APPLICATION OF ADAPTIVE STATE ESTIMATION TO TARGET TRACKING

by

H. Yücel ÖZEL

B.S. in E.E., Boğaziçi University, 1983

Submitted to the Institute for Graduate Studies in Science and Engineering in partial fulfillment of the requirements for the degree of

Master of Science

in

Electrical Engineering

Boğaziçi University

1986

APPLICATION OF ADAPTIVE STATE ESTIMATION

TO

TARGET TRACKING

APPROVED BY

Doç. Dr. Bülent SANKUR
(THESIS SUPERVISOR)

Doç. Dr. Selim ŞEKER

Dr. Emin ANARIN

DATE OF APPROVAL : 21. 2. 1986

TABLE OF CONTENTS

APPLICATION OF ADAPTIVE STATE ESTIMATION TO TARGET TRACKING

ABSTRACT

The development of an adaptive Kalman target tracking filter algorithm for high speed maneuvering fighter aircraft is studied. The extension of our estimator for use in an anti-aircraft gun fire control system is straightforward. The target modeling problem is studied considering such factors as target maneuvering characteristics and required filter settling and prediction times. Among several adaptation mechanisms, the so-called dual-bandwidth adaptation scheme is simulated on a digital computer, since this scheme is found to be the most sophisticated one of all and these adaptation techniques are essentially the same in nature. Implementation problem is overcome by means of using decoupled filters with linearized statistics. The algorithm developed for testing filter optimality has shown however, that the reduction of performance sensitivity to modeling errors was sufficiently good for our specific application.

UYARLANIR DURUM KESTİRİM KURAMININ HEDEF İZLEME SORUNUNA UYGULANMASI

ÖZETÇE

Bu çalışmada, yüksek hız düzeylerinde manevra yapabilme yeteneğine sahip hava hedeflerinin bir uyarlanabilir Kalman süzgeci ile izlenmesi sorunu ele alınmaktadır. Tasarımlamış olduğumuz durum kestirimcinin uçaksavar atış denetim dizgelerine kolaylıkla uygulanabileceği açıktır. Hedef için belli bir matematiksel model seçerken, hedefin ivmelenebilirlik özellikleri ile birlikte süzgecimizin yatışma süreci ve öngörü süresi gibi etmenler göz önünde tutulmuştur. Birkaç uyarlama yöntemi içinden yaptığımız seçim ikili-bantgenişlikli uyarlama yöntemi doğrultusunda olmaktadır. Bunun nedeni ise uyarlama yöntemlerinin özdeki benzerlikleri ve seçilen yöntemin diğerlerini de kısmen içeren en karmaşık yöntem oluşu olarak gösterilebilir. Gerçekleme sorununun çözümünde, istatistiksel verilerin doğrusallaştırılması, süzgeçleme algoritmasında kanalların bağımsız oluşu gibi işlem ve varsayımlar kullanılmaktadır. Ancak, süzgecin çalışma verimliliğini ölçmek için geliştirilen algoritma, bu tür varsayımların --en azından bu uygulama için-- verimliliğin duyarlığını yeteri kadar aza indirgediğini göstermektedir.

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$a = a(t)$	Acceleration
$E\{\cdot\}$	Expectation operator
F	System matrix
G	Input vector
H	Observation matrix
J	Jacobian matrix
P	Estimation error covariance matrix
Q	Process noise covariance matrix
$r = r(\tau)$	Autocorrelation function
R_p	Measurement error covariance matrix in spherical-polar coordinates
R_{CL}	Measurement error covariance matrix in Cartesian coordinates
u	Inhomogeneous driving input
w	White noise process
x	State vector
$\hat{x}$	Estimated state vector
z_p	Measurement vector in spherical polar coordinates
z_C	Measurement vector in Cartesian coordinates
α	Reciprocal of maneuver time constant
ν	Measurement residual sequence
σ	Variance of a probabilistic valu

I. INTRODUCTION

For about twenty years, the estimation theory has found its broadest application in the area of aerospace navigation. In this context the target tracking-trajectory estimation problem has been a good example. It is obvious that a great deal of tactical weapons systems require that manned maneuverable vehicles such as aircraft, ships and submarines be tracked accurately. To achieve such a goal, however, one has to make use of stochastic integrals and stochastic differential equations due to the nonlinear nature of the estimation problem at hand. In this thesis we have used the familiar "formal" manipulations of the Gaussian white process and omitted the sophisticated Itô calculus results. This approach is still an appropriate one for a large class of problems because it is generally impossible to find a mathematical model that is an exact representation of the physical process and as a second reason, the process noise is hardly a strictly white noise process.

For the reasons cited above, our main points of concern have been on the choice of a simple target model that closely represents the ensemble behaviour of a particular class of maneuvering vehicles and on the ease of implementation when used in the appropriate Kalman filter algorithm. The analysis is carried out in a generalized fashion and deals with the problem of tracking and predicting the state of a general target for fire control and other purposes. However, we will confine ourselves to the front end of the system, i.e., the target, sensor, filter and prediction of future target position. The predictor is the ultimate product required for the front end of the loop for this particular application. The target trajectory is obviously not directly available to the system. Instead, the sensor tracks the target's current position, superimposing measurement error in so doing. The purpose of the filter then is to process the noisy position measurements in such a manner as to estimate the parameters required for the prediction model. Such parameters might

include smooth current target position, velocity and acceleration.

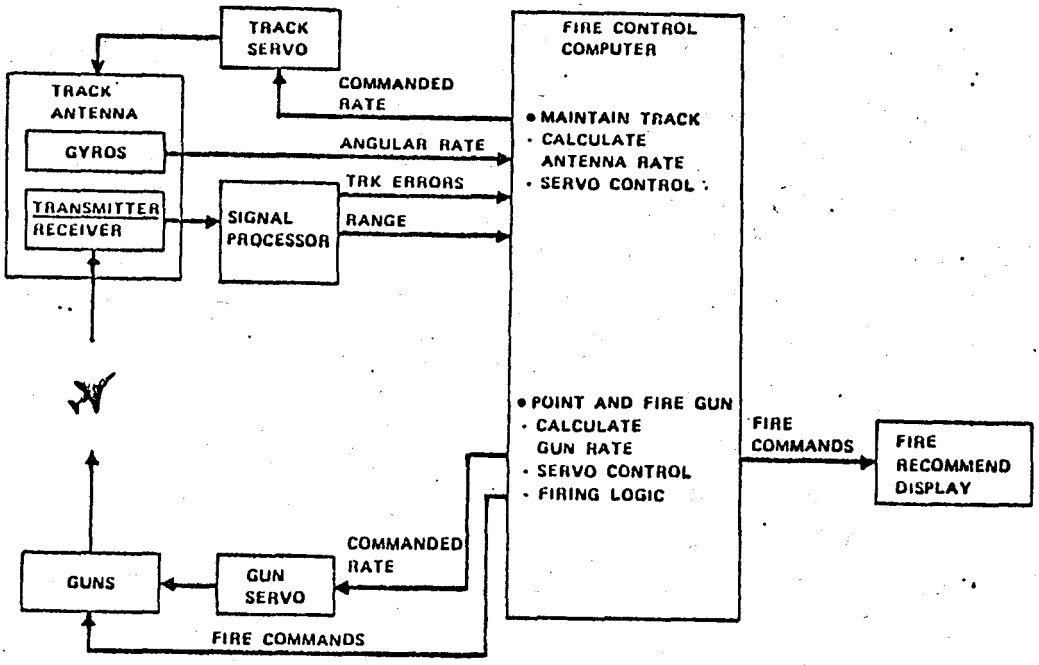


fig 1.1
Automatic Gun Fire Control System

It is assumed that the measurements have been stabilized, i.e ownship angular motion removed, if the system is intended for shipboard tracking of high speed maneuvering targets for weapon control applications and these measurements are then transformed to Cartesian coordinates. The implications of this transformation will be discussed in more detail later on. Tracking, filtering and prediction will therefore be performed in ownship coordinates.

The choice of Cartesian ownship coordinates is due to the fact that the motion of most targets is more closely linear in this system than in others. The prediction error , that is, the

error in determining the target's future position, basically results from two (not necessarily independent) sources: predictor modeling error and filtering or estimation error.

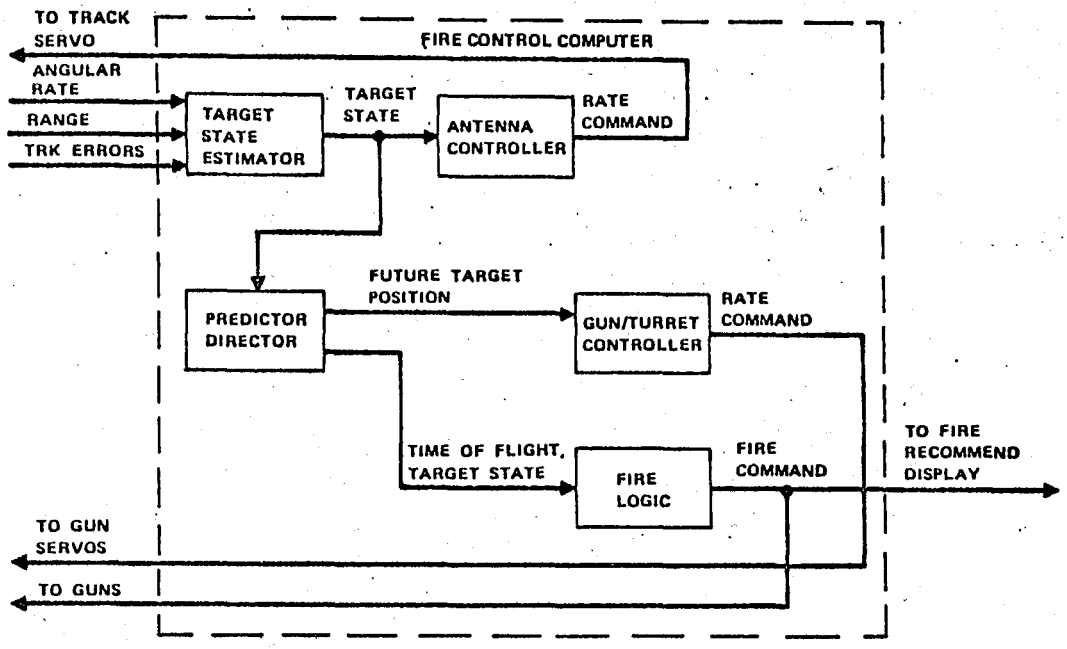


fig. 1.2
Fire control Software Algorithms

In general, a target will be maneuvering in some unknown fashion --at least as far as the tracker is concerned-- when it is in tracking range. The particular reasons for the target maneuvers are quite varied, from evasion or terminal homing to the presence of various sources such as sensor induced errors or atmospheric turbulance. Prediction modeling errors are due to indefiniteness of target strategy, and as a result, utilization of an incorrect functional form of the predictor. Obviously, even in the case of a perfectly modeled process, there would retain estimation errors due to sensor measurement noise, and these would be extrapoted from current time by the predictor.

Finally, we would like to consider the spatial problem. Measurements of target position are taken in a spherical polar frame, but targets are generally not described linearly in such a frame. Therefore, a nonlinearity will appear in the full three dimensional problem. We will be dealing with the formulations to establish our "channel independence" assumption, the most important implication of which is considerable computational savings.

II. DYNAMIC EQUATIONS OF TARGET MOTION

2.1. SYSTEM MODEL

a. Continuous time equations of motion

The problem of modeling targets in a realistic way has contradictory aspects. The target model selected for tracking applications should be sufficiently simple to permit ready implementation in weapons systems for which computation time is the most important factor. On the other hand, the model should be sophisticated enough to provide satisfactory tracking accuracy. The choice of a model to represent an actual target then is a critical one for it directly affects the estimation and prediction performance.

The model used in our study is based on the fact that, without maneuvering, manned vehicles of class under consideration such as aircraft, ships and submarines generally follow straight line constant velocity trajectories. If these vehicles were not able to deviate from these trajectories, i.e. could not maneuver tracking problem could be solved quickly and simply by using standart filter algorithms such as least squares, polynomial fitting, $\alpha - \beta$ techniques.. and so on.

We shall consider these techniques in more detail under the topic "alternative filtering methods" in chapter VI .

We presume here that the target normally moves at constant velocity. Turns, evasive maneuvers and accelerations due to atmospheric turbulance may be regarded as perturbations upon this constant velocity trajectory.

In a single physical dimension, the target equations of motion can be represented by :

$$\dot{\underline{x}}'(t) = \underline{\underline{F}}'\underline{x}'(t) + \underline{G}'a(t) \qquad (2.1)$$

Where

$$\underline{x}(t) = \begin{cases} \text{target position at time t} \\ \text{target velocity at time t} \end{cases}$$

and

$$\underline{\underline{F}}' = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

and

$$\underline{G}' = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

The acceleration a(t) , since it accounts for the target deviations from a straight line trajectory, will henceforht be termed the "target maneuver variable" .

The single dimension maneuver capability can be satisfactorily specified by two quantities; the variance or magnitude of target maneuver, and the time constant or duration of the target maneuver. The target acceleration , and hence the target maneuver is correlated in time; namely, if a target is accelerating at time t, it is likely to be accelerating at time $t + \tau$ for sufficiently small τ. A lazy turn, for instance, will give rise to correlated acceleration inputs up to one minute, evasive maneuvers will provide correlated inputs for periods between ten and thirty seconds, and athmospheric turbulance may provide inputs for one or two seconds.

A typical model of the correlation function $r(\tau)$ associated with the target acceleration is:

$$r(\tau) = E\left[a(t).a(t+\tau)\right] = \sigma_m^2 \exp(-\alpha|\tau|) \qquad (2.2)$$

Where

σ_m^2 : variance of the target acceleration

α : reciprocal of the maneuver time constant, $\alpha \geqslant 0$

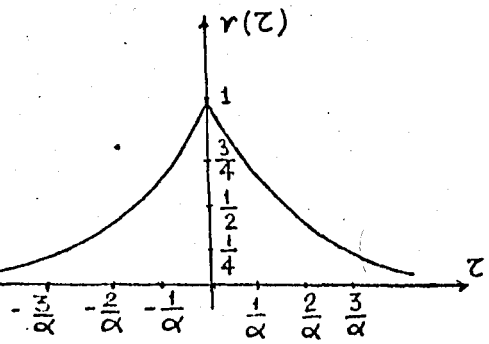


fig. 2.1

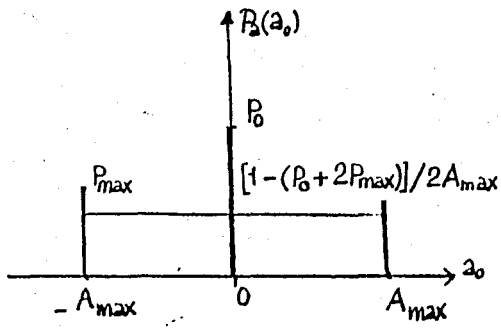


fig. 2.2

The variance σ_m^2 of the target acceleration can be found easily by using the probability density function in fig. 2.1

$$\text{var}\{a\} = E\{[a - E\{a\}]^2\}$$

$$\sigma_m^2 : \text{var}\{a\} = (1/3)\, A_{max}^2 [1 + 4P_{max} - P_o] \qquad (2.3)$$

One can derive above term as follows:

$$\sigma_m^2 = \text{var}\{a\} = E\{[a - E\{a\}]^2\}$$

$$= E\{a^2\} - E^2\{a\}$$

$$= \int_{-A_{max}}^{A_{max}} a_o^2 \left[\frac{1 - P_o - 2P_{max}}{2A_{max}}\right] da_o + \sum_{a_o \in \{a\}} a_o^2 P_a(a_o)$$

$$- \left\{ \int_{-Amax}^{A_{max}} a_o \left[\frac{1 - (P_o + 2P_{max})}{2A_{max}}\right] da_o + \sum_{a_o \in \{a\}} a_o P_a(a_o) \right\}^2$$

$$= \frac{1 - (P_o + 2P_{max})}{2A_{max}} \cdot \frac{a_o^3}{3} \Big|_{-A_{max}}^{A_{max}} + \left((-A_{max}^2)P_{max} + 0 \cdot P_o + (A_{max}^2)P_{max}\right)$$

$$-\left\{\frac{1-(P_o+2P_{max})}{2A_{max}}\cdot\frac{a_o^2}{2}\Big|_{-A_{max}}^{A_{max}}+(-A_{max}P_{max}+0\cdot P_o+A_{max}P_{max})\right\}^2$$

$$=\frac{1-(P_o+2P_{max})}{2A_{max}}\cdot\frac{2A_{max}}{3}\cdot\frac{2A_{max}^2P_{max}}{3}$$

$$=\frac{1}{3}A_{max}^2\left[1+4P_{max}-P_o\right]$$

Now, utilizing the correlation function $r(\tau)$, the acceleration $a(t)$ may be expressed in terms of white noise by the Wiener - Kolmogorov whitening procedure. The Laplace transform of $r(\tau)$, $R(S)$ is given by ;

$$L\{r(\tau)\}=R(S)=-\frac{2\sigma_m^2\alpha}{(S-\alpha)(S+\alpha)} \qquad (2.4)$$

Which can be factored as ;

$$R(S)=H(S).H(-S).W(S)$$

Where

$$H(S)=\frac{1}{S+\alpha}$$

$$H(-S)=\frac{-1}{S-\alpha}$$

$$W(S)=2\alpha\sigma_m^2$$

The quantity H(S) is the transform of the whitening filter for a(t), and W(S) is the transform of the white noise w(t) that drives a(t). The resulting equations are therefore,

$$\dot{a}(t) = -\alpha a(t) + w(t) \tag{2.5}$$

where $\sigma_w^2(\tau)$ is the correlation function of the white noise input, satisfying ;

$$\sigma_w^2(\tau) = 2\alpha\sigma_m^2\,\delta(\tau) \tag{2.6}$$

The target equations of motion in one physical dimension can be expressed in terms of the white noise w(t) as follows :

$$\dot{\underline{x}}(t) = \underline{\underline{F}}\,\underline{x}(t) + \underline{G}\,w(t) \tag{2.7}$$

where

$$x(t) = \begin{cases} \text{target position at time t} \\ \text{target velocity at time t} \\ \text{target acceleration at time t} \end{cases}$$

$w(t)$ = white noise driving function with variance $2\alpha\sigma_m^2$

$$\frac{d}{dt}a(t) = -\alpha a(t) + w(t) \tag{2.8}$$

$$\frac{d}{dt}\left[\frac{d^2}{dt^2}x(t)\right] = -\alpha\left[\frac{d^2}{dt^2}x(t)\right] + w(t) \tag{2.9}$$

Defining new variables,

$$x_1(t) = x(t)$$
$$\dot{x}_1(t) = \dot{x}(t) = x_2(t)$$
$$\dot{x}_2(t) = \ddot{x}(t) = x_3(t)$$
$$\dot{x}_3(t) = \frac{d}{dt}\left[\frac{d^2}{dt^2}x(t)\right] = -\alpha x_3(t) + w(t)$$

We obtain the following matrix formulation :

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\alpha \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w(t) \qquad (2.10)$$

b. Discrete-time equations of motion

Many sensors have a constant data rate, sampling target position every T second . The appropriate discrete time equations of motions for this case are given by :

$$\underline{x}(k+1) = \underline{\underline{\Phi}}(k+1,k)\ \underline{x}(k) + \underline{u}(k) \qquad (2.11)$$

where the state transition matrix is of the form :

$$\underline{\underline{\Phi}}(k+1,k) = \underline{\underline{\Phi}} = \underline{\underline{\Phi}}(T,\alpha)$$

u(k) : inhomogeneous driving input

$$\underline{\underline{\Phi}}(T,\alpha) = \exp(\underline{\underline{F}}T) = L^{-1}\left\{ \left[s\underline{\underline{I}} - \underline{\underline{F}} \right]^{-1} \right\} \qquad (2.12)$$

$$= \begin{bmatrix} 1 & T & \frac{1}{\alpha^2}\left[-1 + \alpha T + \exp(-\alpha T)\right] \\ 0 & 1 & \frac{1}{\alpha}\left[-1 - \exp(-\alpha T)\right] \\ 0 & 0 & \exp(-\alpha T) \end{bmatrix} \qquad (2.13)$$

The input vector u(k) satisfies :

$$\underline{u}(k) = \int_{kT}^{(k+1)T} \exp\left(\underline{\underline{F}}\left[(k+1)T - \tau\right]\right)\ \underline{G}\ w(\tau)\ d\tau \qquad (2.14)$$

$$= \int_{kT}^{(k+1)T} \begin{bmatrix} 1 & (k+1)T - \tau & \frac{1}{\alpha^2}\left[-1 + \alpha\left[(k+1)T - \tau\right] + \exp\left(-\alpha\left[(k+1)T - \tau\right]\right)\right] \\ 0 & 1 & \frac{1}{\alpha}\left[+1 - \exp\left(-\alpha\left[(k+1)T - \tau\right]\right)\right] \\ 0 & 0 & \exp\left(-\alpha\left[(k+1)T - \tau\right]\right) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w\,d\tau$$

$$\int_{kT}^{(k+1)T} \begin{bmatrix} \frac{1}{\alpha^2}\left[-1+\alpha\left[(k+1)T-\tau\right]+\exp\left(-\alpha\left[(k+1)T-\tau\right]\right)\right. \\ \frac{1}{\alpha}\left[1-\exp\left(-\alpha\left[(k+1)T-\tau\right)\right]\right. \\ \exp\left(-\alpha\left[(k+1)T-\tau\right]\right) \end{bmatrix} w(\tau)\,d\tau$$

$$\int_{kT}^{(k+1)T} \begin{bmatrix} \eta_1(\tau) \\ \eta_2(\tau) \\ \eta_3(\tau) \end{bmatrix} w(\tau)\,d\tau \qquad (2.15)$$

Now, since $w(t)$ is white noise, $E\left\{u(k)u^T(k+i)\right\} = \underline{0}$, for $i \neq 0$. So, $u(k)$ is a discrete time white noise sequence.

For the sake of completeness, let us discuss the effects variations in parameters characterizing the maneuver of target under consideration.

The resemblance of this model to the second order polynomial model [7] can immediately be seen if we consider the limiting case $(1/\alpha) \gg T$. The state transition matrix becomes identical to the second order polynomial model if $(1/\alpha)$ vanishes or $(1/\alpha) \ll T$. We find this state transition matrix corresponds to the first order polynomial model. It can therefore be expected that, for finite values of $(1/\alpha)$ term, our model can exhibit convergence properties between those of the first and second order polynomial filters.

Let us make a further definition :

$$\tau_m = \frac{1}{\alpha}$$

and call τ_m as characteristic maneuver time. The choice of a good value of τ_m is a function primarily of the target scenario. Singer [6] recommends $\tau_m = 20$ seconds for manned maneuvering targets exercising evasive maneuvers. The same value has also been found independently by other people who have studied the problem. This rather low "observed" maneuver frequency, $\alpha = (1/20)\,\text{sec}^{-1}$, does not necessarily imply that this is the highest frequency that a particular target might be capable of sustaining. Of course, most fighter aircraft can maneuver much more rapidly if desired. A low maneuver frequency however, is probably typical of air targets maneuvering to achieve

a particular goal. We consider values of τ_m anywhere in the range of 3 to 20 seconds as realistic. The selection of a value for σ_m is more difficult than for τ_m. The acceleration autocorrelation study of the target scenario demonstrates a very wide range of values for σ_m. While some targets are sustaining an rms maneuver level close to one G (approximately 10 yards/sec^2) the others exhibit a very low maneuver level which is due to some athmospheric effects. We are going to devote a more detailed discussion on the choice of parameters in the next sections. Before closing this section, we are to give the formulations for the derivations of covariance matrix of process noise and initial filter state.

2.2. PARAMETERS FOR FILTER INITIATION

The matrix Q(k) is the covariance matrix of the maneuver excitation and as shown in [6] has the form :

$$Q(k) = E\left\{u(k)u^T(k)\right\} = 2\alpha\sigma_m^2 \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{12} & q_{22} & q_{23} \\ q_{13} & q_{23} & q_{33} \end{bmatrix} \quad (2.16)$$

where

$$q_{11} = \frac{1}{2\alpha^5}\left[1 - \exp(-2\alpha T) + \frac{2\alpha^3 T^3}{3} - 2\alpha^2 T^2 - 4\alpha T\exp(-\alpha T) - 2\alpha T\right.$$

$$q_{12} = \frac{1}{2\alpha^4}\left[\exp(-2\alpha T) + 1 - 2\exp(-\alpha T) - 2\alpha T\exp(-\alpha T) - 2\alpha T - \alpha^2 T^2\right]$$

$$q_{13} = \frac{1}{2\alpha^3}\left[1 - \exp(-2\alpha T) - 2\alpha T\exp(-\alpha T)\right]$$

$$q_{22} = \frac{1}{2\alpha^3}\left[-\exp(-2\alpha T) + 2\alpha T - 3 + 4\exp(-2\alpha T)\right]$$

$$q_{23} = \frac{1}{2\alpha^2}\left[\exp(-2\alpha T) + 1 - 2\exp(-\alpha T)\right]$$

$$q_{33} = \frac{1}{2\alpha}\left[1 - \exp(-2\alpha T)\right]$$

For a fixed sensor and target class the α and T terms are fixed so that Q(k) is a constant matrix. When T is sufficiently small so that $\alpha T \ll 1/2$,

$$\lim_{\alpha T \to 0} Q(k) = 2\alpha\sigma_m^2 \begin{bmatrix} T^5/20 & T^4/8 & T^3/6 \\ T^4/8 & T^3/3 & T^2/2 \\ T^3/6 & T^2/2 & T \end{bmatrix} \qquad (2.17)$$

reflecting the fact that for sufficiently short time periods the physical target moves at constant velocity. For a fixed sampling rate, as $\alpha \to \infty$;

$$\lim_{\alpha \to \infty} Q(k) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sigma_m^2 \end{bmatrix} \qquad (2.18)$$

The Kalman filter equations can be initialized by :

$$\hat{x}_1(0/0) = z(0) \qquad (2.19a)$$

$$\hat{x}_2(0/0) = \left[z(1) - z(0)\right]/T \qquad (2.19b)$$

$$\hat{x}_3(0/0) = 0 \qquad (2.19c)$$

Where z(0) and z(1) are, respectively, the first and second measurements received. The covariance initialization equations are :

$$e_1(1/1) = \hat{x}_1(1/1) - x_1(1) = x_1(1) - x_1(1) - v(1) \qquad (2.20a)$$
$$= v(1)$$

$$e_2(1/1) = x_2(1) - \hat{x}_2(1/1) \qquad (2.20b)$$

$$= x_2(0) + \frac{1}{\alpha}\left[1 - \exp(-\alpha T)\right] x_3(0)$$

$$+ u_2(0) - \frac{x_1(1) + v(1) - x_1(0) - v(0)}{T}$$

$$= \frac{v(0)}{T} - \frac{v(1)}{T} +$$

$$\left[\frac{-\exp(-\alpha T)}{\alpha} - \frac{1}{\alpha^2 T} - \frac{\exp(-\alpha T)}{\alpha^2 T}\right] x_3(0) + u_2(0) - \frac{u_1(0)}{T}$$

$$e_3(1/1) = x_3(1) - \hat{x}_3(1/1) \qquad (2.20c)$$

$$= \exp(-\alpha T)\bar{x}_3(0) + u_3(0)$$

Since

$$P_{ij}(1/1) = E\left\{ e_i(1/1) e_j(1/1) \right\} ; \qquad (2.21)$$

it follows that

$$P_{11}(1/1) = \sigma_R^2 \qquad (2.22a)$$

$$P_{12}(1/1) = \frac{\sigma_R^2}{T} \qquad (2.22b)$$

$$P_{13}(1/1) = 0 \qquad (2.22c)$$

$$P_{22}(1/1) = \frac{2\sigma_R^2}{T^2} + \frac{\sigma_M^2}{\alpha^4 T^2}\left[1 - \exp(-\alpha T) - \alpha T \exp(-\alpha T)\right]^2$$

$$+ E\left\{u_2^2\right\} - E\left\{\frac{u_1^2}{T^2}\right\} - \frac{2}{T} E\left\{u_1 u_2\right\} \qquad (2.22d)$$

$$= \left[\frac{2\sigma_R^2}{T^2} + \frac{\sigma_M^2}{\alpha^4 T^2}\right].$$

$$\cdot\left[2 - \alpha^2 T^2 + \frac{2\alpha^3 T^3}{3} - 2\exp(-\alpha T) - 2\alpha T\exp(-\alpha T)\right]$$

$$P_{23}(1/1) = \frac{\sigma_M^2}{\alpha^2 T}\left[\exp(-\alpha T) - \alpha T\exp(-2\alpha T) - \exp(-2\alpha T)\right]$$

$$+ E\{u_2 u_3\} - \frac{E\{u_1 u_3\}}{T} \qquad (2.22e)$$

$$= \frac{\sigma_M^2}{\alpha^2 T}\left[\exp(-\alpha T) - \alpha T - 1\right]$$

$$P_{33} = \sigma_M^2 \exp(-2\alpha T) - E\{u_3^2\} \qquad (2.22f)$$

$$= \sigma_M^2 \exp(-2\alpha T) - \sigma_M^2\left[1 - \exp(-2\alpha T)\right]$$

$$= \sigma_M^2$$

In fact, we do not have to specify initial values for either the covariance matrix P(0/0) or the initial state estimate x(0/0), because under optimal performance conditions, the steady state values of these parameters should be totally independent of these starting values.

III. THREE-DIMENSIONAL FILTERING IN CARTESIAN COORDINATES

We have so far discussed the development of a target model for our filtering and prediction problem and have dealt with one-dimensional description of target motion for the sake of simplicity. Nevertheless, the multi-dimensional aspects of the problem had to be considered at all times during the development. In this section, we will concentrate on this side of the problem.

There are several factors to be considered in selecting a coordinate frame for filtering and prediction applications. Primary consideration must be given to accuracy, since this is a tracking problem, and computational simplicity, since the algorithm must be implemented within the constraints of a given computer. The sensitivity to various nonlinear effects forces us to consider these factors in coordinate system selection.

The study presented in this part of our work is based on the premises :

i. The target is modeled best in Cartesian coordinates. By this we mean the target is more closely linear and well-behaved in Cartesian coordinates than in spherical polar coordinates. For example, if one considers a simple linear (constant velocity) target motion which is canonical in Cartesian coordinates, then using equation (3.7) one finds that second and all higher derivatives appear in the polar frame. This are sometimes described as "pseudo maneuvers" in the literature. These accelerations, if viewed in the polar frame, must either be modeled and propagated nonlinearly by the filter-predictor, or, even worse, tracked adaptively. There are, on the other hand, corresponding polar

canonical targets such as motion along a ray of circular motion centered at the origin. Examination of real life target motions indicates that such motion would be encountered much less frequently than the nominal Cartesian motion. We will therefore assume that the random acceleration target in Cartesian coordinate frame is a good representation of the true target and proceed on this basis. To form the Cartesian target model, an additional assumption of independence of target maneuver among channels was made. That is, if we define the three-dimensional Cartesian state vector as ;

$$\underline{X}_{3D} = \begin{bmatrix} \underline{x} \\ \hline \underline{y} \\ \hline \underline{z} \end{bmatrix} \tag{3.1}$$

where $\underline{x}$, $\underline{y}$ and $\underline{z}$ are each one - dimensional three - element state vectors (position, velocity, and acceleration) governed by the random acceleration target model.

The S.T.M. (state transition matrix) is then ;

$$\underline{\Phi}_{3D} = \left[\begin{array}{c|c|c} \underline{\Phi} & \underline{0} & \underline{0} \\ \hline \underline{0} & \underline{\Phi} & \underline{0} \\ \hline \underline{0} & \underline{0} & \underline{\Phi} \end{array}\right]_{9\times 9} \tag{3.2}$$

where $\underline{\Phi}$ is as defined in section II.

The assumption of complete independence among channels is obviously a weak one, but the implications of this assumption compensates for its unrealistic nature as we shall see later on. It depends , on the other hand, the particular target type and its particular angular orientation in the coordinate frame . The target motion analysis problem discards the importance of this assumption, reqiring more additional attention .

ii. Measurements of target position are obtained in spherical polar coordinates, i.e. range, bearing angle from north, and elevation angle.

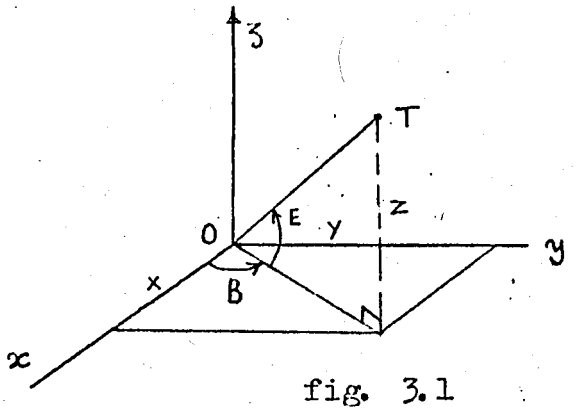


fig. 3.1

$$z_p(k) = \begin{bmatrix} r(k) \\ B(k) \\ E(k) \end{bmatrix} + \begin{bmatrix} n_r(k) \\ n_B(k) \\ n_E(k) \end{bmatrix} \qquad (3.3)$$

where $r(k)$, $B(k)$, and $E(k)$ are true target range, bearing and elevation respectively, and $n_r(k)$, $n_B(k)$, $n_E(k)$ are the corresponding measurement noise components.

It is assumed that $n_r(k)$, $n_B(k)$ and $n_E(k)$ are mutually uncorrelated and that ;

$$E\left\{n_r(k)\right\} = E\left\{n_B(k)\right\} = E\left\{n_E(k)\right\} = 0$$

and

$$R_p(k) = E\left\{n(k)\ n^T(k)\right\} = \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_B^2 & 0 \\ 0 & 0 & \sigma_E^2 \end{bmatrix} \quad \text{for } \forall k \qquad (3.4)$$

The spherical polar - Cartesian transformation equations are ;

$$z_C(k) = \begin{bmatrix} x(k) \\ y(k) \\ z(k) \end{bmatrix} + \begin{bmatrix} v_1(k) \\ v_2(k) \\ v_3(k) \end{bmatrix} \qquad (3.5)$$

where

$$x(k) = r(k)\,\mathrm{Cos}E(k)\,\mathrm{Cos}B(k) \tag{3.6a}$$

$$y(k) = r(k)\,\mathrm{Cos}E(k)\,\mathrm{Sin}B(k) \tag{3.6b}$$

$$z(k) = r(k)\,\mathrm{Sin}E(k) \tag{3.6c}$$

and the corresponding inverse relation is ;

$$r(k) = \left[x^2(k) + y^2(k) + z^2(k) \right]^{1/2} \tag{3.7a}$$

$$B(k) = \arctan(y(k)/x(k)) \tag{3.7b}$$

$$E(k) = \mathrm{arcSin} \frac{z(k)}{\left[x^2(k) + y^2(k) + z^2(k)\right]^{1/2}} \tag{3.7c}$$

To recapitulate, the measurement vector equation is given by :

$$\underline{z}_C(k) = \underline{h}(\underline{x}(k), k) + \underline{v}(k) \tag{3.8}$$

Now , let us form the Jacobian matrix to obtain the linearized measurement error covariance , i.e. ,

$$\underline{\underline{J}} = \frac{\partial \underline{h}}{\partial \underline{z}_p} \tag{3.9}$$

$$\underline{\underline{J}} = \begin{bmatrix} \mathrm{Cos}B\ \mathrm{Cos}E & -r\mathrm{Sin}B\ \mathrm{Cos}E & -r\mathrm{Cos}B\ \mathrm{Sin}E \\ \mathrm{Sin}B\ \mathrm{Cos}E & r\mathrm{Cos}B\ \mathrm{Cos}E & -r\mathrm{Sin}B\ \mathrm{Sin}E \\ \mathrm{Sin}E & 0 & r\mathrm{Cos}E \end{bmatrix} \tag{3.10}$$

The linearized Cartesian measurement error covariance matrix is given by ;

$$\underline{\underline{R}}_{CL} = \underline{\underline{J}}\,\underline{\underline{R}}_p\,\underline{\underline{J}}^T \tag{3.11}$$

After the matrix multiplications have been performed, there

will appear off - diagonal terms in the R_{CL} matrix , the implications of which is the cross-correlation among x, y and z directions. The entries of the covariance matrix of measurement error are found to be ;

$$r_{11} = \sigma_{xx}^2 = \sigma_r^2 \cos^2 B \cos^2 E + r^2 \sigma_B^2 \sin^2 B \cos^2 E + r^2 \sigma_E^2 \cos^2 B \sin^2 E$$

$$r_{12} = \sigma_r^2 \sin^2 B \cos B \cos^2 E - r^2 \sigma_B^2 \cos B \sin B \cos^2 E$$

$$- r^2 \sigma_E^2 \sin B \cos B \sin^2 E$$

$$r_{13} = \sigma_r^2 \cos B \cos E \sin E - r^2 \sigma_E^2 \cos B \cos E \sin E$$

$$r_{22} = \sigma_{yy}^2 = \sigma_r^2 \sin^2 B \cos^2 E + r^2 \sigma_B^2 \cos^2 B \cos^2 E + r^2 \sigma_E^2 \sin^2 E \sin^2 B$$

$$r_{23} = \sigma_r^2 \sin B \sin E \cos E - r^2 \sigma_E^2 \sin B \sin E \cos E$$

$$r_{33} = \sigma_{zz}^2 = \sigma_r^2 \sin^2 E + r^2 \sigma_E^2 \cos^2 E$$

For the Cartesian noise terms, the obvious result follows :

$$\underline{v}_C(k) = \begin{bmatrix} v_1(k) \\ v_2(k) \\ v_3(k) \end{bmatrix} = \underline{\underline{J}} \underline{v}_p(k) \qquad (3.12)$$

$$E\left\{\underline{v}_C(k)\right\} = E\left\{\underline{\underline{J}} \underline{v}_p(k)\right\} = \underline{\underline{J}}\, E\left\{\underline{v}_p(k)\right\} = \underline{0} \qquad \forall k \qquad (3.13)$$

Thus ,

$$E\left\{\underline{\underline{J}} \underline{v}_p(k) \underline{v}_p^T(k) \underline{\underline{J}}^T\right\} = \underline{\underline{J}}\, E\left\{\underline{v}_p(k) \underline{v}_p^T(k)\right\}\, \underline{\underline{J}}^T = \underline{\underline{J}} \underline{\underline{R}}_p \underline{\underline{J}}^T$$

$$= \underline{\underline{R}}_{CL}(k) \qquad (3.14)$$

3.1. Coordinate system rotation

In the preceding section, the radar measurements of target position in spherical polar coordinates are converted to Cartesian frame by the equations :

$$\underline{z}_C(k) = \begin{bmatrix} x(k) \\ y(k) \\ z(k) \end{bmatrix} + \begin{bmatrix} v_1(k) \\ v_2(k) \\ v_3(k) \end{bmatrix} = \begin{bmatrix} r(k)\,\mathrm{Cos}B(k)\,\mathrm{Cos}E(k) \\ r(k)\,\mathrm{Sin}B(k)\,\mathrm{Cos}E(k) \\ r(k)\,\mathrm{Sin}E(k) \end{bmatrix} + \begin{bmatrix} v_1(k) \\ v_2(k) \\ v_3(k) \end{bmatrix}$$

Now, if the x-axis is aligned with the initial bearing angle of the target , the above equations become ;

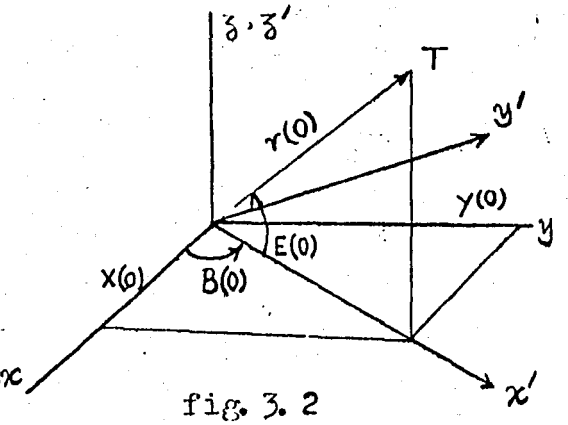


fig. 3. 2

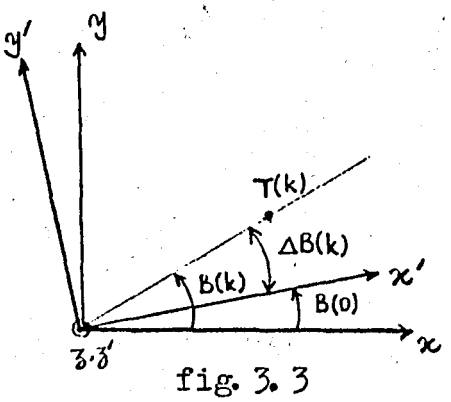


fig. 3. 3

$$\underline{z}_C(k) = \begin{bmatrix} x'(k) \\ y'(k) \\ z'(k) \end{bmatrix} + \begin{bmatrix} v_1(k) \\ v_2(k) \\ v_3(k) \end{bmatrix} = \begin{bmatrix} r(k)\,\mathrm{Cos}\,\Delta B(k)\,\mathrm{Cos}E(k) \\ r(k)\,\mathrm{Sin}\,\Delta B(k)\,\mathrm{Cos}E(k) \\ r(k)\,\mathrm{Sin}E(k) \end{bmatrix} + \begin{bmatrix} v_1(k) \\ v_2(k) \\ v_3(k) \end{bmatrix} \qquad (3.15)$$

where

$$\Delta B(k) = B(k) - B(0) \qquad (3.16)$$

In order to maintain the oriantation of the rotated coordinate system, it is necessary to store the initial bearing $B(0)$ for use in the coordinate conversion process.

The effect of the correlation is diminished in the rotated Cartesian coordinate system, since for an inbound target the bearing will be zero or near zero. If $B(k) = 0$ is assumed, the entries of the measurement error covariance matrix now become :

$$r_{11} = \sigma_{xx}^2 = \sigma_r^2$$

$$r_{12} = \sigma_{xy} = 0 = r_{21}$$

$$r_{13} = \sigma_{xz} = \text{SinE CosE}\left[\sigma_r^2 - r^2\sigma_E^2\right] = r_{31}$$

$$r_{22} = \sigma_{yy}^2 - r^2\sigma_B^2\text{Cos}^2\text{E}$$

$$r_{32} = \sigma_{yz} = 0$$

$$r_{33} = \sigma_{zz}^2 = \sigma_r^2\text{SinE} - r^2\sigma_E^2\text{Cos}^2\text{E}$$

Although the measurement noise is still correlated , the dependence on the target bearing has been eliminated. We shall make now a further assumption :

For the case of a low elevation target, where the measurement noise is expected to be relatively high due to poor tracking conditions in the radar , the approximations $\text{SinE}(k) \cong 0$ and $\text{CosE}(k) \cong 1$ are valid, and therefore ;

$$R_{CL}(k) = \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & r^2(k)\,\sigma_B^2 & 0 \\ 0 & 0 & r^2(k)\,\sigma_E^2 \end{bmatrix} \qquad (3.17)$$

which agrees in form with the uncorrelated noise model proposed by spherical coordinate measurement modeling .

IV. FILTER DEVELOPMENT

4.1 Summary of the conventional filtering algorithm of Kalman

In the previous two chapters, we have dealt with the problem of modeling the targets belonging to the class under consideration and finally obtained the following model ;

$$\underline{x}(k+1) = \underline{\underline{\Phi}}(T,\alpha)\ \underline{x}(k) + \underline{w}(k) \qquad (4.1)$$

where

$$\underline{\underline{\Phi}}(T,\alpha) = \begin{bmatrix} \underline{\underline{\Phi}}_x & \underline{\underline{0}} & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{\Phi}}_y & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{0}} & \underline{\underline{\Phi}}_z \end{bmatrix} \qquad (4.2)$$

and

$$\underline{x}(k) = \begin{bmatrix} x(k) \\ \dot{x}(k) \\ \ddot{x}(k) \\ \hline y(k) \\ \dot{y}(k) \\ \ddot{y}(k) \\ \hline z(k) \\ \dot{z}(k) \\ \ddot{z}(k) \end{bmatrix} \quad ; \quad \underline{w}(k) = \begin{bmatrix} w_x(k) \\ w_y(k) \\ w_z(k) \end{bmatrix} \qquad (4.3)$$

with measurements

$$\underline{z}_c = \left[\begin{array}{ccc|ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right] \left[\begin{array}{c} \underline{x} \\ \hline \underline{y} \\ \hline \underline{z} \end{array}\right] + \begin{bmatrix} v_1(k) \\ v_2(k) \\ v_3(k) \end{bmatrix} \qquad (4.4)$$

or, in a compact form ,

$$\underline{z}_C(k) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{pos}(k) \\ x_{vel}(k) \\ x_{acc}(k) \end{bmatrix} + v_1(k) \qquad (4.5)$$

i. e. partitioning the measurement equation in the above manner the same result can easily be obtained for y and z channel measurements.

Measurement equation in one dimension, on x-axis thus becomes ;

$$z_C(k) = x_{pos}(k) + v_1(k) \qquad (4.6)$$

obviously a scalar equation.

This result makes it possible for us to process the observations sequentially . We will discuss this facility in more detail later on .

Having summarized the system and measurement models, let us go on to the study of random process statistics .

The following equations summarize these statistics :

$$E\{\underline{w}(k)\} = \underline{0} \qquad \forall k$$

$$E\{\underline{v}(k)\} = \underline{0} \qquad \forall k$$

$$E\{\underline{w}(k)\underline{w}^T(k)\} = \underline{\underline{Q}}(k)\delta_{ik}$$

where Q(k) is as obtained in chapter II .

$$E\{\underline{w}(j)\underline{v}^T(k)\} = \underline{0} \qquad \forall j , k$$

$$E\{\underline{v}(k)\underline{v}^T(k)\} = \underline{\underline{R}}(k)\delta_{ik}.$$

where $\underline{\underline{R}}(k)$ is as obtained in chapter III .

The discrete Kalman filter equations are ;

1. The dynamic extrapolation of the preceding state

$$\hat{\underline{x}}(k/k-1) = \underline{\underline{\Phi}}(T, \alpha)\ \hat{\underline{x}}(k-1/k-1)$$

where

$\hat{\underline{x}}(k/k-1)$: MMSE estimate of $x(k)$ given sensor data up to and including time k, i.e. the one sample ahead prediction .

$\hat{\underline{x}}(k-1/k-1)$: MMSE estimate of $\underline{x}(k-1)$ given sensor data up to and including time k , i.e. the filtered estimate

2. Evaluation of the optimal filtered estimate ;

$$\hat{\underline{x}}(k/k) = \hat{\underline{x}}(k/k-1) + \underline{\underline{K}}(k)\left[\underline{z}_C(k) - \underline{\underline{H}}\hat{\underline{x}}(k/k-1)\right]$$

optimal filtered estimate: prediction + correction

where

$$\underline{\underline{K}}(k) = \underline{\underline{P}}(k/k-1)\underline{\underline{H}}^T(k)\left[\underline{\underline{H}}(k)\underline{\underline{P}}(k/k-1)\underline{\underline{H}}^T(k) + \underline{\underline{R}}(k)\right]^{-1}$$

defined as Kalman gain in the literature
and

$$\underline{\underline{P}}(k/k-1) = \underline{\underline{\Phi}}(T,\alpha)\underline{\underline{P}}(k-1/k-1)\underline{\underline{\Phi}}^T(T,\alpha) + \underline{\underline{Q}}(k-1)$$

is the predicted error covariance matrix of the estimate
and finally ,

$$\underline{\underline{P}}(k/k) = \left[\underline{\underline{I}} - \underline{\underline{K}}(k)\underline{\underline{H}}(k)\right]\underline{\underline{P}}(k/k-1)$$

is the new error covariance matrix .

The derivation of these terms can be found easily in the literature, especially in [1] a broad manner .

4. 2. Adaptation

The Kalman filter formulation presented in the previous section assumes complete knowledge of the linear dynamics and the process noise covariance . Normally , however , the particular strategy being exercised by the target is not known by the tracker in advance . In turn , the form of the state vector

and its propagation characteristics is assumed and may or may not represent true target dynamics over long periods of time. Such a situation is usually referred to as "suboptimal modeling" in the sense that no attempt is made to fully model the target dynamics . As we noted in the very beginning of this work , we have overlooked the rigorous treatment of this nonlinear estimation problem and preferred to look at it at an engineer's point of view . The utilization of a suboptimal model leads to large estimation errors , a condition known as filter divergence.

When divergence occurs, an inconsistency between the error covariance calculated by the filter and the actual error covariance occurs . An adaptive filter is basically a method of adjusting parameters in order to effect a more realistic match between the calculated and actual filter error covariances.

In our attempts to model the system in a practical but still realistic way, we had found that no single set of parameters would adequately represent the true target scenario. Instead , it was found that a range of each parameter (σ_m , α_m) could be expected and that we can essentially bound the parameters by a low maneuver level, low frequency (long time constant) set which we will designate $(\sigma_{mA}, \alpha_{mA})$ and a high maneuver , high frequency set, $(\sigma_{mB}, \alpha_{mB})$. Our fundamental assumption in approaching adaptation with the random acceleration model is therefore ,

$$(\sigma_{mA} , \alpha_{mA}) \leqslant \underbrace{(\sigma_m , \alpha_m)}_{\substack{\text{parameter set representing} \\ \text{any actual target}}} < (\sigma_{mB} , \alpha_{mB}) \qquad (4.7)$$

Let us now examine the performance of filters based on each bounding and call the corresponding fixed parameter Kalman filters A and B .

Simulation results has shown that, A filter corresponding to $(\sigma_{mA}, \alpha_{mA}) = (0.5,\ 1/20)$ is narrow banded in that its position estimates are relatively smooth. Unfortunately, A filter values tend to diverge or at least lag severely from the actual (simulated) values whenever the acceleration (maneuver) changes rapidly. The estimates of B filter , whose parameter set is given by $(\sigma_{mB}, \alpha_{mB}) = (5.0, 1/10)$, tend to be unbiased since it is very

wide banded , and never really divergent. Indeed, we can not make an interpretation about the divergence of B , since it never diverges or converges exactly. B filter estimates always contain a lot of uncertainty even when we observe that A filter estimates are quite smooth and accurate with exception that they tend to lag considerably.

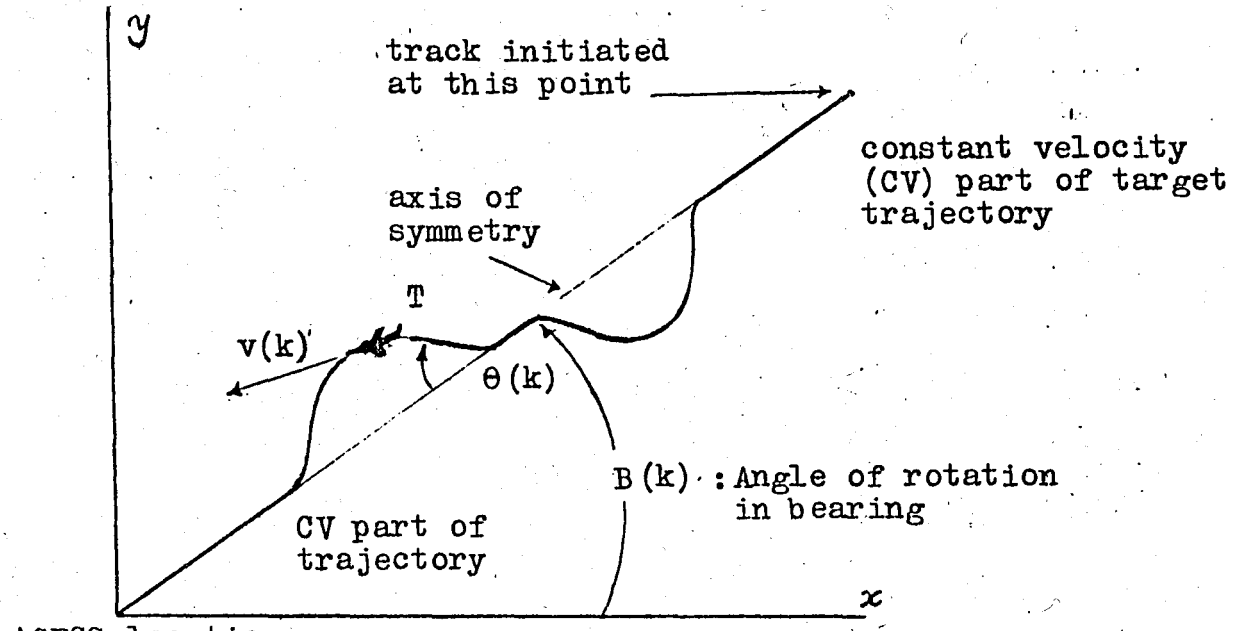


fig. 4. 1. Simulated aircraft trajectory

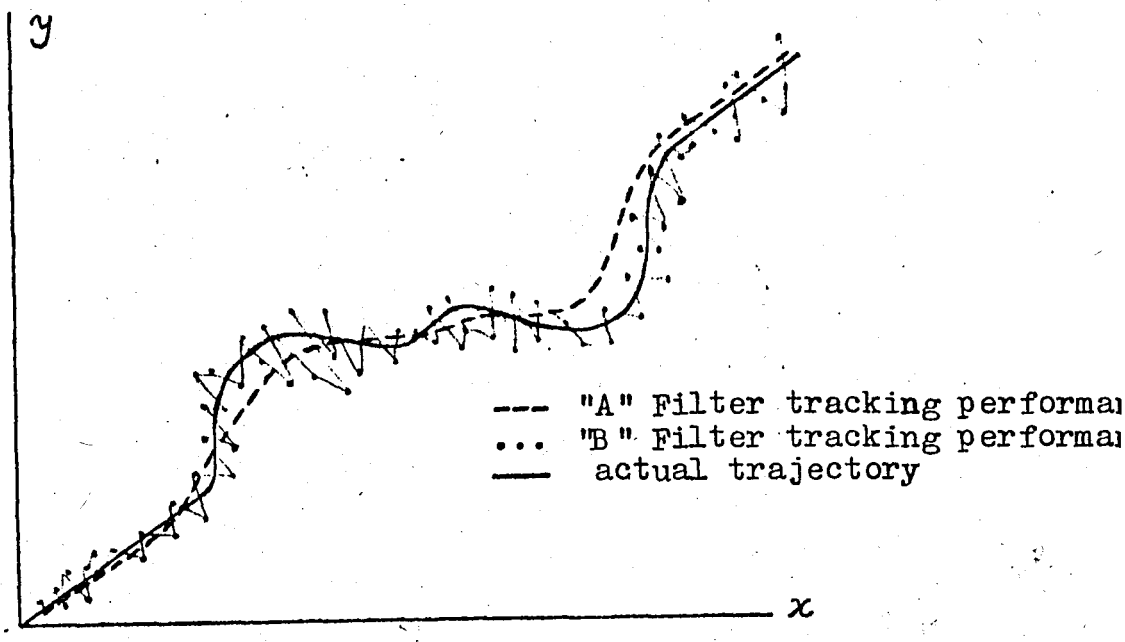


fig. 4. 2. Expected tracking performances

The paradox is that , while each filter has its advantages and disadvantages, neither is really suitable for implementation as a tracking filter. Obviously, we should look for some method of adapting-on-line the bandwidth of the filter to achieve the desired performance.

When divergence occurs, the error vector

$$\tilde{\underline{x}}(k/k) = \underline{x}(k) - \hat{\underline{x}}(k/k) \tag{4.8}$$

grows large. Fortunately , we are able to monitor at least partially the actual performance of the filter at any time. This can be done by observing the innovations sequence and then attempting to the built up of a bias and consequent growth of residuals. We can easily determine the statistics of residuals as :

$$\begin{aligned} E\{\nu(k/k-1)\} &= E\{\underline{z}(k) - \underline{\underline{H}}(k)\hat{\underline{x}}(k/k-1)\} \\ &= E\{\underline{\underline{H}}(k)\underline{x}(k) + v(k) - \underline{\underline{H}}(k)\hat{\underline{x}}(k/k-1)\} \\ &= E\{\underline{\underline{H}}(k)[\underline{x}(k) - \hat{\underline{x}}(k/k-1)] + \underline{v}(k)\} \\ &= \underline{\underline{H}}(k)E\{\tilde{\underline{x}}(k/k-1)\} + E\{\underline{v}(k)\} \\ &= \underline{0} \end{aligned}$$

and covariance ;

$$\begin{aligned} E\{\nu(k/k-1)\nu^T(k/k-1)\} &= E\{[\underline{\underline{H}}(k)[\underline{x}(k) - \hat{\underline{x}}(k/k-1)] + \underline{v}(k)] \\ &\quad \cdot [\underline{v}^T(k) + [\underline{x}(k) - \hat{\underline{x}}(k/k-1)]^T \underline{\underline{H}}^T(k)] \\ &= E\{\underline{\underline{H}}(k)\tilde{\underline{x}}(k/k-1)\underline{v}^T(k) \\ &\quad + \underline{\underline{H}}(k)\tilde{\underline{x}}(k/k-1)\tilde{\underline{x}}^T(k/k-1)\underline{\underline{H}}^T(k) \\ &\quad + \underline{v}(k)\underline{v}^T(k) + \underline{v}(k)\tilde{\underline{x}}(k/k-1)\underline{\underline{H}}^T(k)\} \\ &= \underline{\underline{R}}(k) + \underline{\underline{H}}(k)\underline{\underline{P}}(k/k-1)\underline{\underline{H}}^T(k) \end{aligned}$$

In the above term, substituting the matrices for our model, we

find that the residual sequence should be zero mean and has variance :

$$\sigma_{\nu}(k) = \sigma_r^2 + p_{11}(k/k-1) \qquad (4.9)$$

since

$$H = [1 \quad 0 \quad 0]$$

and

$$\underline{\underline{R}}(k) = \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & r^2(k)\sigma_B^2 & 0 \\ 0 & 0 & r^2(k)\sigma_E^2 \end{bmatrix}$$

i. e. under the assumption of channel independence .

It is then a relatively simple matter to determine the probability that the sampled residual belongs to the population with above statistics . The actual residual is comprised of exactly the same error terms and the sample expected value is related in the same manner to the actual residual error and measurement covariance . Instead of working with an individual, greater statistical significance can be obtained by considering several data samples . For example, we can normalize each residual sample with $\sigma_{\nu}(k)$ to remove the transient behaviour of the error covariance , and compute the normalized sample mean ;

$$\nu_N(k) = \frac{1}{M} \sum_{i=k-M}^{k} \nu(i/i-1) \; / \; \sigma_{\nu}(i) \qquad (4.10)$$

It is easily seen that ν_N is ideally also a zero mean normally distributed random variable of variance $1/M$. This can be verified as follows :

$$\bar{\nu}_N(k) \sim \mathcal{N}(0, 1/M)$$

$$E\{\bar{\nu}_N(k)\} = E\left\{\frac{1}{M}\sum_{i=k-M}^{k}\frac{\nu(i/i-1)}{\sigma_\nu(i)}\right\}$$

$$= \frac{1}{M}\ E\left\{\sum_{i=k-M}^{k}\frac{\nu(i/i-1)}{\sigma_\nu(i)}\right\} = 0$$

as found in the previous proof .
Also ,

$$E\{\bar{\nu}_N^{\,2}(k)\} = E\left\{\left[\frac{1}{M}\sum_{i=k-M}^{k}\frac{\nu(i/i-1)}{\sigma_\nu(i)}\right]^2\right\}$$

$$= E\left\{\frac{1}{M^2}\sum_{i=k-M}^{k}\frac{\nu^2(i/i-1)}{\sigma_\nu^2(i)}\right\}$$

$$+ E\left\{\frac{1}{M^2}\sum_{i=k-M}^{k}\sum_{\substack{i=k-M\\ i\neq j}}^{k}\frac{\nu(i/i-1)}{\sigma_\nu(i)}\cdot\frac{\nu(j/j-1)}{\sigma_\nu(j)}\right\}$$

$$= \left[\frac{1}{M}\right]^2\left[ME\left\{\frac{\nu^2(i/i-1)}{\sigma_\nu^2(i)}\right\} + M(M-1)\left[E\left\{\frac{\nu(i/i-1)}{\sigma_\nu(i)}\right\}\right]^2\right]$$

$$= \frac{1}{M^2}\cdot M\cdot\frac{\sigma_\nu^2(i/i-1)}{\sigma_\nu^2(i)}$$

$$= \frac{1}{M}$$

We will define a maneuver as any target motion that causes the filter performance as measured by $\nu_N(k)$ to exceed some specified value .Namely, a maneuver is declared, if ;

$$\left| \bar{\nu}_N(k) \right| > \frac{1}{\sqrt{M}} c \qquad (4.11)$$

where c is a constant that determines the significance of the test . The probability of false detection or type I error is ;

$$P_{fd} = \text{prob}\left\{ \left| \bar{\nu}_N(k) \right| > c\frac{1}{\sqrt{M}} \right\} \qquad (4.12)$$

The values of P_{fd} as a function of c can be found in most introductory statistics books . In order to choose a value of c , we must consider the false detection probability in conjunction with the cost of such false detections . These costs are generally a function of the type of action taken to adapt whenever a maneuver is declared . A trade-off is involved since there is a cost in increased maneuver detection time, when the maneuver threshold c is raised . There are some techniques to find a functional dependence of c on maneuver detection time t_D , however, such tecniques are not of exact science with some misleading implications they bring about . So , in our work we choose to investigate the effects of the selection of c on simulation studies, rather than sit down to find an incorrect functional form .

Once divergence has been detected, a method of modifying or adapting the filter parameters to correct the situation must be specified .

Let us return to the original assumption that the target maneuver level -- and subsequent process noise -- is bounded by A and B parameter sets . Therefore , in the absence of the maneuver detection, process noise corresponding to the set A will be added and if maneuver is declared, Q_B corresponding to set B will be added . That is ;

$$\text{if} \quad \bar{\nu}_N(k) < \frac{1}{\sqrt{M}} c \quad \text{then} \quad \underline{\underline{Q}}(k-1) = \underline{\underline{Q}}_A \qquad (4.13a)$$

$$\text{if} \quad \bar{\nu}_N(k) > \frac{1}{\sqrt{M}} c \quad \text{then} \quad \underline{\underline{Q}}(k-1) = \underline{\underline{Q}}_B \qquad (4.13b)$$

4.3. Dual-Bandwidth Adaptation

Ideally, when a maneuver has been detected, one would likely reprocess the measurements over some interval immediately preceding current time with a wider bandwidth filter so as to remove the bias error which has presumably been occuring.

Unfortunately, in our application we are supposed to built up a fully recursive filter for implementing the same on a limited real-time computer in an efficient way. It would be very difficult to interrupt normal processing in order to reprocess the past data. One possible solution would be the built-up of a second filter operating in parallel to the "main" filter with a wider bandwidth to remove the bias error once it has been detected,

The dual-bandwidth filter operates as follows:
Two filters, A and B, corresponding to the respective maneuver parameter bounds would operate simultaneously. The filter algorithm would output the state vector of filter A. If divergence of filter A is detected, the state vector of filter B, which is expected to be unbiased, is put into filter A, i.e.

$$\text{if} \quad |\bar{\gamma}_{NA}(k)| > \frac{1}{\sqrt{M}}\, c \qquad \text{then} \quad \hat{\underline{x}}_A(k/k) = \hat{\underline{x}}_B(k/k)$$

Conceptually, we want the adaptation to work as in the block diagram shown in fig.4.3.

Admittedly, the output vector $\hat{x}$ will be discontinuous, but in this situation $\hat{x}_B$ represents the best information available. An important consideration now is the way we should modify the A filter bandwidth. Theoretically, if one attempts to reset the state estimate of A to B, the bandwidth should be similarly reset. This is not an acceptable solution for one pays a high cost of a false detection since long reconvergence time has not yet been eliminated.

Leaving the A filter bandwith unchanged is desirable since this filter did not suffer from these disadvantages. The particular maneuver detector we have used, however, may not do well since the large random errors of the B filter may look like biases to the A filter and consequently, cyclic maneuver detections

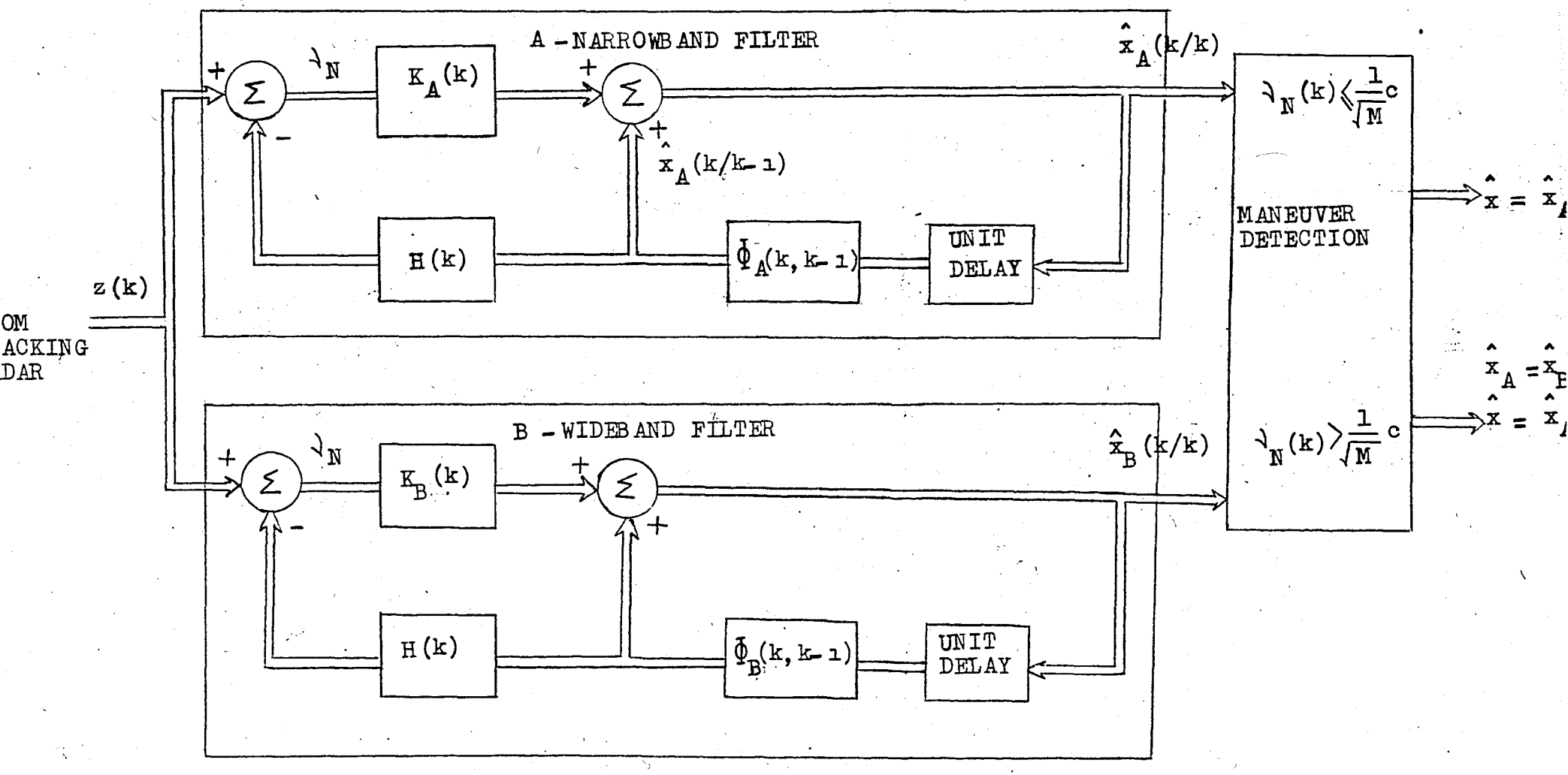

A - NARROWBAND FILTER
ν_N
$K_A(k)$
$\hat{x}_A(k/k)$
$\hat{x}_A(k/k-1)$
$H(k)$
$\Phi_A(k, k-1)$
UNIT DELAY
$z(k)$
OM
ACKING
DAR
B - WIDEBAND FILTER
$K_B(k)$
$\hat{x}_B(k/k)$
$\Phi_B(k, k-1)$
MANEUVER DETECTION
$\nu_N(k) \leqslant \frac{1}{\sqrt{M}} c$
$\nu_N(k) > \frac{1}{\sqrt{M}} c$
$\hat{x} = \hat{x}_A$
$\hat{x}_A = \hat{x}_B$
$\hat{x} = \hat{x}_B$

fig. 4.3

tend to occur .

To eliminate this problem ,

$$\underline{\underline{P}}_A = \underline{\underline{P}}_A + \underline{\underline{Q}}_B$$

is proposed, namely, a gradual widening of the A filter bandwidth which is expected to eliminate the repeated maneuver detections and maintained a low cost of false detections .

4. 4. Alternative Adaptation Techniques

a. Switch-on-range Adaptation

The simplest of the three adaptation techniques presented in this section is the switch-on-range adaptation . This method is based on the idea that the measurement error in Cartesian coordinates is approximately proportional to the target range. Therefore, we can use relatively high filter gains on a short range target without significant degradation of the estimate quality for a constant velocity target and with significant improvement of the estimate quality for a maneuvering target.

Upon target detection, the estimator uses a gain schedule which is based on a small forcing input -- such as 0.1 g -- in each coordinate axis . This low Q gain schedule is continued until the target range is less than a predetermined threshold. Once the target has reached this threshold range, the estimator switches to a steady-state gain matrix which is based on relatively high forcing inputs (high Q) . This adaptation method, however, is proposed for constant velocity target models at long range .

The 'high Q' filter uses a steady-state gain matrix because it is assumed that a good estimate is available at the time switching occurs .

This adaptation scheme has been presented here for reference purposes and since it depends on the utilization of a different model, simulation studies have not been performed .

b. Switch-on-residuals Adaptation (1)

This method of adaptation is more sophisticated and more complex than the switch-on-range adaptation method previously described . The first version of the switch-on-residuals adaptive

filter is designed to track with a 'low Q' gain schedule based on the constant velocity model until a component of the residual vector of the filter ;

$$\underline{\nu}(k) = \underline{z}(k) - \underline{\underline{H}}(k)\hat{\underline{x}}(k/k-1)$$

exceeds a calculated gating level

$$l_i(k) = 2\sqrt{P_{ii}(k/k) + R_{ii}(k)} \quad , \quad i = 1, 2, 3 . \qquad (4.14)$$

on two successive samples.

$P_{ii}(k/k)$ is the theoretical variance of position estimate error and $R_{ii}(k)$ is the theoretical variance of measurement noise in one coordinate direction . These two values are taken from the steady-state theoretical covariance of estimation error matrix for the model from which the 'high Q' gains are calculated . The values for the theoretical variance of measurement noise in the gating level calculation are calculated on-line from the diagonal terms of the covariance matrix of measurement noise given in chapter III , eq'n (3.17) and vary with the target position.

If the occurence of a residual value greater than the gating level at time i is defined as event A_i and probability of this occuring on the next sample is event A_{i+1} , A_i and A_{i+1} are independent events . The probability of having two successive samples with residuals greater than the gating level is :

$$P(A_i A_{i+1}) = P(A_i)P(A_{i+1}) \qquad (4.15)$$

If one considers the $R_{ii}(k)$ term of the equation (4.14), one sees that the gating level is equal to twice the standart deviation of measurement noise . For a normal distribution, the probability of an event differing from the mean by more than two standart deviations is given as :

$$P(A_i) = P(A_{i+1}) = 0.0228$$

Substituting this value into the equation (4.15) yields :

$$P(A_i A_{i+1}) = (0.0228)^2 = 5.20 \cdot 10^{24}$$

which is the probability that a residual is greater than the gating level because of estimation error or measurement noise when a maneuver of the target is experienced .

The filter is independently adaptive in each coordinate direction . For example, when $\gamma(k)$ is greater than $l_1(k)$ for two successive samples, the x-direction states are re-initialized by accepting the latest measurement as the position estimate and beginning at k=0 with a 'high Q' gain schedule based on the random acceleration model with values of σ_m and α_m and assumed forcing inputs chosen to provide good performance against a maneuvering target . The y-direction and z-direction state estimates continue to be based on the CV model . The random acceleration model filter gains continue to be used in making the x-direction estimates until three successive samples have residuals less than one half the gating level .

When the inequality ;

$$\gamma_1(k) = \sqrt{P_{11}(k/k) + R_{11}(k)}$$

is satisfied for three successive samples, the filter switches back to the low Q gain schedule assuming the same time index as would have been used for the current sample if no adaptive switching had occured .

c. Switch-on-residuals Adaptation (2)

This method of adaptation is essentially not a new one since it differs from the first version in that it calculates the switching gate using the equation ;

$$l_i(k) = 3\sqrt{P_{ii}(k/k) - R_{ii}(k)} \quad , \quad i = 1, 2, 3 . \qquad (4.16)$$

Again, switching to the 'high Q ' gain schedule occurs when the above gating level is exceeded by one sample residual .

The switch back to the 'low Q' gain schedule occurs when the residual is $0.88\, l_i(k)$ for two successive samples (the

coefficient 0.88 is an arbitrary choice for a value between 0.5 and 1.) . In this version, when the filter switches back to the 'low Q' gains, the index for the 'high Q' gain is not reset immediately but is delayed one sample so that if the residual exceeds the gating level on the first sample after the filter returns to the 'low Q' gain schedule, the filter does not re-initialize by accepting the position measurement as the estimate but simply switches back to the 'high Q' gain schedule as would have been used if the switch to the 'low Q' gain schedule had not taken place .

The gain schedules used in this version of the switch-on-residuals adaptation are the same as those used in the first version described in the previous section .

V. COMPUTATIONAL CONSIDERATIONS

Several aspects of the problem have so far been investigated in detail but the problem of large computational burden superimposed by Kalman's conventional filtering algorithm has been disregarded deliberately . As we shall discuss in sections dealing with simulation results, this computational difficulty can be ignored as far as large scale digital computers like IBM-4341 or CDC170/815 are concerned . Unfortunately, however, it would not be practical to dedicate this class of computers in applications such as shipboard tracking of low speed surface targets -- even when intolerable costs are taken as granted .

The problem can therefore be stated as follows :

Design an adaptive target tracking filter which is capable of processing high rate data of radar observations and capable of propagating the covariances at a speed that is compatible to the speed of approaching "real" targets .

The main points of concern for a solution should therefore be on the computational requirements of the filtering algorithm which is to be implemented on a small digital computer .

In our development of the three-dimensional filter, the dual-bandwidth adaptive feature serves to multiply these computational requirements. Obviously, anything that we can do to reduce the calculation required by the basic one-dimensional filter could be of importance .

In this section we aim to examine two possible techniques for the reduction of computational burden :

a. Prefiltering

The term "prefiltering" means processing radar data at a higher rate than the filter cycles and if this can be achieved without severely degrading performance , much of the problem is solved . The high cycling rate (time required to process a single set of measurements by the filter) can be offset by this kind of a data compression technique . The real-time propagation of

error covariance on a fixed-point computer of relatively short word-length , say,16 bits poses other problems .

Considering the potential range of the elements of error covariance matrix P which involves the squares of both rather large and rather small numbers during each run, it was found that such a computer would have to perform the covariance calculations in double precision in order to achieve sufficient scaling .These problems can be eliminated through the introduction of error covariance square root which can be propagated in place of the covariance .

Even though the high data rate is one reason of computational burden, it serves to reduce the effective measurement noise level .By this we mean that taking radar data at lower rates would not be of practical significance as a "solution".

Data compression techniques can be used to achieve effective high data rates without actually running the Kalman filter at such rates .Prefiltering in our specific application means the processing of data which is available at a higher rate than the rate at which we wish to cycle the Kalman filter .Suppose, as in fig.5.1 that we wish to process data at an integer rate μ times the filter cycling rate, Δt .We will therefore have μ measurements equally spaced $\Delta t/\mu$ apart, that will have been made since the last filter cycle at time $t(k-1)$ and which we want to process at time $t(k)$ .There are undoubtedly several possible methods of aggregating these additional measurements.. However, we will consider the simplest effective method of doing this, namely , averaging .For short time intervals, where the measurement noise essentially masks any time variations in the signal, data averaging is an effective means of data compression with small loss of information . There is actually some velocity information that could be extracted from measurements. The variance of this velocity estimate, however, is often so large, relative to that already available in the Kalman filter, that its inclusion makes essentially no improvement .There is, however, a significant increase in the computation required to process such a velocity " measurement" .We will therefore compute an equivalent prefiltered measurement based upon a technique

similar to data averaging -- residual averaging .
By averaging the 'a priori' residual as opposed to the current measurements we account for the estimated target motion over the prefiltered interval .

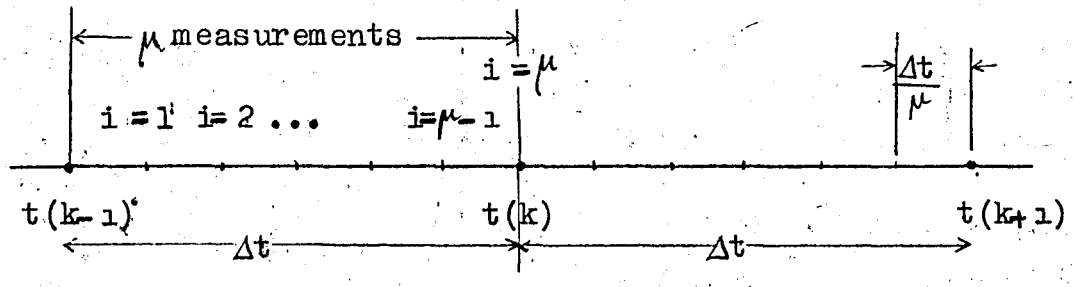


fig. 5.1

The averaged or prefiltered residual is simply :

$$\gamma_{pf}(k/k-1) = \frac{1}{\mu} \sum_{i=1}^{\mu} \gamma(k-1 + \frac{i}{\mu}/k-1) \qquad (5.1)$$

Substituting the definition for the 'a priori' residual, one finds the effective prefiltered measurement as :

$$z_{pf}(k) = z(k) + \mu_2 x_2(k/k-1) - \mu_3 x_3(k/k-1) \qquad (5.2)$$

where

$$\mu_2 = \Delta t \left[1 - \frac{1}{\mu^2} \sum_{i=1}^{\mu} i \right]$$

$$\mu_3 = \tau_m^2 \left[\frac{1}{\mu\gamma} \sum_{i=1} \gamma^{i/\mu} - 1 \right] - \tau_m \mu_2$$

are constants to be computed once before the implementation .

We also noted that, if $\mu = 1$, then μ_2 and μ_3 vanish, and the prefiltered measurement z_{pf} becomes z as before.

Now let us consider the effect of prefiltering on the measurement error statistics. Obviously, an error in z_{pf} is a function of the errors $\tilde{x}_2(k/k-1)$ and $\tilde{x}_3(k/k-1)$, as well as the measurement errors. However, consideration of small values of μ_2 and μ_3 reveal these effects to be usually negligible. Under this assumption, we find that the prefiltered measurement error is simply the average of the individual errors in measurements.

The variance of the prefiltered measurement error is : (assuming the measurement error standart deviation is constant)

$$\frac{\sigma_{pf(k)}^2}{\sigma^2(k)} = \frac{1}{\mu^2} \sum_{i=1} \sum_{j=1} \rho(k)^{|i-j|/\mu} \qquad (5.3)$$

It should be noted that, in white noise case $\rho(k)$ term vanishes $\sigma_{pf}^2(k)/\sigma^2(k)$ then reduces to $1/\mu$.

Let us now consider the effect of prefiltering on the filter performance. It can be found that there are three factors to be considered. The increase in the time increment between filter cycles (a reduction in the filter cycle rate) and the increase in the autocorrelation of the prefiltered error both tend to increase covariance thus degrade performance. Fortunately the decrease in the prefiltered measurement error variance tends to decrease the error covariance thus improve filter performance. Study of the prediction performance vs. μ is given in [7] in more detail. In conclusion, the method of prefiltering based upon residual averaging offers a computationally inexpensive way to achieve high data rate filtering with negligible sacrifice in performance as compared to high data rate full Kalman filtering.

b. Square root formulation of covariance propagation equations

The square root formulation can propagate the error covariance

in single precision as accurately as the conventional error covariance methods do in double precision .

Algorithm for square root filtering :

1. Dynamic extapolation of the preceding state

$$\hat{\underline{x}}(k/k-1) = \underline{\underline{\Phi}}(k,k-1)\hat{\underline{x}}(k-1/k-1) \tag{5.4}$$

$$\begin{matrix} n \\ m \end{matrix}\begin{bmatrix} \underline{\underline{S}}^T(k/k-1) \\ \hline 0 \end{bmatrix} = \underline{\underline{T}} \begin{bmatrix} S^T(k-1/k-1)\ \underline{\underline{\Phi}}^T(k/k-1) \\ Q^{1/2}(k) \end{bmatrix} \tag{5.5}$$

n

where

$$\underline{\underline{P}}(k/k-1) = \underline{\underline{S}}(k/k-1)\underline{\underline{S}}^T(k/k-1) \tag{5.6}$$

In the above equation, the matrix T is orthogonal, but otherwise any matrix making $S^T(k/k-1)$ upper-triangular .

The matrix S in the second of above equation denotes the square root of P .The definition of S ,however is not unique . It is quite simple to write out the matrix transformations to obtain upper and lower triangular S matrices .The construction of T is a task to which much attention should be given. There are methods, however, proposed by Householder, Gram-Schmidt and Givens .These manipulations are rather lengthy and will not be reproduced here .

In chapter III , we have verified the assumption of diagonal measurement error covariance matrix

$$\underline{\underline{R}}(k) = \underline{\underline{\Lambda}}(R^1 , R^2, \ldots , R^r) \qquad \forall k$$

If this is not the case, however, the Cholesky decomposition algorithm is readily applicable to determine a transformation producing a problem in which R is diagonal.

The most useful implication of this assumption is the application of sequential data processing .This approach can

be summarized as follows ;

Let $H^i(k)$ denote the i'th column of $H(k)$ and $z^i(k)$ the i'th entry of $z(k)$ with

$$\hat{\underline{x}}^o = \hat{\underline{x}}(k/k-1) \tag{5.7a}$$

$$\hat{\underline{x}}^r = \hat{\underline{x}}(k/k) \tag{5.7b}$$

$$\underline{\underline{S}}^o = \underline{\underline{S}}(k/k-1) \tag{5.7c}$$

and

$$\underline{\underline{S}}^r = \underline{\underline{S}}(k/k) \tag{5.7d}$$

one obtains for $i = 1, 2, \ldots, r$

$$D^i = (S^{i-1})^T H^i \tag{5.8a}$$

$$\alpha^i = \left[(D^i)^T D^i + R^i \right]^{-1} \tag{5.8b}$$

$$\gamma^i = (1 + \sqrt{\alpha^i R^i})^{-1} \tag{5.8c}$$

$$S^i = S^{i-1} - \gamma^i \alpha^i S^{i-1} D^i D^{iT} \tag{5.8d}$$

and finally ,

$$\hat{x}^i = \hat{x}^{i-1} + \alpha^i S^{i-1} D^i \left[z^i(k) - H^{iT} \hat{x}^{i-1} \right] \tag{5.8e}$$

Of course, these equations define a sequence of updates corresponding to a sequence of scalar measurements , and agree with the original algorithm .

It can be stated that the accuracy is maintained by means of the square root technique implemented on a fixed point computer in which case conventional algorithm in single precision arithmetic would give rise to performance degradation because of round-off errors .

To summarize, we have found that it is possible to reduce the computational burden by a factor of μ by prefiltering with negligible loss of performance .The choice of a value for

depends on the cycling rate of the filter and on the data rate specified for a particular sensor type .By using the square root covariance filter, it is possible to eliminate double precision arithmetic in covariance calculation, thereby significantly reducing the time required for covariance propagation .The combination of these two techniques to overcome computational difficulties is then expected to reduce the processing time of high data rate covariance filter significantly .These reductions are in fact compulsory since the filter design performed in our study requires two filters operating in parallel and the three-dimensional geometry of the system poses the problem of three independent channels with three independent data processing requirements .As pointed out earlier, real-time application of the tracker under consideration must fulfill the computational requirements mentioned throughout this chapter .

VI. ALTERNATIVE FILTERING METHODS

In the very beginning of this study, the nonlinear structure of the problem was discussed. The efforts for modeling the process in a form which is directly accessible by the standart discrete linear filtering algorithms have been involved in the subsequent chapters. In this chapter we aim to survey some filtering techniques which can be considered as alternatives to conventional filter in case when the designer can sacrifice computational time to accuracy in his particular application.

6.1. Extended Kalman Filter

Starting with the nonlinear system ;

$$\frac{d\underline{x}_t}{dt} = \underline{f}(\underline{x}_t, t) + \underline{\underline{G}}(t)\underline{w}_t \qquad t \geqslant t_o\,,\ \underline{x}_{t_o} \sim \mathcal{N}(\hat{\underline{x}}_{t_o}, \underline{\underline{P}}_{t_o}) \tag{6.1a}$$

$$\underline{y}_{t_k} = \underline{h}(\underline{x}_{t_k}, t_k) + \underline{v}_k \tag{6.1b}$$

one obtains the linearized discrete system (linearization is done about the solution of noise free dynamics)

$$\delta\underline{x}_{t_{k-1}} = \underline{\underline{\Phi}}\left[t_{k-1}, t_k;\ \bar{\underline{x}}(t_k)\right]\delta\underline{x}_{t_k} + \underline{w}_{t_{k-1}} \tag{6.2a}$$

$$\delta\underline{y}_{t_k} = \underline{\underline{H}}\left[t_k;\ \bar{\underline{x}}(t_k)\right]\delta\underline{x}_{t_k} + \underline{v}_k \tag{6.2b}$$

with

$$\underline{x}_{t_o} \sim \mathcal{N}(\hat{\underline{x}}_{t_o} - \bar{\underline{x}}(t_o)\ ,\ \underline{\underline{P}}_{t_o})$$

It can be seen that the linear filter is directly applicable to the linearized system .

Instead of 'state' and 'measurement' we now speak of the state deviation and measurement deviation . For a given trajectory and set of measurements, one can compute δy_{t_k} with

$$\delta \underline{y}_{t_k} \triangleq \underline{y}_{t_k} - \bar{\underline{y}}(t_k)$$

and process the measurement deviations through the linear filter to estimate the state deviations . From the definition of perturbation,

$$\delta \underline{x}_{t} \triangleq \underline{x}_{t} - \bar{\underline{x}}(t)$$

it follows that

$$\hat{\underline{x}}(t_k/t_k) = \bar{\underline{x}}_{t_k} - \hat{\delta \underline{x}}(t_k/t_k) \tag{6.3}$$

and corresponding estimation error is given by :

$$\begin{aligned} \tilde{\underline{x}}(t_k/t_k) &\triangleq \underline{x}_{t_k} - \hat{\underline{x}}(t_k/t_k) \\ &= \delta \underline{x}_{t_k} - \hat{\delta \underline{x}}(t_k/t_k) \\ &= \tilde{\delta \underline{x}}(t_k/t_k) \end{aligned} \tag{6.4}$$

so that

$$\underline{\underline{P}}(k/k) = \operatorname{cov}\left\{\tilde{\underline{x}}(t_k/t_k)\right\} = \operatorname{cov}\left\{\tilde{\delta \underline{x}}(t_k/t_k)\right\} \tag{6.5}$$

The remaining problems are the choice of the reference trajectory and the question of validity of the linearized equations.

One would prefer to use a reference trajectory with $\bar{x}(t_o) = \hat{x}_{t_o}$, the prior estimate of the state . Then ,

$$\delta \underline{x}_{t_o} \sim \mathcal{N}(\underline{0}, \underline{\underline{P}}_{t_o})$$

and evidently ,

$$E\left\{\delta\underline{x}_{t_k}\right\} = \underline{0} \qquad \forall t_k$$

As we process the observations ,

$$\delta\underline{x}(t_k/t_k) = E\left\{\delta\underline{x}_{t_k} / \underline{Y}_k\right\} \neq 0 \qquad k>0$$

Generally, the initially wise choice of a reference trajectory may turn out to be a poor one . This is especially true when P_{t_o} is large , which implies higher level of uncertainty in the estimate $\hat{x}_{t_o}$, or if the process noise is large . In this case the estimates of state deviations can become large, violating our linearity assumptions.

Assuming that the process is a noise-free one , then one can imagine that there exists a "true" trajectory of the system, which can be learned in the precence of sufficient observations. Having chosen $\bar{x}(t_o) = \hat{x}_{t_o}$, we filter the batch of observations

$$\{ y_1, y_2, \cdots , y_k \}$$

and obtain $\delta\hat{x}(t_k/t_k)$. Then predicting backward to t_o ,

$$\delta\hat{\underline{x}}(t_o/t_k) = \underline{\underline{\Phi}}\left[t_o, t_k; \bar{\underline{x}}(t_o)\right] \delta\hat{\underline{x}}(t_k/t_k) \qquad (6.6)$$

If our batch of observations is sufficiently large, then we can expect ;

$$\bar{\underline{x}}'(t_o) = \bar{\underline{x}}(t_o) + \delta\hat{\underline{x}}(t_o/t_k) \qquad (6.7)$$

to be closer to the "true" initial state $x(t_o)$ than $\bar{x}(t_o)$ is. Taking $\bar{x}'(t_o)$ now as the reference initial condition we can reprocess our batch of observations, that is , linearize about $x'(t)$, and run the filter for the linearized system over the same batch of observations . This procedure can be repeated

several times until the last equation produces no change in the reference initial condition .Presumably, convergence to the true trajectory will have been obtained .This procedure is often call ed "global iteration " of the Kalman filter .This kind of iteration is commonly used in least squares estimation techniques . More precisely, the global iteration we have described is almost equivalent to iterated least squares estimation techniques and undermines the advantages of Kalman filter .Furthermore, convergence is not guaranteed, i.e. it may diverge as well, requiring alternate guesses of reference trajectory .

As a further step, one can relinearize about each new estimate as these become available .At t_o, linearize about $\hat{x}_{t_o}$, and once y_1 is processed, relinearize about $\hat{x}(t_1/t_1)$ .. and so on .This procedure results in a better reference trajectory as soon as one is obtained .As a consequence , large initial estimation errors are not allowed to propagate through time, not violating our linearity assumptions .It should be noted that this advantage is not available in least squares batch estimation techniques .

Now, if we initially linearize about $\hat{x}_{t_o}$, then

$$\delta\hat{\underline{x}}(t_o/t_o) = \underline{0}$$

and in view of equation (6.13) when applied to deviations

$$\delta\hat{\underline{x}}(t_1/t_1) = \underline{0}$$

since we subsequently linearize about $\hat{x}(t_1/t_1)$

$$\delta\hat{\underline{x}}(t_1/t_1) = 0$$

so that again ,

$$\delta\hat{\underline{x}}(t_2/t_1) = 0$$

and, in general ;

$$\delta\hat{\underline{x}}(t/t_k) = \underline{0} \qquad t_k \leqslant t \leqslant t_{k+1} \ , \ \forall k \tag{6.8}$$

As a result, between observations, the best estimate of the state is the reference state, and accordingly,

$$\frac{d\hat{\underline{x}}(t/t_k)}{dt} = \underline{f}(\hat{\underline{x}}(t/t_k), t) \quad ; \quad t_k \leqslant t \leqslant t_{k+1} \tag{6.9}$$

Prediction is accomplished using the nonlinear dynamics described earlier with $w_t = 0$

The equation for the correction to the estimate at an observation for the linearized system is given by :

$$\delta\hat{\underline{x}}(k+1/k+1) = \delta\hat{\underline{x}}(k+1/k) + \underline{\underline{K}}(k+1)\left[\delta\underline{y}_{k+1} - \underline{\underline{H}}(k+1)\delta\hat{\underline{x}}(k+1/k)\right] \tag{6.10}$$

Since

$$\delta\hat{\underline{x}}(k+1/k+1) = \hat{\underline{x}}(k+1/k+1) - \hat{\underline{x}}(k+1/k)$$

In view of linearization and using (6.8)

$$\hat{\underline{x}}(k+1/k+1) = \hat{\underline{x}}(k+1/k) + \underline{\underline{K}}(k+1)\left[\underline{y}_{k+1} - \underline{h}(\hat{\underline{x}}(k+1/k), t_{k-1})\right] \tag{6.11}$$

The above equation completes the derivation of extended or modified Kalman filter . This derivation can be given in the form of a theorem.

Theorem 6.1. Extended Kalman Filter

The extended Kalman filter for the nonlinear system given by (6.1) consist of the prediction via.

$$\hat{\underline{x}}(t_{k+1}/t_k) = \underline{x}(t_k/t_k) + \int_{t_k}^{t_{k+1}} \underline{f}(\underline{x}(t/t_k), t)dt$$

$$\underline{\underline{P}}(t_{k+1}/t_{k+1}) = \underline{\underline{\Phi}}\left[t_{k+1}, t_k; \hat{\underline{x}}(t_k/t_k)\right] \underline{\underline{P}}(t_k/t_k)\ \underline{\underline{\Phi}}^T\left[t_{k+1}, t_k; \hat{\underline{x}}(t_k/t_k)\right] + \underline{\underline{Q}}(k$$

and at an observation,

$$\hat{\underline{x}}(t_{k+1}/t_{k+1}) = \hat{\underline{x}}(t_{k+1}/t_k) + \underline{\underline{K}}\left[t_{k+1}; \hat{\underline{x}}(t_{k+1}/t_k)\right] \cdot$$

$$\left[\underline{y}_{t_{k+1}} - \underline{h}(\hat{\underline{x}}(t_{k+1}/t_k), t_{k+1})\right]$$

$$\underline{\underline{P}}(t_{k+1}/t_{k+1}) = \left[\underline{\underline{I}} - \underline{\underline{K}}\left\{t_{k+1}; \hat{\underline{x}}(t_{k+1}/t_k)\right\} \underline{\underline{H}}\left\{t_{k+1}; \hat{\underline{x}}(t_{k+1}/t_k)\right\}\right] \underline{\underline{P}}(t_{k+1}/t_k)$$

$$\cdot \left[\underline{\underline{I}} - \underline{\underline{K}}\underline{\underline{H}}\{\cdot\}\right]^T + \underline{\underline{K}}\{\cdot\}\underline{\underline{R}}(k+1)\underline{\underline{K}}^T\{\cdot\}$$

The Kalman gain is ,

$$\underline{\underline{K}}\left[t_{k+1}; \hat{\underline{x}}(t_{k+1}/t_k)\right] = \underline{\underline{P}}(t_{k+1}/t_k)\underline{\underline{H}}^T\left[t_{k+1}; \hat{\underline{x}}(t_{k+1}/t_k)\right]$$

$$\cdot \left[\underline{\underline{H}}\left\{t_{k+1}; \hat{\underline{x}}(t_{k+1}/t_k)\right\}\underline{\underline{P}}(t_{k+1}/t_k)\right.$$

$$\left. \cdot \underline{\underline{H}}^T\left\{t_{k+1}; \hat{\underline{x}}(t_{k+1}/t_k)\right\} + \underline{\underline{R}}(k+1)\right]^{-1}$$

The O and H matrices are those of the linearized system (6.2) . It has been emphasized that these matrices depend on $\hat{x}(t_k/t_k)$ or, equivalently, on $\hat{x}(t_{k+1}/t_k)$ by including these estimates as arguments .

6. 2. Fading Memory Filter

The fading memory filter (sometimes referred to as exponentially aging filter) weights recent data exponentially higher than past data . In the recursive least squares derivation of the linear filter in [1] a detailed discussion is given . We are to omit this derivation and present the results .

The filter gain matrix for this version is ,

$$\underline{\underline{K}}_e(k+1) = \underline{\underline{P}}(k+1/k)\underline{\underline{H}}^T(k+1)\left[\underline{\underline{H}}(k+1)\underline{\underline{P}}(k+1/k)\underline{\underline{H}}^T(k+1) + \right.$$

$$\left. \exp\left(-(t_{k+1}-t_k)/\tau\right)\underline{\underline{R}}(k+1)\right]^{-1} \qquad (6.12)$$

and the recursion for P at an observation

$$\underline{\underline{P}}(k+1/k+1) = \exp(-(t_{k+1}-t_k)/\tau)\left[\underline{\underline{I}} - \underline{\underline{K}}_e(k+1)\underline{\underline{H}}(k+1)\right]\underline{\underline{P}}(k+1/k) \tag{6.13}$$

It should be noted that, the smaller τ is , the faster old observations are "forgotten" .

With measurement noise covariance, data rate, and an assumed modeling error, one can find an optimum τ for minimizing the mean square error .

6. 3. Finite Memory Filter

The finite memory filter assumes a noise-free dynamics ;

$$\underline{x}(k+1) = \underline{\underline{\Phi}}(k+1,k)\underline{x}(k) \tag{6.14}$$

where

$$\underline{x}(0) \sim \mathcal{N}(\hat{\underline{x}}_0, \underline{\underline{P}}_0) \quad , \quad \underline{\underline{P}}_0 > 0$$

with observations

$$\underline{y}(k) = \underline{\underline{H}}(k)\underline{x}(k) + \underline{v}(k) \tag{6.15}$$

where

$$\underline{v}(k) \sim \mathcal{N}(\underline{0}, \underline{\underline{R}}(k)) \quad , \quad \underline{\underline{R}}(k) > 0$$

If the memory of the filter is limited sufficiently, the above set of equations become an adequate approximation to the real system over the limited time intervals .During this time intervals the system is assumed to be noise-free and this assumption is related to the selection of time interval in that the variation of unknown parameter over this interval is trivial .

Let the measurement sequence be denoted by ,

$$\{ y_1 , y_2 , \cdots , y_m ; y_{m+1} , \cdots , y_k \}$$

and

$$N = k - m$$

where N is the total number of measurements desired in the finite memory filter. Let the finite memory filter state estimate and covariance be denoted by $\hat{x}(k/m,k)$ and $P(k/m,k)$ respectively. They can be computed by the following equations :

$$\hat{\underline{x}}(k/m,k) = \underline{\underline{P}}(k/m,k)\left[\underline{\underline{P}}^{-1}(k/k)\hat{\underline{x}}(k/k) - \underline{\underline{P}}^{-1}(k/m)\hat{x}(k/m)\right] \quad (6.16)$$

$$\underline{\underline{P}}^{-1}(k/m,k) = \underline{\underline{P}}^{-1}(k/k) - \underline{\underline{P}}^{-1}(k/m) \quad (6.17)$$

where

$\hat{x}(k/k)$ is the state estimate at time k based upon all data up to and including y_k , and $\hat{x}(k/m)$ is the state estimate at time k based upon all data up to and including y_m .

A possible interpretation of the above equations could be to say that the finite memory estimate is obtained by subtracting an estimate based upon all data prior to the time window from the estimate based on all data .

It should be noted that the finite memory filter requires a batch of N measurement vectors be stored for updating $\hat{x}(k/m)$.

6.4. The $\alpha-\beta-\gamma$ Tracker

In some cases , because of computational constraints, it may be impractical to compute the filter gains in real time . The problem we have been studying is an example of such a case . Under such cases, one must use either a set of precalculated filter gains or a constant gain filter . One commonly used method of constant gain filter implementation is to compute and tabulate the steady-state filter gains as a function of target noise model.

Another popular constant gain filter is the $\alpha-\beta-\gamma$ filter. The main difference between the $\alpha-\beta-\gamma$ filter and the constant gain filter is that the former assumes a complete independence of the three spatial coordinates in the filter update equation.

We noted that the state space and measurement space may be related through a nonlinear function as suggested by the equation (3.5) .In using the $\alpha-\beta-\gamma$ filter, this assumption is not allowed .

Let (r,b,e) denote the radar range, bearing and elevation, respectively .The state vector becomes ;

$$\underline{x} = \left[r \quad \dot{r} \quad \ddot{r} \mid b \quad \dot{b} \quad \ddot{b} \mid e \quad \dot{e} \quad \ddot{e} \right]^T \tag{6.18}$$

Let $\underline{r}$ denote $[r \quad \dot{r} \quad \ddot{r}]^T$ then the filter update equation becomes:

$$\hat{\underline{r}}(k+1/k+1) = \hat{\underline{r}}(k+1/k) + \underline{\underline{K}}(k+1)\left[\underline{z}(k+1) - \hat{\underline{z}}(k+1/k)\right] \tag{6.19}$$

where, obviously, $z(k+1)$ is the range measurement at time $(k+1)$. The gain matrix $K(k+1)$ for the $\alpha-\beta-\gamma$ filter is ;

$$\underline{K}(k+1) = \left[\alpha, \frac{\beta}{t_k - t_{k-1}}, \frac{2\gamma}{(t_{k+1} - t_k)^2} \right] \tag{6.20}$$

The update equations for bearing and elevation can be obtained accordingly .It should be noted that α, β and γ are filter gain components for position, velocity and acceleration, respectively. The structure of the gain equation allows these parameters to be free of time interval between measurements at $t_{k+1} - t_k$.

When a Kalman filter is used, the gain matrix K_k has the dimension (9×3) while the gain matrix utilizing the channel independence assumption reduces to three (3×1) vectors .There is a wide range of methods for choosing the values of α, β and γ .One method is to use the steady-state Kalman gain in the above chosen coordinate .Another method used in practice is to conduct extensive Monte-Carlo studies to define α, β and γ over a variety of cases for a given application .

6.5. A Comparison of Filtering Methods

In this chapter we have discussed the filter designs considering such factors as performance sensitivity to modeling errors and computational efficiency of a particular choice.

Admittedly, none of the four approaches we have presented is likely to be accepted as an alternative to the one selected for implementation in this study. The extended Kalman filter with an appropriately chosen process noise covariance could be a good choice if our problem had not required excessively fast data processing to meet the demands of real-time applications. The $\alpha-\beta-\gamma$ filter offers considerable simplicity of computation at the cost of degraded performance. The finite memory and fading memory filters are only of secondary interest since they do not seem to provide any advantages over the extended Kalman filter while computational requirements are comparable or, depending on application, even more severe.

This brief outline of the most popular approaches to processing of noisy measurements associated with a given dynamical system forces us to make use of advantageous properties of each one of them and synthesize the one to serve best for our specific filter application.

Since a separate chapter has been devoted for justifying our particular approach, we will omit the discussion related to it and prefer to present the simulation studies in the next section.

VII. SIMULATION STUDIES

The Monte-Carlo simulation program described in appendix A performs the realization of dual-bandwidth adaptive filter in conjunction with the accompanying subroutines . The program has been made in a generalized fashion in that it offers several options to the user . Even though these options are explained in detailed comments throughout the program, it has been found helpful to give a brief outline of these options once before the program listing . First of all, they are given in the form of "software switches" to the user, namely, by specifying these input flags, one can modify the course of events in line with his particular purpose . These flags are ;

i. ICHN

This is the flag for channel selection . In chapter III, it has been emphasized that the partitioning of system model would result in a significant mathematical simplicity of implementatio This approximation is now utilized by specifying such that ,

if ICHN = 1 , the program runs for x-channel
if ICHN = 0 , the program runs for y-channel
if ICHN = 1 then the program runs for z-channel

ii. IPHI

The specification of this flag enables the user to choose the following options :

if IPHI = 1 ,	the state transition matrix should be in the input file
if IPHI = 0 ,	the system matrix F (in continuous-time) should be specified. The program then evaluates the discrete-time equivalent of F , i. e. corresponding STM with desired set of parameters in the series expansion.
if IPHI = 1	STM is evaluated directly by the program

iii. ITRK

This is the flag for the choice of assigning a reference trajectory to the program.

if ITRK = 1, the reference trajectory is to be obtained from the noise-free dynamics, namely,

$$\underline{x}(k) = \underline{\underline{\Phi}}(k, k-1)\underline{x}(k-1)$$

if ITRK = 0, the user should specify a reference trajectory in any form

if ITRK = 1, the simulated aircraft trajectory is generated by call of subroutine TRKGEN which will be presented in appendix B in a detailed discussion.

7.1. Test Trajectories.

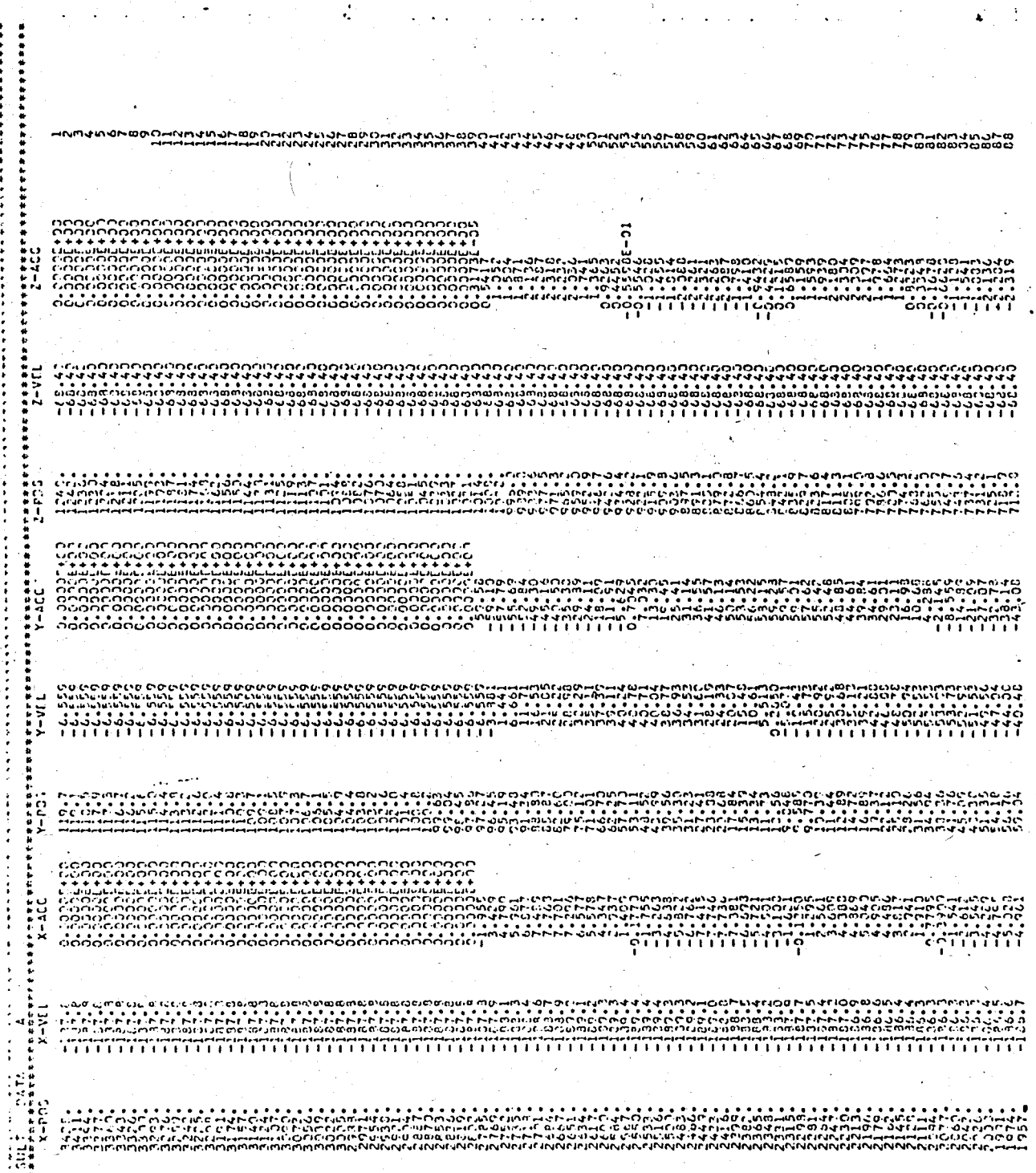
All trajectories used for testing filter performance are the products of track generating subroutine since these trajectories are variable in range, maneuver duration and maneuver amplitude. The utilization of a trajectory which is a result of noise-free dynamics is not found to be a suitable choice since this would mean that the target is modeled perfectly, which is, of course, not realistic. In addition, it was found unnecessary to specify particular trajectories since the present results proved to be sufficient for testing the filter performance.

An example of test trajectories involved in the simulation can be given as ;

maneuver commencing range	:	3000
maneuver duration	:	2000
maneuver amplitude	:	75.5

The following table lists the data generated by the track generating program for this example. The track generating program was borrowed from J. Michael Parr [13] and involved in our simulation program in the form of a subroutine after certain modifications have been made.

Table 7.1.1
An example of test trajectories



1940.	-186.[illegible]	-4.583	69.58	-36.5[illegible]	-47.5[illegible]	706.6	-68.40	-2.234	89
1923.	-187.[illegible]	-3.700	67.56	-32.5[illegible]	-51.34	700.5	-68.40	-2.032	90
1907.	-187.[illegible]	-2.937	63.75	[illegible]	-54.52	694.3	-68.40	-1.712	91
1890.	-187.[illegible]	-1.730	64.51	-22.6[illegible]	-56.9[illegible]	688.1	-68.40	-1.312	92
1873.	-187.[illegible]	-0.3600	64.85	-17.40	-58.64	682.0	-68.40	-0.8312	93
1856.	-187.6	1.119	64.73	-12.70	-59.47	675.8	-68.40	-0.3030	94
1839.	-187.8	-1.256	64.26	-7.116	59.5[illegible]	669.7	-68.40	0.2426	95
1822.	-187.7	0.2000	64.05	-11.61	58.77	663.5	-68.40	0.7743	96
1805.	-187.5	1.592	64.30	-16.95	57.20	657.4	-68.40	1.267	97
1788.	-187.3	2.815	65.10	-22.05	54.83	651.2	-68.40	1.679	98
1771.	-187.[illegible]	3.807	66.24	-27.00	51.73	645.1	-68.40	2.002	99
1754.	-187.0	4.522	67.77	-31.70	47.9[illegible]	638.9	-68.40	2.217	100
1737.	-186.9	4.934	69.66	-36.07	43.59	632.7	-68.40	2.315	101
1720.	-186.7	5.036	71.92	-40.06	38.71	626.6	-68.40	2.294	102
1703.	-186.6	4.835	74.47	-43.63	33.40	620.4	-68.40	2.158	103
1686.	-186.5	4.356	77.27	-46.93	27.77	614.3	-68.40	1.916	104
1669.	-186.4	3.630	80.30	-49.31	21.76	608.1	-68.40	1.583	105
1652.	-186.4	2.723	83.49	-51.34	15.59	602.0	-68.40	1.177	106
1635.	-186.3	1.665	86.81	-52.00	9.265	595.8	-68.40	0.7164	107
1618.	-186.3	0.5200	90.21	-53.67	2.860	589.6	-68.40	0.2236	108
1601.	-186.3	-0.6503	93.63	-53.94	-3.574	583.5	-68.40	-0.2792	109
1584.	-186.3	-1.788	97.01	-53.61	-9.976	577.3	-68.40	-0.7675	110
1567.	-186.3	-2.833	100.3	-52.67	-16.28	571.2	-68.40	-1.225	111
1550.	-186.4	-3.720	103.5	-51.14	-22.44	565.0	-68.40	-1.624	112
1533.	-185.4	-4.423	106.5	-49.05	-28.37	558.9	-68.40	-1.948	113
1516.	-186.5	-4.872	109.3	-46.41	-34.01	552.7	-68.40	-2.178	114
1499.	-186.6	-5.040	111.8	-43.26	-39.28	546.5	-68.40	-2.302	115
1482.	-186.7	-4.904	114.0	-39.64	-44.11	540.4	-68.40	-2.310	116
1465.	-186.9	-4.457	115.9	-35.60	-48.41	534.2	-68.40	-2.199	117
1448.	-187.0	-3.707	117.4	-31.15	-52.11	528.1	-68.40	-1.971	118
1431.	-187.2	-2.687	118.5	-26.46	-55.13	521.9	-68.40	-1.636	119
1414.	-187.3	-1.445	119.1	-21.48	-57.41	515.8	-68.40	-1.211	120
1397.	-187.5	-0.3010E-01	119.4	-16.31	-58.99	509.6	-68.40	-0.7169	121
1380.	-187.7	1.453	119.2	-11.02	-59.55	503.5	-68.40	-0.1921	122
1363.	-187.9	2.946	118.5	-5.666	-59.35	497.3	-68.40	0.3631	123
1346.	-188.0	4.353	117.4	-0.3250	-58.32	491.1	-68.40	0.8875	124
1329.	-188.2	5.596	115.8	4.936	-55.48	485.0	-68.40	1.361	125
1312.	-188.4	6.603	113.8	10.05	-53.87	478.8	-68.40	1.759	126
1296.	-188.5	7.321	111.4	14.95	-50.55	472.7	-68.40	2.060	127
1279.	-188.7	7.713	108.6	19.58	-46.61	466.5	-68.40	2.249	128
1262.	-188.9	7.768	105.4	23.87	-42.11	460.4	-68.40	2.321	129
1245.	-189.0	7.468	101.9	27.78	-37.13	454.2	-68.40	2.273	130
1229.	-189.1	6.857	98.16	31.24	-31.76	448.0	-68.40	2.113	131
1212.	-189.2	5.933	94.12	34.23	-26.06	441.9	-68.40	1.849	132
1195.	-189.2	4.780	89.87	36.69	-20.12	435.7	-68.40	1.499	133
1178.	-189.3	3.424	85.46	38.60	-13.99	429.6	-68.40	1.078	134
1161.	-189.3	1.920	80.94	39.93	-7.733	423.4	-68.40	0.6009	135
1144.	-189.4	0.3530	76.35	40.67	-1.409	417.3	-68.40	0.1119	136
1128.	-189.4	-1.234	71.76	40.80	4.927	411.1	-68.40	-0.3901	137
1111.	-189.4	-2.772	67.20	40.33	11.2[illegible]	405.0	-68.40	-0.8745	138
1094.	-189.3	-4.195	62.74	39.36	17.41	398.8	-68.40	-1.319	139
1077.	-189.3	-5.453	58.41	37.61	23.45	392.6	-68.40	-1.703	140
1061.	-189.2	-6.475	54.27	[illegible]	29.[illegible]	386.5	-68.40	-2.008	141
1044.	-189.1	-7.231	50.37	32.38	34.79	380.3	-68.40	-2.215	142
1027.	-189.0	-7.673	46.74	29.38	39.95	374.2	-68.40	-2.314	143
1010.	-188.9	-7.777	43.42	25.66	44.67	368.0	-68.40	-2.296	144
993.4	-188.7	-7.537	40.46	21.53	48.67	361.9	-68.40	-2.158	145
976.6	-188.6	-6.961	37.87	17.05	52.48	355.7	-68.40	-1.906	146
959.7	-188.4	-6.076	35.69	12.26	55.41	349.5	-68.40	-1.549	147
942.8	-188.3	-4.931	33.93	7.232	57.60	343.4	-68.40	-1.106	148
926.0	-188.1	-3.531	32.45	[illegible]	59.00	337.2	-68.40	-0.6006	149
909.1	-187.9	-2.12[illegible]	31.75	[illegible]	59.57	331.1	-68.40	-0.6078E-01	150
892.2	-187.6	0.0000E+00	31.16	-6.559	0.0000E+00	324.9	-68.40	0.0000E+00	151
875.3	-187.9	0.0000E+00	30.56	-6.559	0.0000E+00	318.8	-68.40	0.0000E+00	152
858.4	-187.8	0.0000E+00	29.97	-6.559	0.0000E+00	312.6	-68.40	0.0000E+00	153
841.5	-187.8	0.0000E+00	29.38	-6.559	0.0000E+00	306.5	-68.40	0.0000E+00	154
824.6	-187.8	0.0000E+00	28.79	-6.559	0.0000E+00	300.3	-68.40	0.0000E+00	155
807.6	-187.8	0.0000E+00	28.20	-6.559	0.0000E+00	294.1	-68.40	0.0000E+00	156
790.7	-187.8	0.0000E+00	27.61	-6.559	0.0000E+00	288.0	-68.40	0.0000E+00	157
773.8	-187.8	0.0000E+00	27.03	-6.559	0.0000E+00	281.8	-68.40	0.0000E+00	158
756.9	-187.8	0.0000E+00	26.43	-6.559	0.0000E+00	275.7	-68.40	0.0000E+00	159
740.0	-187.8	0.0000E+00	25.84	-6.559	0.0000E+00	269.5	-68.40	0.0000E+00	160
723.1	-187.8	0.0000E+00	25.25	-6.559	0.0000E+00	263.4	-68.40	0.0000E+00	161
706.2	-187.8	0.0000E+00	24.66	-6.559	0.0000E+00	257.2	-68.40	0.0000E+00	162
689.3	-187.8	0.0000E+00	24.07	-6.559	0.0000E+00	251.0	-68.40	0.0000E+00	163
672.4	-187.8	0.0000E+00	23.48	-6.559	0.0000E+00	244.9	-68.40	0.0000E+00	164
655.5	-187.8	0.0000E+00	22.89	-6.559	0.0000E+00	238.7	-68.40	0.0000E+00	165
638.6	-187.6	0.0000E+00	22.30	-6.559	0.0000E+00	232.6	-68.40	0.0000E+00	166
621.7	-187.8	0.0000E+00	21.71	-6.559	0.0000E+00	226.4	-68.40	0.0000E+00	167
604.8	-187.8	0.0000E+00	21.12	-6.559	0.0000E+00	220.3	-68.40	0.0000E+00	168
587.9	-187.8	0.0000E+00	20.53	-6.559	0.0000E+00	214.1	-68.40	0.0000E+00	169
571.0	-187.8	0.0000E+00	19.94	-6.559	0.0000E+00	207.9	-68.40	0.0000E+00	170
554.1	-187.8	0.0000E+00	19.35	-6.559	0.0000E+00	201.8	-68.40	0.0000E+00	171
537.2	-187.8	0.0000E+00	18.76	-6.559	0.0000E+00	195.6	-68.40	0.0000E+00	172
520.3	-187.8	0.0000E+00	18.17	-6.559	0.0000E+00	189.5	-68.40	0.0000E+00	173
503.4	-187.8	0.0000E+00	17.58	-6.559	0.0000E+00	183.3	-68.40	0.0000E+00	174
486.5	-187.8	0.0000E+00	16.99	-6.559	0.0000E+00	177.2	-68.40	0.0000E+00	175
469.6	-187.8	0.0000E+00	16.40	-6.559	0.0000E+00	171.0	-68.40	0.0000E+00	176
452.7	-187.8	0.0000E+00	15.81	-6.559	0.0000E+00	164.9	-68.40	0.0000E+00	177
435.8	-187.6	0.0000E+00	15.22	-6.559	0.0000E+00	158.7	-68.40	0.0000E+00	178

418.8	-187.8	0.0000E+00	14.63	-6.559	0.0000E+00	152.5	-68.40	0.0000E+00	179
401.9	-187.8	0.0000E+00	14.04	-6.559	0.0000E+00	146.4	-68.40	0.0000E+00	180
385.0	-187.8	0.0000E+00	13.45	-6.559	0.0000E+00	140.2	-68.40	0.0000E+00	181
368.1	-187.8	0.0000E+00	12.86	-6.559	0.0000E+00	134.1	-68.40	0.0000E+00	182
351.2	-187.8	0.0000E+00	12.27	-6.559	0.0000E+00	127.9	-68.40	0.0000E+00	183
334.3	-187.8	0.0000E+00	11.67	-6.559	0.0000E+00	121.8	-68.40	0.0000E+00	184
317.4	-187.8	0.0000E+00	11.08	-6.559	0.0000E+00	115.6	-68.40	0.0000E+00	185
300.5	-187.8	0.0000E+00	10.49	-6.559	0.0000E+00	109.4	-68.40	0.0000E+00	186
283.6	-187.8	0.0000E+00	9.904	-6.559	0.0000E+00	103.3	-68.40	0.0000E+00	187
266.7	-187.8	0.0000E+00	9.314	-6.559	0.0000E+00	97.13	-68.40	0.0000E+00	188
249.8	-187.8	0.0000E+00	8.723	-6.559	0.0000E+00	90.98	-68.40	0.0000E+00	189
232.9	-187.8	0.0000E+00	8.133	-6.559	0.0000E+00	84.82	-68.40	0.0000E+00	190
216.0	-187.8	0.0000E+00	7.543	-6.559	0.0000E+00	78.66	-68.40	0.0000E+00	191
199.1	-187.8	0.0000E+00	6.953	-6.559	0.0000E+00	72.51	-68.40	0.0000E+00	192
182.2	-187.8	0.0000E+00	6.362	-6.559	0.0000E+00	66.35	-68.40	0.0000E+00	193
165.3	-187.8	0.0000E+00	5.772	-6.559	0.0000E+00	60.20	-68.40	0.0000E+00	194
148.4	-187.8	0.0000E+00	5.182	-6.559	0.0000E+00	54.04	-68.40	0.0000E+00	195
131.5	-187.8	0.0000E+00	4.591	-6.559	0.0000E+00	47.88	-68.40	0.0000E+00	196
114.6	-187.8	0.0000E+00	4.001	-6.559	0.0000E+00	41.73	-68.40	0.0000E+00	197
97.67	-187.8	0.0000E+00	3.411	-6.559	0.0000E+00	35.57	-68.40	0.0000E+00	198
80.76	-187.8	0.0000E+00	2.820	-6.559	0.0000E+00	29.41	-68.40	0.0000E+00	199
63.86	-187.8	0.0000E+00	2.230	-6.559	0.0000E+00	23.26	-68.40	0.0000E+00	200

1

7. 2. Simulation results

This section presents the results of Monte-Carlo simulation program. It should be noted that the continuous time state equations are discretized with specified parameters and then filtering is performed in x, y, and z-axis in a sequential manner . The tracking performance in each coordinate axis is plotted afterwards.

RESULT DATA A

MATRIX F :

0.1000+01 0.1000+01 0.0000+00
0.0000+00 0.1000+01 0.0000+00
0.0000+00 0.0000+00 -.5000-01

DISCRT-TIME EQUIVALENT OF FA;

MATRIX PHIA :

0.1000+01 0.2500+00 0.3110-01
0.2977-07 0.1000+01 0.2487+00
-.4814-06 0.0000+00 0.9880+00

MATRIX F :

0.0000+00 0.1000+01 0.0000+00
0.0000+00 0.0000+00 0.1000+01
0.5000+00 0.0000+00 -.1000+00

DISCRT-TIME EQUIVALENT OF FB;

MATRIX PHIB :

0.1000+01 0.2500+00 0.3100-01
0.3915-04 0.1000+01 0.2470+00
-.3915-05 0.0000+00 0.9750+00

VECTOR H;

0.1000+01
0.0000+00
0.0000+00

INITIAL EST.ERR. COVARIANCE MATRIX P(0/-1)

0.1000+06 0.0000+00 0.0000+00
0.0000+00 0.1000+04 0.0000+00
0.0000+00 0.0000+00 0.1000+04

PROCESS NOISE COVARIANCE MATRIX QA;

0.3720-06 0.4540-03 0.2630-07
0.4540-03 0.5540+01 0.4350-06
0.2630-07 0.4350-06 0.1370-04

PROCESS NOISE COVARIANCE MATRIX QB;

0.1510-06 0.1830-07 0.1440-08
0.1830-07 0.2230+01 0.1720-05
0.1440-06 0.1720-05 0.5360-04

I	X-POS	X-VEL	X-ACC	Y-POS	Y-VEL	Y-ACC	Z-POS	Z-VEL	Z-ACC	
	3426.	-187.8	0.0000E+00	119.7	-6.559	0.0000E+00	1248.	-68.40	0.0000E+00	1
	3411.	-187.8	0.0000E+00	119.1	-6.559	0.0000E+00	1242.	-68.40	0.0000E+00	2
	3396.	-187.8	0.0000E+00	118.5	-6.559	0.0000E+00	1236.	-68.40	0.0000E+00	3
	3381.	-187.8	0.0000E+00	117.9	-6.559	0.0000E+00	1230.	-68.40	0.0000E+00	4
	3366.	-187.8	0.0000E+00	117.3	-6.559	0.0000E+00	1224.	-68.40	0.0000E+00	5
	3351.	-187.8	0.0000E+00	116.7	-6.559	0.0000E+00	1218.	-68.40	0.0000E+00	6
	3336.	-187.8	0.0000E+00	116.1	-6.559	0.0000E+00	1211.	-68.40	0.0000E+00	7
	3321.	-187.8	0.0000E+00	115.5	-6.559	0.0000E+00	1205.	-68.40	0.0000E+00	8
	3306.	-187.8	0.0000E+00	114.9	-6.559	0.0000E+00	1199.	-68.40	0.0000E+00	9
	3291.	-187.8	0.0000E+00	114.3	-6.559	0.0000E+00	1193.	-68.40	0.0000E+00	10
	3276.	-187.8	0.0000E+00	113.7	-6.559	0.0000E+00	1187.	-68.40	0.0000E+00	11
	3259.	-187.8	0.0000E+00	113.0	-6.559	0.0000E+00	1181.	-68.40	0.0000E+00	12
	3243.	-187.8	0.0000E+00	112.4	-6.559	0.0000E+00	1174.	-68.40	0.0000E+00	13
	3228.	-187.8	0.0000E+00	111.8	-6.559	0.0000E+00	1168.	-68.40	0.0000E+00	14
	3213.	-187.8	0.0000E+00	111.2	-6.559	0.0000E+00	1162.	-68.40	0.0000E+00	15
	3198.	-187.8	0.0000E+00	110.6	-6.559	0.0000E+00	1156.	-68.40	0.0000E+00	16
	3183.	-187.8	0.0000E+00	110.0	-6.559	0.0000E+00	1150.	-68.40	0.0000E+00	17
	3168.	-187.8	0.0000E+00	109.4	-6.559	0.0000E+00	1144.	-68.40	0.0000E+00	18
	3153.	-187.8	0.0000E+00	108.8	-6.559	0.0000E+00	1138.	-68.40	0.0000E+00	19
	3138.	-187.8	0.0000E+00	108.2	-6.559	0.0000E+00	1132.	-68.40	0.0000E+00	20
	3122.	-187.8	0.0000E+00	107.6	-6.559	0.0000E+00	1125.	-68.40	0.0000E+00	21
	3107.	-187.8	0.0000E+00	107.0	-6.559	0.0000E+00	1119.	-68.40	0.0000E+00	22
	3092.	-187.8	0.0000E+00	106.4	-6.559	0.0000E+00	1113.	-68.40	0.0000E+00	23
	3077.	-187.8	0.0000E+00	105.8	-6.559	0.0000E+00	1107.	-68.40	0.0000E+00	24
	3062.	-187.8	0.0000E+00	105.2	-6.559	0.0000E+00	1101.	-68.40	0.0000E+00	25
	3047.	-187.8	0.0000E+00	104.6	-6.559	0.0000E+00	1095.	-68.40	0.0000E+00	26
	3032.	-187.8	0.0000E+00	104.0	-6.559	0.0000E+00	1089.	-68.40	0.0000E+00	27
	3017.	-187.8	0.0000E+00	103.4	-6.559	0.0000E+00	1083.	-68.40	0.0000E+00	28
	3002.	-187.8	0.0000E+00	102.8	-6.559	0.0000E+00	1077.	-68.40	0.0000E+00	29
	2987.	-187.8	0.0000E+00	102.2	-6.559	0.0000E+00	1071.	-68.40	0.0000E+00	30
	2971.	-187.8	0.0000E+00	101.6	-6.559	0.0000E+00	1064.	-68.40	0.0000E+00	31
	2956.	-187.8	0.0000E+00	101.0	-6.559	0.0000E+00	1058.	-68.40	0.0000E+00	32
	2941.	-187.8	0.0000E+00	100.4	-6.559	0.0000E+00	1052.	-68.40	0.0000E+00	33
	2926.	-187.8	0.0000E+00	99.84	-6.559	0.0000E+00	1046.	-68.40	0.0000E+00	34
	2911.	-187.8	0.0000E+00	99.24	-6.559	0.0000E+00	1040.	-68.40	0.0000E+00	35
	2896.	-187.8	0.0000E+00	98.64	-6.559	0.0000E+00	1034.	-68.40	0.0000E+00	36
	2880.	-187.8	0.0000E+00	98.04	-6.559	0.0000E+00	1027.	-68.40	0.0000E+00	37
	2865.	-187.8	0.0000E+00	97.44	-6.559	0.0000E+00	1021.	-68.40	0.0000E+00	38

2735.	-137.8	-1.955	97.27	-6.559	59.58	1014.	-68.40	0.31730E-05	39
2769.	-138.0	3.429	96.46	-3.831	59.10	1005.	-68.40	0.5417	40
2752.	-138.1	4.769	94.19	1.441	57.79	995.2	-68.40	1.052	41
2735.	-138.2	5.964	93.48	6.461	56.69	988.3	-68.40	1.504	42
2718.	-138.4	6.877	91.34	11.71	52.84	980.3	-68.40	1.871	43
2701.	-138.6	7.499	88.00	16.53	49.30	973.6	-68.40	2.136	44
2684.	-138.7	7.768	85.87	21.05	45.16	967.5	-68.40	2.287	45
2667.	-138.9	7.701	83.60	25.22	40.53	961.3	-68.40	2.318	46
2651.	-139.0	7.296	79.20	28.98	35.38	955.2	-68.40	2.232	47
2634.	-139.1	6.577	75.12	32.29	29.86	949.0	-68.40	2.036	48
2617.	-139.2	5.579	71.01	35.11	24.11	942.9	-68.40	1.741	49
2600.	-139.3	4.347	65.70	37.39	18.09	936.7	-68.40	1.365	50
2583.	-139.3	2.937	62.25	39.11	11.91	930.6	-68.40	0.9263	51
2567.	-139.4	1.409	57.70	40.24	5.622	924.4	-68.40	0.4652	52
2550.	-139.4	-0.1770	53.11	40.78	0.7045	918.2	-68.40	-0.5598E-01	53
2533.	-139.4	-1.755	48.52	40.71	-4.232	912.1	-68.40	-0.5546	54
2516.	-139.4	-3.263	43.99	40.04	-13.36	905.9	-68.40	-1.028	55
2500.	-139.3	-4.638	39.56	38.77	-19.45	899.8	-68.40	-1.455	56
2483.	-139.2	-5.822	35.30	36.93	-25.41	893.6	-68.40	-1.814	57
2466.	-139.2	-6.765	31.23	34.53	-31.14	887.5	-68.40	-2.080	58
2449.	-139.1	-7.415	27.41	31.60	-36.55	881.3	-68.40	-2.261	59
2433.	-139.0	-7.746	23.88	28.19	-41.57	875.1	-68.40	-2.341	60
2416.	-138.9	-7.736	20.69	24.33	-46.13	869.0	-68.40	-2.369	61
2399.	-138.7	-7.381	17.84	20.07	-50.14	862.8	-68.40	-2.087	62
2382.	-138.5	-6.698	15.39	15.46	-53.53	856.7	-68.40	-1.758	63
2365.	-138.4	-5.721	13.34	10.61	-56.22	850.5	-68.40	-1.410	64
2349.	-138.2	-4.501	11.73	5.513	-58.15	844.4	-68.40	-1.043	65
2332.	-138.0	-3.108	10.56	0.2650	-59.28	838.2	-68.40	-0.6432	66
2315.	-137.9	-1.621	9.848	-5.071	-59.57	832.1	-68.40	-0.2229	67
2298.	-137.7	-0.1250	9.585	-10.43	-59.01	825.9	-68.40	0.1715	68
2281.	-137.5	1.395	9.770	-15.73	-57.62	819.8	-68.40	0.6589	69
2264.	-137.4	2.961	10.39	-20.92	-55.42	813.6	-68.40	1.119	70
2247.	-137.2	4.309	11.44	-25.95	-52.48	807.5	-68.40	1.559	71
2230.	-137.0	4.388	12.89	-30.68	-48.85	801.3	-68.40	1.939	72
2213.	-136.9	4.867	14.72	-35.13	-44.61	795.2	-68.40	2.180	73
2196.	-136.8	5.040	16.89	-39.21	-39.84	789.0	-68.40	2.334	74
2179.	-136.6	4.905	19.37	-42.88	-34.61	782.9	-68.40	2.309	75
2162.	-136.5	4.486	22.12	-46.08	-29.01	776.7	-68.40	2.197	76
2145.	-136.4	3.817	25.10	-48.78	-23.11	770.5	-68.40	1.664	77
2128.	-136.4	2.941	28.26	-50.94	-16.96	764.4	-68.40	1.273	78
2111.	-136.3	1.910	31.57	-52.53	-10.69	758.2	-68.40	0.8223	79
2094.	-136.3	0.7795	34.94	-53.53	-4.288	752.1	-68.40	0.3340	80
2077.	-136.3	-0.3909	38.26	-53.93	2.145	745.9	-68.40	-0.1678	81
2060.	-136.3	-1.541	41.36	-53.73	8.559	739.7	-68.40	-0.6628	82
2043.	-136.3	-2.612	44.09	-52.93	14.89	733.6	-68.40	-1.129	83
2026.	-136.3	-3.568	46.30	-51.52	21.09	727.4	-68.40	-1.541	84
2009.	-136.4	-4.389	51.35	-49.56	27.07	721.2	-68.40	-1.903	85
1991.	-136.5	-5.028	54.18	-47.04	32.78	715.1	-68.40	-2.136	86
1974.	-136.6	-5.561	56.76	-44.00	38.14	708.9	-68.40	-2.284	87
1957.	-136.7	-4.383	59.04	-40.48	43.08	702.7	-68.40	-2.319	88
1940.	-136.8	-3.900	60.98	-36.53	47.50	696.6	-68.40	-2.234	89
1923.	-137.0	-2.937	62.56	-32.20	51.34	690.4	-68.40	-2.032	90
1907.	-137.1	-1.738	63.75	-27.54	54.52	684.3	-68.40	-1.719	91
1890.	-137.3	-0.3603	64.51	-22.61	56.97	678.1	-68.40	-1.312	92
1873.	-137.5	1.119	64.85	-17.46	58.64	672.0	-68.40	-0.8314	93
1856.	-137.6	-1.386	64.73	-12.20	59.47	665.8	-68.40	-0.3030	94
1839.	-137.7	0.2000	64.26	-7.016	59.52	659.7	-68.40	0.2426	95
1822.	-137.7	1.592	64.09	-11.61	58.77	653.5	-68.40	0.7743	96
1805.	-137.5	2.815	64.38	-16.90	57.20	647.4	-68.40	1.292	97
1789.	-137.3	3.807	65.10	-22.05	54.88	641.2	-68.40	1.679	98
1772.	-137.2	4.522	66.24	-27.00	51.73	635.1	-68.40	2.002	99
1755.	-137.0	4.934	67.77	-31.70	47.96	628.9	-68.40	2.217	100
1738.	-136.9	5.036	69.68	-36.07	43.59	622.7	-68.40	2.315	101
1721.	-136.7	4.835	71.92	-40.06	38.71	616.6	-68.40	2.294	102
1705.	-136.6	4.353	74.47	-43.63	33.40	610.4	-68.40	2.158	103
1688.	-136.5	3.638	77.27	-46.73	27.72	604.3	-68.40	1.916	104
1671.	-136.4	2.723	80.30	-49.31	21.74	598.1	-68.40	1.583	105
1654.	-136.4	1.665	83.49	-51.34	15.59	592.0	-68.40	1.177	106
1638.	-136.3	0.5208	86.81	-52.80	9.268	585.8	-68.40	0.7164	107
1621.	-136.3	-0.6502	90.21	-53.67	2.860	579.6	-68.40	0.2536	108
1604.	-136.3	-1.788	93.63	-53.94	-3.574	573.5	-68.40	-0.2092	109
1587.	-136.3	-2.833	97.01	-53.61	-9.976	567.3	-68.40	-0.7695	110
1571.	-136.3	-3.749	100.3	-52.67	-16.26	561.2	-68.40	-1.325	111
1554.	-136.4	-4.473	103.5	-51.14	-22.44	555.0	-68.40	-1.824	112
1537.	-136.4	-4.975	106.5	-49.05	-28.37	548.9	-68.40	-2.243	113
1520.	-136.4	-5.242	109.3	-46.41	-34.01	542.7	-68.40	-2.578	114
1503.	-136.5	-5.274	111.8	-43.26	-39.28	536.5	-68.40	-2.830	115
1486.	-136.6	-5.070	114.0	-39.64	-44.11	530.4	-68.40	-2.999	116
1469.	-136.7	-4.634	115.9	-35.60	-48.41	524.2	-68.40	-3.085	117
1453.	-136.9	-4.457	117.4	-31.19	-52.11	518.1	-68.40	-3.090	118
1436.	-137.0	-3.709	118.5	-26.46	-55.13	511.9	-68.40	-3.014	119
1419.	-137.2	-2.689	119.1	-21.48	-57.41	505.8	-68.40	-2.858	120
1414.	-137.3	-1.443	119.4	-16.31	-58.89	499.6	-68.40	-2.716	121
1397.	-137.5	-0.3811E-01	119.2	-11.02	-59.55	493.5	-68.40	-2.571	122
1380.	-137.7	1.427	118.5	-5.666	-59.35	487.3	-68.40	-2.423	123
1363.	-137.9	2.744	117.4	-0.3250	-58.32	481.1	-68.40	-2.271	124
1346.	-138.0	3.953	115.8	4.936	-56.49	475.0	-68.40	-2.116	125
1329.	-138.2	5.026	113.8	10.05	-53.87	468.8	-68.40	-1.959	126
1312.	-138.4	6.035	111.4	14.95	-50.55	462.7	-68.40	-1.800	127
1296.	-138.5	6.951	108.6	19.58	-46.61	456.5	-68.40	-1.640	128
1279.	-138.7	7.713							

1262.	-192.8	7.765	105.4	23.87	-42.11	460.4	-68.40	2.321	129
1245.	-190.0	7.460	101.9	27.76	-37.13	454.2	-68.40	2.273	130
1229.	-189.1	6.833	98.16	31.24	-31.75	448.0	-68.40	2.113	131
1212.	-187.3	5.931	[illegible]	[illegible]	-26.05	441.7	-68.40	1.843	132
1195.	-187.3	4.781	89.37	36.69	-20.12	435.5	-68.40	1.459	133
1178.	-187.3	3.454	85.46	38.60	-13.97	429.4	-68.40	1.039	134
1161.	-187.4	1.928	80.54	[illegible]	-7.723	423.4	-68.40	0.6039	135
1144.	-187.4	0.3533	75.35	40.67	-1.457	417.3	-68.40	0.1119	136
1128.	-187.3	-1.234	71.76	40.83	4.927	411.1	-68.40	-0.3701	137
1111.	-187.3	-2.772	67.20	40.33	11.22	405.0	-68.40	-0.8745	138
1094.	-187.1	-4.190	62.74	39.26	17.41	398.8	-68.40	-1.319	139
1077.	-187.0	-5.453	58.41	37.61	23.45	392.6	-68.40	-1.723	140
1061.	-187.7	-6.479	54.27	35.39	29.26	386.5	-68.40	-2.030	141
1044.	-187.6	-7.231	50.37	32.63	34.79	380.3	-68.40	-2.215	142
1027.	-187.4	-7.673	46.74	29.36	39.95	374.2	-68.40	-2.314	143
1010.	-187.3	-7.777	43.42	25.66	44.67	368.0	-68.40	-2.296	144
993.4	-187.0	-7.537	40.46	21.53	48.37	361.9	-68.40	-2.150	145
976.5	-187.8	-6.961	37.87	17.28	52.48	355.7	-68.40	-1.906	146
959.7	-187.8	-6.076	35.69	12.77	55.41	349.5	-68.40	-1.547	147
942.8	-187.8	-4.731	33.93	7.996	57.60	343.4	-68.40	-1.136	148
926.0	-187.8	-3.587	32.62	2.947	59.00	337.2	-68.40	-0.6036	149
909.1	-187.8	-2.122	31.75	-3.286	59.57	331.1	-68.40	-0.6370E-01	150
892.3	-187.8	0.0000E+00	31.16	-6.553	0.0000E+00	324.9	-68.40	0.0000E+00	151
875.3	-187.8	0.0000E+00	30.56	-6.559	0.0000E+00	318.8	-68.40	0.0000E+00	152
858.4	-187.8	0.0000E+00	29.97	-6.559	0.0000E+00	312.6	-68.40	0.0000E+00	153
841.5	-187.8	0.0000E+00	29.38	-6.559	0.0000E+00	306.5	-68.40	0.0000E+00	154
824.5	-187.8	0.0000E+00	28.79	-6.559	0.0000E+00	300.3	-68.40	0.0000E+00	155
807.6	-187.8	0.0000E+00	28.20	-6.559	0.0000E+00	294.1	-68.40	0.0000E+00	156
790.7	-187.8	0.0000E+00	27.61	-6.559	0.0000E+00	288.0	-68.40	0.0000E+00	157
773.8	-187.8	0.0000E+00	27.02	-6.559	0.0000E+00	281.8	-68.40	0.0000E+00	158
756.9	-187.8	0.0000E+00	26.43	-6.559	0.0000E+00	275.7	-68.40	0.0000E+00	159
740.0	-187.8	0.0000E+00	25.84	-6.559	0.0000E+00	269.5	-68.40	0.0000E+00	160
723.1	-187.8	0.0000E+00	25.25	-6.559	0.0000E+00	263.4	-68.40	0.0000E+00	161
706.2	-187.8	0.0000E+00	24.66	-6.559	0.0000E+00	257.2	-68.40	0.0000E+00	162
689.3	-187.8	0.0000E+00	24.07	-6.559	0.0000E+00	251.0	-68.40	0.0000E+00	163
672.4	-187.9	0.0000E+00	23.49	-6.559	0.0000E+00	244.9	-68.40	0.0000E+00	164
655.5	-187.9	0.0000E+00	22.89	-6.559	0.0000E+00	238.7	-68.40	0.0000E+00	165
638.6	-187.8	0.0000E+00	22.30	-6.559	0.0000E+00	232.6	-68.40	0.0000E+00	166
621.7	-187.9	0.0000E+00	21.71	-6.559	0.0000E+00	226.4	-68.40	0.0000E+00	167
604.8	-187.8	0.0000E+00	21.12	-6.559	0.0000E+00	220.3	-68.40	0.0000E+00	168
587.9	-187.8	0.0000E+00	20.53	-6.559	0.0000E+00	214.1	-68.40	0.0000E+00	169
571.0	-187.8	0.0000E+00	19.94	-6.559	0.0000E+00	207.9	-68.40	0.0000E+00	170
554.1	-187.8	0.0000E+00	19.35	-6.559	0.0000E+00	201.8	-68.40	0.0000E+00	171
537.2	-187.8	0.0000E+00	18.76	-6.559	0.0000E+00	195.6	-68.40	0.0000E+00	172
520.3	-187.8	0.0000E+00	18.17	-6.559	0.0000E+00	189.5	-68.40	0.0000E+00	173
503.4	-187.8	0.0000E+00	17.58	-6.559	0.0000E+00	183.3	-68.40	0.0000E+00	174
486.5	-187.8	0.0000E+00	16.99	-6.559	0.0000E+00	177.2	-68.40	0.0000E+00	175
469.6	-187.8	0.0000E+00	16.40	-6.559	0.0000E+00	171.0	-68.40	0.0000E+00	176
452.7	-187.8	0.0000E+00	15.81	-6.559	0.0000E+00	164.9	-68.40	0.0000E+00	177
435.8	-187.8	0.0000E+00	15.22	-6.559	0.0000E+00	158.7	-68.40	0.0000E+00	178
418.9	-187.8	0.0000E+00	14.63	-6.559	0.0000E+00	152.5	-68.40	0.0000E+00	179
401.9	-187.8	0.0000E+00	14.04	-6.559	0.0000E+00	146.4	-68.40	0.0000E+00	180
385.0	-187.8	0.0000E+00	13.45	-6.559	0.0000E+00	140.2	-68.40	0.0000E+00	181
368.1	-187.8	0.0000E+00	12.86	-6.559	0.0000E+00	134.1	-68.40	0.0000E+00	182
351.2	-187.8	0.0000E+00	12.27	-6.559	0.0000E+00	127.9	-68.40	0.0000E+00	183
334.3	-187.8	0.0000E+00	11.67	-6.559	0.0000E+00	121.8	-68.40	0.0000E+00	184
317.4	-187.8	0.0000E+00	11.08	-6.559	0.0000E+00	115.6	-68.40	0.0000E+00	185
300.5	-187.8	0.0000E+00	10.49	-6.559	0.0000E+00	109.4	-68.40	0.0000E+00	186
283.6	-187.8	0.0000E+00	9.904	-6.559	0.0000E+00	103.3	-68.40	0.0000E+00	187
266.7	-187.8	0.0000E+00	9.314	-6.559	0.0000E+00	97.13	-68.40	0.0000E+00	188
249.8	-187.8	0.0000E+00	8.723	-6.559	0.0000E+00	90.98	-68.40	0.0000E+00	189
232.9	-187.8	0.0000E+00	8.133	-6.559	0.0000E+00	84.82	-68.40	0.0000E+00	190
216.0	-187.8	0.0000E+00	7.543	-6.559	0.0000E+00	78.66	-68.40	0.0000E+00	191
199.1	-187.8	0.0000E+00	6.953	-6.559	0.0000E+00	72.51	-68.40	0.0000E+00	192
182.2	-187.8	0.0000E+00	6.362	-6.559	0.0000E+00	66.35	-68.40	0.0000E+00	193
165.3	-187.8	0.0000E+00	5.772	-6.559	0.0000E+00	60.20	-68.40	0.0000E+00	194
148.4	-187.8	0.0000E+00	5.182	-6.559	0.0000E+00	54.04	-68.40	0.0000E+00	195
131.5	-187.8	0.0000E+00	4.591	-6.559	0.0000E+00	47.88	-68.40	0.0000E+00	196
114.6	-187.8	0.0000E+00	4.001	-6.559	0.0000E+00	41.73	-68.40	0.0000E+00	197
97.67	-187.8	0.0000E+00	3.411	-6.559	0.0000E+00	35.57	-68.40	0.0000E+00	198
80.76	-187.8	0.0000E+00	2.820	-6.559	0.0000E+00	29.41	-68.40	0.0000E+00	199
63.86	-187.8	0.0000E+00	2.230	-6.559	0.0000E+00	23.26	-68.40	0.0000E+00	200

Fig. 7.1. x-position tracking performance.

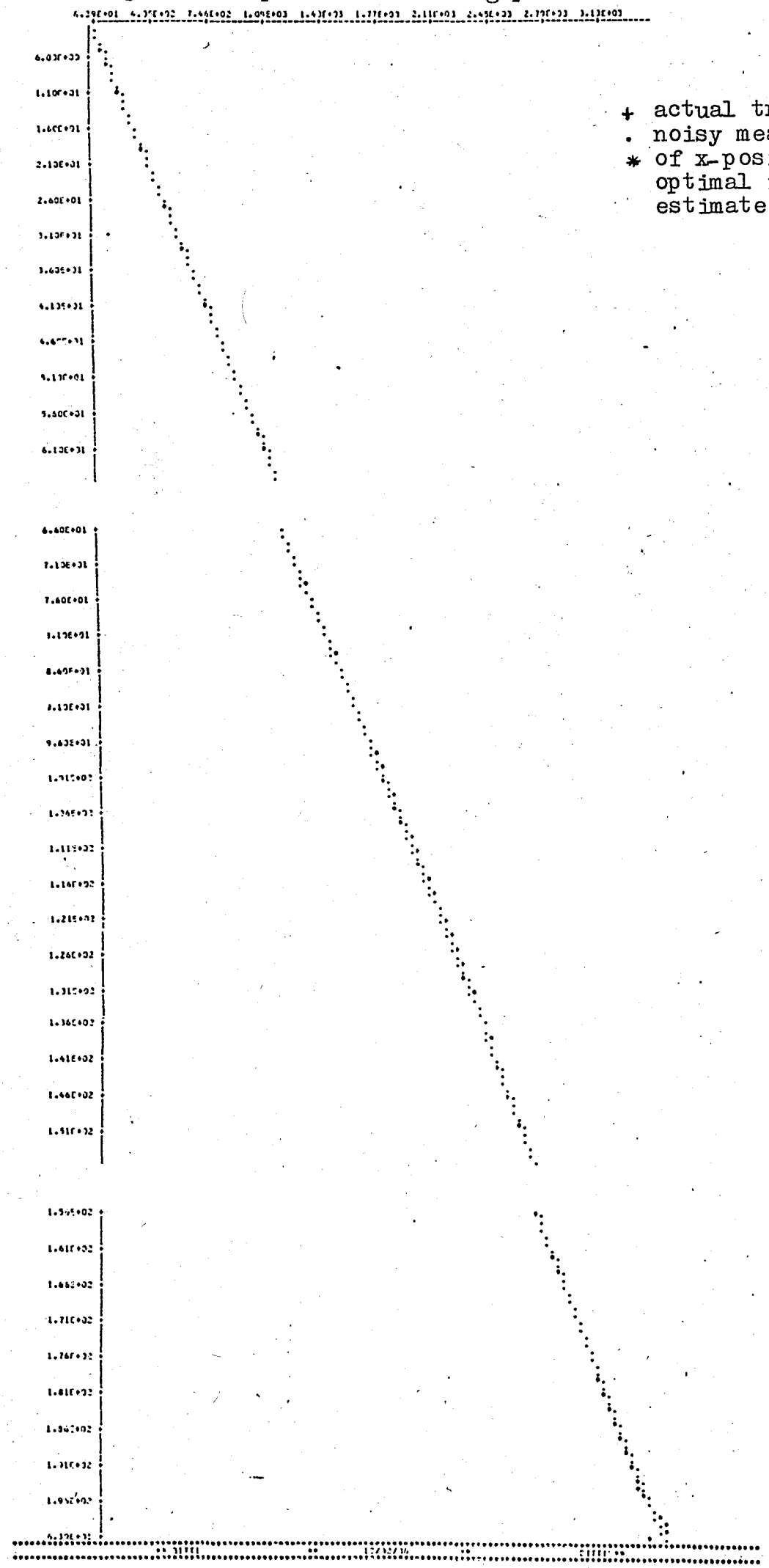


fig. 7.2. y-position tracking performance

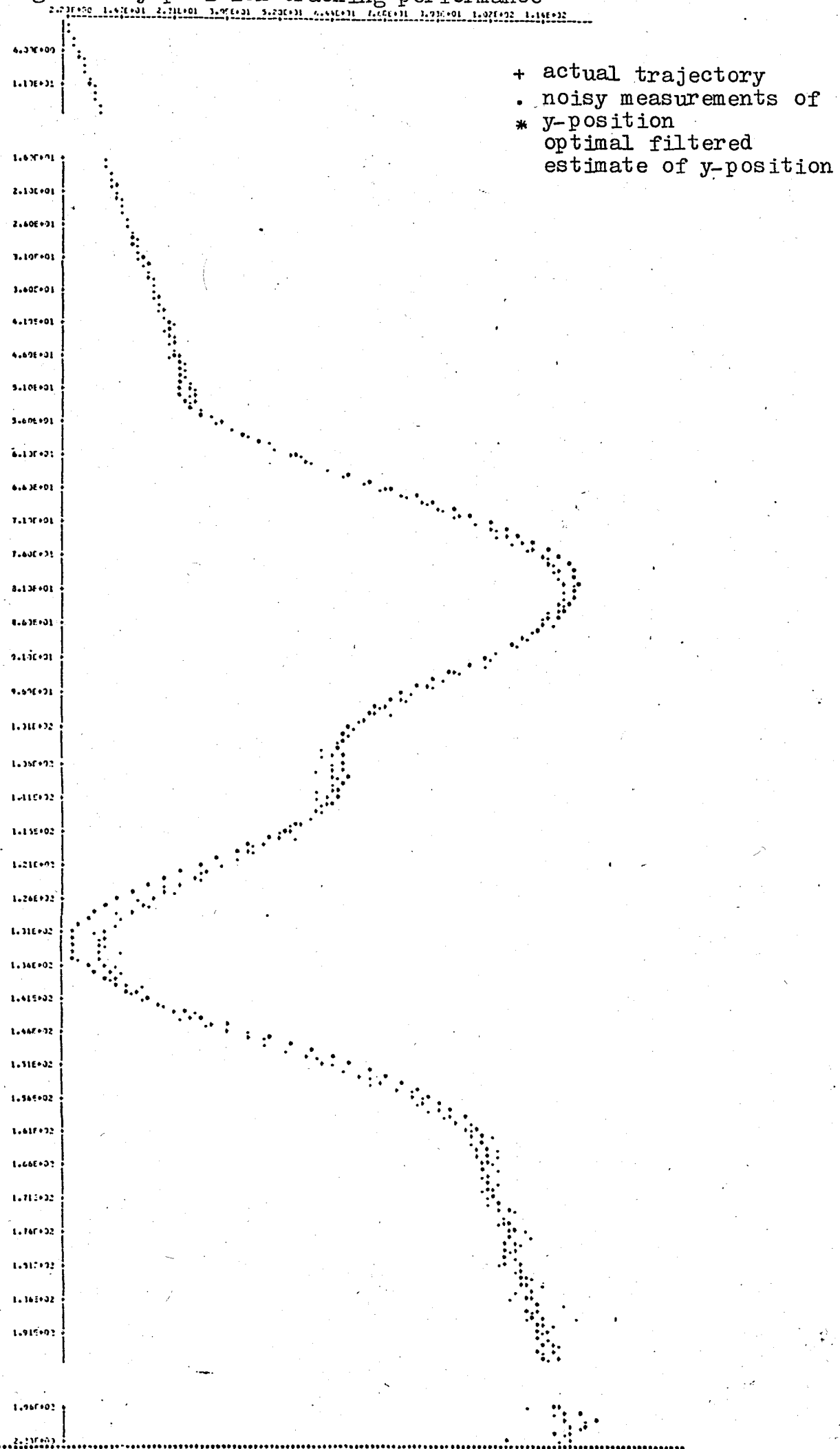


Fig. 7.3. z-position tracking performance

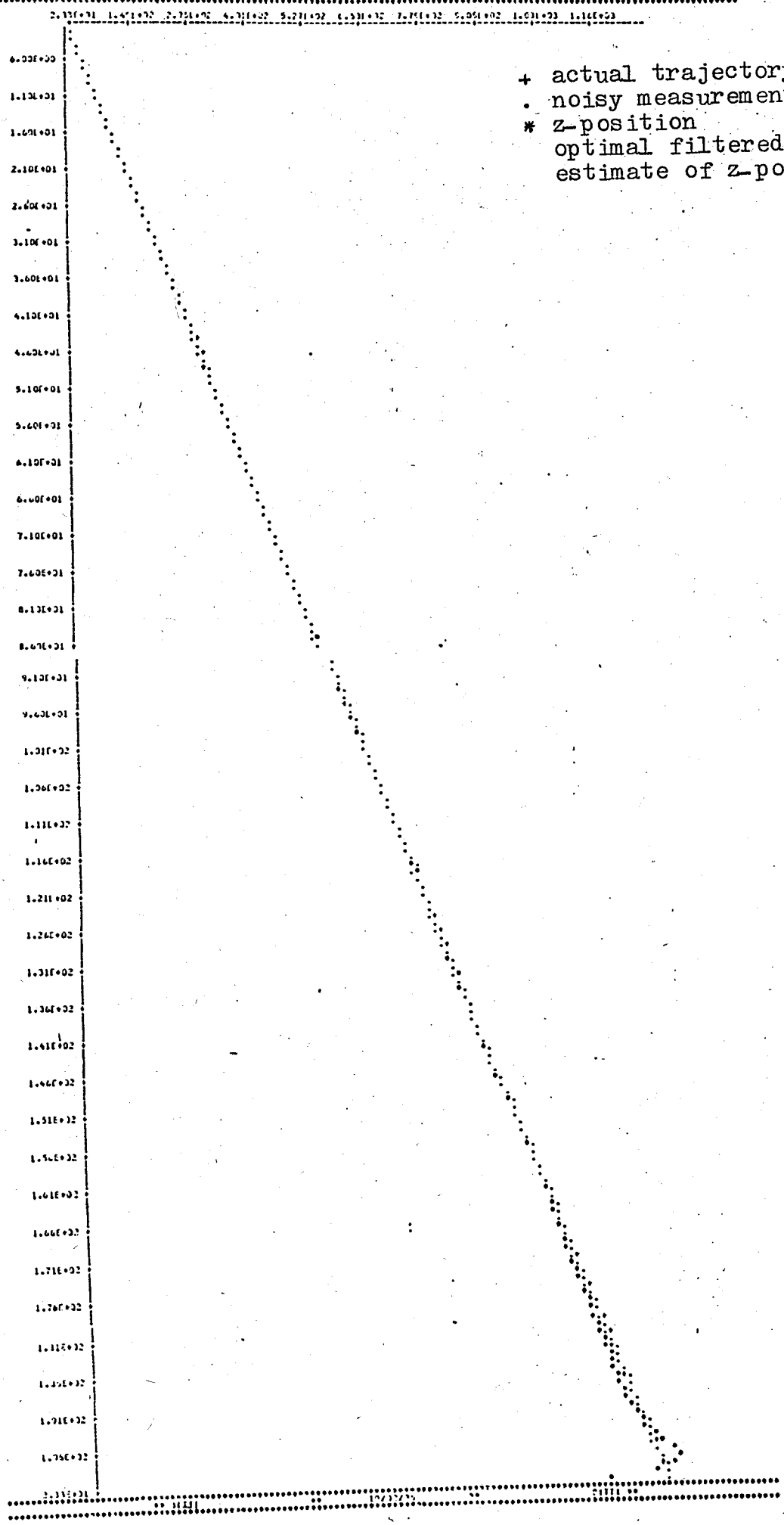
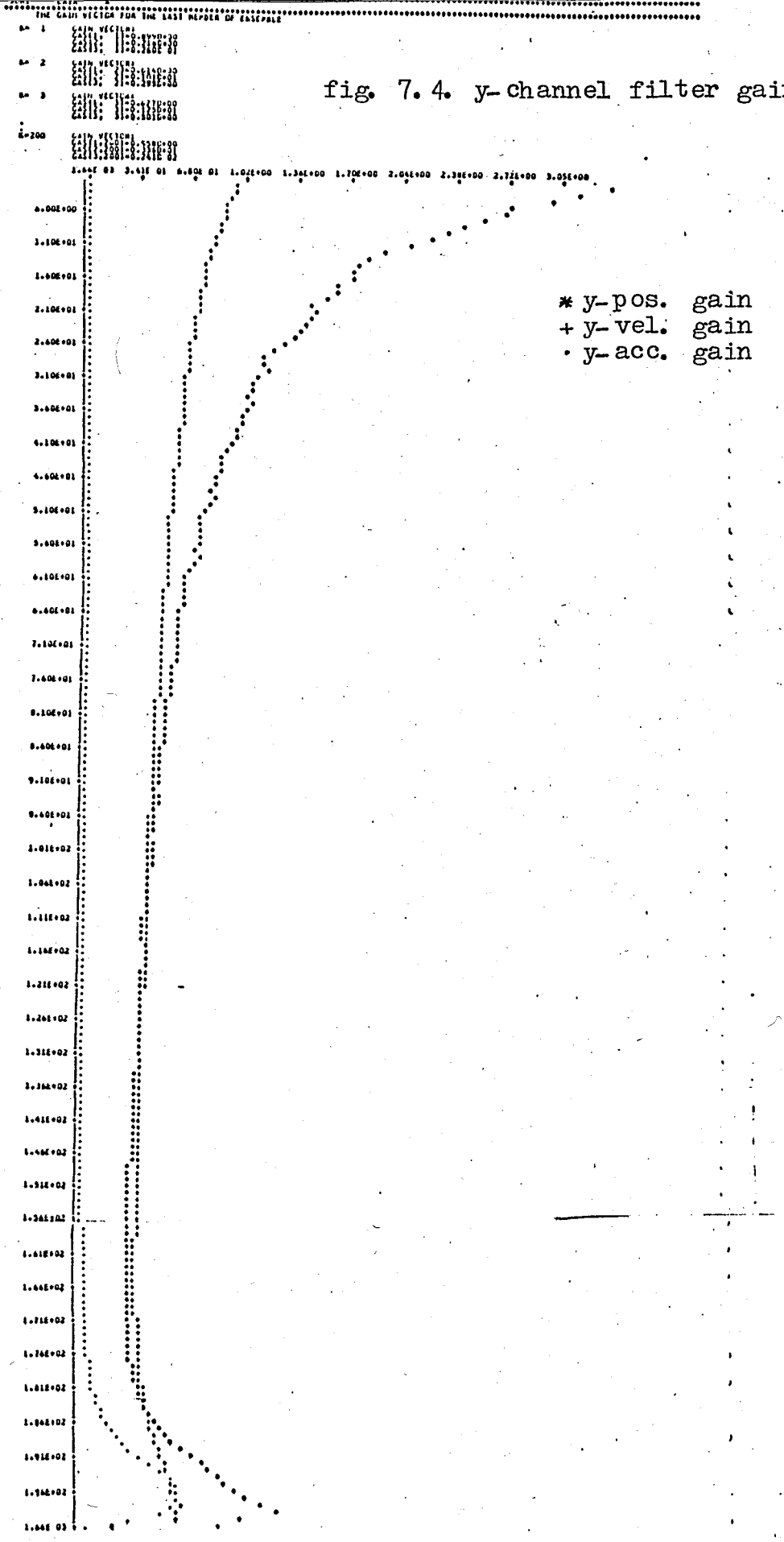


fig. 7.4. y-channel filter gains



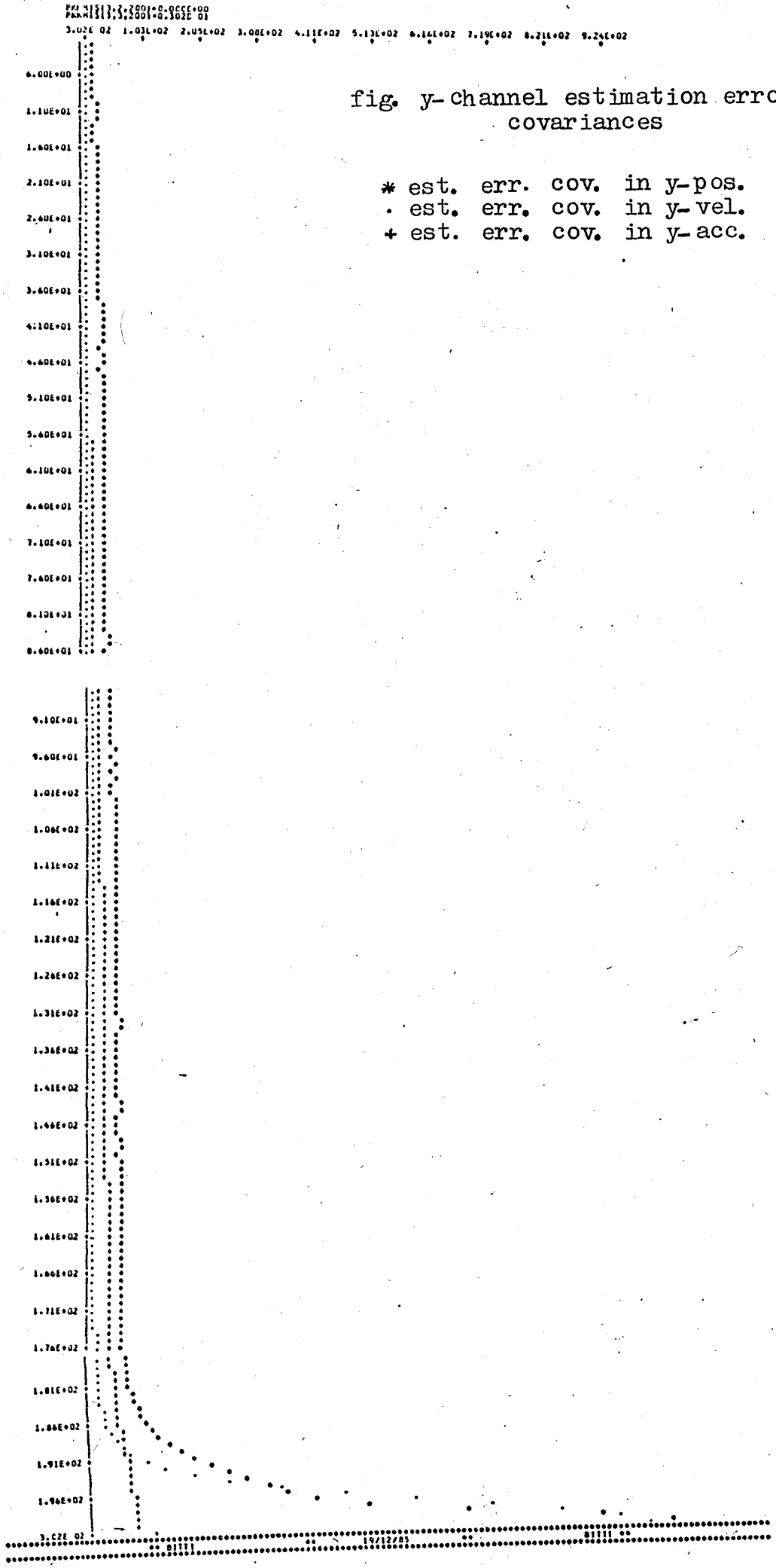


fig. y-channel estimation error covariances

* est. err. cov. in y-pos.
. est. err. cov. in y-vel.
\+ est. err. cov. in y-acc.

VIII. SUMMARY and CONCLUSIONS

In this study, various features of the target tracking problem have been discussed . Simulation results have shown that the dual-bandwidth adaptive filter configuration superimposing upper and lower bounds on the parameters on the random acceleration target model would yield sufficiently accurate estimated values for the system 's state . It is obvious that the basic approaches to the design of a tracking filter have been thoroughly investigated and many problems related to it have been solved . However, the problem of integrating the tracking system into the overall command, control and communication structure to achieve improved performance while minimizing data processing requirements represents the nature of problem that require current attention . More specifically, new problems still emerge and challenge system 's engineers . These problems can be listed as suggestions for further investigations.

i. An optimal nonlinear filter, i. e. the filter which does not take the simplifying assumptions into account during the modeling efforts should be designed to form a basis for testing the optimality of a particular design . This must be the starting point for the designer .

ii. The recent developments in target tracking on the focal plane using an infrared sensor are in the problem areas of both signal processing and target tracking . A survey of problems along with an extensive list of references have been presented in 5 .

iii. Tracking in a dense target environment with multiple sensors still represents the challenging side of the problem.

APPENDIX A

THE MONTE-CARLO SIMULATION PROGRAM

A.1. Program Description

The Monte-Carlo simulation program we have developed is quite simple and flexible in its basic form and offers several options to the user. The various options and many of the input variables and/or data requirements are explained by comments early in the program listing. Every step in the program has been explained by these detailed comments to make the algorithm more easily tractable.

A. 2. The Monte-Carlo program listing

```
C*************************************************************************
C*************************************************************************
C***          THE FOLLOWING PROGRAM PERFORMS THE MONTE-CARLO           ***
C***      SIMULATION OF DISCRETE LINEAR ADAPTIVE KALMAN FILTER         ***
C***      FOR TARGET TRACKING PURPOSES.THE DUAL BANDWIDTH ADAP         ***
C***      TATION SCHEME IS INVOLVED IN THE SIMULATION.THE APP-         ***
C***      ROXIMATION OF COMPLETE INDEPENDENCE AMONG CHANNELS           ***
C***      IS MADE. SO, THE FLAG (ICHN) IS UTILIZED FOR CHANNEL         ***
C***      SELECTION.ALSO,THE DYNAMIC EQUATION OF TARGET MOTION         ***
C***      IS ASSUMED TO BE IDENTICAL IN EACH COORDINATE AXIS..         ***
C*************************************************************************
C*************************************************************************
C          TYPE AND DIMENSION DECLARATIONS;

      DOUBLE PRECISION FA(3,3),PHIA(3,3),PHIAT(3,3),EECA(3,3),QA(3,3)
      DOUBLE PRECISION FB(3,3),PHIB(3,3),PHIBT(3,3),EECB(3,3),QB(3,3)
      DOUBLE PRECISION ALPHAA,E1A,E2A,ATA,SIGMA
      DOUBLE PRECISION ALPHAB,E1B,E2B,ATB,SIGMB
      DOUBLE PRECISION XESTA,CORA,RESA,GA,GA1,GAS
      DOUBLE PRECISION XESTB,CORB,RESB,GB,GB1,GBS
      DOUBLE PRECISION G,XCZ,Z,X1,X4,X7,XCST,ZR,SIGZR,MNVY
      DOUBLE PRECISION RON,ERR,VAR,EECS,EXC,Y1,Z1,RGE,MNVZ
      DOUBLE PRECISION E1,TEMP,TEMP1,TEMP2,H
      COMMON /L1/S(9,200)
      COMMON /L2/E1(3,3),TEMP(3,3),TEMP1(3,3),TEMP2(3),H(3)
      DIMENSION IN(3),X(3,200),Y(3,200),MNVY(200),MNVZ(200)
      DIMENSION XESTA(3),CORA(3,200),RESA(200),GA(3),GA1(200),GAS(3,200)
      DIMENSION XESTB(3),CORB(3,200),RESB(200),GB(3),GB1(200),GBS(3,200)
      DIMENSION XEST(3,200),RGE(200),SIGZR(200),RON(200),Z1(200),XC(3)
      DIMENSION ERR(3,200),VAR(3,3,200),EECS(3,3,200),EXC(3),ZR(200)
      DIMENSION G(200),XCZ(3),Z(200),X1(200),X4(200),X7(200),Y1(200)

      READ(5,*) N,M,NRF,NTS,NENS
C*********************************************************
C N    : ORDER OF THE SYSTEM MODEL                        *
C M    ; DIMENSION OF THE MEASUREMENT VECTOR Z            *
C NRF  : NUMBER OF RANDOM FORCING FUNCTIONS               *
C NTS  : NUMBER OF TIME SAMPLES                           *
C NENS; NUMBER OF MEMBERS IN THE ENSEMBLE                 *
C ICHN: THE FLAG FOR  CHANNEL SELECTION                   *
C IPHI: THE FLAG FOR  INSERTING STM IN THE PROGRAM        *
C ITRK; THE FLAG FOR TRACK MODE SELECTION                 *
C*********************************************************
      READ(5,*) ICHN
      READ(5,*) IPHI
      READ(5,*) ITRK
      READ(5,*) ALPHAA,ALPHAB,SIGMA,SIGMB
      READ(5,*) SIGR,SIGB,SIGE
      READ(5,*) SCONS
      READ(5,*) IX,MNVX,DEC
      READ(5,*) T,K,AX,PI
      CALL VRREAD (H,N)

C     FORM N*N IDENTITY MATRIX IN DOUBLE PRECISION

      DO 5 I=1,N
      DO 5 J=1,N
      E1(I,J)=0.D0
    5 IF(I.EQ.J) E1(I,J)=1.D0

      E1A=DEXP(-ALPHAA*T)
      E2A=DEXP(-2*ALPHAA*T)
      ATA=ALPHAA*T
      E1B=DEXP(-ALPHAB*T)
      E2B=DEXP(-2*ALPHAB*T)
      ATB=ALPHAB*T

      IF(IPHI) 6,7,8
    6 CALL MREAD (PHIA,N)
      CALL MREAD (PHIB,N)
      GO TO 15
    7 CONTINUE
      CALL MREAD (FA,N)
      CALL DISCRT (FA,PHIA,T,K,N,M)
      WRITE(6,601)
  601 FORMAT(/,10X,'DISCRT-TIME EQUIVALENT OF FA,',/)
      WRITE(6,*)'MATRIX PHIA :'
      CALL MWRITE (PHIA,N)

      CALL MREAD (FB,N)
      CALL DISCRT (FB,PHIB,T,K,N,M)
      WRITE(6,602)
  602 FORMAT(/,10X,'DISCRT-TIME EQUIVALENT OF FB,',/)
      WRITE(6,*)'MATRIX PHIB :'
      CALL MWRITE (PHIB,N)
      GO TO 15
    8 CONTINUE

C     STM FOR FILTER 'A' ;
```

```
      PHIA(1,2)=T
      PHIA(1,3)=(1/(ALPHAA**2))*(-1+ATA+E1A)
      PHIA(2,1)=0.D0
      PHIA(2,2)=1.D0
      PHIA(2,3)=(1/ALPHAA)*(1-E1A)
      PHIA(3,1)=0.D0
      PHIA(3,2)=0.D0
      PHIA(3,3)=E1A

      STM FOR FILTER 'B' :

      PHIB(1,1)=1.D0
      PHIB(1,2)=T
      PHIB(1,3)=(1/(ALPHAB**2))*(-1+ATB+E1B)
      PHIB(2,1)=0.D0
      PHIB(2,2)=1.D0
      PHIB(2,3)=(1/ALPHAB)*(1-E1B)
      PHIB(3,1)=0.D0
      PHIB(3,2)=0.D0
      PHIB(3,3)=E1B
      WRITE(6,603)
603   FORMAT(//,10X,'STATE TRANSITION MATRIX A;',/)
      CALL MWRITE (PHIA,N)
      WRITE(6,604)
604   FORMAT(//,10X,'STATE TRANSITION MATRIX B;',/)
      CALL MWRITE (PHIB,N)
 15   CONTINUE
      WRITE(6,605)
605   FORMAT(//,10X,'VECTOR H;',/)
      CALL VWRITE (H,N)

      COVARIANCE INITIALIZATION EQ'S. WE MAKE THE ASSUMPTION THAT
      THE UNCERTAINTY IN THE INITIAL ESTIMATE IS RATHER LARGE.

      EECA(1,1)=1.D03
      EECA(1,2)=0.D0
      EECA(1,3)=0.D0
      EECA(2,1)=0.D0
      EECA(2,2)=1.D03
      EECA(2,3)=0.D0
      EECA(3,1)=0.D0
      EECA(3,2)=0.D0
      EECA(3,3)=1.D03
      DO 25 I=1,N
      DO 25 J=1,N
 25   EECB(I,J)=EECA(I,J)
      WRITE(6,606)
606   FORMAT(//,10X,'INITIAL EST.ERR. COVARIANCE MATRIX P(0/-1)',/)
      CALL MWRITE (EECA,N)

      COVARIANCE MATRIX OF PROCESS NOISE FOR FILTER 'A' ;

      QA(1,1)=AX**2*((T**2/2-(1/(ALPHAA**2))*(-1+ATA+E1A))**2)
      QA(1,2)=AX**2*(T**2/2-(1/(ALPHAA**2))*(-1+ATA+E1A))*(T-(1/ALPHAA)*
     *(1-E1A))
      QA(1,3)=AX**2*(T**2/2-(1/(ALPHAA**2))*(-1+ATA+E1A))*(1-E1A)
      QA(2,1)=QA(1,2)
      QA(2,2)=AX**2*((T-(1/ALPHAA)*E1A))**2
      QA(2,3)=AX**2*(T-(1/ALPHAA)*(1-E1A))*(1-E1A)
      QA(3,1)=QA(1,3)
      QA(3,2)=QA(2,3)
      QA(3,3)=AX**2*((1-E2A)**2)

      COVARIANCE MATRIX OF PROCESS NOISE FOR FILTER 'B' ;

      QB(1,1)=AX**2*((T**2/2-(1/(ALPHAB**2))*(-1+ATB+E1B))**2)
      QB(1,2)=AX**2*(T**2/2-(1/(ALPHAB**2))*(-1+ATB+E1B))*(T-(1/ALPHAB)*
     *(1-E1B))
      QB(1,3)=AX**2*(T**2/2-(1/(ALPHAB**2))*(-1+ATB+E1B))*(1-E1B)
      QB(2,1)=QB(1,2)
      QB(2,2)=AX**2*((T-(1/ALPHAB)*E1B))**2
      QB(2,3)=AX**2*(T-(1/ALPHAB)*(1-E1B))*(1-E1B)
      QB(3,1)=QB(1,3)
      QB(3,2)=QB(2,3)
      QB(3,3)=AX**2*((1-E2B)**2)
      WRITE(6,607)
607   FORMAT(//,10X,'PROCESS NOISE COVARIANCE MATRIX QA;',/)
      CALL MWRITE (QA,N)
      WRITE(6,608)
608   FORMAT(//,10X,'PROCESS NOISE COVARIANCE MATRIX QB;',/)
      CALL MWRITE (QB,N)

      ASSIGN VALUES FOR THE OPTIMAL FILTERED ESTIMATE XCAP(0/-1)
      FOR THE INITIATION OF KALMAN FILTER EQ'S.

      DATA XCZ/0.0,0.0,0.0/

      SET UP ARRAYS FOR COMPUTING STATISTICS

      DO 80 K=1,NTS
```

```
      ERR(I,K)=0.
 80 VAR(I,I,K)=0.

                TRACK MODE SELECTION

      IF(ITRK) 1,2,3
      STORE ACTUAL TRAJECTORY IN S(9,200).IT IS TO BE GENERATED
      FROM THE NOISE-FREE DYNAMICS, I.E. X(K)=PHI(K,K-1)*X(K-1)
       ( USER SHOULD PROVIDE FOR THE APPROPRIATE INSTRUCTIONS )
  1   CONTINUE
      STORE ACTUAL TRAJECTORY IN S(9,200).IT IS TO BE GENERATED
      BY THE USER IN ANY FORM.
       ( USER SHOULD PROVIDE FOR THE APPROPRIATE INSTRUCTIONS )
  2   CONTINUE
      STORE ACTUAL TRAJECTORY IN S(9,200).IT IS TO BE GENERATED
      BY CALL OF SUBROUTINE TRKGEN
  3   CONTINUE

============================
      CALL TRKGEN
============================
      WRITE(6,609)
609 FORMAT('1',4X,'X-POS       X-VEL       X-ACC       Y-POS       Y-VEL
     *       Y-ACC       Z-POS       Z-VEL       Z-ACC',/)
      DO 35 J=1,NTS
      X(1,J)=J
      CHANNEL SELECTION
      IF(ICHN) 501,502,503
501 Y(1,J)=S(1,J)
      GO TO 35
502 Y(1,J)=S(4,J)
      GO TO 35
503 Y(1,J)=S(7,J)
 35 CONTINUE
      DO 40 J=1,NTS
 40 WRITE(6,610) (S(I,J),I=1,9),J
610 FORMAT(' ',9G12.4,5X,I5)

        ACTUAL TRAJECTORY IS GENERATED FOR EACH CHANNEL.NOTE THAT
          S(9,160) IS SPECIFIED FOR X,Y & Z CHANNEL N-VECTORS.

      WRITE(6,611)
611 FORMAT(//,10X,'THE FIRST AND LAST DATA POINTS ON THE TRACK TO BE
     *USED ARE;',/)
      WRITE(6,612) (I,S(I,1),I=1,9)
      WRITE(6,613) (I,S(I,NTS),I=1,9)
612 FORMAT(10X,'S(',I1,',1)=',E9.3)
613 FORMAT(10X,'S(',I1,',200)=',E9.3)

       THE FOLLOWING SECTION IS THE MAIN ITERATION LOOP FOR THE
                  MONTE-CARLO SIMULATION PROGRAM

      DO 101 ITER=1,NENS

      DO 102 I=1,N
102 XC(I)=XCZ(I)
      DO 101 K=1,NTS
      CALL RANDOM (IX,IY,V)
      IX=IX+3
      I1=IX/5
      IF(I1.EQ.(IX/5)) IX=IX+1
      CALL RANDOM (IX,IY,V1)

      CHANNEL SELECTION
      IF(ICHN) 100,200,300

          X-CHANNEL FILTERING.COVARIANCE MATRIX OF MEASUREMENT ERROR
          IS NOW REDUCED TO THE SCALAR SIGR**2
100 RON(K)=SIGR**2
      ICS=0
      X1(K)=S(1,K)
          FORM NOISY MEASUREMENT OF X-POSITION FROM STATE VALUE,I.E.;
      G(K)=SQRT(-2*ALOG(V))*COS(PI*V1)*MNVX
      Z(K)=X1(K)+G(K)
      GO TO 1000

          Y-CHANNEL FILTERING.COVARIANCE MATRIX OF MEASUREMENT ERROR
          IS NOW REDUCED TO THE SCALAR RGE(K)*SIGB**2
200 RGE(K)=S(1,K)**2+S(4,K)**2+S(7,K)**2
      RON(K)=RGE(K)*SIGB**2
      ICS=3
      MNVY(K)=SQRT(RGE(K))*DEC
      Y1(K)=S(4,K)
          FORM NOISY MEASUREMENT OF Y-POSITION FROM STATE VALUE,I.E.;
      G(K)=SQRT(-2*ALOG(V))*COS(PI*V1)*MNVY(K)
      Z(K)=Y1(K)+G(K)
      GO TO 1000

          Z-CHANNEL FILTERING.COVARIANCE MATRIX OF MEASUREMENT ERROR
```

```
      IS NOW REDUCED TO THE SCALAR RGE(K)*SIGE**2
300   RGE(K)=S(1,K)**2+S(4,K)**2+S(7,K)**2
      RON(K)=RGE(K)*SIGE**2
      ICS=6
      MNVZ(K)=SQRT(RGE(K))*DEC
      Z1(K)=S(7,K)
      FORM NOISY MEASUREMENT OF Z-POSITION FROM STATE VALUE, I.E.;
      G(K)=SQRT(-2*ALOG(V))*COS(PI*V1)*MNVZ(K)
      Z(K)=Z1(K)+G(K)
1000  CONTINUE
      X(3,K)=K
      Y(3,K)=Z(K)

          ****************************************
          ******** FILTER COMPUTATIONS ***********
          ****************************************

      1. STORE FILTER STATE XCAP(O/-1),P(O/-1)

      IF(K.GT.1) GO TO 104
      DO 103 I=1,N
      XESTA(I)=XC(I)
      XESTB(I)=XC(I)
103   CONTINUE
104   CONTINUE

      2. COMPUTE THE PREDICTED STATE
      XCAP(K/K-1)=PHI(K,K-1)*XCAP(K,K-1)

      CALL VECMUL (PHIA,XESTA,TEMP2,N)
      DO 105 I=1,N
105   XESTA(I)=TEMP2(I)
      CALL VECMUL (PHIB,XESTB,TEMP2,N)
      DO 106 I=1,N
106   XESTB(I)=TEMP2(I)

      3. COMPUTE THE PREDICTED ERR. COV. MATRIX
      P(K/K-1)=PHI(K,K-1)*P(K-1,K-1)*PHIT(K,K-1)+Q(K)

      CALL TRANS (PHIA,PHIAT,N)
      CALL MATMUL (EECA,PHIAT,TEMP,N)
      CALL MATMUL (PHIA,TEMP,TEMP1,N)
      CALL MADD (TEMP1,QA,EECA,N)

      CALL TRANS (PHIB,PHIBT,N)
      CALL MATMUL (EECB,PHIBT,TEMP,N)
      CALL MATMUL (PHIB,TEMP,TEMP1,N)
      CALL MADD (TEMP1,QB,EECB,N)

      4. COMPUTE FILTER GAIN MATRIX
      K(K)=P(K/K-1)*HT*((H*P(K,K-1)*HT+R(K)**-1

      CALL VECMUL (EECA,H,TEMP2,N)
      CALL SPROD (H,TEMP2,TEMP4,N)
      GA1(K)=1/(TEMP4+RON(K))
      DO 107 I=1,N
107   GA(I)=TEMP2(I)*GA1(K)
      DO 108 I=1,N
108   GAS(I,K)=GA(I)

      CALL VECMUL (EECB,H,TEMP2,N)
      CALL SPROD (H,TEMP2,TEMP4,N)
      GB1(K)=1/(TEMP4+RON(K))
      DO 109 I=1,N
109   GB(I)=TEMP2(I)*GB1(K)
      DO 110 I=1,N
110   GBS(I,K)=GB(I)

      5. PROCESS THE OBSERVATION Z(K)
      XCAP(K/K)=XCAP(K/K-1)+K(K)*((Z(K)-H*XCAP(K/K-1))

      CALL SPROD (H,XESTA,TEMP4,N)
      RESA(K)=Z(K)-TEMP4
      DO 111 I=1,N
111   CORA(I,K)=GAS(I,K)*RESA(K)
      DO 112 I=1,N
112   TEMP2(I)=XESTA(I)+CORA(I,K)
      DO 113 I=1,N
113   XESTA(I)=TEMP2(I)

      CALL SPROD (H,XESTB,TEMP4,N)
      RESB(K)=Z(K)-TEMP4
      DO 114 I=1,N
114   CORB(I,K)=GBS(I,K)*RESB(K)
      DO 115 I=1,N
115   TEMP2(I)=XESTB(I)+CORB(I,K)
      DO 116 I=1,N
116   XESTB(I)=TEMP2(I)
```

```
      ERROR.IF(ABS)ZR(K/K-1)=(Z(K)-H*XCAP(K/K-1)/SIGZR(K) EXCEEDS SOME
      SPECIFIED VALUE SCONS,MANEUVER IS DECLARED.

      SIGZR(K)=SIGR**2+EECA(1,1)
      ZR(K)=RESA(K)/SIGZR(K)
      IF(CABS(ZR(K)).GT.SCONS) GO TO 117
      DO 118 I=1,N
  118 XEST(I,K)=XESTA(I)
      GO TO 119
  117 DO 120 I=1,N
  120 XESTA(I)=XESTB(I)
      DO 121 I=1,N
  121 XEST(I,K)=XESTA(I)
      CALL MADD (EECA,GB,TEMP,N)
      DO 122 I=1,N
      DO 122 J=1,N
  122 EECA(I,J)=TEMP(I,J)
  119 CONTINUE

C     6.COMPUTE THE NEW ERROR COVARIANCE MATRIX
C     P(K/K)=(I-K(K)*H)*P(K/K-1)

      CALL VEKMUL (GA,H,TEMP,N)
      CALL MATSUB (EI,TEMP,TEMP1,N)
      CALL MATMUL (TEMP1,EECA,TEMP,N)
      DO 123 I=1,N
      DO 123 J=1,N
  123 EECA(I,J)=TEMP(I,J)
      DO 124 I=1,N
      DO 124 J=1,N
      EECS(I,J,K)=0.DO
  124 IF(I.EQ.J) EECS(I,J,K)=EECA(I,J)
      CALL VEKMUL (GB,H,TEMP,N)
      CALL MATSUB (EI,TEMP,TEMP1,N)
      CALL MATMUL (TEMP1,EECB,TEMP,N)
      DO 125 I=1,N
      DO 125 J=1,N
  125 EECB(I,J)=TEMP(I,J)

C     7. SET K=K+1 AND RETURN TO STEP 1
C     BEFORE INCREMENTING K WE WILL UPDATE RUNNING
C     SUMS USED IN COMPUTING STATISTICS

      X(2,K)=K
      Y(2,K)=XEST(1,K)
      DO 101 I=1,N
      EXC(I)=XEST(I,K)-S(I+ICS,K)
      ERR(I,K)=ERR(I,K)+EXC(I)
      VAR(I,I,K)=VAR(I,I,K)+EXC(I)**2
  101 CONTINUE

C     PLOT XCAP(K/K) AND S(I,K) ON THE SAME GRAPH
      IN(1)=200
      IN(2)=200
      IN(3)=200
      CALL MPLOT1 (100,200,Y,X,IN,0,3)

C     TO COMPUTE STATISTICS DIVIDE RUNNING SUMS BY NENS

      ENS=NENS
      DO 45 K=1,NTS
      DO 45 J=1,N
      ERR(J,K)=ERR(J,K)/ENS
   45 VAR(J,J,K)=VAR(J,J,K)/ENS+ERR(J,K)**2
      WRITE(6,614)
  614 FORMAT('1',20X,'OUTPUT DATA',///)
      DO 95 K=1,NTS
      X(1,K)=K
      Y(1,K)=ERR(1,K)
      X(2,K)=K
      Y(2,K)=ERR(2,K)
      X(3,K)=K
      Y(3,K)=ERR(3,K)
   95 CONTINUE
      IN(1)=200
      IN(2)=200
      IN(3)=200
      CALL MPLOT1 (100,200,Y,X,IN,0,3)
      WRITE(6,615)
  615 FORMAT(5X,'THE GAIN VECTOR FOR THE LAST MEMBER OF ENSEMBLE')
      DO 90 K=1,NTS
      DO 90 I=1,N
      IF(I.EQ.1) WRITE(6,616) K
   90 WRITE(6,617) I,K,GAS(I,K)
  616 FORMAT(/,2X,'K=',I3,5X,'GAIN VECTOR;')
  617 FORMAT(12X,'GAS(',I1,',',I3,')=',E0.3)
      DO 50 K=1,NTS
      X(1,K)=K
```

```
      Y(2,K)=GAS(2,K)
      X(3,K)=K
      Y(3,K)=GAS(3,K)
   50 CONTINUE
      IN(1)=200
      IN(2)=200
      IN(3)=200
      CALL MPLOT1 (100,200,Y,X,IN,0,3)
      WRITE(6,650)
  650 FORMAT(2X,'COV. MATRIX OF EST. ERROR FOR THE LAST MEMBER  OF THE
     *ENSEMBLE',/)
      DO 55 K=1,NTS
      DO 55 I=1,N
      DO 55 J=1,N
      IF(I.EQ.1.AND.J.EQ.1) WRITE(6,618) K
   55 WRITE(6,619) I,J,K,EECS(I,J,K)
  618 FORMAT(/,2X,'K=',I3,5X,'MATRIX  P(K/K-1):')
  619 FORMAT(12X,'P(',I1,',',I1,',',I3,')=',E9.3)
      DO  60 K=1,NTS
      X(1,K)=K
      Y(1,K)=EECS(1,1,K)
      X(2,K)=K
      Y(2,K)=EECS(2,2,K)
      X(3,K)=K
      Y(3,K)=EECS(3,3,K)
   60 CONTINUE
      IN(1)=200
      IN(2)=200
      IN(3)=200
      CALL MPLOT1 (100,200,Y,X,IN,0,3)
      WRITE(6,620)
      WRITE(6,621)
      WRITE(6,622)
  621 FORMAT(T5,'TIME',T16,'VECTOR COM-',T34,'    SAMPLE MEAN OF',
     *T60,'SAMPLE VARIANCE OF')
  622 FORMAT(T5,'INDEX',T16,'PONENT INDEX',T34,'    ESTIMATION ERROR',
     *T60,'ESTIMATION ERROR',/)
  620 FORMAT('1')
  623 FORMAT(6X,I3,10X,I1,10X,2(10X,E9.3))
  624 FORMAT(/)
      DO 75 K=1,NTS
      WRITE(6,624)
      DO 75 I=1,N
   75 WRITE(6,623) K,I,ERR(I,K),VAR(I,I,K)
      STOP
      END

      SUBROUTINE DISCRT (F,PHI,T,KH,N,M)

      THIS SUBROUTINE PERFORMS DISCRETIZATION OF  CONTINUOUS TIME
      STATE EQUATIONS OF A GIVEN SYSTEM.

      DOUBLE PRECISION PHI(3,3),PSI(3,3),FS(3),F(3,3),TAVS(3)
      DOUBLE PRECISION EI,TEMP,TEMP1,TEMP2,H
      COMMON /L2/EI(3,3),TEMP(3,3),TEMP1(3,3),TEMP2(3),H(3)
      WRITE(6,*) 'MATRIX F ;'
      CALL MWRITE (F,N)
      DO 1 I=1,N
      DO 1 J=1,N
    1 PSI(I,J)= EI(I,J)
      K=KH
   35 IF(K.EQ.1) GO TO 40
      CALL SCALE (F,T/K,FS,N)
      CALL MATMUL (FS,PSI,TEMP,N)
      CALL MADD (EI,TEMP,PSI,N)
      K=K-1
      GO TO 35
   40 CONTINUE
      CALL VECMUL (PSI,H,TEMP2,N)
      CALL VSCALE (TEMP2,T,TAVS,N)
      CALL MATMUL (F,PSI,TEMP,N)
      CALL SCALE (TEMP,T,TEMP1,N)
      CALL MADD (EI,TEMP1,PHI,N)
      RETURN
      END

      SUBROUTINE SCALE (X,CONS,XS,N)

      THIS SUBROUTINE MULTIPLIES THE N*N MATRIX X BY A  CONSTANT
      (CONS) STORING THE RESULT IN XS

      DOUBLE PRECISION X(N,N),XS(N,N)
      DO 1 I=1,N
      DO 1 J=1,N
    1 XS(I,J)=X(I,J)*CONS
      RETURN
      END
```

```
      DOUBLE PRECISION X(N,N),Y(N,N),SUM(N,N)
      DO 1 I=1,N
      DO 1 J=1,N
    1 SUM(I,J)=X(I,J)+Y(I,J)
      RETURN
      END

      SUBROUTINE MATSUB (X,Y,DIFF,N)

C     THIS SUBROUTINE PERFORMS THE SUBTRACTION OF TWO N*N MATRICES X & Y
C     STORING  THE RESULT IN DIFF
      DOUBLE PRECISION X(N,N),Y(N,N),DIFF(N,N)
      DO 1 I=1,N
      DO 1 J=1,N
    1 DIFF(I,J)=X(I,J)-Y(I,J)
      RETURN
      END

      SUBROUTINE MATMUL (X,Y,Z,N)

C     THIS SUBROUTINE PERFORMS THE MULTIPLICATION OF TWO MATRICES X & Y
C     AND STORES THE RESULT IN MATRIX Z ( X,Y & Z ARE N*N MATRICES)

      DOUBLE PRECISION X(N,N),Y(N,N),Z(N,N)
      DO 2 I=1,N
      DO 2 J=1,N
      Z(I,J)=0.
      DO 2 K=1,N
    2 Z(I,J)=Z(I,J)+X(I,K)*Y(K,J)
      RETURN
      END

      SUBROUTINE MWRITE (A,N)

C     THIS SUBROUTINE WRITES THE ENTRIES OF THE N*N MATRIX A
C     ACCORDING TO FORMAT E9.3

      DOUBLE PRECISION A(N,N)
      I=1
    2 WRITE(6,1) (A(I,J),J=1,N)
    1 FORMAT(5X,10(2X,E9.3))
      IF(I.EQ.N) GO TO 3
      I=I+1
      GO TO 2
    3 CONTINUE
      RETURN
      END

      SUBROUTINE MREAD (A,N)

C     THIS SUBROUTINE READS THE ENTRIES OF THE N*N MATRIX A
C     (FORMAT-FREE READING)
      DOUBLE PRECISION A(N,N)
      DO 5 I=1,N
    5 READ(5,*) (A(I,J),J=1,N)
      RETURN
      END

      SUBROUTINE VREAD (A,N)

C     THIS SUBROUTINE READS THE ENTRIES OF THE N-VECTOR A
C     (FORMAT-FREE READING)
      DOUBLE PRECISION A(N)
      READ(5,*) (A(J),J=1,N)
      RETURN
      END

      SUBROUTINE TRANS (A,AT,N)

C     THIS SUBROUTINE TRANSPOZES AN N*N MATRIX A
C     STORING THE RESULT IN AT
      DOUBLE PRECISION A(N,N),AT(N,N)
      DO 5 I=1,N
      DO 5 J=1,N
      AT(J,I)=A(I,J)
    5 CONTINUE
      RETURN
      END

      SUBROUTINE VECMUL (A,X,Z,N)

C     THIS SUBROUTINE PERFORMS THE MULTIPLICATION OF N*N MATRICES
C     BY  N-VECTORS AND STORES THE RESULT IN MATRIX Z

      DOUBLE PRECISION A(N,N),X(N),Z(N)
      DO 3 I=1,N
      Z(I)=0.
      DO 3 J=1,N
```

```
      RETURN
      END

      SUBROUTINE VWRITE (V,N)

C     THIS SUBROUTINE WRITES THE ENTRIES OF  N-VECTOR V
C     ACCORDING TO FORMAT E9.3
      DOUBLE PRECISION V(N)
      WRITE(6,1) (V(I),I=1,N)
    1 FORMAT(10X,E9.3)
      RETURN
      END

      SUBROUTINE VSCALE (V,CONS,VS,N)

C     THIS SUBROUTINE MULTIPLIES AN N-VECTOR BY A CONSTANT
C     AND STORES THE RESULT IN VS

      DOUBLE PRECISION V(N),VS(N)
      DO 1 I=1,N
    1 VS(I)=V(I)*CONS
      RETURN
      END

      SUBROUTINE SPROD (V1,V2,SCA,N)

C     THIS SUBROUTINE MULTIPLIES AN N-VECTOR BY ITS TRANSPOSE
C     AND STORES THE RESULT IN SCA ,I.E. THE INNER PRODUCT <V1,V2>

      DOUBLE PRECISION V1(N),V2(N)
      SCA=0.0
      DO 1 I=1,N
    1 SCA=SCA+V1(I)*V2(I)
      RETURN
      END

      SUBROUTINE VEKMUL (V1,V2,RM,N)

C     THIS SUBROUTINE PERFORMS THE MULTIPLICATION OF TWO N-VECTORS
C     V1 & V2 ,GIVING N*N MATRIX RM

      DOUBLE PRECISION RM,V1,V2
      DIMENSION V1(N),V2(N),RM(N,N)
      DO 1 I=1,N
      DO 1 J=1,N
    1 RM(I,J)=0.
      DO 2 I=1,N
      DO 2 J=1,N
    2 RM(I,J)=V1(I)*V2(J)
      RETURN
      END

      SUBROUTINE TRKGEN

C     THIS SUBROUTINE GENERATES  SIMULATED AIRCRAFT TRAJECTORIES.

      COMMON /L1/S(9,200)
      DIMENSION Y(4),F(4),TEMPL(200)
      DOUBLE PRECISION B,E,Y,TEMPL,PHI
      DATA RATE,V,A,XPER,DR,L,RMAX,TRKEND/0.09,200,75.5,2000,2,20,3000,
     *-9999.90/
      DATA CPA,HO,NSAM1,N1,ICOORD/0,0,200,9,0/
      S(1,1)=3650
      S(2,1)=-200
      DO 2 I=3,9
    2 S(I,1)=0.0

C     THE SYSTEM STATES DEFINED AS FOLLOWS;
C          S(1,K)=X-POSITION
C          S(2,K)=X-VELOCITY
C          S(3,K)=X-ACCELERATION
C          S(4,K)=Y-POSITION
C          S(5,K)=Y-VELOCITY
C          S(6,K)=Y-ACCELERATION
C          S(7,K)=Z-POSITION
C          S(8,K)=Z-VELOCITY
C          S(9,K)=Z-ACCELERATION
C          K=1,2,3,.......,NSAM1
      K=1

C     THE FOLLOWING SECTION GENERATES A STRAIGHT SECTION OF TRACK
C     BEGINNING AT S(1,1),AND ENDING AT RMAX (OR WHEN K=NSAM1 IF A
C     STRAIGHT TRACK IS DESIRED)

    3 K=K+1
      M=K-1
      DO 4 I=3,6
    4 S(I,K)=0.0
      S(2,K)=-V
```

```
      S(1,K)=S(1,M)+S(2,M)*RATE
      IF(K.GE.NSAM1.OR.S(1,K).LE.TRKEND) GO TO 10
      IF(S(1,K).LE.RMAN) GO TO 5
      GO TO 3

C     THE FOLLOWING SECTION OF THE PROGRAM GENERATES THE SIN**2
C     MANEUVER,BEGINNING AT RMAN AND HAVING A DURATION XPER.

    5 Y(1)=XPER
      Y(2)=S(2,K)
      Y(3)=S(4,K)
      Y(4)=S(5,K)
      B=6.283185308/XPER
      KHOLD=K
    6 COS1=DCOS(B*Y(1))
      SIN1=DSIN(B*Y(1))
      COSSQ=COS1**2
      SINSQ=SIN1**2
      NSIGN=2.0*Y(1)/XPER
      P=(-1)**NSIGN
      F(1)=Y(2)
      F(2)=-4.*(A**2)*(B**3)*Y(2)*V*SIN1*COS1*(1.-2.*COSSQ)/(1.+4.*(A*
     **2)*(B**2)*COSSQ*SINSQ)**1.50
      F(3)=Y(4)
      F(4)=2.0*P*A*B*(F(2)*COS1*SIN1-B*(Y(2)**2)*(1.0-2.*COSSQ))
      K=K+1
      Y(1)=Y(1)+RATE*Y(2)
      B1=Y(3)
      Y(3)=P*A*SINSQ
      Y(4)=(B1-Y(3))/RATE
      S(1,K)=S(1,KHOLD)+Y(1)-XPER
      S(2,K)=Y(2)
      S(3,K)=F(2)
      S(4,K)=Y(3)
      S(5,K)=Y(4)
      S(6,K)=F(4)
      IF(K.GE.NSAM1) GO TO 10
      IF(Y(1).LE.0.0) GO TO 3
      GO TO 6

C     THE FOLLOWING SECTION GENERATES A STRAIGHT SECTION OF TRACK WHIC
C     EXTENDS FROM THE COMPLETION OF THE MANEUVER TO TRKEND.

    8 K=K+1
      M=K-1
      DO 9 I=3,6
    9 S(I,K)=0.0
      S(2,K)=-V
      S(1,K)=S(1,M)+S(2,M)*RATE
      IF(K.GE.NSAM1) GO TO 10
      IF(S(1,K).LE.TRKEND.OR.S(1,K).LE.0) GO TO 10
      GO TO 8

C     THE FOLLOWING SECTION ROTATES THE TRACK THROUGH THE ANGELS BR
C     AND E AS REQUIRED BY THE INPUT DATA

   10 B=BR/57.29577951
      E=E/57.29577951
      COSB=DCOS(B)
      SINB=DSIN(B)
      COSE=DCOS(E)
      SINE=DSIN(E)

      DO 13 KL=1,K
      DO 11 JL=1,6
   11 TEMPL(JL)=S(JL,KL)

      IF(N1.NE.9) GO TO 12
      S(7,KL)=TEMPL(1)*SINE+HO
      S(8,KL)=TEMPL(2)*SINE
      S(9,KL)=TEMPL(3)*SINE
   12 IF(N1.NE.9) COSE=1.0

      R= DSQRT(TEMPL(1)**2+TEMPL(4)**2)
      THETA=DATAN2(TEMPL(4),TEMPL(1))
      PHI=THETA+B
      S(1,KL)=R*DCOS(PHI)*COSE
      S(2,KL)=(TEMPL(2)*COSB-TEMPL(5)*SINB)*COSE
      S(3,KL)=(TEMPL(3)*COSB-TEMPL(6)*SINB)*COSE
      S(4,KL)=R*COSE*DSIN(PHI)+CPA
      S(5,KL)=TEMPL(5)*COSB+TEMPL(2)*SINB*COSE
      S(6,KL)=TEMPL(6)*COSB+TEMPL(3)*SINB*COSE
   13 CONTINUE

C     CALL SUBROUTINE ICOORD IF SPHERICAL COORDINATE DATA IS DESIRED

      IF(ICOORD.EQ.0) RETURN
      CALL COORD (S,NSAM1,K)
      RETURN
```

```
      SUBROUTINE COORD (S,N,NTS)
C
C     THIS SUBROUTINE CONVERTS THE CARTESIAN COORDINATE STATE
C     VALUES TO SPHERICAL COORDINATES..
C
      DOUBLE PRECISION X,Y,Z,R
      DIMENSION S(6,200)
      DO 10 K=1,NTS
      X=S(1,K)
      Y=S(4,K)
      Z=S(7,K)
      VX=S(2,K)
      VY=S(5,K)
      VZ=S(8,K)
      AX=S(3,K)
      AY=S(6,K)
      AZ=S(9,K)
    5 CONTINUE
      R=DSQRT(X**2+Y**2+Z**2)
      B=DATAN2(Y,X)
      E=DASIN(Z/R)
      RDOT=(VX*X+VY*Y+VZ*Z)/R
      XYSQ=X**2+Y**2
      IF(XYSQ.LT.0.0001) XYSQ=0.0001
      BDOT=(X*VY-VX*Y)/XYSQ
      RTRZ=DSQRT(R**2-Z**2)
      IF(RTRZ.LT.0.0001) RTRZ=0.0001
      EDOT=(R*VZ-RNOK*Z)/(RTRZ*R)
      R2DOT=(R*(AX*X+VX**2+AY*Y+VY**2+AZ*Z+VZ**2)-RNOK*(VX*X+VY*Y+VZ*Z))
     */R**2
      B2DOT=(X*AY-AX*Y)/XYSQ-2*((X*VX+Y*VY)*(X*VY-VX*Y))/XYSQ**2
      E2DOT=(R*AZ-R2NCK*Z)/RTRZ-(RNOK*RTRZ**2+R*(RNOK*R-VZ*Z)*(R*VZ-RNOK
     **Z))/(R*RTRZ**3)
      S(1,K)=R
      S(2,K)=RDOT
      S(3,K)=R2DOT
      S(4,K)=B
      S(5,K)=BDOT
      S(6,K)=B2DOT
      IF(N.LT.9) GO TO 10
      S(7,K)=E
      S(8,K)=EDOT
      S(9,K)=E2DOT
   10 CONTINUE
      WRITE(6,15)
   15 FORMAT('1',T30,'OUTPUT IN SPHERICAL COORDINATES',/)
      RETURN
      END

      SUBROUTINE MPLOT1 (IX,IY,X,Y,IN,NL,IDIM)

C     THIS SUBROUTINE PERFORMS THE PLOTTING OF RESULTS.

C     IX   :LENGTH OF X AXIS
C     IY   ;LENGTH OF Y AXIS
C     Y    :Y AXIS
C     X    :X AXIS
C     IN   ;NUMBER OF POINTS
C     NL   :LOGARITMIC INDICATOR
C     IDIM:NUMBER OF GRAPHES

      DIMENSION X(3,200),Y(3,200),U(111),V(200),D(40),E(12),IN(3)
      CHARACTER *1 A(200,111),F(111)*1
      REAL MIN,MIN1,MAX,MAX1,INCX,INCY
      DO 5 I=1,200
      DO 5 J=1,111
      A(I,J)=' '
    5 CONTINUE
      IF(IX.GT.111.OR.IX.LT.2) IX=111
      IF(IY.GT.200.OR.IY.LT.2) IY=200
      IF(NL.EQ.1.OR.NL.EQ.2) THEN
      DO 10 M=1,IDIM
      DO 10 I=1,IN(M)
      IF(X(M,I).NE.0) THEN
      X(M,I)=ALOG10(X(M,I))
      ENDIF
   10 CONTINUE
      END IF
      IF(NL.EQ.1.OR.NL.EQ.3) THEN
      DO 15 M=1,IDIM
      DO 15 I=1,IN(M)
      IF(Y(M,I).NE.0) THEN
      Y(M,I)=ALOG10(Y(M,I))
      ENDIF
   15 CONTINUE
      END IF
      DO 20 I=1,IX
      F(I)='-'
   20 CONTINUE
```

```
      F(I)='+'
   25 CONTINUE
      MIN=X(1,1)
      MAX=X(1,1)
      MIN1=Y(1,1)
      MAX1=Y(1,1)
      DO 35 M=1,IDIM
      DO 30 I=1,IN(M)
      IF(X(M,I).GT.MAX) MAX=X(M,I)
      IF(X(M,I).LT.MIN) MIN=X(M,I)
      IF(Y(M,I).GT.MAX1) MAX1=Y(M,I)
      IF(Y(M,I).LT.MIN1) MIN1=Y(M,I)
   30 CONTINUE
   35 CONTINUE
      INCX=(MAX-MIN)/(IX-1)
      INCY=(MAX1-MIN1)/(IY-1)
      U(1)=MIN
      V(1)=MIN1
      DO 40 I=1,IX-1
      U(I+1)=U(I)+INCX
   40 CONTINUE
      DO 45 I=1,IY-1
      V(I+1)=V(I)+INCY
   45 CONTINUE
      DO 65 M=1,IDIM
      DO 60 I=1,IN(M)
      M1=1
      DIFF=ABS(X(M,I)-U(1))
      DO 50 J=1,IX-1
      IF(ABS(X(M,I)-U(J+1)).LT.DIFF) THEN
      DIFF=ABS(X(M,I)-U(J+1))
      M1=J+1
      END IF
   50 CONTINUE
      L=1
      DIFF1=ABS(Y(M,I)-V(1))
      DO 55 J=1,IY-1
      IF(ABS(Y(M,I)-V(J+1)).LT.DIFF1) THEN
      DIFF1=ABS(Y(M,I)-V(J+1))
      L=J+1
      END IF
   55 CONTINUE
      IF(M.EQ.1) A(L,M1)='+'
      IF(M.EQ.2) A(L,M1)='*'
      IF(M.EQ.3) A(L,M1)='.'
   60 CONTINUE
   65 CONTINUE
      L2=0
      L3=0
      DO 70 I=1,IY,5
      L2=L2+1
      IF(NL.EQ.1.OR.NL.EQ.3) D(L2)=10**V(I)
      IF(NL.NE.1.AND.NL.NE.3) D(L2)=V(I)
   70 CONTINUE
      DO 75 I=1,IX,10
      L3=L3+1
      IF(NL.EQ.1.OR.NL.EQ.2) E(L3)=10**U(I)
      IF(NL.NE.1.AND.NL.NE.2) E(L3)=U(I)
   75 CONTINUE
      IF(NL.EQ.1.OR.NL.EQ.2) THEN
      DO 80 M=1,IDIM
      DO 80 I=1,IN(M)
      X(M,I)=10**X(M,I)
   80 CONTINUE
      END IF
      IF(NL.EQ.1.OR.NL.EQ.3) THEN
      DO 85 M=1,IDIM
      DO 85 I=1,IN(M)
      Y(M,I)=10**Y(M,I)
   85 CONTINUE
      END IF
      K7=((IY-1)/5)*5+1
      IP=(IX-1)/10+1
      PRINT 95
      PRINT 115, (E(I),I=1,IP)
      PRINT 110, (F(I),I=1,IX)
      MK=IY/5+2
      DO 90 I=IY,1,-1
      IF(I.EQ.K7) THEN
      K7=K7-5
      L=I/5+1
      L=MK-L
      PRINT 100, D(L),(A(I,J),J=1,IX)
      GO TO 90
      END IF
      PRINT 105, (A(I,J),J=1,IX)
   90 CONTINUE
      RETURN
   95 FORMAT(1H1,/)
```

```
100 FORMAT(3X,1PE10.2,1X,'+',111A1)
105 FORMAT(14X,'I',111A1)
110 FORMAT(15X,111A1,/)
115 FORMAT(11X,11(1PE9.2,1X),1PE9.2,/)
    END

    SUBROUTINE RANCOM(IX,IY,RND)

    IY=IX*1220703125
    IF(IY) 3,4,4
  3 IY=IY+2147483647+1
  4 RND=IY
    RND=RND*0.4656613E-9
    IX=IY
    RETURN
    END
```

APPENDIX B

COMPUTATIONAL REQUIREMENTS

Kalman filtering is a popular method of estimating states from noisy measurements due to the ease of implementation on a digital computer . Kalman filter stability, convergence and sensitivity properties have been thoroughly examined in the literature . In chapter V we presented certain modifications which can be incorporated in the conventional filter algorithm in order to reduce the computational burden . In previous discussions, a computationally efficient algorithm was of the primary concern, even though the term 'computational requirements' was not explained. In [12] Mendel gives a clear definition :

Computational requirements : Computation time per iteration and Necessary memory size for storage

In his paper Mendel gives computational requirements as explicit functions of the dimensions of the system's state, measurement and disturbance vectors within the constraints of an assumed computer configuration .

Basic operations required for programming the Kalman filter equations are matrix addition, matrix subtraction, matrix multiplication, matrix transpose multiplication, multiplication by a scalar and matrix inversion, which has been eliminated by the sequential processing property of our program .

We shall make use of the results obtained in [12] to determine the number of operations to estimate the system's state for one iteration .

The calculation of execution time required for processing one set of measurements was deliberately omitted because the Monte-Carlo simulation program in our work is intended not only for the implementation of Kalman filter but also for the analysis of results .

The present discussion, however, gives a criterion for the evaluation of computational efficiency of a particular filter design for real-time applications .

TABLE I

DISCRETE KALMAN FILTER COMPUTATION TIME REQUIREMENTS

Variable	Defining Equation[b]	Computations	Number of Multiplications	Number of Additions	Logic Time
$\hat{x}(k+1\|k)$	$\Phi(k+1,k)\hat{x}(k\|k)+\Psi(k+1,k)u(k)$	$\Phi\hat{x}$	n^2	n^2-n	$10+6n^2+37n$
		Ψu	n	0	$8n$
		$\Phi\hat{x}+\Psi u$	0	n	MUL + 27 + $5n$
$\Sigma(k+1\|k)$	$\Phi(k+1,k)\Sigma(k\|k)\Phi'(k+1,k)$ $+\Gamma(k+1,k)\Xi(k)\Gamma'(k+1,k)$	$\Sigma\Phi'$	n^3	n^3-n^2	$10+6n^3+21n^2+16n$
		$\Phi(\Sigma\Phi')$	n^3	n^3-n^2	$10+6n^3+21n^2+16n$
		$\Xi\Gamma'$	ns^2	ns^2-ns	$10+6ns^2+21ns+16s$
		$\Gamma(\Xi\Gamma')$	n^2s	n^2s-n^2	$10+6n^2s+21n^2+16n$
		$\Phi(\Sigma\Phi')+\Gamma(\Xi\Gamma')$	0	n^2	MUL + 27 + $5n^2$
$K(k+1)$	$\Sigma(k+1\|k)C'(k+1)$ $\cdot[C(k+1)\Sigma(k+1\|k)C'(k+1)$ $+\Theta(k+1)]^{-1}$	$\Sigma C'$	n^2r	n^2r-nr	$10+6n^2r+21nr+16n$
		$C(\Sigma C')$	nr^2	nr^2-r^2	$10+6nr^2+21r^2+16r$
		$C(\Sigma C')+\Theta$	0	r^2	MUL + 27 + $5r^2$
		$[C(\Sigma C')+\Theta]^{-1}$	r^3	r^3	$10+7.5r^3+41r^2+139.5r$ + 92 + DIV(r + $2r^2$) + MUL(2.5r + $0.5r^2$)
		$\Sigma C'[C(\Sigma C')+\Theta]^{-1}$	nr^2	nr^2-nr	$10+6nr^2+21nr+16n$
$\hat{x}(k+1\|k+1)$	$\hat{x}(k+1\|k)+K(k+1)$ $\cdot[z(k+1)-C(k+1)\hat{x}(k+1\|k)]$	$C\hat{x}$	nr	$nr-r$	$10+6nr+37r$
		$z-(C\hat{x})$	0	r	MUL + 27 + $5r$
		$K[z-(C\hat{x})]$	nr	$nr-n$	$10+6nr+37n$
		$\hat{x}+K[z-(C\hat{x})]$	0	n	MUL + 27 + $5n$
[a] $\Sigma(k+1\|k+1)$	$[I-K(k+1)C(k+1)]\Sigma(k+1\|k)$	KC	n^2r	n^2r-n^2	$10+6n^2r+21n^2+16n$
		$I-(KC)$	0	n^2	MUL + 27 + $5n^2$
		$[I-(KC)]\Sigma$	n^3	n^3-n^2	$10+6n^3+21n^2+16n$

[a] $\Sigma(k+1|k+1)$ is symmetric and positive definite. If it is difficult to maintain these properties using the defining equation, the following alternate, symmetric form can be used:

$$\Sigma(k+1|k+1)=[I-K(k+1)C(k+1)]\Sigma(k+1|k)[I-K(k+1)C(k+1)]'+K(k+1)\Theta(k+1)K'(k+1)$$

Computational time will increase. For example, the total numbers of multiplications and additions required to obtain $\Sigma(k+1|k+1)$ increase by n^3+n^2r+nr and $n^3+n^2r-n^2$, respectively.

[b] All variables which appear in these equations are defined in Table II

TABLE II

FILTER VARIABLES AND STORAGE REQUIREMENTS

Variable	Definition	Dimension	Storage Requirement
$\hat{x}(k\|k)$	State estimate at t_k given $z(k)$	$n\times 1$	n
$\Sigma(k\|k)$	Covariance matrix of the error in $\hat{x}(k\|k)$	$n\times n$	n^2
$\Phi(k+1,k)$	State transition matrix (from t_k to t_{k+1})	$n\times n$	n^2
$\Gamma(k+1,k)$	System disturbance distribution matrix	$n\times s$	ns
$\Xi(k)$	System disturbance covariance matrix	$s\times s$	s^2
$\hat{x}(k+1\|k)$	State estimate at t_{k+1} given z_k	$n\times 1$	n
$\Sigma(k+1\|k)$	Covariance matrix of the error in $\hat{x}(k+1\|k)$	$n\times n$	n^2
$C(k+1)$	Measurement matrix	$r\times n$	nr
$\Theta(k+1)$	Measurement noise covariance matrix	$r\times r$	r^2
$K(k+1)$	Filter (Kalman) gain matrix at t_{k+1}	$n\times r$	nr
$z(k+1)$	Measurement (observation) at t_{k+1}	$r\times 1$	r
$\Psi(k+1,k)$	Control distribution vector	$n\times 1$	n
$u(k)$	Control at t_k	Scalar[a]	1

Making use of the table I with appropriate substitutions for our system yields the following results :

number of multiplications
required in filter computations (for one channel) = 120

number of additions
required in filter computations (for one channel) = 91

number of multiplications
required in maneuver detection (for one channel) = 1

number of additions
required in maneuver detection (for one channel) = 9

It has to be noted that the algorithm involved in the simulations was based on the assumption that the system was made up of three independent channels with two filters, A and B operating simultaneously.

The estimate of system's state in three-dimensional modeling therefore requires

$$121 \times 2 \times 3 = 726 \text{ multiplications}$$

and

$$101 \times 2 \times 3 = 606 \text{ additions .}$$

APPENDIX C

PREDICTION ACCURACY CALCULATION

The ultimate product of a filter which is designed for target tracking applications is the prediction of future target position for an automatic gun fire control system (or, AGFCS for short). The accuracy of the predictions are of primary concern since the gun is aimed at the predicted target position which can be extrapolated from each sample estimate. Therefore, the estimation errors might either serve to multiply the prediction errors, or these errors cancel each other out.

A relatively simple procedure for measuring the prediction accuracy over realistic time intervals is described in [9]. We reproduce the same procedure here for reference purposes.

The time interval over which the predictions are made is based on a straight line approximation of the time of flight(TOF) versus range curve for a 5"/54 caliber projectile as indicated in fig. 1.

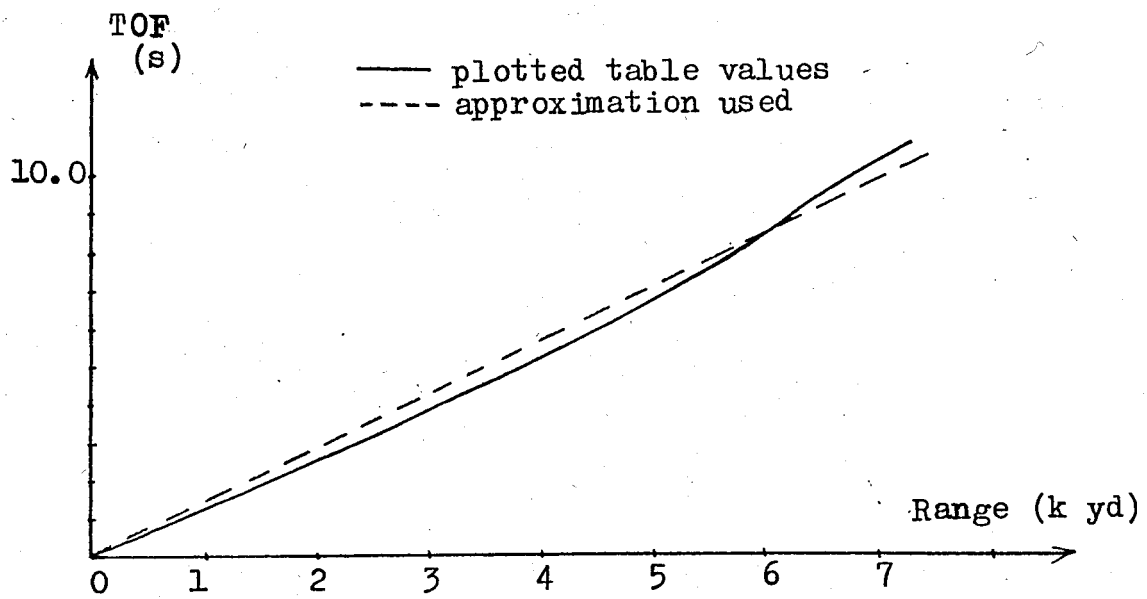


fig. 1 .Plot of time of flight vs. range for a 5"/54 cal. projectile

Measurement of the slope of the straight line approximation in fig. 1 shows that

$$TOF = 1.43 \times 10^{-3} (s/yd) \quad Range(yd)$$

Taking the maximum effective range of the gun against an air target into consideration, the time interval over which predictions are made is limited to a maximum of ten seconds . This corresponds to a maximum effective range of approximately 7000 yards .

A further approximation is made in order to simplify the matrix manipulations required . The TOF is truncated to an integer multiple of the sample period . The prediction is based on the equation ;

$$\underline{\hat{x}}(k/k-N) = \underline{\underline{\Phi}}^{N}\, \underline{\hat{x}}(k-N/k-N) \tag{1}$$

where N is the integer number of sample periods over which the prediction is made .

The prediction miss distance is calculated by finding the square root of the sum of the squares of the difference between the predicted position states and the respective true position states , i. e.

$$\begin{aligned}(\text{miss distance})^2 = &\left[\hat{x}_1(k/k-N) - x_1(k)\right]^2 \\ &+ \left[\hat{x}_2(k/k-N) - x_2(k)\right]^2 \\ &+ \left[\hat{x}_3(k/k-N) - x_3(k)\right]^2\end{aligned} \tag{2}$$

The N-step STM is given by

$$\underline{\underline{\Phi}}^{N} = \begin{bmatrix} \underline{\underline{\Phi}}_i & \underline{\underline{0}} & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{\Phi}}_i & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{0}} & \underline{\underline{\Phi}}_i \end{bmatrix} \tag{3}$$

where

$$\underline{\underline{\Phi}}_i = \begin{bmatrix} 1.0 & NT & \Phi_{13} \sum_{n-1}^{N} \Phi_{33}^{n-1} + T\Phi_{23} \sum_{n-1}^{N-1} [(N-n)]\, \Phi_{33}^{n-1} \\ 0 & 1.0 & \Phi_{23} \sum_{n-1}^{N} \Phi_{33}^{n-1} \\ 0 & 0 & \Phi_{33}^{N} \end{bmatrix} \tag{4}$$

In equation (4), the matrix elements Φ_{ij} are the same as the ones which we have described in chapter II .

It shouldbe noted that equation (1) is not evaluated by raising the $n \times n$ matrix Φ to the N^{th} power by direct matrix algebra, which would be very time consuming, but is calculated by operating on the individual components of Φ .

Equation (4) is therefore derived by self-multiplying the matrix of equation (2.13) and establishing a pattern .

APPENDIX D

A TARGET TRAJECTORY GENERATING PROGRAM

D.1 Description of the Program

In order to test filter's performance against maneuvering targets it is desirable to have a realistic trajectory for which position, velocity and acceleration states are known at each sample point .

The maneuver decided upon is a periodic, planar maneuver described by the equations ;

$$y(x,t) = pA\sin^2(kx) \tag{1}$$

where

$$p = (-1)^n, \qquad n = \begin{cases} 0, & kx < \pi \\ 1, & kx \geq \pi \end{cases}$$

and

$$k = \frac{2\pi}{\left[\begin{array}{l}\text{the distance traveled along the} \\ \text{x-axis during the maneuver}\end{array}\right]}$$

$$\dot{y}(x,\dot{x},t) = 2pAk\dot{x}\cos(kx)\sin(kx) \tag{2}$$

and

$$\ddot{y}(x,\dot{x},\ddot{x},t) = 2pAk\{\ddot{x}\cos(kx)\sin(kx)+k\dot{x}^2[2\cos^2(kx)-1]\} \tag{3}$$

The form of maneuver is shown in fig.1 . The variables $x, \dot{x}$, and $\ddot{x}$ have been used to represent $x(t)$, $\dot{x}(t)$, and $\ddot{x}(t)$ respectively in equations (1) through (14) to make the notation less awkward . The two dimensional geometry of the maneuver yields the relationship ;

$$v^2 = \dot{x}^2 + \dot{y}^2$$

where v, the speed of the aircraft tangent to the x-y plane is

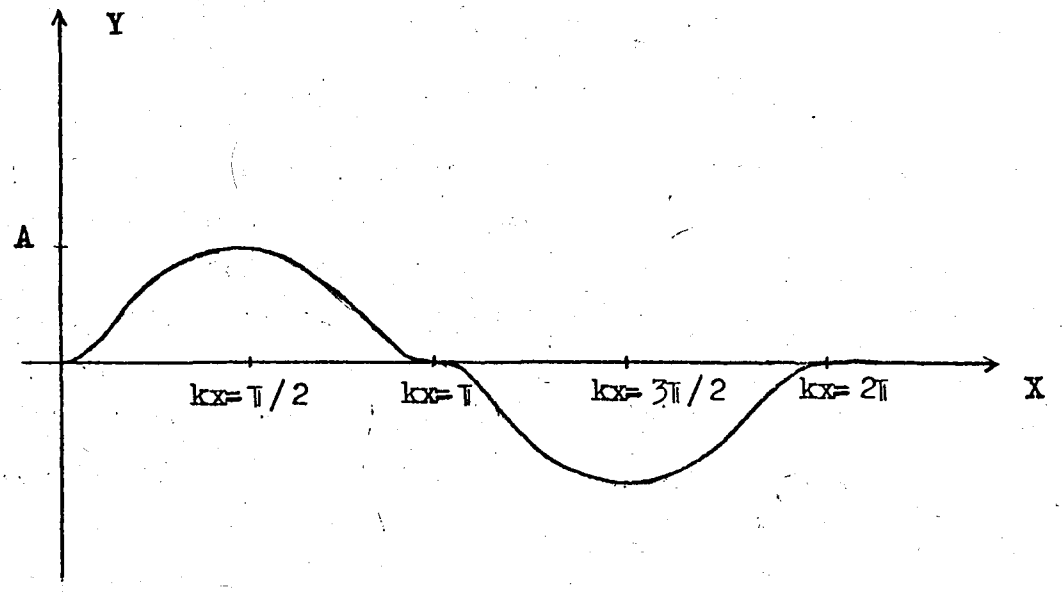


fig. 1. The Planar Maneuver

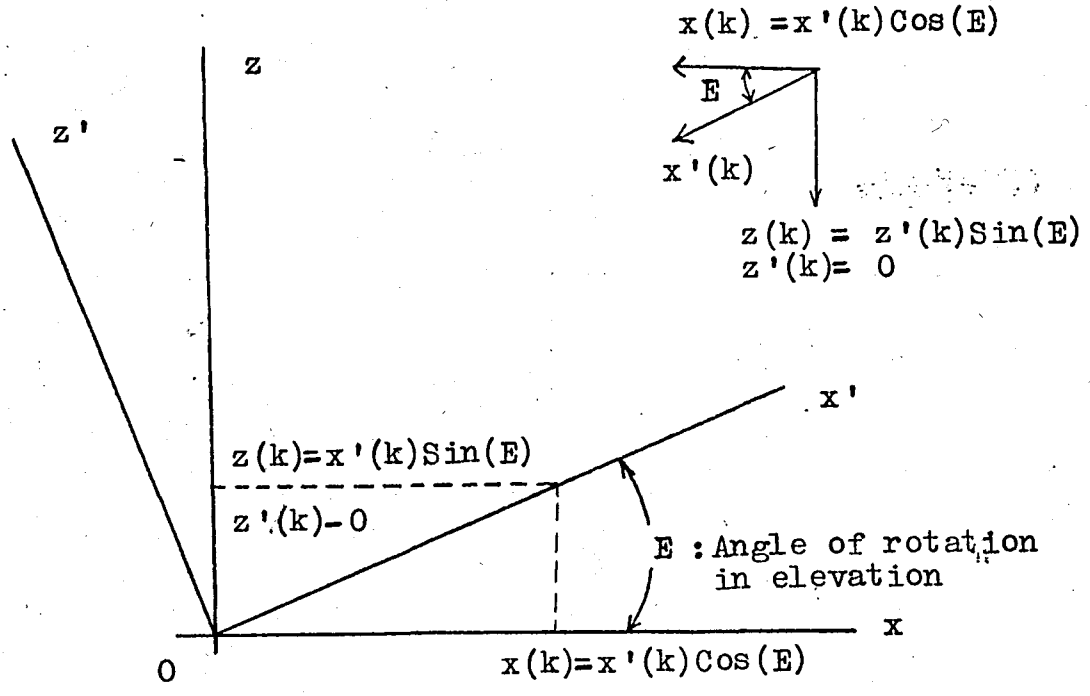


fig. 2. Rotation of Track in elevation

assumed to be constant .

Transposing the terms of equation (4), substituting the right side of equation (2) for y, and taking the square root of both sides yields ;

$$\dot{x} = \frac{v}{\left[1 - 4A^2k^2\cos^2(kx)\sin^2(kx)\right]^{1/2}} \tag{5}$$

Equation (5) is differentiated with respect to time with the result ;

$$\ddot{x} = \frac{4A^2k^2v\dot{x}\sin(kx)\cos(kx)\left[1 - 2\cos^2(kx)\right]}{\left[1 - 4A^2k^2\cos^2(kx)\sin^2(kx)\right]^{3/2}} \tag{6}$$

If the system states are defined as

$$x_1 = x(t) \tag{7}$$
$$x_2 = \dot{x}(t) \tag{8}$$
$$x_3 = y(x,t) \tag{9}$$
$$x_4 = \dot{y}(x,\dot{x},t) \tag{10}$$

the resulting nonlinear state equations are ;

$$\dot{x}_1 = x_2 \tag{11}$$
$$\dot{x}_2 = \frac{4A^2k^2vx_2\sin(kx_1)\cos(kx_1)\left[1 - 2\cos^2(kx_1)\right]}{\left[1 + 4A^2k^2\cos^2(kx_1)\sin^2(kx_1)\right]^{3/2}} \tag{12}$$
$$\dot{x}_3 = x_4 \tag{13}$$
$$\dot{x}_4 = 2pAk\left\{\dot{x}_2\cos(kx_1)\sin(kx_1) - kx_2^2\left[1 - 2\cos^2(kx_1)\right]\right\} \tag{14}$$

The state equations are used in a computer program to make up the maneuvering portion of the trajectory .

In order that filter's performance against a target which maneuvers after the filter gains have reached steady-state can be studied, provision is made for a constant velocity section of target trajectory preceding the maneuver . A constant-velocity

section of track following the maneuvers allows the filter's recovery time to be examined .

Although the trajectory is originally generated with the x-axis as its axis of symmetry, it may be rotated to any bearing angle and any first quadrant elevation angle . The geometry involved in the rotation of the axis of symmetry is illustrated in fig. 2 through fig. 4 . As indicated in fig. 2 for the elevation angle rotation, a new coordinate system is established which places the axis of symmetry , now labeled the x'-axis, at the desired elevation angle, E.

The x and z coordinate states can be calculated from the x' coordinate states as ;

$$x(k) = x'(k)\mathrm{Cos}(E) \tag{15}$$

$$\dot{x}(k) = \dot{x}'(k)\mathrm{Cos}(E) \tag{16}$$

$$\ddot{x}(k) = \ddot{x}'(k)\mathrm{Cos}(E) \tag{17}$$

$$z(k) = x'(k)\mathrm{Sin}(E) \tag{18}$$

$$\dot{z}(k) = \dot{x}'(k)\mathrm{Sin}(E) \tag{19}$$

$$\ddot{z}(k) = \ddot{x}'(k)\mathrm{Sin}(E) \tag{20}$$

The y-coordinate states are not affected by the elevation angle rotation since the rotation is about the y-axis . The bearing angle rotation is depicted in fig. 3 -- the case shown is for a zero elevation angle . The axis of symmetry is rotated through the angle B, which results in a point on the trajectory curve being displaced from the x-axis by an angle of $B + \theta$. Fig. 3 shows the relationship between the x and y-direction velocities before and after rotation . The z-coordinate states are not affected by the bearing rotation since the rotation is about the z-axis . The values of x and y-coordinate states after rotation of the trajectory's axis of symmetry through elevation angle E and bearing angle B are given by :

$$x(k) = R(k)\mathrm{Cos}(B - \theta(k))\mathrm{Cos}(E) \tag{21}$$

$$\dot{x}(k) = \left[\dot{x}'(k)\mathrm{Cos}(B) - \dot{y}'(k)\mathrm{Sin}(B)\right]\mathrm{Cos}(E) \tag{22}$$

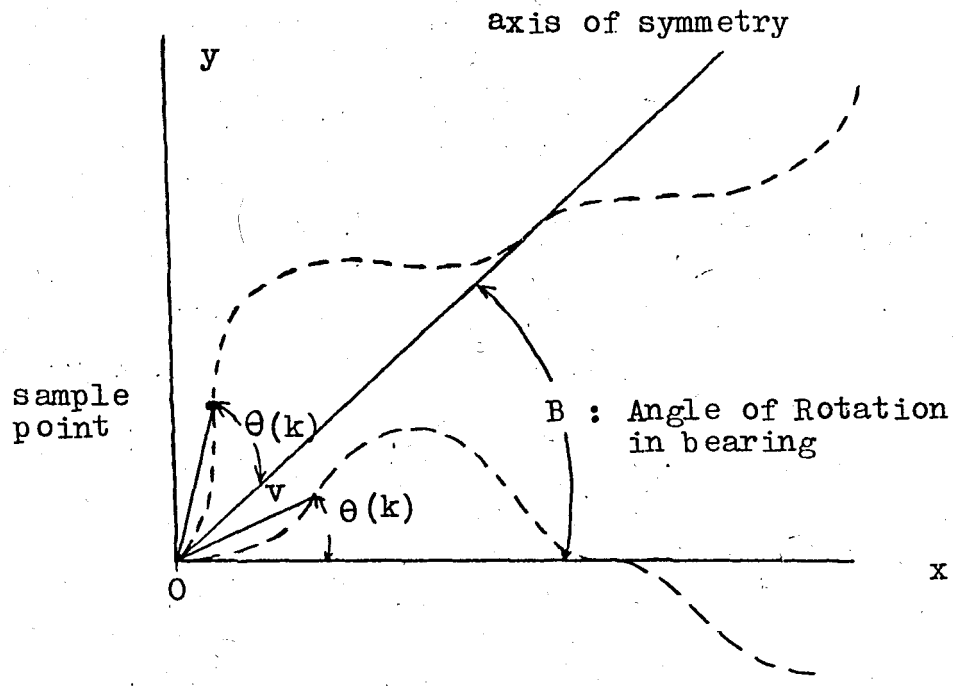
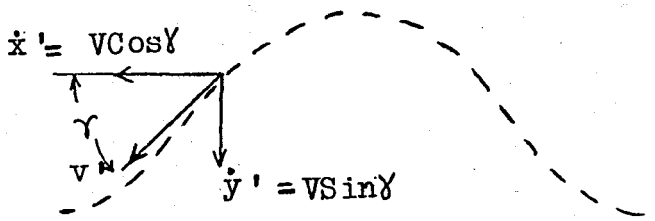
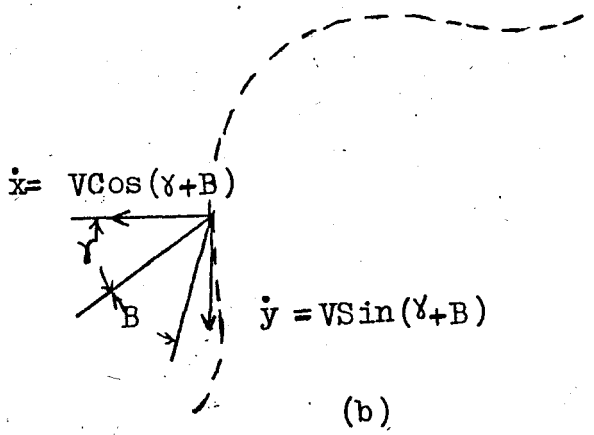


fig. 3. Rotation of track in bearing



(a)



(b)

fig. 4. Velocity vector before (a) and after (b) rotation through angle B

$$\ddot{x}(k) = \left[\ddot{x}'(k)\mathrm{Cos}(B) - \ddot{y}'(k)\mathrm{Sin}(B)\right]\mathrm{Cos}(E) \tag{23}$$

$$y(k) = R(k)\mathrm{Sin}[B + \theta(k)]\mathrm{Cos}(E) \tag{24}$$

$$\dot{y}(k) = \dot{x}'(k)\mathrm{Cos}(E)\mathrm{Sin}(B) + \dot{y}'(k)\mathrm{Cos}(B) \tag{25}$$

$$\ddot{y}(k) = \ddot{x}'(k)\mathrm{Cos}(E)\mathrm{Sin}(B) + \ddot{y}'\mathrm{Cos}(B) \tag{26}$$

where

$$R(k) = \left[x'(k)^2 + y'(k)^2\right]^{1/2}$$
$$= \left[x(k)^2 + y(k)^2 + z(k)^2\right]^{1/2} \tag{27}$$

and

$$\theta(k) = \arctan[y'(k)/x'(k)] \tag{28}$$

Equation (22) and (25) are derived from the trigonometric relationship shown in fig. 3 . Equations (23) and (26) are derived from a similar relationship between the acceleration states .

Provision is made for the trajectory to have nonzero intercepts with the y- and/or z-axis . This is accomplished simply by adding any desired value to the y- and/or z-axis position states at each sample point .

A subroutine is provided which performs conversion of all state values from Cartesian to spherical coordinate quantities when required . The spherical coordinate data is in units of yards in range and radians in bearing and elevation .

D. 2 Relationship Between Velocity, Period of Maneuver, Amplitude of Maneuver, and Acceleration

The magnitude of total acceleration for the x-y plane maneuver is given by

$$a = \left[\ddot{x}^2 + \ddot{y}^2\right]^{1/2} \tag{29}$$

where x and y are given by equations (6) and (3) respectively. Making the subtitions as indicated in equation (29) and solving for a yields

$$a = \frac{2Ak^2v^2\left[1 - 2\mathrm{Cos}^2(k)\right]}{1 - A^2k^2\mathrm{Cos}^2(kx)\mathrm{Sin}^2(kx)} \tag{30}$$

which can be further simplified by utilizing the trigonometric relationship

$$\mathrm{Cos}2x = 2\mathrm{Cos}^2x - 1 \tag{31}$$

and

$$\mathrm{Sin}2x = 2\mathrm{Sin}x\mathrm{Cos}x \tag{32}$$

with the result

$$a = \frac{-2Ak^2v^2\mathrm{Cos}(2kx)}{1 + A^2k^2\mathrm{Sin}^2(2kx)} \tag{33}$$

To find the point, with respect to x-position, at which the max. acceleration occurs during the maneuver it is necessary to differentiate the equation (33) with respect to x and set the result equal to zero. This yields

$$4A^2k^2v^2\mathrm{Sin}(2kx)\left[1 + 4Ak^2\mathrm{Sin}^2(2kx) - Ak^2\mathrm{Cos}^2(2kx)\right] = 0 \tag{34}$$

which is true when

$$\mathrm{Sin}(2kx) = 0,$$

$$\text{i.e. } 2kx = n\pi, \quad n = 0, 1, 2, \ldots$$

Substitution of $2kx = n\pi$ into equation (28) yields

$$|a|_{max} = 2Ak^2v^2 \tag{35}$$

Equaiton (35) gives the relationship between target speed, maneuver amplitude, maneuver period, and the maximum acceleration required to complete the maneuver. This relationship is useful as an aid in choosing values for the variables involved.

D. 3 Examples of Track Generating Program Use

Example 1. A trajectory is to be generated which approaches the originfrom a range of 3500 yards with a 20 degree dive angle at a constant speed of 200 yd/s. A 2.5 g maneuver of approximately

ten seconds duration is to commence when the target is at a range of 2700 yards . Sixty samples are to be generated at a four hertz sampling rate .

The first task is to insert the track specifications into the data required by the track generating program .

A maneuver duration of ten seconds is specified . To calculate the distance travelled along the x-axis of symmetry during a 2.5 g , ten second, Sin^2 maneuver would not be a simple task . However experience has shown that the change in range rate during such a maneuver is relatively small, therefore the distance is approximately ,

$$x_{per} = (10s)(200yd/s) = 2000yd. \qquad (37)$$

Equation (36) can now be used to solve for the amplitude of the maneuver.

$$A = \frac{2.5(g)\quad 10.73(yd/s.g)}{2\left[\frac{2}{2000(yd)}\right]^2 \left[\,200(yd/s)\,\right]^2}$$

$$= 33.97 \text{ yd.} \qquad (38)$$

The four hertz sampling rate inverts to a 0.25 s sampling period .

The data is now available in the proper form to make up the data input cards for the track generating program . The cards must contain the following information .

CARD 1 :

S(1,1) = initial range . . . . . = 3500.0
S(2,1) = initial range rate . = -200.0

CARD 2 :

RATE	=	sampling period	=	0.25
V	=	target speed	=	200.0
A	=	maneuver amplitude	=	33.97
XPER	=	maneuver duration	=	2000.0

BR	=	bearing angle	=	0.0
E	=	elevation angle	=	20.0
RMAN	=	maneuver commencement range	=	2700.0
TRKEND	=	trajectory end range (the track is to be ended by the sample count)	=	-9999.9

CARD 3 :

CPA	=	y-direction displacement	=	0.0
HO	=	z-direction displacement	=	0.0

CARD 4 :

NSAM	=	number of samples desired	=	60
N	=	number of states generated	=	9
ICOORD	=	coordinate system indicator= (data is to be in Cartesian coordinates)		0

Table D.3.1 lists the data generated by the track generating program for this example .

Example 2. A level, constant velocity trajectory beginning at the point x = 5000 yards, y = 1000 yards and z = 333.3 yards in Cartesian coordinates and proceeding in the negative x-direction with a speed of 200 yds/s until the x-direction displacement is 2800 yards is to be produced . The sampling period is 0.25 s . The output data is to be in spherical coordinates .

This trajectory can be generated quite simply by utilizing the CPA and HO features of the program . Since the trajectory is generated initially with the x-axis as its axis of symmetry and the y and z-direction displacements are added afterward, the specified values of initial x-position and velocity are taken as the initial range and range rate respectively . The other required input values are given directly in the track specifications .

CARD 1 :

S(1,1)	=	initial range (x-pos.)	=	5000.0
S(2,1)	=	initial range rate (x-vel.)	=	-200.0

CARD 2 :

RATE	=	sampling period	=	0.25

V	=	target speed	=	200.0
A	=	maneuver amplitude	=	0.0
XPER	=	maneuver duration	=	0.0
BR	=	bearing angle	=	0.0
E	=	elevation angle	=	0.0
RMAN	=	maneuver commencement rg.	=	0.0
TRKEND	=	trajectory end range	=	2800.0

CARD 3 :

CPA	=	y-direction displacement	=	1000.0
HO	=	z-direction displacement	=	333.3

CARD 4 :

NSAM	=	number of samples	=	200
N	=	number of states generated	=	9
ICOORD	=	coordinate system indicator (spherical coordinate data)	=	1

Table D. 3. 2. lists the data generated by the track generating program for this example .

X-POSITION	X-VELOCITY	X-ACCELER	Y-POSITION	Y-VELOCITY	Y-ACCELER	Z-POSITION	Z-VELOCITY	Z-ACCELER	
3289.	-187.9	0.0	0.0	0.0	0.0	1197.	-68.40	0.0	1
3242.	-187.9	0.0	0.0	0.0	0.0	1180.	-68.40	0.0	2
3195.	-187.9	0.0	0.0	0.0	0.0	1163.	-68.40	0.0	3
3148.	-187.9	0.0	0.0	0.0	0.0	1146.	-68.40	0.0	4
3101.	-187.9	0.0	0.0	0.0	0.0	1129.	-68.40	0.0	5
3054.	-187.9	0.0	0.0	0.0	0.0	1112.	-68.40	0.0	6
3007.	-187.9	0.0	0.0	0.0	0.0	1094.	-68.40	0.0	7
2960.	-187.9	0.0	0.0	0.0	0.0	1077.	-68.40	0.0	8
2913.	-187.9	0.0	0.0	0.0	0.0	1060.	-68.40	0.0	9
2866.	-187.9	0.0	0.0	0.0	0.0	1043.	-68.40	0.0	10
2819.	-187.9	0.0	0.0	0.0	0.0	1026.	-68.40	0.0	11
2772.	-187.9	0.0	0.0	0.0	0.0	1009.	-68.40	0.0	12
2725.	-187.9	0.0	0.0	0.0	0.0	991.9	-68.40	0.0	13
2678.	-187.9	0.0	0.0	0.0	0.0	974.8	-68.40	0.0	14
2631.	-187.9	0.0	0.0	0.0	0.0	957.7	-68.40	0.0	15
2584.	-187.9	0.0	0.0	0.0	0.0	940.6	-68.40	0.0	16
2537.	-187.9	0.0	0.0	0.0	0.0	923.5	-68.40	0.0	17
2490.	-187.8	0.7887	-0.7599	-6.501	-25.45	906.4	-68.37	0.2871	18
2443.	-187.6	1.269	-3.002	-12.42	-21.54	889.3	-68.27	0.4618	19
2396.	-187.2	1.261	-6.499	-17.10	-15.56	872.2	-68.15	0.4591	20
2350.	-187.0	0.7803	-10.90	-20.09	-8.188	855.2	-68.06	0.2840	21
2303.	-186.9	0.1167D-01	-15.78	-21.13	-0.1164	838.2	-68.02	0.4247D-02	22
2256.	-187.0	-0.7612	-20.67	-20.15	7.965	821.2	-68.05	-0.2771	23
2209.	-187.2	-1.254	-25.10	-17.21	15.37	804.2	-68.15	-0.4563	24
2162.	-187.6	-1.276	-28.63	-12.58	21.39	787.1	-68.27	-0.4643	25
2116.	-187.8	-0.8078	-30.91	-6.682	25.38	770.0	-68.36	-0.2940	26
2069.	-187.9	-0.2401D-01	-31.71	-0.1011	25.82	752.9	-68.40	-0.4358D-02	27
2022.	-187.8	0.7693	-30.95	6.499	25.53	735.8	-68.37	0.2800	28
1975.	-187.6	1.261	-28.71	12.45	21.68	718.8	-68.27	0.4591	29
1928.	-187.3	1.269	-25.20	17.17	15.75	701.7	-68.15	0.4617	30
1881.	-187.0	0.7991	-20.77	20.21	8.411	684.7	-68.06	0.2909	31
1834.	-186.9	0.3501D-01	-15.86	21.31	0.3491	667.7	-68.02	0.1374D-01	32
1788.	-187.0	-0.7419	-10.92	20.36	-7.742	650.7	-68.05	-0.2700	33
1741.	-187.2	-1.246	-6.436	17.49	-15.17	633.6	-68.14	-0.4534	34
1694.	-187.6	-1.282	-2.835	12.91	-21.25	616.6	-68.26	-0.4667	35
1647.	-187.8	-0.8767	-0.4722	7.041	-25.30	599.5	-68.36	-0.3009	36
1600.	-187.9	-0.4801D-01	0.4176	0.4705	-26.82	582.4	-68.40	-0.1747D-01	37
1553.	-187.8	0.7496	1.167	6.454	-25.60	565.3	-68.37	0.2728	38
1506.	-187.6	1.254	3.404	12.43	-21.82	548.2	-68.28	0.4563	39
1459.	-187.3	1.275	6.912	17.19	-15.55	531.2	-68.16	0.4642	40
1413.	-187.0	0.8177	11.35	20.28	-6.633	514.1	-68.06	0.2976	41
1366.	-186.9	0.5833D-01	16.29	21.44	0.5818	497.1	-68.02	0.2123D-01	42
1319.	-187.0	-0.7224	21.26	20.57	7.518	480.1	-68.05	-0.2629	43
1272.	-187.2	-1.237	25.80	17.74	14.58	463.1	-68.14	-0.4503	44
1226.	-187.5	-1.288	29.46	13.19	21.10	446.1	-68.26	-0.4669	45
1179.	-187.8	-0.8453	31.90	7.357	25.21	429.0	-68.36	-0.3077	46
1132.	-187.9	-0.7200D-01	32.86	0.7972	26.81	411.9	-68.40	-0.2620D-01	47
1085.	-187.9	0.7297	32.27	-5.820	25.68	394.8	-68.37	0.2656	48
1038.	-187.6	1.246	30.18	-11.82	21.96	377.7	-68.28	0.4533	49
990.9	-187.3	1.282	26.81	-16.64	16.14	360.6	-68.16	0.4666	50
944.1	-187.0	0.8360	22.50	-19.77	8.854	343.6	-68.06	0.3043	51
897.4	-186.9	0.8164D-01	17.67	-20.99	-0.8145	326.6	-68.02	0.2971D-01	52
850.6	-187.0	-0.7026	12.80	-20.17	7.293	309.6	-68.05	-0.2557	53
803.9	-187.2	-1.228	8.347	-17.39	14.78	292.6	-68.14	-0.4471	54
757.0	-187.5	-1.294	4.759	-12.89	20.95	275.5	-68.26	-0.4710	55
710.1	-187.8	-0.8637	2.392	-7.084	25.13	258.5	-68.36	-0.3143	56
663.1	-187.9	-0.9596D-01	1.489	-0.5363	26.80	241.4	-68.40	-0.3493D-01	57
616.1	-187.9	0.7095	2.146	6.089	25.74	224.3	-68.37	0.2583	58
569.2	-187.9	0.0	0.0	0.0	0.0	207.2	-68.40	0.0	59
522.2	-187.9	0.0	0.0	0.0	0.0	190.1	-68.40	0.0	60

TABLE I TRACK GENERATING PROGRAM OUTPUT FOR EXAMPLE 2

OUTPUT DATA IN SPHERICAL COORDINATES

RANGE	RDOT	R2DOT	BEARING	BDOT	B2DOT	ELEVATION	EDOT	E2DOT	
5110.	-195.7	0.3331	0.1974	0.7892D-02	0.5917E-03	0.6522D-01	0.2504D-02	0.4703	1
5061.	-195.6	0.3429	0.1993	0.7842D-02	0.6089E-03	0.6591D-01	0.2551D-02	0.4786	2
5012.	-195.5	0.3530	0.2013	0.7997D-02	0.6267E-03	0.6655D-01	0.2600D-02	0.4872	3
4963.	-195.4	0.3635	0.2033	0.8156D-02	0.6452E-03	0.6721D-01	0.2651D-02	0.4959	4
4914.	-195.3	0.3745	0.2054	0.8319D-02	0.6644E-03	0.6788D-01	0.2702D-02	0.5049	5
4865.	-195.3	0.3859	0.2075	0.8486D-02	0.6844E-03	0.6856D-01	0.2756D-02	0.5141	6
4817.	-195.2	0.3977	0.2096	0.8656D-02	0.7052E-03	0.6926D-01	0.2811D-02	0.5235	7
4768.	-195.1	0.4100	0.2118	0.8841D-02	0.7269D-03	0.6997D-01	0.2867D-02	0.5332	8
4719.	-194.9	0.4229	0.2141	0.9025D-02	0.7494E-03	0.7069D-01	0.2925D-02	0.5432	9
4671.	-194.8	0.4362	0.2163	0.9216D-02	0.7728E-03	0.7143E-01	0.2985D-02	0.5534	10
4622.	-194.7	0.4502	0.2187	0.9411D-02	0.7972E-03	0.7218D-01	0.3047D-02	0.5638	11
4573.	-194.6	0.4647	0.2210	0.9614D-02	0.8227E-03	0.7295E-01	0.3110D-02	0.5746	12
4525.	-194.5	0.4798	0.2235	0.9823D-02	0.8492E-03	0.7374D-01	0.3176D-02	0.5856	13
4476.	-194.4	0.4957	0.2260	0.1004D-01	0.8768E-03	0.7454D-01	0.3243D-02	0.5969	14
4427.	-194.2	0.5121	0.2285	0.1026D-01	0.9056E-03	0.7536D-01	0.3313D-02	0.6085	15
4378.	-194.1	0.5294	0.2311	0.1049D-01	0.9357E-03	0.7620D-01	0.3385D-02	0.6204	16
4328.	-194.0	0.5474	0.2337	0.1073D-01	0.9670E-03	0.7705E-01	0.3459D-02	0.6327	17
4282.	-193.8	0.5662	0.2365	0.1098D-01	0.9996E-03	0.7793D-01	0.3535D-02	0.6452	18
4233.	-193.7	0.5858	0.2392	0.1123D-01	0.1034E-02	0.7882D-01	0.3614D-02	0.6582	19
4183.	-193.6	0.6064	0.2421	0.1149D-01	0.1070E-02	0.7974D-01	0.3696D-02	0.6714	20
4133.	-193.4	0.6279	0.2450	0.1176D-01	0.1107E-02	0.8067D-01	0.3780D-02	0.6850	21
4082.	-193.2	0.6504	0.2480	0.1205D-01	0.1146E-02	0.8163D-01	0.3867D-02	0.6990	22
4030.	-193.1	0.6741	0.2510	0.1234D-01	0.1187E-02	0.8260D-01	0.3957D-02	0.7134	23
3978.	-192.9	0.6988	0.2541	0.1264D-01	0.1230E-02	0.8360D-01	0.4050D-02	0.7281	24
3926.	-192.7	0.7247	0.2573	0.1295D-01	0.1275E-02	0.8463D-01	0.4146D-02	0.7433	25
3873.	-192.5	0.7519	0.2606	0.1328D-01	0.1322E-02	0.8568D-01	0.4245D-02	0.7588	26
3819.	-192.3	0.7805	0.2640	0.1361D-01	0.1372E-02	0.8675D-01	0.4346D-02	0.7748	27
3765.	-192.1	0.8105	0.2674	0.1396D-01	0.1423E-02	0.8785D-01	0.4450D-02	0.7912	28
3709.	-191.9	0.8420	0.2709	0.1433D-01	0.1478E-02	0.8898D-01	0.4556D-02	0.8081	29
3653.	-191.7	0.8752	0.2746	0.1470D-01	0.1535E-02	0.9013D-01	0.4667D-02	0.8254	30
3596.	-191.5	0.9100	0.2783	0.1509D-01	0.1595E-02	0.9132D-01	0.4779D-02	0.8431	31
3537.	-191.3	0.9467	0.2821	0.1550D-01	0.1658D-02	0.9253E-01	0.4892D-02	0.8614	32
3477.	-191.0	0.9854	0.2861	0.1592D-01	0.1724E-02	0.9378D-01	0.5008D-02	0.8801	33
3416.	-190.8	1.026	0.2901	0.1636D-01	0.1794E-02	0.9506D-01	0.5179D-02	0.8993	34
3354.	-190.5	1.069	0.2942	0.1682D-01	0.1867E-02	0.9637D-01	0.5351D-02	0.9190	35
3291.	-190.2	1.114	0.2985	0.1730D-01	0.1945E-02	0.9772D-01	0.5458D-02	0.9392	36
3227.	-190.0	1.162	0.3029	0.1781D-01	0.2026E-02	0.9910D-01	0.5606D-02	0.9599	37
3162.	-189.7	1.213	0.3074	0.1834D-01	0.2112E-02	0.1005	0.5759D-02	0.9811	38
3096.	-189.4	1.266	0.3120	0.1889D-01	0.2203E-02	0.1020	0.5918D-02	1.003	39
3029.	-189.0	1.323	0.3168	0.1947D-01	0.2299E-02	0.1035	0.6083D-02	1.025	40
2961.	-188.7	1.382	0.3216	0.2008D-01	0.2400E-02	0.1050	0.6255D-02	1.046	41
2892.	-188.3	1.446	0.3268	0.2071D-01	0.2507E-02	0.1066	0.6434D-02	1.071	42
2822.	-188.0	1.513	0.3321	0.2137D-01	0.2620E-02	0.1082	0.6619D-02	1.095	43
2751.	-187.6	1.584	0.3375	0.2206D-01	0.2740E-02	0.1099	0.6813D-02	1.119	44
2679.	-187.2	1.660	0.3430	0.2278D-01	0.2866E-02	0.1116	0.7014D-02	1.143	45

BIBLIOGRAPHY

1. Jazwinski, A. H., "Stochastic Processes and Filtering Theory," New York Academic Press, 1970.

2. Mc Garty, T. , "Stochastic Systems and State Estimation," M. I. T. Press, 1973.

3. Meditch, J. S., "Stochastic Optimal Linear Estimation and Control," Mc Graw Hill Inc. U. S. A., 1969.

4. Sage, A. P., and Melsa, J. L., "Estimation Theory with Applications to Communications and Control," Mc Graw Hill Inc., U. S. A. 1971.

5. Chang, C. B. and Tabaczynski, J. A. , "Application of State Estimation to Target Tracking," IEEE Transactions on Automatic Control, Vol. AC-29 , No. 2, pp98-110 , February 1984.

6. Singer, R. A., "Estimating Optimal Filter Tracking Performance for Manned Maneuvering Targets," IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-6, No. 4, pp. 473-483, July 1970.

7. Clark, B. L., "The Development of an Adaptive Kalman Target Tracking Filter," Proc. AIAA Guidance Control Conference, pp. 365-383, August 1976.

8. Berg, R. F., "Estimation and Prediction for Maneuvering Target Trajectories," IEEE Transactions on Automatic Control, Vol. AC-28, No. 3, pp. 294-303, March 1983.

9. Kirk, D. E., "Evaluation of State Estimators and Predictors for Fire Control Systems," Naval PostGraduate School Report, NPS-52KI74101, October 1974.

10. Fitzgerald, R. J., "Divergence of the Kalman Filter," IEEE Transactions on Automatic Control, Vol. AC-16, No. 6, pp. 736-747, December 1971.

11. Mendel, J. M., "Computational Requirements of a Discrete Kalman Filter," IEEE Transactions on Automatic Control, Vol. AC-16, No. 6, pp. 748-758, December 1971 .

12. Singer, R. A. and Sea, R. G., "Increasing the Computational Efficiency of Discrete Kalman Filters," IEEE Transactions on Automatic Control, Vol. AC-16, No. 2, pp. 254-257, June 1971.

13. Parr, J.M., "An Estimator for an Anti-Aircraft Gun Fire Control System," M.S. Thesis, Naval Postgraduate School, 1974.