}_{mmae}(t_i) = \sum_{j=1}^{J} p_j(t_i)\hat{\mathbf{x}}_j(t_i) \tag{2.12}$$

Equation (2.12) is valid for both t_i^- (just before a measurement update) and t_i^+ (just after a measurement update). This type of multiple model filtering algorithm is referred to as Bayesian MMAE. Figure 2.1 illustrates the structure of the Bayesian MMAE.

The hypothesis conditional probabilities $p_j(t_i)$, $j = 1, 2, \ldots, J$, are calculated at each sample time t_i, by the recursive equation [18]:

$$p_k(t_i) = \frac{f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i \mid \mathbf{a}_k, \mathbf{Z}_{i-1}) \cdot p_k(t_{i-1})}{\sum_{j=1}^{J} f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i \mid \mathbf{a}_j, \mathbf{Z}_{i-1}) \cdot p_j(t_{i-1})} \qquad (2.13)$$

where

$$f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i \mid \mathbf{a}_k, \mathbf{Z}_{i-1}) = \frac{1}{(2\pi)^{m/2}\,|\mathbf{A}_k(t_i)|^{1/2}} \exp\{\cdot\}$$

$$\{\cdot\} = \{-\tfrac{1}{2}\mathbf{r}_k^{T}(t_i)\mathbf{A}_k^{-1}(t_i)\mathbf{r}_k(t_i)\}$$

$$\mathbf{r}_k(t_i) = \mathbf{z}(t_i) - \mathbf{H}_k(t_i)\hat{\mathbf{x}}_k(t_i^{-})$$

$$\mathbf{A}_k(t_i) = \mathbf{H}_k(t_i)\mathbf{P}_k(t_i^{-})\,\mathbf{H}_k^{T}(t_i) + \mathbf{R}_k(t_i)$$

$\mathbf{P}_k(t_i^{-})$ = k-th filter's computed state error covariance before the measurement update at time t_i is incorporated.

Note that the denominator of Equation (2.13) is the sum of the numerator terms for $j = 1, 2, \ldots, J$. This ensures that the sum of $p_j(t_i)$, $j = 1, 2, \ldots, J$, equals one.

Also note that the hypothesis conditional probabilities at time t_i are functions of the hypothesis conditional probabilities at time t_{i-1}. Due to the recursive nature of the calculations, it is essential that an artificial lower bound be established for $p_j(t_i)$ [18]. Without this lower bound, the hypothesis conditional probability of a filter with a totally invalid parameter set, $\mathbf{a}_q$, at some point in time may go to zero and remain zero for all time: once $p_q(t_{i-1})$ reaches zero, $p_q(t_i)$ and all subsequent p_q would be zero. Thus the q-th filter is essentially removed from the bank. Should the actual system change its characteristics so that the true parameter set $\mathbf{a}$ matches $\mathbf{a}_q$ at some future time, $p_q(t_i)$ would remain at zero and the MMAE would produce undesirable results.

It is also possible for a single filter in an MMAE to diverge, since each filter is tuned for only a specific part of the system parameter space. Since the probability weighting on each filter has an artificial lower bound, a divergent filter could drive the entire MMAE divergent. Further, when the system actually moves into the portion of the parameter space for which the now-divergent filter was designed, the filter doesn't know the correct state and can't make accurate predictions. To avoid this problem, each filter is given a divergence threshold. If the residuals of a filter exceed the threshold chosen by the designer, the filter is declared divergent. The MMAE forms an estimate of the future state based on the nondivergent filters, and it assigns those state values to the divergent filter to "restart" it. The filter is then assigned a probability weighting equal to its lower bound and returned to the normal MMAE process.

The residual of the j-th filter plays a major role in determining $p_j(t_i)$. As evident from Equation (2.13), the filter with the consistently smallest value of $\mathbf{r}_j^T(t_i)\mathbf{A}_j^{-1}(t_i)\,\mathbf{r}_j(t_i)$ assumes the largest conditional hypothesis probability. Thus the hypothesis probability algorithm is consistent with the heuristic intuition that the residuals of a well-matched filter should be smaller (relative to the filter's internally computed residual covariance, $\mathbf{A}_j$) than the residuals of a miss-matched filter.

The conditional covariance matrix, $\mathbf{P}_{mmae}(t_i^+)$, can be computed from [18]:

$$\mathbf{P}_{mmae}(t_i^+) = \sum_{j=1}^{J} p_j(t_i)\left[\mathbf{P}_j(t_i^+) + \mathbf{y}_j(t_i^+)\mathbf{y}_j^T(t_i^+)\right] \tag{2.14}$$

where

$$\mathbf{y}_j(t_i^+) = \hat{\mathbf{x}}_j(t_i^+) - \hat{\mathbf{x}}_{mmae}(t_i^+)$$

$p_j(t_i)$ = j-th filter's conditional hypothesis probability

$\mathbf{P}_j(t_i^+)$ = j-th filter's state error covariance matrix after the measurement is incorporated

In linear systems, the $\mathbf{P}_j(t_i^+)$ values for each of the elemental filters can be calculated a priori, but $\mathbf{P}_{mmae}(t_i^+)$ cannot. This is because actual measurements are required to calculate $p_j(t_i)$, $\hat{\mathbf{x}}_j(t_i^+)$, and $\hat{\mathbf{x}}_{mmae}(t_i^+)$. Note, however, that Equation (2.14) need not be computed for online MMAE.

Finally, it should be noted that MMAE performance is generally degraded when too much pseudonoise is added to the elemental filters for tuning purposes. This is due to the fact that "correct" $p_j(t_i)$ values are assigned when the residuals of filters based on correct models look small relative to the filter-computed residual covariance, while the residuals of filters based on "incorrect" models look large. The addition of pseudonoise tends to blur the defining characteristics of the filter residuals, and distinguishing well-matched filters from miss-matched filters becomes increasingly difficult.

2.3 Summary

This chapter introduced the theory of the Kalman filter and the MMAE. The Kalman filter provides optimal estimates when the system is linear, driven by white, Gaussian noise and can be modeled exactly (with no uncertain parameters within the model structure). The MMAE uses a bank of Kalman filters to overcome the inaccuracies of presumed parameter values within models, by blending (through a probability-weighted average) the estimates of Kalman filters using different hypothesized parameter values.

3. Filter Models

3.1 Introduction

This chapter presents the dynamics models of the elemental Kalman filters used in the MMAE. Generally, for all the models used in this research the dynamics are expressed as

$$\dot{\mathbf{x}}(t) = \mathbf{F}\mathbf{x}(t) + \mathbf{G}\mathbf{w}(t) \tag{3.1}$$

The time notation is removed from **F** and **G** because they are constants; **w**(t) is stationary and thus its strength matrix, **Q**, is also devoid of a time argument.

3.2 First-Order Gauss-Markov Acceleration (FOGMA) Model

Using the Kalman filter theory developed in Chapter 2, the FOGMA model was derived by Millner [20], and adapted by Russell [24]. The FOGMA model has been used successfully to track airborne targets [19]. Since tracking airborne targets is a common pilot task, it is reasonable to expect a FOGMA model to predict pilot head motion accurately. Russell's results are given here.

The dynamics are described in terms of postion, **p**, velocity, **v**, and acceleration, **a**, by

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \mathbf{a} \\ \dot{\mathbf{a}} &= -\frac{1}{\tau}\mathbf{a} + \mathbf{w} \end{aligned} \tag{3.2}$$

Here τ is the first-order lag time coefficient, and the correlation time of head angular acceleration. This is of the form of Equation (3.1), with the appropriate vectors and matrices defined as

$$\mathbf{x} = \begin{bmatrix} p_x \\ p_y \\ p_z \\ v_x \\ v_y \\ v_z \\ a_x \\ a_y \\ a_z \end{bmatrix} \quad \mathbf{F} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1/\tau & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1/\tau & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1/\tau \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{Q} = \begin{bmatrix} \frac{2\sigma^2}{\tau} & 0 & 0 \\ 0 & \frac{2\sigma^2}{\tau} & 0 \\ 0 & 0 & \frac{2\sigma^2}{\tau} \end{bmatrix} \tag{3.3}$$

where σ^2 is the mean squared value of the acceleration vector components a_x, a_y, and a_z. Converting **Q** into equivalent discrete-time [17] $\mathbf{Q}_d$ yields

$$\mathbf{Q}_d = \begin{bmatrix} q_{11} & 0 & 0 & q_{14} & 0 & 0 & q_{17} & 0 & 0 \\ 0 & q_{11} & 0 & 0 & q_{14} & 0 & 0 & q_{17} & 0 \\ 0 & 0 & q_{11} & 0 & 0 & q_{14} & 0 & 0 & q_{17} \\ q_{14} & 0 & 0 & q_{44} & 0 & 0 & q_{47} & 0 & 0 \\ 0 & q_{14} & 0 & 0 & q_{44} & 0 & 0 & q_{47} & 0 \\ 0 & 0 & q_{14} & 0 & 0 & q_{44} & 0 & 0 & q_{47} \\ q_{17} & 0 & 0 & q_{47} & 0 & 0 & q_{77} & 0 & 0 \\ 0 & q_{17} & 0 & 0 & q_{47} & 0 & 0 & q_{77} & 0 \\ 0 & 0 & q_{17} & 0 & 0 & q_{47} & 0 & 0 & q_{77} \end{bmatrix} \tag{3.4}$$

where

$$q_{11} = \sigma^2\{2/3\ \tau^2\Delta t^3 - 2\tau^2\Delta t^2 - 4[\tau^2\Delta t \exp(-\Delta t/\tau)] + 2\tau^3\Delta t - \tau^4 \exp(-2\Delta t/\tau) + \tau^4\}$$

$$q_{14} = \sigma^2\{\tau\Delta t^2 + 2[\tau^2\Delta t \exp(-\Delta t/\tau)] + \tau^3 - 2\tau^3 \exp(-\Delta t/\tau) - 2\tau^2\Delta t + \tau^3 \exp(-2\Delta t/\tau)\}$$

$$q_{17} = \sigma^2\{-2\tau\Delta t \exp(-\Delta t/\tau) + \tau^2 - \tau^2 \exp(-2\Delta t/\tau)\}$$

$$q_{44} = \sigma^2\{2\tau\Delta t - 3\tau^2 + 4\tau^2 \exp(-\Delta t/\tau) - \tau^2 \exp(-2\Delta t/\tau)\}$$

$$q_{47} = \sigma^2\{\tau - 2\tau \exp(-\Delta t/\tau) + \tau \exp(-2\Delta t/\tau)\}$$

$$q_{77} - \sigma^2\{1 - \exp(-2\Delta t/\tau)\}$$

σ^2 = variance (and mean squared value) of acceleration

Δt = sample period

τ = correlation time of angular head acceleration.

The state transition matrix Φ is:

$$\Phi = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 & \phi_{16} & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & \phi_{16} & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & \phi_{16} \\ 0 & 0 & 0 & 1 & 0 & 0 & \phi_{46} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \phi_{46} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \phi_{46} \\ 0 & 0 & 0 & 0 & 0 & 0 & \phi_{66} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \phi_{66} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \phi_{66} \end{bmatrix} \tag{3.5}$$

where

$$\phi_{16} = \tau[\Delta t - \tau(1 - \exp(-\Delta t/\tau))]$$

$$\phi_{46} = \tau[1 - \exp(-\Delta t/\tau)]$$

$$\phi_{66} = \exp(-\Delta t/\tau)$$

Notice that, as τ approaches infinity (1/τ approaches zero), the FOGMA model converges to a constant acceleration model, since $\dot{\mathbf{a}} = -\frac{1}{\tau}\mathbf{a} + \mathbf{w}$ becomes $\dot{\mathbf{a}} = \mathbf{0} + \mathbf{w}$ as τ is allowed to become very large. The dynamics model is representative of constant acceleration, with pseudonoise **w** added for tuning and "admitting" that a constant acceleration model is not really a perfect model of the observed head motion. At the other extreme, as τ approaches zero, even a small acceleration results in a large jerk in the opposite direction, and the acceleration becomes white noise in the limit as τ approaches 0. Thus, the model converges to a constant velocity model, with pseudonoise added for filter tuning; i.e., $\dot{\mathbf{v}} = \mathbf{0} + \mathbf{a}$, with **a** being zero-mean white Gaussian noise.

The values of σ and τ for each filter can be adjusted by the designer to give the best overall elemental filter performance for a particular type of head motion. The objective behind choosing τ values is to discretize the parameter space so that an acceptably accurate model is available for all head motions used by a pilot in flight simulation while requiring only a small number of discrete parameter values. The σ values provide a measure of rms acceleration (or velocity) of the head, and also of the inadequacies of each model. The value of σ in each filter is most important in relation to the σ values of the other filters, and relative to the measurement noise covariance, **R**. In this research σ and τ were adjusted for each filter by trial-and-error. One filter was tuned for benign motion - staring and small, slow movements. The second filter was tuned for moderate motion - tracking a slowly manuevering target. The final filter was tuned for aggressive motion - reacquisition of a manuevering target. After each filter was tuned, the three filters were put together into an MMAE algorithm. Further tuning was then required to ensure the probability weight shifted to the correct elemental filter at the appropriate time, i.e., as simulated head motion transitioned from benign to moderate to aggressive motion, and back again. Actual values for σ and τ are given in Chapter 4.

3.3 First-Order Gauss-Markov Velocity (FOGMV) Model

As mentioned in Chapter 1, Liang used a single FOGMV model with some success to predict head motion. Liang's design provides a benchmark to measure the effectiveness of new designs. Therefore, it makes sense to experiment with the same model type he used. There is no substantive data from actual head motion studies to support or discount any specific type of model.

The dynamics of the FOGMV model are:

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= -\frac{1}{\tau}\mathbf{v} + \mathbf{w} \end{aligned} \tag{3.6}$$

Here, τ becomes the correlation time of the head angular velocity rather than angular accleration as in the FOGMA model. Since acceleration is not considered in this model, the state vector, **x**, could be reduced to six elements. However, to minimize the software changes and to yield filters with the same state dimensionality (even if some components are dummy variables only), the **x** vector remains nine elements and the acceleration terms are simply zeroed out:

$$\mathbf{x} = \begin{bmatrix} p_x \\ p_y \\ p_z \\ v_x \\ v_y \\ v_z \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.7}$$

$$\mathbf{F}=\begin{bmatrix}0&0&0&1&0&0&0&0&0\\0&0&0&0&1&0&0&0&0\\0&0&0&0&0&1&0&0&0\\0&0&0&-\frac{1}{\tau}&0&0&0&0&0\\0&0&0&0&-\frac{1}{\tau}&0&0&0&0\\0&0&0&0&0&-\frac{1}{\tau}&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\end{bmatrix} \tag{3.8}$$

For the corresponding equivalent discrete-time model,

$$\Phi=\begin{bmatrix}1&0&0&\phi_{14}&0&0&0&0&0\\0&1&0&0&\phi_{14}&0&0&0&0\\0&0&1&0&0&\phi_{14}&0&0&0\\0&0&0&\phi_{44}&0&0&0&0&0\\0&0&0&0&\phi_{44}&0&0&0&0\\0&0&0&0&0&\phi_{44}&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\end{bmatrix} \tag{3.9}$$

where

$$\phi_{14} = \tau[1-\exp(-\Delta t/\tau)]$$

$$\phi_{44} = \exp(-\Delta t/\tau)$$

$$\mathbf{Q}_d = \begin{bmatrix} q_{11} & 0 & 0 & q_{14} & 0 & 0 & 0 & 0 & 0 \\ 0 & q_{11} & 0 & 0 & q_{14} & 0 & 0 & 0 & 0 \\ 0 & 0 & q_{11} & 0 & 0 & q_{14} & 0 & 0 & 0 \\ q_{14} & 0 & 0 & q_{44} & 0 & 0 & 0 & 0 & 0 \\ 0 & q_{14} & 0 & 0 & q_{44} & 0 & 0 & 0 & 0 \\ 0 & 0 & q_{14} & 0 & 0 & q_{44} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.10}$$

where

$$q_{11} = \sigma^2\{2\tau\Delta t - 3\tau^2 + 4\tau^2 \exp(-\Delta t/\tau) - \tau^2 \exp(-2\Delta t/\tau)\}$$

$$q_{14} = \sigma^2\{\tau - 2\tau \exp(-\Delta t/\tau) + \tau \exp(-2\Delta t/\tau)\}$$

$$q_{44} = \sigma^2\{1 - \exp(-2\Delta t/\tau)\}$$

and σ^2 is now the variance (and mean squared value) of head angular velocity (versus angular acceleration in the FOGMA model).

Notice that, as τ approaches infinity ($1/\tau$ approaches zero), the FOGMV model converges to a constant velocity model, since $\dot{\mathbf{v}} = -\frac{1}{\tau}\mathbf{v} + \mathbf{w}$ becomes $\dot{\mathbf{v}} = \mathbf{0} + \mathbf{w}$ as τ is allowed to become very large. The dynamics model is representative of constant velocity, with pseudonoise **w** added for tuning and "admitting" that a constant velocity model is not really a perfect model of the observed head motion. At the other extreme, as τ approaches zero, even a small velocity results in a large acceleration in the opposite direction,and the velocity process converges to white noise as τ approaches 0; thus the model converges to a constant position filter with pseudonoise: $\dot{\mathbf{p}} = \mathbf{0} + \mathbf{v}$ with **v** being a zero-mean white noise. It is reasonable to assume the time constant τ for the FOGMV model will be longer than for the corresponding FOGMA model, since velocity persists longer than acceleration.

Liang calculated optimal values for σ and τ by using a hill climbing algorithm to minimize the cost function

$$L_k(\sigma,\tau) = \sum_{j=1}^{n-k} \sum_{m=1}^{3} | p_j^{(m)} - q_{j+k}^{(m)} | \qquad (3.11)$$

where p is the predicted value of the unit vector (p_x , p_y , p_z) which points in the same direction as the position sensor, q is the measured value of that same unit vector (q_x , q_y , q_z), k is the number of time steps into the future for which the prediction is being made, $p^{(m)}$ is the m th component of the unit vector, and j is the current time step. Liang's values for a 150 ms prediction period were $\sigma^2 = 0.2$ (radians/sec)^2 and $\tau = 0.115$ seconds. These values were used as a starting point for design of the moderate motion filter. The benign and reacquistion filters were developed to bracket the moderate filter. Specific numerical values are given in Chapter 4.

3.4 Constant Position Model

During very benign motion, such as during instrument conditions when the pilot tries to keep his head steady and change views by only moving his eyes, a constant position model may most accurately describe head motion.

A constant-position filter is based on the dynamics model

$$\dot{\mathbf{p}} = \mathbf{0} + \mathbf{w} \qquad (3.12)$$

If **Q** is finite, and the measurement noise covariance **R** = **0**, the output of the filter will be the same as the raw output of the position measuring device. If **Q** and **R** are both non-zero, the filter will smooth the output of the measuring device at the cost of adding to lag by the filter's computation time. The design philosophy for this filter is to use a very small **R** value so the filter will closely track the measurements and adjust **Q** to draw MMAE probability weighting when there is no motion and quickly give up weighting when motion begins. A single constant-position filter will be used later in an MMAE with FOGMA and/or FOGMV elemental filters. Actual tuning values are given in Chapter 4. As in the previous section, the state dimension is kept at nine, but only the first three states are actually estimated:

$$\mathbf{x} = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.13}$$

$$\mathbf{F} = \mathbf{0} \tag{3.14}$$

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.15}$$

The dynamics noise strength matrix is described as:

$$\mathbf{Q} = \begin{bmatrix} Q_{11} & 0 & 0 \\ 0 & Q_{11} & 0 \\ 0 & 0 & Q_{11} \end{bmatrix} \tag{3.16}$$

or, in the equivalent discrete time form, the defining matrices are:

$$\mathbf{\Phi} = \mathbf{I} \tag{3.17}$$

$$\mathbf{Q}_d = \begin{bmatrix} Q_{11}\,\Delta t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & Q_{11}\,\Delta t & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & Q_{11}\,\Delta t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.18}$$

3.5 Measurement Model

The Polhemus head-tracker describes the subject's head orientation with a unit vector pointing in the same direction as the subject's head. All calculations in the MMAE algorithm are done on in Cartesian coordinates. Once a prediction is made, the coordinates are transformed to heading and pitch and output to the rendering software. Each of the above models uses the same sampled-data measurement model:

$$\mathbf{z}(t_i) = \mathbf{H}\mathbf{x}(t_i) + \mathbf{v}(t_i) \tag{3.19}$$

where

$\mathbf{z} = (z_x , z_y , z_z)$ the unit vector which points in the same direction as the subject's head.

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.20}$$

$\mathbf{v}$ = zero-mean, white Gaussian measurement noise with covariance

$$\mathbf{R} = \begin{bmatrix} R_{11} & 0 & 0 \\ 0 & R_{11} & 0 \\ 0 & 0 & R_{11} \end{bmatrix}.$$

3.6 Summary

This chapter presented the dynamics and measurement models which will be used in elemental Kalman filters of the MMAE. The MMAE may consist of three FOGMA models, three FOGMV models or a combination of FOGMA, FOGMV and constant position models.

4. Performance Analysis

The results obtained from an MMAE using three elemental filters (a constant position, a FOGMV and a FOGMA filter) clearly outperformed the single Kalman filter predictor Liang proposed (12). This chapter will show the advantages and disadvantages of several possible designs and cover the development of the final MMAE design.

Researchers at Armstrong Laboratories (8) collected head motion data, sampled at 60 Hz, from simulator missions flown by an experienced fighter pilot. The head movements were categorized into seven groups (circular, hourglass, chaos, vertical, horizonal, staring, and figure-eight), each group containing five sets of data. To form a single, representative data set, displaying benign, moderate and aggressive motion, data sets from the hourglass and chaos categories were concantenated. Unless otherwise mentioned, the test runs in this chapter use this concantenated path, sampled at 10 Hz. Appendix A shows the final design run against several other sets of motion data and against the original concatenated path sampled at 20 Hz. The heading position, velocity and acceleration for this original path are shown in Figure 4.1. Note that position is scaled by a factor of 10 to allow it to be put on the same graph with velocity and data. The vertical scale is in degrees * 10 for position, degrees/second for velocity and degrees/second^2 for acceleration. Further, note the difference in the vertical scales from the top graph to the bottom graph. The first eight seconds are benign motion. Eight to 14 seconds is moderate motion and 19 to 25 seconds is reacquisition (or aggressive) motion.

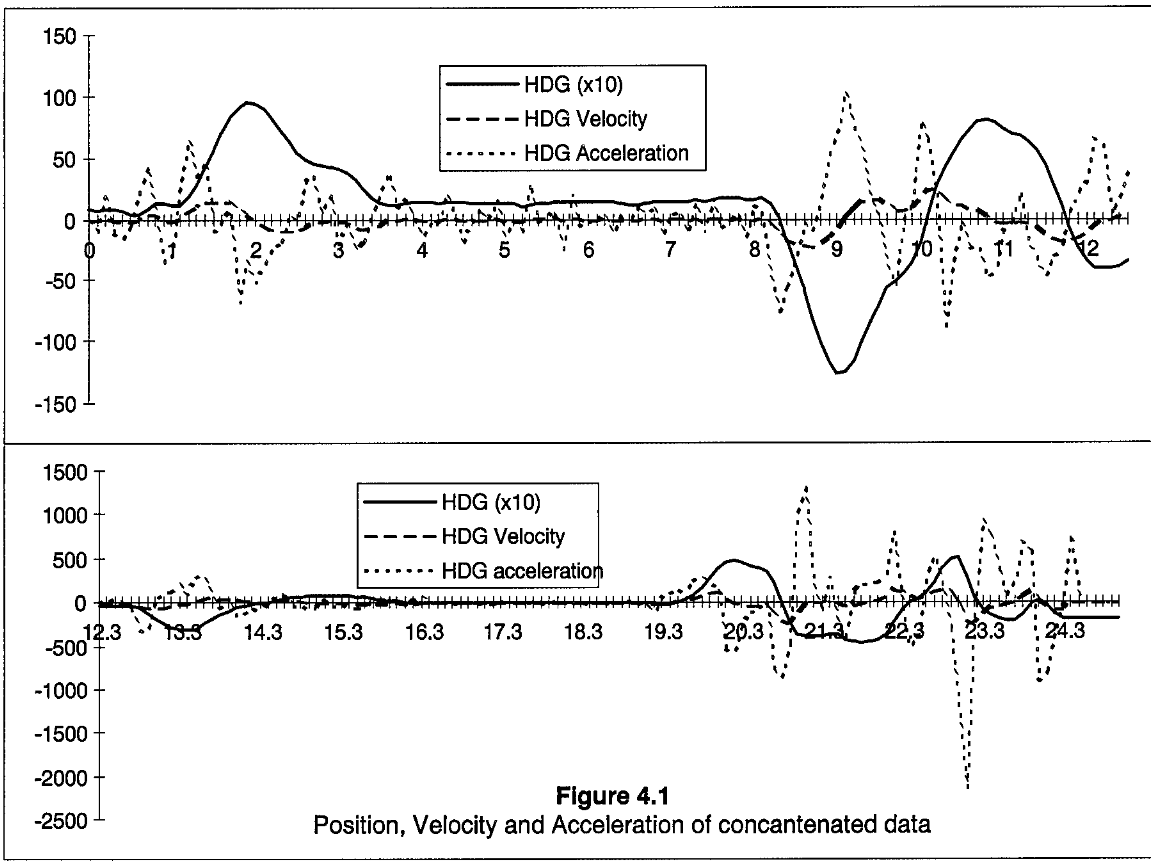


Figure 4.1
Position, Velocity and Acceleration of concantenated data

Two benchmarks were used to measure relative performance of designs. First, MMAE predictor performance was compared to a system with no predictor. The no-predictor system has a constant one-sample-period lag and no overshoot or ringing. Second, the MMAE predictor was compared to Liang's nonadaptive FOGMV-model predictor [12]. No single elemental filter tested in this research could significantly outperform Liang's design. That is, any predictor which gave significantly less lag also produced significantly more overshoot and/or ringing. Since the MMAE structure is more complex than a nonadaptive filter, the MMAE must outperform it to make the extra computational effort worthwhile. For this reason, the actual look-angle path (perfectly known and with no lag, and thus the unachievable result of a "perfect" predictor), and the two benchmarks for comparison given by the output of both the no-predictor system and

Liang's nonadaptive predictor, are plotted with each proposed design. It is desired that the MMAE performance fall between the "perfect prediction" case and the two benchmarks, and as close to the "perfect prediction" case as possible.

4.1 Initial Design

Jim Russell [19] developed the virtual environment and MMAE software used in this research. Figure 4.2 shows the performance of Russell's final MMAE design against the standard concatenated path of head motion. Note that, in this and all other performance plots the uppermost plot, corresponds to benign head motion, the middle plot to moderate head motion, and the bottom plot to aggressive target-reacquisition motion of a head (as seen in Figure 4.1). Since his focus was on creating the virtual environment, his results basically show an untuned MMAE. Russell's design uses three FOGMA filters, each with a different time constant (τ) and noise rms value (σ). This design was very preliminary, and it is outperformed by Liang's nonadaptive filter, as seen in Figure 4.2. It is important to note the characteristics of Liang's filter. Compared to no predictor, it significantly reduces lag. Without a predictor, the system has a constant lag of 0.1 seconds. Liang's filter cuts the lag by a variable amount of time, at the expense of overshoots whenever the path changes velocity quickly. The trade-off between reducing lag and increasing overshoot is the dominant factor in designing a predictor.

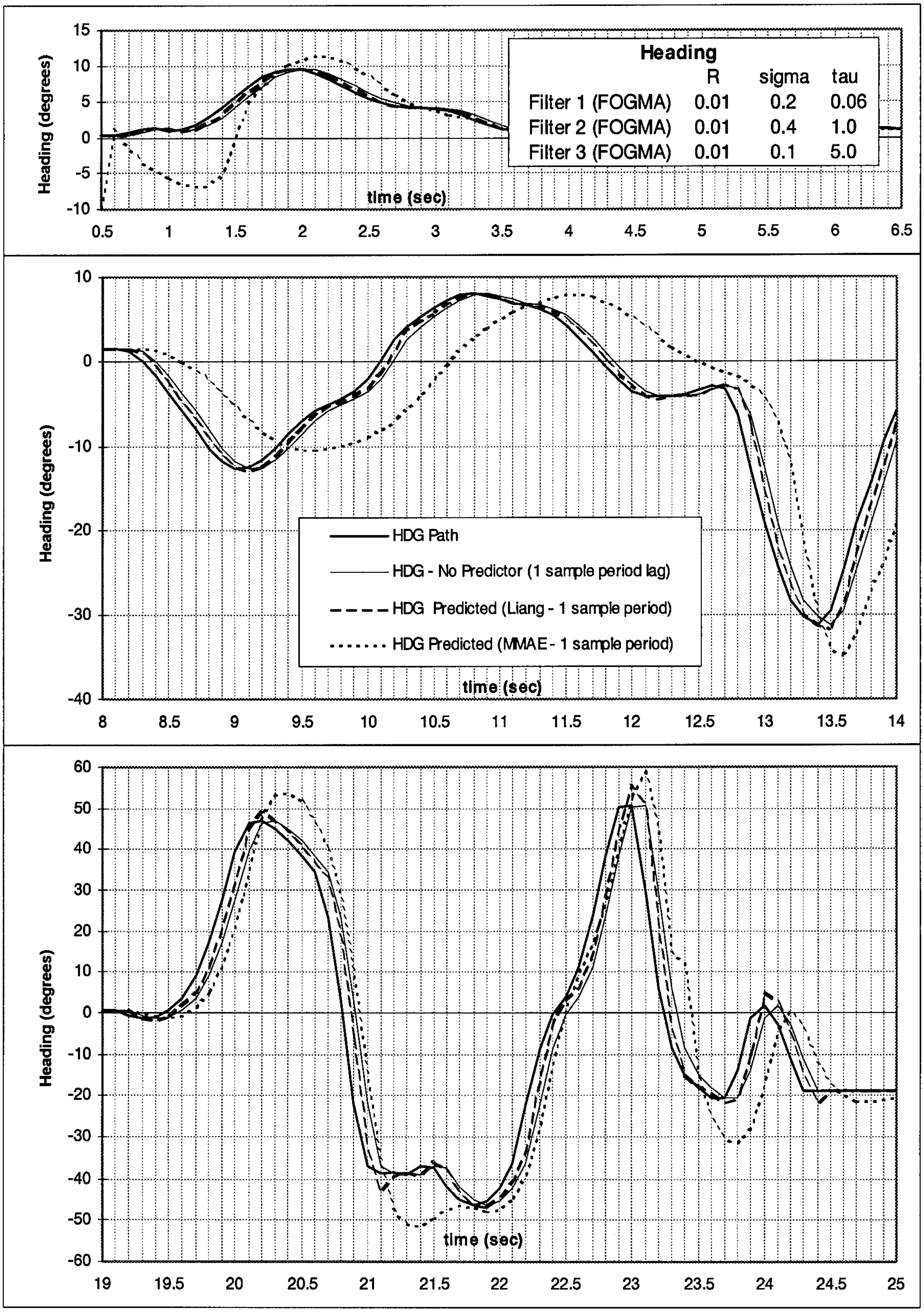


Figure 4.2

Russell's preliminary design.

4.2 Design Approach

In this research, MMAEs were designed by tuning elemental filters for specific types of motion (benign, moderate and reacquisition). The elemental filters were then combined to form MMAEs. The individual filters were then further tuned to get the best overall performance.

Although Liang empirically determined the measurement noise of his system to have variance R = 0.001 degrees^2, in this research predictors performed better with R = 0.0001 degrees^2. This small value for R brought the updated look-angle estimate very close to the measured value. This is notable because the measurements were thus declared to be far more accurate than any sensor actually tested. Such a small value of R raises the risk of introducing jitter from noisy measurements. The empirically observed variance of the measurement noise varied greatly depending on the position of the transmitter and receiver relative to each other and other electronic and metal objects in the laboratory. In the final configuration, jitter was negligible, even with R= 0.0001 degrees^2. A different electromagnetic environment, or different look-angle measuring device, may demand a different value for R.

The initial transient on all the filters is a result of the initial conditions, $\hat{\mathbf{x}}_0$ and $\mathbf{P}_0$. The algorithm requires initial conditions, but quickly adjusts the state estimate and covariance in response to measurements. The initial conditions were arbitrarily set, resulting in a transient as the algorithm converged to the correct values. Since this initial transient was not of substantial practical interest, all of the performance plots in this chapter start at 0.5 seconds into the simulation, after the initial transient effects have died out.

4.3 Elemental FOGMA Filters

Individual FOGMA filters were tested over a range of time constants, τ, mean squared accelerations, σ^2, and measurement noise variances, R. Figure 4.3 shows a single FOGMA filter with a $\tau = 10$ seconds (the other parameter values are listed in the figure). Recall from Chapter 3 that, for large values of τ (such as $\tau = 10$ seconds), the FOGMA model approaches a constant acceleration model. Figure 4.4 shows a single FOGMA filter with $\tau = 0.01$ seconds. This filter approximates a constant velocity model. Comparing the two FOGMA filters shows that the nearly-constant-acceleration model displays less lag (see especially t = 2.0 through 2.5 seconds, t = 8.3 through 9.0 seconds) and generally larger overshoots (particularly at t = 9.1, 13.2, 24.7 seconds) when compared to the nearly-constant-velocity model. Since the nearly-constant-acceleration model expects any acceleration to persist, it tends to yield position predictions that oscillate (or ring) about the true measurement (t = 3.5 through 6.5 seconds, and t = 10.3 through 11.3 seconds) during benign and moderate motion when acceleration actually decays rapidly. Figures 4.5 and 4.6 show models using the same σ value as the previous two models and τ values between the nearly-constant-velocity and nearly-constant-acceleration values. As expected, as τ increases, the predictions tend to be more aggressive (make larger changes from one time sample to the next). The filters with the larger τ also have more ringing and

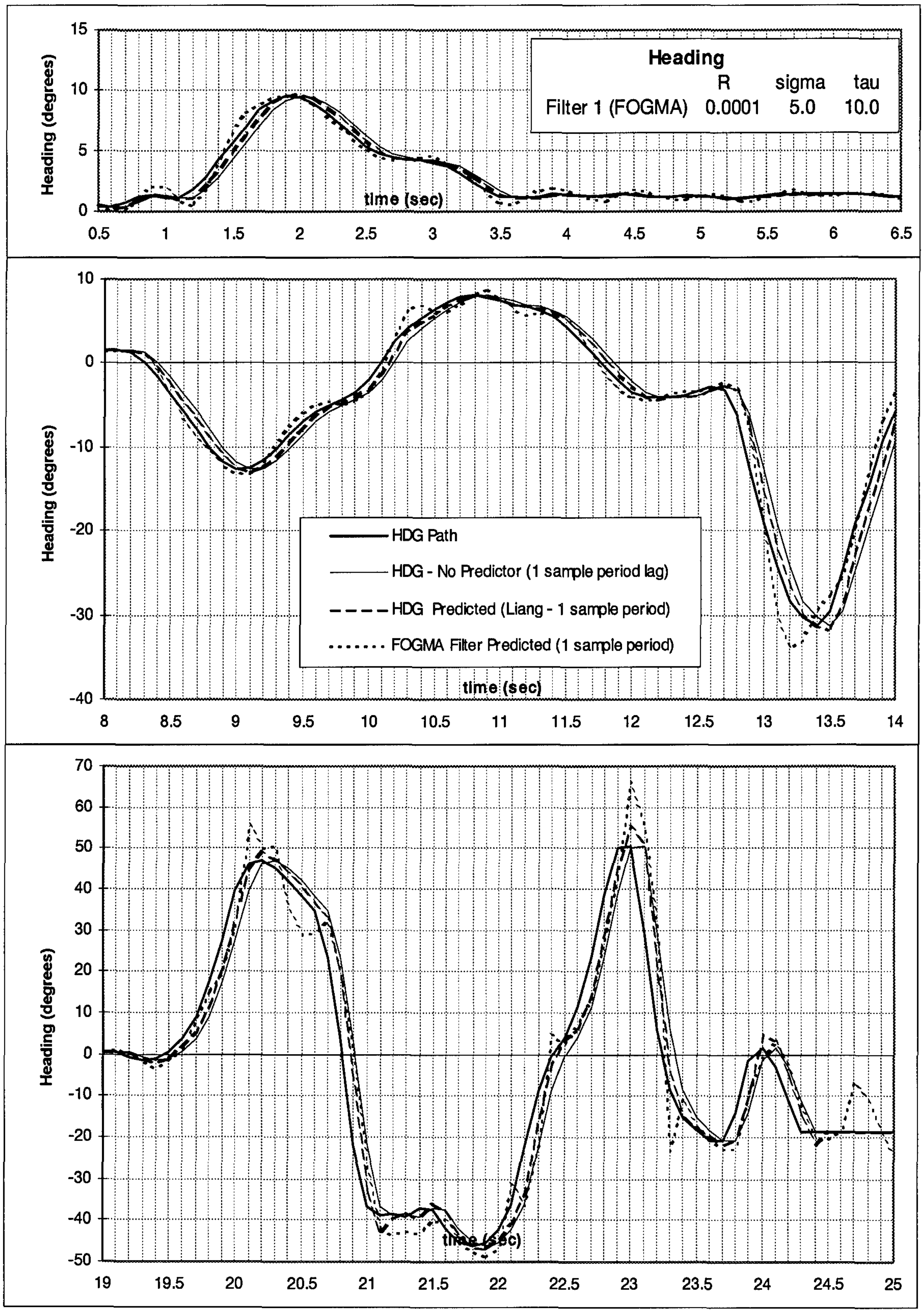


Figure 4.3
Nearly-Constant-Acceleration FOGMA.

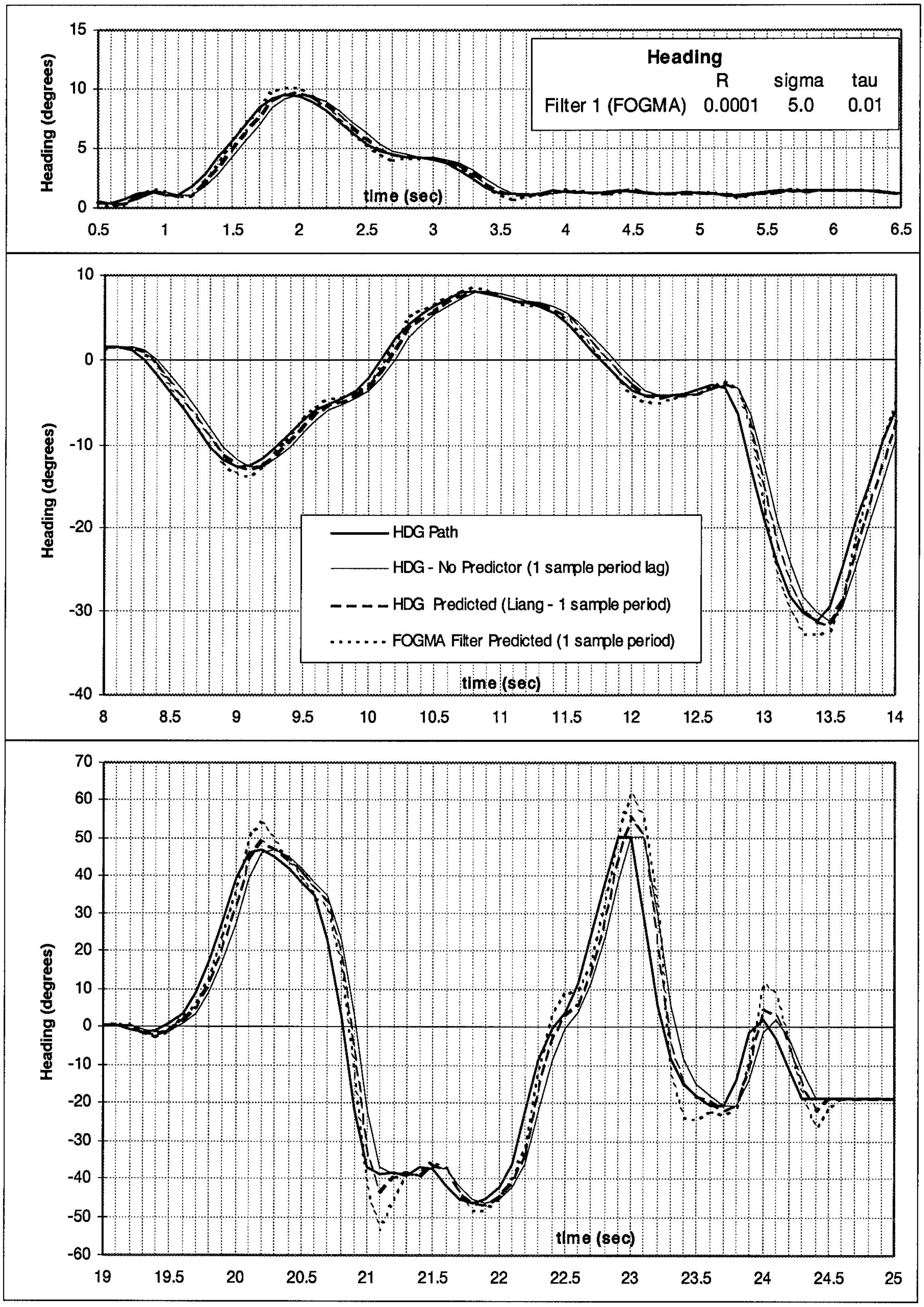


Figure 4.4
Nearly-Constant-Velocity FOGMA.

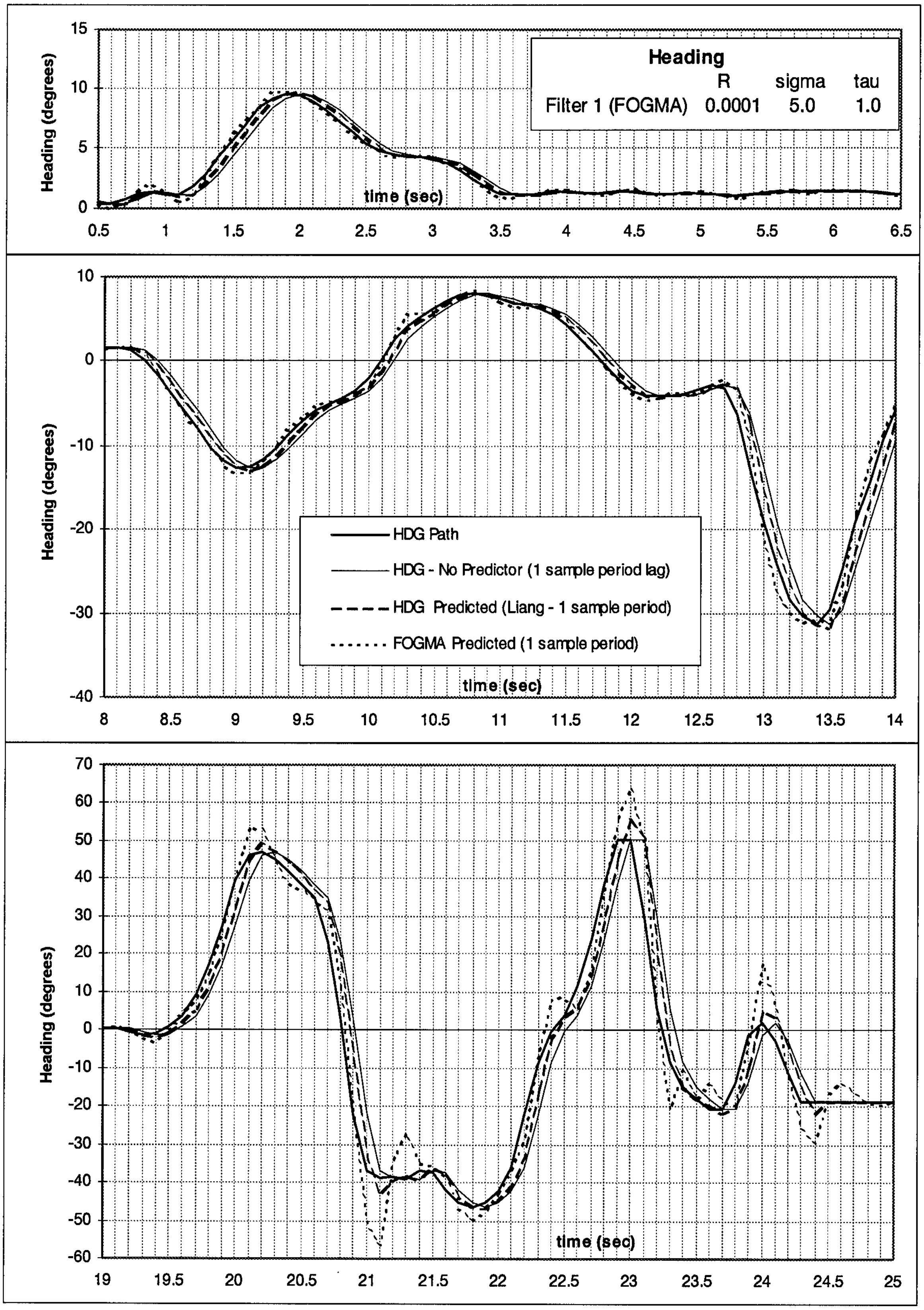


Figure 4.5
FOGMA with a mid-range time constant, τ.

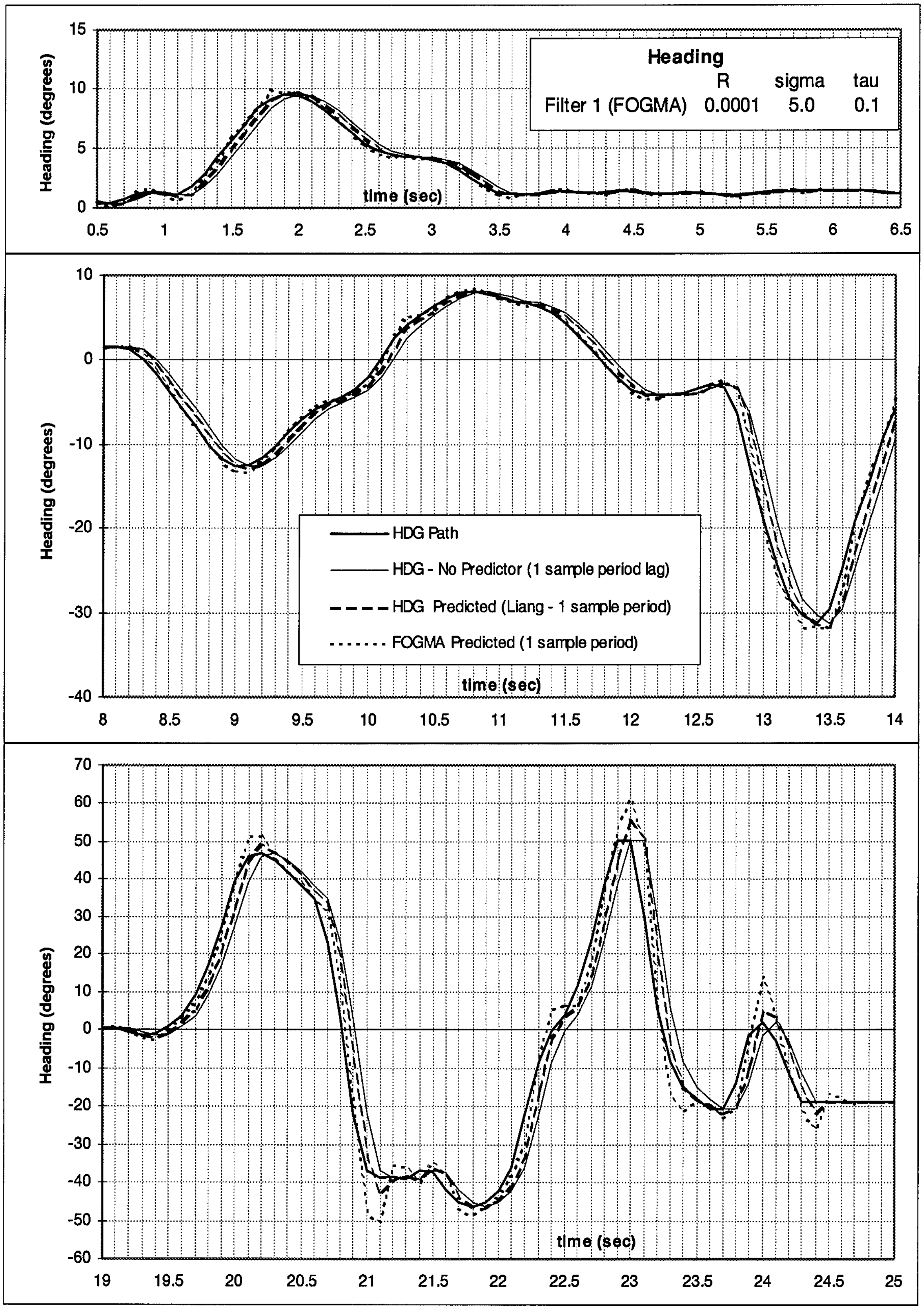


Figure 4.6

FOGMA with a mid-range time constant, τ.

Figures 4.7 and 4.8 show the effect of varying σ while holding R and τ constant. Figure 4.7 shows the output of a filter with $\tau = 1.0$ second and $\sigma = 0.5$ degrees/second^2. Figure 4.8 uses the same filter except σ is changed to 50.0 degrees/second^2. Compare these figures with Figure 4.5 with $\tau = 1.0$ second and $\sigma = 5.0$ degrees/second^2. Clearly, as σ increases, the predictions get better. This is because the filter puts less relative weight on the internal dynamics model and more on the look-angle measurement. A large σ, relative to R, results in the updated estimate, $\hat{\mathbf{x}}(t_i^+)$, closely matching $\mathbf{z}(t_i)$. Thus, the estimate for each period is propagated starting very close to the measurement made at the last sample period. The danger in making σ too large is that measurement noise will cause erratic predictions (image jitter). In this research, proposed filters were tested in the virtual environment mentioned earlier, and any that caused significant jitter were eliminated. When filters are integrated into an MMAE, σ may be further adjusted to cause the proper shifting of probability weighting between filters.

In an attempt to reduce the lag further, the propagation equation for a single sample period was changed from the standard equation:

$$\mathbf{x}(\mathbf{t}_{i+1}) = \Phi\, \mathbf{x}(\mathbf{t}_i) \qquad \textbf{(4-1)}$$

to

$$\mathbf{x}(\mathbf{t}_{i+1}) = \Phi^2\, \mathbf{x}(\mathbf{t}_i) \qquad \textbf{(4-2)}$$

Simply put, what would normally be the prediction for 2 sample periods into the future becomes the prediction for a single sample period. A "Φ^2" model will predict larger changes in position between sample periods than a standard FOGMA model with the same time constant. This approach is not physically motivated and could result in filter

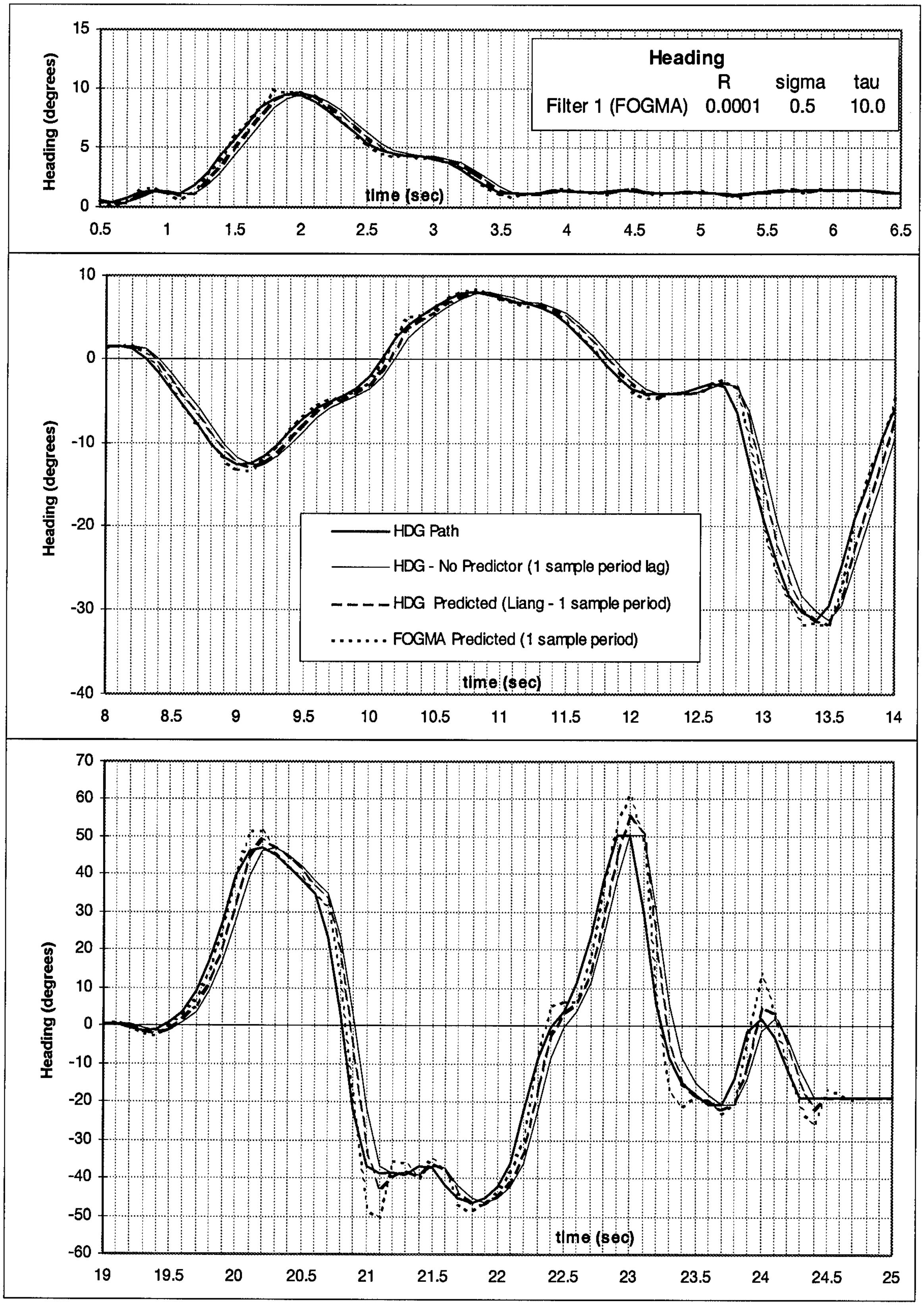


Figure 4.7

FOGMA with a small dynamic noise covariance, σ. Compare to Figures 4.3 and 4.8.

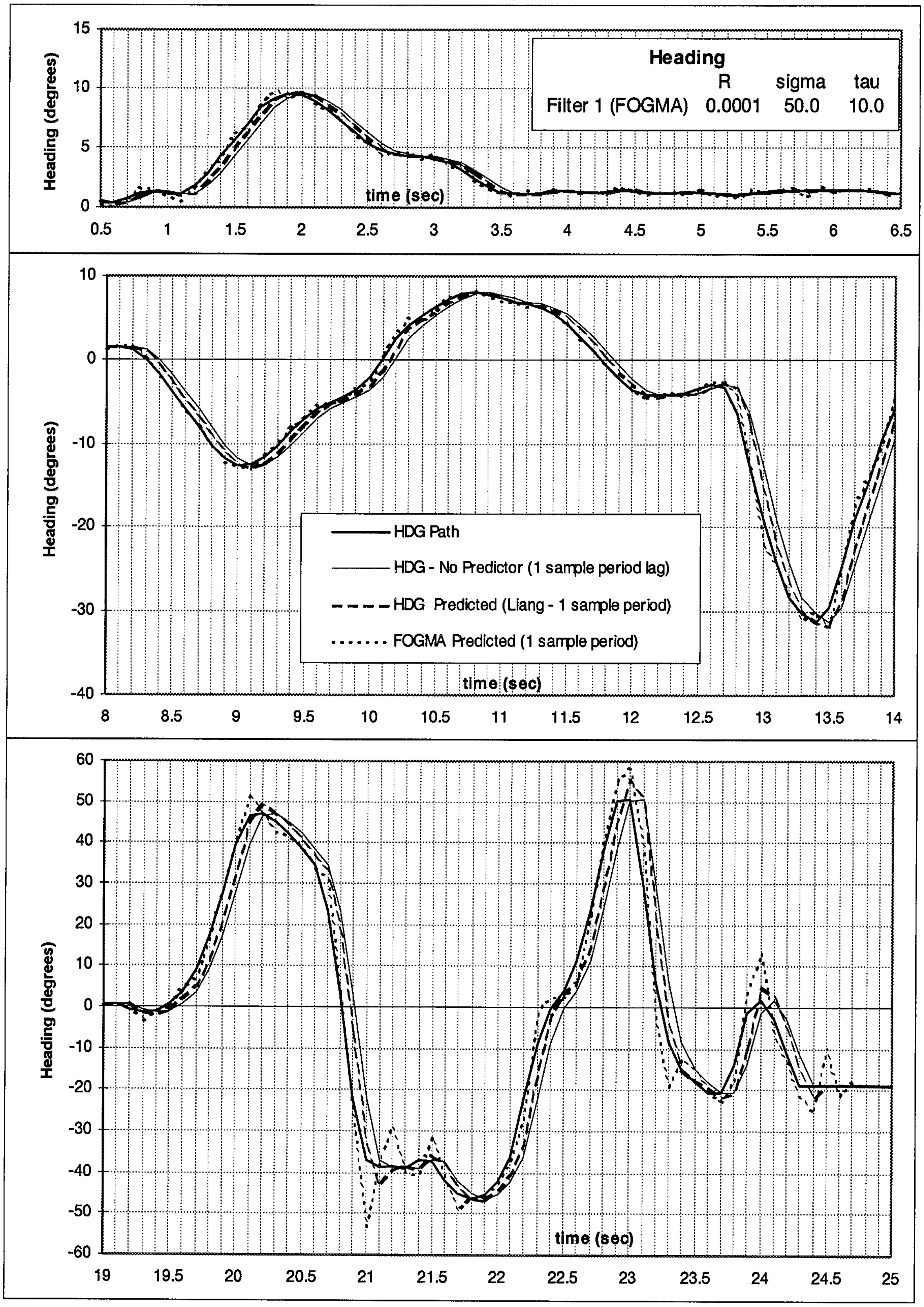


Figure 4.8

FOGMA with a large dynamic noise covariance, σ. Compare to Figures 4.3 and 4.7.

instability. Squaring Φ is also somewhat arbitrary and easy to compute; a final design may use Φ appropriate for predicting over an interval of time other than two sample periods (e.g., 1.5 or 1.6 sample periods) - whatever period of time that yielded the *best* performance. Using Φ^2 is simply a convenient first attempt at further reducing lag. Figure 4.9 shows the result of using a single "Φ^2 filter" with a state transition matrix Φ evaluated for a time constant of $\tau = 5$ seconds. The performance is especially striking during moderate levels of head motion ($t = 8$ through 14 seconds). Lag is almost completely eliminated, and the overshoots are no greater than in Liang's filter. However, ringing remains a significant problem (as seen during $t = 13.0$ through 13.6 seconds, $t = 21.0$ through 22.0 seconds, and $t = 24.0$ through 25.0 seconds).

4.4 Elemental FOGMV Filters

Figures 4.10 and 4.11 show extreme cases of very large and very small τ values used in a FOGMV filter. In Figure 4.10, $\tau = 10$ seconds models the dynamics as nearly constant velocity. Note that this model gives almost identical results as the FOGMA nearly-constant-velocity model of Figure 4.4. In Figure 4.11, $\tau = 0.01$ seconds models the dynamics as nearly constant position. This model is most effective during times of small, slow movements. It also tends to smooth the displayed image by filtering out measurement noise. But, it shows a lag very near what the system would experience with no predictor at all. Figure 4.12 shows a FOGMV filter with an intermediate time constant of $\tau = 1.0$ seconds. Close comparison of Figures 4.10 and 4.12 show, as expected, the larger τ of Figure 4.10 results in less lag and more overshoot. Figure 4.13 shows that

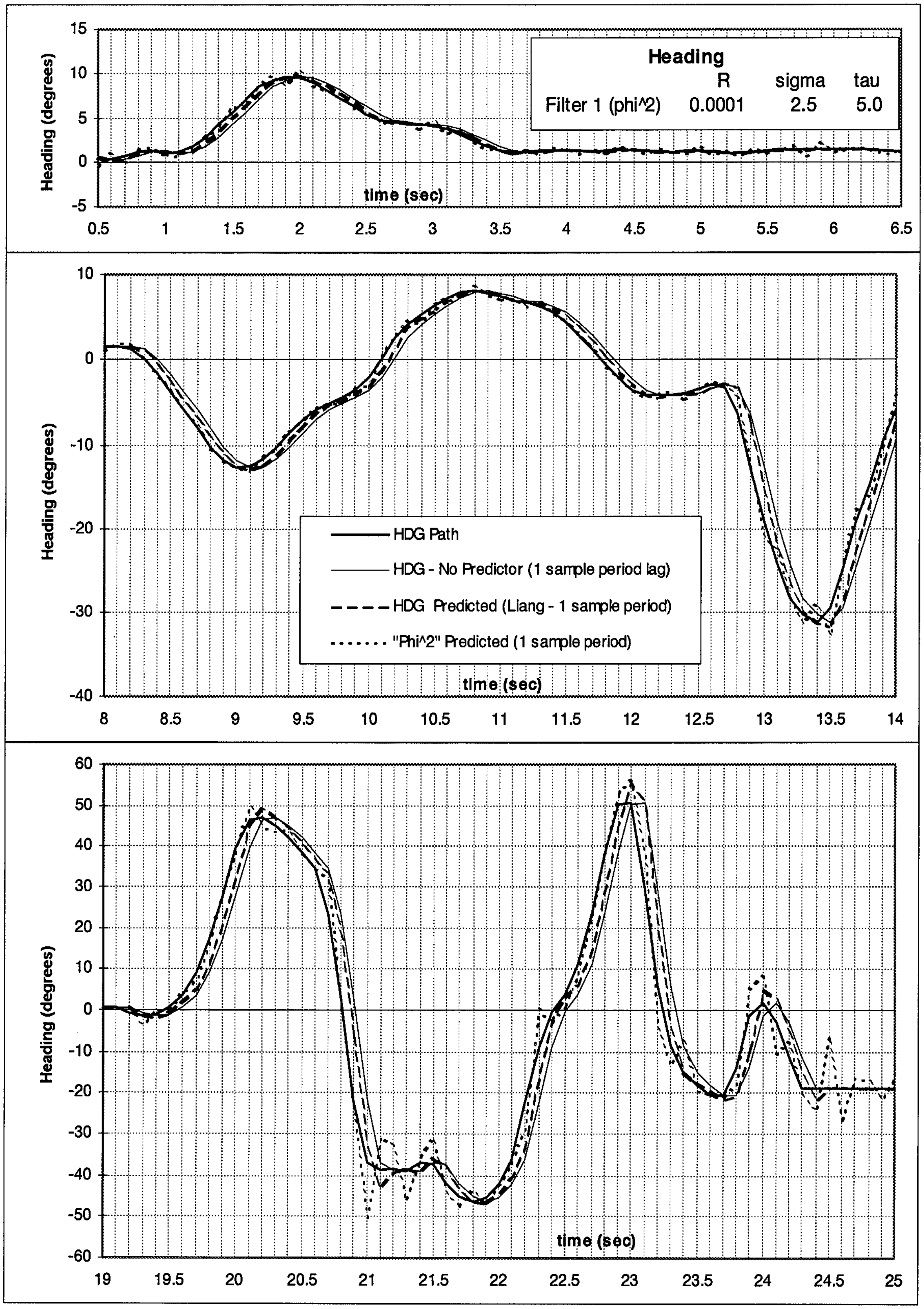


Figure 4.9
Initial tuning of a "Φ^2 filter."

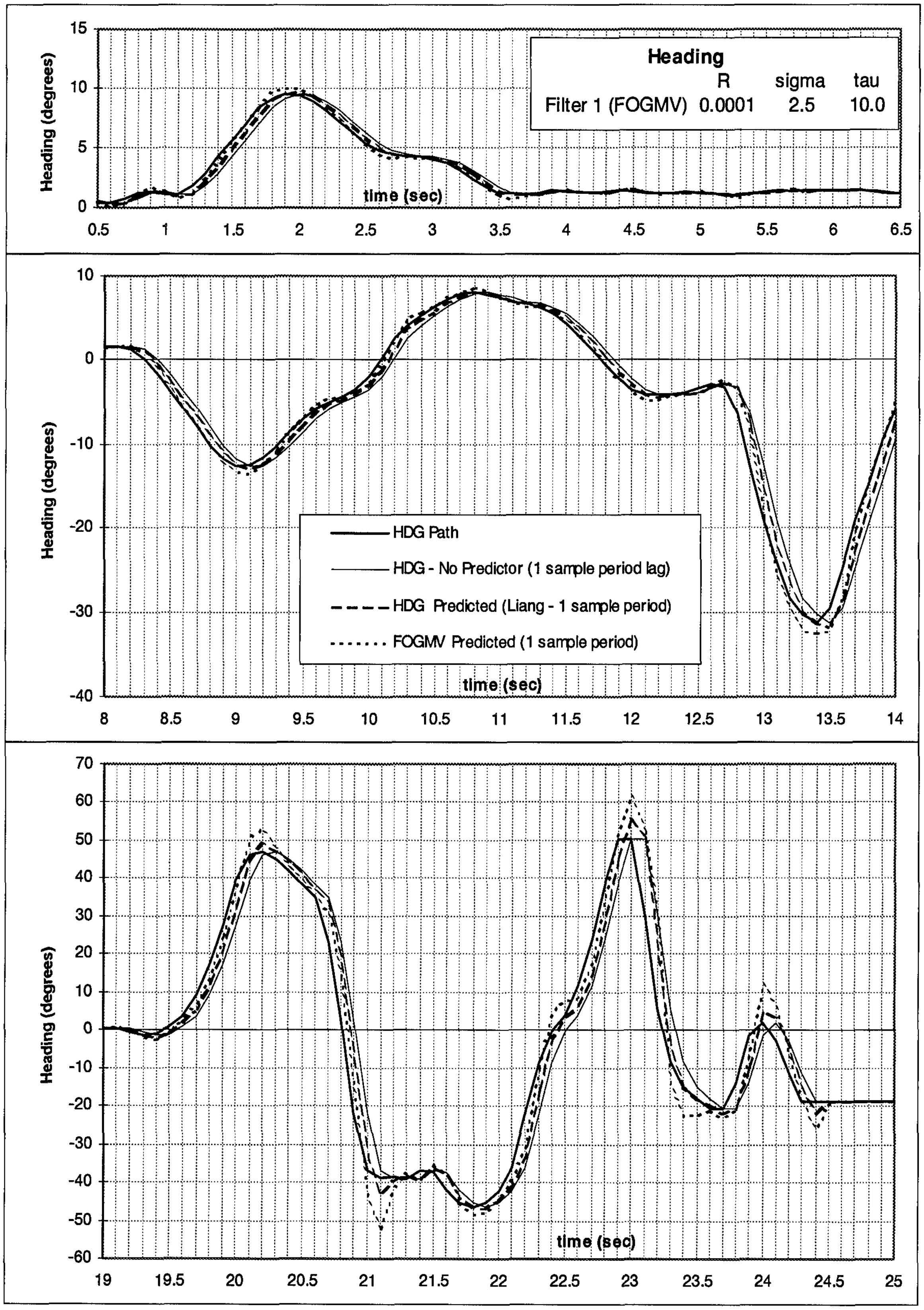


Figure 4.10

Nearly-Constant-Velocity FOGMV

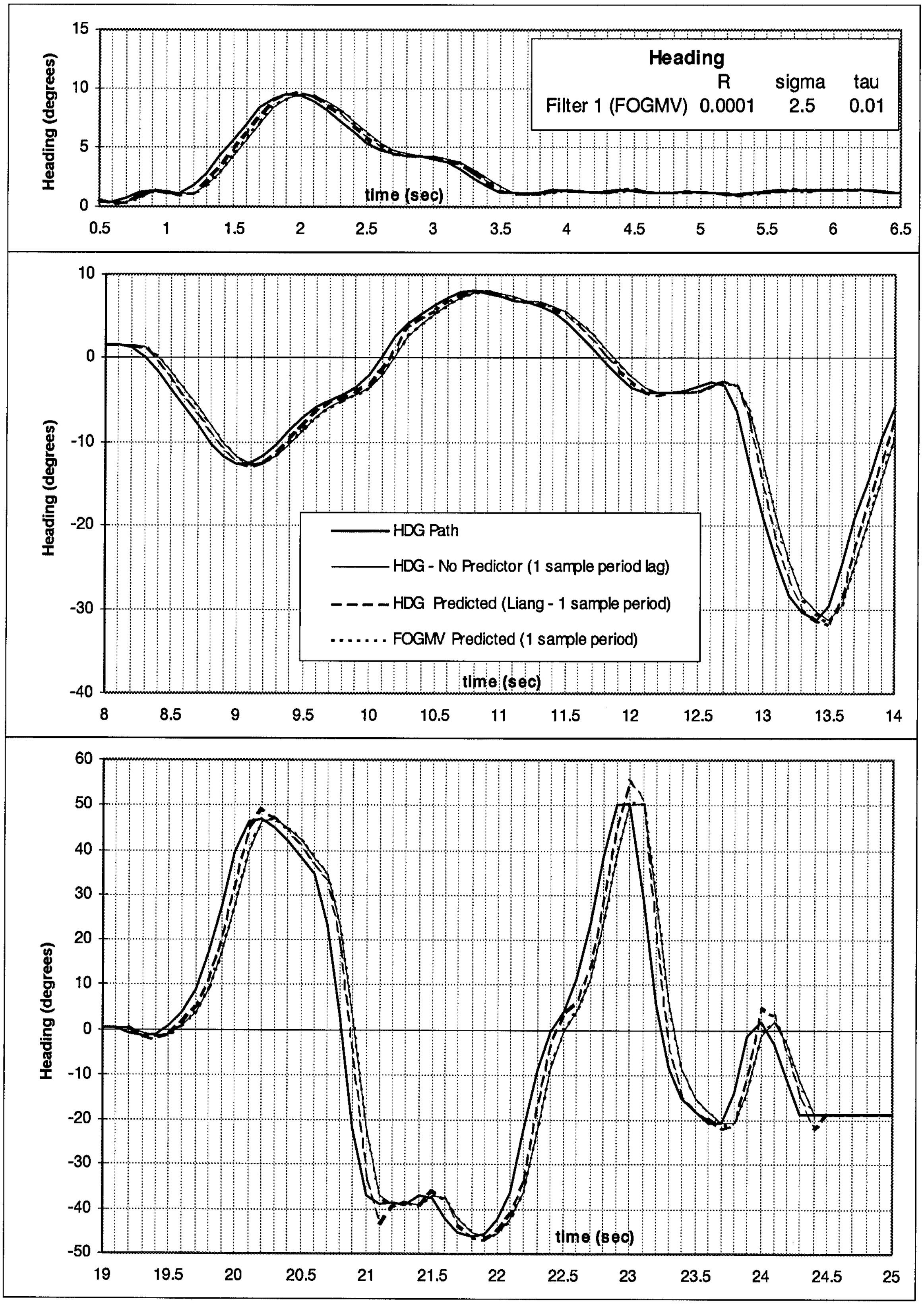


Figure 4.11
Nearly-Constant-Position FOGMV.

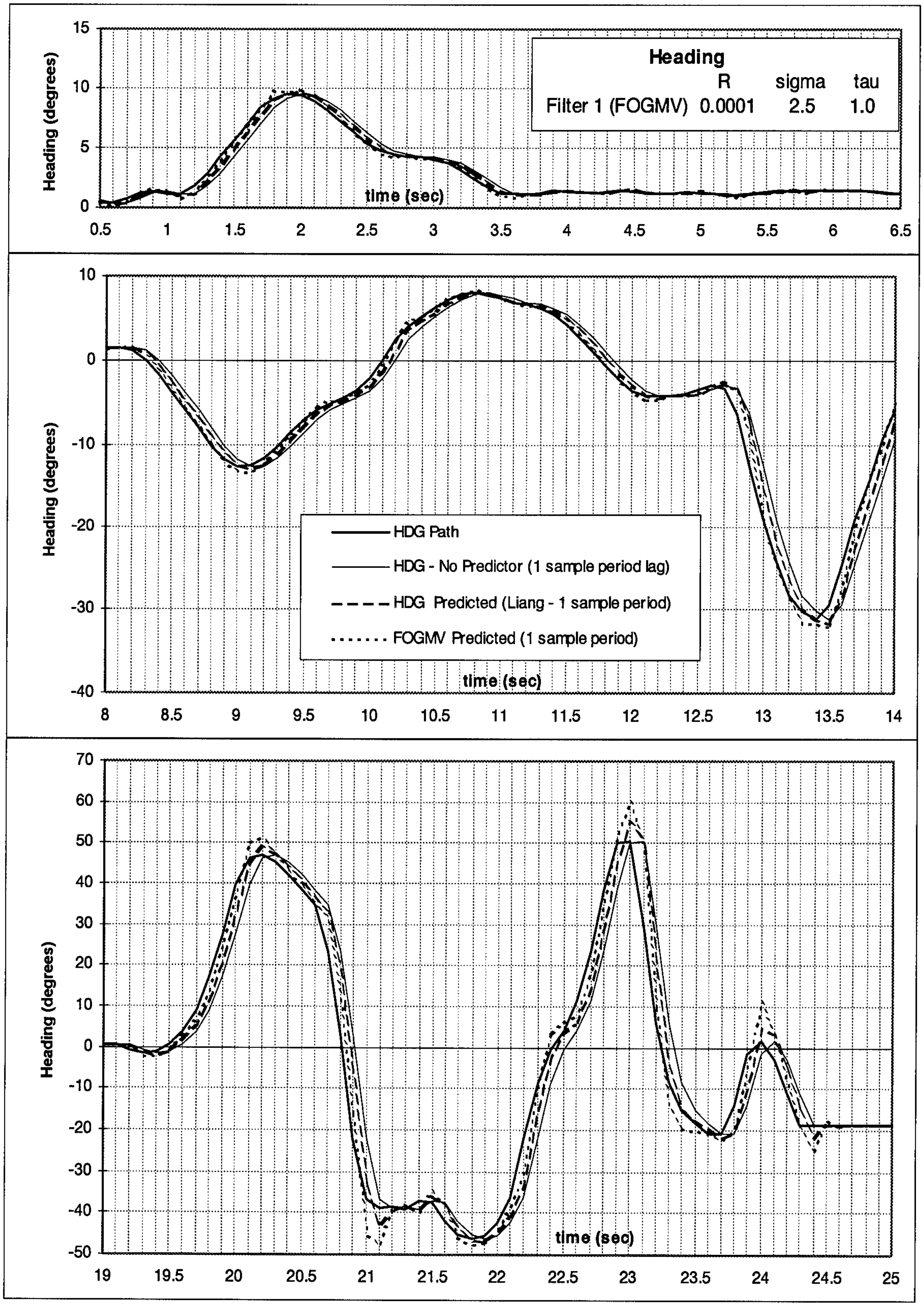


Figure 4.12

FOGMV with a mid-range time constant, τ.

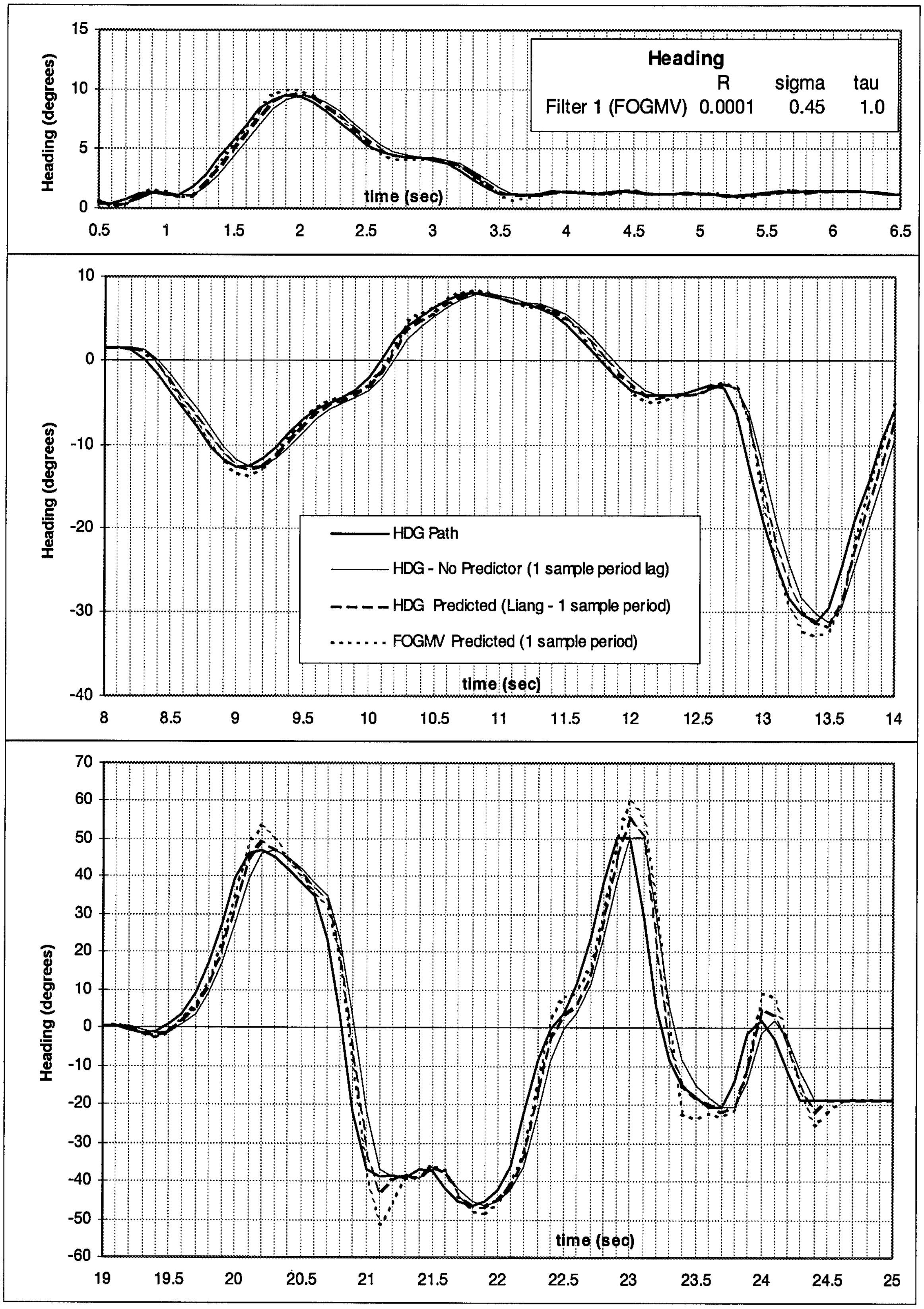


Figure 4.13

FOGMV with a small dynamics noise covariance, σ. Compare to Figure 4.12.

holding τ constant (compared to that of Figure 4.12) and decreasing σ degrades the performance of the filter ($\sigma = 0.45$ was chosen since it is close to Liang's non-adaptive FOGMV value). This is the same effect discussed in the previous section. Note that the best σ value for the FOGMV model ($\sigma = 0.45$) is much smaller than the best σ value for the FOGMA model ($\sigma = 5.0$). There are two explanations for the difference in σ values. First, accelerations have larger magnitudes than velocities (see Figure 4.1). Second, the FOGMV model may be more accurate over the entire range of head motion. A more accurate model requires less tuning noise.

4.5 MMAE Design

The MMAE attempts to form the best possible prediction by taking a properly weighted average of the elemental filter predictions. If, at a particular time, t_i, filter 1 has an error of -5 degrees, filter 2 has an error of -3 degrees and filter 3 has an error of -2 degrees, then the best possible MMAE solution is to have all the probability weight on filter 3, resulting in an error of -2 degrees. If the elemental filters bracket the true look-angle (one has a positive error and another has a negative error) an ideal solution would blend the two predictions to get the exact answer. Practically, it is impossible to know a priori, what the correct weighting will be for a given time. The appropriate weighting for a prediction at time t_i must be based on the information available at t_{i-1}. With this information designers can look at the performance of individual elemental filters when choosing the specific elemental filters to be incorporated into an MMAE. The best combination would be filters which consistently complement each other. That is, any possible head motion could be accurately modelled by a weighted average of the elemental filters. Since all predictors display lag, to different extents, it is impossible to bracket the true look-angle at all times.

The first step in designing an MMAE consisting of three FOGMA elemental filters is to look at the performance of the individual filters. The filter of Figure 4.5 gives the best performance over almost the entire range of motions. It consistently displays less lag and equal or less overshoot than any of the other FOGMA filters. For this reason, it would most likely be unproductive to put another FOGMA elemental filter into an MMAE with the filter of Figure 4.5.

From inspection of the FOGMV elemental filters' performances (Figures 4.10 through 4.13), it appears a bank of FOGMV filters may outperform any single FOGMV. The nearly-constant-position filter of Figure 4.11 gives excellent performance during benign motion (t = 4.0 through 8.0 seconds and t = 16.2 through 19.0 seconds), but when there is any significant motion, it displays a lag of one full sample period. The nearly-constant-velocity filter of Figure 4.10 removes significant amounts of lag, compared to the nearly-constant-position filter (t = 8.5 through 15.5 seconds), but suffers from overshoot. Combining these two filters properly, along with a filter having characteristics between the constant-velocity and constant-position models, in an MMAE should give better results than those obtained from either filter operating alone. Figure 4.14 shows the output of an MMAE using three FOGMV elemental filters. Comparing Figure 4.14 to Figures 4.10 and 4.11 shows that the MMAE yields less lag than the nearly-constant-position filter and less overshoot than the nearly-constant-velocity filter. The lags when the look-angle

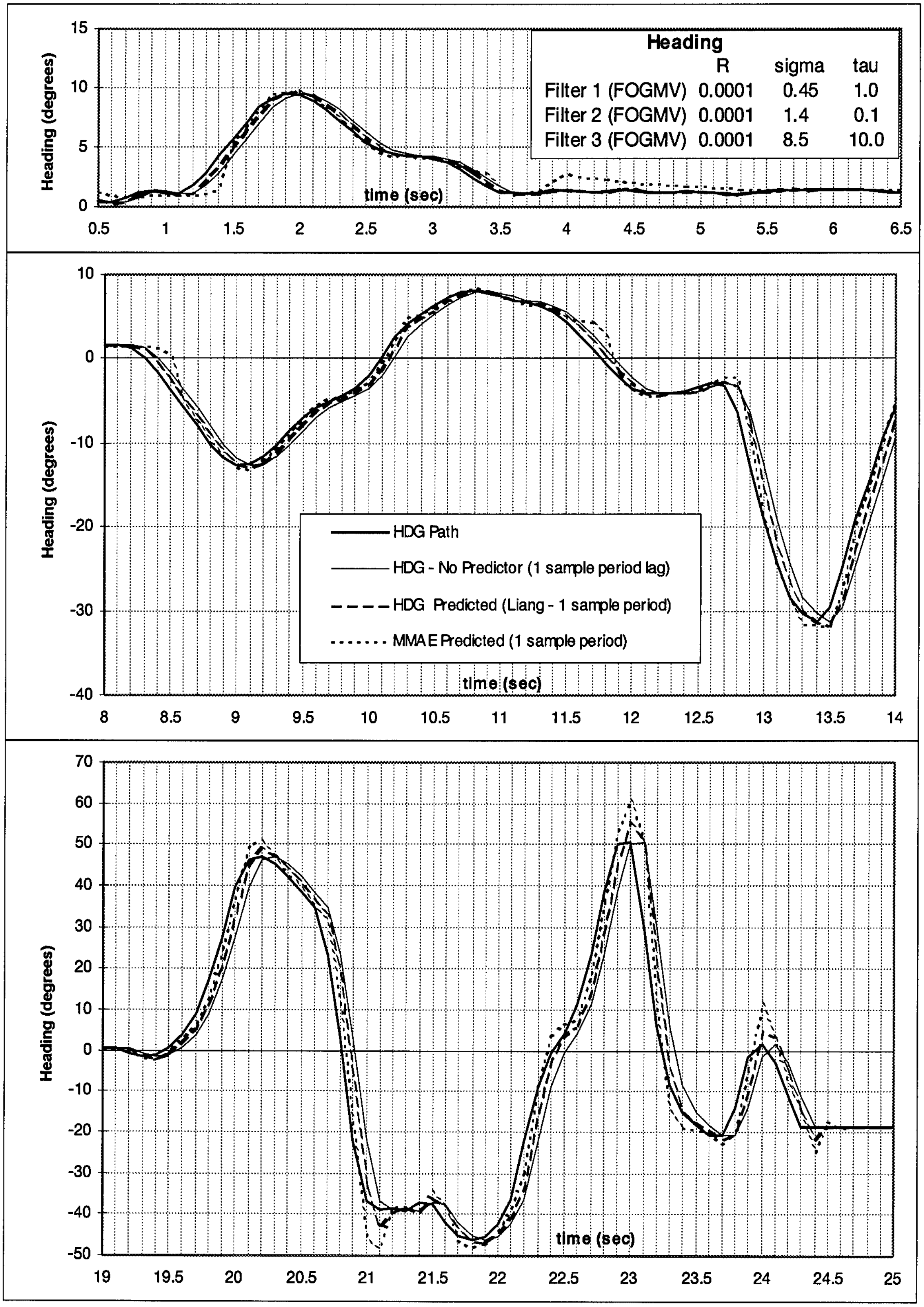


Figure 4.14(a)
MMAE consisting 3 FOGMV elemental filters.

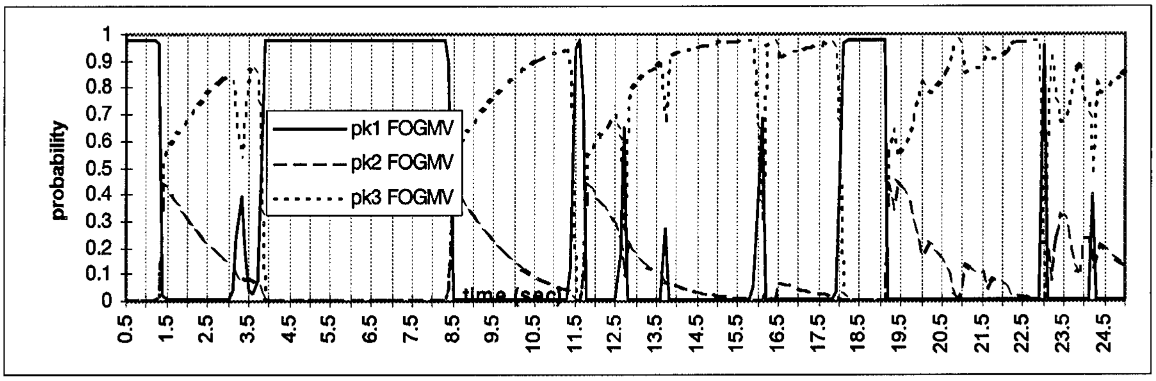


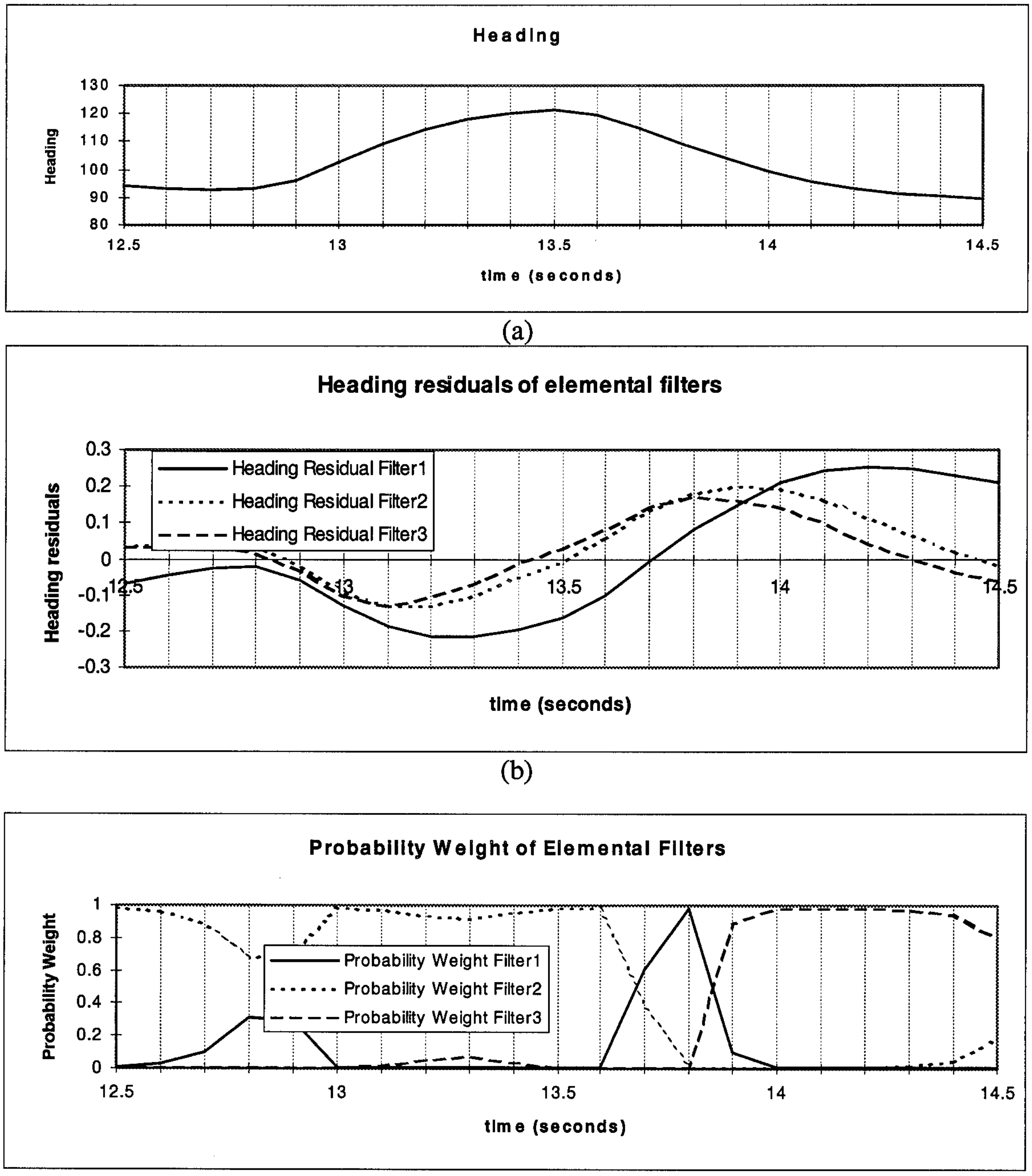
Figure 4.14(b)
Probability Weighting of 3-FOGMV MMAE.

transitions from zero velocity are caused by the dynamics noise strength, related directly to σ, on filter one (nearly-constant-position) being low relative to the measurement noise variance, R_{11}. When the heading is stable (zero velocity), filter one absorbs all the probability. When motion starts, the MMAE is slow to recognize the motion as more than just noise. Once motion is established, filter one quickly loses probability. Further tuning may reduce overshoots. Instead of following this path, this research aimed to reduce lag further, necessating an elemental filter with smaller lag characteristics than achievable with the FOGMV model.

Before discussing mixed MMAEs (MMAEs with combinations of FOGMV, FOGMA, and “Φ^2” filters), it is worth mentioning two characteristics of this research. Both of these problems, as discussed in Sections 4.6 and 4.7, affect any MMAE design for virtual displays.

4.6 Spike Phenomena

Some tunings of the MMAE using three filters displayed a visually disconcerting characteristic dubbed the “spike phenomena”. When an observer, using a virtual display, rapidly changes directions of head motion, the displayed image adjusts to the motion (with some lag and overshoot discussed earlier). A few sample periods after the change in direction, the displayed image makes a jump change back toward the look-angle where the change in direction took place. The image then returns to its previous trajectory. The



(a)

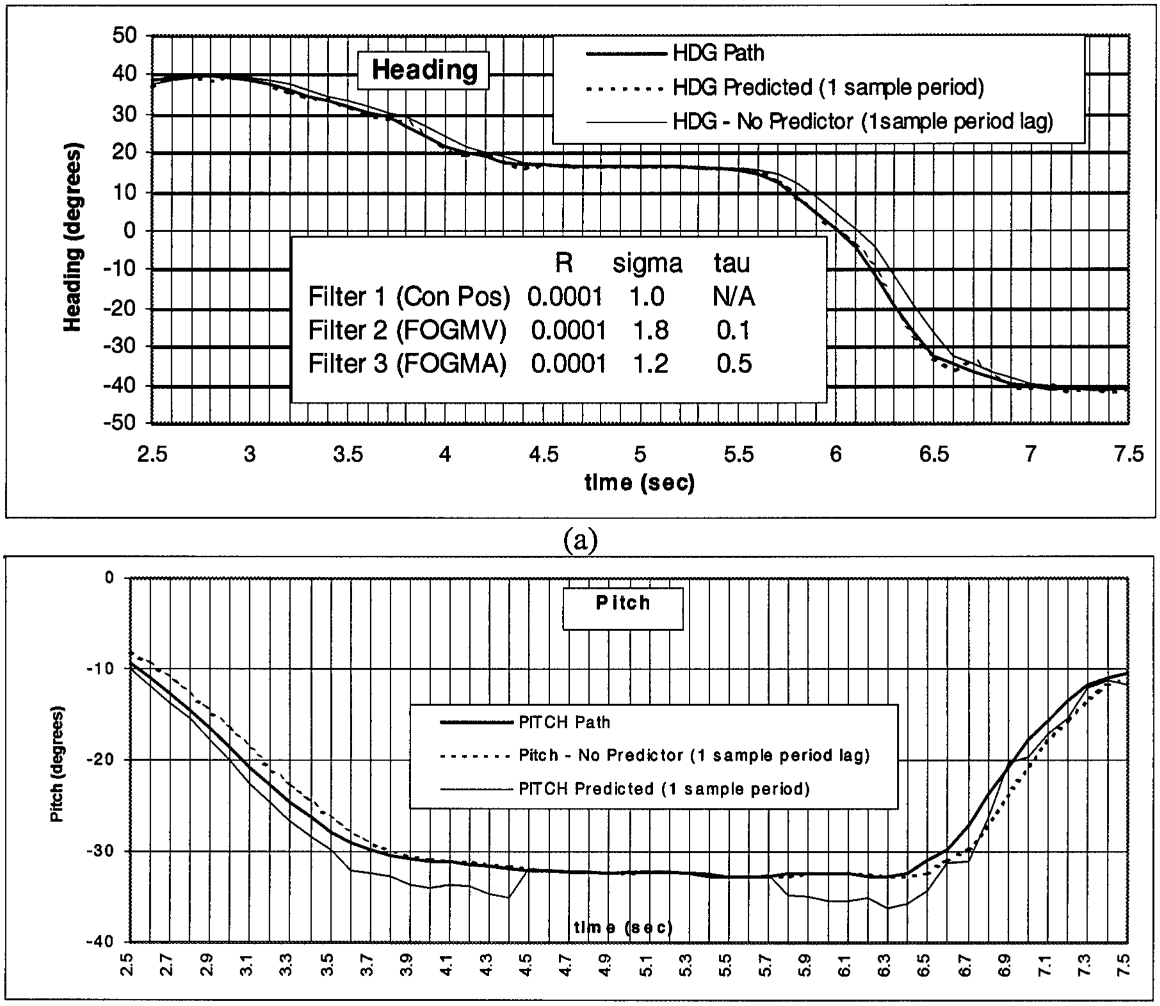
(b)

Figure 4.15 (c)
"Spike Phenomena." Demonstrated using 3 FOGMV elemental filters, having time constants, τ, of 5, 1, and 0.1 seconds.

reason for this phenomena is shown in Figure 4.15. Figure 4.15(a) shows that, at t = 13.5 seconds, heading velocity changes direction, from positive to negative. Figure 4.15(b) shows that, at this time, all three filters were undershooting the actual look-angle. That is, as the magnitude of the look-angle increased, the prediction of the magnitude was too small. When the direction of motion changes, each filter estimate transitions from an undershoot to an overshoot. As this happens, the residual of each filter changes sign. If the MMAE contains one filter with a much larger time constant than the other two, the residuals of the filter with the large time constant (slow reaction time) will cross the zero axis much later than the residuals of the other filters. At the sampling time when the slower filter is crossing the zero axis, the magnitude of the slow filter's residual is much smaller than the residuals of the faster filters. The MMAE will shift the majority of its probability to the slower filter (Figure 4.15c), and the image will jump to the estimate made by the slower filter. With time, the slow filter residuals continue to increase in magnitude and the MMAE shifts the probability back to the filters with smaller time constants. Users consistently complained when they experienced the spike phenomena. Any design that produced any significant spike phenomena was considered unusable.

4.7 Interaction of Heading and Pitch

In this research a single MMAE weights the filter predictions based on the residual vector $\mathbf{r}(t_i)$. The head tracker outputs a unit vector in Cartesian coordinates [x, y, z] pointing in the direction the head is looking. Heading is computed from the x and y components of the vector (heading = tan(x/y)). Pitch depends only on the z component (pitch = sin z). However, the weighting of filter outputs depends on the total residual. If the heading is changing rapidly (driving the MMAE to weight the most aggressive filter) and pitch is changing slowly (driving the MMAE to weight the most benign filter), the final weighting will tend to be too aggressive for the pitch prediction and not aggessive enough for the heading prediction. Figure 4.16 shows the predictions made by a 3-FOGMV MMAE for both heading and pitch over the same time period. Since heading uses two of the three vector components, heading predictions tend to be more accurate than pitch predictions. But, the predictions in both axes are degraded from the best



(a)

Figure 4.16(b)
Comparison of Pitch and Heading predictors.

performance achievable (by considering heading and pitch channels separately in the MMAE). Solutions to this problem are discussed in Chapter 5.

4.8 Mixed MMAE Designs

To gain the advantages of the "Φ^2 filter" (a FOGMA filter using the square of the state transition matrix to propagate a single sample period) and reduce or eliminate the ringing, the "Φ^2 filter" can be combined with Liang's filter in an MMAE. Figure 4.17 shows an initial attempt to combine the two filters into an MMAE. Notice that the ringing is still present. Several other tunings, adjusting the dynamics noise strength (through the σ parameter) and time constants (τ) for both filters, produced unsatisfactory ringing, overshoot and/or lag.

To improve the results, a constant-position elemental filter was added to the MMAE. The constant-position filter gives excellent results when the pilot's head is motionless, or moving very slowly. Most importantly, it can absorb the probability (via the p_k computations in the MMAE) while the "Φ^2 filter" is ringing after a sudden stop, since its residuals will be smaller than those of the "Φ^2 filter" at such times.

Figure 4.18(a) shows the first attempt at using the three filters together in an MMAE. The obvious problem here is the step changes in position which occur when the constant-position filter is dominant (during benign motion, e.g., t = 1.0 through 3.0 seconds, and 8.0 through 12.0 seconds). This happens because the constant-position filter

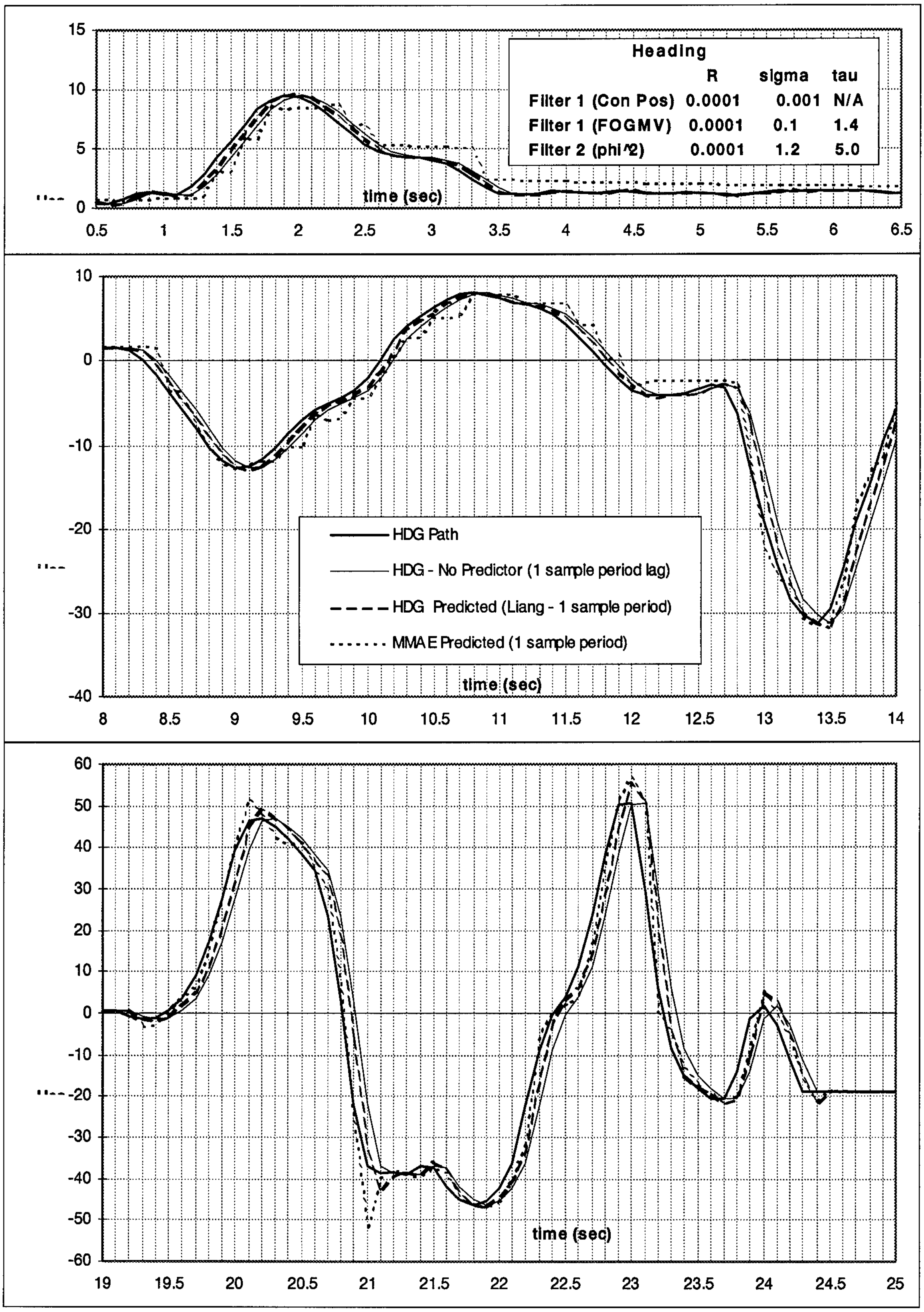


Figure 4.17
MMAE using a FOGMV and a "Φ^2" filter.

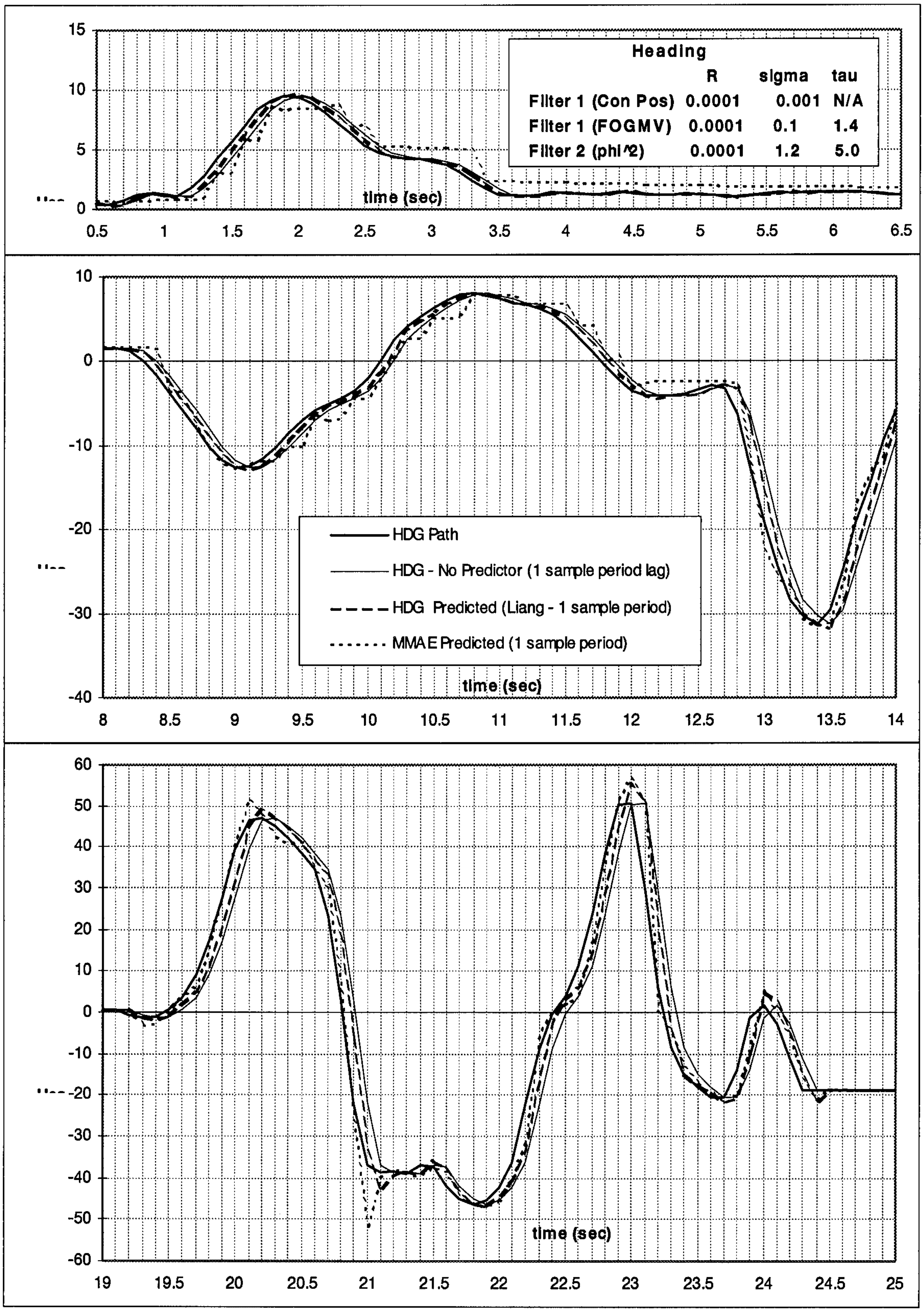


Figure 4.18 (a)

MMAE consisting of a constant-position, FOGMV, and "Φ^2" filter.

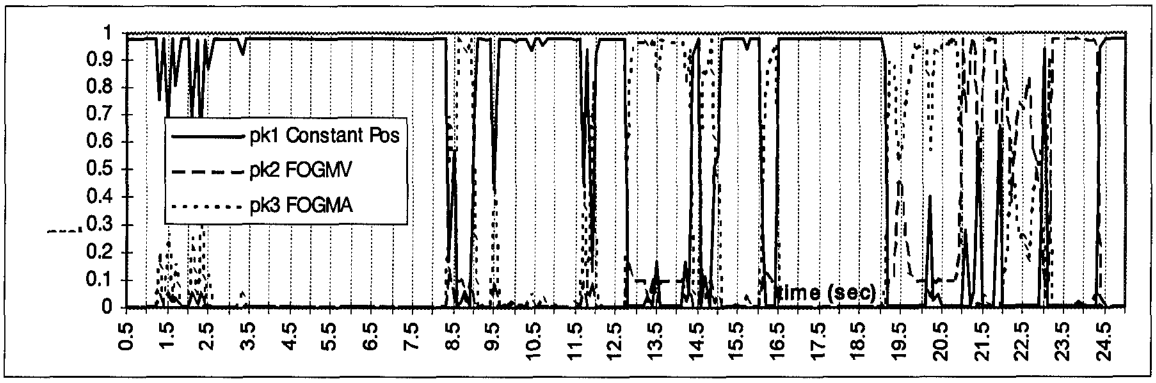


Figure 4.18(b)
Probability Weighting of MMAE consisting of a constant-position, FOGMV, and "Φ^2" filter.

gives too much weighting to the model and not enough to the measurement. As a result, even when the measurement indicates motion, the filter tends to choose the model's prediction of no motion. To eliminate this problem, either the dynamics driving noise strength can be increased (by increasing the σ parameter), or the measurement noise variance, R, can be decreased. In this research, σ was increased. This increases the filter's weighting on the measurement and allows the constant position filter to dominate the other two filters during the appropriate trajectories. In addition to adjusting σ for the constant-position filter, the "Φ^2 filter" was retuned to reduce the overshoot (t = 20.0, 21.0 and, 23.0 seconds).

Figure 4.19 shows the final tuning of the MMAE and the best results obtained in this research. Along with the noise values (σ and R) and time constant (τ), notice the divergence thresholds. When a filter's $\mathbf{r}(t_i)\mathbf{A}^{-1}(t_i)\mathbf{r}(t_i)$ value exceeds its divergence threshold, the filter is declared divergent. Its prediction is given zero weighting by the MMAE and the position states of the divergent filter are replaced with the MMAE position states computed without the divergent filter (this is called an elemental filter

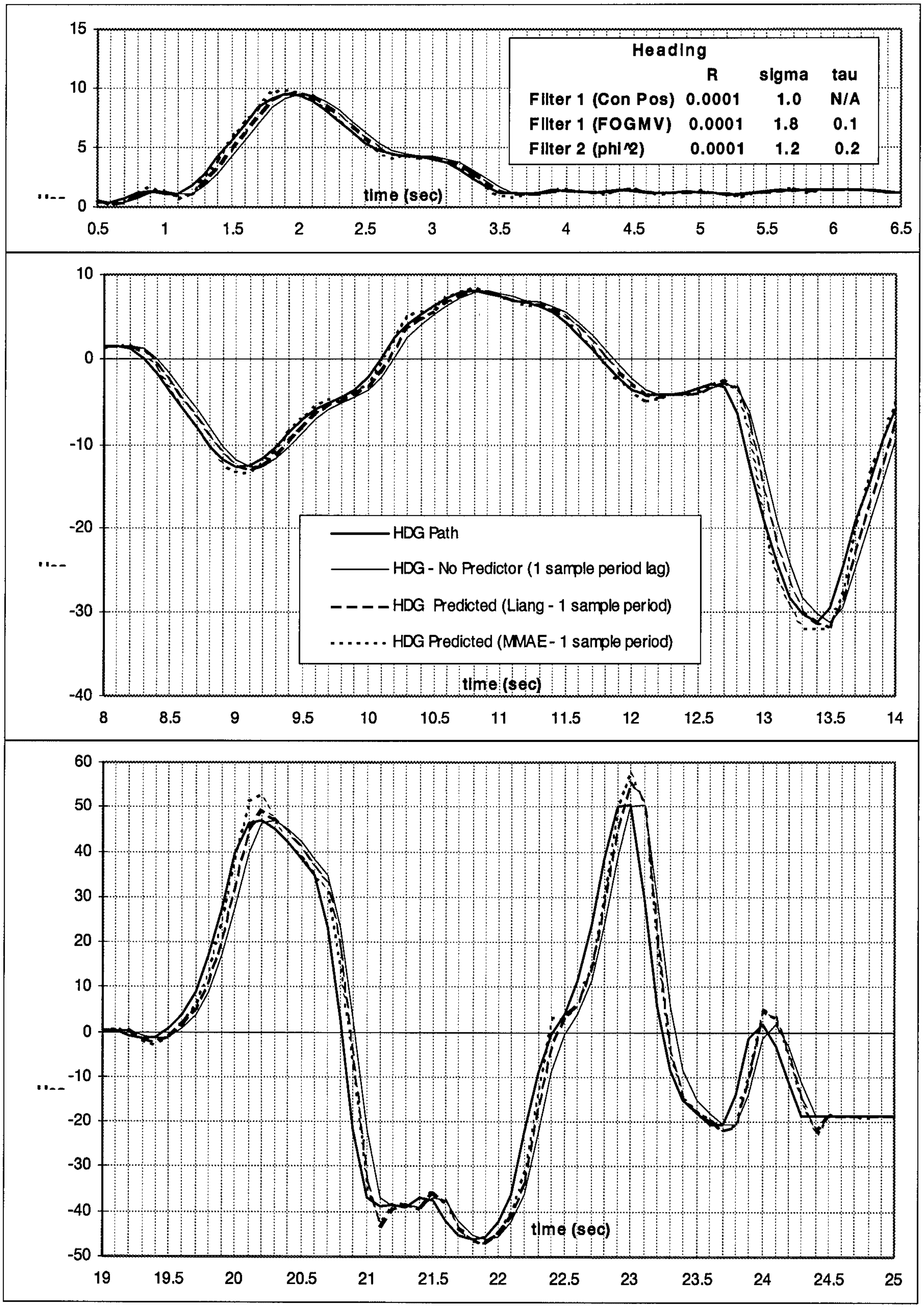


Figure 4.19(a)
Final MMAE design.

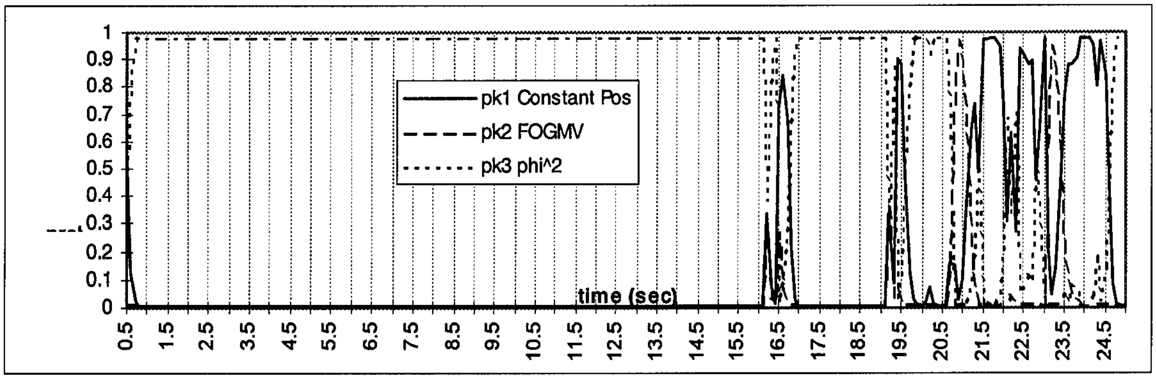


Figure 4.19(b)
Probability Weighting of Final MMAE design.

"restart"). Since the constant-position filter has a relatively large lag (comparable to the lag with no predictor at all), it is only effective when there is very little head motion. To ensure the MMAE quickly discounts the constant-position filter's prediction when motion begins, its divergence threshold is set very low. The result is that the constant-position filter is being declared divergent and being restarted almost continuously, except when head movements are very slow.

Notice that during benign motion, lag is almost completely eliminated (t = 0.5 through 6.5 seconds - the dotted line representing the MMAE prediction is obscured by the solid line of the true path). During moderate motion, the performance is almost as good (t = 8.0 through 14.0 seconds). The MMAE prediction oscillates about the true path with errors much smaller than those of Liang's filter, and with overshoots much less than 5 degrees. In fact, the MMAE gives a smaller error than Liang's filter throughout the period of moderate motion except for individual sample periods when Liang's filter overshoots the true path later than the MMAE (t = 9.0, 13.4 seconds). During very rapid head motion (reacquisition: t = 19.0 through 24.5 seconds), after an initial overshoot (t =

20.2 seconds), the MMAE performs very much like Liang's filter. It is important to note that the MMAE could be tuned to reduce lag during the reacquisition phase further by forcing more probability to the "Φ^2 filter", but that would increase overshoots. Instead, the MMAE was tuned to demonstrate lag could be significantly reduced during benign and moderate head motion (i.e., during tracking tasks by the pilot, when fidelity of simulation is most important, rather than during aggressive and short-term head motions representative of acquiring a new target), without a significant increase in the overshoots (compared to Liang's nonadaptive Kalman filter) or degredation in performance during reacquisition motions.

Table 4.1 compares the overshoot and lag performance of the MMAE, Liang's nonadaptive filter and no predictor (one sample period lag). Each of the three types of motion (benign, moderate and reacquisition) last six seconds, yielding 61 data points each. The "% improvement row is a measure of the improvement relative to no predictor. This table was generated by dividing each estimate into one of three categories: accurate (the estimate was within 0.5 degrees of the true heading, as particularly between t = 3.7 seconds and t = 6.5 seconds), lag (the estimated heading trailed the true heading in time), and overshoot (the estimate did not have a direct connection with a specific true heading, such as at t = 23.0 seconds). Overshoot was calculated as the magnitude of the difference between the predicted and true headings, giving all positive numbers. Lag was calculated as the time between the true position reaching a heading and the predicted position reaching that same heading (using linear interpolation between sample times to compute the exact time when the estimate reached the same value as the actual heading at a given (earlier) sample time). Notice that the MMAE outperforms Liang's filter in every category except overshoot in the Reacquisition phase. Also, note that the lag in the MMAE varies much more than in Liang's design. The most dramatic improvement is in the predictions of moderate motion. The MMAE gives 65% more points within the 0.5 degree "accurate" tolerance and a 75% reduction in lag points. To achieve this the number of overshoot points increased, but the average overshoot is very small.

	No Predictor	Liang			MMAE		
		Benign	Moderate	Reacquisition	Benign	Moderate	Reacquisition
Lag							
Number of lag points	61	9	33	39	1	8	41
Average lag (seconds)	0.1	0.058	0.058	0.059	0.048	0.030	0.051
% improvement		42	42	41	52	70	49
Overshoot							
Number of overshoots	0	0	2	10	1	10	12
average overshoot (degrees)	0	0	1.37	2.98	0.65	1.12	3.75

Number of predictions within 0.5 degrees (out of a possible 61)	47(benign)
	15(moderate)
	14(reacquisition)
		52	26	12	59	43	8

Table 4.1
Comparison of lag and overshoot between MMAE, Liang's nonadaptive filter and no predictor.

Table 4.2 compares the temporally average and maximum error magnitudes (i.e., error absolute values) of the three designs. The MMAE gives the best improvements during benign and moderate motion (substantially better than those of Liang's filter), but even during reacquisiton motion the MMAE gives substantial error reduction. The largest errors occur during periods of lag, not during overshoot.

4.9 Extending the Predictions

Since this research is intended to be independent of a specific sampling time, it is worth noting the performance of the final design when it predicts over longer periods. As expected, extending look-angle predictions further into the future increases the error. Fig 4.20 shows the results of predicting 200 msec (2 sample periods) into the future. Overshoots are more common and larger than seen in the corresponding results of Figure 4.19 for the case of predicting 100 msec (1 sample period) instead. The 200 msec prediction is what some researchers used in their virtual environment. The most troublesome part of the errors was the lag when motion reversed direction. It is evident from Figure 4.20 that the MMAE would need a significantly different tuning to be used for 2-sample-period predictions.

4.10 Summary

This chapter showed the development of an MMAE to reduce lag in virtual displays. The final MMAE design clearly outperformed Liang's single nonadaptive Kalman filter predictor. The MMAE can be tuned to acheive the best trade-off between good lag reduction and undesirable overshoot.

Average |error| for each type of motion (degrees)	No Predictor	Liang	MMAE
benign	0.34	0.23	0.167
moderate	1.68	0.99	0.57
reacquisition	5.90	4.05	3.42
average	2.64	1.76	1.39
% improvement (compared to no predictor)			
benign		33.5	50.7
moderate		40.9	65.8
reacquisition		31.3	42.0
average over all 3 types of motion		33.4	47.4
Maximum error (degrees)			
benign	1.47	1.00	0.71
moderate	6.48	5.06	3.11
reacquisition	24.93	21.22	20.27
% improvement (compared to no predictor)			
benign		31.6	51.9
moderate		21.9	51.9
reacquisition		14.9	18.7

Table 4.2

Comparison of performance of MMAE with Liang's nonadaptive filter and no predictor.

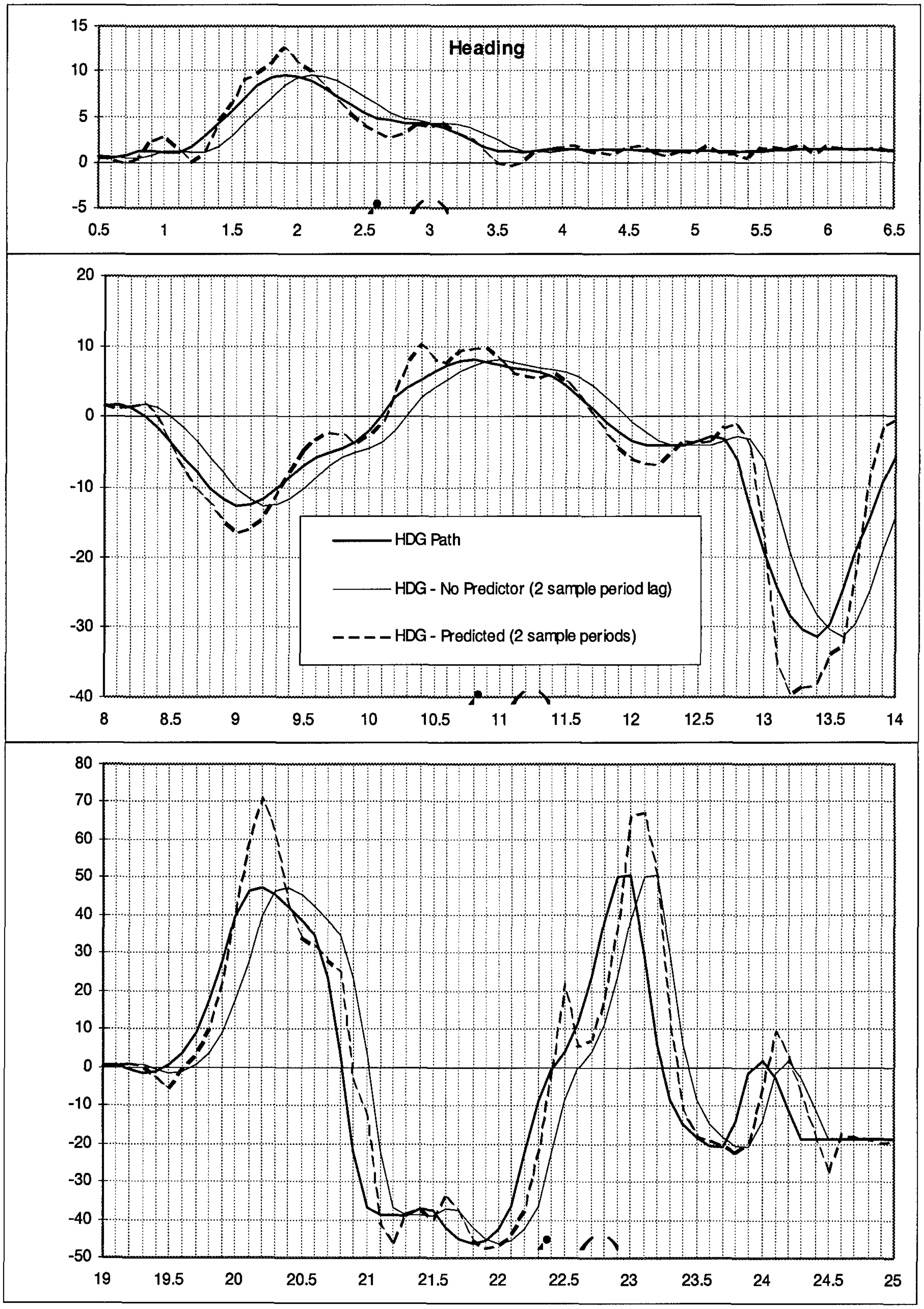


Figure 4.20

Final MMAE design predicting 2 sample periods into the future.

5. Conclusions and Recommendations

This chapter presents the conclusions drawn from the results of Chapter 4. This is followed by recommendations for future research.

5.1 Conclusions

Multiple Model Adaptive Estimation has been shown to reduce lag significantly in virtual displays. It results in less lag than the single nonadaptive Kalman filter designed by Liang [12] and its overshoot characteristics show a slightly larger magnitude, but equal duration (usually one sample period). The improved performance is most dramatic during benign and moderate head motion, i.e., for the tracking most common in virtual environment simulations. During abrupt reacquisition motions, the MMAE still significantly reduces lag, but at the expense of overshoot and some ringing. Various design schemes showed different amounts of lag and overshoot, but all except the constant position filter displayed both lag and overshoot. Lag cannot be eliminated without a priori information about future head motion. Overshoot is inherent in any predictive scheme, especially when position is being predicted based only upon position measurements (velocity and acceleration are not currently measureable).

Motion estimations in each axis (pitch and heading) interfered with each other. The residuals for heading and pitch contribute to decisions about the type of motion present (benign, tracking, or reacquisition). This implies that both heading and pitch are experiencing the same type of motion. In reality, heading and pitch change independently. For example, consider a pilot tracking a nearby target on the ground from low altitude. His look-angle heading will change rapidly as he flies past the target, looking to the side to see the target and then straight ahead to clear his path. During this same time his look-angle pitch will change very little. The MMAE will form a single estimation of look-angle dynamics which will be a compromise between what is needed for pitch and heading predictions.

MMAE can be very helpful if a single sample period overshoot is acceptable. It seems overshoot is unavoidable in a predictive algorithm based only on position measurements. A single filter cannot model all the look-angle trajectories typical of a single simulator mission. Adaptive schemes (such as MMAE) cannot change models until after elemental filter residuals reveal the need for an altered hypothesis (i.e., a different elemental filter appropriately gaining the highest probability weighting). The MMAE in this research responded to changing dynamics in a *single* sample period (as quickly as possible) and drove the prediction error to nearly zero. Higher sampling rates have two advantages during times of changing dynamics. First, overshoot errors have smaller magnitude. Second, the overshoot error is displayed for a shorter period. If the sampling rate is high enough to make a single sample period overshoot imperceivable, or at least acceptable to the user, MMAE can greatly enhance head-mounted visual displays by reducing lag. The lag from an MMAE varies. It is unclear how variable lag, overshoot and oscillation affect the realism of virtual displays or how well training using such displays will transfer to the actual task.

5.2 Recommendations

This thesis has explored the advantages and disadvantages of MMAE used to reduce lag in a virtual environment flight simulator. Significant research remains to be done. Based on the lessons learned from this thesis work, the following recommendations are made:

1. Use a separate estimator for each axis. The problem of using a single estimator for two axes was demonstrated in the previous section, 5.1. To overcome this problem, a separate MMAE might be used for each axis. The MMAE for each axis will require slightly less computation since it is only taking measurements and making predictions for one axis. The total computation would be greater since two full MMAE's with separate filter banks will be needed. Ideally, these MMAE's will run in parallel, making the separate axis structure run faster than the single MMAE structure used in this research. This parallel structure will also simplify the addition of a roll axis to account for head tilt.

2. Test subjects to find the effect of overshoot, oscillation, and lag. If these effects can be quantified, an MMAE can be tuned to give the best overall result. This will guide designers in the future as they are faced with making trade-offs between these errors.

3. Possibly try α-β and α-β-γ filters [6] to reduce computational loading further. The α-β and α-β-γ filters are simpler than the FOGMV and FOGMA filters, respectively, thus reducing the on-line computational loading. However, they will also give lower performance than the the FOGMV and FOGMA filters. This trade-off should be

investigated.

The α-β-γ filter uses a precomputed filter gain K and the same dynamics as the FOGMA filter in the limit as τ goes to infinity. Said another way, the filter assumes that constant acceleration paths provide good descriptions of the look-angles being predicted. It also assumes complete independence of the spatial coordinates in the filter update equation. That is, heading is unrelated to pitch. The α-β-γ filter requires less on-line computation than the FOGMA filter at the cost of providing a less accurate model.

The α-β filter uses the same assumptions as the α-β-γ model, with the additional assumption that acceleration is zero. It employs essentially a constant velocity model, and is related to the FOGMV in the limit as $1/\tau$ goes to zero.

4. Measure acceleration data as well as position data about head motion. Being able to measure acceleration would greatly enhance filtering and prediction accuracy. In this research, acceleration could only be computed by differentiating position information twice, using 3 measurements taken over a 200 msec period. The final computation is an average acceleration over the 200 msec period. This may be a poor indication of acceleration over the next sample period. Further, differentiating tends to amplify noise. Taking two derivatives greatly amplifies noise and degrades accuracy of the acceleration calculation. Measuring acceleration directly from accelerometers mounted on a helmet or head mounted display will give an indication of instantaneous acceleration and would be free of the noise amplification caused by differentiation.

Having acceleration measurements available may change the models used in the predictive filters (although initially one would design filters based on the same dynamics models as developed in this research, but with augmented measurement models of both positions and accelerations). In any event, the MMAE will still be a viable technique to form the best prediction possible.

6. References

1. Adelstein, Bernard D., Eric R. Johnston, Stephen Ellis, "Spatial Sensor Lag in Virtual Environment Systems," *Proceedings of the SPIE*, volume 1833, 1993.

2. Arnold, J. E., "Design and Evaluation of a Predictor for Remote Control Synthesis with Signal Transmission." *Tech Report TN D-2229*, Washington, D. C: NASA, 1963.

3. Arthur, Kevin W., Kellogg, S. Booth, Colin Ware, Evaluating 3D Task Performance for Fish Tank Virtual Worlds," *ACM Transactions on Information Systems*, volume 11, issue 3, July 1993.

4. Boff, Kenneth R., Janet E. Lincoln (Editors*), Engineering Data Compendium: Human Perception and Performance*, Volume 3, Armstrong Aeromedical Research Laboratory, Wright-Patterson AFB, Ohio, 1988.

5. Bryson, Steve, and Scott S. Fisher, "Defining, Modeling and Measuring System Lag in Virtual Environments," Proceeding of the SPIE - The International Society for Optical Engineering, volume 1256, 1990.

6. Chang, Chaw-Bing, and John A. Tabaczynski, "Application of State Estimation to Target Tracking," *IEEE Transactions on Automatic Control*, volume AC-29, number 2, February 1984.

7. Crane, D. F., "Flight simulator visual display delay compensation," *Proceedings 1981 Winter Computer Simulation Conference*, San Diego, CA: Society for Computer Simulation, 1981.

8. D'Azzo, John, J. and Constantine Houpis, *Linear Control System Analysis and Design*, Third edition, McGraw-Hill Publishing Company, 1988.

9. Eggleston, Dr. Robert, Director, Human Interface Technology Branch, Armstrong Laboratory, Wright-Patterson AFB, OH. Personal Interview. 10 May 1995.

10. Friedman, Martin, Thad Starner, Alex Pentland, "Device Synchronization Using an Optimal Linear Filter," *Proceedings of the 1991 ACM Symposium on Interactive 3D Graphics*, 1992

11. *Graphics Library Programming Guide, Volume 1*, Document Number 007-1210-050, Silicon Graphics, Incorporated., 1992

12. Jagacinski, Richard J., "A Qualitative Look at Feedback Control Theory as a Style of Describing Behavior," Human Factors, volume 19, number 4, 1977.

13. Kozak, J. J., P. A. Hancock, E. J. Arthur, S. T. Chrysler, "Transfer of Training from Virtual Reality," *Ergonomics*, volume 36, number 7, 1993.

14. Liang, Jiandong, Chris Shaw, Mark Green, "On Temporal-Spatial Realism in the Virtual Reality Environment," *Proceedings of the ACM Symposium on User Interface Software and Technology*, 1991

15. Longinow, Nicholas E., "Predicting Pilot Look-Angle With a Radial Basis Function Network," *IEEE Transactions on Systems, Man and Cybernetics*, volume 24, no. 10, October, 1994.

16. Maybeck, Peter S., Professor of Electrical Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH. Course Notes EENG 768. Spring, 1995.

17. Maybeck, Peter S. *Stochastic Models, Estimation and Control. Volume 1.* New York: Academic Press, Inc., 1979.

18. Maybeck, Peter S. *Stochastic Models, Estimation and Control. Volume 2.* Arlington, VA, Navtech Book & Software Store, 1994.

19. Maybeck, Peter S., Professor, Air Force Institute of Technology, Wright-Patterson AFB, OH. Personal Interview, March 1995.

20 . Millner, Paul P., *Enhanced Tracking of Airborne Targets Using a Correlator/Kalman Filter.* MS Thesis, AFIT/GE/EE/82D-50. School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1982.

21. O'Connor, William, and Douglas Blake, unpublished notes and software for a Multiple Model Adaptive Estimator developed at the Air Force Institute of Technology, 1992.

22. Oman, Charles M., "Motion Sickness: A Synthesis and Evaluation of Sensory Conflict Theory," *Canadian Journal of Physiology and Pharmacology*, volume 68, 1990.

23. *PT-01 User's Manual*; O1 (a division of Optics 1, Incorporated); 3050 Hillcrest Drive, Suite 100, Westlake Village, CA.

24. Russell, James R., *Multiple Model Adaptive Estimation and Head Motion Tracking in a Virtual Environment: an Engineering Approach.* MS Thesis, AFIT/GCS/ENG/94D-21. School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1994.

25. So, R. H., M. J. Griffin, "Compensating lags in head-coupled displays using head position prediction and image deflection," *Journal of Aircraft*, 1992.

26. *3 Space User's Manual*, Document Number OPM3609-002B; Polhemus Incorporated, Colchester, Vermont, November 1992.

Vita

Capt David W. Kyger [redacted]. He graduated from Sandia High School in 1980 and accepted an appointment to the United States Air Force Academy. In 1983 he participated in a 6 month work-study program at the Air Force Weapons Laboratory, Kirtland AFB. Capt Kyger earned a Bachelor of Science in Electrical Engineering from the Air Force Academy with academic distinction in May 1985. After completing pilot training at Ft. Rucker, Alabama he was assigned to Holloman AFB to fly UH-1N Hueys. In September 1986 Captain Kyger was transferred to F. E. Warren AFB. He served in Desert Shield/Desert Storm as a Search and Rescue Duty Officer. In 1991 Captain Kyger transitioned to the HH-60 Pave Hawk and moved to Misawa AB, Japan. He entered the Air Force Institute of Technology in May 1994. Upon graduation he will be assigned to the Avionics Directorate at Wright Laboratories, Wright-Patterson AFB.

[redacted]

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
December 1995

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
Reducing Lag in Virtual Displays
Using Multiple Model Adaptive Estimation

5. FUNDING NUMBERS

6. AUTHOR(S)
David W. Kyger
Captain, USAF

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

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GE/ENG/95D-11

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Dr. Robert Eggleston
AL/CFHP
Wright-Patterson AFB, OH 45433

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Multiple Model Adaptive Estimation is an effective method for reducing lag in virtual environment displays. Lag in displays (the time from head motion to the appearance of the proper image on the display) is a significant detriment to realism in virtual envirnoments. Increasing the speed of the computers which control the virtual display is not a final answer. No matter how fast the processors work, there will always be demands to do more. Predicting angular head positions (look-angles) can reduce the lag by allowing the computer to calculate the appropriate scene before it is needed on the display. Single predictors cannot adequately cover the dynamic range of head motion. Using a bank of three elemental filters, lag in head orientation can be significantly reduced when compared to systems with no predictor or the single, nonadaptive Kalman filter predictor proposed by Liang. Predictions lead to small overshoots when the angular velocity of the head reverses direction, but the overshoot lasts for only one frame and is not significantly larger than the nonadaptive predictor.

14. SUBJECT TERMS
Multiple Model Adaptive Estimation, Lag, Virtual, Filtering, Kalman

15. NUMBER OF PAGES
88

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
UL

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	Contract	**PR**	Project
G	Grant	**TA**	Task
PE	Program Element	**WU**	Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD -	See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE -	See authorities.
NASA -	See Handbook NHB 2200.2.
NTIS -	Leave blank.

Block 12b. Distribution Code.

DOD -	Leave blank.
DOE -	Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA -	Leave blank.
NTIS -	Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

★U.S.GPO:1993-0-336-043