U.S. DEPARTMENT OF COMMERCE
National Technical Information Service

AD-A024 329

RECURSIVE FILTERING ALGORITHMS FOR SHIP TRACKING

NAVAL RESEARCH LABORATORY

6 APRIL 1976

NRL Report 7969

Recursive Filtering Algorithms for Ship Tracking

WARREN W. WILLMAN

Systems Research Branch
Space Systems Division

April 6, 1976

NAVAL RESEARCH LABORATORY
Washington, D.C.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE — READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER NRL Report 7969	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) RECURSIVE FILTERING ALGORITHMS FOR SHIP TRACKING		5. TYPE OF REPORT & PERIOD COVERED Interim report on one phase of a continuing NRL Problem. 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Warren W. Willman		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Research Laboratory Washington, D.C. 20375		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NRL Problem B01-10 Project RR003-02-41-6152
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Arlington, Va. 22217		12. REPORT DATE April 6, 1976 13. NUMBER OF PAGES 49
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Adaptive driving noise
Command and control
Kalman filtering
Ocean surveillance
Recursive filtering
Ship tracking
Track smoothing

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Some recursive filtering algorithms were developed for tracking ships when observations are sporadic and imprecise. Tracking with position-only observations was emphasized, but a procedure was also developed for utilizing possible independent observations of ship velocity. Two basic algorithms are considered: a Kalman filter with adaptive driving noise for generating estimates (and containment ellipses) for current and future ship positions, and a corresponding

(Continued)

DD FORM 1473, 1 JAN 73 — EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Bayesian smoother for generating estimates of past positions. The driving noise was treated as a velocity term in a continuous-time model of ship's motion. The details of these two algorithms were developed for tracking on a plane, on a sphere in geographical coordinates, and on a sphere in three-dimensional rectilinear coordinates. A Fortran implementation and some corresponding numerical results were developed for the planar case.

CONTENTS

RECURSIVE FILTERING ALGORITHMS FOR SHIP TRACKING

INTRODUCTION

Existing naval forces can be deployed and concentrated more effectively with a better knowledge of current and probable future locations of the surrounding shipping. Making good use of available surveillance assets in performing this tracking function requires the correlation of data coming from many sources at unpredictable times. Because of the large volume of such data, there is interest in creating a capability for automatic ship tracking using sporadic and noisy observations of position only. This report investigates the use of certain recursive filtering techniques, in particular ones based on Kalman filters, for this purpose. This work provides alternatives to existing algorithms for possible use in automatic ship tracking.

The ship tracking algorithms were designed for these anticipated uses:

Track Generation (Kalman Filter)

- Estimation of present location
- Prediction of future locations
- Generation of "gates" (position confidence regions) for report-to-track correlation at present time in a multitarget environment.

Track Smoothing (Bayesian Smoother)

- Generation of "gates" for report-to-track correlation at previous times (i.e., for out-of-sequence reports) in a multitarget environment
- Estimation of previous locations (i.e., track history).

As input data, these algorithms require reports specifying time of observation, observed ship position, and a covariance matrix for the errors in the observed position (or equivalent information in the form of a confidence region, containment ellipse, etc.). If an observed velocity is also reported at a given observation time, it also must be accompanied by a corresponding error covariance matrix in order to be utilized. Because of the adaptive nature of the Kalman filter used for track generation, no additional information (such as estimated heading, speed, or maneuverability) need be specified externally, and a track can be initiated with a single observation. The time intervals between successive observations may be variable. The input reports are normally processed recursively in their natural time sequence and need not be recalled after their initial use. The incorporation of an out-of-sequence report, however, requires that the intervening reports be available for

Manuscript submitted December 23, 1975.

reprocessing. As output, these algorithms provide (a) estimates of ship position, at present or future times for the track generator and at past times for the smoother, (b) error covariance matrices, or equivalent containment ellipses, that correspond to these estimates. The track generation algorithm also provides estimates of average velocity (two components) and maneuverability (two parameters), which are revised after the receipt of each observation.

There are two salient features of these particular tracking algorithms. First, they are based on a continuous-time ship motion model. This feature allows observations that are unevenly spaced in time to be processed in a statistically consistent manner by the track generator. It also enables the smoother to consistently interpolate the tracks for processing out-of-sequence observations. Second, the ship motion is approximated in this model as the vector sum of a constant average velocity and a two-dimensional Brownian motion. These two velocity terms are processed concurrently but separately by the track generation algorithm. A standard Kalman filter is used to estimate the average velocity with the position. Another recursive procedure is used to estimate the intensity statistics of the Brownian motion from the "residuals" of this Kalman filter. These estimates then are used as "driving noise" parameters in the Kalman filter to adaptively modify its subsequent operation. The purpose of this adaptive modification of the basic Kalman filter algorithm is to make it flexible enough to track a wide variety of ship motions without prior external specification of the motion type.

The track generation algorithm operates recursively in time. Basically it propagates the track forward between observations by dead reckoning and updates it whenever a new report is received. The track smoothing algorithm operates recursively in reverse time using the output of the track generator (position estimate and covariance matrix) as input. These two algorithms are first developed here for tracking on a planar surface. Then they are extended to tracking on the surface of a sphere, both in geographical coordinates of latitude and longitude and in three-dimensional rectilinear coordinates. The algorithms for the planar case are implemented as experimental Fortran programs and tested on both realistic and idealized ship tracks.

SOME COMMENTS ON RECURSIVE FILTERING TERMINOLOGY

Let x be a state vector describing a ship's location such that

$$\dot{x}(t) = F(t)x(t) + w(t)\,, \tag{1}$$

where F is a matrix time function and w is a Gaussian white noise process with mean $\bar{w}(t)$ and (known) normalized covariance matrix $Q(t)$. This normalization refers to the limiting value of

$$\frac{1}{\Delta}\, E\left\{[w(t+\Delta) - \bar{w}(t+\Delta) - w(t) + \bar{w}(t)]\,[w(t+\Delta) - \bar{w}(t+\Delta) - w(t) + \bar{w}(t)]^T\right\}.$$

The state vector x will contain two position components defining the ship's location, and possibly other components, such as velocity, as well. At sporadic times t_i, $i = 0, 1, 2, \ldots$, noisy observations z_i of $x(t_i)$ are received such that

$$z_i = H_i x(t_i) + n_i \,, \tag{2}$$

where H_i is a matrix and $\{n_i\}$ is a sequence of independent random vectors such that

$$n_i \text{ is Normal } [0, R_i]$$

and is statistically independent of w. The observation times are ordered such that $i > j \Leftrightarrow t_i > t_j$ but are otherwise arbitrary. Also, a prior probability distribution is assigned to the state vector at the initial observation time such that

$$x(t_0) \text{ is Normal } [\bar{x}_0, M_0] \,. \tag{3}$$

For any pair of times t and T such that $T > t > t_0$, the conditional probability distribution of $x(t)$, given the prior distribution and all the observations contained in the interval $[t_0, T]$, happens to be multivariate Normal. The moments determining this distribution are denoted as follows.

Definitions

$$\eta(T, t) = E[x(t)] \quad \text{given all data in } [t_0, T]$$

$$K(T, t) = E\left\{[x(t) - \eta(T, t)][x(t) - \eta(T, t)]^T\right\} \quad \text{given all data in } [t_0, T]$$

$$\hat{x}(t) = \eta(t, t) = E[x(t)] \quad \text{given all previous data}$$

$$P(t) = K(t, t) = E\,[x(t) - \hat{x}(t)][x(t) - \hat{x}(t)]^T \quad \text{given all previous data}$$

Capital letters denote matrices, and lower case ones vectors; A^T denotes the transpose of A; E denotes expected value.

The Kalman filter corresponding to Eqs. (1), (2), and (3) generates statistics recursively from the observations. These statistics are, at any time t, an estimate of the state vector $x(t)$ and an error covariance matrix for this estimate. These statistics have the property that

$$\hat{x}(t) = \text{filter estimate of } x(t)$$

and

$$P(t) = \text{filter error covariance matrix for } \hat{x}(t)\,.$$

Hence, this Kalman filter may be regarded as a real-time conditional probability computer. Time t may be at or between observation times.

Another recursive algorithm, called the Bayesian smoother for Eqs. (1) through (3), can be used in conjunction with the Kalman filter algorithm to compute $\eta(T, t)$ and $K(T, t)$. The details of both algorithms are shown in Appendix A. $\eta(T, t)$ is called the smoothed estimate of the state vector $x(t)$ at time T, and $K(T, t)$ is called the error covariance matrix of $\eta(T, t)$.

In the limiting case where det $[M_0] \to \infty$ for the prior distribution of $x(t_0)$ (i.e., a "flat prior"), $\eta(T, t)$ and $K(T, t)$ may be interpreted as the first and second moments, respectively, of the *normalized* likelihood function (i.e., integrates to unity) of $x(t)$ for the observation values received and the motion model postulated. This normalized likelihood function is also multivariate Normal. No prior distribution for $x(t_0)$ is involved in this interpretation, which includes the Kalman filter statistics $\hat{x}(t)$ and $P(t)$ as special cases of $\eta(T, t)$ and $K(T, t)$.

UNDERLYING SHIP MOTION MODEL

The tracking algorithms are based on the Kalman filter and Bayesian smoother for a specific motion model in which a ship's motion is approximated as the vector sum of a constant (average) velocity and a two-dimensional (random) Brownian motion. The intensity of the Brownian motion, which is actually specified by three independent parameters, is selected to correspond to the extent of maneuvering performed by the ship with respect to a constant-speed, great-circle course. This particular motion model was selected as a basis for these tracking algorithms for the following reasons.

- The general recursive filtering algorithms of Appendix A reduce to particularly simple forms for this model if the earth's curvature is neglected. Modifications to account for this curvature are also relatively simple.

- The motion model has sufficient flexibility to give at least a rough approximation to a wide variety of ship motions.

- The smoothed tracks generated are great circles between smoothed observation points.

- Unevenly spaced observations can be accommodated in a systematic way.

- Tracks can be initiated with a single observation, so no qualitative distinction between tracks and unassociated observations is necessary. Track initiation and observation-to-track association can be regarded as special cases of track-to-track association.

- The linear size of the containment ellipse generated by the corresponding Kalman filter (i.e., the track propagation algorithm) often grows only as the square root of the time elapsed since the last observation. Since the gates used in observation-to-track association algorithms often correspond roughly to these containment ellipses, this is possibly an important element in achieving good observation-to-track association performance with sparse observations at high shipping densities.

The execution of the tracking algorithms based on this model requires that each report of a ship's location specify the time, the observed position, and the (2×2) covariance matrix of the observation errors. The average velocity and Brownian motion intensity parameters are estimated from the observation data and need not be specified externally. As output, the algorithms are capable of providing the following information about a ship at any given time.

Track Generation Algorithm (Kalman Filter)

- Estimates of current position and average velocity (four components altogether)

- A (4 × 4) covariance matrix for the errors in these estimates

- An estimate of the three Brownian motion intensity parameters describing the ship's maneuvering.

Track Smoothing Algorithm

- Estimates of position at any past time

- A (2 × 2) covariance matrix for the errors in these estimates.

The information concerning the ship's position is the output of primary importance. Estimates of average velocity and maneuvering are included chiefly for the ulterior purposes of estimating future and past positions.

This motion model is tailored for tracking with position-only observations. It is also possible to incorporate independent velocity observations into the tracking procedures, but this is not a completely straightforward extension. The difficulty is basically that a ship's velocity in this model has two components—an average velocity, which is being estimated, and a completely random velocity, which is not. What is observed, however, is the sum of these two components; thus some additional assumption must be specified about the relation of the observed velocity to the constant-velocity component of the model. Possible procedures for making such a modification are discussed in Appendix B.

TRACKING OF PLANAR MOTION

It is convenient to begin the detailed development of these tracking algorithms with the consideration of a special case. In this case a ship's motion is restricted to a portion of the earth's surface which is small enough to be adequately approximated by a plane. The resulting algorithms are thus easier to understand and can easily be generalized to algorithms for tracking on a sphere. In fact, this generalization is basically just a matter of rotating the coordinate axes at each time of interest, usually an observation time, to realign the y axis with local north.

In this planar context, an approximation of the ship's motion is described by a state vector consisting of two rectangular position coordinates, x and y, which satisfy the differential equation

$$\begin{bmatrix} \dot{x} \\ \hdashline \dot{y} \end{bmatrix} = \begin{bmatrix} w_y \\ \hdashline w_y \end{bmatrix}, \tag{4}$$

where

$\begin{bmatrix} w_x \\ --- \\ w_y \end{bmatrix}$ is a two-dimensional Gaussian white noise process with constant mean

$\begin{bmatrix} u \\ --- \\ v \end{bmatrix}$ (the ship's average velocity) and constant normalized covariance matrix Q (representing maneuvering about this average velocity) such that

$$Q = \left[\begin{array}{c|c} q_{xx} & q_{xy} \\ \hline q_{xy} & q_{yy} \end{array}\right].$$

It is assumed that bias has been removed from the position observations so that the observation at time t_i can reasonably be approximated as the 2-vector

$$\begin{bmatrix} z_{xi} \\ --- \\ z_{yi} \end{bmatrix} = \begin{bmatrix} x(t_i) \\ --- \\ y(t_i) \end{bmatrix} + \begin{bmatrix} n_{xi} \\ --- \\ n_{yi} \end{bmatrix}, \tag{5}$$

where

$\begin{bmatrix} n_{xi} \\ --- \\ n_{yi} \end{bmatrix}$ is a zero-mean bivariate Normal random variable with covariance matrix R_i such that

$$R_i = \left[\begin{array}{c|c} r_{xxi} & r_{xyi} \\ \hline r_{xyi} & r_{yyi} \end{array}\right].$$

The Q and R_i matrices have been kept deliberately in general two-dimensional form. No significant computational reduction appears to be possible, unless the observation errors are assumed to be statistically independent in those rotated coordinates which also diagonalize the Q matrix. Although such an assumption may be reasonable in some situations, it might constitute a serious inefficiency in the use of the data when the "error ellipses" of successive position observations are long and narrow and differ widely in orientation, which is a case of major interest here.

Track Generation—Adaptive Kalman Filter

Although a Kalman filter for Eqs. (4) and (5) could be constructed directly, it could not be implemented in practice for position estimation because the average velocity components u and v and the Brownian motion intensity matrix Q are not known. They must be estimated from the observation data as the track is being generated. It is assumed, however, that the covariance matrix of each observation error is known to the tracker (i.e., both the observed position and error covariance matrix are necessary parts of the input data).

If the average velocity components are adjoined to the state vector of the ship motion model, the motion can be described in terms of the augmented state vector by the differential equation

$$\begin{bmatrix} \dot{x} \\ \hline \dot{y} \\ \hline \dot{u} \\ \hline \dot{v} \end{bmatrix} = \left[\begin{array}{c|c|c|c} 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \end{array}\right] \begin{bmatrix} x \\ \hline y \\ \hline u \\ \hline v \end{bmatrix} + \begin{bmatrix} w_x \\ \hline w_y \\ \hline 0 \\ \hline 0 \end{bmatrix}, \tag{6}$$

where w_x and w_y now denote zero-mean noise components and Q denotes the corresponding 2×2 partition of the 4×4 driving noise matrix, the other components of which are zero. Furthermore, the position observations can also be expressed in terms of this augmented state vector as

$$\begin{bmatrix} z_{xi} \\ \hline z_{yi} \end{bmatrix} = \left[\begin{array}{c|c|c|c} 1 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 \end{array}\right] \begin{bmatrix} x(t_i) \\ \hline y(t_i) \\ \hline u \\ \hline v \end{bmatrix} + \begin{bmatrix} n_{xi} \\ \hline n_{yi} \end{bmatrix}. \tag{7}$$

Hence, if Q were known, u and v could be estimated concurrently with x and y by the use of the Kalman filter corresponding to Eqs. (6) and (7). As an ad hoc procedure based on this concept, this filtering algorithm is used as if Q were a known constant matrix, except that the value of Q used in these computations is updated at each observation time by another recursive algorithm. The initial estimates of u, v, and Q are all taken as zero, which in effect gives priority to estimating the average velocity (u, v) over estimating the "maneuvering matrix" Q.

The Basic Kalman Filter

If Q is treated as a known constant for the moment, the evolution of the conditional mean and covariance matrix of the state vector between generic observation times t_i and t_{i+1} can be described by specializing the results of Appendix A to Eqs. (6) and (7), giving the following differential equations:

$$\begin{aligned}
\dot{\hat{x}} &= \hat{u} \\
\dot{\hat{y}} &= \hat{v} \\
\dot{\hat{u}} &= 0 \\
\dot{\hat{v}} &= 0 \\
\dot{p}_{xx} &= 2p_{xu} + q_{xx} \\
\dot{p}_{xy} &= p_{xv} + p_{yu} + q_{xy} \\
\dot{p}_{yy} &= 2p_{yv} + q_{yy} \\
\dot{p}_{xu} &= p_{uu} \\
\dot{p}_{xv} &= p_{uv} \\
\dot{p}_{yu} &= p_{uv} \\
\dot{p}_{yv} &= p_{vv} \\
\dot{p}_{uu} &= 0 \\
\dot{p}_{uv} &= 0 \\
\dot{p}_{vv} &= 0 .
\end{aligned}$$

In this case, the differential equations can be integrated analytically to give

$$\hat{x}(t_{i+1}^-) = \hat{x}(t_i^+) + \tau\hat{u}(t_i^+) \tag{8}$$

$$\hat{y}(t_{i+1}^-) = \hat{y}(t_i^+) + \tau\hat{v}(t_i^+) \tag{9}$$

$$\hat{u}(t_{i+1}^-) = \hat{u}(t_i^+) \tag{10}$$

$$\hat{v}(t_{i+1}^-) = \hat{v}(t_i^+) \tag{11}$$

$$m_{xx} = p_{xx}(t_i^+) + 2p_{xu}(t_i^+)\tau + p_{uu}(t_i^+)\tau^2 + q_{xx}\tau \tag{12}$$

$$m_{xy} = p_{xy}(t_i^+) + [p_{xv}(t_i^+) + p_{yu}(t_i^+)]\tau + p_{uv}(t_i^+)\tau^2 + q_{xy}\tau \tag{13}$$

$$m_{yy} = p_{yy}(t_i^+) + 2p_{yv}(t_i^+)\tau + p_{vv}(t_i^+)\tau^2 + q_{yy}\tau \tag{14}$$

$$m_{xu} = p_{xu}(t_i^+) + p_{uu}(t_i^+)\tau \tag{15}$$

$$m_{xv} = p_{xv}(t_i^+) + p_{uv}(t_i^+)\tau \tag{16}$$

$$m_{yu} = p_{yu}(t_i^+) + p_{uv}(t_i^+)\tau \tag{17}$$

$$m_{yv} = p_{yv}(t_i^+) + p_{vv}(t_i^+)\tau \tag*{(13)}$$

$$m_{uu} = p_{uu}(t_i^+) \tag{19}$$

$$m_{uv} = p_{uv}(t_i^+) \tag{20}$$

$$m_{vv} = p_{vv}(t_i^+) , \tag{21}$$

where $\tau = t_{i+1} - t_i$, and the m variables denote the corresponding components of the P matrix at time t_{i+1}^-. After the observation

$$\begin{bmatrix} z_x(i+1) \\ \hline z_y(i+1) \end{bmatrix}$$

at time t_{i+1}, the conditional mean and covariance matrix components are updated according to the equations

$$\begin{bmatrix} \hat{x}_{i+1} \\ \hline \hat{y}_{i+1} \end{bmatrix} = \begin{bmatrix} \hat{x}_i \\ \hline \hat{y}_i \end{bmatrix} + \tau \begin{bmatrix} \hat{u}_i \\ \hline \hat{v}_i \end{bmatrix} + \begin{bmatrix} p_{xx} & p_{xy} \\ \hline p_{xy} & p_{yy} \end{bmatrix}_{i+1} R_{i+1}^{-1} \begin{bmatrix} z_x(i+1) - \hat{x}_i - \tau\hat{u}_i \\ \hline z_y(i+1) - \hat{y}_i - \tau\hat{v}_i \end{bmatrix} \tag{22}$$

$$\begin{bmatrix} \hat{u}_{i+1} \\ \hline \hat{v}_{i+1} \end{bmatrix} = \begin{bmatrix} \hat{u}_i \\ \hline \hat{v}_i \end{bmatrix} + \begin{bmatrix} p_{xu} & p_{yu} \\ \hline p_{xv} & p_{yv} \end{bmatrix}_{i+1} R_{i+1}^{-1} \begin{bmatrix} z_x(i+1) - \hat{x}_i - \tau\hat{u}_i \\ \hline z_y(i+1) - \hat{y}_i - \tau\hat{v}_i \end{bmatrix}, \tag{23}$$

where the integer subscripts i and $i + 1$ indicate the value of variables at t_i^+ and t_{i+1}^+ and the P matrix components at t_{i+1}^+ are computed from the equation

$$\begin{bmatrix} p_{xx} & p_{xy} & p_{xu} & p_{xv} \\ p_{xy} & p_{yy} & p_{yu} & p_{yv} \\ p_{xu} & p_{yu} & p_{uu} & p_{uv} \\ p_{xv} & p_{yv} & p_{uv} & p_{vv} \end{bmatrix}_{i+1} = \begin{bmatrix} m_{xx} & m_{xy} & m_{xu} & m_{xv} \\ m_{xy} & m_{yy} & m_{yu} & m_{yv} \\ m_{xu} & m_{yu} & m_{uu} & m_{uv} \\ m_{xv} & m_{yv} & m_{uv} & m_{vv} \end{bmatrix} - \begin{bmatrix} m_{xx} & m_{xy} \\ m_{xy} & m_{yy} \\ m_{xu} & m_{yu} \\ m_{xv} & m_{yv} \end{bmatrix}$$

$$\times \begin{bmatrix} m_{xx} + r_{xx}(i+1) & m_{xy} + r_{xy}(i+1) \\ \hline m_{xy} + r_{xy}(i+1) & m_{yy} + r_{yy}(i+1) \end{bmatrix}^{-1}$$

$$\times \begin{bmatrix} m_{xx} & m_{xy} & m_{xu} & m_{xv} \\ \hline m_{xy} & m_{yy} & m_{yu} & m_{yv} \end{bmatrix}. \tag{24}$$

Once the initial conditions are specified, Eqs. (8) through (24) define the track generation procedure under the assumption of known Q. A convenient practice for initiating this procedure is to start tracking at time t_0^+, immediately after the first observation, with

$$\begin{aligned} \hat{x}(t_0^+) &= z_{xo} \\ \hat{y}(t_0^+) &= z_{yo} \\ \hat{u}(t_0^+) &= 0 \\ \hat{v}(t_0^+) &= 0 \end{aligned} \tag{25}$$

$$\begin{aligned} p_{xx}(t_0^+) &= r_{xxo} \\ p_{xy}(t_0^+) &= r_{xyo} \\ p_{yy}(t_0^+) &= r_{yyo} \\ p_{uu}(t_0^+) &= p_{vv}(t_0^+) = \frac{1}{2} V^2 \end{aligned} \tag{26}$$

$$\text{all other components of } P(t_0^+) = 0 \,,$$

where V is the (externally specified) average speed of ships covered by the tracking system. It is possible to avoid specifying a value of V by using $p_{uu}(t_0^+) = p_{vv}(t_0^+) = \infty$, which corresponds to the use of a flat prior for the initial state vector to generate its normalized likelihood function. This modification would make the tracking somewhat less efficient, however, and is probably needless since a reliable estimate of V, or at least a finite upper bound, would usually be available.

Adaptive Modification for Recursive Estimation of Q

To account for unknown Q, note that the term

$$\left[\begin{array}{c} z_x(i+1) - \hat{x}_i - \tau\hat{u}_i \\ \hdashline z_y(i+1) - \hat{y}_i - \tau\hat{v}_i \end{array}\right]$$

in Eqs. (22) and (23) can be expressed as

$$\left[\begin{array}{c} x_i - \hat{x}_i \\ \hdashline y_i - \hat{y}_i \end{array}\right] + \tau \left[\begin{array}{c} u - \hat{u}_i \\ \hdashline v - v_i \end{array}\right] + \int_{t_i}^{t_{i+1}} w\, dt + \left[\begin{array}{c} n_x(i+1) \\ \hdashline n_y(i+1) \end{array}\right],$$

which, if Q were known, would have a zero mean and covariance matrix

$$\left[\begin{array}{c:c} m_{xx} & m_{xy} \\ \hdashline m_{xy} & m_{yy} \end{array}\right] + \left[\begin{array}{c:c} \tau q_{xx} & \tau q_{xy} \\ \hdashline \tau q_{xy} & \tau q_{yy} \end{array}\right] + \left[\begin{array}{c:c} r_{xx}(i+1) & r_{xy}(i+1) \\ \hdashline r_{xy}(i+1) & r_{yy}(i+1) \end{array}\right].$$

Furthermore, if Q were known and used in Eqs. (8) through (24), the vectors

$$\begin{bmatrix} \epsilon_x(i+1) \\ \hline \epsilon_y(i+1) \end{bmatrix} = \begin{bmatrix} z_x(i+1) - \hat{x}_i - \tau\hat{u}_i \\ \hline z_y(i+1) - \hat{y}_i - \tau\hat{v}_i \end{bmatrix}$$

would be statistically independent, in which case a reasonable statistic $\hat{Q}(i+1)$ for approximating Q at time t_{i+1}^+ would be given by the equations

$$\hat{q}_{xx}(i-1) = \frac{1}{i+1} \sum_{j=1}^{i+1} \frac{1}{\tau_j} \left[\epsilon_{xj}^2 - p_{xx}(t_j^-) - r_{xxj}\right]$$

$$\hat{q}_{xy}(i-1) = \frac{1}{i+1} \sum_{j=1}^{i+1} \frac{1}{\tau_j} \left[\epsilon_{xj}\epsilon_{yj} - p_{xy}(t_j^-) - r_{xyj}\right]$$

and

$$\hat{q}_{yy}(i+1) = \frac{1}{i+1} \sum_{j=1}^{i+1} \frac{1}{\tau_j} \left[\epsilon_{yj}^2 - p_{yy}(t_j^-) - r_{yyj}\right],$$

where τ_j denotes $t_j - t_{j-1}$. These statistics cannot be used directly to estimate Q because Q is needed to compute the ϵ's and p's. However, these statistics obey the following recursion equations:

$$\hat{q}_{xx}(i+1) = \hat{q}_{xx}(i) + \frac{1}{(i+1)}\left[\frac{\epsilon_x^2(i+1) - m_{xx} - r_{xx}(i+1)}{\tau} - \hat{q}_{xx}(i)\right] \tag{27}$$

$$\hat{q}_{xy}(i+1) = \hat{q}_{xy}(i) + \frac{1}{(i+1)}\left[\frac{\epsilon_x(i+1)\epsilon_y(i+1) - m_{xy} - r_{xy}(i+1)}{\tau} - \hat{q}_{xy}(i)\right] \tag{28}$$

and

$$\hat{q}_{yy}(i+1) = \hat{q}_{yy}(i) + \frac{1}{(i+1)}\left[\frac{\epsilon_y^2(i+1) - m_{yy} - r_{yy}(i+1)}{\tau} - q_{yy}(i)\right], \tag{29}$$

where τ, m_{xx}, m_{xy} and m_{yy} are defined as in Eqs. (8) through (24). As an ad hoc procedure, Eqs. (27) through (29) are used recursively to generate estimates of the components of Q, starting with $\hat{q}_{xx}(o) = \hat{q}_{xy}(o) = \hat{q}_{yy}(o) = 0$. It is also desirable to constrain these estimates so that they form a positive semidefinite matrix which is diagonalized by a rotation to coordinates aligned with the estimated average velocity vector (i.e., maneuvering is assumed symmetric about the ship's average heading). One way which has been found to accomplish this is to use the following as estimates for the time interval (t_i, t_{i+1}) in the context of Eqs. (8) through (24):

$$q_{xx} = \frac{1}{2}\left[\xi + \frac{\hat{u}_i^2 - \hat{v}_i^2}{\hat{u}_i^2 + \hat{v}_i^2}\lambda\right] \tag{30}$$

$$q_{xy} = \frac{\hat{u}_i\hat{v}_i}{\hat{u}_i^2 + \hat{v}_i^2}\lambda \tag{31}$$

and

$$q_{yy} = \frac{1}{2}\left[\xi - \frac{\hat{u}_i^2 - \hat{v}_i^2}{\hat{u}_i^2 + \hat{v}_i^2}\lambda\right], \tag{32}$$

where

$$\xi = \max\{0, [\hat{q}_{xx}(i) + \hat{q}_{yy}(i)]\} \tag{33}$$

$$\lambda = \begin{cases} \xi & \text{if} \quad \dfrac{\hat{u}_i^2 + \hat{v}_i^2}{\hat{u}_i\hat{v}_i}\,\hat{q}_{xy}(i) > \xi \\[2ex] -\xi & \text{if} \quad \dfrac{\hat{u}_i^2 + \hat{v}_i^2}{\hat{u}_i\hat{v}_i}\,\hat{q}_{xy}(i) < -\xi \\[2ex] \dfrac{\hat{u}_i^2 + \hat{v}_i^2}{\hat{u}_i\hat{v}_i}\,\hat{q}_{xy}(i) & \text{otherwise}. \end{cases} \tag{34}$$

The justification for this procedure is given in Appendix C.

Final Algorithm

This completes the specification of the recursive track generation procedure. To summarize this procedure, tracking begins immediately after the initial observation at time t_0 with initial conditions given by Eqs. (5), (25), (26), and $\hat{q}_{xx}(o) = \hat{q}_{xy}(o) = \hat{q}_{yy}(o) = 0$. From time t_i^+ to time t_{i+1}^+, $i = 0, 1, \ldots,$ the track is generated as follows:

Track Propagation

- Generate q_{xx}, q_{xy}, q_{yy} from $\hat{q}_{xx}(i)$, $\hat{q}_{xy}(i)$, $\hat{q}_{yy}(i)$ with Eqs. (30) through (34).

- Use these values in Eqs. (8) through (21) to generate $\hat{x}(t_{i+1}^-)$, $\hat{y}(t_{i+1}^-)$, $\hat{u}(t_{i+1}^-)$, $\hat{v}(t_{i+1}^-)$, and the m's.

Track Updating

- Use these values of the m's in Eqs. (22) through (24) to generate the new estimates $\hat{x}_{i+1}$, $\hat{y}_{i+1}$, $\hat{u}_{i+1}$, $\hat{v}_{i+1}$ from the observations $z_x(i+1)$, $z_y(i+1)$.

• Use Eqs. (27) through (29) to compute $\hat{q}_{xx}(i+1)$, $\hat{q}_{xy}(i+1)$, $\hat{q}_{yy}(i+1)$, where (as defined earlier)

$$\epsilon_x(i+1) = z_x(i+1) - \hat{x}_i - \tau\hat{u}_i$$

$$\epsilon_y(i+1) = z_y(i+1) - \hat{y}_i - \tau\hat{v}_i \, .$$

Note that track initiation is accomplished with only one position observation in this algorithm. Hence an unassociated observation can be regarded as a one-point track. In this regard, it is perhaps helpful to consider a track as consisting of the time history of the conditional mean and covariance matrix of the entire state vector, not just the time history of $\hat{x}$ and $\hat{y}$. With this interpretation, no qualitative distinction between report-to-track association and track-to-track association is necessary. The implementation of this algorithm requires that a total of 17 quantities be carried, propagated, and updated for each ship being tracked (i.e., the estimates $\hat{x}$, $\hat{y}$, $\hat{u}$, $\hat{v}$, the 10 independent "p" components of the corresponding covariance matrix, and the maneuvering parameter estimates $\hat{q}_{xx}$, $\hat{q}_{xy}$, and $\hat{q}_{yy}$).

In actual operation, it might be well to reinitialize this tracking algorithm, perhaps at the discretion of a human operator, if a sequence of consistently large residuals ϵ_x and ϵ_y are encountered for a given ship; such an event would imply an abrupt change in maneuvering behavior. Another possibility would be to limit the $i+1$ factor in Eqs. (27) through (29) to some maximum to prevent the estimates of Q from depending too heavily on old observations.

Prediction of Future Positions

Although the preceding algorithm is contemplated mainly for the updating of position estimates after the receipt of an additional observation (one-point updating), it can also be used for computing the conditional probability distribution, given all currently available observation data, of a ship's position and velocity at a future time. If t_i is the time of the last observation, this can be done with the track propagation steps of the above algorithm by regarding the future time in question as t_{i+1}. The conditional distribution is then Normal with mean (predicted position and velocity)

$$\begin{bmatrix} \hat{x}(t_{i+1}^-) \\ \hline \hat{y}(t_{i+1}^-) \\ \hline \hat{u}(t_{i+1}^-) \\ \hline \hat{v}(t_{i+1}^-) \end{bmatrix}$$

and covariance matrix M, as defined by Eqs. (12) through (21). Aside from its obvious tactical value, this information can also be used for the construction of position and/or velocity gates in observation-to-track correlation; in the latter use, t_{i+1} is the time of the observation possibly being correlated. In either case, however, unless a new observation is actually used to update the track, the "track updating" steps are not performed and any subsequent track propagation or updating proceeds from time t_i as if these computations had never taken place.

Track Smoothing

In addition to keeping track of the conditional distribution of a ship's current position and velocity, it is occasionally useful to know the current distribution of its state vector at a previous time as well (i.e., the smoothed track statistics η and K). The main use foreseen for this information is in observation-to-track association for out-of-sequence observations. A smoothed track can be considerably more precise in practice than the past history of the track generated by the corresponding Kalman filter. This extra precision would enable out-of-sequence observations to be correlated to tracks more accurately in areas of high shipping density.

This track smoothing algorithm is the specialization of the generic Bayesian smoother of Appendix A to the particular ship motion model adopted above. As a simplifying approximation, however, it is assumed that the current estimates of the velocity and maneuvering parameters u, v, q_{xx}, q_{xy}, and q_{yy} at the time of smoothing are the exact values of these quantities—with one exception. The estimates $\hat{q}_{xx}$, $\hat{q}_{xy}$, and $\hat{q}_{yy}$ are first adjusted according to Eqs. (30) through (34) to insure that the resulting "maneuvering" matrix, denoted by $\bar{Q}$, is positive semidefinite. Then $\bar{Q}$ is further modified to compensate for the uncertainty in the velocity estimate, which is suppressed by the assumption that $\hat{u}$ and $\hat{v}$ are precisely known constants. It has been found by numerical experimentation that this modification can be achieved reasonably well by replacing $\bar{Q}$ with a matrix Q such that

$$Q = \bar{Q} + (T - t_0)\left[\begin{array}{c|c} P_{uu}(T) & P_{uv}(T) \\ \hline p_{uv}(T) & p_{vv}(T) \end{array}\right], \tag{35}$$

where T is the time of smoothing and t_0 is the time of track initiation. This simplification makes it possible to use the Bayesian smoother corresponding to Eqs. (4) and (5), rather than to Eqs. (6) and (7), a reduction from four state variables to two. To implement this smoother at time T requires that the quantities $\hat{x}(t_i^+)$, $\hat{y}(t_i^+)$, $p_{xx}(t_i^+)$, $p_{xy}(t_i^+)$, and $p_{yy}(t_i^+)$ be available for all observations times t_i such that $t_i < T$.

For clarity of notation, the components of $\eta(T, t)$ are denoted here by

$$\left[\begin{array}{c} \bar{x}(t) \\ \hline \bar{y}(t) \end{array}\right],$$

and those of $K(T, t)$ by

$$\left[\begin{array}{c|c} k_{xx}(t) & k_{xy}(t) \\ \hline k_{xy}(t) & k_{yy}(t) \end{array}\right].$$

The variables η and K are continuous in t and obey the following differential equations between generic observation times t_i and t_{i+1} (T is considered a fixed parameter here):

$$\dot{\eta}\left(= \frac{\partial \eta(T,t)}{\partial t}\right) = \left[\begin{array}{c} u \\ \hline v \end{array}\right] + QP^{-1}\left(\eta - \left[\begin{array}{c} \hat{x} \\ \hline \hat{y} \end{array}\right]\right) \tag{36}$$

and

$$\dot{K} = QP^{-1}K + KP^{-1}Q - Q, \tag{37}$$

where

$$\begin{cases} \hat{x}(t) = \hat{x}(t_i^+) + (t - t_i)u \\ \hat{y}(t) = \hat{y}(t_i^+) + (t - t_i)v \\ P(t) = \left[\begin{array}{c|c} p_{xx}(t_i^+) & p_{xy}(t_i^+) \\ \hline p_{xy}(t_i^+) & p_{yy}(t_i^+) \end{array}\right] + (t - t_i)Q \end{cases}$$

is the Kalman filter solution corresponding to Eq. (4). Equations (36) and (37) can be integrated analytically in this case by noting that the quantities

$$P^{-1}\left(\eta - \left[\begin{array}{c} \hat{x} \\ \hline \hat{y} \end{array}\right]\right)$$

and

$$P^{-1}(K - P)P^{-1}$$

are constant on the interval (t_i, t_{i+1}). Therefore, by continuity, $\eta(t)$ and $K(t)$ can be computed for any $t \epsilon (t_i, t_{i+1})$ in terms of their values η_{i+1} and K_{i+1} at time t_{i+1} as follows:

- Let

$$M_{i+1} = P(t_{i+1}^-) = P(t_i^+) + Q(t_{i+1} - t_i), \tag{38}$$

$$\tilde{x}_{i+1} = \hat{x}(t_{i+1}^-) = \hat{x}(t_i^+) + u(t_{i+1} - t_i), \tag{39}$$

and

$$\tilde{y}_{i+1} = \hat{y}(t_{i+1}^-) = \hat{y}(t_i^+) + v(t_{i+1} - t_i). \tag{40}$$

- Compute

$$\eta(T, t) = \begin{bmatrix} \hat{x}(t) \\ --- \\ \hat{y}(t) \end{bmatrix} + P(t)M_{i+1}^{-1}\left(\eta_{i+1} - \begin{bmatrix} \tilde{x}_{i+1} \\ --- \\ \tilde{y}_{i+1} \end{bmatrix}\right) \tag{41}$$

and

$$K(T, t) = P(t) + P(t)M_{i+1}^{-1}(K_{i+1} - M_{i+1})M_{i+1}^{-1}P(t). \tag{42}$$

Thus, it is easy to compute η and K recursively, starting with

$$\eta(T, T) = \begin{bmatrix} \hat{x}(T) \\ --- \\ \hat{y}(T) \end{bmatrix} \quad \text{and} \quad K(T, T) = P(T)$$

and using Eqs. (38) through (42) on the interobservation intervals in reverse sequence. Note that setting $t = t_i$ in Eqs. (40) and (41) gives η_i and K_i. Equations (38) through (42) can be computed component by component by first defining

$$\tau = t_{i+1} - t_i$$

and

$$s = t - t_i,$$

then computing

$$p_1 = p_{xx}(t_i^+) + q_{xx}s,$$

$$p_2 = p_{xy}(t_i^+) + q_{xy}s,$$

$$p_3 = p_{yy}(t_i^+) + q_{yy}s,$$

$$m_1 = p_{xx}(t_i^+) + q_{xx}\tau,$$

$$m_2 = p_{xy}(t_i^+) + q_{xy}\tau,$$

$$m_3 = p_{yy}(t_i^+) + q_{yy}\tau,$$

$\tilde{x}_{i+1}$ from Eq. (39),

and

$\tilde{y}_{i+1}$ from Eq. (40),

and then computing

$$\bar{x}(t) = \hat{x}(t) + \frac{p_1 m_3 - p_2 m_2}{m_1 m_3 - m_2^2} (\bar{x}_{i+1} - \tilde{x}_{i+1}) + \frac{p_2 m_1 - p_1 m_2}{m_1 m_3 - m_2^2} (\bar{y}_{i+1} - \tilde{y}_{i+1}) \tag{43}$$

$$\bar{y}(t) = \hat{y}(t) + \frac{p_2 m_3 - p_3 m_2}{m_1 m_3 - m_2^2} (\bar{x}_{i+1} - \tilde{x}_{i+1}) + \frac{p_3 m_1 - p_2 m_2}{m_1 m_3 - m_2^2} (\bar{y}_{i+1} - \tilde{y}_{i+1}), \tag{44}$$

$$k_{xx}(t) = p_{xx}(t) + \Big\{k_{xx}(t_{i+1})[p_1 m_3 - p_2 m_2]^2 + 2k_{xy}(t_{i+1})[p_2 m_1 - p_1 m_2] \times [p_1 m_3 - p_2 m_2] + k_{yy}(t_{i+1})[p_2 m_1 - p_1 m_2]^2\Big\} \left\{\frac{1}{m_1 m_3 - m_2^2}\right\}^2, \tag{45}$$

$$k_{xy}(t) = p_{xy}(t) + \{k_{xx}(t_{i+1})[p_1 m_3 - p_2 m_2][p_2 m_3 - p_3 m_2] + k_{xy}(t_{i+1}) \times [(p_2 m_1 - p_1 m_2)(p_2 m_3 - p_3 m_2) + (p_1 m_3 - p_2 m_2)(p_3 m_1 - p_2 m_2)] + k_{yy}(t_{i+1})[p_2 m_1 - p_1 m_2][p_3 m_1 - p_2 m_2]\} \left\{\frac{1}{m_1 m_3 - m_2^2}\right\}^2, \tag{46}$$

and

$$k_{xx}(t) = p_{yy}(t) + \Big\{k_{xx}(t_{i+1})[p_2 m_3 - p_3 m_2]^2 + 2k_{xy}(t_{i+1})[p_3 m_1 - p_2 m_2] \times [p_2 m_3 - p_3 m_2] + k_{yy}(t_{i+1})[p_3 m_1 - p_2 m_2]^2\Big\} \left\{\frac{1}{m_1 m_3 - m_2^2}\right\}^2. \tag{47}$$

Note that Eqs. (43), (44), and the corresponding Kalman filter equations for $\hat{x}$ and $\hat{y}$ imply that the smoothed tracks $(\bar{x}, \bar{y})$ for this type of motion model have constant velocities between observation times. The tracks are continuous, but there are, in general, discontinuities in the velocities at the observation times.

Numerical Performance

These planar tracking algorithms have been implemented as experimental Fortran programs. These implementations have been tested on both idealized and realistic ship tracks.

Figure 1 shows the performance of these algorithms on a realistic ship track consisting of 17 observations. Covariance matrix information here is depicted in terms of a corresponding "two-sigma ellipse," a level curve of the (Bivariate Normal) probability density function which contains 86% of the probability mass of the random variable in question.

Figure 2 shows the comparative sizes of the Kalman filter's and Bayesian smoother's 86% containment ellipses for some representative previous positions for this same ship track. The smaller size of the smoother's containment ellipses is significant in the observation-to-track association problem for out-of-sequence observations, as explained earlier.

Figure 3 is a listing of the Fortran program that generated the results in Figs. 1 and 2. Execution time on a CDC 3800 computer was 2 s for this example. This is just an experimental program, but it gives a rough indication of what is involved in the implementation of these tracking algorithms.

Figure 4 shows the tracking algorithms' performance on an idealized track. In this case the true track is displayed to show the tracking accuracy.

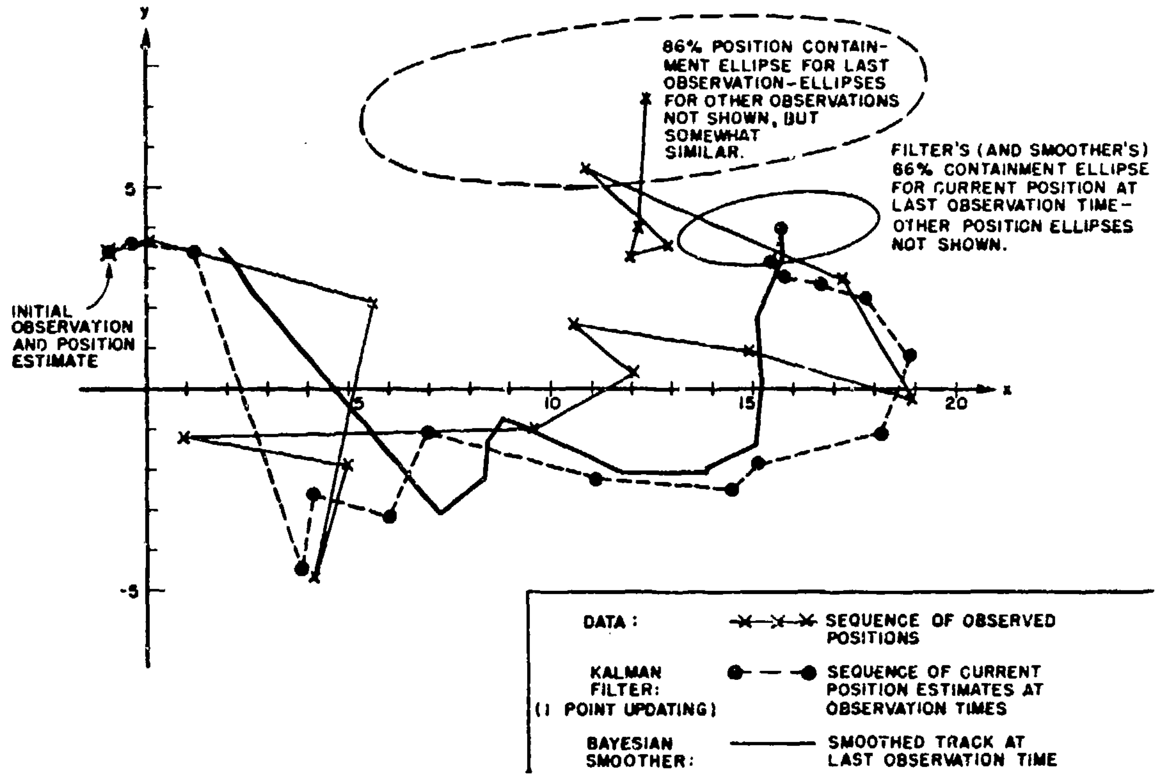

Fig. 1—Performance of recursive tracking algorithms on realistic ship track (true track not shown)

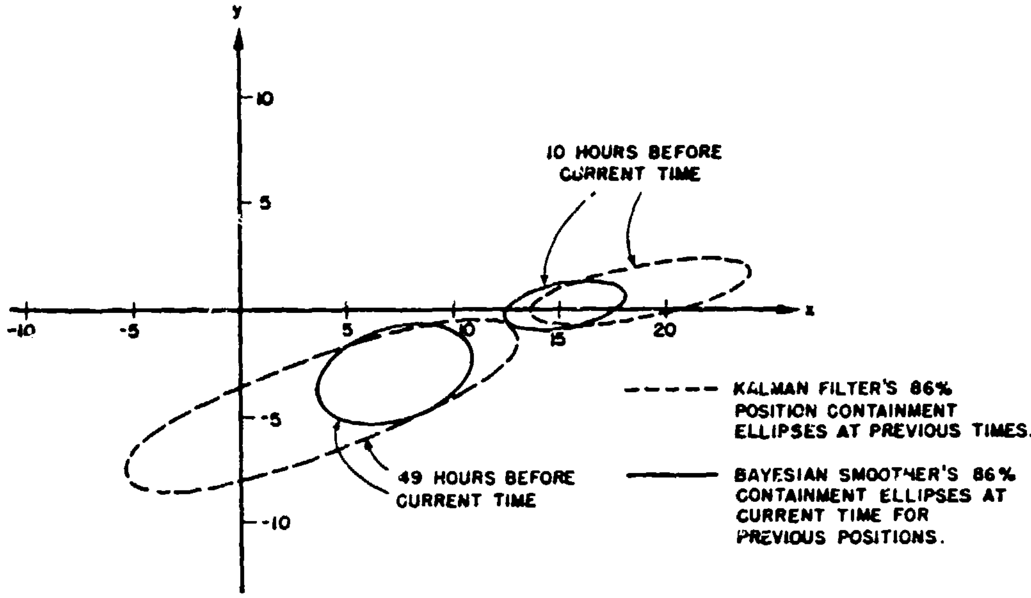

Fig. 2—Containment ellipses at last observation time for previous positions on realistic ship track

TRACKING ON A SPHERE: GEOGRAPHICAL COORDINATES

The ship tracking algorithms developed in the preceding section can be extended to include the effects of earth curvature if it is assumed that this curvature is negligible within a ship's position sigma ellipse generated by the filter or the smoother. In this case, the ship's motion can continually be approximated by Eqs. (4) and (5) in local rectangular coordinates. Although it has been found convenient for some other tracking algorithms of this type to keep these local x, y coordinates aligned with the estimated velocity vector for reasons of symmetry, it seems simpler here to keep them aligned with local north because the motion and observation models are fully two-dimensional anyway. This section contains an extension of the planar filtering and smoothing algorithms to tracking on a sphere when a ship's location is described in geographical latitude and longitude coordinates. Alternate algorithms are developed in the following section for tracking on a sphere in rectilinear coordinates, which have certain computational advantages.

Track Generation

The basic procedure using the recursive filter is to perform the track propagation step with dead reckoning along a great-circle path using the estimated average velocity. Track updating is accomplished by first establishing a rectangular coordinate system centered at the current propagated position and aligned with local north, then updating as in the section "Tracking of Planar Motion," and finally computing the latitude and longitude of the updated position. Between successive observations, say at times t_i and t_{i+1}, the details of this procedure are as follows.

```
      PROGRAM MAIN
C
C                    KALMAN FILTER WITH ADAPTIVE DRIVING NOISE
C
      DIMENSION T(999),XX(999),YY(999),XS(999),YS(999),PXX(999),AX(999)
      DIMENSION AY(999),PXY(999),PYY(999),ARXX(999),ARXY(999),ARYY(999)
      DIMENSION SMAJ(999),SMIN(999),TH(999)
C
C                    READ PARAMETER VALUES
C
C                    N = NUMBER OF DETECTIONS (= NO. OF DATA CARDS)
C                    VELVAR = PRIOR SPEED VARIANCE
C
      READ 5071,N,VELVAR
 5071 FORMAT(I3,F10.5)
C
C                    READ (AND STORE) DATA FOR EACH DETECTION
C
C         T = TIME
C         AX,AY = OBSERVED LOCATION COORDINATES
C         SMA = SEMIMAJOR AXIS OF 86 PERCENT CONTAINMENT ELLIPSE
C               FOR OBSERVATION
C         SMI = SEMIMINOR AXIS OF CONTAINMENT ELLIPSE
C         THT = ORIENTATION OF SEMIMAJOR AXIS (DEGREES CLOCKWISE
C               FROM Y-AXIS)
C
 5070 FORMAT(6F10.4)
      DO 9 I=1,N
      READ 5070,T(I),AX(I),AY(I),SMA,SMI,THT
      THT=THT/57.3
      ARXX(I)=((SMA*SIN(THT))**2+(SMI*COS(THT))**2)/4.
      ARXY(I)=SIN(THT)*COS(THT)*(SMA*SMA-SMI*SMI)/4.
    9 ARYY(I)=((SMI*SIN(THT))**2+(SMA*COS(THT))**2)/4.
C
C                    INITIALIZATION
C
      XX(1)=AX(1)
      YY(1)=AY(1)
      PXX(1)=ARXX(1)
      PXY(1)=ARXY(1)
      PYY(1)=ARYY(1)
      C1=PXX(1)+PYY(1)
      C2=SQRT((PXX(1)-PYY(1))**2+4.*PXY(1)**2)
      C1=.5*(C1+C2)
      C2=C1-C2
      SMAJ(1)=2.*SQRT(C1)
      SMIN(1)=2.*SQRT(C2)
      TH(1)=57.3*ATAN((PXX(1)-C1)/PXY(1))+90.
      QXX=0.
      QXY=0.
      QYY=0.
      U=0.
      V=0.
      PXU=0.
      PXV=0.
      PYU=0.
      PYV=0.
      PUU=.5*VELVAR
      PUV=0.
```

Fig. 3—Program listing

```
      PVV=PUU
      CXX=0.
      CXY=0.
      CYY=0.
C
C                 RECURSIVE STATE VECTOR ESTIMATION
C
      DO 1 I=2,N
      ZX=AX(I)
      ZY=AY(I)
      RXX=ARXX(I)
      RXY=ARXY(I)
      RYY=ARYY(I)
      TAU=T(I)-T(I-1)
      XBAR=XX(I-1)+U*TAU
      YBAR=YY(I-1)+V*TAU
      GXX=PXX(I-1)+2.*PXU*TAU+PUU*TAU*TAU+QXX*TAU
      GXY=PXY(I-1)+(PXV+PYU)*TAU+PUV*TAU*TAU+QXY*TAU
      GYY=PYY(I-1)+2.*PYV*TAU+PVV*TAU*TAU+QYY*TAU
      GXU=PXU+PUU*TAU
      GXV=PXV+PUV*TAU
      GYU=PYU+PUV*TAU
      GYV=PYV+PVV*TAU
      DET=(GXX+RXX)*(GYY+RYY)-(GXY+RXY)**2
      HXX=(GYY+RYY)/DET
      HXY=-(GXY+RXY)/DET
      HYY=(GXX+RXX)/DET
      PXX(I)=GXX-GXX*GXX*HXX-2.*GXX*GXY*HXY-GXY*GXY*HYY
      PXY(I)=GXY-GXX*GXY*HXX-(GXX*GYY+GXY*GXY)*HXY-GXY*GYY*HYY
      PYY(I)=GYY-GYY*GYY*HYY-2.*GYY*GXY*HXY-GXY*GXY*HXX
      C1=PXX(I)+PYY(I)
      C2=SQRT((PXX(I)-PYY(I))**2+4.*PXY(I)**2)
      C1=.5*(C1+C2)
      C2=C1-C2
      SMAJ(I)=2.*SQRT(C1)
      SMIN(I)=2.*SQRT(C2)
      TH(I)=57.3*ATAN((PXX(I)-C1)/PXY(I))+90.
      PXU=GXU-GXX*GXU*HXX-(GXX*GYU+GXY*GXU)*HXY-GXY*GYU*HYY
      PXV=GXV-GXX*GXV*HXX-(GXX*GYV+GXY*GXV)*HXY-GXY*GYV*HYY
      PYU=GYU-GYY*GYU*HYY-(GYY*GXU+GXY*GYU)*HXY-GXY*GXU*HXX
      PYV=GYV-GYY*GYV*HYY-(GYY*GXV+GXY*GYV)*HXY-GXY*GXV*HXX
      PUU=PUU-GXU*GXU*HXX-2.*GXU*GYU*HXY-GYU*GYU*HYY
      PUV=PUV-GXU*GXV*HXX-(GXU*GYV+GYU*GXV)*HXY-GYU*GYV*HYY
      PVV=PVV-GXV*GXV*HXX-2.*GXV*GYV*HXY-GYV*GYV*HYY
      DET=RXX*RYY-RXY*RXY
      HXX=RYY/DET
      HXY=-RXY/DET
      HYY=RXX/DET
      XX(I)=XBAR+(PXX(I)*HXX+PXY(I)*HXY)*(ZX-XBAR)
      XX(I)=XX(I)+(PXX(I)*HXY+PXY(I)*HYY)*(ZY-YBAR)
      YY(I)=YBAR+(PXY(I)*HXX+PYY(I)*HXY)*(ZX-XBAR)
      YY(I)=YY(I)+(PXY(I)*HXY+PYY(I)*HYY)*(ZY-YBAR)
      U=U+(PXU*HXX+PYU*HXY)*(ZX-XBAR)+(PXU*HXY+PYU*HYY)*(ZY-YBAR)
      V=V+(PXV*HXX+PYV*HXY)*(ZX-XBAR)+(PXV*HXY+PYV*HYY)*(ZY-YBAR)
C
C                 UPDATE DRIVING NOISE ESTIMATE
C
      RI=FLOAT(I)
      CXX=CXX+(((ZX-XBAR)**2-GXX-RXX)/TAU-CXX)/RI
```

Fig. 3—Program listing (continued)

```
      CXY=CXY+(((ZX-XBAR)*(ZY-YBAR)-GXY-RXY)/TAU-CXY)/RI
      CYY=CYY+(((ZY-YBAR)**2-GYY-RYY)/TAU-CYY)/RI
      RK=CXY*(U*U+V*V)/U/V
      DXX=CXX
      DYY=CYY
      IF(DXX.LT.0.) DXX=0.
      IF(DYY.LT.0.) DYY=0.
      IF(RK.GT.DXX+DYY) RK=DXX+DYY
      IF(RK.LT.-DXX-DYY) RK=-DXX-DYY
      QXX=.5*(DXX+DYY+RK*(U*U-V*V)/(U*U+V*V))
      QXY=RK*U*V/(U*U+V*V)
1     QYY=QXX-RK*(U*U-V*V)/(U*U+V*V)
C
C                     OUTPUT
C
C        I = OBSERVATION INDEX
C        T = TIME
C        (XX, YY) = CURRENT POSITION ESTIMATE IN X-Y COORDINATES
C        SMAJ = SEMIMAJOR AXIS OF 86 PERCENT CONTAINMENT ELLIPSE FOR
C               CURRENT POSITION
C        SMIN = SEMIMINOR AXIS OF CONTAINMENT ELLIPSE
C        TH = ORIENTATION OF SEMIMAJOR AXIS (DEG. CLOCKWISE FROM Y-AXIS)
C
      DO 3 I=1,N
    3 PRINT 7,T(I),XX(I),YY(I),SMAJ(I),SMIN(I),TH(I)
    7 FORMAT(20X,3F10.2,5X,3F10.2/)
      PRINT 8
    8 FORMAT(20X///)
C
C                     TRACK SMOOTHER
C
      XS(N)=XX(N)
      YS(N)=YY(N)
      NM1=N-1
      DEN=T(N)-T(1)
      QXX=QXX+PUU*DEN
      QXY=QXY+PUV*DEN
      QYY=QYY+PVV*DEN
      DO 2 K=1,NM1
      I=N-K
      TAU=T(I+1)-T(I)
      P1=PXX(I)
      P2=PXY(I)
      P3=PYY(I)
      DEN=(P1+QXX*TAU)*(P3+QYY*TAU)-(P2+QXY*TAU)**2
      HXX=P1*(P3+QYY*TAU)-P2*(P2+QXY*TAU)
      HXY=P2*(P1+QXY*TAU)-P1*(P2+QXY*TAU)
      HYX=P2*(P3+QYY*TAU)-P3*(P2+QXY*TAU)
      HYY=P3*(P1+QXX*TAU)-P2*(P2+QXY*TAU)
      XS(I)=XX(I)+(HXX*(XS(I+1)-XX(I)-U*TAU)+HXY*(YS(I+1)-YY(I)-V*TAU))/
     1DEN
2     YS(I)=YY(I)+(HYX*(XS(I+1)-XX(I)-U*TAU)+HYY*(YS(I+1)-YY(I)-V*TAU))/
     1DEN
      SXX=PXX(N)
      SXY=PXY(N)
      SYY=PYY(N)
      C1=SXX+SYY
      C2=SQRT((SXX-SYY)**2+4.*SXY**2)
      C1=.5*(C1+C2)
```

Fig. 8—Program listing (continued)

```
      C2=C1-C2
      SMAJ(N)=2.*SQRT(C1)
      SMIN(N)=2.*SQRT(C2)
      TH(N)=57.3*ATAN((SXX-C1)/SXY)+90.
      DO 5 L=2,N
      I=N-L+1
      TAU=T(I+1)-T(I)
      P1=PXX(I)
      P2=PXY(I)
      P3=PYY(I)
      A1=P1+QXX*TAU
      A2=P2+QXY*TAU
      A3=P3+QYY*TAU
      DET=A1*A3-A2*A2
      H1=A3/DET
      H2=-A2/DET
      H3=A1/DET
      B1=P1*H1+P2*H2
      B2=P1*H2+P2*H3
      B3=P2*H1+P3*H2
      B4=P2*H2+P3*H3
      D1=SXX-A1
      D2=SXY-A2
      D3=SYY-A3
      E1=B1*D1+B2*D2
      E2=B1*D2+B2*D3
      E3=B3*D1+B4*D2
      E4=B3*D2+B4*D3
      SXX=P1+E1*B1+E2*B2
      SXY=P2+E1*B3+E2*B4
      SYY=P3+E3*B3+E4*B4
      C1=SXX+SYY
      C2=SQRT((SXX-SYY)**2+4.*SXY**2)
      C1=.5*(C1+C2)
      C2=C1-C2
      SMAJ(I)=2.*SQRT(C1)
      SMIN(I)=2.*SQRT(C2)
    5 TH(I)=57.3*ATAN((SXX-C1)/SXY)+90.
C
C                   OUTPUT
C
C        I = OBSERVATION INDEX
C        T = TIME
C        (XS, YS) = SMOOTHED POSITION IN X-Y COORDINATES
C        SMAJ = SEMIMAJOR AXIS OF 86 PERCENT CONTAINMENT ELLIPSE FOR
C               SMOOTHED POSITION
C        SMIN = SEMIMINOR AXIS OF CONTAINMENT ELLIPSE
C        TH = ORIENTATION OF SEMIMAJOR AXIS (DEG. CLOCKWISE FROM Y-AXIS)
C
      DO 4 I=1,N
    4 PRINT 7,T(I),XS(I),YS(I),SMAJ(I),SMIN(I),TH(I)
      END
```

Fig. 3—Program listing (continued)

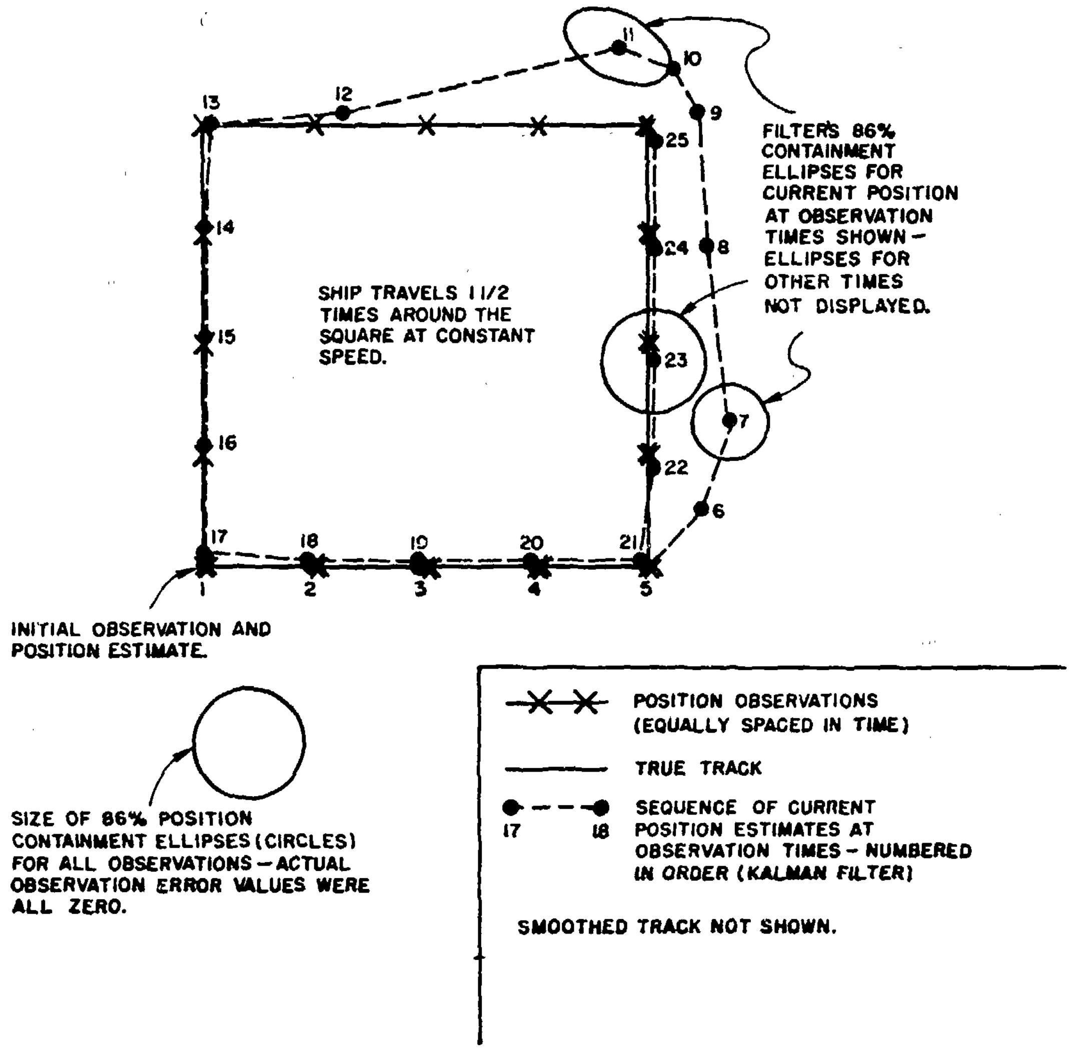

Fig. 4—Performance of recursive ship tracking algorithms on idealized track

• Beginning at time t_i^+ with the y axis oriented toward local north and the x axis toward local east, let ϕ_i and ψ_i be the latitude and longitude of the estimated position at that time. In these local rectangular coordinates, the velocity estimates $\hat{u}_i$, $\hat{v}_i$, the (4×4) state covariance matrix P_i, and the maneuvering matrix estimate $\hat{Q}_i$ are also available. For convenience here, denote south latitudes and west longitudes as negative.

• Propagating the ship's position by dead reckoning to that at the time t_{i+1} of the next observation, adopt a new coordinate system with origin at that position (ϕ_{i+1}, ψ_{i+1}) with y' and x' aligned along local north and east. These parameters are specified by the equations

$$\tau = t_{i+1} - t_i \tag{48}$$

$$f = \sqrt{\hat{u}_i^2 + \hat{v}_i^2} \tag{49}$$

$$\gamma = \frac{f\tau}{R_e}; \quad R_e = \text{earth radius} \tag{50}$$

$$\left\{ \begin{aligned} &\tilde{\phi}_{i+1} = \sin^{-1}\left[\sin\phi_i \cos\gamma + \frac{\hat{u}_i}{f}\cos\phi_i \sin\gamma\right] \\ &\tilde{\phi}_{i+1} \epsilon \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \end{aligned} \right\} \tag{51}$$

$$\left\{ \begin{aligned} &\sin\delta = \frac{\hat{u}_i \sin\gamma}{f\cos\tilde{\phi}_{i+1}} \\ &\cos\delta = \frac{\cos\phi_i\cos\gamma - \dfrac{\hat{v}_i}{f}\sin\phi_i\sin\gamma}{\cos\tilde{\phi}_{i+1}}. \end{aligned} \right\} \tag{52}$$

Take $\delta \epsilon [0, 2\Pi)$

$$\left\{ \begin{aligned} &\tilde{\psi}_{i+1} = \psi_i + \delta \\ &\text{if } \tilde{\psi}_{i+1} > \pi, \quad \tilde{\psi}_{i+1} = \tilde{\psi}_{i+1} - 2\pi. \end{aligned} \right\} \tag{53}$$

• Compute

$$\tilde{v}_{i+1} = \frac{\hat{v}_i \cos\phi_i \cos\gamma - f\sin\phi_i \sin\gamma}{f\cos\tilde{\phi}_{i+1}}$$

$$\tilde{u}_{i+1} = \hat{u}_i \frac{\cos \phi_i}{\cos \tilde{\phi}_{i+1}} .$$

These are the values of $\hat{v}, \hat{u}$ at time t_{i+1}^- in local rectangular coordinates.

- Define the rotation matrix θ_{i+1} as

$$\theta_{i+1} = \frac{1}{f^2} \left[\begin{array}{c|c} \hat{v}_i\tilde{v}_{i+1} + \hat{u}_i\tilde{u}_{i+1} & \hat{u}_i\tilde{v}_{i+1} - \hat{v}_i\tilde{u}_{i+1} \\ \hline \hat{v}_i\tilde{u}_{i+1} - \hat{u}_i\tilde{v}_{i+1} & \hat{v}_i\tilde{v}_{i+1} + \hat{u}_i\tilde{u}_{i+1} \end{array}\right]$$

$$= \left[\begin{array}{c|c} \frac{\hat{v}_i}{f} & \frac{\hat{u}_i}{f} \\ \hline \frac{-\hat{u}_i}{f} & \frac{\hat{v}_i}{f} \end{array}\right] \left[\begin{array}{c|c} \frac{\tilde{v}_{i+1}}{f} & \frac{-\tilde{u}_{i+1}}{f} \\ \hline \frac{\tilde{u}_{i+1}}{f} & \frac{\tilde{v}_{i+1}}{f} \end{array}\right] .$$

- Compute $\bar{M}_{i+1}$ from covariance propagation equation for rectangular coordinates, Eqs. (30) through (34) and (12) through (21) for this motion model.

- Rotate to updated coordinates:

$$M_{i+1} = \left[\begin{array}{c|c} \theta_{i+1} & 0 \\ \hline 0 & \theta_{i+1} \end{array}\right] \bar{M}_{i+1} \left[\begin{array}{c|c} \theta_{i+1}^T & 0 \\ \hline 0 & \theta_{i+1}^T \end{array}\right] .$$

- Set

$$\left.\begin{aligned} \hat{Q}_{i+1} &= \theta_{i+1}\hat{Q}_i\theta_{i+1}^T \\ \bar{x}_{i+1} &= 0 \\ \bar{y}_{i+1} &= 0 . \end{aligned}\right\} \text{values of } \hat{Q}, \hat{x}, \hat{y} \text{ at time } t_{i+1}^- \text{ in local coordinates}$$

- Compute with Eqs. (22) through (24) and (27) through (34)

$$\left.\begin{array}{l} \hat{x}_{i+1} \\ \hat{y}_{i+1} \\ \hat{u}_{i+1} \\ \hat{v}_{i+1} \\ P_{i+1} \\ \hat{Q}_{i+1} . \end{array}\right\} \text{estimated values at time } t_{i+1}^{+} \text{ in local rectangular coordinates}$$

- Compute latitude and longitude of (updated) position estimate at time t_{i+1}^{+}:

$$\phi_{i+1} = \tilde{\phi}_{i+1} + \frac{\hat{y}_{i+1}}{R_e}$$

$$\psi_{i+1} = \tilde{\psi}_{i+1} + \frac{\hat{x}_{i+1}}{R_e} \cos \tilde{\phi}_{i+1} .$$

— End of Procedure —

Track Smoothing

If the preceding track generation procedure has been followed through the Nth observation time, there are $N + 1$ observation times t_i, $i = 0, \ldots, N$, with corresponding Kalman filter estimates of latitude and longitude ϕ_i and ψ_i. The Bayesian smoother can be implemented in this context by computing "smoothed" offsets d_i to these position estimates, and their corresponding error covariance matrices K_i. These offsets are 2-vectors in linear distance units aligned with local north (i.e., the unit vector $\vec{e}_{yi}$ points north at ϕ_i, ψ_i). These offset and cov: 'ance matrices can be computed recursively as follows.

- Denote the current (at time t_N^{+}) positive semidefinite estimate of Q by Q_N and the estimate of the velocity vector

$$\begin{bmatrix} u \\ --- \\ v \end{bmatrix}$$

by u_N. These variables are all expressed in the local rectangular coordinate frame of observation time t_N.

- Set $d_N = 0$, $K_N = P_N$
- Loop through the following steps for $i = N-1, N-2, \ldots, 0$.

1. Rotate the offset vector d_{i+1} and its error covariance matrix K_{i+1} to the local coordinate frame at ϕ_i, ψ_i by computing

$$a_i = \theta_{i+1}^T d_{i+1}$$

and

$$L_i = \theta_{i+1}^T K_{i+1} \theta_{i+1}$$

where

a_i is the rotated offset vector

L_i is the rotated (2×2) covariance matrix

θ_{i+1} is the rotation matrix defined in the previous subsection for the Kalman filter (which could also be obtained from ϕ_i, ψ_i, ϕ_{i+1}, ψ_{i+1} to within the degree of approximation adopted here).

2. Rotate the velocity vector u_{i+1} to the ith local coordinate frame by computing

$$u_i = \theta_{i+1}^T u_{i+1} \; .$$

Compute $\tilde{\phi}_{i+1}$ and $\tilde{\psi}_{i+1}$ from ϕ_i and ψ_i with Eqs. (48) through (53), except that $\hat{u}_i$ and $\hat{v}_i$ are replaced by the components of u_i.

3. Let

$$b_i = \left[\begin{array}{c} R_e \cos \phi_{i+1} (\tilde{\psi}_{i+1} - \psi_{i+1}) \\ \hdashline R_e (\tilde{\phi}_{i+1} - \phi_{i+1}) \end{array} \right] .$$

4. Compute

$$\bar{M}_{i+1} = P_i + Q_i \tau_i \, , \quad \text{where} \quad Q_i = \theta_{i+1}^T Q_{i+1} \theta_{i+1} \quad \text{and} \quad \tau_i = t_{i+1} - t_i \, .$$

5. Compute

$$d_i = P_i \bar{M}_{i+1}^{-1} (a_i - b_i)$$

and

$$K_i = P_i + P_i \bar{M}_{i+1}^{-1} (L_i - \bar{M}_{i+1}) \bar{M}_{i+1}^{-1} P_i$$

which are the offset and the smoother's corresponding error covariance matrix for the ship position at the ith observation time, expressed in the ith local coordinate frame. The component-by-component details of this computation are not shown here, but are analogous to Eqs. (43) through (47).

— End of Loop —

If desired, the latitude and longitude $\bar{\phi}_i$ and $\bar{\psi}_i$ of the smoothed position estimate at time t_i can be computed as follows.

$$\bar{\phi}_i = \phi_i + \frac{d_i}{R_e \cos \phi_i}$$

$$\bar{\psi}_i = \psi_i + \frac{d_i}{R_e}.$$

TRACKING ON A SPHERE: RECTILINEAR COORDINATES

Ship tracking on a sphere is sometimes performed in terms of three-dimensional coordinates instead of geographical latitude and longitude. The use of such a coordinate system enables the computational effort to be reduced considerably, largely by the avoidance of trigonometric computations, at the expense of a slight increase in storage requirements. This section develops an adaptation of the above track generation and smoothing algorithms for a spherical earth in these rectilinear coordinates. This particular adaptation circumvents the potential problem of singular covariance matrices.

In this system a target's position on the earth's surface is described by a vector

$$\mathbf{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

whose components are the coordinates of the target's position in an earth-centered rectangular coordinate system. Specifically, the x axis is taken as intersecting the equator at zero longitude, the y axis as intersecting it at 90°E and the z axis as aligned along the North Pole (a right-handed coordinate system). Boldface lower case letters are used here to denote such 3-vectors.

Motion along a great-circle path is represented by a vector normal to the plane of the great circle. The magnitude of this vector is the *angular* speed of the motion and its sense is such that eastward motion along the equator is represented by a vector aligned with the North Pole. Hence, the motion of a target at position $\mathbf{r}$ and velocity $\mathbf{r}$ is represented by a vector

$$\boldsymbol{\omega} = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}$$

such that $\dot{\mathbf{r}} = \boldsymbol{\omega} \times \mathbf{r}$.

The track of a target is generated recursively from a sequence of noisy observations $\mathbf{z}_i$, $i = 0, 1, \ldots$, of the target's position $\mathbf{r}(t_i)$ in the rectilinear coordinate system. Also given with each observation $\mathbf{z}_i$ is a symmetric 2×2 covariance matrix R_i for the components of the observation error vector $\mathbf{z}_i - \mathbf{r}(t_i)$ in the directions of local east and local north. These components of $\mathbf{z}_i$ could easily be computed by the usual spherical coordinate formulas from a position report of geocentric latitude and longitude.

WARNING: Because of the earth's oblateness, the *geocentric* latitude differs from the more commonly used *geodetic* latitude (based on the orientation of the local horizon) by about 10 mi at middle latitudes.

Track Generation

For a particular target, the above-mentioned track generation algorithm can be implemented in this rectilinear coordinate system by computing and storing estimates of $\mathbf{r}$ and $\boldsymbol{\omega}$, a 4×4 symmetric covariance matrix P for the errors in this estimate, and an estimate of a 2×2 symmetric driving noise matrix Q after the receipt of each successive observation. The components of the P and Q matrices refer to the local east and local north components of the errors in position and velocity and the driving noise. For a time t between observation times, the target's position is estimated by dead reckoning (along a great circle path) from its estimated position and velocity at the time t_i of the last observation. Hence, using the circumflex to denote estimated values,

$$\hat{\mathbf{r}}(t) = \hat{\mathbf{r}}(t_i) \cos [(t - t_i)|\hat{\boldsymbol{\omega}}(t_i)|] + \frac{\hat{\boldsymbol{\omega}}(t_i) \times \hat{\mathbf{r}}(t_i)}{(t - t_i)|\hat{\boldsymbol{\omega}}(t_i)|} \sin [(t - t_i)|\hat{\boldsymbol{\omega}}(t_i)|]$$

$$\hat{\boldsymbol{\omega}}(t) = \hat{\boldsymbol{\omega}}(t_i)$$

If desired, the value of the P matrix can be computed for such an intermediate time as it is in the section "Tracking of Planar Motion." The x subscripts there correspond to local east components here, and the y subscripts correspond to local north components.

The item of major importance is the manner in which the estimates of $\mathbf{r}$, $\boldsymbol{\omega}$, and Q, and the covariance matrix P are updated immediately after each observation. We initiate this process immediately after the initial observation $\mathbf{z}_0$ by setting

$$\hat{\mathbf{r}}_0 = \mathbf{z}_0 \qquad \text{(3 vector)}$$

$$\hat{\boldsymbol{\omega}}_0 = 0 \qquad \text{(3 vector)}$$

$$Q_0 = 0 \qquad (2 \times 2 \text{ matrix})$$

$$P_0 = \left[\begin{array}{c|cc} R_0 & \multicolumn{2}{c}{0} \\ \hline 0 & \sigma^2 & 0 \\ & 0 & \sigma^2 \end{array}\right] \qquad (4 \times 4 \text{ matrix})$$

where $\hat{\mathbf{r}}_0$ denotes $\hat{\mathbf{r}}(t_0)$, etc., and σ^2 is an externally specified prior (*linear*) velocity component variance. The general updating step begins immediately after observation time t_i with the estimates $\hat{\mathbf{r}}_i$, $\hat{\omega}_i$, Q_i, and the covariance matrix P_i. The components of the matrices P_i and Q_i are denoted for convenience as follows:

$$P_i = \left[\begin{array}{c|c|c|c} p_{ee} & p_{en} & p_{e\dot{e}} & p_{e\dot{n}} \\ \hline & p_{nn} & p_{n\dot{e}} & p_{n\dot{n}} \\ \hline \text{Symmetric} & & p_{\dot{e}\dot{e}} & p_{\dot{e}\dot{n}} \\ \hline & & & p_{\dot{n}\dot{n}} \end{array}\right]; \quad Q_i = \left[\begin{array}{c|c} q_{ee} & q_{en} \\ \hline & q_{nn} \end{array}\right]$$

where e and n stand for local east and north, respectively. When the next observation $(\mathbf{z}_{i+1}, R_{i+1})$ is received at time t_{i+1}, the following procedure can be used to compute the updated quantities $\hat{\mathbf{r}}_{i+1}$, $\hat{\omega}_{i+1}$, Q_{i+1}, and P_{i+1}.

1. Compute the quantity

$$a_i = \sqrt{\hat{x}_i^2 + \hat{y}_i^2}\,.$$

It is then possible to express the components of the local east unit vector $\mathbf{e}$ and the local north unit vector $\mathbf{n}$ (at location $\hat{\mathbf{r}}_i$) as

$$\mathbf{e}_i = \begin{bmatrix} -\hat{y}_i/a_i \\ \hline \hat{x}_i/a_i \\ \hline 0 \end{bmatrix} \quad \text{and} \quad \mathbf{n}_i = \begin{bmatrix} \left(\dfrac{-\hat{x}_i\hat{z}_i}{a_iR_e}\right) \\ \hline \left(\dfrac{-\hat{y}_i\hat{z}_i}{a_iR_e}\right) \\ \hline \left(\dfrac{a_i}{R_e}\right) \end{bmatrix},$$

It is not necessary to compute these unit vector components separately, but they are used in deriving some of the following steps.

2. Compute the local east and local north components, $\hat{u}_i$ and $\hat{v}_i$, of the estimated velocity at t_i as $(\hat{\omega}_i \times \hat{\mathbf{r}}_i) \cdot \mathbf{e}_i$ and $(\hat{\omega}_i \times \hat{\mathbf{r}}_i) \cdot \mathbf{n}_i$; this yields

$$\hat{u}_i = a_i\hat{\gamma}_i - \frac{\hat{z}_i}{a_i}(\hat{\alpha}_i\hat{x}_i + \hat{\beta}_i\hat{y}_i)\,, \qquad \text{(east component)}$$

$$\hat{v}_i = \left(\frac{\hat{y}_i\hat{\alpha}_i - \hat{x}_i\hat{\beta}_i}{a_i}\right) R_e\,, \qquad \text{(north component)}\,.$$

3. Compute

$$\widetilde{\mathbf{r}}_{i+1} \quad \begin{bmatrix} \widetilde{x}_{i+1} \\ \hdashline \widetilde{y}_{i+1} \\ \hdashline \widetilde{z}_{i+1} \end{bmatrix}$$

such that

$$\widetilde{\mathbf{r}}_{i+1} = \widetilde{\mathbf{r}}_i \cos(\tau b_i) + \frac{(\hat{\omega}_i \times \hat{\mathbf{r}}_i)}{\tau b_i} \sin(\tau b_i)$$

where

$$b_i = \sqrt{\hat{\alpha}_i^2 + \hat{\beta}_i^2 + \hat{\gamma}_i^2} = |\hat{\omega}_i|$$

$$\tau = t_{i+1} - t_i .$$

This is position propagation by dead reckoning along a great-circle path. As a simplifying approximation, one could compute instead

$$\widetilde{\mathbf{r}}_{i+1} = \hat{\mathbf{r}}_i \left(1 - \frac{\tau^2}{2}\, \hat{\omega}_i \cdot \hat{\omega}_i\right) + (\hat{\omega}_i \times \hat{\mathbf{r}}_i) .$$

4. Compute the new value

$$a_{i+1} = \sqrt{\widetilde{x}_{i+1}^2 + \widetilde{y}_{i+1}^2} .$$

5. Compute the new local east and local north velocity components as in step 2, giving

$$\widetilde{u}_{i+1} = a_{i+1}\hat{\gamma}_i - \frac{\hat{z}_{i+1}}{a_{i+1}} (\hat{\alpha}_i \hat{x}_{i+1} - \hat{\beta}_i \hat{\gamma}_{i+1})$$

$$\widetilde{v}_{i+1} = \left(\frac{\widetilde{y}_{i+1}\hat{\alpha}_i - \widetilde{x}_{i+1}\hat{\beta}_i}{a_{i+1}}\right) R_e .$$

6. Compute the rotation matrix θ_{i+1} such that

$$\theta_{i+1} = \frac{1}{b_i^2} \left[\begin{array}{c:c} \hat{v}_i\widetilde{v}_{i+1} + \hat{u}_i\widetilde{u}_{i+1} & \hat{u}_i\widetilde{v}_{i+1} - \hat{v}_i\widetilde{u}_{i+1} \\ \hdashline \hat{v}_i\widetilde{u}_{i+1} - \hat{u}_i\widetilde{v}_{i+1} & \hat{v}_i\widetilde{v}_{i+1} + \hat{u}_i\widetilde{u}_{i+1} \end{array}\right] .$$

7. Compute a positive semidefinite approximation to Q_i; denote it by

$$Q_i^* = \left[\begin{array}{c|c} q_{ee}^* & q_{en}^* \\ \hline q_{en}^* & q_{nn}^* \end{array}\right].$$

One reasonable method of doing this is described in the section "Tracking of Planar Motion."

8. Compute the symmetric 4×4 matrix $\bar{M}$ such that

$$\bar{M} = \left[\begin{array}{c|c|c|c} \bar{m}_{ee} & m_{en} & \bar{m}_{e\dot{e}} & \bar{m}_{e\dot{n}} \\ \hline & \bar{m}_{nn} & \bar{m}_{n\dot{e}} & \bar{m}_{n\dot{n}} \\ \hline \text{Symmetric} & & p_{\dot{e}\dot{e}} & p_{\dot{e}\dot{n}} \\ \hline & & & p_{\dot{n}\dot{n}} \end{array}\right]$$

and

$$\bar{m}_{ee} = p_{ee} + 2p_{e\dot{e}}\tau^2 + p_{\dot{e}\dot{e}}\tau^2 + q_{ee}^*\tau$$

$$\bar{m}_{en} = p_{en} + (p_{e\dot{n}} + p_{n\dot{e}})\tau + p_{\dot{e}\dot{n}}\tau^2 + q_{en}^*\tau$$

$$\bar{m}_{nn} = p_{nn} + 2p_{n\dot{n}}\tau + p_{\dot{n}\dot{n}}\tau^2 + q_{nn}^*\tau$$

$$\bar{m}_{e\dot{e}} = p_{e\dot{e}} + p_{\dot{e}\dot{e}}\tau$$

$$\bar{m}_{e\dot{n}} = p_{e\dot{n}} + p_{\dot{e}\dot{n}}\tau$$

$$\bar{m}_{n\dot{e}} = p_{n\dot{e}} + p_{\dot{e}\dot{n}}\tau$$

$$\bar{m}_{n\dot{n}} = p_{n\dot{n}} + p_{\dot{n}\dot{n}}\tau .$$

9. Compute the rotated matrices

$$M = \left[\begin{array}{c|c} \theta_{i+1} & 0 \\ \hline 0 & \theta_{i+1} \end{array}\right] \bar{M} \left[\begin{array}{c|c} \theta_{i+1}^T & 0 \\ \hline 0 & \theta_{i+1}^T \end{array}\right] \qquad (4 \times 4)$$

and

$$\bar{Q}_{i+1} = \theta_{i+1} Q_i \theta_{i+1}^T . \qquad (2 \times 2)$$

Denote their components with analogous subscripts.

10. Compute the local east and local north components ϵ_e and ϵ_n of the residual vector

$$(\mathbf{z}_{i+1} - \widetilde{\mathbf{r}}_{i+1}) = \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \end{bmatrix}$$

by taking dot products with the unit vectors $\mathbf{e}_{i+1}$ and $\mathbf{n}_{i+1}$ (not shown) used in step 2, which gives

$$\epsilon_e = \frac{1}{a_{i+1}} (\widetilde{x}_{i+1}\epsilon_y - \widetilde{y}_{i+1}\epsilon_x)$$

and

$$\epsilon_n = \frac{1}{a_{i+1}R_e} \left(-\epsilon_x\widetilde{z}_{i+1}\widetilde{x}_{i+1} - \epsilon_y\widetilde{y}_{i+1}\widetilde{z}_{i+1} + \epsilon_z a_{i+1}^2\right) .$$

11. Compute the new state covariance matrix P_{i+1} as

$$P_{i+1} \triangleq \left[\begin{array}{c|c|c|c} p_{ee} & p_{en} & p_{e\dot{e}} & p_{e\dot{n}} \\ \hline p_{en} & p_{nn} & p_{n\dot{e}} & p_{n\dot{n}} \\ \hline p_{e\dot{e}} & p_{n\dot{e}} & p_{\dot{e}\dot{e}} & p_{\dot{e}\dot{n}} \\ \hline p_{e\dot{n}} & p_{n\dot{n}} & p_{\dot{e}\dot{n}} & p_{\dot{n}\dot{n}} \end{array}\right]_{i+1} = M_{i+1} - \left[\begin{array}{c|c} m_{ee} & m_{en} \\ \hline m_{en} & m_{nn} \\ \hline m_{e\dot{e}} & m_{n\dot{e}} \\ \hline m_{e\dot{n}} & m_{n\dot{n}} \end{array}\right]$$

$$\times \left(\left[\begin{array}{c|c} m_{ee} & m_{en} \\ \hline m_{en} & m_{nn} \end{array}\right] + R_{i+1} \right)^{-1} \left[\begin{array}{c|c|c|c} m_{ee} & m_{en} & m_{e\dot{e}} & m_{e\dot{n}} \\ \hline m_{en} & m_{nn} & m_{n\dot{e}} & m_{n\dot{n}} \end{array}\right] .$$

12. Compute Kalman filter correction terms δ in local east and local north coordinates:

$$\left[\begin{array}{c} \delta_e \\ \hline \delta_n \\ \hline \delta_{\dot{e}} \\ \hline \delta_{\dot{n}} \end{array}\right] = \left[\begin{array}{c|c} p_{ee} & p_{en} \\ \hline p_{en} & p_{nn} \\ \hline p_{e\dot{e}} & p_{n\dot{e}} \\ \hline p_{e\dot{n}} & p_{n\dot{n}} \end{array}\right]_{i+1} R_{i+1}^{-1} \left[\begin{array}{c} \epsilon_e \\ \hline \epsilon_n \end{array}\right] .$$

13. Use the Kalman filter correction terms to compute the new estimates $\hat{\mathbf{r}}_{i+1}$ and $\hat{\omega}_{i+1}$, the components of which are

$$\hat{x}_{i+1} = -\frac{1}{a_{i+1}} \left[\tilde{y}_{i+1}\delta_e + \frac{1}{R_e} (\tilde{x}_{i+1}\tilde{z}_{i+1}\delta_n) \right] + \tilde{x}_{i+1}$$

$$\hat{y}_{i+1} = \frac{1}{a_{i+1}} \left[\tilde{x}_{i+1}\delta_e - \frac{1}{R_e} (\tilde{y}_{i+1}\tilde{z}_{i+1}\delta_n) \right] + \tilde{y}_{i+1}$$

$$\hat{z}_{i+1} = \frac{a_{i+1}}{R_e} \delta_n + \tilde{z}_{i+1}$$

$$\hat{\alpha}_{i+1} = \hat{\alpha}_i + \frac{1}{R_e a_{i+1}} \left[\tilde{y}_{i+1}\delta_{\dot{n}} - \frac{1}{R_e} (\tilde{x}_{i+1}\tilde{z}_{i+1}\delta_{\dot{e}}) \right]$$

$$\hat{\beta}_{i+1} = \hat{\beta}_i - \frac{1}{R_e a_{i+1}} \left[\tilde{x}_{i+1}\delta_{\dot{n}} - \frac{1}{R_e} (\tilde{y}_{i+1}\tilde{z}_{i+1}\delta_{\dot{e}}) \right]$$

$$\hat{\gamma}_{i+1} = \hat{\gamma}_i + \frac{a_{i+1}}{R_e^2} \delta_{\dot{e}} .$$

Also, it might be wise to renormalize $\hat{\mathbf{r}}_{i+1}$ so that its magnitude is R_e to prevent an accumulation of roundoff errors.

14. Compute the new estimate of the driving noise matrix Q_{i+1}, as $Q_{i+1} = 0$ if $i = 0$, otherwise as

$$Q_{i+1} = \bar{Q}_{i+1} + \frac{1}{i+1} \left(\frac{1}{\tau} \left[\begin{array}{c|c} \epsilon_e^2 - m_{ee} & \epsilon_e\epsilon_n - m_{en} \\ \hline \epsilon_e\epsilon_n - m_{en} & \epsilon_n^2 - m_{nn} \end{array} \right] - \frac{1}{\tau} R_{i+1} - \bar{Q}_{i+1} \right).$$

— End of Update Cycle —

Simplifying Approximations

It might be possible to reduce the computation required to implement this tracking algorithm with little sacrifice in accuracy by adopting some approximations. In addition to that mentioned in step 3 for generating $\tilde{\mathbf{r}}_{i+1}$ from $\hat{\mathbf{r}}_i$ and $\hat{\omega}_i$, another likely approximation is to neglect the rotation of local east and local north between successive observations. With this approach it is possible to omit steps 1, 2, 6, and 9 in the update cycle by the use of $M_{i+1} = \bar{M}_{i+1}$ in step 11 and $\bar{Q}_{i+1} = Q_i$ in step 14.

Track Smoothing

The basic approach here is the same as in pages 27 to 29. The smoother is implemented as a backward recursion to compute offsets d_i to the Kalman filter position estimates $\hat{\mathbf{r}}_i$ and also the smoothed error covariance matrices K_i corresponding to the d_i. The offsets are computed in local east and north coordinates; thus the d_i are 2-vectors and the K_i are 2×2 matrices. If smoothing is to take place immediately after time t_N, it is assumed that the Kalman filter outputs $\hat{\omega}_N$, Q_N^*, $\hat{\mathbf{r}}_i$, and P_i, $i = 0, \ldots, N$, are all available. The procedure can be implemented as described below.

- Set

$$d_N = 0$$

$$K_N = P_N$$

$$Q_N = Q_N^* .$$

- Loop through the following steps for $i = N-1, N-2, \ldots, 0$.

1. Follow steps 1 through 5 of the filter update cycle of the preceding subsection, except with $\hat{\omega}_i$ replaced by $\hat{\omega}_N$ to compute $\theta_{i+1}, \tilde{x}_{i+1}, \tilde{y}_{i+1}, \tilde{z}_{i+1}$, $\mathbf{e}_i$ and $\mathbf{n}_i$.

2. Rotate the offset vector d_{i+1} and covariance matrix K_{i+1} to the ith local Cartesian coordinate frame:

$$a_i = \theta_{i+1}^T d_{i+1} \qquad \text{(offset 2-vector)}$$

$$L_i = \theta_{i+1}^T K_{i+1} \theta_{i+1} \quad (2 \times 2 \text{ covariance matrix}) .$$

3. Compute the local east and north components of $\hat{\mathbf{r}}_{i+1} - \bar{\mathbf{r}}_{i+1}$ in the ith coordinate frame:

$$\epsilon_e = (\hat{\mathbf{r}}_{i+1} - \bar{\mathbf{r}}_{i+1}) \cdot \mathbf{e}_i$$

$$\epsilon_n = (\hat{\mathbf{r}}_{i+1} - \bar{\mathbf{r}}_{i+1}) \cdot \mathbf{n}_i .$$

4. Compute

$$\bar{M}_{i+1} = P_i + Q_i \tau_i ,$$

where

$$Q_i = \theta_{i+1}^T Q_{i+1} \theta_{i+1}$$

$$\tau_i = t_{i+1} - t_i .$$

5. Compute the new offset and covariance matrix:

$$d_i = P_i \bar{M}_{i+1}^{-1} \left(\begin{bmatrix} \epsilon_e \\ -- \\ \epsilon_n \end{bmatrix} + a_i \right)$$

$$K_i = P_i + P_i \bar{M}_{i+1}^{-1} (L_i - \bar{M}_{i+1}) \bar{M}_{i+1}^{-1} P_i .$$

— End of Loop —

At this point, one could also compute the smoothed position vectors $\bar{\mathbf{r}}_i$ as

$$\bar{\mathbf{r}}_i = \hat{\mathbf{r}}_i + [\mathbf{e}_i \mid \mathbf{n}_i] \, d_i ,$$

where

$[\mathbf{e}_i \mid \mathbf{n}_i]$ denotes a 3×2 matrix.

Appendix A

GENERAL RECURSIVE FILTER AND SMOOTHING ALGORITHMS

Suppose that x is a state vector describing a ship's motion in the context of Eqs. (1), (2), and (3), and that the corresponding Kalman filter is being used to track the ship. The conditional moments η and K can be computed recursively from the data by means of the following equations.

Forward Equations (Kalman Filter for Eqs. (1) Through (3))

$$\left.\begin{aligned} \dot{\hat{x}} &= F\hat{x} + \bar{w} \qquad &\text{(A1)} \\ \dot{P} &= FP + PF^T + Q \qquad &\text{(A2)} \end{aligned}\right\} \text{between observations}$$

$$\left.\begin{aligned} P(t_i^+) &= [P^{-1}(t_i^-) + H_i^T R_i^{-1} H_i]^{-1} ; \; P(t_0^-) = M_0 \qquad &\text{(A3a)} \\ &= P(t_i^-) - H_i^T [H_i P(t_i^-) H_i^T + R_i]^{-1} H_i P(t_i^-) \qquad &\text{(A3b)} \\ \hat{x}(t_i^+) &= \hat{x}(t_i^-) + P(t_i^+) H_i^T R_i^{-1} [z_i - H_i \hat{x}(t_i^-)] ; \; \hat{x}(t_0^-) = \bar{x}_0 \,. \qquad &\text{(A4)} \end{aligned}\right\} \text{at observation time } t_i$$

For the "flat prior" case, use $P^{-1}(t_0^-) = 0$ in Eq. (A3a).

Backward Equations (Bayesian Smoother for Eqs. (1) Through (3))

$$\left.\begin{aligned} \dot{\hat{x}} &= F\hat{x} + \bar{w} \qquad &\text{(A5)} \\ \dot{P} &= FP + PF^T + Q \qquad &\text{(A6)} \end{aligned}\right\} \text{between observations}$$

$$\left.\begin{aligned} P(t_i^-) &= [P^{-1}(t_i^+) - H_i^T R_i^{-1} H_i]^{-1} \qquad &\text{(A7a)} \\ &= P(t_i^+) + P(t_i^+) H_i^T [R_i - H_i P(t_i^+) H_i^T]^{-1} H P(t_i^+) \qquad &\text{(A7b)} \\ \hat{x}(t_i^-) &= \hat{x}(t_i^+) - P(t_i^-) H_i^T R_i^{-1} [z_i - H_i \hat{x}(t_i^+)] \qquad &\text{(A8)} \end{aligned}\right\} \text{at observation time } t_i$$

$$\left.\begin{aligned} \frac{\partial\eta(T,t)}{\partial t} &= \dot{\eta}(T,t) = F\eta + \bar{w} + QP^{-1}[\eta(T,t) - \hat{x}]\,;\ \eta(T,T) = \hat{x}(T) \qquad \text{(A9)} \\ \dot{K}(T,t) &= [F + QP^{-1}]K(T,t) + K(T,t)[F^T + P^{-1}Q] - Q\,; \\ K(T,T) &= P(T) \qquad \text{(A10)} \end{aligned}\right\} \text{always}.$$

The Kalman filter statistics $\hat{x}(t)$ and $P(t)$ are computed using only the forward equations from time t_0 to time t. The moments $\eta(T, t)$ and $K(T, t)$, for $T > t$, are computed by first obtaining $\hat{x}(T)$ and $P(T)$ using the forward equations from time t_0 to time T. Then these values are used as boundary conditions at time T for the backward equations, which are then used recursively from time T to time t to obtain $\eta(T, t)$ and $K(T, t)$.

These results are developed in greater detail in Bryson and Ho.*

*A. E. Bryson and Y. C. Ho, *Applied Optimal Control*, Blaisdell, Waltham, Mass., ——1968.

Appendix B

INCLUSION OF OCCASIONAL VELOCITY OBSERVATIONS

In the context of the planar motion of the section "Tracking of Planar Motion," suppose that a noisy, independent measurement of the ship's velocity is obtained at observation time t_i in addition to the position measurement z_i. We assume that this velocity measurement can be adequately described as

$$\zeta_i = s(t_i) + \sigma_i \tag{B1}$$

where

ζ_i = measured velocity (2-vector)

s = actual instantaneous velocity in x-y coordinates (2-vector)

σ_i = zero-mean, Bivariate Normal, random variable with covariance matrix Σ_i.

Unfortunately, the ship motion model (pages 4 and 5) is an inadequate approximation to the actual motion for the purpose of dealing with such velocity observations because an instantaneous velocity cannot be defined for the Brownian motion component. The model can be refined for this purpose, however, by specifying a probability distribution for the difference between the average velocity and the observed, instantaneous velocity. One convenient way of doing this is to assume that these differences are statistically independent for different observation times and distributed with a zero-mean Bivariate Normal distribution with covariance matrix D_i at observation time t_i. As a default procedure to avoid specifying D_i externally, it might be reasonable to use

$$D_i = \frac{i}{t_i - t_0}\left[\begin{array}{c|c} q_{xx} & q_{xy} \\ \hline q_{xy} & q_{yy} \end{array}\right],$$

where q_{xx}, q_{xy}, and q_{yy} are as defined in Eqs. (30) through (34). This procedure is based on matching the variances of the observed fluctuations in position.

With this refinement, it follows from Eqs. (4) and (B1) that the velocity measurement ζ_i can be expressed as

$$\zeta_i = \begin{bmatrix} u \\ \hline v \end{bmatrix} + \left(s(t_i) - \begin{bmatrix} u \\ \hline v \end{bmatrix} + \sigma_i \right), \tag{B2}$$

the latter term in parentheses being a Bivariate Normal random variable with mean zero and covariance matrix $(\Sigma_i + D_i)$. Therefore, the composite observation vector

$$\begin{bmatrix} z_i \\ \hdashline \zeta_i \end{bmatrix}$$

can be regarded as a noisy observation of the state vector

$$\begin{bmatrix} x \\ \hdashline y \\ \hdashline u \\ \hdashline v \end{bmatrix}$$

at time t_i in the context of Appendix A. The (4×4) covariance matrix of the observation noise η_i in this context is

$$\left[\begin{array}{c:c} R_i & 0 \\ \hdashline 0 & \Sigma_i + D_i \end{array}\right].$$

Specializing the results of Appendix A to this particular case shows that the track generation procedure (pages 6 to 14) applies here also, except that Eqs. (22) through (24) are replaced respectively by the following three equations whenever a velocity measurement ζ_{i+1} is obtained at time t_{i+1}:

$$\begin{bmatrix} \hat{x}_{i+1} \\ \hdashline \hat{y}_{i+1} \end{bmatrix} = \begin{bmatrix} \hat{x}_i \\ \hdashline \hat{y}_i \end{bmatrix} + \tau \begin{bmatrix} \hat{u}_i \\ \hdashline \hat{v}_i \end{bmatrix} + \left[\begin{array}{c:c} p_{xx} & p_{xy} \\ \hdashline p_{xy} & p_{yy} \end{array}\right]_{i+1} R_{i+1}^{-1} \begin{bmatrix} z_x(i+1) - \hat{x}_i - \tau\hat{u}_i \\ \hdashline z_y(i+1) - \hat{y}_i - \tau\hat{v}_i \end{bmatrix}$$

$$+ \left[\begin{array}{c:c} p_{xu} & p_{xv} \\ \hdashline p_{yu} & p_{yv} \end{array}\right]_{i+1} \cdot (\Sigma_{i+1} + D_{i+1})^{-1} \left(\zeta_{i-1} - \begin{bmatrix} \hat{u}_i \\ \hdashline \hat{v}_i \end{bmatrix} \right) \quad \text{(B3)}$$

$$\left[\begin{array}{c}\hat{u}_{i+1}\\ \hline \hat{v}_{i+1}\end{array}\right] = \left[\begin{array}{c}\hat{u}_i\\ \hline \hat{v}_i\end{array}\right] + \left[\begin{array}{c|c}p_{xu} & p_{yu}\\ \hline p_{xv} & p_{yv}\end{array}\right]_{i+1} R_{i+1}^{-1} \left[\begin{array}{c}z_x(i+1) - \hat{x}_i - \tau\hat{u}_i\\ \hline z_y(i+1) - \hat{y}_i - \tau\hat{v}_i\end{array}\right]$$

$$+ \left[\begin{array}{c|c}p_{uu} & p_{uv}\\ \hline p_{uv} & p_{vv}\end{array}\right]_{i+1} (\Sigma_{i+1} + D_{i+1})^{-1}\left(\zeta_{i+1} - \left[\begin{array}{c}\hat{u}_i\\ \hline \hat{v}_i\end{array}\right]\right) \tag{B4}$$

$$P_{i+1} = M - M\left(M + \left[\begin{array}{c|c}R_{i+1} & 0\\ \hline 0 & \Sigma_{i+1} + D_{i+1}\end{array}\right]\right)^{-1} M. \tag{B5}$$

The M matrix components are as defined in Eqs. (12) through (21). Since the track propagation steps are unchanged, there are no additional modifications for velocity observations in the spherical track propagation algorithms because the track updating steps are performed in Cartesian coordinates there also.

The results of Appendix A can also be specialized to this case for track smoothing. It would be necessary to retain all four components of the state vector for this purpose because velocity observations must be taken directly into account. This would be a drastic departure from the other smoothing algorithms developed in this report, which are based on a two-component, position-only, state vector approximation. Therefore the details of this four-dimensional smoother are not developed here. A two-dimensional smoother can always be used, anyway, by ignoring the velocity measurements.

Appendix C

DEVELOPMENT OF CONSTRAINT ON MANEUVERING MATRIX ESTIMATE

A procedure is described in Eqs. (30) through (34) for modifying the estimates $\hat{q}_{xx}(i)$, $\hat{q}_{xy}(i)$, and $\hat{q}_{yy}(i)$ of the maneuvering matrix components to ensure that the resulting estimates form a positive semidefinite matrix whose principal axes are aligned with the current estimate

$$\begin{bmatrix} \hat{u}_i \\ \hline \hat{v}_i \end{bmatrix}$$

of the average velocity vector. The justification for this particular procedure is discussed below.

Suppose that the estimated average velocity vector is as shown in Fig. C1, where the i and c axes are in-track and cross-track coordinates. From this figure, the formulas for rotation of coordinates give

$$\begin{bmatrix} x \\ \hline y \end{bmatrix} = \left[\begin{array}{c|c} \cos\theta & -\sin\theta \\ \hline \sin\theta & \cos\theta \end{array}\right] \begin{bmatrix} i \\ \hline c \end{bmatrix},$$

from which

$$\cos^2\theta = \frac{\hat{u}_i^2}{\hat{u}_i^2 + \hat{v}_i^2}, \tag{C1}$$

$$\sin^2\theta = \frac{\hat{v}_i^2}{\hat{u}_i^2 + \hat{v}_i^2}, \tag{C2}$$

and

$$\sin\theta\cos\theta = \frac{\hat{u}_i\hat{v}_i}{\hat{u}_i^2 + \hat{v}_i^2}. \tag{C3}$$

For a maneuvering noise covariance matrix which has the diagonal form

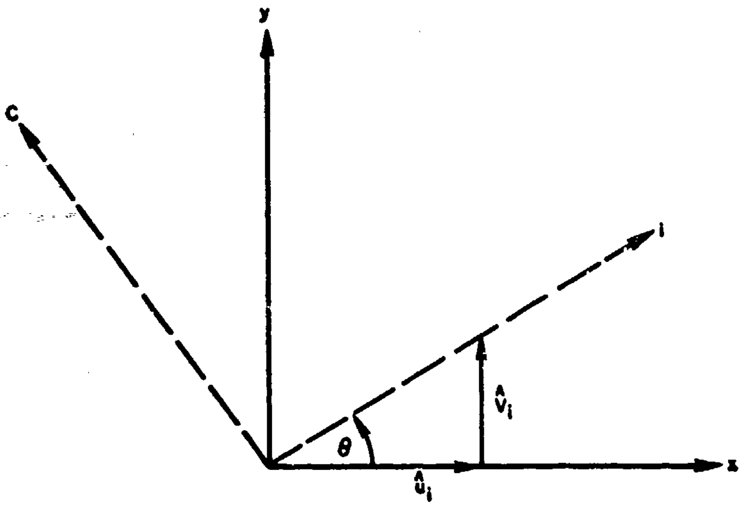

Fig. C1—Coordinate rotation

$$\left[\begin{array}{c|c} q_i & 0 \\ \hline 0 & q_c \end{array}\right] \tag{C4}$$

in the *i*, *c* coordinate system, the corresponding covariance matrix in *x*, *y* coordinates is, by standard formulas for linear coordinate transformations,

$$\left[\begin{array}{c|c} q_i \cos^2\theta + q_c \sin^2\theta & (q_i - q_c)\sin\theta\cos\theta \\ \hline (q_i - q_c)\sin\theta\cos\theta & q_i \sin^2\theta + q_c \cos^2\theta \end{array}\right]. \tag{C5}$$

If this is to be equal to an observed symmetric matrix

$$\left[\begin{array}{c|c} \hat{q}_{xx} & \hat{q}_{xy} \\ \hline \hat{q}_{xy} & \hat{q}_{yy} \end{array}\right],$$

then

$$q_i = \frac{\hat{q}_{xy}}{\cos\theta \sin\theta} + q_c .$$

Therefore,

$$q_i = \hat{q}_{yy} - \frac{\sin\theta}{\cos\theta}\,\hat{q}_{xy} = \hat{q}_{xx} + \frac{\sin\theta}{\cos\theta}\,\hat{q}_{xy} - \frac{\hat{q}_{xy}}{\sin\theta\cos\theta}$$

and

$$q_c = \hat{q}_{xx} + \frac{\sin\theta}{\cos\theta}\,\hat{q}_{xy} = \hat{q}_{yy} - \frac{\sin\theta}{\cos\theta}\,\hat{q}_{xy} + \frac{\hat{q}_{xy}}{\sin\theta\cos},$$

from which we get

$$q_i = \frac{1}{2}\left[\hat{q}_{xx} + \hat{q}_{yy} + \frac{\hat{q}_{xy}}{\sin\theta\cos\theta}\right] \tag{C6}$$

and

$$q_c = \frac{1}{2}\left[\hat{q}_{xx} + \hat{q}_{yy} - \frac{\hat{q}_{xy}}{\sin\theta\cos\theta}\right]. \tag{C7}$$

For arbitrary $\hat{q}_{xx}$, $\hat{q}_{xy}$, and $\hat{q}_{yy}$, Eqs. (C1) through (C3) can be used in modifying Eqs. (C6) and (C7) as follows so that the diagonal matrix (C4) is always positive semidefinite:

$$q_i = \frac{1}{2}(\xi + \lambda)$$

$$q_c = \frac{1}{2}(\xi - \lambda),$$

where

$$\xi = \max\{0,\ (\hat{q}_{xx} + \hat{q}_{yy})\} \tag{C8}$$

$$\lambda = \begin{cases} \xi & \text{if} \quad \dfrac{\hat{q}_{xy}\left(\hat{u}_i^2 + \hat{v}_i^2\right)}{\hat{u}_i\hat{v}_i} > \xi \\[2ex] -\xi & \text{if} \quad \dfrac{\hat{q}_{xy}\left(u_i^2 + v_i^2\right)}{\hat{u}_i v_i} < -\xi \\[2ex] \hat{q}_{xy}\,\dfrac{\hat{u}_i^2 + \hat{v}_i^2}{\hat{u}_i\hat{v}_i} & \text{otherwise}. \end{cases} \tag{C9}$$

Using Eqs. (C1) through (C3) in covariance matrix (C5) for this modified matrix in x, y coordinates gives

$$\hat{q}_{xx} = \frac{1}{2}\left[\xi + \frac{\hat{u}_i^2 - \hat{v}_i^2}{\hat{u}_i^2 + \hat{v}_i^2}\lambda\right]$$

$$\hat{q}_{xy} = \frac{\hat{u}_i\hat{v}_i}{\hat{u}_i^2 + \hat{v}_i^2}\lambda$$

$$\hat{q}_{xy} = \frac{1}{2}\left[\xi - \frac{\hat{u}_i^2 - \hat{v}_i^2}{\hat{u}_i^2 + \hat{v}_i^2}\lambda\right],$$

where ξ and λ are defined by Eqs. (C8) and (C9).