TRACKING MANEUVERING TARGETS VIA SEMI-MARKOV MANEUVER MODELING

by

Norman H. Gholson

Dissertation submitted to the Graduate Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

ELECTRICAL ENGINEERING

APPROVED:

Richard L Moose

R. L. Moose, Chairman

H. F. VanLandingham

R. A. Thompson

T. E. Bechert

J. A. Nachlas

February, 1977

Blacksburg, Virginia

LD
5655
V856
1977
G47
c. 2

MHS 4-7-77

ACKNOWLEDGEMENTS

The author wishes to express thanks to the many people who assisted him in the research and writing of this dissertation. Without their support, this effort might not have been completed.

Perhaps the most appreciation should be directed to Dr. Richard L. Moose, major professor and close friend. His guidance through the past three years of graduate work has helped insure a productive effort for my sponsors, as well as for myself. He was always available for discussion and quick to offer helpful suggestions and moral support.

A special thanks goes to other members of my committee whose main contribution has been excellent instruction in many fine graduate courses. Drs. H. F. VanLandingham and T. E. Bechert provided invaluable background in modern and classical control theory. Dr. J. A. Nachlas' expertise in system analysis provided powerful tools for designing stochastic models. A note of appreciation must also go to Dr. R. A. Thompson who allowed the author to present his research at an Electrical Engineering Graduate Seminar.

The Naval Surface Weapons Center and Dr. W. A. Blackwell deserve much credit for their unselfish efforts in providing financial support.

Finally, a big thanks to my wife Elizabeth who provided moral support and endured the trying circumstances involved in pursuing an advanced engineering degree.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.0 INTRODUCTION

The problem of tracking maneuvering targets via radar data has been a topic of concern for many researchers. The problem is quite simply defined in that one seeks to estimate the target's state (position, velocity, etc.) in such a manner to minimize some error cost function. A solution, however, is quite difficult and can best be summarized in two parts. Initially, a mathematical model must be developed to describe the target's dynamic behavior and statistical properties of the target's maneuvers. Secondly, an efficient algorithm must be designed to incorporate the modeling along with available measurements to obtain some optimal estimate of the target's state. Throughout the development, computational efficiency must be respected in order to obtain an implementable design. Utility of such work can readily be seen in areas such as automatic gun fire control systems.

The advent of high speed digital computers has made it possible to implement much more sophisticated algorithms than was previously possible. Microprocessor technology is becoming extremely attractive for algorithms featuring the possibility of bulk parallel computation. Even with this wealth of modern technology, computational burden can often be a limiting factor. Because of this, strong emphasis is placed on recursive type solutions with fixed memory requirements.

Linear mean square estimation was first considered and solved by Wiener and independently at about the same time (1940) by Kolmogoroff[1,2]. Their investigations addressed primarily the steady state continuous time estimation problem.

In 1960 the famous work of Kalman and Bucy led to the familiar Kalman filter[3]. This work revolutionized the estimation field by providing an optimal solution to a class of problems described by linear systems and Gaussian random processes.

Singer applied the Kalman filter to maneuvering target tracking by modeling target accelleration as a Gauss-Markov process[4]. Clark designed a very successful estimator for gun fire control systems using Singer's technique and the concept of a "dual bandwidth" Kalman filter[5,6].

Other work in this area includes Jazwiniski's limited memory filtering whereby the filter gains were maintained sufficiently large to allow accurate tracking[7]. A technique of switching between two filters upon detecting a maneuver is described by Thorp[8]. Regression techniques have been applied to target tracking by Springarn and Weideman[9].

This dissertation describes the development of an adaptive estimator based on modeling target maneuvers as a semi-Markov process. This process was analyzed by Howard[10] and first explored for tracking use by Moose[11,12].

In order to attack the development of a new adaptive algorithm, some fundamental background is needed. Chapter 2 briefly deals with analysis of semi-Markov maneuver modeling and the basic Kalman filter structure.

Chapter 3 developes the recursive semi-Markov adaptive algorithm. This algorithm augments the conventional Kalman filter to yield an adaptive state estimator.

Chapter 4 discusses target dynamics and explores two techniques of applying conventional (non adaptive) Kalman filtering to the target

tracking problem.

Implementation of the discrete semi-Markov adaptive algorithm is presented and evaluated in Chapter 5. Results are shown illustrating the technique's powerful ability to recognize maneuver commands influencing a target when the commands can be described by a finite set of discrete levels.

Chapter 6 describes the development of a continuous semi-Markov algorithm. Performance is analyzed by comparisons with the discrete semi-Markov algorithm and Singer's correlated acceleration algorithm.

Finally, Chapter 7 contains a summary of the methods and some suggestions for further work.

The effort described by this dissertation concluded that approximate spherical target dynamic modeling provides an adequate filter model for adaptive and non-adaptive estimation. Due to computational burden, it was concluded the spherical technique is actually superior to the more exact rectangular technique.

The continuous semi-Markov algorithm (described in Chapter 6.0) was concluded to be the more versatile adaptive estimator for target tracking. This conclusion was made based on the previous (discrete semi-Markov algorithm) "oscillation" problem being virtually eliminated by the continuous technique.

2.0 FUNDAMENTAL BACKGROUND

2.1 Introduction

In order to effectively modify the Kalman filter algorithm for incorporating semi-Markov maneuver modeling, it is necessary to understand the basic mechanics of the model and filter. This chapter reviews both in sufficient detail to fortify the adaptive filter derivation presented in Chapter 3.

2.2 Semi-Markov Maneuver Modeling

Semi-Markov maneuver modeling provides a probabilistic description of maneuver dynamics. This model is based on two very intuitively appealing assumptions.

(1) The maneuver command vector, $\underline{u}$, is randomly selected from a set of N maneuver vectors $\underline{u}^{(i)}$, i = 1,2,---, N. This selection process is described by the matrix P of N^2 elements p_{ij} where $p_{ij} \triangleq$ prob { next maneuver command is $\underline{u}^{(j)}$ | last maneuver command was $\underline{u}^{(i)}$}.

(2) A maneuver command is sustained for a random holding time. The random holding time, τ_{ij}, (described by density $h_{ij}(\tau)$) is the time sustained by maneuver command $\underline{u}^{(i)}$ given the next maneuver command is $\underline{u}^{(j)}$. Therefore, $h_{ij}(\tau)$ is the conditional probability density function of time sustained by maneuver command $\underline{u}^{(i)}$ given the next transition is to maneuver command $u^{(j)}$.

The objective of this section is to compute the matrix $\Theta(t)$ of N^2 elements $\theta_{ij}(t) \triangleq$ prob $\{E_{ij}(t)\}$, where $E_{ij}(t)$ is defined as the event {maneuver command is $\underline{u}^{(j)}$ at time $t + t_o$ | maneuver command $\underline{u}^{(i)}$ was

selected at time t_o}. The process is assumed stationary, i.e., $\theta(t)$ is independent of t_o. An implicit expression for $\theta(t)$ can be formed from the following reasoning. Event $E_{ij}(t)$ can occur by first a direct transition from $\underline{u}^{(i)}$ to $\underline{u}^{(k)}$ at time τ, and second, a transition of any type from $\underline{u}^{(k)}$ to $\underline{u}^{(j)}$ (and staying at $\underline{u}^{(j)}$) during the remaining time $t-\tau$. Here a "direct" transition is defined as a transition involving no intermediate maneuver commands. Figure 1 illustrates the proposed path from $\underline{u}^{(i)}$ to $\underline{u}^{(j)}$. Notice the second part of the above path is equivalent to event $E_{kj}(t-\tau)$. Since the transitions from $\underline{u}^{(i)}$ to $u^{(k)}$ and $\underline{u}^{(k)}$ to $\underline{u}^{(j)}$ are statistically independent, the joint probability of event $\{E_{ij}(t)$, and via $\underline{u}^{(k)}$, and transition to $\underline{u}^{(k)}$ occurs at time = $\tau\}$ can be expressed as

$$\text{prob}\ \{E_{ij}(t),\ k,\tau\} = \text{prob}\ \{\underline{u}^{(i)}\ \text{to}\ \underline{u}^{(k)}\ \text{and at time} = \tau\} \cdot \text{prob}\{E_{kj}(t-\tau)\}. \tag{2-1}$$

By the definitions of p_{ik}, h_{ik} and θ_{kj}, equation (2-1) can be expressed as

$$\text{prob}\ \{E_{ij}(t),\ k,\ \tau\} = p_{ik} \cdot h_{ik}(\tau) \cdot \theta_{kj}(t-\tau)d\tau. \tag{2-2}$$

Integrating equation (2-2) over time and summing over k results in the marginal probability

$$\text{prob}\ \{E_{ij}\ (t)\} = \int_o^t \sum_{k=1}^{N} p_{ik} \cdot h_{ik}(\tau)d\tau \cdot \theta_{kj}(t-\tau).$$

Rearranging and substituting $\theta_{ij}(t) \triangleq \text{prob}\ \{E_{ij}(t)\}$ yields the implicit formulation for $\theta(t)$

$$\theta_{ij}(t) = \sum_{k=1}^{N} p_{ik} \int_o^t \theta_{kj}(t-\tau)\ h_{ik}(\tau)d\tau\ . \tag{2-3}$$

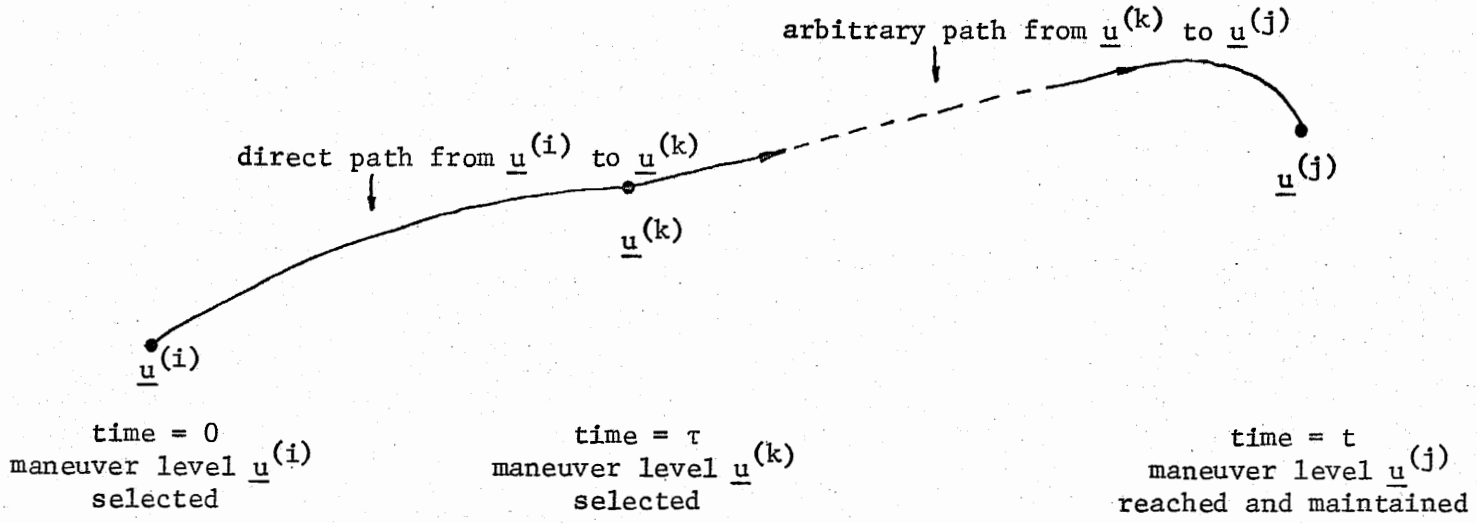


FIGURE 1 Path From $\underline{u}^{(i)}$ to $\underline{u}^{(j)}$

Expression 2-3 is complete except for the case of j=i. Notice event E_{ii} may occur if the random holding time τ_i is greater than t, as well as by the chain of events described above. A joint probability for $\tau_i > t$ can be obtained from the conditional probability

$$\text{prob } \{\tau_i > t \mid \text{next command is } \underline{u}^{(\ell)}\} = \int_t^\infty h_{i\ell}(\tau)d\tau$$

by applying Bayes' rule to yield

$$\text{prob } \{\tau_i > t, \text{ next command is } \underline{u}^{(\ell)}\} = \int_t^\infty h_{i\ell}(\tau)d\tau \; p_{i\ell} \tag{2-4}$$

Summing the joint probability of equation 2-4 over ℓ results in the marginal probability

$$\text{prob } \{\tau_i > t\} = \sum_{\ell=1}^{N} p_{i\ell} \int_t^\infty h_{i\ell}(\tau)d\tau \; . \tag{2-5}$$

Since the event just discussed and the previous chain of events are mutually exclusive, equations (2-3) and (2-5) may be added to obtain the implicit expression

$$\theta_{ij}(t) = \left[\sum_{\ell=1}^{N} p_{i\ell} \int_t^\infty h_{i\ell}(\tau)d\tau\right] \delta_{ij} + \sum_{k=1}^{N} p_{ik} \int_0^t \theta_{kj}(t-\tau)h_{ik}(\tau)d\tau \; . \tag{2-6}$$

where δ_{ij}, the Kroneker delta function, insures the first term contributes only when j=i. Examining equation (2-6) we see that it is an integral equation in $\Theta(t)$. However, it is easily solvable by noting the last term represents a convolution in the time domain which becomes a product in the complex frequency domain. Hence, we have upon taking the Laplace transform

$$\theta_{ij}(s) = w_i(s) \; \delta_{ij} + \sum_{k=1}^{N} p_{ik} \; h_{ik}(s) \; \theta_{kj}(s) \tag{2-7}$$

where $w_i(s)$ = Laplace transform of $\sum_{\ell=1}^{N} p_{i\ell} \int_t^\infty h_{i\ell}(\tau)d\tau$.

Howard[10] defines a matrix "box" product by C=A * B where the elements of C are related to the elements of A and B by $c_{ij}=a_{ij}\ b_{ij}$. Applying the box notation, equation (2-7) can be expressed in matrix form by

$$\Theta(s) = W(s) + (P * H)\,\Theta(s) \tag{2-8}$$

where H(s) is a matrix of N^2 elements $h_{ij}(s)$ and W(s) is a diagonal matrix of elements $w_i(s)$ i=1,2,---,N. Solving equation (2-8) for $\Theta(s)$ results in the transform pair

$$\begin{aligned} \Theta(s) &= \left[I - P * H(s)\right]^{-1} W(s) \\ \Theta(t) &= L^{-1}\{\Theta(s)\} \end{aligned} \tag{2-9}$$

where I is an NxN identity matrix.

2.2.1 Exponential Holding Times

The case of exponentially distributed random holding times is of particular interest in the adaptive estimator design. For this reason, a closed form expression for $\Theta(t)$ will be derived in this section. Assuming the random holding time density functions are given by $h_{ij}(t) = ae^{-at}$ allows the matrices H(s) and W(s) to be easily calculated yielding

$$\begin{aligned} h_{ij}(s) &= \frac{a}{s+a} \\ W(s) &= \frac{1}{s+a}\ I \end{aligned} \tag{2-10}$$

Substituting W(s) and H(s) into equation (2-9) and arranging terms results in the familiar expression

$$\Theta(s) = \left[sI - a(P-I)\right]^{-1} \tag{2-10a}$$

of inverse transform

$$\Theta(t) = \varepsilon^{a(P-I)t} \tag{2-10b}$$

For the case of N fairly small, $\Theta(t)$ can be evaluated directly by inverting the Laplace transform of equation (2-10a). Also, for small N, $\Theta(t)$ can be computed from equation (2-10b) using matrix diagonalization procedures discussed in Ogata[13]. For large N, numerical techniques such as presented by Healey[14] are convenient. As an example, $\Theta(t=2$ sec.) was computed for two random holding time density functions using the P matrix shown below.

$$P = \begin{bmatrix} .6364 & .2341 & .0861 & .0317 & .0117 \\ .1915 & .5306 & .1915 & .0765 & .0259 \\ .0675 & .1834 & .4984 & .1834 & .0675 \\ .0259 & .0705 & .1915 & .5206 & .1915 \\ .0117 & .0317 & .0861 & .2341 & .6364 \end{bmatrix}$$

Using the power series expansion technique, Θ was computed to be

$$\Theta(t=2\text{sec}, \bar{\tau}=50\text{sec}) = \begin{bmatrix} .9856 & .0092 & .0034 & .0013 & .0005 \\ .0075 & .9811 & .0075 & .0028 & .0010 \\ .0027 & .0072 & .9802 & .0072 & .0027 \\ .0010 & .0028 & .0075 & .9811 & .0075 \\ .0005 & .0013 & .0034 & .0092 & .9856 \end{bmatrix}$$

for random holding time density function $.02\varepsilon^{-.02t}$ (mean random holding time of 50sec). Changing the mean holding time to 5sec results in the following Θ matrix.

$$\Theta(t=2\text{sec},\bar{\tau}=5\text{sec}) = \begin{bmatrix} .8682 & .0806 & .0326 & .0132 & .0054 \\ .0659 & .8313 & .0654 & .0266 & .0108 \\ .0255 & .0626 & .8238 & .0626 & .6255 \\ .0108 & .0266 & .0654 & .8313 & .0659 \\ .0054 & .0132 & .0326 & .0806 & .8683 \end{bmatrix}$$

Notice that reducing the mean holding time tends to reduce θ_{ii} while increasing $\theta_{ij}(i\neq j)$.

2.2.2 A More General Form of the Conditional Probabilities

The particular form of the conditional probabilities $\theta_{ij}(t)$ differs from that necessary for implementation in the adaptive estimator. This will be discussed later in more detail while presently an effort will be directed toward obtaining the matrix $\Theta^*(t)$ of N^2 elements $\theta^*_{ij} \triangleq$ prob $\{\underline{u} = \underline{u}^{(j)}$ at time = $t + t_o \mid$ given $\underline{u} = \underline{u}^{(i)}$ at time = $t_o\}$. Notice that $\Theta^*(t)$ differs from $\Theta(t)$ in that $\Theta^*(t)$ drops the requirement of maneuver command $\underline{u}^{(i)}$ being <u>selected</u> at time = t_o. An explicit equation for $\Theta^*(t)$ can be formulated to yield

$$\theta^*_{ij}(t) = \left[\sum_{\ell=1}^{N} p^*_{i\ell} \int_k^\infty h^*_{i\ell}(\tau)d\tau\right]\delta_{ij} + \sum_{k=1}^{N} p^*_{ik} \int_o^t \theta_{kj}(t-\tau)h^*_{ik}(\tau)d\tau \quad (2\text{-}11)$$

Moose[15] shows that p_{ij}^* and h_{ij}^* are given by

$$p^*_{ij} = \frac{\bar{\tau}_{ij}p_{ij}}{\sum_{j=1}^{N} p_{ij}\bar{\tau}_{ij}} \quad (2\text{-}12)$$

$$h^*_{ij}(\tau) = \frac{1 - \int_o^\tau h_{ij}(t)dt}{\bar{\tau}_{ij}}$$

where $\bar{\tau}_{ij} = \int_0^\infty \tau\, h_{ij}(\tau)d\tau$.

Substituting $h_{ij}(t) = a\varepsilon^{-at}$ into equations (2-12) results in $p_{ij}^* = p_{ij}$ and $h_{ij}^* = h_{ij}$. Therefore, for the case of identical exponentially distributed holding times, $\theta_{ij}^*(t) = \theta_{ij}(t)$. The importance of this result will become apparent in the adaptive estimator derivation.

2.3 Basic Structure of the Kalman Filter

Conventional Kalman filtering algorithms are designed about the following familiar set of linear difference equations describing the plant (target) dynamics and measurement process.

$$\begin{aligned} \underline{x}_{k+1} &= \phi_k \underline{x}_k + \Gamma_k \underline{w}_k \\ \underline{z}_{k+1} &= H_{k+1} \underline{x}_{k+1} + \underline{v}_{k+1} \end{aligned} \qquad (2\text{-}13)$$

where $\underline{x}$ = n x 1 system state vector

ϕ = n x n system state transition matrix

Γ = n x p input matrix

$\underline{w}$ = p x 1 zero mean white Gaussian disturbance vector

$\underline{z}$ = r x 1 observation vector

H = r x n matrix describing linear relation of observation to system state

$\underline{v}$ = r x 1 vector of zero mean white Gaussian observation errors (independent of $\underline{w}$)

The Kalman filter processes measurements to obtain an estimate ($\hat{\underline{x}}$) of the system state vector by exercising the filter equation shown below.

$$\hat{\underline{x}}_{k+1} = \phi_k \hat{\underline{x}}_k + K_{k+1}(\underline{z}_{k+1} - H_{k+1}\, \phi_k\, \hat{\underline{x}}_k) \qquad (2\text{-}14)$$

The matrix K_{k+1} is the Kalman gain matrix which is computed from the following set of recursive equations.

$$P_{k+1|k} = \phi_k \, P_{k|k} \, \phi_k^T + \Gamma_k \, Q_k \, \Gamma_k^T$$

$$K_{k+1} = P_{k+1|k} H_{k+1}^T \left(H_{k+1} \, P_{k+1|k} \, H_{k+1}^T + R_{k+1}\right)^{-1} \qquad (2\text{-}15)$$

$$P_{k+1|k+1} = \left[I - K_{k+1} \, H_{k+1}\right] P_{k+1|k}$$

where $\quad Q_k \triangleq E[\underline{w}_k \, \underline{w}_k^T]$

$$R_k \triangleq E[\underline{v}_k \, \underline{v}_k^T]$$

$$E[\cdot] \triangleq \text{expected value}$$

Matrix $P_{k+1|k+1}$ is the error covariance matrix defined by

$$P_{k+1|k+1} = E[(\hat{\underline{x}}_{k+1} - \underline{x}_{k+1})(\hat{\underline{x}}_{k+1} - \underline{x}_{k+1})^T] \quad .$$

Implementing the Kalman filter yields the minimum variance state estimate given by the conditional expectation $\hat{\underline{x}}_{k+1} = E[\underline{x}_{k+1} | \hat{\underline{x}}_k, \underline{z}_{k+1}]$. Figure 2 illustrates the Kalman filter structure by a block diagram of the filter equation.

2.3.1 Application to Maneuvering Target Tracking

In general, the vector $\underline{w}$ of equation (2-13) represents white Gaussian disturbances influencing a plant. For a maneuvering target, $\underline{w}$ represents small perturbations (such as turbulence, etc.) influencing the target trajectory. Clearly this vector does not account for large scale effects such as evasive or even normal target maneuvers. For this reason, a more realistic target model can be formulated by including a deterministic input vector representing large scale pilot commands. The term "deterministic" is used to indicate the command is known by the

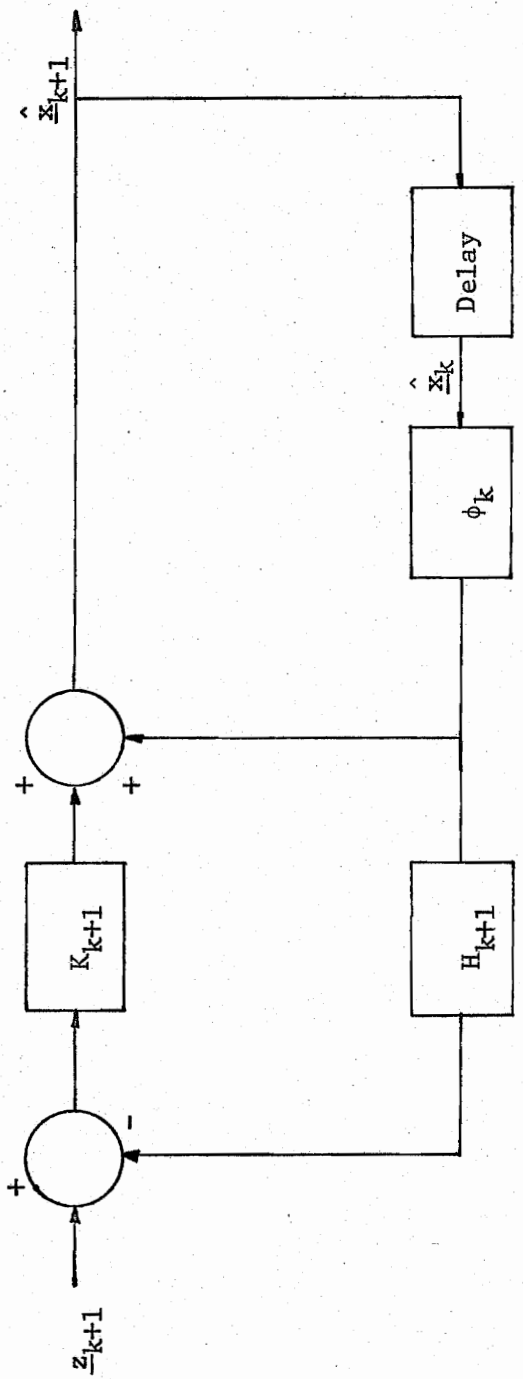


FIGURE 2 Block Diagram of Filter Equation

pilot or guidance system. Such a model is shown below where the deterministic input at time t_k is represented by $\underline{u}_k$.

$$\underline{x}_{k+1} = \phi_k \underline{x}_k + \Gamma_k (\underline{w}_k + \underline{u}_k) \qquad (2\text{-}16)$$

A simple technique of accounting for the presence of $\underline{u}$ is to design the Kalman filter assuming the covariance of $\underline{w}$ is much larger than its actual value. This has the effect of increasing Kalman gain or widening the filter "bandwidth" but has the shortcoming of modeling target maneuvers as a white Gaussian process. A casual glance at typical maneuvering target trajectories reveals this to be a poor maneuver model. The adaptive estimator described in this dissertation recursively computes an estimate of $\underline{u}$ based on semi-Markov maneuver modeling. Derivation of the recursive algorithm is presented in the next chapter.

3.0 ADAPTIVE ESTIMATOR DERIVATION

3.1 Introduction

The adaptive estimator described in this dissertation consists of a Kalman filter augmented by a recursive algorithm for estimating the maneuver command vector. The command vector, $\underline{u}$, at time t_k influences the target state vector, $\underline{x}$, and measurement vector, $\underline{z}$, at time t_{k+1} in the following manner.

$$\begin{aligned} \underline{x}_{k+1} &= \phi_k \underline{x}_k + \Gamma_k(\underline{u}_k + \underline{w}_k) \\ \underline{z}_{k+1} &= H_{k+1}\underline{x}_{k+1} + \underline{v}_{k+1} \end{aligned} \tag{3-1}$$

As discussed earlier, we assume the command vector $\underline{u}$ is randomly selected from a set of N vectors $\underline{u}^{(i)}$, i=1, 2...N in a semi-Markov manner. The recursive algorithm estimates $\underline{u}$ by computing the expected value of $\underline{u}$ given semi-Markov statistical knowledge and all available measurement data. The purpose of this chapter is to derive the recursive algorithm. Much of the presentation is taken directly from Moose[12].

3.2 Bank of Kalman Filters

In deriving the adaptive estimator, the discrete form of conditional expectation is used:

$$\hat{\underline{x}}_{k+1} \triangleq E[\underline{x}_{k+1} | \hat{\underline{x}}_k, \underline{z}_{k+1}] = \sum_{\underline{x}} \underline{x} \, \text{prob}(\underline{x}_{k+1} = \underline{x} | \hat{\underline{x}}_k, \underline{z}_{k+1}) \tag{3-2}$$

Defining $\text{prob}(\underline{u}_k = \underline{u}^{(i)} | \hat{\underline{x}}_k, \underline{z}_{k+1})$ as the probability of $\underline{u}$ at time t_k being $\underline{u}^{(i)}$ given $\hat{\underline{x}}_k$ and $\underline{z}_{k+1}$, allows $\text{prob}(\underline{x}_{k+1} = \underline{x} | \hat{\underline{x}}_k, \underline{z}_{k+1})$ to be expressed as

$$\begin{aligned} \text{prob}(\underline{x}_{k+1} = \underline{x} | \hat{\underline{x}}_k, \underline{z}_{k+1}) = \sum_{i=1}^{N} & \text{prob}(\underline{x}_{k+1} = \underline{x} | \hat{\underline{x}}_k, \underline{z}_{k+1}, \underline{u}_k = \underline{u}^{(i)}) \cdot \\ & \text{prob}(\underline{u}_k = u^{(i)} | \hat{\underline{x}}_k, \underline{z}_{k+1}) \end{aligned} \tag{3-3}$$

Combining equations (3-2) and (3-3) yields, upon interchanging the order of summation,

$$\underline{\hat{x}}_{k+1} = \sum_{i=1}^{N} \underline{\hat{x}}_{k+1}^{(i)} \text{ prob}(\underline{u}_k=\underline{u}^{(i)} | \underline{\hat{x}}_k, \underline{z}_{k+1}) \qquad (3\text{-}4)$$

where

$$\underline{\hat{x}}_{k+1}^{(i)} = \sum_{\underline{x}} \underline{x} \text{ prob}(\underline{x}_{k+1}=\underline{x} | \underline{\hat{x}}_k, \underline{z}_{k+1}, \underline{u}_k=\underline{u}^{(i)}) \equiv E[\underline{x}_{k+1} | \underline{\hat{x}}_k, \underline{z}_{k+1}, \underline{u}_k=\underline{u}^{(i)}] \qquad (3\text{-}5)$$

Equation (3-5) represents a Kalman filter conditioned on $\underline{u}=\underline{u}^{(i)}$ at time t_k. Therefore, equation (3-4) represents the weighed estimate of a bank of N Kalman filters. The N weighting probabilities $\text{prob}(\underline{u}_k=\underline{u}^{(i)} | \underline{\hat{x}}_k, \underline{z}_{k+1})$, i=1,2,...,N will be determined in the next section.

3.3 Weighting Probabilities

The weighting probabilities can be expanded by Bayes' rule to yield

$$\text{prob}(\underline{u}_k=\underline{u}^{(i)} | \underline{\hat{x}}_k, \underline{z}_{k+1}) = \frac{p(\underline{z}_{k+1} | \underline{u}_k=\underline{u}^{(i)}, \underline{\hat{x}}_k) \text{prob}(\underline{u}_k=\underline{u}^{(i)} | \underline{\hat{x}}_k)}{p(\underline{z}_{k+1} | \underline{\hat{x}}_k)} \qquad (3\text{-}6)$$

It should be pointed out that the denominator term is common for all weighting probabilities and can be replaced by D_k. The term D_k can easily be computed from $\sum_{i=1}^{N} \text{prob}(\underline{u}_k=\underline{u}^{(i)} | \underline{\hat{x}}_k, \underline{z}_{k+1}) = 1$. The first numerator term, $p(\underline{z}_{k+1} | \underline{u}_k=\underline{u}^{(i)}, \underline{\hat{x}}_k)$, is approximately normally distributed and will be represented by a Gaussian density function of mean and covariance established from the Kalman filtering algorithm conditioned on $\underline{u}_k=\underline{u}^{(i)}$, i. e.

$$N\{[H_{k+1}\phi_k\underline{\hat{x}}_k + H_{k+1}\Gamma_k\underline{u}^{(i)}]; C_{k+1}\}$$

where $C_{k+1} = H_{k+1}[\phi_k P_{k/k} \phi_k^T + \Gamma_k Q_k \Gamma_k^T] H_{k+1}^T + R_{k+1}$.

The matrix C_{k+1} is a byproduct of computing the Kalman gain (see equation (2-15)). The probability $\text{prob}(\underline{u}_k = \underline{u}^{(i)} | \underline{\hat{x}}_k)$ of equation (3-6) is the one

step predicted probability of $\underline{u}$ obtained from semi-Markov modeling. Expanding $\text{prob}(\underline{u}_k = \underline{u}^{(i)} | \hat{\underline{x}}_k)$ provides

$$\text{prob}(\underline{u}_k = \underline{u}^{(i)} | \hat{\underline{x}}_k) = \sum_{\alpha=1}^{N} \text{prob}(\underline{u}_k = \underline{u}^{(i)} \Big| \underline{u}_{k-1} = \underline{u}^{(\alpha)}, \hat{\underline{x}}_k) \cdot$$

$$\text{prob}(\underline{u}_{k-1} = \underline{u}^{(\alpha)} | \hat{\underline{x}}_{k-1}, \underline{z}_k) \qquad (3\text{-}7)$$

where the information $(\hat{\underline{x}}_{k-1}, \underline{z}_k)$ is equivalent to $(\hat{\underline{x}}_k)$. The first probability of the summation is conditioned on $\hat{\underline{x}}_k$ as well as $\underline{u}_{k-1} = \underline{u}^{(\alpha)}$. From equation (3-1) we see that $\underline{u}_k$ does not influence $\underline{x}_k$ ($\underline{u}_k$ first influences $\underline{x}_{k+1}$) and therefore does not influence $\hat{\underline{x}}_k$. Because of this, conditioning the probability on $\underline{u}_{k-1} = \underline{u}^{(\alpha)}$ and $\hat{\underline{x}}_k$ adds little information over conditioning on just $\underline{u}_{k-1} = \underline{u}^{(\alpha)}$. Therefore, it is reasonable to make the approximation

$$\text{prob}(\underline{u}_k = \underline{u}^{(i)} | \underline{u}_{k-1} = \underline{u}^{(\alpha)}, \hat{\underline{x}}_k) \simeq \text{prob}(\underline{u}_k = \underline{u}^{(i)} | \underline{u}_{k-1} = \underline{u}^{(\alpha)}). \qquad (3\text{-}8)$$

The second term of the summation (equation (3-7)) is known from the previous recursive calculation. Combining equations (3-8), (3-7) and (3-6) yields for the time varying weighting probabilities,

$$\text{prob}(\underline{u}_k = \underline{u}^{(i)} | \hat{\underline{x}}_k, \underline{z}_{k+1}) = \left(\frac{1}{D_k}\right) p(\underline{z}_{k+1} | \underline{u}_k = \underline{u}^{(i)}, \hat{\underline{x}}_k) \sum_{\alpha=1}^{N} \text{prob}(\underline{u}_k = \underline{u}^{(i)} | \underline{u}_{k-1} =$$

$$\underline{u}^{(\alpha)}) \ \text{prob}(\underline{u}_{k-1} = \underline{u}^{(\alpha)} | \hat{\underline{x}}_{k-1}, \underline{z}_k). \qquad (3\text{-}9)$$

The only undefined term in equation (3-9) is $\text{prob}(\underline{u}_k = \underline{u}^{(i)} | \underline{u}_{k-1} = \underline{u}^{(\alpha)})$. Formally expressing this term as $\text{prob}(\underline{u} = \underline{u}^{(i)}$ at time t_k given $\underline{u} = u^{(\alpha)}$ at time t_{k-1}) reveals that $\text{prob}(\underline{u}_k = \underline{u}^{(i)} | \underline{u}_{k-1} = u^{(\alpha)}) = \theta^*_{\alpha i}(t_k - t_{k-1})$ where θ^* is formulated in Chapter II. Therefore, equation (3-9) can be expressed as

$$\text{prob}(\underline{u}_k = \underline{u}^{(i)} | \hat{\underline{x}}_k, \underline{z}_{k+1}) = \left(\frac{1}{D_k}\right) p(\underline{z}_{k+1} | \underline{u}_k = \underline{u}^{(i)}, \hat{\underline{x}}_k) \sum_{\alpha=1}^{N} \text{prob}(\underline{u}_{k-1} = \underline{u}^{(\alpha)} | \hat{\underline{x}}_{k-1},$$

$$\underline{z}_k) \cdot \theta^*_{\alpha i}(t_k - t_{k-1}). \qquad (3\text{-}10)$$

Using the weighting probabilities, an adaptive estimator can be formed from a bank of N Kalman filters conditioned on $\underline{u}_k = \underline{u}^{(i)}$, i = 1, 2, ...,N.

3.4 Reduction to A Single Filter

Equation (3-4) represents the weighted estimate of N Kalman filters. The i^{th} Kalman filter has the form

$$\hat{\underline{x}}^{(i)}_{k+1} = \phi^{(i)}_k \hat{\underline{x}}_k + \Gamma^{(i)}_k \underline{u}^{(i)} + K^{(i)}_{k+1}[\underline{z}_{k+1} - H^{(i)}_{k+1}\phi^{(i)}_k \hat{\underline{x}}_k - H^{(i)}_{k+1}\Gamma^{(i)}_k \underline{u}^{(i)}] \quad (3\text{-}11)$$

where superscript "i" denotes "conditioned on $\underline{u}_k = \underline{u}^{(i)}$". Consider the very practical case where the matrices ϕ, Γ, H, K are independent of $\underline{u}$. In this case, the i^{th} Kalman filter becomes (after rearranging terms)

$$\hat{\underline{x}}^{(i)}_{k+1} = \phi_k \hat{\underline{x}}_k + K_{k+1}[\underline{z}_{k+1} - H_{k+1}\phi_k \hat{\underline{x}}_k] + [\Gamma_k - K_{k+1}H_{k+1}\Gamma_k]\underline{u}^{(i)}. \quad (3\text{-}12)$$

Substituting equation (3-12) into equation (3-4) results in

$$\hat{\underline{x}}_{k+1} = \sum_{i=1}^{N} \text{prob}(\underline{u}_k = \underline{u}^{(i)} | \hat{\underline{x}}_k, \underline{z}_{k+1})[\phi_k \hat{\underline{x}}_k + K_{k+1}[\underline{z}_{k+1} - H_{k+1}\phi_k \hat{\underline{x}}_k]] +$$

$$\sum_{i=1}^{N} \text{prob}(\underline{u}_k = \underline{u}^{(i)} | \hat{\underline{x}}_k, \underline{z}_{k+1})[\Gamma_k - K_{k+1}H_{k+1}\Gamma_k]\underline{u}^{(i)}$$

which may be written in the form

$$\hat{\underline{x}}_{k+1} = \phi_k \hat{\underline{x}}_k + K_{k+1}[\underline{z}_{k+1} - H_{k+1}\phi_k \hat{\underline{x}}_k] + [\Gamma_k - K_{k+1}H_{k+1}\Gamma_k] \sum_{i=1}^{N} \underline{u}^{(i)} \text{prob}$$

$$(\underline{u}_k = \underline{u}^{(i)} | \hat{\underline{x}}_k, \underline{z}_{k+1}) \quad (3\text{-}13)$$

where use has been made of $\sum_{i=1}^{N} \text{prob}(\underline{u}_k = \underline{u}^{(i)} | \hat{\underline{x}}_k, \underline{z}_{k+1}) = 1$. Defining

$$\hat{\underline{u}}_k \triangleq \sum_{i=1}^{N} \underline{u}^{(i)} \text{prob}(\underline{u}_k = \underline{u}^{(i)} | \hat{\underline{x}}_k, \underline{z}_{k+1})$$

and substituting into equation (3-13) yields

$$\hat{\underline{x}}_{k+1} = \phi_k \hat{\underline{x}}_k + K_{k+1}[\underline{z}_{k+1} - H_{k+1}\phi_k \hat{\underline{x}}_k] + [\Gamma_k - K_{k+1}H_{k+1}\Gamma_k]\hat{\underline{u}}_k.$$

Rearranging terms results in the final form of the single filter adaptive estimator.

$$\hat{\underline{x}}_{k+1} = \phi_k \hat{\underline{x}}_k + \Gamma_k \hat{\underline{u}}_k + K_{k+1}[\underline{z}_{k+1} - H_{k+1}\phi_k \hat{\underline{x}}_k - H_{k+1}\Gamma_k \hat{\underline{u}}_k].$$

$$\hat{\underline{u}}_k = \sum_{i=1}^{N} \underline{u}^{(i)} \text{prob}[\underline{u}_k = \underline{u}^{(i)} | \hat{\underline{x}}_k, \underline{z}_{k+1}]. \qquad (3\text{-}14)$$

Equation (3-14) has the form of a Kalman filter augmented by an external learning feature estimating $\underline{u}$, the unknown maneuver command vector. This system is similar to that of Ackerson and Fu[16] with the primary difference being that the time varying weighting probabilities are now computed from a semi-Markov maneuver model.

4.0 FORMULATION OF THE NON-ADAPTIVE TARGET TRACKING FILTER

4.1 Introduction

The semi-Markov adaptive estimator is based on the conventional (non-adaptive) Kalman filter. Therefore, an investigation of non-adaptive filtering must be conducted before effectively implementing the adaptive algorithm. The purpose of this chapter is to develop and compare two techniques of non-adaptive design.

4.2 Target Dynamics

Accurate target dynamics are seldom known since the target is usually of enemy origin. Even if an accurate model were available, it would certainly be in the form of complex high order nonlinear differential equations. Such equations would, of course, be very difficult to implement.

Intuitively, a linear target model in rectangular coordinates is appealing despite the spherical form of radar data. Target dynamics assumed in this study are described in state space notation by

$$\dot{\underline{x}} = A\,\underline{x} + B(\underline{u} + \underline{w}) \tag{4-1}$$

where $\underline{x} = [x|\dot{x}|y|\dot{y}|z|\dot{z}]^T$

x,y,z = rectangular target coordinates

$\underline{u} = [u_x|u_y|u_z]^T$ (maneuver command vector of x,y,z components)

$\underline{w} = [w_x|w_y|w_z]^T$ zero mean white Gaussian disturbance process of x,y,z components

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\rho & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\rho & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -\rho \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The viscous drag coefficient, ρ, is generally chosen from 0 to 0.32. A ρ of 0.32 allows a maneuver command causing 10G acceleration from standstill to result in a terminal speed of 1000ft/sec.

Assuming the vectors $\underline{u}$ and $\underline{w}$ approximately constant over an iteration interval, equation (4-1) can be discretized to yield the following vector difference equation describing target dynamics.

$$\underline{x}_{k+1} = \phi \underline{x}_k + \Gamma(\underline{u}_k + \underline{w}_k) \qquad (4\text{-}2)$$

$$\phi = \begin{bmatrix} 1 & P_1 & 0 & 0 & 0 & 0 \\ 0 & P_1^* & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & P_1 & 0 & 0 \\ 0 & 0 & 0 & P_1^* & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & P_1 \\ 0 & 0 & 0 & 0 & 0 & P_1^* \end{bmatrix}$$

$$\Gamma = \begin{bmatrix} P_2 & 0 & 0 \\ P_1 & 0 & 0 \\ 0 & P_2 & 0 \\ 0 & P_1 & 0 \\ 0 & 0 & P_2 \\ 0 & 0 & P_1 \end{bmatrix}$$

$$P_1 = \frac{1 - \varepsilon^{-\rho T}}{\rho}$$

$$P_1^* = \varepsilon^{-\rho T}$$

$$P_2 = \frac{\varepsilon^{-\rho T} - 1 + \rho T}{\rho^2}$$

T = Time difference between time t_{k+1} and time t_k

The zero mean white Gaussian disturbance vector $\underline{w}_k$ is described by its covariance matrix Q, where Q is defined by

$$Q_k \triangleq E[\underline{w}_k \, \underline{w}_k^T] \quad .$$

In this study, we initially assume $\underline{w}$ consists of statisically independent x,y,z components. This choice is not based on any stringent physical considerations but merely for mathematical convenience. Under the above assumption, the matrix Q is given by

$$Q = \begin{bmatrix} \sigma^2_{w_x} & 0 & 0 \\ 0 & \sigma^2_{w_y} & 0 \\ 0 & 0 & \sigma^2_{w_z} \end{bmatrix}$$

where $\sigma^2_{w_x}$, $\sigma^2_{w_y}$ and $\sigma^2_{w_z}$ are variances of the x,y,z components respectively. Typical assumed variances are $\sigma^2_{w_x} = \sigma^2_{w_y} = \sigma^2_{w_z} = 1000 \text{ft}^2/\text{sec}^4$.

A sample nonmaneuvering trajectory (x-direction) generated from this discrete time model is shown in Figure 3. The trajectory was generated using the following conditions.

$$\underline{u} = \underline{0}$$

$$\rho = 0$$

$$\sigma^2_{w_x} = 1000\text{ft}^2/\text{sec}^4$$

$$x_o = 1000 \text{ ft}$$

$$T = 0.2 \text{ sec}$$

4.3 Measurement Process

The observation equation describes how the measurement process relates to the target's state vector. For convenience, the observation equation is repeated below.

$$\underline{z}_{k+1} = H_{k+1}\,\underline{x}_{k+1} + \underline{v}_{k+1}$$

where $\underline{z}$ = observation (measurement) vector

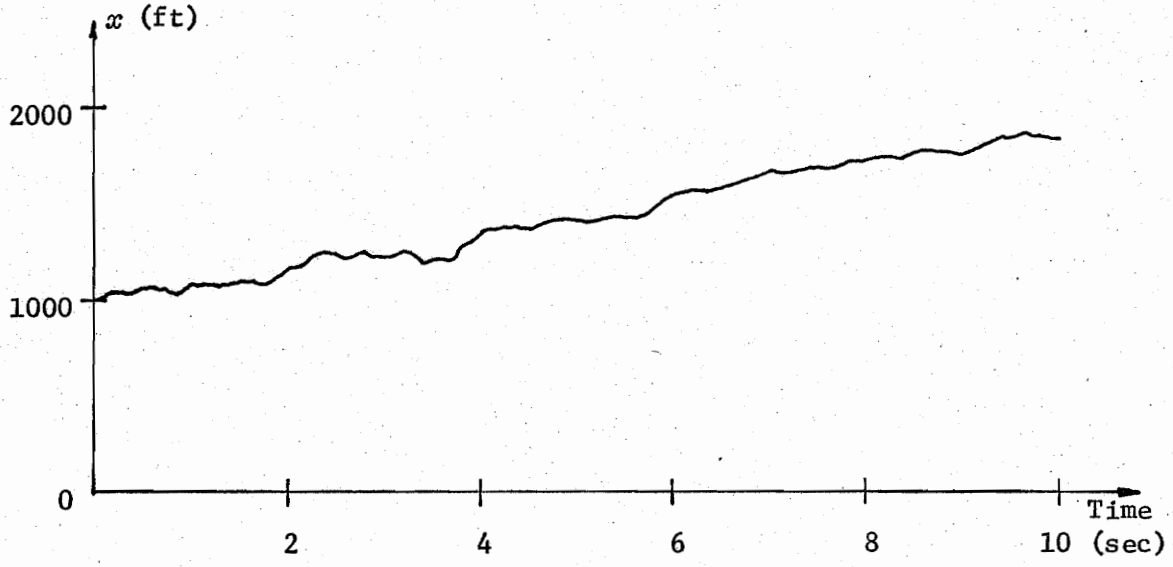
H = observation matrix relating $\underline{z}$ to the state vector

$\underline{x}$ = state vector

$\underline{v}$ = vector of white Gaussian measurement errors

For radar tracking, measurements consist of measured target range, elevation and bearing. For the rectangular dynamic model discussed in the previous section, $\underline{x} = [x|\dot{x}|y|\dot{y}|z|\dot{z}]^T$. Adopting the coordinate system shown in Figure 4 provides the following relationship between spherical and rectangular coordinates

$$\begin{aligned} x &= r\cos(e)\cos(b) \\ y &= r\cos(e)\sin(b) \\ z &= r\sin(e) \end{aligned} \qquad (4\text{-}3)$$



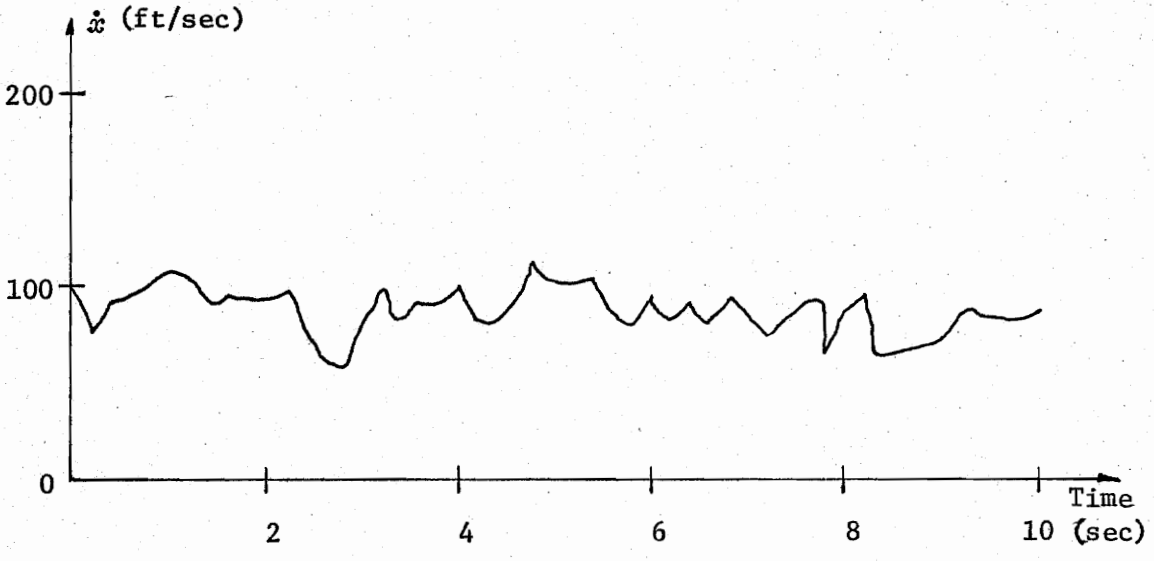


FIGURE 3 Sample Non-maneuvering Trajectory

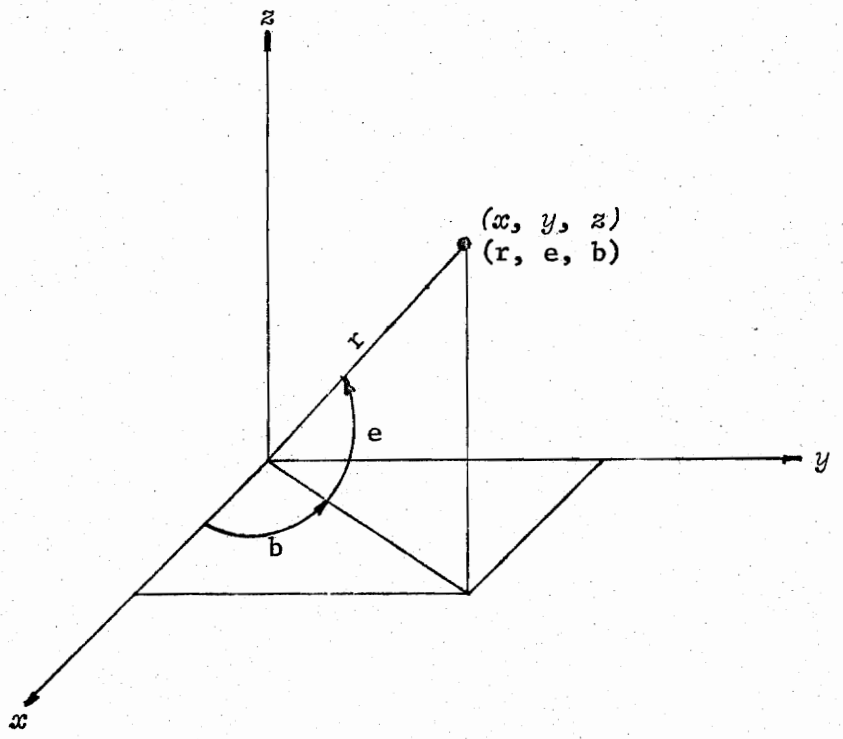


FIGURE 4 Coordinate Systems

where r,e,b represent target range, elevation and bearing respectively. Clearly, equation (4-3) indicates a nonlinear relationship between the rectangular target state and the spherical radar data. Because of this, conventional Kalman filtering cannot be directly applied. The following sections describe two techniques for overcoming this problem.

4.4 Data Linearization

The principle of data linearization is to generate an observation vector that is nearly a linear function of the system state vector. An account of this technique appears in Gelb[17] and Meditch[18]. Basically, this technique is exercised by expanding the nonlinear rectangular to spherical transformation in a first order Taylor series about the one step predicted state vector $\hat{\underline{x}}_{k+1|k}$, obtained from the tracking filter. The technique for finding an approximate $\underline{z}$ and H follows:

In vector notation, the Taylor expansion is given by

$$\underline{s}_{k+1} \approx \underline{s}_p + \left.\frac{\partial \underline{s}(\underline{x})}{\partial \underline{x}}\right|_{\underline{x}_p} (\underline{x}_{k+1} - \underline{x}_p) \tag{4-4}$$

where $\underline{s} = [r|e|b] \Rightarrow \underline{s} = \left[\left[\sqrt{x^2\ y^2\ z^2}\ \middle|\ \sin^{-1}\left(\frac{z}{\sqrt{x^2+y^2+z^2}}\right)\middle|\cos^{-1}\left(\frac{x}{\sqrt{x^2+y^2}}\right)\right]\right]^T$

$$\underline{x}_p = \hat{\underline{x}}_{k+1|k} = \phi\, \hat{\underline{x}}_k$$

$$\underline{s}_p = \underline{s} \text{ evaluated at } \underline{x}_p$$

For clarity, equation (4-4) is expressed in scalar form yielding

$$\begin{aligned}
r_{k+1} &\approx r_p + \left.\frac{\partial r}{\partial x}\right|_{\underline{x}_p}(x_{k+1} - x_p) + \left.\frac{\partial r}{\partial y}\right|_{\underline{x}_p}(y_{k+1} - y_p) + \left.\frac{\partial r}{\partial z}\right|_{\underline{x}_p}(z_{k+1} - z_p) \\
e_{k+1} &\approx e_p + \left.\frac{\partial e}{\partial x}\right|_{\underline{x}_p}(x_{k+1} - x_p) + \left.\frac{\partial e}{\partial y}\right|_{\underline{x}_p}(y_{k+1} - y_p) + \left.\frac{\partial e}{\partial z}\right|_{\underline{x}_p}(z_{k+1} - z_p) \qquad (4\text{-}5) \\
b_{k+1} &\approx b_p + \left.\frac{\partial b}{\partial x}\right|_{\underline{x}_p}(x_{k+1} - x_p) + \left.\frac{\partial b}{\partial y}\right|_{\underline{x}_p}(y_{k+1} - y_p) + \left.\frac{\partial b}{\partial z}\right|_{\underline{x}_p}(z_{k+1} - z_p)
\end{aligned}$$

Performing the operations indicated in equations (4-5), an approximate linear observation equation can be expressed as

$$\underline{s}_{k+1} - \underline{s}_p + \hat{H}_{k+1}\,\underline{x}_p \approx \hat{H}_{k+1}\,\underline{x}_{k+1} \tag{4-6}$$

$$\hat{\underline{H}}_{k+1} = \left[\begin{array}{cccccc} \frac{\partial r}{\partial x} & 0 & \frac{\partial r}{\partial y} & 0 & \frac{\partial r}{\partial z} & 0 \\ \frac{\partial e}{\partial x} & 0 & \frac{\partial e}{\partial y} & 0 & \frac{\partial e}{\partial z} & 0 \\ \frac{\partial b}{\partial x} & 0 & \frac{\partial b}{\partial y} & 0 & \frac{\partial b}{\partial z} & 0 \end{array}\right]\Bigg|_{\underline{x}_p}$$

Defining the measurement vector $\underline{z}$ as a vector of measured range elevation and bearing, we can express $\underline{z}$ in terms of $\underline{x}$ by

$$\underline{z}_{k+1} = \underline{s}_{k+1} + \underline{v}_{k+1} \tag{4-7}$$

Where $\underline{v}_{k+1}$ is a vector of white Gaussian measurement errors in range elevation and bearing. The statistics of $\underline{v}$ are given by the measurement error covariance matrix R, where $R_k \triangleq E\,[\underline{v}_k \underline{v}_k^T]$. Substituting equation (4-7) into equation (4-6) and evaluating $\hat{H}$ results the linearized observation equation

$$\underline{z}^*_{k+1} \approx \hat{H}_{k+1}\,\underline{x}_{k+1} + \underline{v}_{k+1} \tag{4-8}$$

where

$$z^*_{k+1} = \underline{z}_{k+1} - \underline{s}_p + \hat{H}_{k+1}\,\underline{x}_p$$

$$\hat{H}_{k+1} = \left[\begin{array}{cccccc} \frac{x_p}{r_p} & 0 & \frac{y_p}{r_p} & 0 & \frac{z_p}{r_p} & 0 \\ \frac{-x_p z_p}{D_p} & 0 & \frac{-y_p z_p}{D_p} & 0 & \frac{x_p^2+y_p^2}{D_p} & 0 \\ \frac{-y_p}{x_p^2+y_p^2} & 0 & \frac{x_p}{x_p^2+y_p^2} & 0 & 0 & 0 \end{array}\right]$$

$$r_p = \sqrt{x_p^2 + y_p^2 + z_p^2}$$

$$D_p = r_p^2 \sqrt{x_p^2 + y_p^2}$$

A block diagram of the non-adaptive Kalman filter using the data linearization process is shown in Figure 5. Notice the one step predicted state vector is used to compute the observation equation parameters.

4.5 Model Linearization

The previous technique (data linearization) attempts to find an approximate linear relation between the rectangular target state vector and the spherical radar data. The technique of model linearization is very nearly the reverse, i.e. find an approximate target state model in spherical coordinates. Modeling in spherical coordinates allows the measurement vector to be expressed directly as a linear function of the target state vector. An exact solution in spherical coordinates is not only difficult to achieve, but contains many undesirable nonlinear expressions. The approach developed here attempts to find an approximate spherical companion of equation (4-2) via the following expansions:

$$\begin{aligned} r_{k+1} &\approx r_k + \frac{\partial r}{\partial \underline{x}^T}\Big|_{\underline{x}_k} (\underline{x}_{k+1} - \underline{x}_k) \\ e_{k+1} &\approx e_k + \frac{\partial e}{\partial \underline{x}^T}\Big|_{\underline{x}_k} (\underline{x}_{k+1} - \underline{x}_k) \\ b_{k+1} &\approx b_k + \frac{\partial b}{\partial \underline{x}^T}\Big|_{\underline{x}_k} (\underline{x}_{k+1} - \underline{x}_k) \end{aligned} \qquad (4\text{-}9)$$

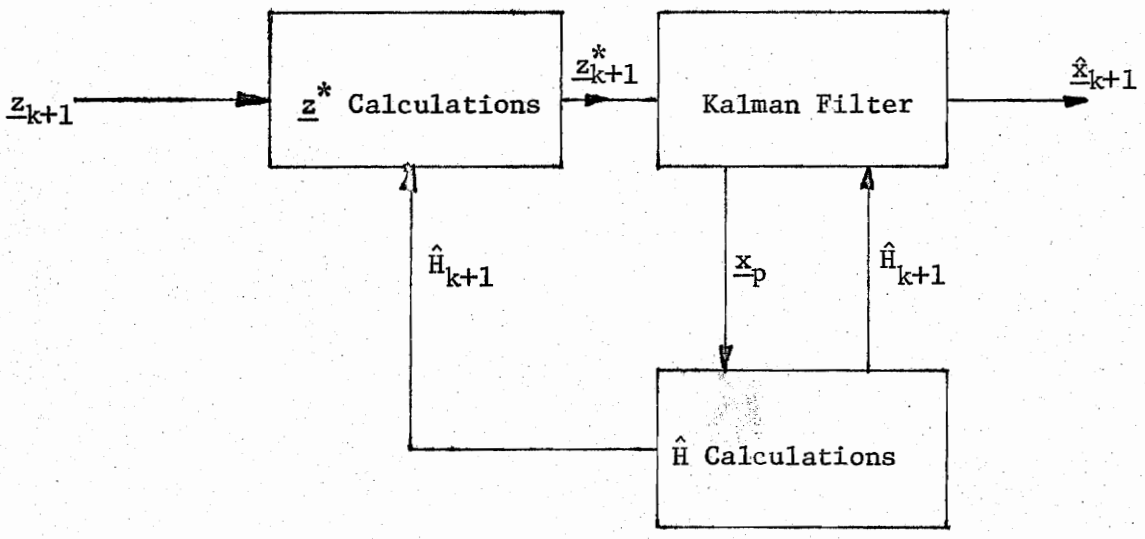


FIGURE 5 Kalman Filter Using Data Linearization

where r_{k+1}, e_{k+1}, and b_{k+1} represent target range, elevation and bearing respectively at time t_{k+1}. Substituting the dynamic description of $\underline{x}$ defined by equation (4-2) into equations (4-9) and performing the indicated expansions results in the approximate spherical model

$$\underline{s}_{k+1} \approx \phi^{(s)} \underline{s}_k + \Psi^{(s)} (\underline{u}_k + \underline{w}_k)$$

where

$$\underline{s} = \left|\underline{r} \middle| \dot{r} \middle| e \middle| \dot{e} \middle| b \middle| \dot{b}\right|^T$$

$$\underline{u} = \left|\underline{u_x} \middle| u_y \middle| u_z\right|^T$$

$$\underline{w} = \left|\underline{w_x} \middle| w_y \middle| w_z\right|$$

the derivation of $\phi^{(s)}$ and $\Psi^{(s)}$ is presented in Appendix A. The matrix $\Psi^{(s)}$ is rather complex and also produces coupling between the components of state vector $\underline{s}$. A much simpler expression can be obtained if $\underline{w}$ and $\underline{u}$ are assumed forces acting independently in the spherical directions r,e, b, thus expressing the disturbance and command vectors by

$$\underline{w} = \left|\underline{w}_r \middle| w_e \middle| \underline{w}_b\right|^T$$

$$\underline{u} = \left|\underline{u}_r \middle| u_e \middle| \underline{u}_b\right|^T$$

As discussed earlier, the original rectangular form of $\underline{w}$ and $\underline{u}$ was merely for mathematical convenience. Therefore, no loss in modeling accuracy is realized by adopting the spherical form. This assumption has been used successfully by Brands[19] for estimators used in gunfire control systems.

The system state and measurement model now becomes

$$\underline{s}_{k+1} = \phi^{(s)} \underline{s}_k + \Gamma^{(s)}_k (\underline{u}_k + \underline{w}_k) \qquad (4\text{-}10)$$

$$\underline{z}_{k+1} = H \underline{s}_{k+1} + \underline{v}_{k+1}$$

where

$$\phi^{(s)} = \begin{bmatrix} 1 & p_1 & 0 & 0 & 0 & 0 \\ 0 & p_1^* & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & p_1 & 0 & 0 \\ 0 & 0 & 0 & p_1^* & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & p_1 \\ 0 & 0 & 0 & 0 & 0 & p_1^* \end{bmatrix}$$

$$\Gamma^{(s)}_k = \begin{bmatrix} p_2 & 0 & 0 \\ p_1 & 0 & 0 \\ 0 & p_{2/d_1} & 0 \\ 0 & p_{1/d_1} & 0 \\ 0 & 0 & p_{2/d_2} \\ 0 & 0 & p_{1/d_2} \end{bmatrix}$$

$\underline{z}$ = vector of measured range elevation and bearing

$d_1 = \hat{r}_k \cos \hat{e}_k$

$p_1 = \frac{1-\varepsilon^{-\rho T}}{\rho}$

$p_1^* = \varepsilon^{-\rho T}$

$p_2 = \frac{\varepsilon^{-\rho T} - 1 + \rho T}{\rho^2}$

The derivation of $\Gamma_k^{(s)}$ also appears in Appendix A. Notice equation (4-10) is actually three uncoupled linear difference equations and therefore allows each coordinate direction to be filtered independently as shown in Figure 6. The importance of this decoupled approximate spherical modeling will become more obvious in Chapter 5.

4.6 Simulation Results

The two non-adaptive estimators of Figures 5 and 6 were compared via computer simulation using the rectangular model of equation (4-2) to generate target trajectories. An iteration interval of 0.05 sec was used. Viscous drag coefficient, ρ, was set to zero. The vector $\underline{w}$, of rectangular components, was generated zero mean white Gaussian with standard deviation approximately 3G in each direction. Measurement errors were generated as zero mean white Gaussian sequences of standard deviation 25 ft., 1 deg. and 1 deg. for range, elevation and bearing respectively. The purpose of the simulations is to compare the more mathematically convenient decoupled spherical modeling technique (model linearization) with the more precise rectangular modeling technique (data linearization). The comparison was performed via a series of experiments involving nonmaneuvering and maneuvering targets. For brevity, the results of two experiments are presented.

4.6.1 Experiement #1 (Nonmaneuvering Target)

The first experiment involves a nonmaneuvering target, i.e. $u_x = u_y = u_z \equiv 0$ for all time. Initial conditions are given by

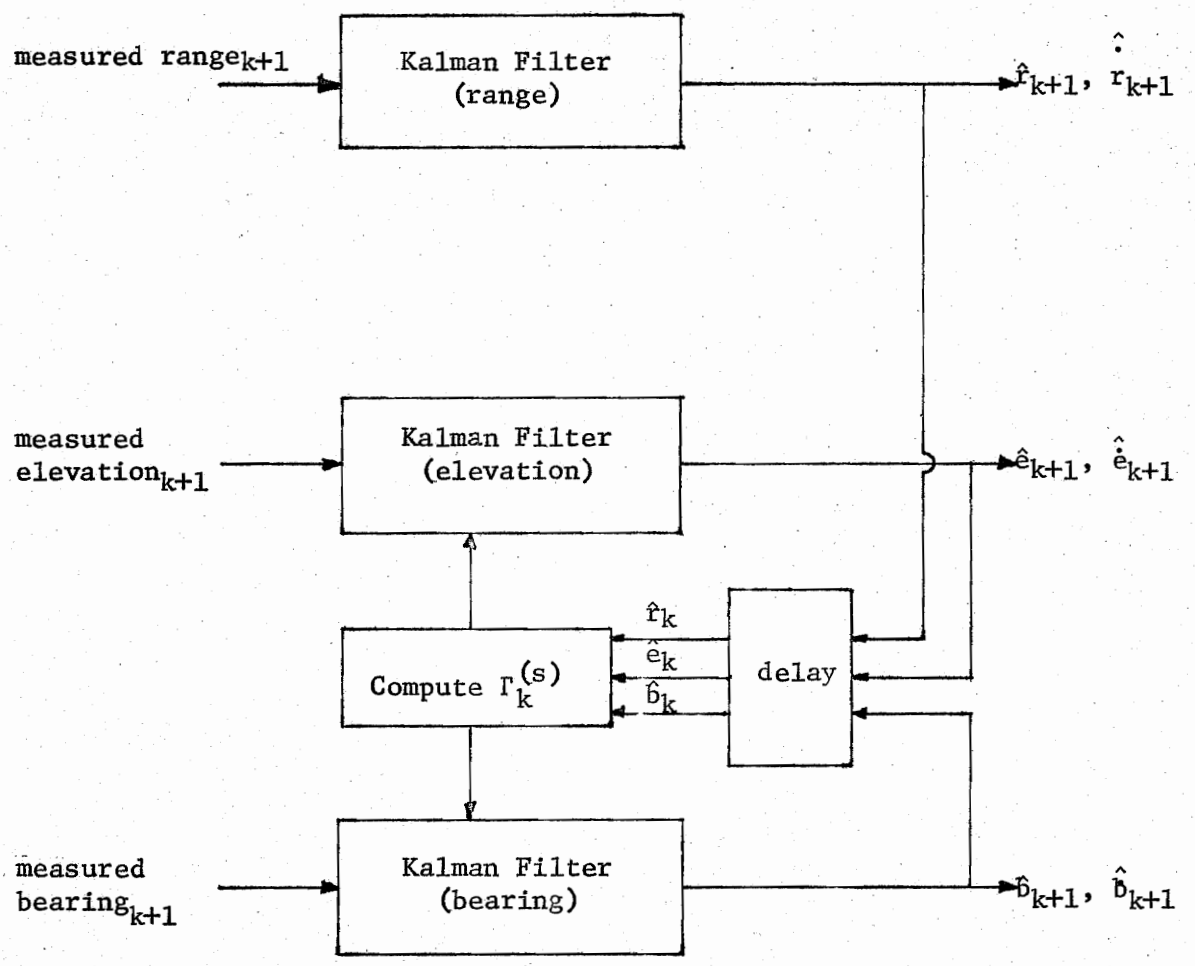


FIGURE 6 Kalman Filter Using Decoupled Spherical Model

$\underline{x}_o$ = [22000ft., 1000ft./sec, 1500ft., 0, 2400ft., 100ft./sec]^T.

Results are presented in Figures 7, 8 and 9. Filter "R" is a rectangular filter (data linearization) while filter "S" is a spherical filter (decoupled approximate spherical model). For purposes of comparison, the spherical estimates are converted to rectangular coordinates. Comparing results shows the more precise rectangular estimator to achieve only slight tracking superiority over the decoupled spherical estimator.

4.6.2 Experiment #2 (Maneuvering Target)

The second experiment is identical to the first except for a deterministic input (unknown to the estimators) occuring at time = 7 seconds causing a 10G acceleration in the x-direction. Figures 10, 11 and 12 compare the estimators' performance. Notice the relatively large bias errors that develop shortly after the maneuver initiates. Also notice that again the decoupled spherical estimator compares favorably with the rectangular estimator. This experiment is very important since the non-adaptive estimator needs to be as robust as possible with respect to large scale maneuvers. Experiments involving more radical maneuvering trajectories have shown the decoupled spherical estimator to actually outperform the rectangular estimator. This is most probably due to the spherical estimator being slightly less sensitive to poor estimates of $\Gamma^{(s)}$ than the rectangular estimator to poor estimates of H.

4.7 Conclusions from Non-Adaptive Experiments

From many experiments, such as the two discussed above, it has been concluded that the decoupled spherical technique (model linearization)

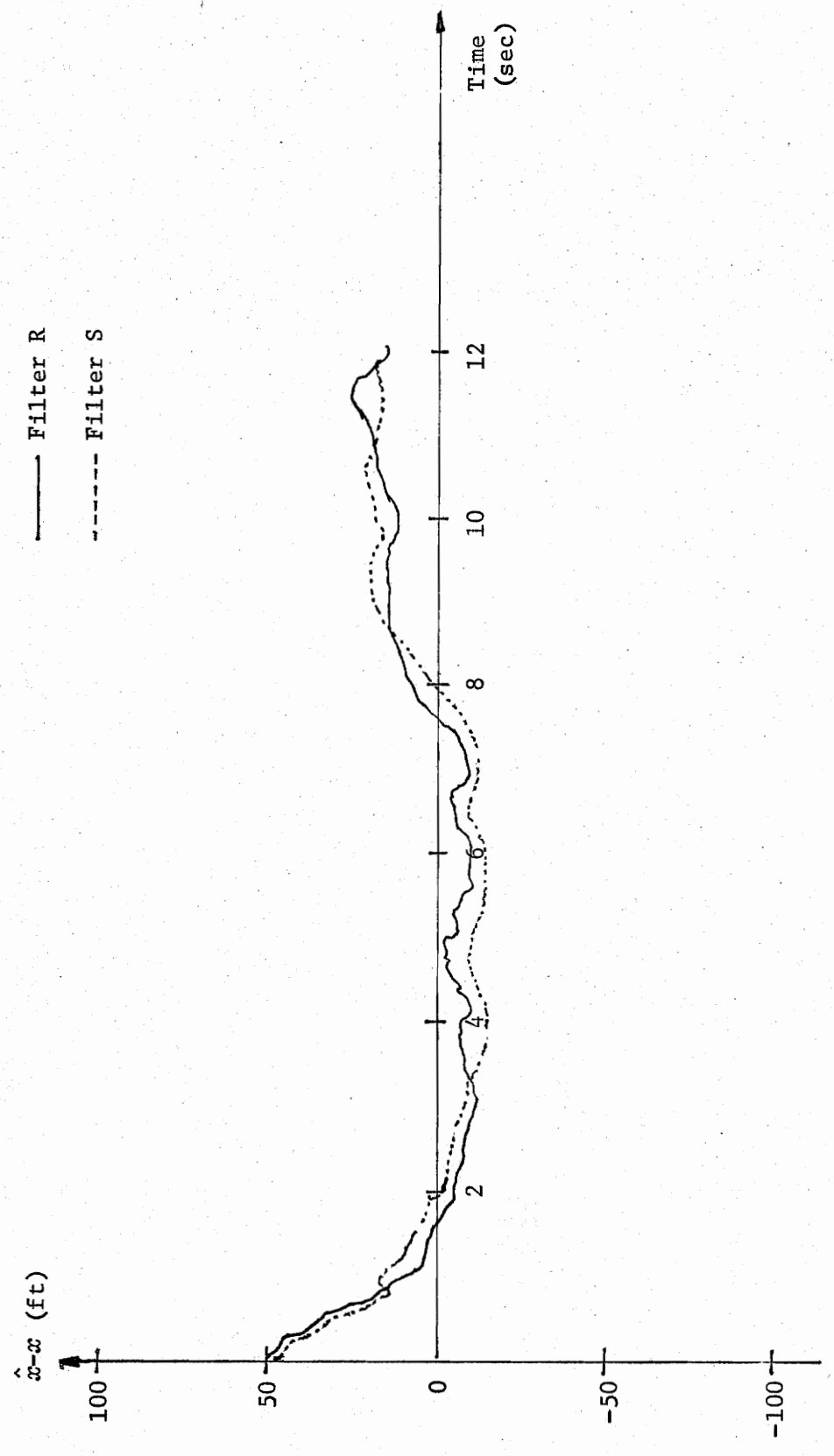


FIGURE 7 x-Direction Tracking Errors

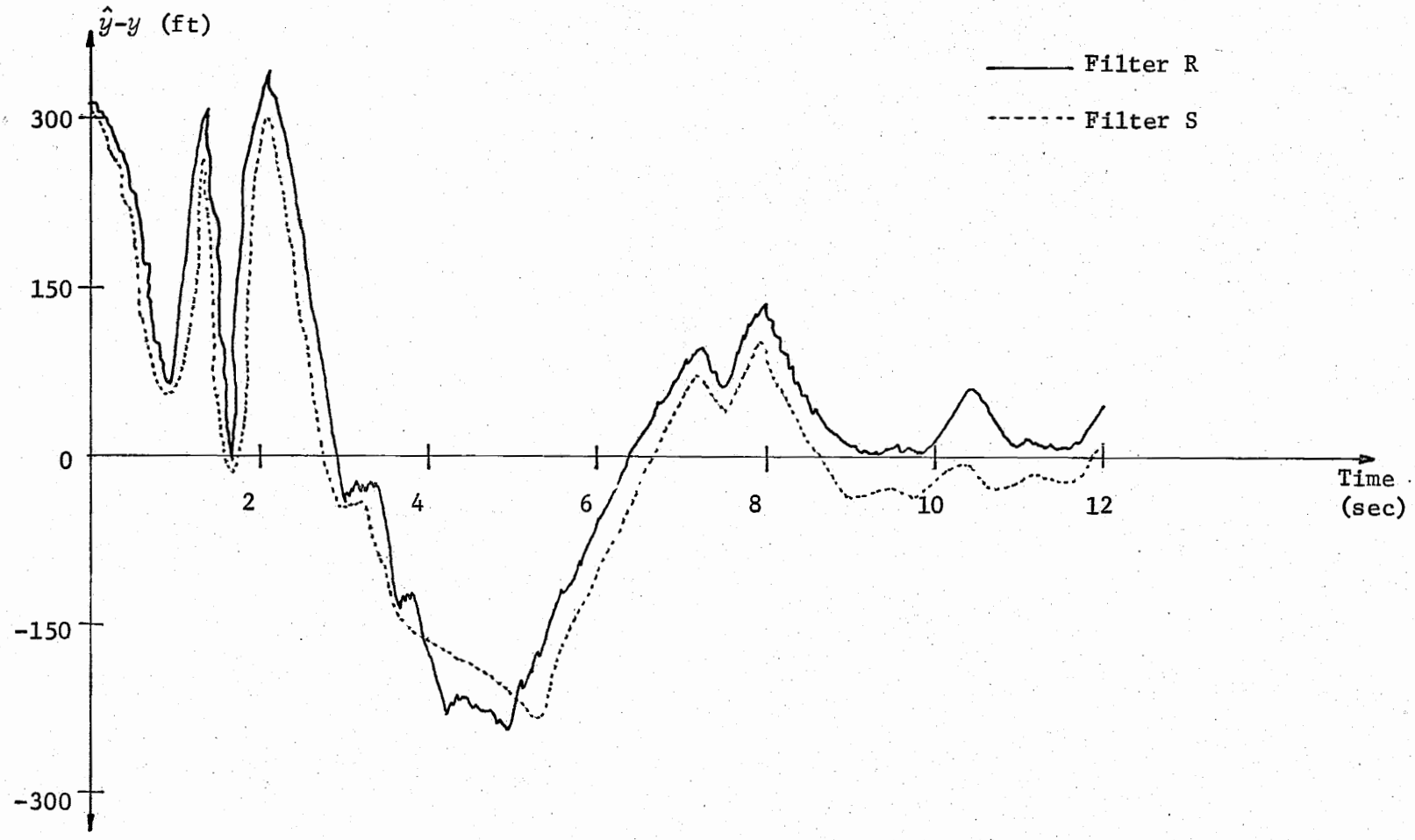


FIGURE 8 y-Direction Tracking Errors

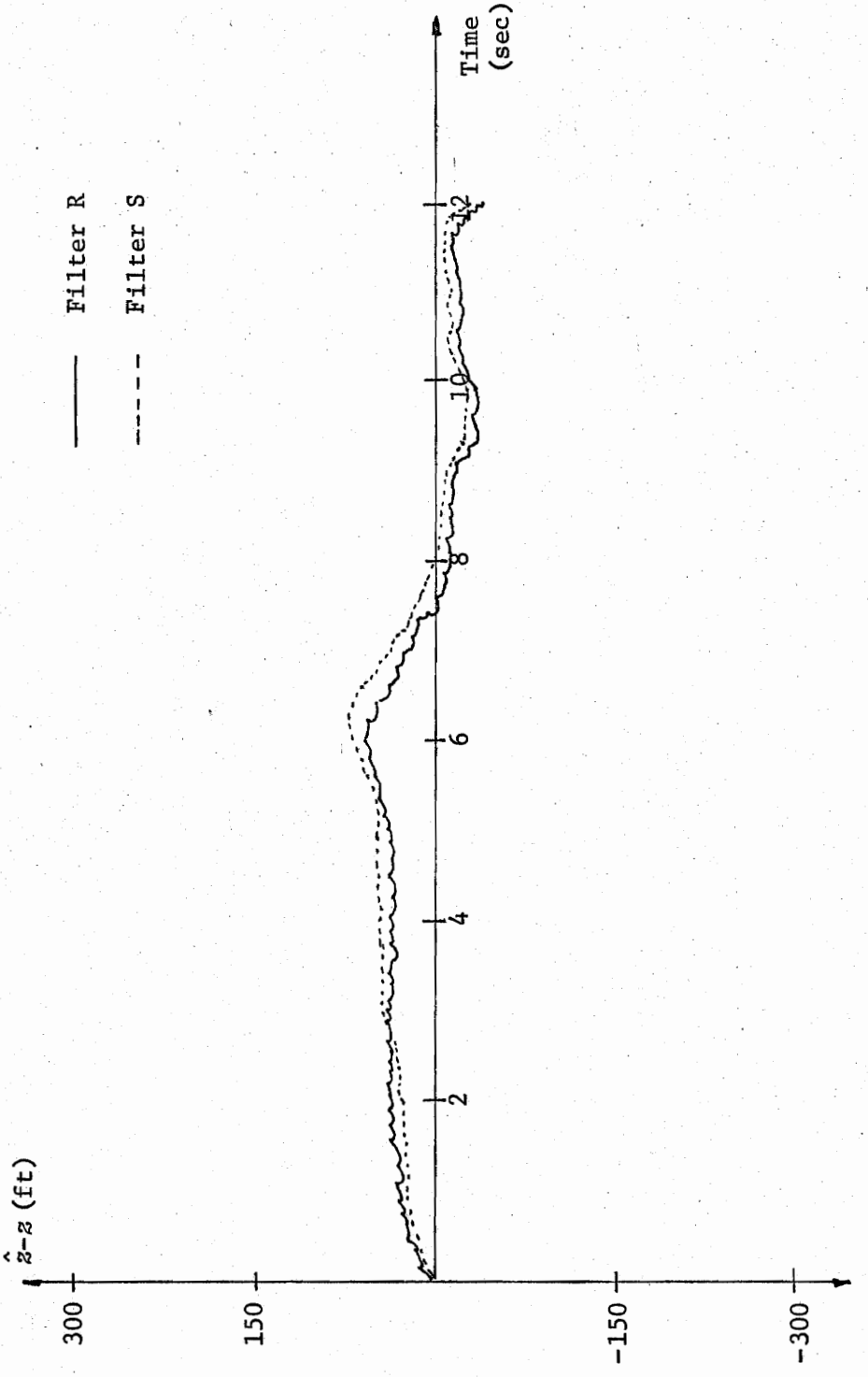


FIGURE 9 z-Direction Tracking Errors

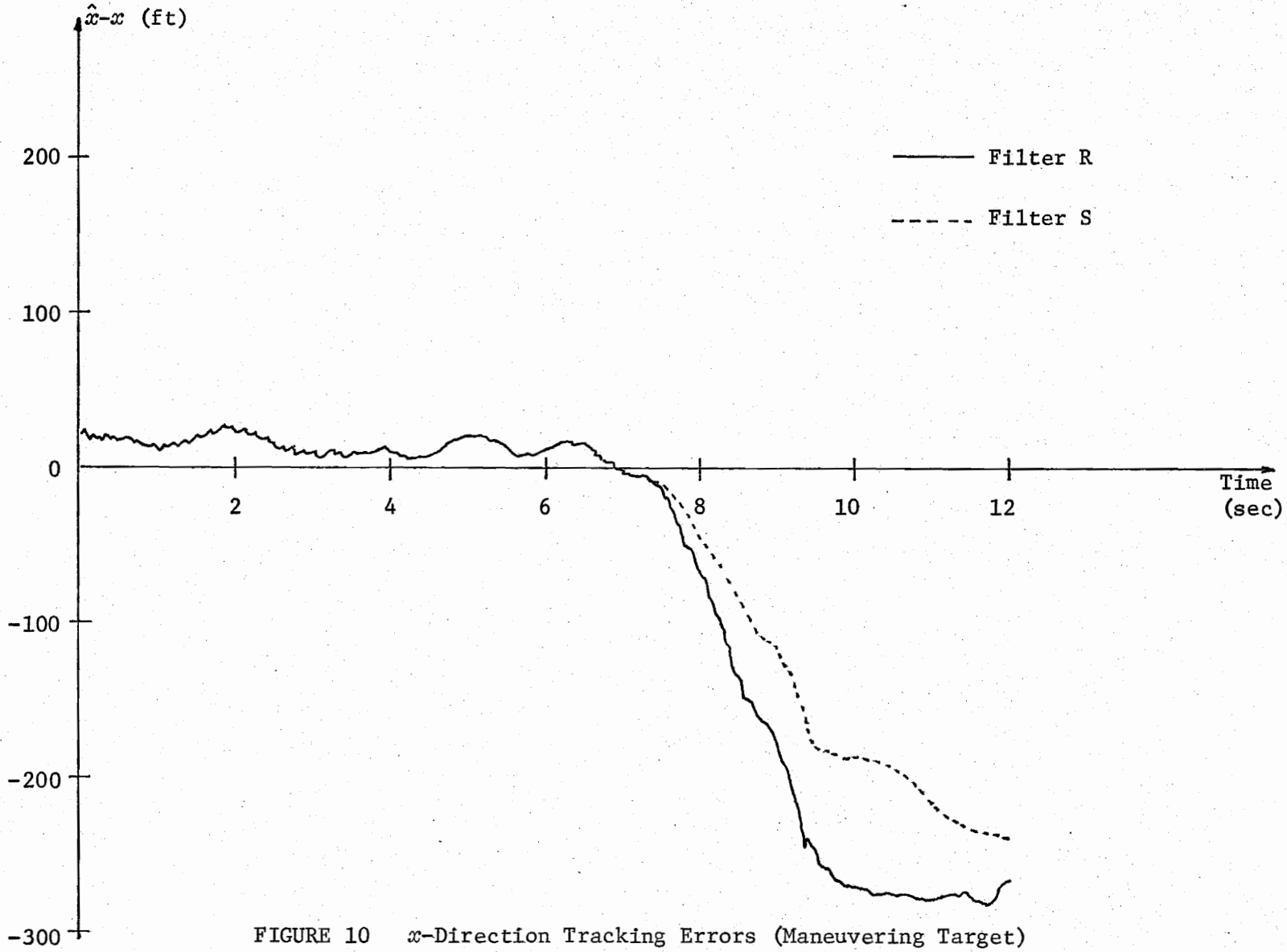


FIGURE 10 x-Direction Tracking Errors (Maneuvering Target)

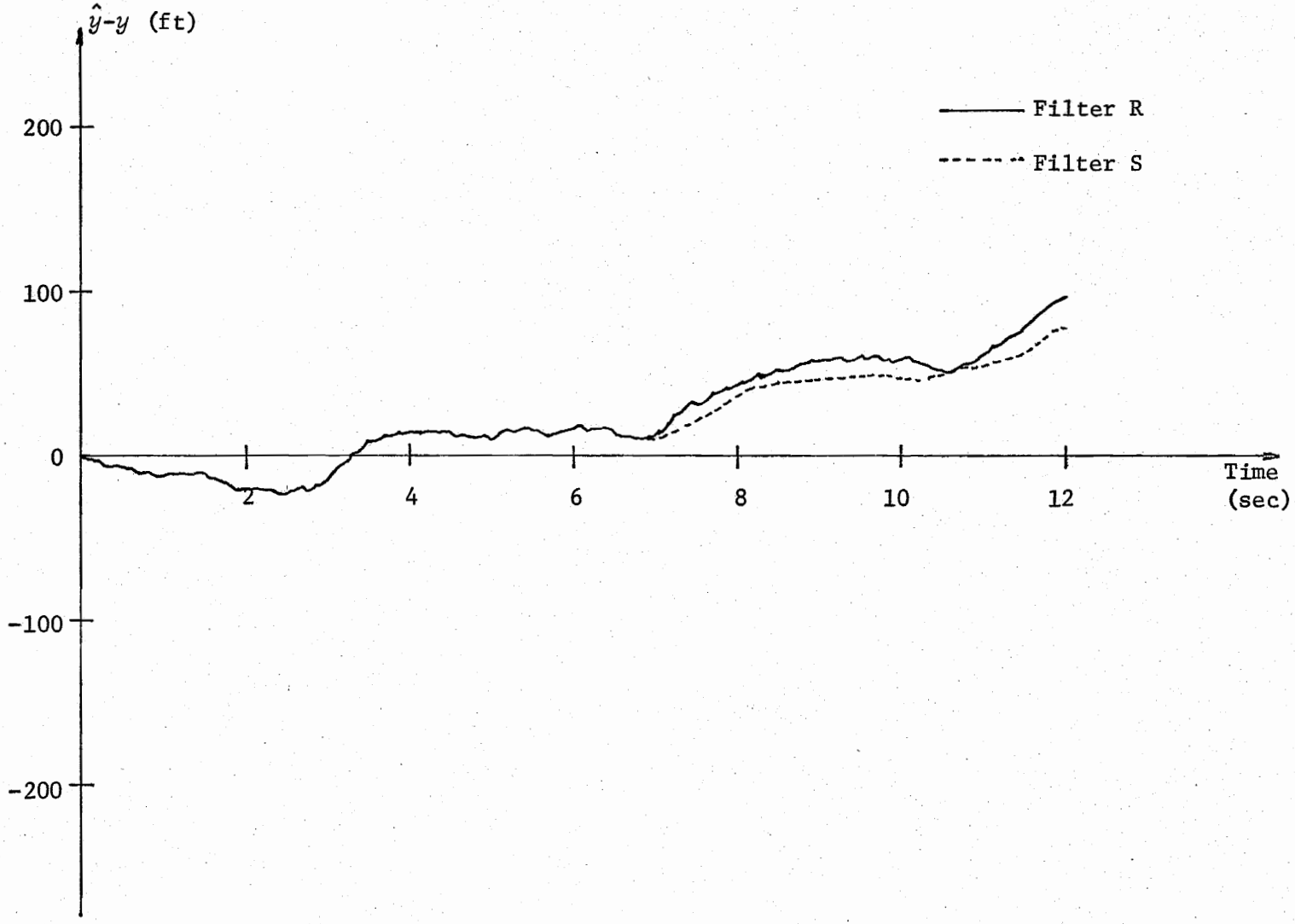


FIGURE 11 y-Direction Tracking Errors (Maneuvering Target)

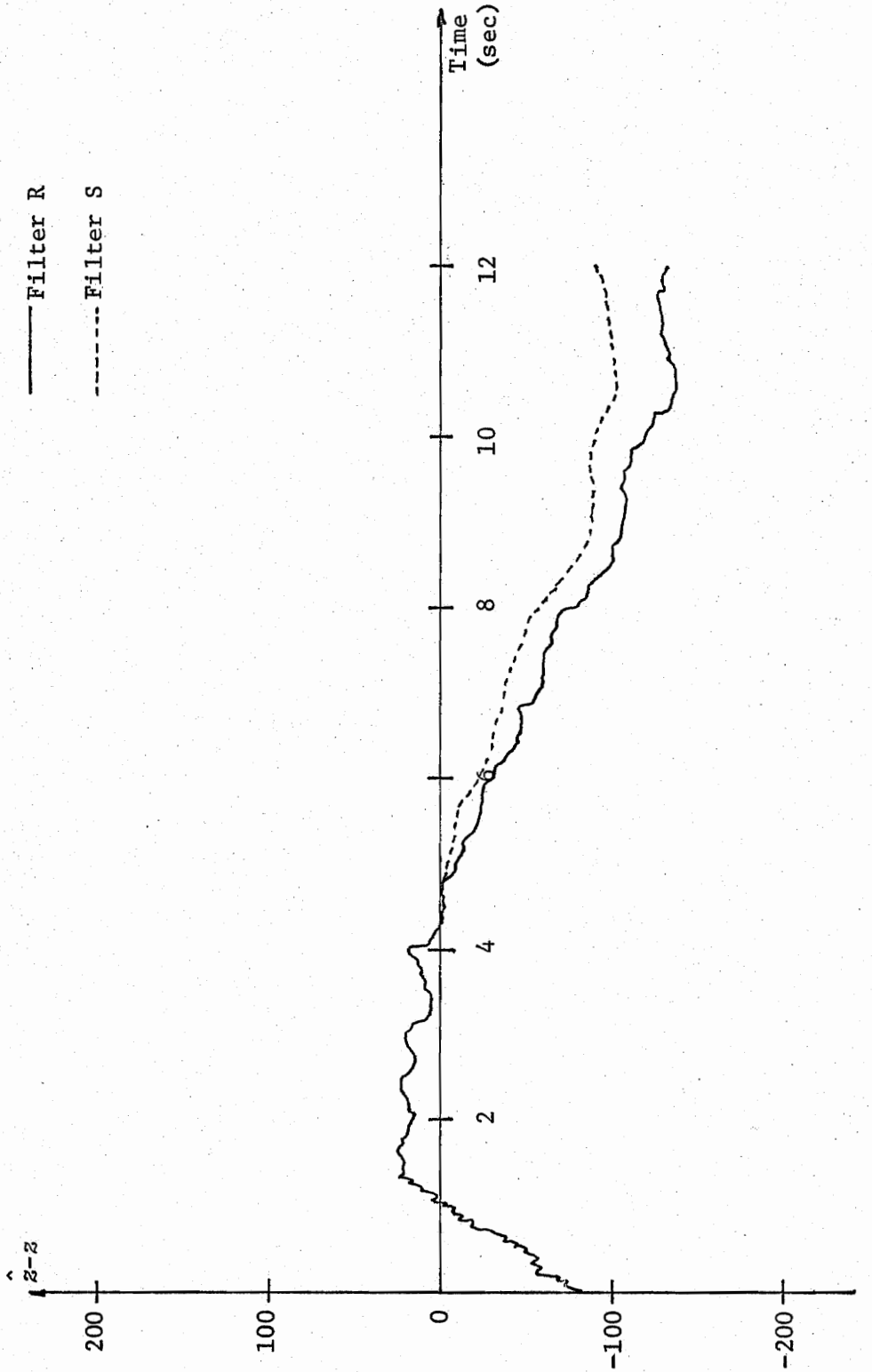


FIGURE 12 z-Direction Tracking Errors (Maneuvering Target)

is a very useful approximation to the more precise rectangular technique (data linearization). The rectangular estimator shows a slight advantage in nonmaneuvering experiments while the decoupled spherical estimator shows some advantage in experiments involving large scale maneuvers. The rectangular estimator's superiority in nonmaneuvering cases can easily be attributed to its more accurate model of target dynamics. In actual implementation, neither model (rectangular or spherical) will be a very accurate representation of target dynamics. Therefore, it is concluded that a valid choice between the two techniques can be made based on ease of implementation for a particular (adaptive or non-adaptive) state estimation scheme. The importance of this free choice will become quite obvious in the next chapter when computational burden is investigated.

5.0 DISCRETE SEMI-MARKOV ADAPTIVE ESTIMATOR

5.1 Introduction

The discrete semi-Markov adaptive estimator is a solution to the maneuvering target tracking problem when the maneuvers can be described by a finite set of maneuver commands. In this study, N maneuver commands are modeled in each coordinate direction to yield N^3 possible combinations of commands. Further, it is assumed the commands in a given direction are statistically independent of commands in any other direction. Target dynamics are modeled by equations (4-2) using a non-zero viscous drag coefficient.

The adaptive estimator consists of a conventional Kalman tracking filter (described in Chapter 4.0) augmented by the recursive maneuver command estimation algorithm developed in Chapter 3.0. The recursive algorithm is based on the semi-Markov maneuver modeling described in Chapter 2.0.

The purpose of this chapter is to develop two techniques of implementing the adaptive algorithm. Comparisons will be made concerning computational burden, as well as tracking performance.

5.2 Implementation via Data Linearization

The technique of data linearization is fully developed in Chapter 4.0. For convenience, the technique's system modeling equations are repeated below.

$$\begin{aligned} \underline{x}_{k+1} &= \phi \underline{x}_k + \Gamma(\underline{u}_k + \underline{w}_k) \\ \underline{z}^*_{k+1} &\simeq \hat{H}_{k+1} \underline{x}_{k+1} + \underline{V}_{k+1} \end{aligned} \tag{5-1}$$

where $\underline{u}_k$ is the deterministic maneuver command at time t_k. The parameters $\underline{z}^*$ and $\hat{H}$ are computed from equation (4-8) using radar measurements and the one step predicted state vector.

Application of the semi-Markov adaptive algorithm follows in basically a straightforward manner. The filter equation is modified to incorporate $\underline{\hat{u}}$ yielding the adaptive filter equation

$$\underline{\hat{x}}_{k+1} = \phi\underline{\hat{x}}_k + \Gamma\underline{\hat{u}}_k + K_{k+1}[\underline{z}^*_{k+1} - \hat{H}_{k+1}\phi\underline{\hat{x}}_{k+1} - \hat{H}_{k+1}\Gamma\underline{\hat{u}}_k] \tag{5-2}$$

The maneuver command estimate, $\underline{\hat{u}}_k$ is provided by the recursive algorithm developed in Chapter 3.0. The algorithm defines $\underline{\hat{u}}_k$ as the expected value of $\underline{u}_k$ given $\underline{z}^*_{k+1}$ and $\underline{\hat{x}}_k$ and thus computes $\underline{\hat{u}}_k$ from

$$\underline{\hat{u}}_k = \sum_{q=1}^{N^3} \underline{u}^{(q)} \text{prob}[\underline{u}_k = \underline{u}^{(q)} | \underline{z}^*_{k+1}\underline{\hat{x}}_k]. \tag{5-3}$$

The probabilities in equation (5-3) are recursively computed from

$$\text{prob}[\underline{u}_k=\underline{u}^{(q)} | \underline{z}^*_{k+1}, \underline{\hat{x}}_k] = (\text{const})p(\underline{z}^*_{k+1} | \underline{u}_k=u^{(q)}, \underline{\hat{x}}_k) \sum_{\omega=1}^{N^3} \text{prob}(\underline{u}_{k-1} =$$

$$\underline{u}^{(\omega)} | \underline{\hat{x}}_{k-1}, \underline{z}^*_k) \cdot \theta^*_{\omega q}(t_k - t_{k-1}) \tag{5-4}$$

where "const" is evaluated from $\sum_{q=1}^{N^3} \text{prob}[\underline{u}_k=\underline{u}^{(q)} | \underline{z}^*_{k+1}, \underline{\hat{x}}_k] = 1$, and $p(\underline{z}^*_{k+1} | \underline{u}_k=\underline{u}^{(q)}, \underline{\hat{x}}_k)$ is evaluated from a Gaussian density function described by

$$\underline{\text{mean}}^{(q)} = \hat{H}_{k+1}\phi\underline{\hat{x}}_k + \hat{H}_{k+1}\Gamma\underline{u}^{(q)}$$

$$\text{covariance} = \hat{H}_{k+1}P_{k+1|k}\hat{H}^T_{k+1} + R .$$

The probability $\theta^*_{\omega q}$ is obtained from semi-Markov maneuver modeling. Notice that $\text{prob}[\underline{u}_{k-1} = \underline{u}^{(\omega)} | \underline{\hat{x}}_{k-1}, \underline{z}^*_k]$ is just the previously obtained probability. Also notice the summations of equations (5-3) and (5-4) extend to N^3 since there are N^3 possible maneuver combinations.

5.2.1 Computational Procedure

The above equations are mathematically correct but cumbersome for computer application. A more convenient formulation can be obtained by expressing the maneuver command vector in terms of its scalar components u_x, u_y and u_z. Equation (5-3) becomes

$$\hat{u}_{x_k} = \sum_{i=1}^{N} u^{(i)} \text{ prob}[u_{x_k} = u^{(i)} | \underline{z}^*_{k+1}, \hat{\underline{x}}_k]$$

$$\hat{u}_{y_k} = \sum_{i=1}^{N} u^{(i)} \text{prob}[u_{y_k} = u^{(i)} | \underline{z}^*_{k+1}, \hat{\underline{x}}_k] \qquad (5\text{-}5)$$

$$\hat{u}_{z_k} = \sum_{i=1}^{N} u^{(i)} \text{prob}[u_{z_k} = u^{(i)} | \underline{z}^*_{k+1}, \hat{\underline{x}}_k]$$

The marginal probabilities necessary for evaluating equations (5-5) can be computed from

$$\text{prob}\left[u_{x_k} = u^{(i)} | \underline{z}^*_{k+1}, \hat{\underline{x}}_k\right] = \sum_{j=1}^{N} \sum_{\ell=1}^{N} \text{prob}\left[\begin{array}{l} u_{x_k} = u^{(i)} \\ \text{and} \\ u_{y_k} = u^{(j)} \\ \text{and} \\ u_{z_k} = u^{(\ell)} \end{array} \middle| \underline{z}^*_{k+1}, \hat{\underline{x}}_k\right]$$

$$\text{prob}\left[u_{y_k} = u^{(i)} | \underline{z}^*_{k+1}, \hat{\underline{x}}_k\right] = \sum_{j=1}^{N} \sum_{\ell=1}^{N} \text{prob}\left[\begin{array}{l} u_{x_k} = u^{(j)} \\ \text{and} \\ u_{y_k} = u^{(i)} \\ \text{and} \\ u_{z_k} = u^{(\ell)} \end{array} \middle| \underline{z}^*_{k+1}, \hat{\underline{x}}_k\right] \qquad (5\text{-}6)$$

$$\text{prob}\left[u_{z_k}=u^{(i)} \,\middle|\, \underline{z}^*_{k+1},\ \underline{\hat{x}}_k\right] = \sum_{j=1}^{N}\sum_{\ell=1}^{N} \text{prob}\left[\begin{array}{l} u_{x_k}=u^{(j)} \\ \text{and} \\ u_{y_k}=u^{(\ell)} \\ \text{and} \\ u_{z_k}=u^{(i)} \end{array} \,\middle|\, \underline{z}^*_{k+1},\ \underline{x}_k\right]$$

The probability update equation (equation (5-4)) now becomes

$$\text{prob}\left[\begin{array}{l} u_{x_k}=u^{(i)} \\ \text{and} \\ u_{y_k}=u^{(j)} \\ \text{and} \\ u_{z_k}=u^{(\ell)} \end{array} \,\middle|\, \underline{z}^*_{k+1},\ \underline{x}_k\right] = (\text{const})\,p\left[\underline{z}^*_{k+1} \,\middle|\, \begin{array}{l} u_{x_k}=u^{(i)} \\ \text{and} \\ u_{y_k}=u^{(j)} \\ \text{and} \\ u_{z_k}=u^{(\ell)} \\ \text{and} \\ \underline{\hat{x}}_k \end{array}\right] \sum_{\alpha=1}^{N}\sum_{\beta=1}^{N}\sum_{\gamma=1}^{N} \cdot$$

$$\left[\begin{array}{l} u_{z_k}=u^{(\alpha)} \\ \text{and} \\ u_{y_{k-1}}=u^{(\beta)} \\ \text{and} \\ u_{z_{k-1}}=u^{(\gamma)} \end{array} \,\middle|\, \underline{z}^*_k,\ \underline{\hat{x}}_{k-1}\right] \text{prob}\left[\begin{array}{l} u_{x_k}=u^{(i)} \\ \text{and} \\ u_{y_k}=u^{(j)} \\ \text{and} \\ u_{z_k}=u^{(\ell)} \end{array} \,\middle|\, \begin{array}{l} u_{x_{k-1}}=u^{(\alpha)} \\ \text{and} \\ u_{y_{k-1}}=u^{(\beta)} \\ \text{and} \\ u_{z_{k-1}}=u^{(\gamma)} \end{array}\right] \qquad (5\text{-}7)$$

where const is evaluated by summing probabilities. Since u_x, u_y and u_z are assumed statistically independent, the $\text{prob}\left[u_{x_k}=u^{(i)},\ u_{y_k}=u^{(j)},\ u_{z_k}=u^{(\ell)} \,\middle|\, u_{x_{k-1}}=u^{(\alpha)},\ u_{y_{k-1}}=u^{(\beta)},\ u_{z_{k-1}}=u^{(\gamma)}\right]$ can be replaced by

$$\theta^{*}_{\alpha i}(t_k - t_{k-1})\theta^{*}_{\beta j}(t_k - t_{k-1})\theta^{*}_{\gamma \ell}(t_k - t_{k-1}).$$

To clarify the computational procedure, a chronological summary of the required steps is presented. The necessary data for starting the sequence is $\underline{\hat{x}}_k$ and the current measurement $\underline{z}_{k+1}$. Also, the N^3 probabilities, $\text{prob}\left[u_{x_{k-1}}=u^{(i)}, u_{y_{k-1}}=u^{(j)}, u_{z_{k-1}}=u^{(\ell)} \,|\, \underline{\hat{x}}_{k-1}, \underline{z}^{*}_{k}\right]$ must be available from the last computer cycle.

(1) Obtain the one step predicted state vector from $\underline{x}_p = \phi\underline{\hat{x}}_k$.

(2) Using $\underline{x}_p$ and $\underline{z}_{k+1}$, compute $\hat{H}_{k+1}$ and $\underline{z}^{*}_{k+1}$ from equation (4-8).

(3) Compute Kalman gain and update error covariance matrix using equations (2-15). Retain the matrix $\left[\hat{H}_{k+1} P_{k+1|k} \hat{H}^{T}_{k+1} + R\right]^{-1}$ and label as c^{-1}.

(4) Compute $p(\underline{z}^{*}_{k+1} | u_{x_k}=u^{(i)}, u_{y_k}=u^{(j)}, u_{z_k}=u^{(\ell)}, \underline{\hat{x}}_k)$ for i = 1, 2, ..., N; j = 1, 2, ..., N; ℓ = 1, 2, ..., N; using $p(\underline{z}^{*}_{k+1} | u_{x_k}=u^{(i)}, u_{y_k}=u^{(j)}, u_{z_k}=u^{(\ell)}, \underline{\hat{x}}_k) \sim \exp\left[-\frac{1}{2}[\underline{z}^{*}_{k+1} - \hat{H}_{k+1}\underline{x}_p - \hat{H}_{k+1}\Gamma\underline{u}^{(i,j,\ell)}]^{T} \cdot c^{-1}[\underline{z}^{*}_{k+1} - \hat{H}_{k+1}\underline{x}_p - \hat{H}_{k+1}\Gamma u^{(i,j,\ell)}]\right]$ where $\underline{u}^{(i,j,\ell)} \triangleq [u^{(i)} | u^{(j)} | u^{(\ell)}]^{T}$ and "$\sim$" denotes "proportional to".

(5) Evaluate equation (5-7) for i = 1, 2, ..., N; j = 1, 2, ..., N; ℓ = 1, 2, ..., N. Save these probabilities for the next computer cycle.

(6) Use equations (5-6) and (5-5) to evaluate $\hat{u}_{x_k}$, $\hat{u}_{y_k}$, $\hat{u}_{z_k}$. Form $\hat{u}_k$ by $\hat{u}_k = \left[\hat{u}_{x_k} \middle| \hat{u}_{y_k} \middle| \hat{u}_{z_k}\right]^{T}$.

(7) Use equation (5-2) to obtain $\underline{\hat{x}}_{k+1}$.

Figure 13 is a block diagram of the discrete semi-Markov adaptive estimator using data linearization.

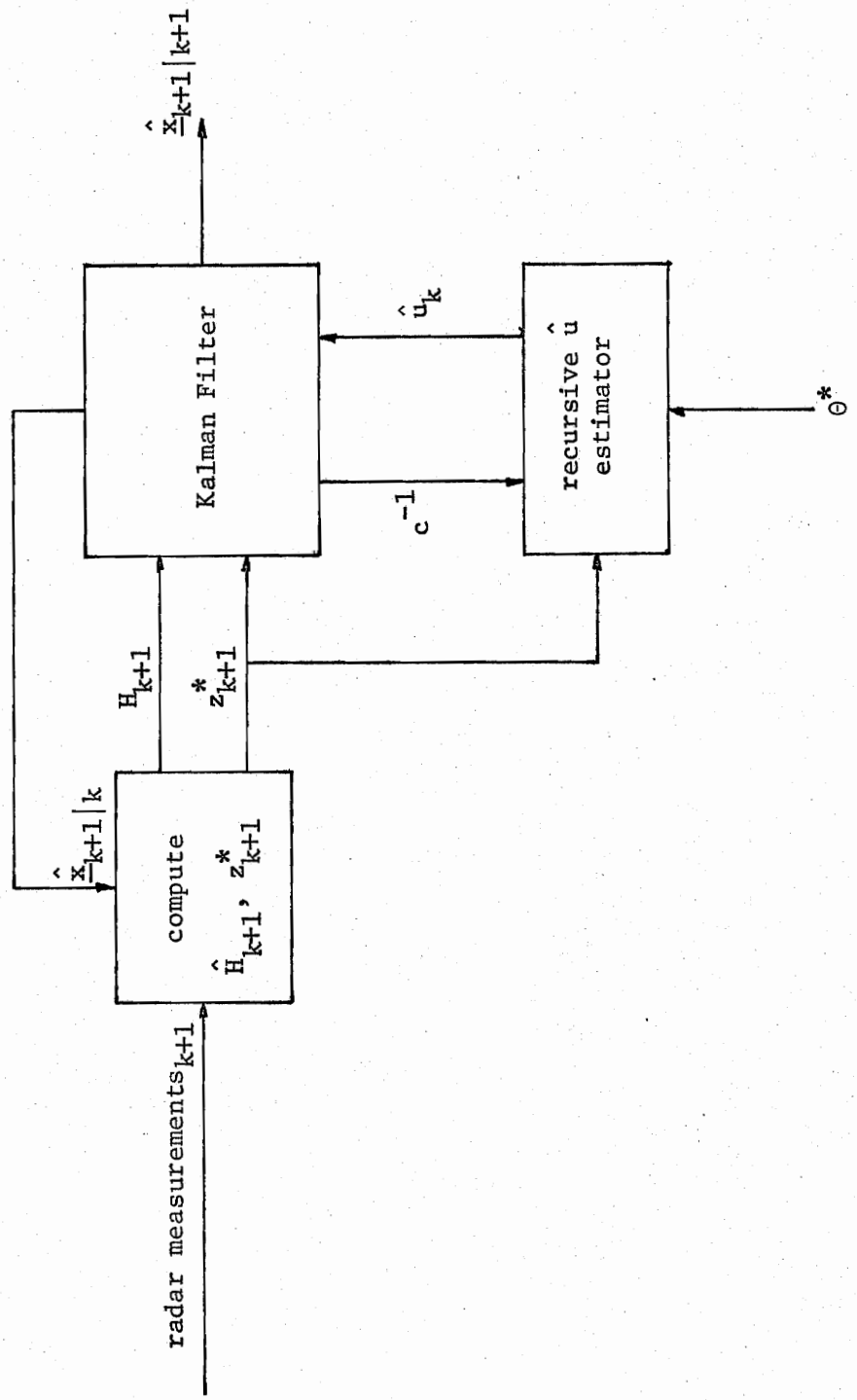


FIGURE 13 Semi-Markov Adaptive Estimator Using Data Linearization

5.2.2 Computational Burden

An indication of computational burden associated with the adaptive portion of the estimator can be obtained by combining equations (5-3) and (5-4) to yield the following expression for $\underline{\hat{u}}_k$.

$$\underline{\hat{u}}_k = (\text{const}) \sum_{q=1}^{N^3} \underline{u}^{(q)} p(\underline{z}^*_{k+1} | \underline{u}_k = \underline{u}^{(q)}, \underline{\hat{x}}_k) \sum_{\omega=1}^{N^3} \text{prob}[\underline{u}_{k-1} = \underline{u}^{(\omega)} | \underline{\hat{x}}_{k-1}, \underline{z}^*_k) \cdot$$

$$\theta^*_{\omega q}(t_k - t_{k-1}) \tag{5-8}$$

Noting equation (5-8) has "nested" summations to N^3 reveals that N^6 "computations" are necessary to compute $\underline{\hat{u}}_k$, where "computations" has been used informally.

5.3 Implementation Via Decoupled Spherical Modeling (Model Linearization)

The decoupled approximate spherical model of equation (4-10) provides independent system modeling equations involving range (r), elevation (e) and bearing (b).

$$\begin{aligned} \underline{x}^{(r)}_{k+1} &= \phi \underline{x}^{(r)}_k + \Gamma^{(r)}(w_{r_k} + u_{r_k}) \\ z^{(r)}_{k+1} &= H\underline{x}^{(r)}_{k+1} + v_{r_{k+1}} \end{aligned} \tag{5-9a}$$

$$\begin{aligned} \underline{x}^{(e)}_{k+1} &= \phi \underline{x}^{(e)}_k + \Gamma^{(e)}_k(w_{e_k} + u_{e_k}) \\ z^{(e)}_{k+1} &= H\underline{x}^{(e)}_{k+1} + v_{e_{k+1}} \end{aligned} \tag{5-9b}$$

$$\begin{aligned} \underline{x}^{(b)}_{k+1} &= \phi \underline{x}^{(b)}_k + \Gamma^{(b)}_k(w_{b_k} + u_{b_k}) \\ z^{(b)}_{k+1} &= H\underline{x}^{(b)}_{k+1} + v_{b_{k+1}} \end{aligned} \tag{5-9c}$$

where $\underline{x}^{(r)} = [r | \dot{r}]^T$

$\underline{x}^{(e)} = [e | \dot{e}]^T$

$\underline{x}^{(b)} = [b|\dot{b}]^T$

$z^{(r)}$, $z^{(e)}$, $z^{(b)}$ = measured range, elevation and bearing, respectively

v_r, v_e, v_b = independent zero mean white Gaussian measurement errors in range, elevation and bearing, respectively

w_r, w_e, w_b = independent zero mean white Gaussian disturbances in range, elevation and bearing, respectively

u_r, u_e, u_b = deterministic maneuver commands in range, elevation and bearing directions, respectively

$H = [1|0]$

The matrices ϕ, $\Gamma^{(r)}$, $\Gamma_k^{(e)}$ and $\Gamma_k^{(b)}$ are defined in Appendix A. Using the modeling equations of (5-9), a nonadaptive estimator can be formulated using three independent Kalman filters as shown in Figure 6.

Implementation of the semi-Markov adaptive algorithm is now much simpler due to the uncoupled modeling. The adaptive filter equations become

$$\underline{\hat{x}}_{k+1}^{(r)} = \phi\underline{\hat{x}}_k^{(r)} + \Gamma^{(r)}\hat{u}_{r_k} + K_{k+1}^{(r)}[z_{k+1}^{(r)} - H\phi\underline{\hat{x}}_k^{(r)} - H\Gamma^{(r)}\hat{u}_{r_k}]$$

$$\underline{\hat{x}}_{k+1}^{(e)} = \phi\underline{\hat{x}}_k^{(e)} + \Gamma_k^{(e)}\hat{u}_{e_k} + K_{k+1}^{(e)}[z_{k+1}^{(e)} - H\phi\underline{\hat{x}}_k^{(e)} - H\Gamma_k^{(e)}\hat{u}_{e_k}]$$

$$\underline{\hat{x}}_{k+1}^{(b)} = \phi\underline{\hat{x}}_k^{(b)} + \Gamma_k^{(b)}\hat{u}_{b_k} + K_{k+1}^{(b)}[z_{k+1}^{(b)} - H\phi\underline{\hat{x}}_{k+1}^{(b)} - H\Gamma_k^{(b)}\hat{u}_{b_k}].$$

Similarly, the determination of $\hat{u}_r$, $\hat{u}_e$ and $\hat{u}_b$ is handled by three independent sets of calculations. As an example, the equations for calculating $\hat{u}_{e_k}$ are presented below.

$$\hat{u}_{e_k} = \sum_{i=1}^{N} u^{(i)} \text{prob}(u_{e_k} = u^{(i)} | \underline{\hat{x}}_k^{(e)}, z_{k+1}^{(e)})$$

where $\text{prob}(u_{e_k} = u^{(i)} | \underline{\hat{x}}_k^{(e)}, z_{k+1}^{(e)}) = (\text{const}) p(z_{k+1}^{(e)} | u_{e_k}, \underline{\hat{x}}_k^{(e)}) \sum_{\alpha=1}^{N} \text{prob}\{u_{k-1}^{(e)} = u^{(\alpha)} | \underline{\hat{x}}_{k-1}^{(e)}, z_k^{(e)}\} \cdot \theta_{\alpha i}^{*}(t_k - t_{k-1})$.

Figure 14 is a block diagram of the decoupled spherical adaptive estimator. The sequence of calculations for implementing the spherical adaptive technique is similar to that of the linearized data technique and therefore will not be presented.

5.3.1 Computational Burden

Due to the decoupled nature of the approximate spherical technique, computational burden is much less than that of the linearized data (rectangular modeling) technique. Combining equations (5-3) and (5-4) (N^3 goes to N) reveals that only N^2 "computations" are necessary per coordinate direction. Therefore, $3N^2$ "computations" are necessary to compute $\hat{a}_{r_k}$, $\hat{a}_{e_k}$ and $\hat{a}_{b_k}$. Again, "computations" has been used informally. To emphasize the tremendous savings, Table 1 is presented comparing the relative computational burdens. Referring to Table 1, we see that 71 commands per coordinate direction can be modeling using the spherical technique with less computational burden than modeling 5 commands per coordinate direction using the rectangular technique. Simulation results have indicated that modeling many commands is desirable thus making the approximate spherical technique attractive.

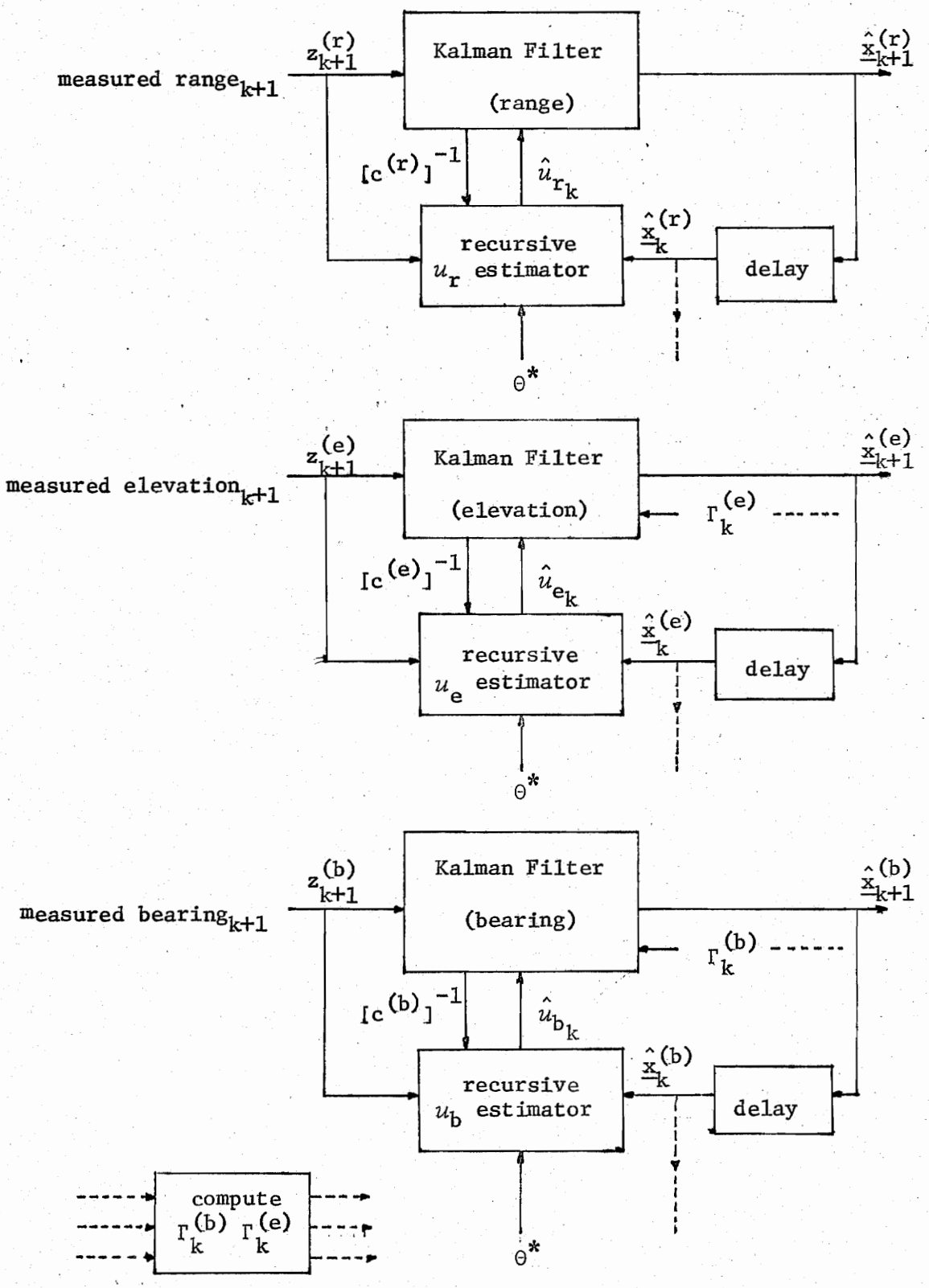


FIGURE 14 Decoupled Spherical Adaptive Estimator

Table 1. Relative Computational Burden Associated with Rectangular and Decoupled Spherical Implementation of Semi-Markov Adaptive Algorithm

Number of Commands Modeled Per Coordinate Direction	Relative Computational Burden	
	Rectangular Implementation	Decoupled Spherical Implementation
2	64	12
5	15625	75
10	10^6	300
71	$(71)^6$	15123

5.4 Simulation Results

Computer simulations (via CDC 6700) were used to compare the tracking ability of the two adaptive estimators shown in Figures (13) and (14). Target trajectories were generated by the rectangular model of equation (4-2) with $\rho = 0.32$, except for the closing target of experiment #2 which was derived from actual radar data. Radar measurement errors were generated as independent white Gaussian sequences of standard deviation 25 feet, 2 milliradians and 2 milliradians for range, elevation and bearing, respectively. A relatively slow data rate of 0.5 Hz was used to insure sufficient time for implementing the adaptive algorithm.

The primary purpose of these simulations is to determine how well the computationally attractive decoupled spherical adaptive estimator compares with the rectangular (linearized data) adaptive estimator. The comparison consists of a series of experiments involving maneuvering targets. In the experiments, design parameters (such as number of commands modeled) were varied to determine their effects on tracking performance. During this process, it was found that 5 commands per direction was a practical limit (due to computation time) for the rectangular estimator while the spherical estimator could be extended to over 100 commands/direction. The following results are taken from four of the experiments. Note that in every case spherical estimates are transformed to rectangular estimates for the purpose of comparison.

5.4.1 Experiment #1

The first experiment involves an initially unforced target. At time = 20 seconds, maneuver command $u_x = +160$ (corresponding to a velocity command of 500 ft./sec.) is applied. Figures 15, 16 and 17 com-

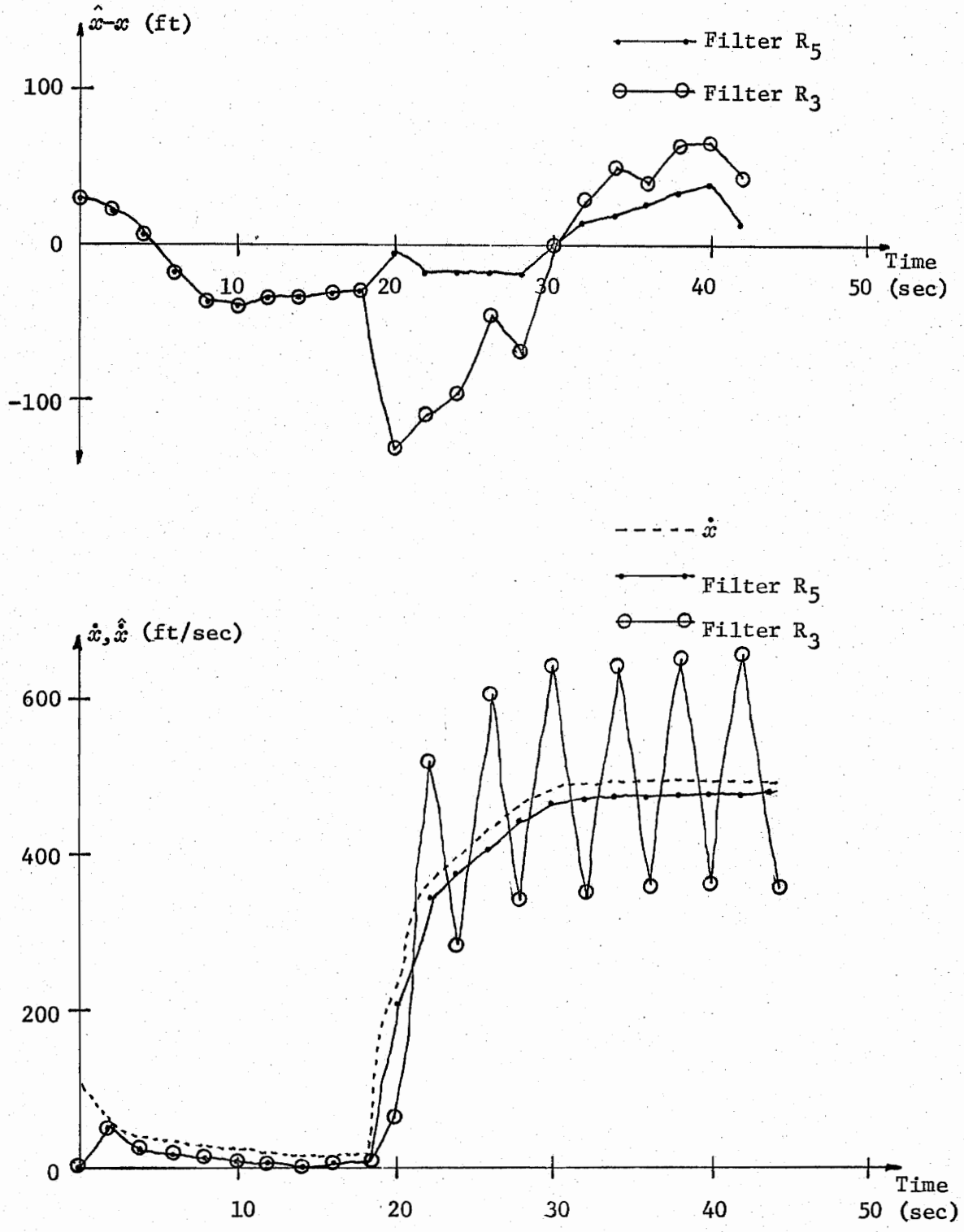


FIGURE 15 x- Direction Estimates

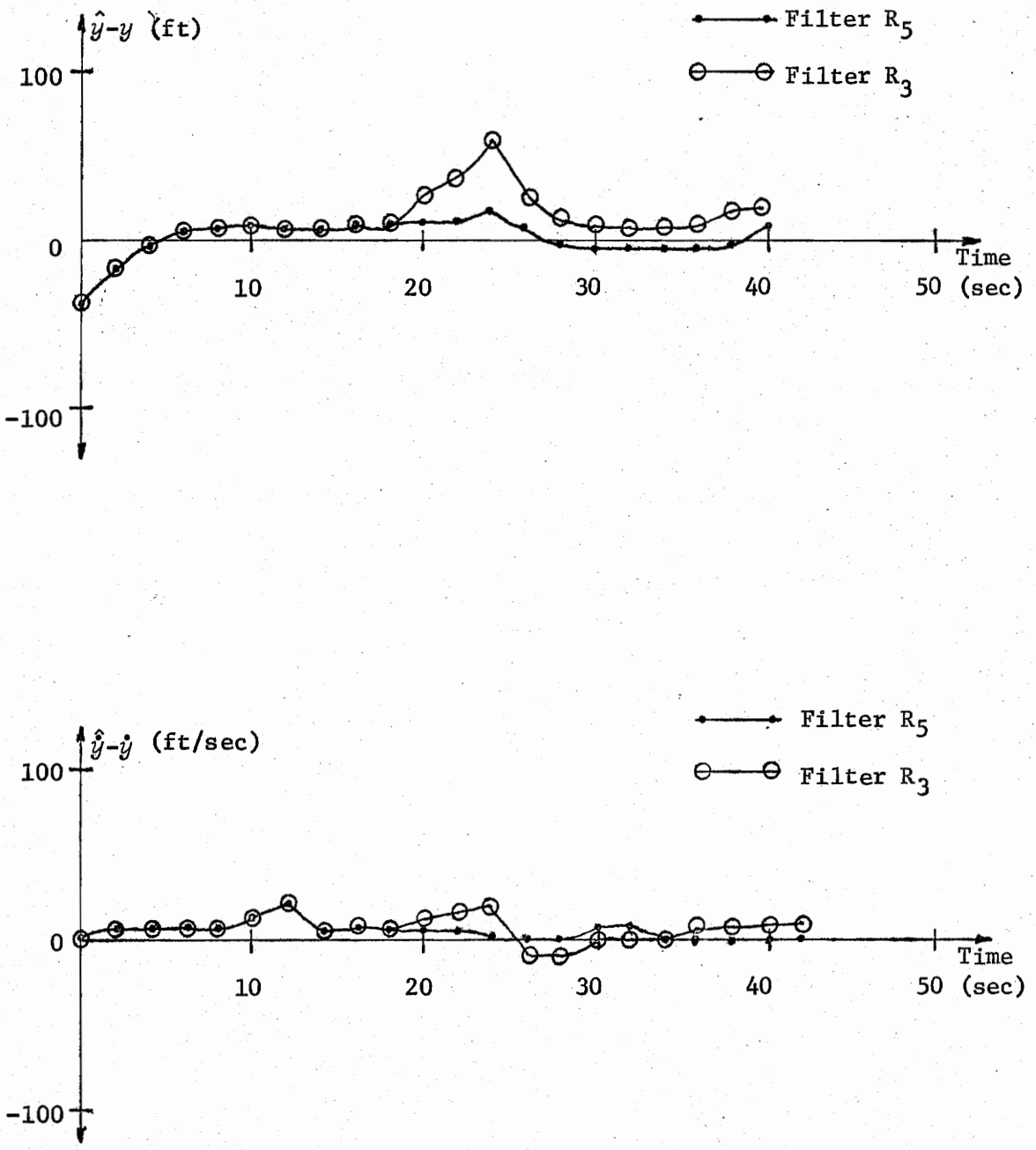
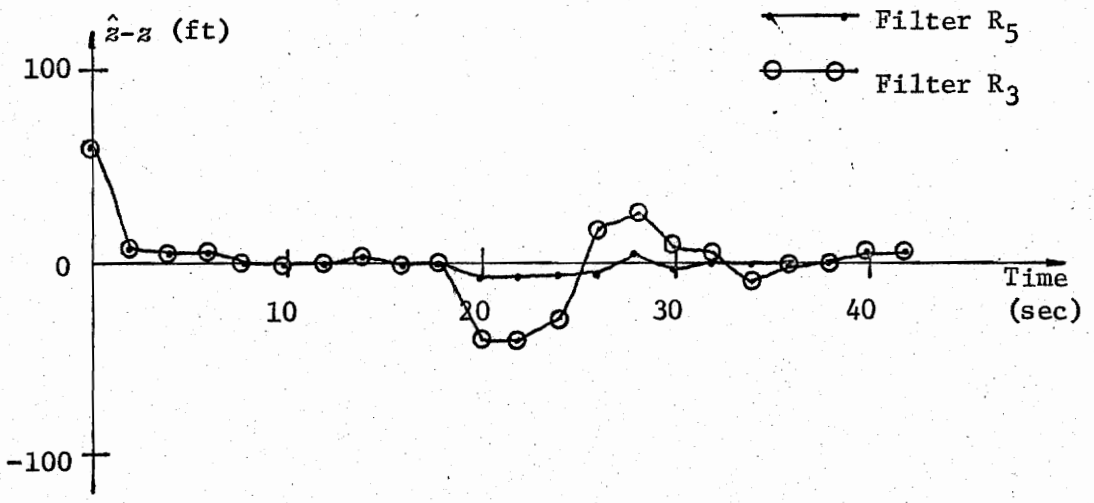


FIGURE 16 y-Direction Estimates



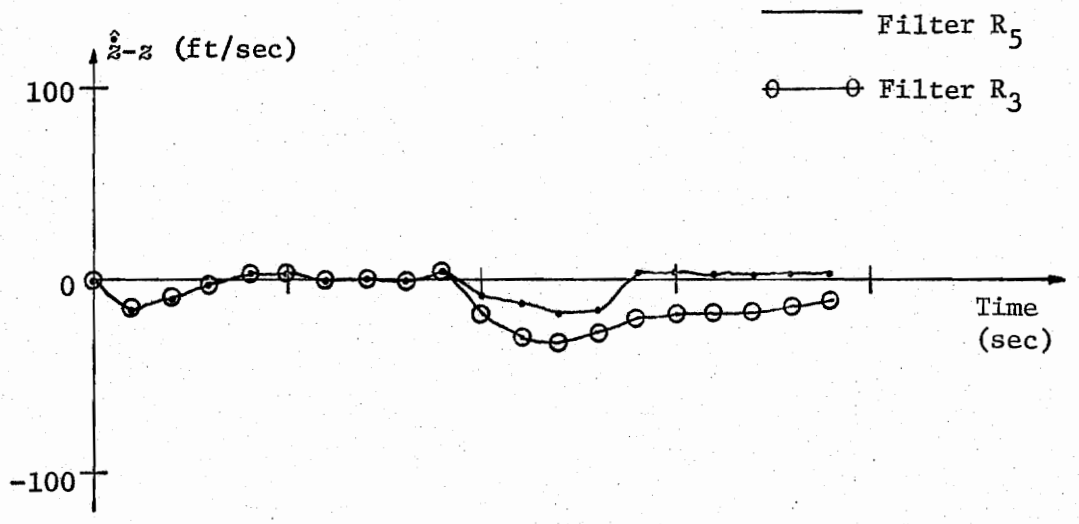


FIGURE 17 z-Direction Estimates

pare the performance of two rectangular (linearized data) adaptive estimators. Filter R_5 models 5 commands per coordinate direction {-320,-160, 0,+160,+320} while filter R_3 models only 3 commands per coordinate direction {-320, 0, +320}. The superior tracking of filter R_5 is due to the maneuver command being a member of filter R_5's command set but not of filter R_3's set. Figure 18 compares the $\hat{u}_x$ of the two estimators. Notice filter R_3 tends to oscillate about the true maneuver command.

5.4.2 Experiment #2 (Actual Radar Data)

The second experiment involves tracking a closing target. The trajectory is virtually motionless in the y and z directions and therefore only x-direction tracking data is presented. Filter R_5 is the same as in the first experiment. Filter S_{11} is an adaptive spherical estimator modeling 11 commands per coordinate direction {-320, -256, -292, ---, 0, ---, 292, 256, 320}. Figure 19 compares the x-velocity tracking of the two filters. Notice that both filters oscillate about the true velocity but filter S_{11}'s oscillation is much less severe.

5.4.3 Experiment #3

This experiment uses filter R_5 and filter S_{11} of second experiment to track a "pop up" target. The trajectory is rather uneventful in the x and y directions and, therefore, only z-direction data is presented. Figure 20 compares the z-velocity tracking performance. As in the previous simulation, filter S_{11}, with more commands modeled, generally outperforms filter R_5. Also, notice the rather severe oscillation of filter R_5. Again, this is due to the maneuver commands not being "near" a modeled command.

5.4.4 Experiment #4

The previous experiments have not revealed the influence of

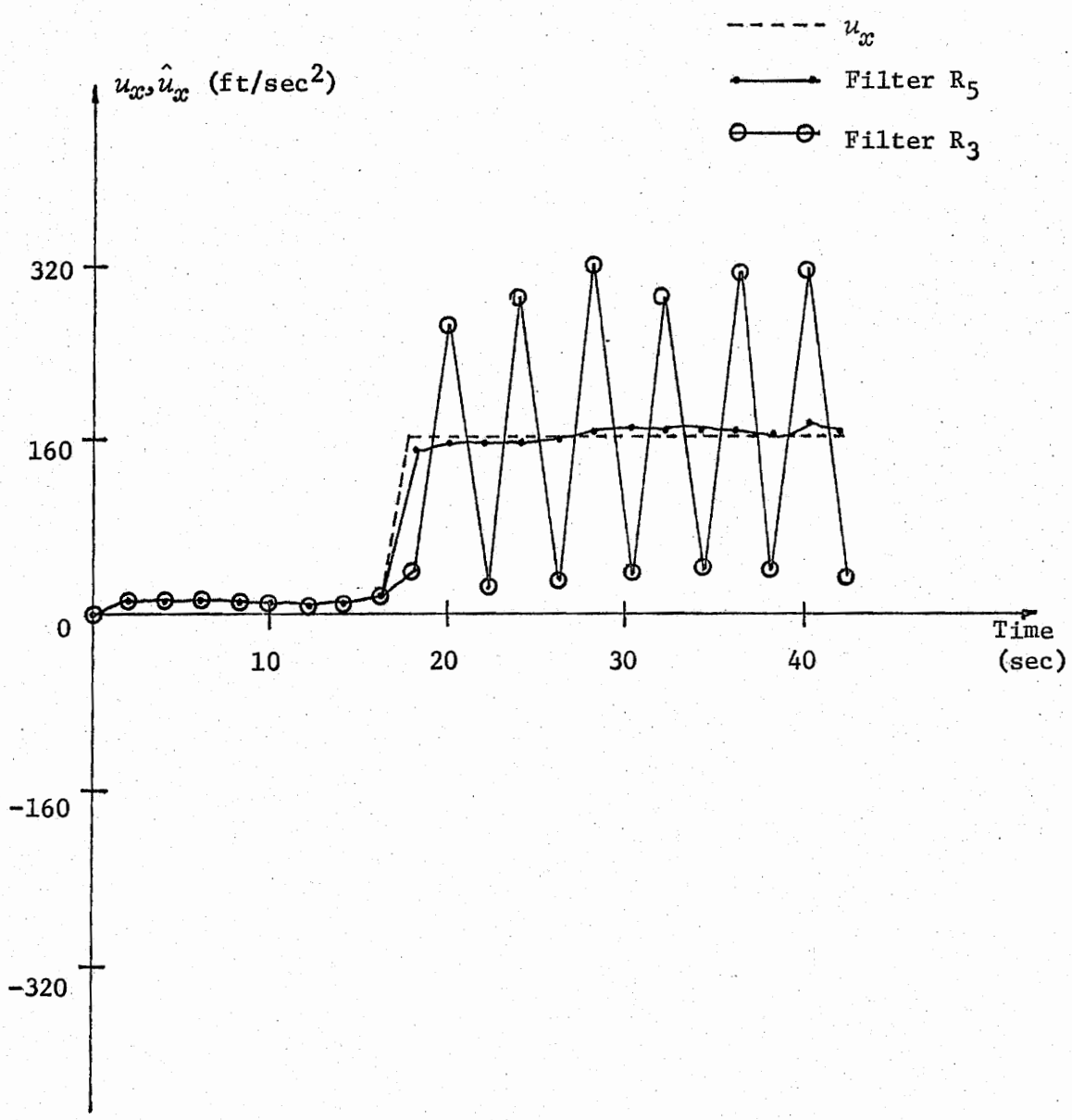


FIGURE 18 $\hat{u}_x$ Estimates

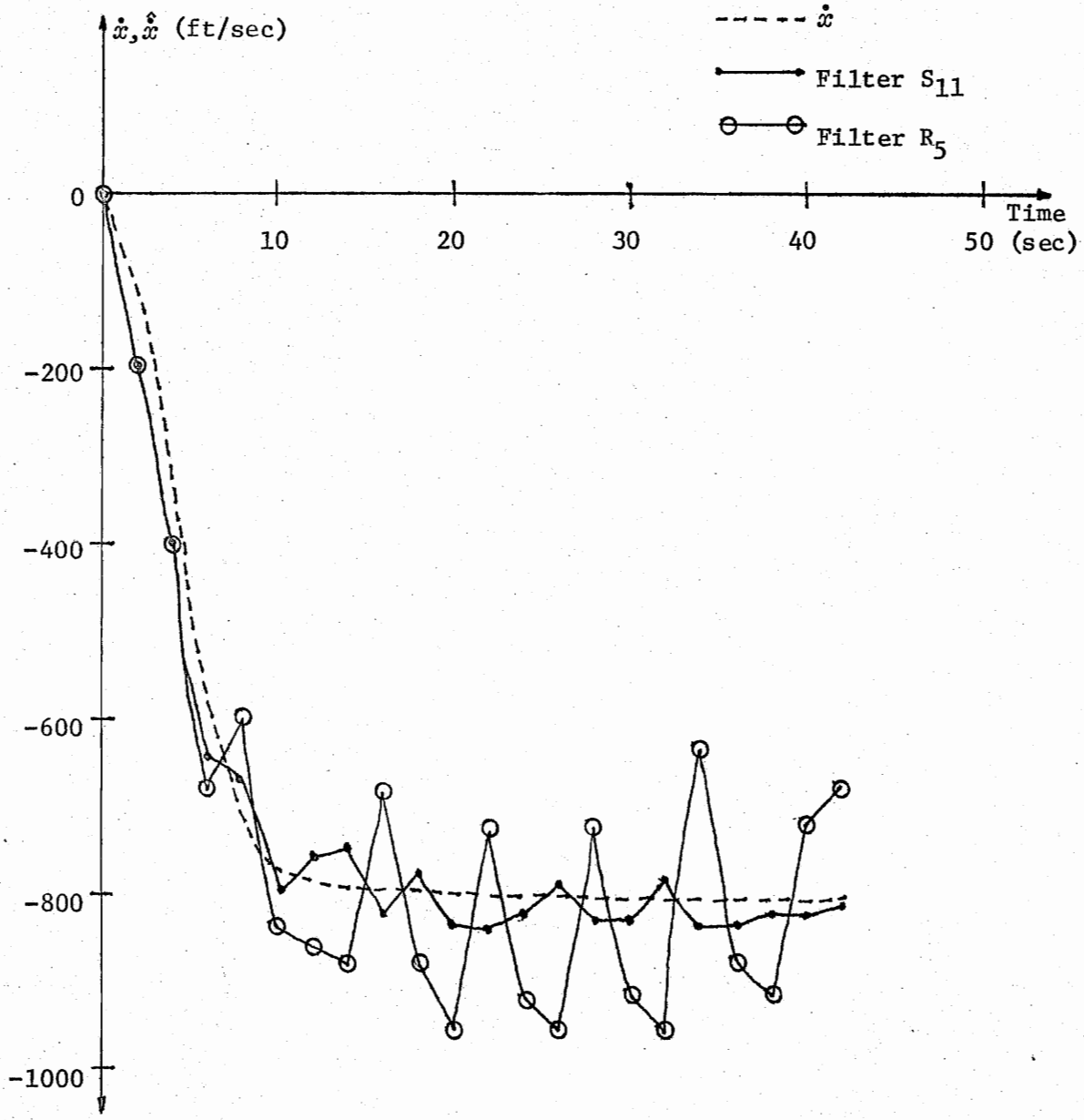


FIGURE 19 x-Direction Estimates

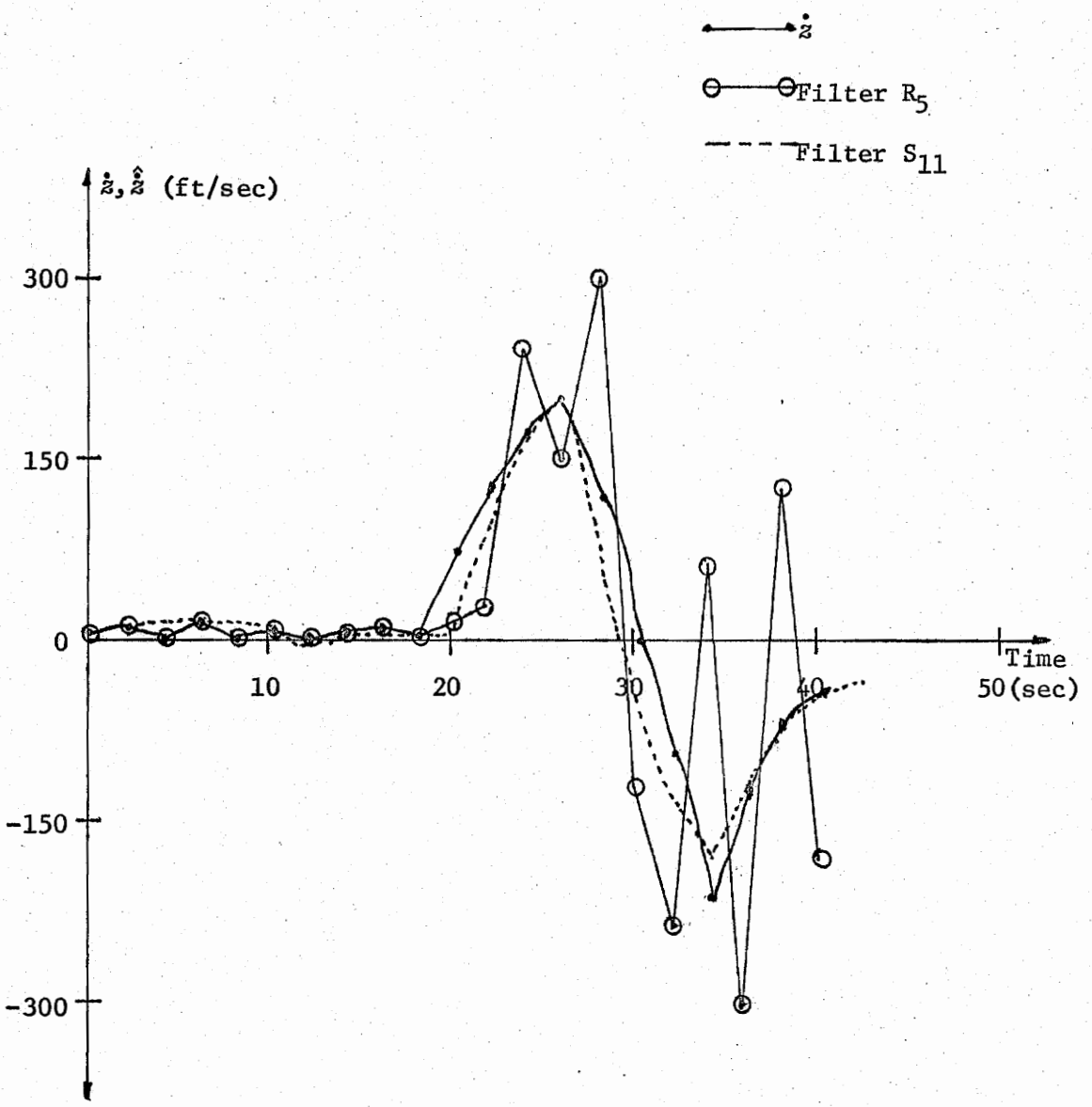


FIGURE 20 z-Direction Estimates

probability matrix θ^* on the estimator's performance characteristics. Repeating the previous experiments using different θ^* matrices would be lengthy and perhaps not as illustrative as the simple experiment presented here. The experiment was conducted in one dimension by using only the r-direction filter of filter S_5 (5 commands modeled). Also, the data rate was increased to 5 Hz to enhance the visibility of "rise time". Figure 21 shows how $\hat{u}$ tracks the deterministic input, u, for two adaptive estimators. Filter S_5' is identical to filter S_5 except the assumed mean random holding time is much greater. The greater holding time has the effect of increasing θ^*_{ii} while decreasing θ^*_{ij} $(i \neq j)$. Notice filter S_5' produces "smoother" estimates of the step input. Also notice that this advantage has been achieved at the expense of much slower "rise time".

5.5 Conclusions from Discrete Semi-Markov Adaptive Experiments

From the experiments conducted, it has been concluded that overall filter performance is a strong function of the number of maneuver commands modeled. The decoupled spherical technique allows many commands to be modeled and was therefore judged the superior implementation of the discrete semi-Markov algorithm.

From the fourth experiment it was concluded that "noise rejection" can be improved by increasing the assumed mean random holding time. This advantage is realized, however, at the expense of poorer response to very rapid changes in maneuver command input. The first conclusion mentioned above is precisely the motivation for developing the continuous semi-Markov algorithm presented in the next chapter.

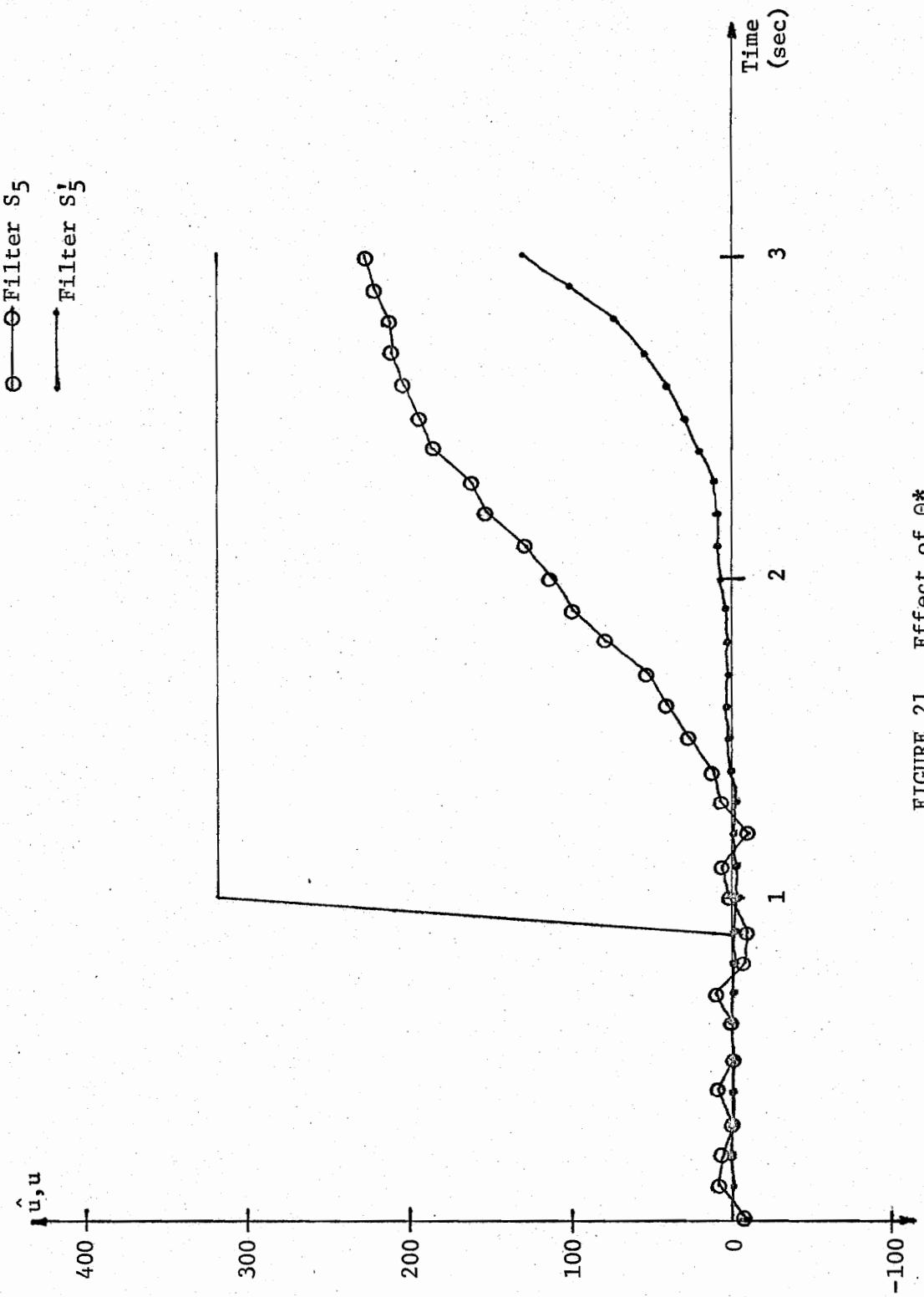


FIGURE 21 Effect of θ^*

6.0 CONTINUOUS SEMI-MARKOV ALGORITHM

6.1 Introduction

From Chapters 4.0 and 5.0, two very important conclusions were reached. The first being that overall adaptive estimator performance can be improved by increasing the number of modeled commands. The second being that computational advantages of the decoupled approximate spherical modeling technique greatly outweigh any performance deficiencies. The first conclusion is precisely the motivation for developing a continuous semi-Markov algorithm that effectively models an infinite number of maneuver commands. The second conclusion was likewise influential in that it justified designing about the much simpler decoupled spherical estimator.

The continuous semi-Markov algorithm not only models an infinite number of maneuver commands but simultaneously reduces computational burden. Also, the algorithm provides error covariance of $\hat{u}$. The importance of this feature will be emphasized later.

6.2 Extension of Discrete Formulation

The continuous algorithm is basically an extension of the discrete algorithm described in Chapter 3.0. For this reason, derivation of the continuous algorithm will proceed in close parallel to Chapter 3.0 and Moose[12]. Notice that in the following derivation, u_k is a continuous random variable at discrete time t_k as opposed to the discrete formulation used earlier. Also, since the decoupled spherical scheme has been adopted, $\underline{x} \triangleq [x|\dot{x}]^T$ where x may be range, elevation or bearing.

Using the continuous form of conditional expectation, $\hat{\underline{x}}_{k+1}$ can be expressed by

$$\underline{\hat{x}}_{k+1} \triangleq E[\underline{x}_{k+1} | \underline{\hat{x}}_k, z_{k+1}] = \int \underline{x} p(\underline{x}_{k+1} = \underline{x} | \underline{\hat{x}}_k, z_{k+1}) d\underline{x} \qquad (6\text{-}1)$$

where $\int_{\underline{x}}(\ )d\underline{x}$ denotes integration over x and $\dot{x}$. Defining $p(u_k = i | \underline{\hat{x}}_k, z_{k+1})$ as the conditional probability density function of u_k given $\underline{\hat{x}}_k$ and z_{k+1}, $p(\underline{x}_{k+1} = \underline{x} | \underline{\hat{x}}_k, z_{k+1})$ of equation (6-1) can be formulated as

$$p(\underline{x}_{k+1} = \underline{x} | \underline{\hat{x}}_k, z_{k+1}) = \int_{i=-\infty}^{i=+\infty} p(\underline{x}_{k+1} = \underline{x} | \underline{\hat{x}}_k, z_{k+1}, u_k = i) p(u_k = i | \underline{\hat{x}}_k, z_{k+1}) di \qquad (6\text{-}2)$$

Notice that both z and u are scalars since the decoupled spherical model has been adopted. Combining equations (6-2) and (6-1) yields upon changing order of integration

$$\underline{\hat{x}}_{k+1} \triangleq \int_{i=-\infty}^{i=+\infty} \underline{\hat{x}}_{k+1}^{(i)} p(u_k = i | \underline{\hat{x}}_k, z_{k+1}) di \qquad (6\text{-}3)$$

where $\underline{\hat{x}}_{k+1}^{(i)}$ is the estimate of a Kalman filter at time t_{k+1} conditioned on $u_k = i$ as well as all available measurement data. Therefore, $\underline{\hat{x}}_{k+1}^{(i)}$ is given by

$$\underline{\hat{x}}_{k+1}^{(i)} \triangleq E[\underline{x}_{k+1} | \underline{\hat{x}}_k, z_{k+1}, u_k = i] = \int_{\underline{x}} \underline{x} p(\underline{x}_{k+1} = \underline{x} | \underline{\hat{x}}_k, z_{k+1}, u_k = i) d\underline{x} \qquad (6\text{-}4)$$

As in the discrete formulation, equation (6-3) represents a bank of Kalman filters. In this case, however, the bank consists of an infinite number of filters. Clearly, it is important that the bank be reduced to a single filter.

Bayes' rule can be used to expand $p(u_k = i | \underline{\hat{x}}_k, z_{k+1})$ of equation (6-3) to yield

$$p(u_k = i | \underline{\hat{x}}_k, z_{k+1}) = \frac{p(z_{k+1} | u_k = i, \underline{\hat{x}}_k) p(u_k = i | \underline{\hat{x}}_k)}{D_{k+1}} \qquad (6\text{-}5)$$

where $D_{k+1} = p(z_{k+1} | \underline{\hat{x}}_k)$.

Expanding the second numerator term of equation (6-5) provides $p(u_k = i|\underline{\hat{x}}_k) = \int_{\alpha=-\infty}^{\alpha=+\infty} p(u_k = i|u_{k-1} = \alpha, \underline{\hat{x}}_k)p(u_{k-1} = \alpha|\underline{\hat{x}}_k)d\alpha$. Realizing that $\underline{\hat{x}}_k$ is optimally obtained from $(\hat{x}_{k-1}, z_k)$, the information $(\underline{\hat{x}}_k)$ can be expressed as $(\hat{x}_{k-1}, z_k)$. Also, from the argument presented to justify equation (3-8), the probability density $p(u_k = i|u_{k-1} = \alpha, \hat{x}_k)$ can be very closely approximated by $p(u_k = i|u_{k-1} = \alpha)$. With this in mind, the above equation can be expressed as

$$p(u_k = i|\underline{\hat{x}}_k) = \int_{\alpha=-\infty}^{\alpha=+\infty} p(u_k = i|u_{k-1} = \alpha)p(u_{k-1} = \alpha|\underline{\hat{x}}_{k-1}, z_k)d\alpha \quad (6\text{-}6)$$

Substituting equation (6-6) into equation (6-5) yields an expression for computing the "weighting" density function needed in equation (6-3).

$$p(u_k = i|\underline{\hat{x}}_k, z_{k+1}) = \frac{p(z_{k+1}|u_k = i, \underline{\hat{x}}_k)\int_{\alpha=-\infty}^{\alpha=+\infty} p(u_k = i|u_{k-1} = \alpha)p(u_{k-1} = \alpha|\hat{x}_{k-1}, z_k)d\alpha}{D_{k+1}} \quad (6\text{-}7)$$

As before, $p(z_{k+1}|u_k = i, \hat{x}_k)$ can be approximated by a Gaussian density function of mean μ_{k+1} variance, c_{k+1} given by

$$\mu_{k+1} = H\phi\underline{\hat{x}}_k + H\Gamma_k i$$

$$c_{k+1} = HP_{k+1|k}H^T + R$$

The variance, c_{k+1}, is provided by the nonadaptive portion of the Kalman filter (see equation (2-15)). The conditional density function $p(u_k = i|u_{k-1} = \alpha)$ is given by $\theta^{**}_{\alpha i}$ where the second superscript * denotes probability density function of u_k as opposed to the discrete probabilities $\theta^*_{\alpha i}$ used in the discrete semi-Markov derivation. Notice that $p(u_{k-1} = \alpha|\underline{\hat{x}}_{k-1}, z_k)$ is just a previously computed "weighting" density function.

Again, D_{k+1} is easily computed from $\int_{i=-\infty}^{i=+\infty} p(u_k = i|\underline{\hat{x}}_k, z_{k+1})di \equiv 1$. Therefore, equation (6-7) provides the recursive formulation for updating the conditional probability density function of u_k. This density is used as a "weighting" function in equation (6-3).

6.2.1 Reduction to A Single Filter

Equation (6-4) represents a Kalman filter conditioned on $u_k = i$. For the case where ϕ, Γ, H and K of the Kalman filter are independent of u, $\underline{\hat{x}}_{k+1}^{(i)}$ becomes

$$\underline{\hat{x}}_{i+1}^{(i)} = \phi\underline{\hat{x}}_k + \Gamma_k i + K_{k+1}[z_{k+1} - H\phi\underline{\hat{x}}_k - H\Gamma_k i].$$

Substituting the above equation into equation (6-3) yields

$$\underline{\hat{x}}_{k+1} = \int_{i=-\infty}^{i=+\infty} [\phi\underline{\hat{x}}_k + \Gamma_k i + K_{k+1}[z_{k+1} - H\phi\underline{\hat{x}}_k - H\Gamma_k i]]p(u_k=i|\underline{\hat{x}}_k, z_{k+1})di$$

which may be rewritten as

$$\underline{\hat{x}}_{k+1} = \phi\underline{\hat{x}}_k + K_{k+1}[z_{k+1} - H\phi\underline{\hat{x}}_k] + (\Gamma_k - K_{k+1}H\Gamma_k)\int_{i=-\infty}^{i=+\infty} ip(u_k = i|\underline{\hat{x}}_k, z_{k+1})di$$

where use has been made of $\int_{i=-\infty}^{i=+\infty} p(u_k = i|\underline{\hat{x}}_k, z_{k+1})di \equiv 1$. By defining $\hat{u}_k$ as the expected value of u_k given $(\underline{\hat{x}}_k, z_{k+1})$, the adaptive estimator can be realized by the single Kalman filter

$$\underline{\hat{x}}_{k+1} = \phi\underline{\hat{x}}_k + \Gamma_k\hat{u}_k + K_{k+1}[z_{k+1} - H\phi\underline{\hat{x}}_k - H\Gamma_k\hat{u}_k] \tag{6-8}$$

where

$$\hat{u}_k = \int_{i=-\infty}^{i=+\infty} ip(u_k = i|\underline{\hat{x}}_k, z_{k+1})di \tag{6-9}$$

and $p(u_k = i|\underline{\hat{x}}_k, z_{k+1})$ is computed recursively from equation (6-7).

6.3 Choice of Θ^{**}

Previously, $\Theta^*_{\alpha i}$ has been computed from transition probabilities and random holding time density functions. Moose[11] points out, however,

that this computation can often be bypassed by using engineering insight to directly choose an appropriate $\theta^{*}_{\alpha i}$. This is precisely the approach used here.

Due to the reduction in computational burden realized by the continuous algorithm, much faster sample rates are possible. As an example, a data rate of 0.5 Hz is practical using the discrete semi-Markov algorithm whereas a data rate of 10 Hz is quite practical using the continuous algorithm. Intuitively, large changes in u during this short sample interval are much less probable than small changes. Therefore, it is reasonable to choose $\theta^{**}_{\alpha i}$ a rapidly decreasing function of $|\alpha-i|$. The particular form of $\theta^{**}_{\alpha i}$ used in this derivation is of Gaussian nature given by

$$\theta^{**}_{\alpha i} \sim \exp\{-s(i-\eta\alpha)^2\} \tag{6-10}$$

where $0 \leq \eta \leq 1$ and $\sim$ denotes "proportional to".

Limiting η to less than unity is necessary to insure the modeling of u is of finite steady state variance. This can be verified by realizing that $\theta^{**}_{\alpha i}$ modeling of u can be statistically represented by the process

$$u_{k+1} = \eta u_k + f_k \tag{6-11}$$

where f_k is a zero mean white process of variance σ_f^2. From equation (6-11) we see that

$$E[u_{k+1}u_{k+1}] = \eta^2 E[u_k u_k] + \sigma_f^2.$$

Solving for steady state variance of u, yields

$$\sigma_{ss}^2 = \frac{\sigma_f^2}{1 - \eta^2}$$

Therefore, η must be limited to less than unity to avoid unrealistic modeling of the maneuver command process.

6.4 Closed Form Solution of Adaptive Algorithm

Implementing equations (6-7) and (6-9) via digital computer is clearly cumbersome if not impossible. If the equations could be solved in closed form, however, a very efficient formulation could be achieved. The purpose of this section is to derive a closed form solution suitable for digital computer implementation.

Assuming the probability density function of u_{k-1} given $\underline{\hat{x}}_{k-1}$ and z_k is Gaussian of mean μ_{k-1} and variance σ^2_{k-1} allows the function to be represented by

$$p(u_{k-1} = \alpha | \underline{\hat{x}}_{k-1}, z_k) \sim \exp\{\frac{-(\alpha - \mu_{k-1})^2}{2\sigma^2_{k-1}}\}$$

where "$\sim$" denoted "proportional to". Clearly, if we define $\hat{u}_{k-1}$ as the expected value of u_{k-1} given $\underline{\hat{x}}_{k-1}$ and z_k, then $\hat{u}_{k-1} = \mu_{k-1}$. Also, the variance of $\hat{u}_{k-1}$ is given by σ^2_{k-1}. The above expression therefore can be written as

$$p(u_{k-1}=\alpha | \underline{\hat{x}}_{k-1}, z_k) \sim \exp\{\frac{-(\alpha - \hat{u}_{k-1})^2}{2\sigma^2_{\hat{u}_{k-1}}}\} . \qquad (6\text{-}12)$$

Substituting expressions (6-12) and (6-10) into equation (6-7) yields

$$p(u_k=i | \underline{\hat{x}}_k, z_{k+1}) \sim p(z_{k+1} | u_k=i, \underline{\hat{x}}_k) \int_{\alpha=-\infty}^{\alpha=+\infty} \exp\{-s(i-\eta\alpha)^2\}$$

$$\cdot \exp\{\frac{-(\alpha - \hat{u}_{k-1})^2}{2\sigma^2_{\hat{u}_{k-1}}}\} \cdot d\alpha$$

where D_{k+1}, not being a function of i, has been absorbed by "$\sim$". From

Papoulis[20], the above convolution can be solved to yield

$$p(u_k=i|\underline{\hat{x}}_k,z_{k+1}) \sim p(z_{k+1}|u_k=i,\underline{\hat{x}}_k)\exp\left[\frac{-(\frac{i}{\eta}-\hat{u}_{k-1})^2}{2\sigma^2_{\hat{u}_{k-1}}+\frac{1}{s\eta^2}}\right] \qquad (6\text{-}13)$$

As discussed earlier, the density $p(z_{k+1}|u_k = i, \underline{\hat{x}}_k)$ is approximately Gaussian of mean = $H\phi\underline{\hat{x}}_k + H\Gamma_k i$ and variance = $HP_{k+1|k}H^T + R$. Denoting the variance by $\sigma^2_{Res_{k+1}}$, the density function can be expressed by

$$p(z_{k+1}|u_k = i,\underline{\hat{x}}_k) \sim \exp\{\frac{-(z_{k+1} - H\phi\underline{\hat{x}}_k - H\Gamma_k i)^2}{2\sigma^2_{Res_{k+1}}}\} \qquad (6\text{-}14)$$

Substituting expression (6-14) into (6-13) results in

$$p(u_k = i|\underline{\hat{x}}_k,z_{k+1}) = (\text{const})\exp\{\frac{-(z_{k+1} - H\phi\underline{\hat{x}}_k - H\Gamma_k i)^2}{2\sigma^2_{Res_{k+1}}}\}$$

$$\cdot \exp\{\frac{-(\frac{i}{\eta}-\hat{u}_{k-1})^2}{2\sigma^2_{\hat{u}_{k-1}}+\frac{1}{s\eta^2}}\} \qquad (6\text{-}15)$$

In Appendix B, it is shown that equation (6-15) can be written in the form

$$p(u_k = i|\underline{\hat{x}}_k,z_{k+1}) = (\text{const}')\exp\{\frac{-(i-\mu_k)^2}{2\sigma^2_k}\} \qquad (6\text{-}16)$$

where $\mu_k = \dfrac{H\Gamma_k[\frac{1}{2s}+\eta^2\sigma^2_{\hat{u}_{k-1}}][z_{k+1} - H\phi\underline{\hat{x}}_k] + \eta\sigma^2_{Res_{k+1}}\hat{u}_{k-1}}{B_k}$

$$\sigma^2_k = \frac{\sigma^2_{Res_{k+1}}[\frac{1}{2s}+\eta^2\sigma^2_{\hat{u}_{k-1}}]}{B_k}$$

$$B_k = (H\Gamma_k)^2 \frac{1}{2s} + \eta^2 (H\Gamma_k)^2 \sigma^2_{\hat{u}_{k-1}} + \sigma^2_{Res_{k+1}}$$

If we again define $\hat{u}_k$ as the expected value of u_k given $\hat{x}_k$, z_{k+1}, then $\hat{u}_k = \mu_k$. Also, the variance of $\hat{u}_k$ is given by σ^2_k. Therefore, equation (6-16) can be written as

$$p(u_k = i \,|\, \underline{\hat{x}}_k, z_{k+1}) = (\text{const}') \exp\left\{\frac{-(i - \hat{u}_k)^2}{2\sigma^2_{\hat{u}_k}}\right\} \qquad (6\text{-}17)$$

Comparing equation (6-17) with (6-12) reveals a very important result. The form of the updated probability density function is identical to its original form. Since the form of $p(z_{k+1} \,|\, u_k = i, \underline{\hat{x}}_k)$ is also time invariant, the following recursive equations can be used to update $\hat{u}$ and $\sigma^2_{\hat{u}}$.

$$\hat{u}_k = \frac{H\Gamma_k \left[\frac{1}{2s} + \sigma^2 \eta^2_{\hat{u}_{k-1}}\right] \left[z_{k+1} - H\phi\underline{\hat{x}}_k\right] + \eta\sigma^2_{Res_{k+1}} \hat{u}_{k-1}}{B_k}$$

$$\sigma^2_{\hat{u}_k} = \frac{\sigma^2_{Res_{k+1}} \left[\frac{1}{2s} + \eta^2 \sigma^2_{\hat{u}_{k-1}}\right]}{B_k} \qquad (6\text{-}18)$$

where $B_k = (H\Gamma_k)^2 \frac{1}{2s} + \eta^2 (H\Gamma_k)^2 \sigma^2_{\hat{u}_{k-1}} + \sigma^2_{Res_{k+1}}$

and $\sigma^2_{Res_{k+1}} = HP_{k+1|k}H^T + R$ (available from non-adaptive portion of the estimator).

6.5 Utilization of $\sigma^2_{\hat{u}}$

The variance, $\sigma^2_{u_k}$, provides an indication of the quality of estimate $\hat{u}_k$. To understand how this parameter might be used, it is necessary to examine the adaptive Kalman filter equation which is repeated below.

$$\underline{\hat{x}}_{k+1} = \phi\underline{\hat{x}}_k + \Gamma_k\hat{u}_k + K_{k+1}[z_{k+1} - H\phi\underline{\hat{x}}_k - H\Gamma_k\hat{u}_k] \tag{6-19}$$

The Kalman gain, K_{k+1}, is computed from the following recursive equations.

$$P_{k+1|k} = \phi P_{k|k}\phi^T + \Gamma_k Q\Gamma_k^T \tag{6-20}$$

$$K_{k+1} = P_{k+1|k}H^T[HP_{k+1|k}H^T + R]^{-1} \tag{6-21}$$

$$P_{k+1|k+1} = [I - K_{k+1}H]P_{k+1|k} \tag{6-22}$$

Equation (6-22) computes the updated error covariance assuming u_k is known exactly. Since this is not true, ($\sigma^2_{\hat{u}_k} \neq 0$), equation (6-22) does not provide an accurate representation of error covariance. Deriving an exact error covariance update equation is quite complicated. In practice, however, an approximate expression has been used with great success. The approximate expression is derived below.

Equation (6-19) can be written in the form

$$\underline{\hat{x}}_{k+1} = (\phi - K_{k+1}H\phi)\underline{\hat{x}}_k + (\Gamma_k - K_{k+1}H\Gamma_k)\hat{u}_k + K_{k+1}z_{k+1}$$

which leads to the error covariance expression

$$P_{k+1|k+1} \triangleq E[(\underline{\hat{x}}_{k+1} - \underline{x}_{k+1})(\underline{\hat{x}}_{k+1} - \underline{x}_{k+1})^T]$$

$$= E\Big[((\phi - K_{k+1}H\phi)\underline{\hat{x}}_k + (\Gamma_k - K_{k+1}H\Gamma_k)\hat{u}_k + K_{k+1}z_{k+1} - \phi\underline{x}_k - \Gamma_k(u_k + w_k))$$

$$\cdot ((\phi - K_{k+1}H\phi)\underline{\hat{x}}_k + (\Gamma_k - K_{k+1}H\Gamma_k)\hat{u}_k + K_{k+1}z_{k+1} - \phi\underline{x}_k - \Gamma_k(u_k + w_k))^T\Big] \tag{6-23}$$

Ignoring cross products involving $\hat{u}_k$, equation (6-23) can be approximated by

$$P_{k+1|k+1} \simeq [I - K_{k+1}H]P_{k+1|k} + (\Gamma_k - K_{k+1}H\Gamma_k)\sigma^2_{\hat{u}_k}(\Gamma_k - K_{k+1}H\Gamma_k)^T \tag{6-24}$$

The form of the adaptive estimator is illustrated by Figure 22.

6.6 Simulation Results

The continuous semi-Markov adaptive estimator was compared with Singer's correlated acceleration estimator, as well as the discrete semi-Markov adaptive estimator. For simplicity, the comparisons are performed in one dimension. This is justifiable since the continuous algorithm is designed for implementation via the decoupled approximate spherical target model. The test trajectories were generated using the rectangular model of equation (4-2). Deterministic commands (u's) were arbitrarily selected to simulate large scale target maneuvers. Due to the reduced computational burden of the continuous semi-Markov algorithm, a date rate of 5 Hz was used. As before, the simulation consists of a series of experiments. Three of these experiments are presented below.

6.6.1 Experiment #1

The first experiment compares the continuous semi-Markov estimator with the discrete semi-Markov estimator. Filter C is a continuous estimator with s and η of equation (6-10) set to 5.0×10^{-2} and 0.997, respectively. Filter D is a discrete semi-Markov estimator modeling the 5-command set {-320, -160, 0, +160, +320}. Initially, the deterministic input is zero. At time = 2.4 seconds, a velocity command of 200 ft/sec (u = +64) is applied. Position and velocity tracking errors are plotted in Figures 23 and 24. Notice the discrete filter performs quite well until the 200 ft/sec velocity command is applied. The oscillation is due to the command (u = +64) not being a member of the modeled set. Also, notice the continuous filter (filter C) shows very little tendency

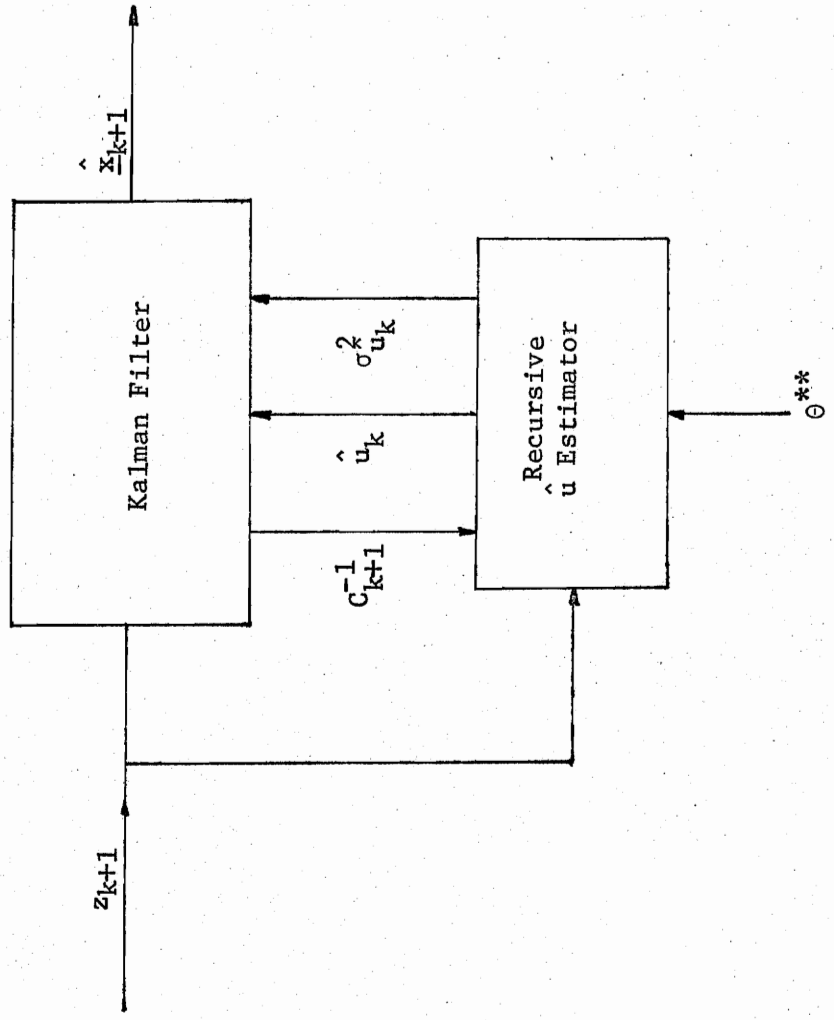


FIGURE 22 Single Dimension Continuous Semi-Markov Adaptive Estimator

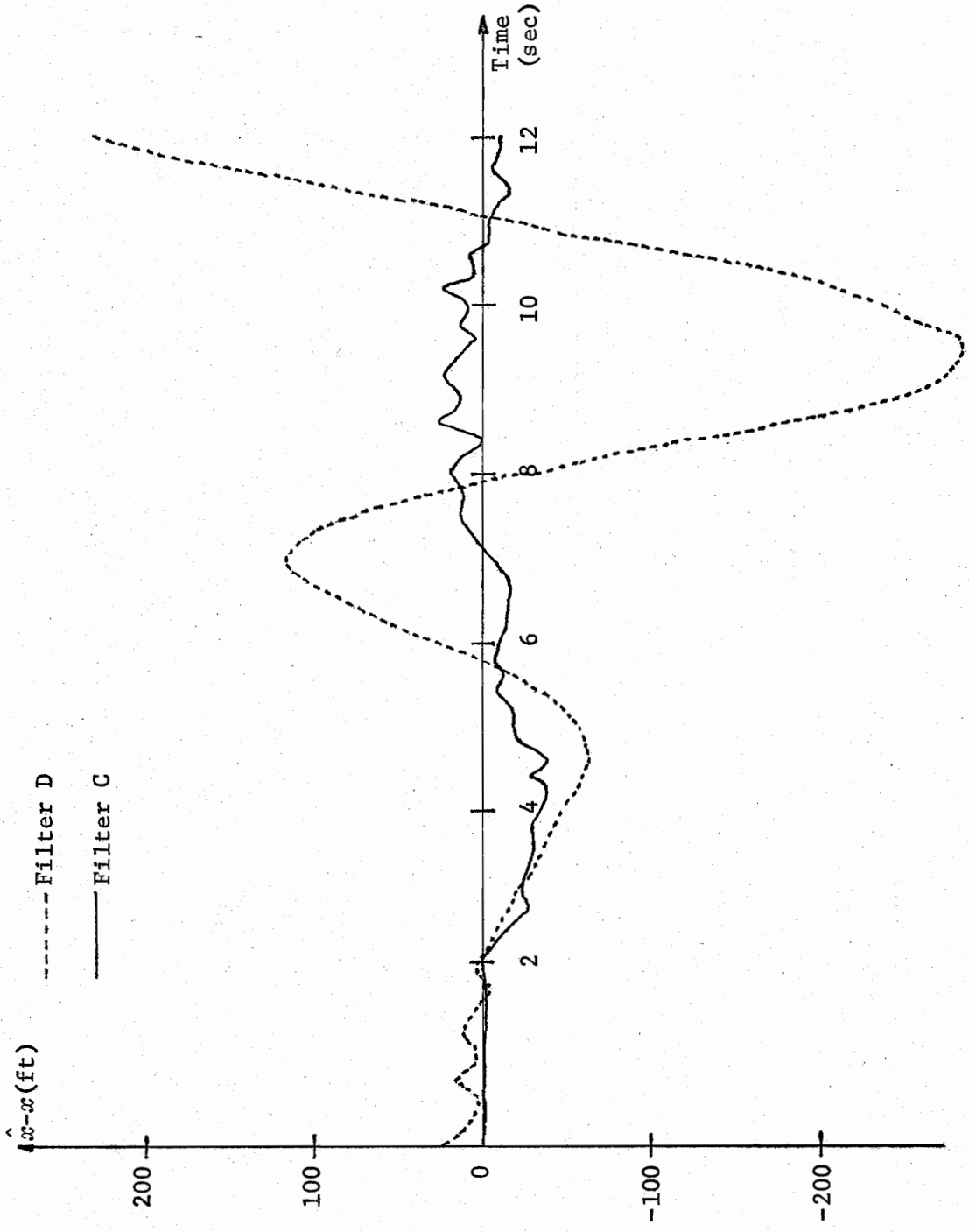


FIGURE 23 Discrete Semi-Markov vs. Continuous Semi-Markov (Position)

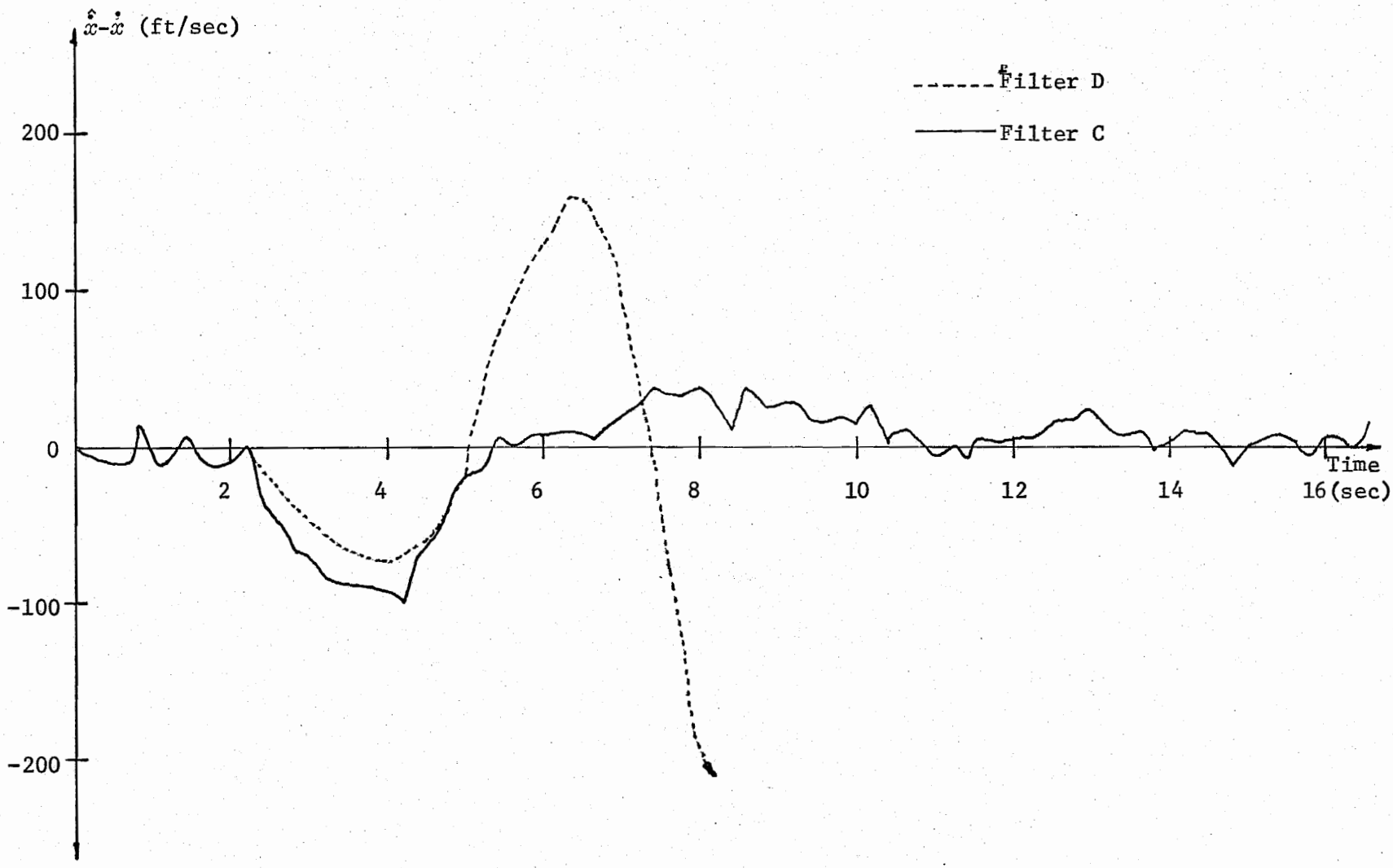


FIGURE 24 Discrete Semi-Markov vs. Continuous Semi-Markov (Velocity)

to oscillate and, therefore, performs quite well over the entire trajectory.

6.6.2 Experiment #2

The second experiment compares Singer's correlated acceleration estimator with filter C of the first experiment. The target performs several maneuvers as outlined in Figures 25 and 26. Notice the singer filter (filter S) behaves very much like the continuous semi-Markov filter. An important difference, however, is that the continuous semi-Markov algorithm achieves this performance without requiring state augmentation.

6.6.3 Experiment #3

This experiment serves to illustrate the influence of θ^{**} of equation (6-10) on the continuous semi-Markov filter performance. Filter C is the same as in the first experiment. Filter C' is also the same except s of equation (6-10) is 1.33×10^{-3}. This parameter change has the effect of modeling a larger steady state variance of u. The test trajectory is unforced until u = +200 is applied at time = 2.7 seconds. Figure 27 compares $\hat{u}$ estimates of the two filters. Notice filter C' provides quicker "rise time" at the expense of "noisier" estimates of a step input. This performance compromise is exactly that observed in Chapter 5.0 experiments concerning the discrete semi-Markov estimator.

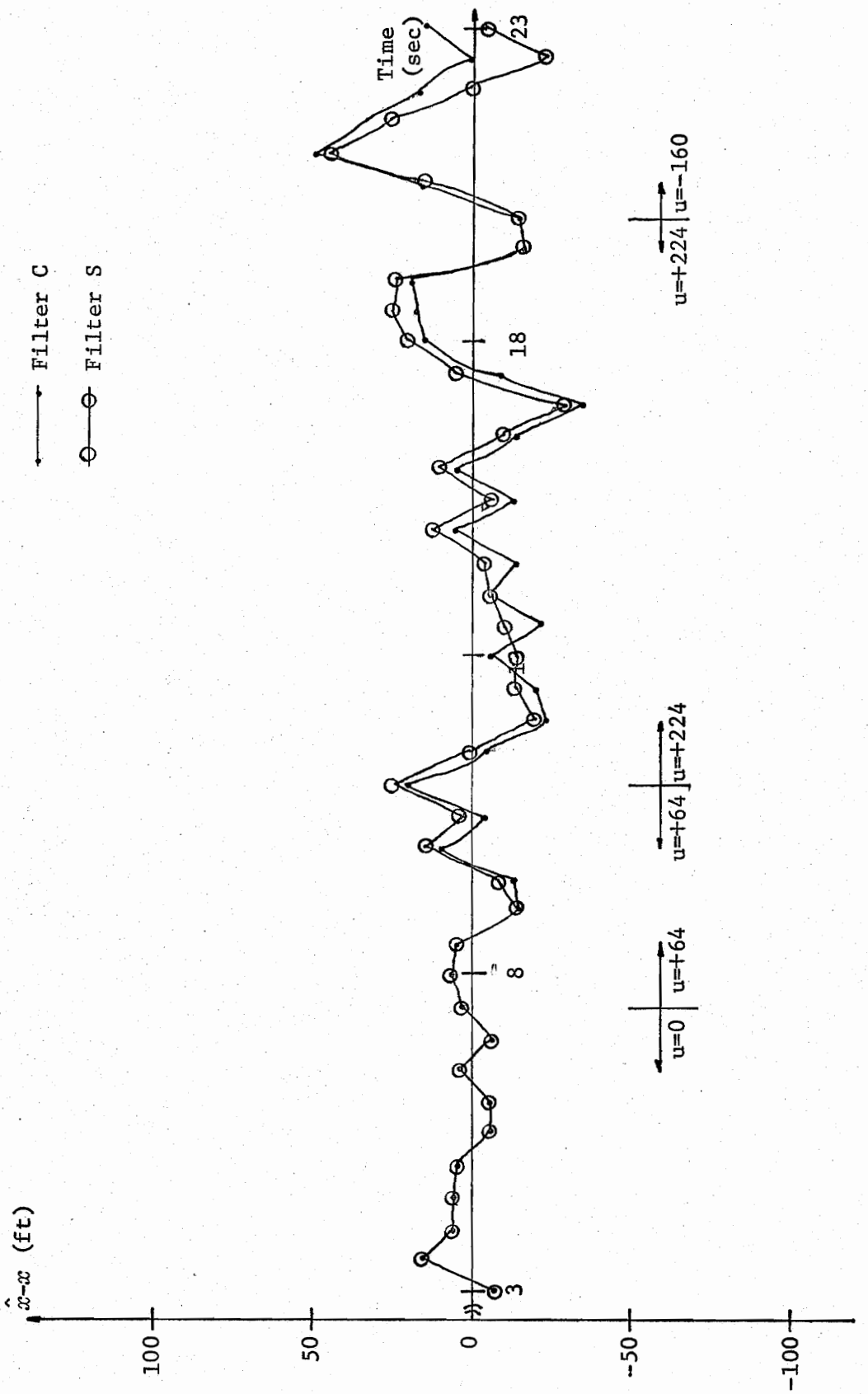


FIGURE 25 Singer vs. Continuous Semi-Markov

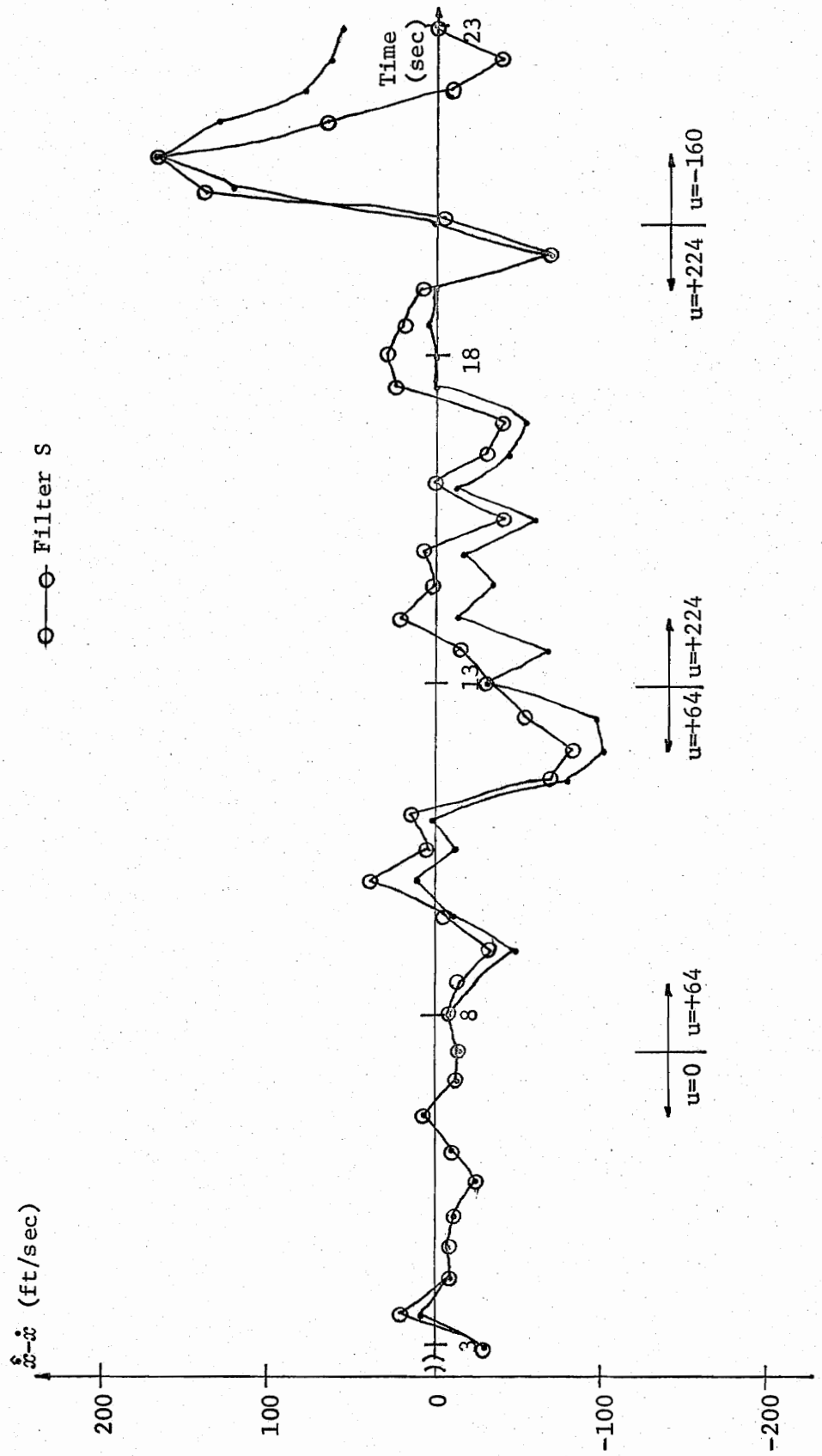


FIGURE 26 Singer vs. Continuous Semi-Markov

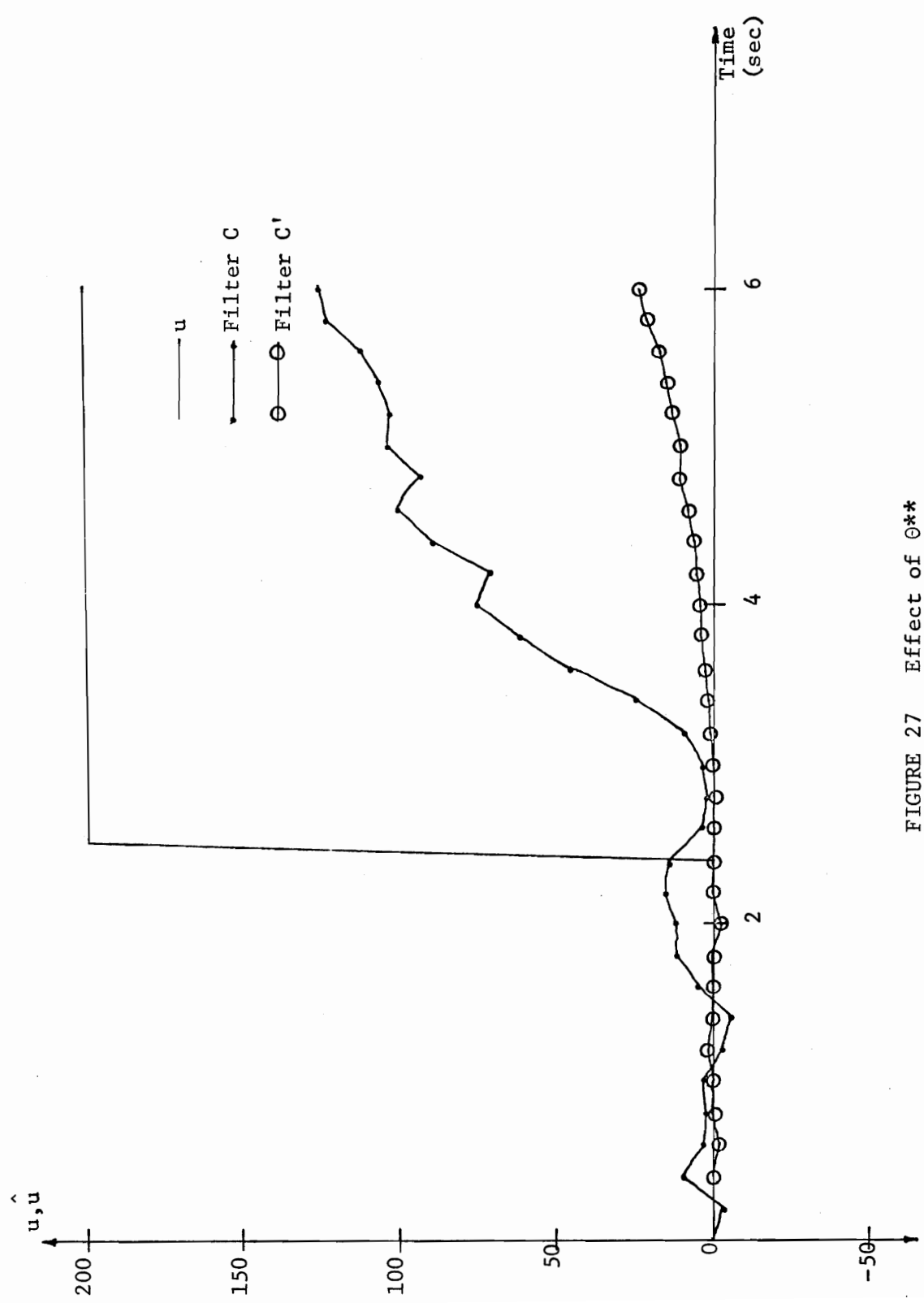


FIGURE 27 Effect of Θ**

7.0 SUMMARY AND CONCLUSIONS

This dissertation addressed the problem of tracking maneuvering targets via radar data. Several techniques were explored, all of which rely on semi-Markov modeling of target maneuvers. These techniques were compared via computer simulation with emphasis on computational burden as well as tracking performance.

The discrete semi-Markov adaptive algorithm was extended from Moose's planar application to three dimensional tracking. Implementation via rectangular and decoupled approximate spherical modeling was investigated. It was found in both cases that overall filter performance could be improved by increasing the number of maneuver commands modeled. For this reason, the spherical implementation was judged the superior since it allowed many more commands to be modeled before computational burden became excessive.

The continuous semi-Markov adaptive algorithm was developed as an extension of the discrete algorithm. In the extension, summations became integrals while discrete probabilities became continuous probability density functions. The extension resulted in a maneuver command estimate equation involving double integrals as opposed to the double summations associated with the original formulation. The integral expression is very difficult, if not impossible, to evaluate via digital computer. By choosing an appropriate semi-Markov probability function, however, it was possible to solve the integrals in closed form. This result led to a very efficient recursive formulation that substantially reduced computational burden associated with the adaptive portion of the estimator.

Also, the continuous formulation provides the variance of the estimated maneuver command. This variance has been used to compute more optimal Kalman gains by modifying the error covariance update equation. Simulation results have indicated that the continuous algorithm performs much better than the discrete algorithm in circumstances where the maneuver commands are not accurately described by a finite set of discrete levels. Simulations comparing the continuous semi-Markov algorithm with Singer's correlated acceleration algorithm shows the performances to be very similar. An important difference, however, is that the continuous semi-Markov adaptive algorithm achieves this performance without requiring state augmentation.

This dissertation explored the use of semi-Markov maneuver modeling for three dimensional target tracking. Much work, however, is left to future researchers interested in discrete or continuous semi-Markov modeling.

The discrete algorithm, in its present state, suffers from oscillation whenever the maneuver command is not "near" a modeled command. The oscillation generally has a mean value near the desired estimate. Perhaps this property could be exploited by "low pass filtering" the maneuver estimates and thus eliminating the ill effects of oscillation. If such a scheme were feasible, it may be possible to effectively implement the discrete semi-Markov algorithm using a very small number of modeled maneuver commands.

The continuous semi-Markov algorithm was derived in closed form for a particular semi-Markov probability density function. Closed form solutions for different density functions would definitely be a major

contribution in this area. Such solutions would allow the continuous semi-Markov adaptive algorithm to offer many features in addition to those already mentioned.

In summary, the research described in this dissertation has developed semi-Markov adaptive filter algorithms that quite effectively track maneuvering targets in three dimensions. It is hoped that others will continue in this area to produce worthwhile contributions in the field of adaptive state estimation.

APPENDIX A

Spherical State Models

This appendix derives the spherical approximate target state models (coupled and decoupled) used in the spherical tracking algorithms. For convenience, the r-direction expression of equation (4-9) is repeated below

$$r_{k+1} \approx r_k + \frac{\partial r}{\partial \underline{x}^T}\Bigg|_{\underline{x}_k} (\underline{x}_{k+1} - \underline{x}_k) \tag{A-1}$$

where $\underline{x}_k = [x|\dot{x}|y|\dot{y}|z|\dot{z}]^T$

$r = \sqrt{x^2+y^2+z^2}$

Expanding the vector notation of equation (A-1) and performing the indicated partial derivatives results in

$$r_{k+1} \approx r_k + \frac{x_k}{\sqrt{x_k^2+y_k^2+z_k^2}} (x_{k+1} - x_k) + \frac{y_k}{\sqrt{x_k^2+y_k^2+z_k^2}} (y_{k+1} - y_k)$$

$$+ \frac{z_k}{\sqrt{x_k^2+y_k^2+z_k^2}} (z_{k+1} - z_k) \tag{A-2}$$

From equation (4-2) the following expressions can be easily obtained

$$\begin{aligned} x_{k+1} - x_k &= P_1 \dot{x}_k + P_2 u_{x_k} \\ y_{k+1} - y_k &= P_1 \dot{y}_k + P_2 u_{y_k} \\ z_{k+1} - z_k &= P_1 \dot{z}_k + P_2 u_{z_k} \end{aligned} \tag{A-3}$$

The expressions P_1 and P_2 are defined in equation (4-2). Substituting equation (A-3) into equation (A-2) yields

$$r_{k+1} \approx r_k + P_1 \left[\frac{x_k \dot{x}_k}{\sqrt{x_k^2+y_k^2+z_k^2}} + \frac{y_k \dot{y}_k}{\sqrt{x_k^2+y_k^2+z_k^2}} + \frac{z_k \dot{z}_k}{\sqrt{x_k^2+y_k^2+z_k^2}} \right]$$

$$+ \frac{P_2}{\sqrt{x_k^2+y_k^2+z_k^2}} \left[x_k u_{x_k} + y_k u_{y_k} + z_k u_{z_k} \right] \qquad \text{(A-4)}$$

By noting the coefficient of P_1 in equation (A-4) is identically $\dot{r}$ at time t_k, the following approximate discrete time model results for the r-direction

$$r_{k+1} \approx r_k + P_1 \dot{r}_k + \frac{P_2}{\sqrt{x_k^2+y_k^2+z_k^2}} \left[x_k u_{x_k} + y_k u_{y_k} + z_k u_{z_k} \right] \qquad \text{(A-5)}$$

which leads directly to

$$\dot{r}_{k+1} \approx P_1^* \dot{r}_k + \frac{P_1}{\sqrt{x_k^2+y_k^2+z_k^2}} \left[x_k u_{x_k} + y_k u_{y_k} + z_k u_{z_k} \right] \qquad \text{(A-6)}$$

where P_1^* is defined in equation (4-2).

Using the procedure outlined above, approximate expressions can be derived for the elevation and bearing coordinates. For brevity, only the results are presented below

$$e_{k+1} \approx e_k + P_1 \dot{e}_k + P_2 \frac{\left[-x_k z_k u_{x_k} - y_k z_k u_{y_k} + (x_k^2+y_k^2) u_{z_k} \right]}{(x_k^2+y_k^2+z_k^2)\sqrt{x_k^2+y_k^2}}$$

$$\dot{e}_{k+1} \approx P_1^* \dot{e}_k + P_1 \frac{\left[-x_k z_k u_{x_k} - y_k z_k u_{y_k} + (x_k^2+y_k^2) u_{z_k} \right]}{(x_k^2+y_k^2+z_k^2)\sqrt{x_k^2+y_k^2}} \qquad \text{(A-7)}$$

$$b_{k+1} \approx b_k + P_1 \dot{b}_k + \frac{P_2}{x_k^2+y_k^2} \left[-y_k u_{x_k} + x_k u_{y_k} \right]$$

$$\dot{b}_{k+1} \approx P_1^* \dot{b}_k + \frac{P_1}{x_k^2 + y_k^2} \left[-y_k u_{x_k} + x_k u_{y_k} \right] \qquad \text{(A-7)}$$

Equations (A-7) have a linear form and can therefore be expressed in linear state space notation by

$$\underline{s}_{k+1} \approx \phi^{(s)} \underline{s}_k + \Psi_k^{(s)} (\underline{u}_k + \underline{w}_k) \qquad \text{(A-8)}$$

where $\underline{s} = [r|\dot{r}|e|\dot{e}|b|\dot{b}]^T$

$\underline{u} = [u_x|u_y|u_z]$

$w = [w_x|w_y|w_z]$

$$\phi^{(s)} = \begin{bmatrix} 1 & P_1 & 0 & 0 & 0 & 0 \\ 0 & P_1^* & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & P_1 & 0 & 0 \\ 0 & 0 & 0 & P_1^* & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & P_1 \\ 0 & 0 & 0 & 0 & 0 & P_1^* \end{bmatrix}$$

$$\Psi_k^{(s)} = \begin{bmatrix} P_2 \frac{x_k}{D_k} & P_2 \frac{y_k}{D_k} & P_2 \frac{z_k}{D_k} \\ P_1 \frac{x_k}{D_k} & P_1 \frac{y_k}{D_k} & P_1 \frac{z_k}{D_k} \\ \frac{-P_2 x_k z_k}{D_k^2 C_k} & \frac{-P_2 y_k z_k}{D_k^2 C_k} & \frac{P_2(x_k^2+y_k^2)}{D_k^2 C_k} \\ \frac{-P_1 x_k z_k}{D_k^2 C_k} & \frac{-P_1 y_k z_k}{D_k^2 C_k} & \frac{P_1(x_k^2+y_k^2)}{D_k^2 C_k} \\ \frac{-P_2 y_k}{x_k^2+y_k^2} & \frac{P_2 x_k}{x^2{}_k+y^2{}_k} & 0 \\ \frac{-P_1 y_k}{x^2{}_k+y^2{}_k} & \frac{P_1 x_k}{x^2{}_k+y^2{}_k} & 0 \end{bmatrix}$$

$$D_k = \sqrt{x_k^2 + y_k^2 + z_k^2}$$

$$C_k = \sqrt{x_k^2 + y_k^2}$$

Decoupled Spherical Model

From equation (A-8) it can be seen that range elevation and bearing are coupled by terms in the matrix $\Psi^{(s)}$. This coupling can be eliminated, however, by assuming the spherical form for $\underline{w}$

$$\underline{w} = [w_r | w_e | w_b]^T$$

where w_r, w_e, w_b are independent forces acting in the range, elevation and bearing directions respectively. The maneuver command vector, $\underline{u}$,

is likewise modeled in spherical form resulting in

$$\underline{u} = [u_r | u_e | u_b]$$

From Figure A-1, the spherical force components can be expressed in rectangular coordinates to yield

$$u_r = u_x \text{ Cos b Cos e} + u_y \text{ Sin b Cos e} + u_z \text{ Sin e}$$

$$u_e = -u_x \text{Cos b Sin e} - u_y \text{ Sin b Sin e} + u_z \text{ Cos e}$$

$$u_b = -u_x \text{Sin b} + u_y \text{ Cos b}$$

Substituting for the trigometric functions

$$u_r = \frac{x\, u_x}{\sqrt{x^2+y^2+z^2}} + \frac{y\, u_y}{\sqrt{x^2+y^2+z^2}} + \frac{z\, u_z}{\sqrt{x^2+y^2+z^2}}$$

$$u_e = \frac{-xz\, u_x}{\sqrt{x^2+y^2}\,\sqrt{x^2+y^2+z^2}} - \frac{yz\, u_y}{\sqrt{x^2+y^2}\,\sqrt{x^2+y^2+z^2}} + \frac{\sqrt{x^2+y^2}\, u_z}{\sqrt{x^2+y^2+z^2}} \qquad \text{(A-9)}$$

$$u_b = \frac{-y\, u_x}{\sqrt{x^2+y^2}} + \frac{x\, u_y}{\sqrt{x^2+y^2}}$$

Comparing equations (A-9) with (A-8) reveals that the approximate spherical model can be expressed in decoupled form by

$$\underline{s}_{k+1} \approx \phi^{(s)}\, \underline{s}_k + \Gamma_k^{(s)}\, (\underline{u}_k + \underline{w}_k)$$

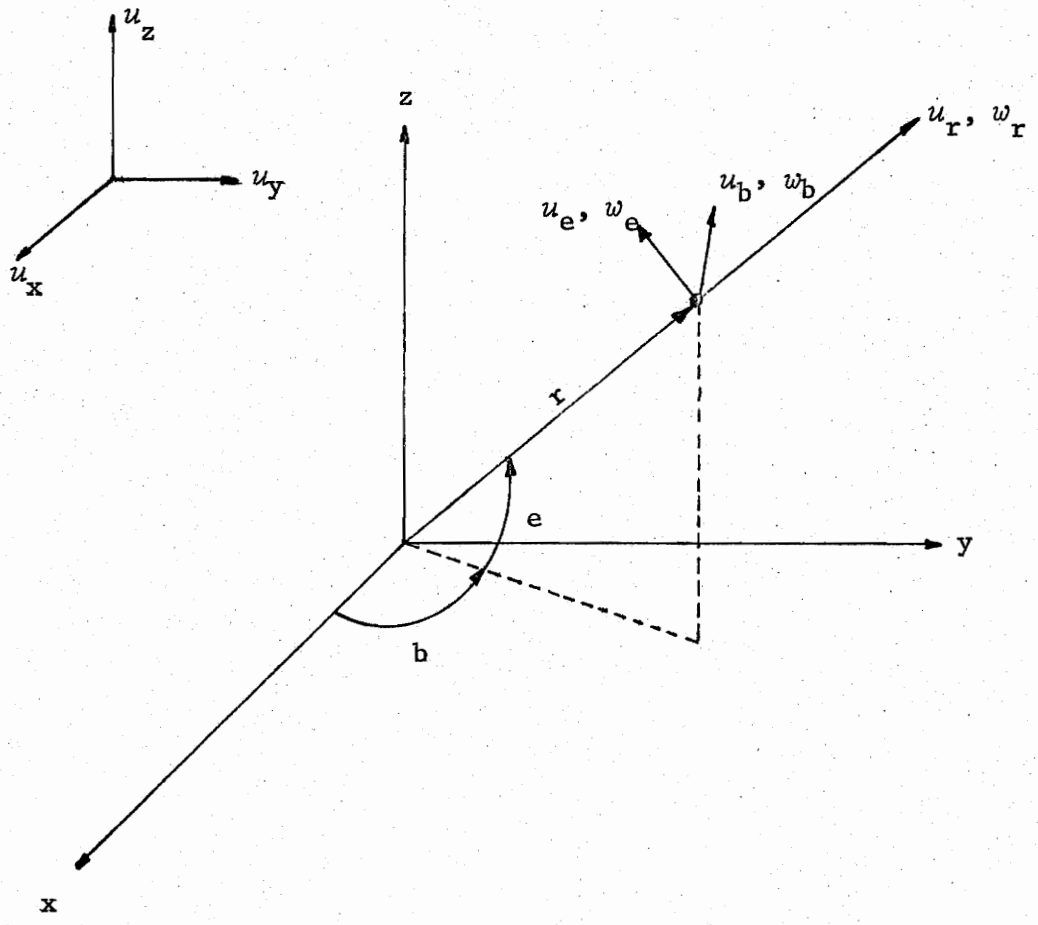


FIGURE A-1 Spherical Force Components In Rectangular Coordinates

where

$$\Gamma_k^{(s)} = \begin{bmatrix} P_2 & 0 & 0 \\ P_1 & 0 & 0 \\ 0 & \dfrac{P_2}{\sqrt{x_k^2+y_k^2+z_k^2}} & 0 \\ 0 & \dfrac{P_1}{\sqrt{x_k^2+y_k^2+z_k^2}} & 0 \\ 0 & 0 & \dfrac{P_2}{\sqrt{x_k^2+y_k^2}} \\ 0 & 0 & \dfrac{P_1}{\sqrt{x_k^2+y_k^2}} \end{bmatrix}$$

$$\underline{u} = [u_r | u_e | u_b]^T$$

$$\underline{w} = [w_r | w_e | w_b]^T$$

APPENDIX B

Derivation of Equation (6-16)

Equation (6-15) is presented below using a more compact notation.

$$p(u_k=i|\hat{\underline{x}}_k, z_{k+1}) = (\text{const}) \exp\left\{\frac{-(R-\gamma_i)^2}{2\sigma_R^2}\right\} \exp\left\{\frac{(\frac{i}{\eta}-m)^2}{2\beta^2}\right\} \qquad \text{(B-1)}$$

where

$$R = z_{k+1} - H\,\phi\,\hat{x}_k$$

$$\gamma = H\,\Gamma_k$$

$$\sigma_R^2 = \sigma_{Res_{k+1}}$$

$$m = \hat{u}_{k-1}$$

$$\beta^2 = \sigma^2_{\hat{u}_{k-1}} + \frac{1}{2s\eta^2}$$

Rewriting the product of exponents in terms of a single exponent yields

$$p(u_k=i|\hat{\underline{x}}_k, z_{k+1}) = (\text{const}) \exp\left\{\frac{-1}{2\sigma_R^2\beta^2}\left[\beta^2(R-\gamma_i)^2 + \sigma_R^2\,(\tfrac{i}{\eta} - m)^2\right]\right\} \qquad \text{(B-2)}$$

Expanding the expressions $(R-\gamma_i)^2$ and $(\frac{i}{\eta}-m)^2$ of equation (B-2) results in

$$p(u_k=i|\hat{\underline{x}}_k, z_{k+1}) = (\text{const}) \exp\left\{\frac{-1}{2\sigma_R^2\beta^2}\left[\beta^2\,R^2 + \sigma_R^2\,m^2\right] - \tfrac{1}{2}\,f(i)\right\} \qquad \text{(B-3)}$$

where $f(i) = i^2\,(\gamma^2\beta^2 + \frac{\sigma_R^2}{\eta^2}) - 2i\,(\gamma\beta^2 R + \sigma_R^2\,\frac{m}{\eta})\,/(\sigma_R^2\beta^2)$

Equation (B-3) can now be written in the form

$$p(u_k=i|\hat{\underline{x}}_k,\, z_{k+1}) = (\text{const}^*) \exp\left\{-\tfrac{1}{2}\,f(i)\right\} \qquad \text{(B-4)}$$

where $\exp\{-\frac{1}{2\sigma^2}\beta^2[\beta^2R^2+\sigma_R^2 m^2]\}$, not being a function of i, has been absorbed in (const*). By defining two more dummy variables, the expression for f(i) may be written as

$$f(i) = \frac{(i-\frac{\alpha}{\omega})^2 - (\frac{\alpha^2}{\omega^2})}{\sigma_R^2 \beta^2/\omega} \qquad \text{(B-5)}$$

where

$$\alpha = \gamma \beta^2 R + \frac{\sigma_R^2 m}{\eta}$$

$$\omega = \gamma^2\beta^2 + \frac{\sigma_R^2}{\eta^2}$$

Substituting equation (B-5) into equation (B-4) yields

$$p(u_k=i|\hat{\underline{x}}_k, z_{k+1}) = (\text{const}^*) \exp\{-\frac{\omega}{2\sigma_R^2\beta^2}(i-\frac{\alpha}{\omega})^2 + \frac{\alpha^2}{2\omega\beta^2\sigma_R^2}\}$$

which may be rewritten as

$$p(u_k=i|\hat{\underline{x}}_k, z_{k+1}) = (\text{const}') \exp\{\frac{-(i-\mu_k)^2}{2\sigma_k^2}\}$$

where $\exp\{\frac{\alpha^2}{2\omega\beta^2\sigma_R^2}\}$ has been absorbed by (const')

and $\sigma_k^2 = \sigma_R^2 \beta^2/\omega$

$$\mu_k = \frac{\alpha}{\omega}$$

Removing the dummy variables, the derivation is concluded by

$$p(u_k = i|\hat{\underline{x}}_k, z_{k+1}) = (\text{const}') \exp\{\frac{-(i-\mu_k)^2}{2\sigma_k^2}\} \qquad \text{(B-6)}$$

where $\sigma_k^2 = \dfrac{\sigma^2_{Res_{k+1}} \left(\frac{1}{2s} + \eta^2 \sigma^2_{\hat{u}_{k+1}} \right)}{(H\Gamma_k)^2 \left(\eta^2 \sigma^2_{\hat{u}_{k-1}} + \frac{1}{2s}\right) + \sigma^2_{Res_{k+1}}}$

$$\mu_k = \frac{H\Gamma_k \left[\frac{1}{2s} + \eta^2 \sigma^2_{\hat{u}_{k-1}} \right] \left[z_{k+1} - H\phi \hat{\underline{x}}_k \right] + \eta \sigma^2_{Res_{k+1}} \hat{u}_{k-1}}{(H\Gamma_k)^2 \left(\eta^2 \sigma^2_{\hat{u}_{k-1}} + \frac{1}{2s}\right) + \sigma^2_{Res_{k+1}}}$$

BIBLIOGRAPHY

1. Weiner, N. The Extrapolation, Interpolation and Smoothing of Stationary Time Series, John Wiley & Sons, Inc., N.Y., 1949.

2. Kolmogorow, A. "Interpolation and Extrapolation Ven Stationären Zufälligen Folgen," Bulletin de L'Academie des sciences de U.R.S.S., Ser. Math 5, 3-14, 1941.

3. Kalman, R. E. and Bucy, R. S. "New Results in Linear Filtering and Prediction Theory," Trans. ASME, J. Basic Engrg., ser. D., Vol. 83, pp. 95-108, March 1961.

4. Singer, R. A. (1970), "Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets," IEEE Transactions on Aerospace and Electronic Systems, AES-6, 473-483.

5. Clark, B. L. (1975), "Development of An Adaptive Kalman Target Tracking Filter and Predictor for Fire Control Applications," Naval Surface Weapons Center Technical Report TR-3445.

6. Clark, B. L. "The Development of An Adaptive Kalman Target Tracking Filter," 1976 AIAA Guidance and Control Conference.

7. Jazwinski, A. H. "Limited Memory Optimal Filtering," IEEE T-AC, Vol. AC-13, Oct. 1968.

8. Thorp, J. S. "Optimal Tracking of Maneuvering Targets," IEEE T-AES, Vol. AES-9, No. 4, July 1973.

9. Spingarm, K. and Weidmann, H. L. "Linear Regression and Prediction for Tracking Maneuvering Aircraft Targets," IEEE T-AES, Vol. AES-8, No. 6, Nov. 1972.

10. Howard, R. A. "System Analysis of Semi-Markov Processes," IEEE Trans. on Military Electronics, Vol. 8, pp. 114-124, April 1964.

11. Moose, R. L. "An Adaptive Estimator for Range and Depth Determination of a Maneuvering Target," 1972, Naval Underwater Systems Center Technical Report.

12. Moose, R. L. "An Adaptive State Estimation Solution To The Maneuvering Target Problem," IEEE T-AC, Vol. AC-20, June 1975.

13. Ogata, K. "State Space Analysis of Control Systems," Prentice-Hall, Inc., 1967.

14. Healey, M. "Study of Methods of Computing Transition Matrices," Proceedings of the Institute of Electrical Engineering on Control and Science, Vol. 120.

TRACKING MANEUVERING TARGETS VIA SEMI-MARKOV MANEUVER MODELING

by

Norman Hamilton Gholson

(ABSTRACT)

Adaptive algorithms for state estimation are currently of tremendous interest. Such estimation techniques have particular military usefulness in automatic gunfire control systems. The conventional Kalman filter, developed by Kalman and Bucy, optimally solves the state estimation problem concerning linear systems with Gaussian disturbance and error processes. The maneuvering target tracking problem generally involves nonlinear system properties as well as non-Gaussian disturbance processes. The study presented here explores several solutions to this problem.

An adaptive state estimator centered about the familiar Kalman filter has been developed for applications in three-dimensional maneuvering target tracking. Target maneuvers are modeled in a general manner by a semi-Markov process. The semi-Markov modeling is based on very intuitively appealing assumptions. Specifically, target maneuvers are randomly selected from a range (possibly infinite) of maneuver commands. The selected command is sustained for a random holding time before another command is selected. Dynamics of the selection and holding process may be stationary or time varying. By incorporating the semi-Markov modeling into a Baysian estimation scheme, an adaptive state estimator can be designed to identify the particular maneuver command influencing the target. The algorithm has the distinct advantages of

requiring only one Kalman filter and non-growing computer storage requirements.

Several techniques of implementing the adaptive algorithm have been developed. The merits of rectangular and spherical modeling have been explored. Most importantly, the planar discrete level semi-Markov algorithm, originally developed for sonar applications, has been extended to a continuum of levels, as well as extended to three-dimensional tracking.

The developed algorithms have been fully evaluated by computer simulations. Emphasis has been placed on computational burden as well as overall tracking performance. Results are presented that show that the developed estimators largely eliminate severe tracking errors that occur when more simplistic target models are incorporated.